SAPIENZA
UNIVERSITÀ DI ROMA

Enhancing Single Image Super-Resolution via Deep Learning: Stability, Perceptual Quality, and Multi-Scale Modeling

Dipartimento di Ingegneria dell'Informazione, Elettronica e Telecomunicazioni (DIET)
Dottorato in Tecnologie Dell'Informazione e Delle Comunicazioni (ICT) (XXXVIII cycle)

Alireza Momenzadeh
ID number 1942818

Advisor
Prof. Enzo Baccarelli

Academic Year 2025/2026

Thesis defended on June 4, 2026
in front of a Board of Examiners composed by:

Prof. Stefano Buzzi
Prof. Riccardo Berta
Prof. Fulvio Schettino (chairman)

Enhancing Single Image Super-Resolution via Deep Learning: Stability, Perceptual Quality, and Multi-Scale Modeling
PhD thesis. Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: alireza.momenzadeh@uniroma1.it

Dedicated to
Sima, my dear wife

Abstract

The goal of Single Image Super-Resolution (SISR) is to reconstruct a high-resolution (HR) image from a single low-resolution (LR) observation. This task is ill-posed: the formation of LR image removes high-frequency information, so multiple distinct HR images can map to the same LR input. In medical imaging, this ambiguity becomes severe because hallucinated structures or reconstruction artifacts can mislead diagnosis. Generative models such as Generative adversarial Networks (GANs) and Diffusion Models (DMs) can produce visually sharp super-resolved images, however, they typically introduce non-deterministic textures, artifacts, or hallucinations. In addition, GAN training is very unstable and sensitive to hyperparameters, and DMs require many iterative denoising steps that makes the inference computationally expensive.

Motivated by these constraints, this thesis investigates how to bridge the perceptual-quality gap between regression-based Super-Resolution (SR) and generative methods, while preserving the stability, determinism, and artifact-awareness that are needed in medical applications. The main idea is that regression-based models can be improved by (i) architectural designs that enhance representation learning, (ii) a constructed set of perceptual and structural losses that target texture, edges, and frequency content, and (iii) stabilization of convolutional layers that are inspired by weight scaling techniques.

The proposed contributions are threefold:

First, we develop Twinned Residual Auto-Encoder architecture (TRAE) for denoising SR, including a multi-resolution extension that produces consistent reconstructions over multiple upscaling factors: Multi-Resolution Twinned Residual Auto-Encoder architecture (MR-TRAE).

Second, we introduce a perceptual SR model based on Convolutional Neural Network (CNN) backbones that is trained without GANs' adversarial losses but equipped with a set of losses. We combine a robust Charbonnier content loss with feature-based perceptual loss, gradient/edge preservation, frequency-domain alignment using masked Fourier magnitudes, and Gram-matrix style/texture matching.

Third, to mimic realistic clinical degradation, we use a stochastic LR synthesis model that includes probabilistic blur (Gaussian kernels with varying variance and kernel size), diverse resampling operators, and additive Gaussian noise, to reduce sensitivity to idealized bicubic assumptions and improving robustness.

Our experiments on medical Computed Tomography (CT) imagery (including COVIDx CT-2A) show that the proposed regression-based models improve perceptual fidelity and structural consistency while maintaining stable training. Quantitative comparisons with representative baselines such as EDSR and SRGAN models confirm competitive reconstruction quality, with strong structural similarity behavior that is aligned with the avoidance of artifact required by medical imaging.

Acknowledgments

I am deeply grateful to professor Enzo Baccarelli for his expert and patient guidance, thoughtful feedback, and continuous support throughout the research.

Contents

List of Figures

List of Tables

List of Acronyms

AE Auto-Encoder.

BN Batch Normalization.

cGAN Conditional Generative Adversarial Network.

CNN Convolutional Neural Network.

CP Common Pneumonia.

CT Computed Tomography.

DDPM Denoising Diffusion Probabilistic Model.

DL Deep Learning.

DM Diffusion Model.

EDSR Enhanced Deep Super-Resolution Network.

FLOP Floating Point Operation.

GAN Generative Adversarial Network.

GPU Graphics Processing Unit.

HR High-Resolution.

HU Hounsfield Units.

LPIPS Learned Perceptual Image Patch Similarity.

LR Low-Resolution.

MR-TRAE Multi-resolution Twinned Residual Auto-Encoders.

MRI Magnetic Resonance Imaging.

MSE Mean Squared Error.

NCP Novel Coronavirus Pneumonia.

NN Neural Network.

PReLU Parametric ReLU.

PSNR Peak Signal-to-Noise Ratio.

RCAB Residual Channel Attention Block.

RCAN Residual Channel Attention Network.

SISR Single Image Super-Resolution.

SR Super-Resolution.

SR-CNN Super-Resolution Convolutional Neural Network.

SRCNN Super-Resolution Convolutional Neural Network.

SRGAN Super-Resolution Generative Adversarial Network.

SRResNet Super-Resolution Residual Network.

SSIM Structural Similarity Index Measure.

TGSR Task Grouping based real-SR.

TRAE Twinned Residual Auto-Encoder.

VDSR Very Deep Super-Resolution.

VGG Visual Geometry Group.

VRAM Video Random Access Memory.

Chapter 1

Introduction

1.1 Single image super-resolution: definition

Single Image Super-Resolution (SISR) is the task of recovering a High-Resolution (HR) image from a single Low-Resolution (LR) image.

Let $\mathbf{X}_{\text{HR}} \in \mathbb{R}^{H \times W \times C}$ denote the unknown HR image and $\mathbf{Y}_{\text{LR}} \in \mathbb{R}^{h \times w \times C}$ its observed LR image, where $s = \frac{H}{h} = \frac{W}{w} > 1$ is the upscaling factor. We denote the vectorized forms by $\mathbf{x}_{\text{HR}} \in \mathbb{R}^N$ and $\mathbf{y}_{\text{LR}} \in \mathbb{R}^M$, where $N = HWC$ and $M = hwC$.

We express LR images as degraded versions of HR images:

$$\mathbf{y}_{\text{LR}} = \mathcal{T}(\mathbf{x}_{\text{HR}}; \mathcal{D}) + \mathbf{n}, \tag{1.1}$$

where $\mathcal{T}(\cdot; \mathcal{D})$ is the degradation operator parameterized by $\mathcal{D}$ (for example, blur kernel, downsampling rule, compression), and $\mathbf{n}$ is additive noise such as zero-mean Gaussian noise.

The goal of SISR is to estimate $\widehat{\mathbf{x}_{\text{HR}}}$ given only $\mathbf{y}_{\text{LR}}$:

$$\widehat{\mathbf{x}_{\text{HR}}} = f_\theta(\mathbf{y}_{\text{LR}}), \tag{1.2}$$

where f_θ is a learned reconstruction rule, typically a Neural Network (NN).

The inverse nature of this task leads to some problems:

- The mapping $\mathcal{T}$ is generally non-invertible, meaning that multiple HR images correspond to the same LR observation.
- High-frequency information (such as edges and fine textures) is lost during the degradation. This makes the the faithful reconstruction difficult.
- The process must balance between perceptual realism and fidelity to the input LR image, especially in critical applications such as medical imaging.

1.2 Motivations

SISR is useful because improving the quality and detail of an image helps both people and computer systems understand the image better and make more accurate decisions:

- Medical imaging: high quality Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and ultrasound images are important for accurate diagnosis. However, acquisition devices often trade resolution for speed or dose reduction. In such domains, preserving anatomical fidelity and avoiding hallucinated structures is very important. Super-Resolution (SR) can improve image quality without additional patient exposure.
- Remote sensing: In satellite or aerial imaging, the resolution is limited because of sensor limitations and so on. SR can give better target recognition and environmental monitoring.
- Photography and media restoration: SR plays an important role in photo enhancement, old film restoration, video streaming upscaling, and improving visual content in consumer electronics.

In all of these applications, we often have only one single LR image per scene, and the problem is that SISR relies on prior knowledge to restore the lost high-frequency information.

1.3 Why SISR is ill-posed?

1.3.1 Under-determined inverse problem

In many practical settings, the degradation can be approximated by a linear operator (after vectorization):

$$\mathbf{y}_{\mathrm{LR}} = \mathbf{A}\mathbf{x}_{\mathrm{HR}} + \mathbf{n}, \tag{1.3}$$

where $\mathbf{A} \in \mathbb{R}^{M \times N}$ combines blur and downsampling, and typically $M \ll N$ because $M = \dfrac{N}{s^2}$. Hence the system is under-determined:

$$\mathrm{rank}(\mathbf{A}) \leq M < N \quad \Longrightarrow \quad \mathrm{null}(\mathbf{A}) \neq \{\mathbf{0}\}, \tag{1.4}$$

so there exist nonzero $\mathbf{v}$ such that $\mathbf{A}\mathbf{v} = \mathbf{0}$. Therefore, if $\mathbf{x}_{\mathrm{HR}}$ is a solution to the noiseless problem $\mathbf{y}_{\mathrm{LR}} = \mathbf{A}\mathbf{x}_{\mathrm{HR}}$, then $\mathbf{x}_{\mathrm{HR}} + \mathbf{v}$ is also a solution for any $\mathbf{v} \in \mathrm{null}(\mathbf{A})$. This proves *non-uniqueness*.

1.3.2 Instability (sensitivity to noise)

Even if a reconstruction rule exists, inversion is unstable because blur/downsampling remove high-frequency information and $\mathbf{A}$ is ill-conditioned. Small perturbations in $\mathbf{y}_{\mathrm{LR}}$ can cause large changes in $\widehat{\mathbf{x}_{\mathrm{HR}}}$. According to Hadamard, a problem is well-posed if: (i) a solution exists, (ii) it is unique, and (iii) it depends continuously on the observations. In SISR, conditions (ii) and (iii) fail because of the information loss

from $\mathcal{T}$ (or $\mathbf{A}$) through down-sampling and blur, and because $M \ll N$. Therefore, SISR is mathematically an ill-posed inverse problem.

Formally, for two observations $\mathbf{y}_{\mathrm{LR}}^{(1)}$, $\mathbf{y}_{\mathrm{LR}}^{(2)}$ and corresponding estimates $\widehat{\mathbf{x}_{\mathrm{HR}}}^{(1)}$, $\widehat{\mathbf{x}_{\mathrm{HR}}}^{(2)}$, $\|\widehat{\mathbf{x}_{\mathrm{HR}}}^{(1)} - \widehat{\mathbf{x}_{\mathrm{HR}}}^{(2)}\|$ can be large even when $\|\mathbf{y}_{\mathrm{LR}}^{(1)} - \mathbf{y}_{\mathrm{LR}}^{(2)}\|$ is small, that shows the lack of "continuous dependence" on the data.

1.3.3 Need for regularization and priors

To select a meaningful HR estimate among the infinitely many cases, additional constraints are needed that are shown by regularization or priors:

$$\widehat{\mathbf{x}_{\mathrm{HR}}} = \arg\min_{\mathbf{x}_{\mathrm{HR}}} \Big(\|\mathbf{y}_{\mathrm{LR}} - \mathcal{T}(\mathbf{x}_{\mathrm{HR}}; \mathcal{D})\|_2^2 + \lambda\, \mathcal{R}(\mathbf{x}_{\mathrm{HR}}) \Big), \tag{1.5}$$

where $\mathcal{R}$ encodes prior assumptions (smoothness, sparsity, self-similarity, natural-image statistics) and $\lambda > 0$ controls the fidelity–prior trade-off.

1.3.4 Bayesian view: posterior, MAP, and sampling

A probabilistic view of (1.1) models the unknown HR image, $\mathbf{x}_{\mathrm{HR}}$, as a random variable.

The degradation model induces a likelihood $p(\mathbf{y}_{\mathrm{LR}} \mid \mathbf{x}_{\mathrm{HR}}; \mathcal{D})$, and prior knowledge on plausible HR images is encoded by a prior distribution $p(\mathbf{x}_{\mathrm{HR}})$. The Bayesian formulation leads to the posterior:

$$p(\mathbf{x}_{\mathrm{HR}} \mid \mathbf{y}_{\mathrm{LR}}; \mathcal{D}) = \frac{p(\mathbf{y}_{\mathrm{LR}} \mid \mathbf{x}_{\mathrm{HR}}; \mathcal{D})\, p(\mathbf{x}_{\mathrm{HR}})}{p(\mathbf{y}_{\mathrm{LR}}; \mathcal{D})}, \tag{1.6}$$

where the evidence (marginal likelihood) is

$$p(\mathbf{y}_{\mathrm{LR}}; \mathcal{D}) = \int p(\mathbf{y}_{\mathrm{LR}} \mid \mathbf{x}_{\mathrm{HR}}; \mathcal{D})\; p(\mathbf{x}_{\mathrm{HR}})\; \mathrm{d}\mathbf{x}_{\mathrm{HR}}. \tag{1.7}$$

The ill-posedness of SISR is reflected by the fact that the posterior $p(\mathbf{x}_{\mathrm{HR}} \mid \mathbf{y}_{\mathrm{LR}}; \mathcal{D})$ is generally broad and often multi-modal which means that many distinct HR images can be compatible with the same LR image.

If the noise term in (1.1) is modeled as i.i.d. Gaussian, $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$, then

$$p(\mathbf{y}_{\mathrm{LR}} \mid \mathbf{x}_{\mathrm{HR}}; \mathcal{D}) = \mathcal{N}(\mathbf{y}_{\mathrm{LR}};\, \mathcal{T}(\mathbf{x}_{\mathrm{HR}}; \mathcal{D}),\, \sigma_n^2 \mathbf{I}), \tag{1.8}$$

and the negative log-likelihood, up to an additive constant, is

$$-\log p(\mathbf{y}_{\mathrm{LR}} \mid \mathbf{x}_{\mathrm{HR}}; \mathcal{D}) = \frac{1}{2\sigma_n^2} \|\mathbf{y}_{\mathrm{LR}} - \mathcal{T}(\mathbf{x}_{\mathrm{HR}}; \mathcal{D})\|_2^2 + \text{const}. \tag{1.9}$$

A classical point-estimation approach is Maximum A Posteriori (MAP):

$$\widehat{\mathbf{x}_{\mathrm{HR}}}^{\mathrm{MAP}} = \arg\max_{\mathbf{x}_{\mathrm{HR}}} p(\mathbf{x}_{\mathrm{HR}} \mid \mathbf{y}_{\mathrm{LR}}; \mathcal{D}) = \arg\min_{\mathbf{x}_{\mathrm{HR}}} \Big(-\log p(\mathbf{y}_{\mathrm{LR}} \mid \mathbf{x}_{\mathrm{HR}}; \mathcal{D}) - \log p(\mathbf{x}_{\mathrm{HR}}) \Big), \tag{1.10}$$

which reduces to (1.5) with $\mathcal{R}(\mathbf{x}_{\mathrm{HR}}) \propto -\log p(\mathbf{x}_{\mathrm{HR}})$. In contrast, **posterior sampling** draws multiple plausible HR reconstructions:

$$\mathbf{x}_{\mathrm{HR}}^{(i)} \sim p(\mathbf{x}_{\mathrm{HR}} \mid \mathbf{y}_{\mathrm{LR}}; \mathcal{D}), \quad i = 1, \ldots, M, \tag{1.11}$$

presenting the ambiguity of SISR (multi-modality).

By substituting (1.9) into (1.10), we obtain the regularized least-squares form:

$$\widehat{\mathbf{x}_{\mathrm{HR}}}^{\mathrm{MAP}} = \arg\min_{\mathbf{x}_{\mathrm{HR}}} \frac{1}{2\sigma_n^2} \|\mathbf{y}_{\mathrm{LR}} - \mathcal{T}(\mathbf{x}_{\mathrm{HR}}; \mathcal{D})\|_2^2 + \underbrace{(-\log p(\mathbf{x}_{\mathrm{HR}}))}_{\triangleq \lambda \mathcal{R}(\mathbf{x}_{\mathrm{HR}})}. \tag{1.12}$$

So, the regularizer $\mathcal{R}$ is the negative log-prior (up to scaling), and the parameter λ controls the prior strength relative to the data fidelity term.

1.4 Why deep learning?

In practice, the prior $p(\mathbf{x}_{\mathrm{HR}})$ or $\mathcal{R}$ are unknown and difficult to model analytically for complex image distributions. Classical methods try to design $\mathcal{R}$ or $p(\mathbf{x}_{\mathrm{HR}})$ (total variation, sparse coding), then solve (1.5) iteratively. They face limitations:

- Handcrafted priors poorly approximate the complex distribution of real images, especially textures.
- Iterative solvers are slow and sensitive to parameter tuning.
- Different domains (medical vs natural images) require different priors and degradation.

The difficulty of SISR lies in learning an effective image prior and in performing inference efficiently under the resulting posterior.

Deep Learning (DL) addresses these by learning data-driven priors. Regression-based networks can be interpreted as learning an implicit regularizer through architecture and training objective, while generative models such as Generative Adversarial Networks (GANs) and Diffusion Models (DMs) can learn distributions and sample from multi-modal posteriors.

DL offers several advantages.

- Deep NNs learn image priors from large datasets that enables richer modeling than analytical priors.
- Once trained, the networks can perform SR in a single forward pass to avoid iterative optimization.
- Convolutional Neural Networks (CNNs) capture multi-scale features and textures that align with human perception.
- Loss functions can optimize not only pixel-level accuracy but also perceptual quality.

1.5 Thesis scope

This thesis focuses on DL approaches to SISR, with an emphasis on: (i) stability and fidelity in sensitive domains (like medical imaging), (ii) perceptual quality, and (iii) computational efficiency.

The rest of this thesis is structured as follows:

- Chapter 2 presents the main DL paradigms for SISR—regression models, GAN-based models, and DM—together with their objectives, mathematical foundations, and known limitations.
- Chapter 3 presents a detailed review of existing SISR methods, including CNN-based, GAN-based, and DM models.
- Chapter 4 describes the proposed Twinned Residual Auto-Encoder (TRAE) model, its architecture, loss functions, training algorithm.
- Chapter 5 introduces the Multi-resolution Twinned Residual Auto-Encoders (MR-TRAE), its training mechanism, and multi-scale architecture for.
- Chapter 6 presents the regression-based SR model, Super-Resolution Convolutional Neural Network (SR-CNN), enhanced by perceptual and structural losses, offering a GAN-free alternative with high perceptual quality.
- Chapter 7 gives a comparative analysis of all models, summarizes results, discusses limitations, and highlights trade-offs between quality and stability.
- Chapter 8 concludes the thesis with key findings and gives directions for future research in the field of DL-based SR.

1.6 Publications related to this thesis

This section presents the publications produced during the PhD research. This includes peer-reviewed articles and a manuscript[1]. Two chapters are written about previously published and collaborative work, with important methodological and architectural extensions developed within the scope of this study.

Paper [3] introduced a training strategy and loss function formulation for SR by introducing the TRAE. In that publication, the candidate contributed primarily to the writing of the manuscript. In the present thesis, Chapter 4 extends the original work by improving the training strategy and the design of loss functions.

Paper [40] presented the architectural components of the overall network model of MR-TRAE. In this collaborative work, the candidate contributed to the writing of the manuscript, as well as to the model design and training procedures. Chapter 5 of this thesis advances that work by introducing improvements to the components of the network, with major enhancements to the overall loss formulation and training strategy that results in a more optimized framework.

[1] Perceptual super-resolution via CNN with advanced loss functions.

SR-CNN, the model presented in Chapter 6, forms the basis of an additional manuscript. The architectural design, training framework, and loss functions in Chapter 6 were implemented entirely by the candidate.

Chapter 2

Deep learning foundations for single image super-resolution

2.1 Problem setting

We consider paired training data $\{(\mathbf{y}_{\mathrm{LR}}^{(i)}, \mathbf{x}_{\mathrm{HR}}^{(i)})\}_{i=1}^{N}$, where $\mathbf{y}_{\mathrm{LR}}^{(i)}$ is an LR observation generated from $\mathbf{x}_{\mathrm{HR}}^{(i)}$ via (1.1) and N is the number of available image pairs.

Deep SISR methods differ in what they learn:

- Regression models learn a deterministic mapping $\widehat{\mathbf{x}_{\mathrm{HR}}} = f_\theta(\mathbf{y}_{\mathrm{LR}})$.
- GANs learn a generator G_θ that produces perceptually sharp outputs that are guided by a discriminator.
- DMs learn a conditional generative process that can sample from $p(\mathbf{x}_{\mathrm{HR}} \mid \mathbf{y}_{\mathrm{LR}})$.

2.2 Regression-based (distortion-oriented) super-resolution

2.2.1 Deterministic mapping and pixel-wise objectives

SISR is formulated as a supervised regression problem. Let $(\mathbf{y}_{\mathrm{LR}}, \mathbf{x}_{\mathrm{HR}}) \sim p(\mathbf{y}_{\mathrm{LR}}, \mathbf{x}_{\mathrm{HR}})$ denote a pair of LR and HR images drawn from the unknown joint distribution.

Regression-based SR tries to find a parametric mapping

$$\widehat{\mathbf{x}_{\mathrm{HR}}} = f_\theta(\mathbf{y}_{\mathrm{LR}}), \tag{2.1}$$

where θ are trainable parameters.

A common objective is empirical risk minimization under an ℓ_2 distortion:

$$\min_\theta \ \mathbb{E}_{(\mathbf{y}_{\mathrm{LR}}, \mathbf{x}_{\mathrm{HR}})}\Big[\|f_\theta(\mathbf{y}_{\mathrm{LR}}) - \mathbf{x}_{\mathrm{HR}}\|_2^2\Big], \tag{2.2}$$

or an ℓ_1 loss:

$$\min_\theta \ \mathbb{E}_{(\mathbf{y}_{\mathrm{LR}}, \mathbf{x}_{\mathrm{HR}})}\Big[\|f_\theta(\mathbf{y}_{\mathrm{LR}}) - \mathbf{x}_{\mathrm{HR}}\|_1\Big]. \tag{2.3}$$

Under a conditional Gaussian noise model, minimizing (2.2) corresponds to maximizing a Gaussian likelihood and yields the conditional mean estimator, which explains the tendency toward *over-smoothed* reconstructions.

In fact, (2.2) is approximated by the empirical mean over a dataset $\{(\mathbf{y}_{\mathrm{LR}}^{(i)}, \mathbf{x}_{\mathrm{HR}}^{(i)})\}_{i=1}^{N}$ assumed i.i.d. from $p(\mathbf{y}_{\mathrm{LR}}, \mathbf{x}_{\mathrm{HR}})$:

$$\mathbb{E}[\,\cdot\,] \approx \frac{1}{N}\sum_{i=1}^{N}(\cdot). \tag{2.4}$$

When f_θ is parameterized by a CNN: $F_{\boldsymbol{\theta}}$, this results in the standard Mean Squared Error (MSE) training objective used in deep SISR:

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_{\mathrm{HR}}^{(i)} - F_{\boldsymbol{\theta}}(\mathbf{y}_{\mathrm{LR}}^{(i)})\|_2^2. \tag{2.5}$$

2.2.2 Connection to maximum likelihood estimation

We formalize the link between the ℓ_2 objective in (2.2) and Gaussian maximum likelihood.

Assumption 2.1 (Conditional Gaussian model)**.** Assume that the conditional distribution of $\mathbf{x}_{\mathrm{HR}}$ given $\mathbf{y}_{\mathrm{LR}}$ satisfies

$$\mathbf{x}_{\mathrm{HR}} \mid \mathbf{y}_{\mathrm{LR}} \sim \mathcal{N}(f_\theta(\mathbf{y}_{\mathrm{LR}}), \sigma^2\mathbf{I}).$$

Proposition 2.2. *Under the conditional Gaussian model, maximizing the log-likelihood with respect to θ is equivalent to minimizing* (2.2).

Sketch of Proof. The conditional density is

$$p(\mathbf{x}_{\mathrm{HR}} \mid \mathbf{y}_{\mathrm{LR}}; \theta) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}_{\mathrm{HR}} - f_\theta(\mathbf{y}_{\mathrm{LR}})\|_2^2\right),$$

where d is the dimensionality of $\mathbf{x}_{\mathrm{HR}}$. We take the logarithm:

$$\log p(\mathbf{x}_{\mathrm{HR}} \mid \mathbf{y}_{\mathrm{LR}}; \theta) = -\frac{d}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|\mathbf{x}_{\mathrm{HR}} - f_\theta(\mathbf{y}_{\mathrm{LR}})\|_2^2.$$

The first term does not depend on θ. Therefore, maximizing the log-likelihood is equivalent to minimizing

$$\|\mathbf{x}_{\mathrm{HR}} - f_\theta(\mathbf{y}_{\mathrm{LR}})\|_2^2.$$

Taking expectation over $(\mathbf{y}_{\mathrm{LR}}, \mathbf{x}_{\mathrm{HR}})$ leads to (2.2). ■

2.2.3 Characterization of the optimal estimator

The ℓ_2 risk admits a well-known characterization.

Proposition 2.3. *The minimizer of* (2.2) *over all measurable functions satisfies*

$$f_\theta^*(\mathbf{y}_{\mathrm{LR}}) = \mathbb{E}[\mathbf{x}_{\mathrm{HR}} \mid \mathbf{y}_{\mathrm{LR}}].$$

Sketch of Proof. For fixed $\mathbf{y}_{\mathrm{LR}}$, define

$$R(f) = \mathbb{E}[\|\mathbf{x}_{\mathrm{HR}} - f(\mathbf{y}_{\mathrm{LR}})\|_2^2 \mid \mathbf{y}_{\mathrm{LR}}].$$

Expanding:

$$R(f) = \mathbb{E}[\|\mathbf{x}_{\mathrm{HR}}\|_2^2 \mid \mathbf{y}_{\mathrm{LR}}] - 2f(\mathbf{y}_{\mathrm{LR}})^\top \mathbb{E}[\mathbf{x}_{\mathrm{HR}} \mid \mathbf{y}_{\mathrm{LR}}] + \|f(\mathbf{y}_{\mathrm{LR}})\|_2^2.$$

Differentiating with respect to $f(\mathbf{y}_{\mathrm{LR}})$ and setting to zero gives

$$f(\mathbf{y}_{\mathrm{LR}}) = \mathbb{E}[\mathbf{x}_{\mathrm{HR}} \mid \mathbf{y}_{\mathrm{LR}}].$$

■

Thus, ℓ_2 regression yields the conditional mean estimator.

2.2.4 Implications for SR

In SR, the conditional distribution $p(\mathbf{x}_{\mathrm{HR}} \mid \mathbf{y}_{\mathrm{LR}})$ is typically multi-modal: multiple HR images correspond to the same LR observation due to information loss in downsampling.

The conditional mean therefore averages over plausible high-frequency realizations and the variations in detailed regions are reduced. This fact represents the over-smoothing that can be observed in distortion-oriented SR models.

Instead, (2.3) results in the conditional median, which is more robust to residuals but still remains a deterministic point estimator.

2.3 Evaluation metrics for SR

The performance of SR methods is assessed using *distortion-based* and *perceptual* quality metrics. Distortion metrics quantify pixel-wise fidelity to the ground-truth HR image, while perceptual metrics aim to measure similarity in a feature space more aligned with human visual perception.

Let $\mathbf{x}_{\mathrm{HR}} \in \mathbb{R}^N$ denote the ground-truth HR image (vectorized form of $\mathbf{X}_{\mathrm{HR}}$) and $\widehat{\mathbf{x}_{\mathrm{HR}}} \in \mathbb{R}^N$ its reconstruction.

1. Peak signal-to-noise ratio

Peak Signal-to-Noise Ratio (PSNR) is defined in terms of the MSE:

$$\mathrm{MSE}(\mathbf{x}_{\mathrm{HR}}, \widehat{\mathbf{x}_{\mathrm{HR}}}) = \frac{1}{N} \|\mathbf{x}_{\mathrm{HR}} - \widehat{\mathbf{x}_{\mathrm{HR}}}\|_2^2 . \tag{2.6}$$

The PSNR (in decibels) is then given by

$$\mathrm{PSNR}(\mathbf{x}_{\mathrm{HR}}, \widehat{\mathbf{x}_{\mathrm{HR}}}) = 10 \log_{10} \left(\frac{L^2}{\mathrm{MSE}(\mathbf{x}_{\mathrm{HR}}, \widehat{\mathbf{x}_{\mathrm{HR}}})} \right), \tag{2.7}$$

where L denotes the dynamic range of the image intensities (e.g., $L = 255$ for 8-bit images, or $L = 1$ for normalized images).

PSNR is a distortion-based metric related to the ℓ_2 reconstruction error. While it is widely used due to its simplicity, it does not correlate well with perceptual quality.

2. Structural similarity index measure

The Structural Similarity Index Measure (SSIM) measures similarity by comparing local luminance, contrast, and structural information between images. For two image patches extracted from $\mathbf{x}_{\mathrm{HR}}$ and $\widehat{\mathbf{x}_{\mathrm{HR}}}$, SSIM is defined as

$$\mathrm{SSIM}(\mathbf{x}_{\mathrm{HR}}, \widehat{\mathbf{x}_{\mathrm{HR}}}) = \frac{(2\mu_x \mu_{\widehat{x}} + C_1)(2\sigma_{x\widehat{x}} + C_2)}{(\mu_x^2 + \mu_{\widehat{x}}^2 + C_1)(\sigma_x^2 + \sigma_{\widehat{x}}^2 + C_2)}, \tag{2.8}$$

where:

- μ_x and $\mu_{\widehat{x}}$ denote local mean intensities,
- σ_x^2 and $\sigma_{\widehat{x}}^2$ denote local variances,
- $\sigma_{x\widehat{x}}$ denotes the local covariance,
- $C_1, C_2 > 0$ are small constants for numerical stability.

In practice, SSIM is computed over sliding windows and averaged over spatial locations. Compared to PSNR, SSIM better reflects structural distortions.

3. Learned perceptual image patch similarity

Learned Perceptual Image Patch Similarity (LPIPS) is a feature-based perceptual metric computed in the embedding space of a pretrained CNN.

Let $\phi_l(\cdot)$ denote the feature map extracted from layer l of a pretrained network like Visual Geometry Group (VGG), and let $w_l \geq 0$ denote learned layer-wise weights. LPIPS is defined as

$$\mathrm{LPIPS}(\mathbf{x}_{\mathrm{HR}}, \widehat{\mathbf{x}_{\mathrm{HR}}}) = \sum_l w_l \frac{1}{H_l W_l} \|\phi_l(\mathbf{x}_{\mathrm{HR}}) - \phi_l(\widehat{\mathbf{x}_{\mathrm{HR}}})\|_2^2 , \tag{2.9}$$

where H_l and W_l denote the spatial dimensions of the feature maps at layer l.

By measuring distances in a deep feature space, LPIPS correlates more strongly with human perceptual judgments than pixel-wise metrics. However, unlike PSNR and SSIM, it does not directly quantify reconstruction fidelity in the image domain.

2.3.1 Perceptual losses as feature-space distortions

To reduce over-smoothing and move beyond pixel-wise conditional mean estimation while remaining within a regression framework, perceptual losses measure distortion in a learned feature space defined by a fixed network $\phi(\cdot)$, typically extracted from a pretrained network such as VGG.

A common choice is to compare deep features at multiple layers using either an ℓ_1 or ℓ_2 distance:

$$\mathcal{L}_{\text{perc}}(\theta) = \sum_{\ell \in \mathcal{L}} \frac{w_\ell}{d_\ell} \left\| \phi_\ell(f_\theta(\mathbf{y}_{\text{LR}})) - \phi_\ell(\mathbf{x}_{\text{HR}}) \right\|_1, \tag{2.10}$$

where $\phi_\ell(\cdot) \in \mathbb{R}^{d_\ell}$ denotes the feature representation at layer ℓ, and $w_\ell \geq 0$ are layer-wise weights. Alternatively, an ℓ_2 formulation can be used:

$$\mathcal{L}_{\text{perc}}(\theta) = \sum_{\ell \in \mathcal{L}} \frac{w_\ell}{d_\ell} \left\| \phi_\ell(f_\theta(\mathbf{y}_{\text{LR}})) - \phi_\ell(\mathbf{x}_{\text{HR}}) \right\|_2^2. \tag{2.11}$$

Such objectives improve perceptual similarity by replacing pixel-space distortion with feature-space distortion and shift the estimator away from the conditional mean in image space. In this context, while perceptual similarity can be improved, the optimality under ℓ_2 distortion in the image domain, and therefore the PSNR and usually the SSIM are reduced.

2.3.2 Typical architectures

Most regression models rely on CNN backbones [10]. As shown in Fig. 2.1, a regression-based SR model learns a mapping from the LR observation to an HR estimate, i.e., $\widehat{\mathbf{x}_{\text{HR}}} = f_\theta(\mathbf{y}_{\text{LR}})$.

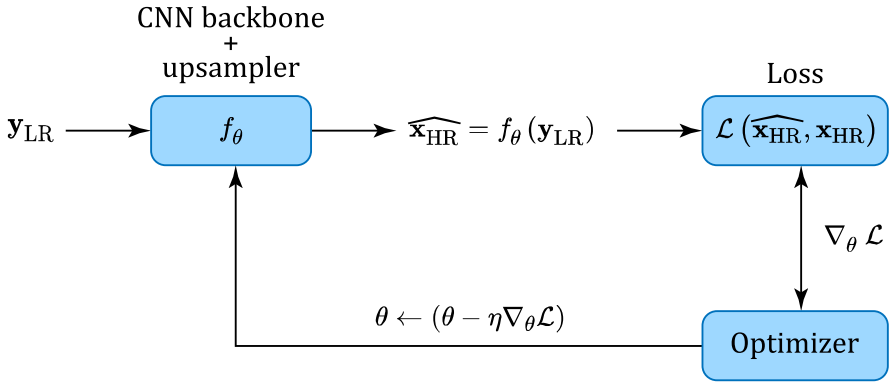

Figure 2.1. A schematic inference of pure regression-based SISR: the LR image $\mathbf{y}_{\text{LR}}$ is mapped to the super-resolved estimate $\widehat{\mathbf{x}_{\text{HR}}}$ by the network f_θ: $\widehat{\mathbf{x}_{\text{HR}}} = f_\theta(\mathbf{y}_{\text{LR}})$. A loss $\mathcal{L}(\widehat{\mathbf{x}_{\text{HR}}}, \mathbf{x}_{\text{HR}})$ compares the estimate to the ground-truth HR image, and an optimizer updates the parameters θ using the gradient of the loss. $\eta > 0$ denotes the learning rate.

A canonical residual SR model parameterizes this mapping as

$$f_\theta(\mathbf{y}_{\text{LR}}) = \mathbf{U}(\mathbf{y}_{\text{LR}}) + g_\theta(\mathbf{y}_{\text{LR}}), \tag{2.12}$$

where $\mathbf{U}$ is a fixed upsampling operator (e.g., bicubic interpolation) and g_θ predicts the HR residual. This decomposition can be interpreted as learning corrections in the high-frequency subspace, while preserving the low-frequency structure through $\mathbf{U}$.

During training (Fig. 2.1), the parameters θ are optimized by minimizing a loss $\mathcal{L}(\widehat{\mathbf{x}_{\mathrm{HR}}}, \mathbf{x}_{\mathrm{HR}})$ between the prediction and the ground-truth HR image.

2.4 GANs for SR

2.4.1 Classical (unconditional) GAN

GANs introduce two components: a generator G_θ and a discriminator D_ψ [15]. These two components cooperate with each other and progressively strengthen their individual performance as well as that of the other component. The generator maps a latent random vector $\mathbf{z} \sim p(\mathbf{z})$ (e.g., $\mathcal{N}(\mathbf{0}, \mathbf{I})$) to a synthetic sample

$$\widetilde{\mathbf{x}} = G_\theta(\mathbf{z}), \tag{2.13}$$

intended to resemble real samples $\mathbf{x} \sim p_{\mathrm{data}}(\mathbf{x})$. The discriminator outputs $D_\psi(\mathbf{x}) \in (0, 1)$, interpreted as the probability that $\mathbf{x}$ is real (from the data distribution) rather than generated.

The original logistic GAN objective is

$$\max_{\psi} \; \mathcal{L}_D(\psi; \theta) = \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}}\big[\log D_\psi(\mathbf{x})\big] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\big[\log\big(1 - D_\psi(G_\theta(\mathbf{z}))\big)\big], \tag{2.14}$$

$$\min_{\theta} \; \mathcal{L}_{\mathrm{adv}}(\theta; \psi) = \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\big[-\log D_\psi(G_\theta(\mathbf{z}))\big], \tag{2.15}$$

where (2.15) corresponds to the commonly used non-saturating generator loss. Training alternates between updating ψ to improve discrimination and updating θ to produce samples that satisfy the discriminator.

2.4.2 Why unconditional GANs are not suitable for SISR

SISR is a *conditional* generation task: the goal is to generate an HR image that is consistent with a specific LR input. In our notation, a regression-based LR model estimates $\widehat{\mathbf{x}_{\mathrm{HR}}} = f_\theta(\mathbf{y}_{\mathrm{LR}})$, and a generative SR model similarly aims to produce $\widehat{\mathbf{x}_{\mathrm{HR}}}$ conditioned on $\mathbf{y}_{\mathrm{LR}}$.

A classical (unconditional) GAN is mismatched to this setting for two reasons:

- the discriminator $D_\psi(\mathbf{x})$ in an unconditional GAN never observes $\mathbf{y}_{\mathrm{LR}}$. So, it cannot verify whether a generated HR image is consistent with the input LR measurement. So, the generator may produce realistic-looking textures that are not supported by the observed LR content,
- because SISR is ill-posed, many HR images can appear plausible for a given LR input. Without an explicit conditioning mechanism, the adversarial objective can increase the risk of hallucinated structures, which is an unacceptable mode in medical imaging.

These limitations motivate *conditional* GAN formulations for SISR.

2.4.3 Conditional GANs for SR

In Conditional Generative Adversarial Networks (cGANs) [47], both generator and discriminator are conditioned on the observed input. In SISR, the generator produces

$$\widehat{\mathbf{x}_{\mathrm{HR}}} = G_\theta(\mathbf{y}_{\mathrm{LR}}), \tag{2.16}$$

and the discriminator receives the pair $(\mathbf{x}_{\mathrm{HR}}, \mathbf{y}_{\mathrm{LR}})$ and outputs $D_\psi(\mathbf{x}_{\mathrm{HR}}, \mathbf{y}_{\mathrm{LR}}) \in (0, 1)$.

Conditioning can be implemented by concatenation, feature-wise modulation, or cross-attention. In all cases, the discriminator is informed about the LR observation used to form the SR candidate.

A standard logistic cGAN objective is

$$\begin{aligned} \max_{\psi} \; \mathcal{L}_D(\psi;\theta) &= \mathbb{E}_{(\mathbf{y}_{\mathrm{LR}},\mathbf{x}_{\mathrm{HR}})\sim p_{\mathrm{data}}}\big[\log D_\psi(\mathbf{x}_{\mathrm{HR}}, \mathbf{y}_{\mathrm{LR}})\big] \\ &\quad + \mathbb{E}_{\mathbf{y}_{\mathrm{LR}}\sim p_{\mathrm{data}}}\big[\log\big(1 - D_\psi(\widehat{\mathbf{x}_{\mathrm{HR}}}, \mathbf{y}_{\mathrm{LR}})\big)\big], \qquad (2.17) \\ \min_{\theta} \; \mathcal{L}_{\mathrm{adv}}(\theta;\psi) &= \mathbb{E}_{\mathbf{y}_{\mathrm{LR}}\sim p_{\mathrm{data}}}\big[-\log D_\psi(\widehat{\mathbf{x}_{\mathrm{HR}}}, \mathbf{y}_{\mathrm{LR}})\big], \qquad (2.18) \end{aligned}$$

with $\widehat{\mathbf{x}_{\mathrm{HR}}} = G_\theta(\mathbf{y}_{\mathrm{LR}})$ defined in (2.16).

2.4.4 SR specific generator loss

In practice, GAN-based SR does not rely on the adversarial term alone. A typical generator objective combines adversarial learning with content fidelity and perceptual constraints:

$$\min_{\theta} \; \mathcal{L}_G(\theta) = \lambda_{\mathrm{cont}}\,\mathcal{L}_{\mathrm{cont}}\big(\widehat{\mathbf{x}_{\mathrm{HR}}}, \mathbf{x}_{\mathrm{HR}}\big) + \lambda_{\mathrm{perc}}\,\mathcal{L}_{\mathrm{perc}}\big(\widehat{\mathbf{x}_{\mathrm{HR}}}, \mathbf{x}_{\mathrm{HR}}\big) + \lambda_{\mathrm{adv}}\,\mathcal{L}_{\mathrm{adv}}(\theta;\psi), \tag{2.19}$$

where $\lambda_{\mathrm{cont}}, \lambda_{\mathrm{perc}}, \lambda_{\mathrm{adv}} \geq 0$. Common choices include $\mathcal{L}_{\mathrm{cont}}(\widehat{\mathbf{x}_{\mathrm{HR}}}, \mathbf{x}_{\mathrm{HR}}) = \|\widehat{\mathbf{x}_{\mathrm{HR}}} - \mathbf{x}_{\mathrm{HR}}\|_1$ (or MSE) and a VGG-based perceptual loss.

Representative GAN-based SR methods such as Super-Resolution Generative Adversarial Network (SRGAN) [33] demonstrate that adversarial training can produce sharp textures and visually pleasing results. However, the fundamental issue of the trade-off between perceptual realism and fidelity still remains.

2.4.5 Limitations in Medical SISR

Despite perceptual improvements, GAN-based SR can be problematic for medical imaging: (i) adversarial learning may introduce hallucinated details not supported by the measurement $\mathbf{y}_{\mathrm{LR}}$; (ii) training is sensitive and may be unstable because of the minimax game; and (iii) outputs may vary in fine texture in ways that are undesirable for diagnostic cases.

For these reasons, this thesis emphasizes GAN-free approaches that aim to improve perceptual quality while maintaining structural fidelity and stable training.

2.5 Diffusion models for SR

2.5.1 Conditional generative perspective

DMs aim to learn a conditional generative process that can sample from $p(\mathbf{x}_{\mathrm{HR}} \mid \mathbf{y}_{\mathrm{LR}}; \mathcal{D})$, rather than producing a single point estimate. This aligns with the Bayesian sampling view in (1.11).

2.5.2 Forward diffusion (noising) process

Let $\mathbf{x}_0$ denote the clean HR image (vectorized), i.e., $\mathbf{x}_0 \triangleq \mathbf{x}_{\mathrm{HR}}$, and define a Markov forward process:

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}), \tag{2.20}$$

In Denoising Diffusion Probabilistic Model (DDPM) with Gaussian transitions [22]:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \mathbf{I}), \tag{2.21}$$

where $\{\beta_t\}_{t=1}^{T}$ is a variance schedule. This implies the closed form:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2.22}$$

with $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.

2.5.3 Reverse (denoising) process and conditioning on LR

The learned reverse process is:

$$p_\theta(\mathbf{x}_{0:T} \mid \mathbf{y}_{\mathrm{LR}}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{y}_{\mathrm{LR}}), \tag{2.23}$$

where $p(\mathbf{x}_T) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. In this formulation, the terminal noisy variable $\mathbf{x}_T$ is assumed to be independent of the conditioning LR observation $\mathbf{y}_{\mathrm{LR}}$, since after sufficient forward diffusion steps $\mathbf{x}_T$ approaches an isotropic Gaussian noise distribution. The conditioning information from $\mathbf{y}_{\mathrm{LR}}$ is therefore introduced through the learned reverse transitions $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{y}_{\mathrm{LR}})$.

2.5.4 Noise-prediction training objective

A common training objective minimizes noise prediction error:

$$\min_{\theta} \; \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}} \left[\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{y}_{\mathrm{LR}}) \|_2^2 \right], \tag{2.24}$$

where $\mathbf{x}_t$ is obtained by (2.22).

At inference, the sampling requires iterative denoising from $t = T$ down to 0, which is computationally expensive.

2.5.5 Strengths and drawbacks in SISR

The strengths of DMs are stable training (no adversarial minimax), strong perceptual realism, and ability to produce diverse outputs. Their drawbacks are slow inference due to many denoising steps although acceleration methods exist but still cost more than one-pass regression or GAN generators.

The trade-off Regression models approximate a point estimator (often conditional mean) that favor high PSNR and usually high SSIM but smoothing textures. GANs optimize perceptual realism but can hallucinate and are unstable to train. DMs approximate sampling from $p(\mathbf{x} \mid \mathbf{y})$ and result in high-quality diverse results, at the cost of high computation during both training and inference. Moreover, in medical imaging, sampling-based generative SR can introduce high-frequency structures that are plausible under the learned prior but not necessarily supported by the measurement $\mathbf{y}_{\text{LR}}$. This raises concerns about hallucinations and artifacts.

Given the computational cost of DMs and the risk of generating misleading details, this thesis focuses on improving regression-based SR models. Our goal is to narrow the perceptual-quality gap with generative methods while preserving deterministic behavior, fast single-pass inference, and improved training stability. These properties are particularly important in "clinical" and "resource-constrained" settings.

2.6 Thesis contributions

This thesis aims to improve the performance of SISR by addressing the limitations of existing DL-based approaches. We focus on improving the perceptual quality, stability, and efficiency of CNN-based regression models, while avoiding the drawbacks of adversarial training and high computational overhead associated with GANs and DMs.

The main contributions of this research are as follows:

1. We propose the TRAE, an architecture based on a pair of Auto-Encoders (AEs) (Master and Follower). The Master AE processes and pretrained on HR images and transfers its knowledge to the Follower AE that learns on LR inputs. The This structure enhances the SR capability of regression-based networks, where the Follower AE sends the learned signals to upsampler CNN.

2. MR-TRAE builds no the TRAE framework. It produces super-resolved outputs at multiple upscaling factors in a single forward pass. In addition to the Follower AE that receives high frequency information from the pre-trained Master, the MR-TRAE incorporates cascaded up-sampling modules that learn from the features of the pretrained Master AE.

3. To improve the perceptual fidelity of regression-based SR models without relying on adversarial loss, we develop a GAN-free architecture equipped with a comprehensive loss function. This includes:

    - robust content loss,

- perceptual loss,
- gradient-based edge preservation loss,
- frequency-domain alignment loss,
- style (Gram matrix) loss.

These losses are integrated into a unified optimization framework to enhance texture sharpness and structural integrity.

4. The proposed methods are evaluated on both synthetic (bicubic) and more diverse degradation settings. Performance is compared with a CNN, GAN using metrics like PSNR, SSIM, LPIPS and perceptual evaluations.

Chapter 3

Literature review

3.1 Foundations of SISR

SISR is a fundamentally ill-posed inverse problem in which we seek to reconstruct an HR image from an LR observation. Since multiple HR images correspond to the same LR input, the quality of the reconstruction depends on the prior assumptions imposed by the underlying model. For this reason, the historical development of SISR can be understood as a progression in the design of priors: from hand-crafted reconstruction assumptions, to sparse patch-based models, and later to deep learned mappings. This progression is particularly important in the context of the present thesis, because it explains both the strengths and the limitations of modern SR models.

In addition, the motivation for SR is compelling in medical imaging. In clinical contexts, image acquisition is constrained by hardware limitations, signal-to-noise considerations, and scan duration. As a result, directly acquiring images with very high spatial resolution may be difficult, costly, or impractical. SR therefore emerges not only as a visual enhancement task, but as a valuable strategy for improving the interpretation of images under constraints [16]. In addition, reconstructed structures must remain reliable, plausible, and faithful. This tension between visual quality and structural trustworthiness is key to this thesis and already appears in the early literature.

3.1.1 Medical imaging motivation

The review in [16] is important because it frames SR from the perspective of medical imaging requirements rather than generic enhancement. Instead of proposing a single algorithmic architecture, this work surveys the role of SR across several medical imaging modalities, and emphasizes that resolution enhancement in these settings is closely tied to modality-specific physical and clinical constraints. The paper highlights that many medical imaging systems inherently face a trade-off between spatial resolution, temporal resolution, and signal quality [16].

The main contribution of [16] is conceptual rather than architectural. It establishes

that, in medical imaging, the goal of SR is not simply to generate sharper-looking images, but to improve image usefulness while respecting acquisition physics and preserving clinically relevant structures. This is a key point, because it distinguishes fidelity-critical SR from purely perceptual image enhancement. A visually enhanced reconstruction can be still undesirable if it introduces artificial structures or changes diagnostic details.

The main strength of [16] is therefore its application-level framing. It provides a strong motivation for why SR is valuable in medical contexts and why reliability must be emphasized alongside visual improvement. Its limitation is that it does not discuss CNN-based regression, perceptual feature losses, adversarial learning, or diffusion-based generation. Consequently, in this thesis, [16] serves primarily as a foundational medical motivation rather than a state-of-the-art algorithmic reference.

3.1.2 Sparse-representation learning before deep networks

A major pre-deep-learning work in SISR is the sparse-representation-based framework [61, 7, 1, 27, 28, 12]. The work in [61] is historically significant because it provided one of the most influential learning-based formulations before CNN became dominant [24, 53, 54, 69, 52, 48, 19, 11], and it introduced several ideas that remained conceptually important even after the transition to DL [43, 49, 8, 50, 55, 13, 59, 5, 45, 44].

Unlike later DL methods, [61] is based on a patch-wise sparse coding pipeline rather than an end-to-end trainable convolutional network [31, 2]. The assumption is that an HR image patch can be represented sparsely on an over-complete HR dictionary, and that its corresponding LR patch shares the same sparse coefficients with respect to a paired LR dictionary [60, 71, 42, 57, 70, 38, 26, 66, 25, 56, 51, 42]. To use this assumption, the method learns two coupled dictionaries: one for LR patches and one for HR patches. During inference, each LR patch is first encoded as a sparse combination of atoms in the LR dictionary, typically through an ℓ_1-regularized sparse optimization problem. The recovered sparse coefficients are then transferred to the HR dictionary to reconstruct the corresponding HR patch [61].

The method also includes a global reconstruction refinement [23] step after local patch recovery. This stage provides consistency between the reconstructed HR and the observed LR image under the assumed degradation model. This helps reduce mismatch between local patch reconstruction and the full-image observation process.

The strengths of [61] are substantial for its time. First, it provides a principled prior-sparsity to regularize the ill-posed SR problem. Second, by learning compact coupled dictionaries, it improves on earlier example-based methods that relied more directly on raw patch retrieval. Third, the global reconstruction step improves consistency and can reduce patch-level artifacts. The model is also relatively interpretable, since the prior, the dictionaries, and the inference process are all explicitly defined.

However, from the perspective of later DL methods, there are several limitations. The inference procedure is computationally expensive because it requires sparse optimization for many overlapping patches. The patch-wise formulation also limits the model's ability to capture large contextual dependencies as efficiently as deep

networks with large receptive fields. In addition, the final performance depends heavily on the quality of the learned dictionaries and on the validity of the assumed degradation model. So, while [61] is slower and less scalable than later feed-forward CNNs, it remains an important precursor because it clearly expresses the pre-deep-learning logic of learning a prior-guided mapping from LR observations to HR reconstructions.

3.1.3 Deep learning: SRCNN

The DL era of SISR begins with the Super-Resolution Convolutional Neural Network (SRCNN), introduced in [10] and later extended in [9]. This line of work is widely recognized as the first influential demonstration that a CNN can learn an end-to-end mapping from an interpolated LR image to its HR counterpart, so it can replace multi-stage hand-crafted pipelines with a single trainable model.

SRCNN is simple. It uses a three-layer convolutional design that can be interpreted as: (i) patch extraction and representation, (ii) non-linear mapping, and (iii) reconstruction [10, 9]. In the original design, the first layer uses a relatively large receptive field to extract patch-level features, the second layer performs a non-linear transformation of these features, and the final layer reconstructs the HR output. The later version in [9] studies larger filters, deeper mapping, and the extension from luminance-only processing to simultaneous color-channel reconstruction.

In terms of optimization, SRCNN is trained with a pixel-wise Euclidean objective, i.e. an ℓ_2 reconstruction loss between the predicted HR image and the ground-truth HR image [10, 9]. This reflects the dominant distortion-oriented philosophy of early deep SR, where PSNR and related fidelity metrics were the primary benchmarks.

The strength of SRCNN is in its conceptual clarity. It established the end-to-end CNN paradigm for SISR and showed that learned convolutional mappings could outperform several classical approaches and remain simple at inference time. Relative to [61], SRCNN can be seen as replacing explicit sparse inference and patch dictionaries with an implicit learned non-linear mapping.

At the same time, SRCNN has notable limitations. Because it operates on a bicubic-upsampled input, computation is performed in HR space, which is inefficient in memory and runtime. Its shallow depth also limits the effective receptive field and restricts contextual modeling. In addition, the model is generally scale-specific, requiring separate training for different enlargement factors. Because it is optimized with a pixel-wise ℓ_2 loss, its reconstructions often achieve strong distortion metrics while appearing very smooth and lacking high-frequency sharpness.

From the first SRCNN in [10] to its later version in [9], we see refinement rather than replacement.

Although [10] and [9] belong to the same SR CNN family, the later version should not be considered as a redundant citation. [9] refines and extends the original contribution by providing broader analysis, larger architectural variants, and more complete evaluation. Both of them, however, remain within the same generation of methods: shallow, pre-upsampling, and fidelity-oriented. The work in [9] improves

the framework, it does not remove the basic limitations of shallow receptive fields, HR-space computation, or smoothing induced by pixel-wise regression.

3.1.4 VDSR: deeper regression through residual learning

The Very Deep Super-Resolution (VDSR) in [32] created a major step beyond SRCNN. This work showed that much deeper convolutional models can greatly improve SISR performance when the optimization problem is formulated carefully.

In contrast to the shallow structure of SRCNN, VDSR uses a 20-layer convolutional architecture built from repeated small 3×3 filters [32]. This design increases the effective receptive field while maintaining parameter efficiency. An important innovation is residual learning [62, 18, 20, 14, 65]. Instead of directly predicting the full HR image, the network predicts the residual image, i.e. the missing high-frequency details relative to the interpolated LR input. The final SR output is then obtained by adding this residual to the input. The network also uses zero-padding so that image size is preserved across layers.

VDSR is trained using a pixel-wise MSE loss function, again an ℓ_2 loss. However, the greater depth makes optimization more difficult. To address this, [32] combines residual learning with adjustable gradient clipping to enable the use of very high learning rates and faster convergence.

Compared with SRCNN, VDSR offers several advantages. Its deeper receptive field allows it to exploit broader contextual information, which is useful for larger scaling factors. Residual prediction improves optimization efficiency, and the method introduces the practical benefit of training a single model for multiple scale factors. In this sense, VDSR is a great improvement over the SRCNN generation.

However, important limitations remain. Like SRCNN, VDSR still relies on bicubic pre-upsampling and therefore performs most computation in HR space. It also remains fundamentally distortion-oriented because it is trained with a pixel-wise ℓ_2 objective. As a result, while VDSR improves reconstruction accuracy, it still tends to produce smooth textures rather than highly perceptual fine details. so, [32] represents a major architectural advance within the same regression paradigm.

3.1.5 Perceptual loss as a critique of pure pixel-wise regression

An important shift appears in [29]. Although this work is not a standard benchmark SR paper in the same sense as SRCNN or VDSR, it it directly questions the per-pixel losses for image transformation tasks, including SR.

The idea in [29] is to train a feed-forward image transformation network using perceptual losses computed in the feature space of a fixed pretrained classification network, specifically a VGG-based loss network. Instead of relying only on low-level pixel-wise discrepancies, the method measures differences in higher-level feature representations. In this way, the optimization objective becomes more aligned with visual perception than with strict pixel-wise matching.

For SR, the importance of [29] is that it shows a model trained only for distortion

minimization does not necessarily produce the most visually convincing result. By using feature-space losses, the method shows that reconstructions can become sharper and perceptually more plausible than those obtained under pure pixel-wise supervision.

The strength of [29] is therefore conceptual. It shows the mismatch between distortion-based objectives and human visual perception, and provides a practical mechanism for training feed-forward models with perceptual guidance. This insight later became a foundation for perceptual SR, including adversarial training methods.

Its limitation, particularly from the perspective of medical imaging, is that perceptual improvement does not guarantee exact structural faithfulness. Because feature-space optimization relaxes strict pixel correspondence. It can encourage plausible details that are not faithful to the original structure. So, [29] is understood as an important bridge. It exposes why regression trained only with ℓ_1/ℓ_2 losses often yields visually conservative results.

3.1.6 EDSR: stronger residual regression

The regression-based line was further strengthened by the Enhanced Deep Super-Resolution Network (EDSR)) in [35]. This work builds on the success of residual learning. This work re-evaluates which components of residual networks are actually beneficial for low-level restoration tasks.

EDSR begins from an architecture like Super-Resolution Residual Network (SR-ResNet) backbone but removes Batch Normalization (BN) layers from the residual blocks [35]. This is a significant design choice. The authors argue that batch normalization may be sub-optimal for SISR because it normalizes feature statistics and can restrict the range needed for precise reconstruction. Removing these layers also reduces memory usage and allows larger models to be trained. The final single-scale EDSR model increases both depth and width through more residual blocks and more feature channels. To stabilize the training of such a large network, the method also introduces residual scaling. In residual scaling, the residual branch is multiplied by a constant factor before being added back to the main path.

In addition, [35] proposes a multi-scale variant of the EDSR in which a large portion of the parameters is shared across scales while only scale-specific modules are retained near the input and output stages (MDSR).

In image tasks, one part of the network may focus on fine details (small scales) and another on global structure (large scales). This can improve performance because each part becomes very good at its specific resolution or feature size. But the problem is that if every scale has its own completely separate parameters, the model becomes large and inefficient. However, the downside is that too much parameter sharing may reduce the model's ability to specialize for each scale.

The method in [35] provides a good compromise between small scales and large scales. The work in [35] shares parameters across scales. This strategy reduces memory usage, overfitting, and training time.

With respect to loss design, [35] is also important because it reflects the field's movement from ℓ_2-dominated optimization toward ℓ_1-based regression. This change preserves deterministic training while often improving reconstruction sharpness and optimization behavior. The work in [35], actually showed that using ℓ_1-based regression leads to better results.

The strengths of EDSR are clear. It improves reconstruction accuracy over earlier residual models. It shows that simplification can outperform direct adoption of classification-oriented designs, and demonstrates that large residual regression models can be trained effectively. It also improves computational logic relative to SRCNN and VDSR by extracting features largely before the final upsampling stage.

However, its limitations remain tied to the broader regression framework. Even with a stronger architecture, EDSR is still fundamentally a fidelity-oriented deterministic model. As a result, it can produce highly accurate reconstructions. however, it still under-represents highly stochastic or ambiguous textures with quite controlled degradation. In addition, its larger capacity increases computational and memory demands.

3.1.7 RCAN: very deep regression with channel attention

Residual Channel Attention Network (RCAN), proposed in [68], further advances regression-based SISR by combining very deep residual learning with channel-wise feature reweighting. This work can be seen as a direct response to two remaining challenges in earlier regression models: the difficulty of training very deep networks for SR, and the inefficiency of treating all feature channels as equally informative.

The RCAN architecture is more complex than SRCNN and VDSR. It consists of shallow feature extraction, deep feature extraction through a residual-in-residual (RIR) structure, an upscaling module, and a final reconstruction layer [68]. Its structure contains multiple residual groups, each of which contains several residual channel attention blocks. Both long and short skip connections are used to improve optimization and facilitate information flow. This hierarchical residual organization allows low-frequency information to bypass the main transformation path efficiently, enabling the deeper core to focus more strongly on high-frequency detail reconstruction.

The second major contribution is the channel attention mechanism. Instead of processing all channels uniformly, RCAN adaptively re-scales channel-wise responses by modeling inter-dependencies among them [68]. This is important because not all feature channels contribute equally to accurate reconstruction. By re-weighting channels dynamically, the network improves representational selectivity and emphasizes more informative responses.

RCAN is trained with an ℓ_1 reconstruction loss [68], continuing the trend established in stronger regression models beyond the earlier ℓ_2 era. The use of ℓ_1 helps preserve optimization stability while often yielding somewhat sharper reconstructions than mean squared error.

At the same time, RCAN does not eliminate the core limitations of regression-

based SR. Although it reconstructs sharper and more accurate details than earlier deterministic models, it remains a supervised regression method optimized under pixel-space fidelity.

3.1.8 Comparative discussion

With each other, [61], [10, 9], [32], [29], [35], and [68] reveal a clear and meaningful progression in the SISR literature.

First, the field moves from explicit priors to implicit learned mappings.

Second, the field moves from shallow convolution to deep residual regression. SRCNN establishes the end-to-end CNN paradigm, but its shallow structure limits context modeling. VDSR in [32] shows that much deeper models become practical and more effective when residual learning is used. EDSR in [35] then refines this idea by tailoring residual architectures specifically for low-level restoration rather than borrowing them unchanged from high-level vision tasks.

Third, the literature evolves from generic deep feature extraction to selective feature modulation. RCAN in [68] makes this progression explicit by demonstrating that channel-wise attention can improve the representational efficiency of very deep regression networks. This indicates that increasing network depth alone is insufficient.

More critically, this progression shows that architectural improvements alone do not fully solve the perceptual limitation of regression-based SR. SRCNN, VDSR, EDSR, and RCAN progressively improve feature extraction, optimization, and representation capacity. However, they remain mainly deterministic reconstruction models. Therefore, when they are trained primarily with pixel-wise losses, they still tend to favor fidelity-oriented outputs rather than truly recovering uncertain high-frequency textures.

Fourth, the literature recognizes that distortion minimization and perceptual quality are not identical objectives. This issue is articulated most directly in [29], which shows that feature-space perceptual losses can produce more visually convincing results than pure pixel-wise optimization. This insight is especially important because it motivates the later emergence of GAN-based and DM methods. At the same time, it also raises a critical caution for fidelity-sensitive applications: perceptual realism does not automatically imply structural correctness.

This distinction is particularly important for medical SR. In natural-image SR, visually plausible texture may be acceptable even when it is not exactly faithful to the ground truth. In medical imaging, however, the same behavior can be problematic, because artificial edges, textures, or anatomical-looking details may affect interpretation. Therefore, perceptual improvement must be considered together with structural fidelity, stability, and diagnostic reliability.

These things are highly relevant. Regression-based methods remain attractive because of their stability, determinism, and relatively controllable optimization behavior. They are especially appealing in domains such as medical imaging, where reliability and structural faithfulness are essential. However, the same deterministic losses that

make them stable also tend to average over multiple plausible HR solutions, which suppresses fine stochastic textures and can lead to overly smooth reconstructions. Thus, the regression literature simultaneously provides the methodological foundation and exposes the perceptual limitation that later GAN-based and DM methods attempt to address.

In summary, [16] provides the medical motivation and fidelity-sensitive application context, [61] represents a key pre-deep-learning learning-based precursor, and [10, 9, 32, 35, 68] define the major stages of regression-based deep SISR. Meanwhile, [29] gives a conceptual turning point by clarifying why pixel-wise regression alone is often insufficient for perceptually convincing reconstruction. Together, these works establish the foundation upon which the later GAN-based and diffusion-based super-resolution literature is built.

3.2 GAN-Based SR

While regression-based SR methods provide stable and often highly accurate reconstructions in terms of distortion metrics, they remain limited by their deterministic pixel-wise objectives. In particular, when the mapping from LR to HR is highly ambiguous, optimization with ℓ_1 or ℓ_2 losses tends to average over multiple plausible HR solutions, producing overly smooth textures. This limitation motivated the introduction of adversarial learning into super-resolution, where the objective is no longer only to minimize pixel-wise discrepancy, but also to encourage reconstructed images to lie closer to the manifold of realistic HR images. In this context, the GAN-based literature represents a major conceptual shift from *distortion-oriented reconstruction* to *perceptually driven generation*.

3.2.1 Adversarial learning

The theoretical foundation of GAN-based SR is the adversarial learning framework introduced by [15]. In the standard GAN setting, two networks are trained simultaneously: a generator G, which maps a latent variable or conditioned input to a synthesized sample, and a discriminator D, which attempts to distinguish real samples from generated ones. The original training formulation is a two-player minimax game in which D maximizes its discrimination accuracy while G attempts to generate outputs that fool D [15].

From an architectural viewpoint, the original GAN framework is intentionally general: it does not prescribe a specific image SR design, but rather defines a learning principle that can be embedded into later task-specific systems. Its core value for the SR literature is therefore not a particular SR architecture, but the introduction of an adversarial objective that can function as a learned, adaptive perceptual criterion.

The main strength of [15] in the present context is that it explains why adversarial learning can move generated outputs toward the target data distribution in a way that fixed pixel-wise losses cannot. This is especially important for SR because the missing high-frequency details are not uniquely determined by the LR input.

3.2.2 SRGAN: the key turning point in perceptual SISR

The most influential transition from adversarial learning theory to practical super-resolution is SRGAN, proposed in [33]. This work is a major landmark in SISR because it explicitly argues that optimizing only for mean squared error yields high PSNR but poor perceptual realism, and it proposes a GAN-based framework to address this mismatch.

Paper [33] employs a deep residual generator and a convolutional discriminator. The generator is built around a very deep residual network with residual blocks, skip connections, and two learned sub-pixel upsampling layers for 4× enlargement. The discriminator is a convolutional classifier designed to distinguish generated SR images from real HR images. In this way, the generator is trained not only to reconstruct, but also to produce outputs that the discriminator considers realistic.

An important contribution of [33] is its *perceptual loss*. Instead of relying solely on a pixel-wise MSE objective, SRGAN combines a content loss with an adversarial loss. The content term is defined in feature space using VGG features, rather than in raw pixel space, while the adversarial term encourages the output to lie closer to the natural image manifold [33]. This design directly operates the conceptual shift by perceptual-loss literature and turns it into a practical SR framework.

The strengths of SRGAN are substantial. First, it establishes GAN-based perceptual SR as a credible alternative to regression-only methods. Second, it demonstrates that outputs with lower PSNR can nevertheless be judged as more photo-realistic by human observers. Third, it defines an influential template for later perceptual SR systems: a strong CNN generator combined with feature-based content loss and adversarial supervision.

[33] also exposes the fundamental trade-off of adversarial SR. Although SRGAN improves perceptual sharpness and texture realism, it may also introduce details that are visually plausible but not strictly faithful to the ground-truth image.

This is problematic in fidelity contexts such as medical imaging, where hallucinated texture may be unacceptable.

3.2.3 Conditional adversarial refinement for SISR

A later refinement of adversarial SR is presented in [47], which proposes an SISR method based on conditional GANs, referred to as SRCGAN. This work is important because it attempts to improve the stability and fidelity of GAN-based SR by making the adversarial decision more explicitly conditioned.

[47] keeps the generator–discriminator structure of GAN-based SR, but modifies the discriminator by introducing the ground-truth HR image as a conditional variable. In effect, the discriminator evaluates the generated SR image relative to the target HR reference, rather than operating as a purely unconditional real/fake classifier. The generator itself is a deep residual network that processes the LR image directly, removes batch normalization from its residual blocks following the design philosophy of EDSR, applies residual scaling to stabilize training, and performs upsampling

through sub-pixel convolution for 4× SR [47].

The loss design in [47] is also more composite than that of SRGAN. The proposed perceptual loss combines MSE loss, VGG feature loss, total variation loss, and adversarial loss. This design reflects an attempt to balance fidelity, perceptual sharpness, smoothness regularization, and adversarial realism in a more controlled manner than a simple GAN-plus-content-loss setting.

The strengths of [47] are twofold. First, by conditioning the discriminator on the HR target, the method attempts to reduce the degree of unconstrained adversarial freedom and thereby improve training stability. Second, by incorporating multiple complementary loss terms, it seeks a more balanced compromise between quantitative metrics and visual quality. Relative to SRGAN, this can be interpreted as a move toward a more supervised and constrained adversarial SR formulation.

Its limitations, however, should be noted. Although the conditioning mechanism can improve guidance, the method still remains within the GAN framework and therefore suffers from adversarial instability to a meaningful extent.

Moreover, because the objective still includes an adversarial term, the possibility of generating visually plausible but not strictly correct details is not fully removed. In addition, the loss design becomes more complex, and the interaction among multiple weighted losses can make optimization sensitive to hyperparameter selection.

3.2.4 GAN-based SR in medical imaging

An important medical extension of adversarial SR is [63], which introduces GAN-CIRCLE for CT SR. It applies GAN-based SR directly to a fidelity-sensitive medical imaging modality, where image enhancement is meaningful only if structural and anatomical reliability are preserved.

[63] proposes a CT SR framework built on GANs and constrained by an *identical*, *residual*, and *cycle learning* ensemble. The method is designed as a semi-supervised deep learning approach for recovering HR CT images from LR inputs while promoting denoising, deblurring, and structural preservation. A key aspect of the method is the inclusion of cycle-consistency-style constraints together with residual learning and adversarial training, so that the mapping from LR to HR is not driven by realism alone, but also by additional structural consistency constraints [63].

From an architectural perspective, [63] goes beyond a plain SRGAN-style formulation by combining adversarial learning with multiple task-specific constraints intended to preserve fidelity in CT reconstruction. The available description also indicates the use of CNN-based feature extraction, residual learning, and network-in-network style compression modules to improve efficiency relative to simply scaling depth and complexity.

The strengths of [63] are important for a medical-imaging contexts. Unlike natural-image SR papers that primarily prioritize visual realism, GAN-CIRCLE explicitly addresses a domain in which anatomical detail and structural consistency matter. The addition of identical, residual, and cycle-related constraints can be interpreted as

an attempt to reduce the risk of unconstrained hallucination and to make adversarial SR more acceptable in clinical-style reconstruction settings. This makes [63] a meaningful bridge between perceptual GAN-based SR and medically constrained image restoration.

At the same time, the limitations remain significant. [63] is still adversarial and therefore cannot fully escape the core risks of GAN-based generation. In medical imaging, even small visually plausible but incorrect structures may have consequences.

3.2.5 Training stability

A key reference for understanding the weaknesses of GAN-based SR is [30]. Although this paper is not itself an SR method, it is relevant because it addresses a central practical issue that affects all adversarial SR systems: instability in GAN optimization.

The main contribution of [30] is a training methodology in which both the generator and discriminator are grown progressively from low spatial resolution to higher resolution. Instead of learning the entire HR image distribution at once, the networks first learn coarse image structure at small resolutions and then gradually incorporate finer details as training proceeds. The paper also introduces several training refinements, including smooth layer fade-in during resolution transitions, batch standard deviation for improving variation, equalized learning rate, and generator-side feature normalization [30].

The strength of [30] in the present chapter is primarily interpretive. It provides strong evidence that many GAN failures are not merely incidental implementation problems, but arise from the intrinsic difficulty of adversarial optimization, especially at high resolution. Since SR tasks often require fine high-frequency synthesis, this insight is directly relevant to GAN-based SR: the sharper and more detailed the desired output, the harder stable adversarial training may become.

Its limitation for the present thesis is that it is not a SR paper and does not solve the LR-to-HR inverse problem directly. Nevertheless, it is extremely useful as a supporting reference because it helps explain why GAN-based SR methods such as [33], [47], and [63] can produce impressive perceptual results while still remaining harder to train and less predictable than regression-based approaches.

3.2.6 Comparative discussion of GAN-based SR methods

Taken together, [15], [33], [47], [63], and [30] define a coherent progression in the adversarial SR literature.

First, the progression moves from general adversarial theory to task-specific perceptual SR. The foundational adversarial framework in [15] establishes the minimax principle and the learned realism prior, but it is [33] that translates this idea into a practical SISR model through SRGAN. In this sense, SRGAN is the key paper that turns GANs from a general generative paradigm into a concrete solution for perceptual SR.

Second, the literature evolves from unconstrained adversarial realism toward more guided adversarial supervision. In [33], the discriminator mainly encourages photo-realism, while in [47] the discriminator is conditioned on the HR target to provide more explicit guidance. This makes [47] more tightly coupled to supervised reconstruction than the earlier SRGAN formulation. Similarly, [63] extends the idea further in a medical context by supplementing adversarial learning with identity, residual, and cycle-related constraints that aim to preserve structure more strongly.

However, these additional constraints do not fully remove the central limitation of GAN-based SR. The adversarial objective still encourages distribution-level realism rather than strict measurement fidelity. As a result, even when conditioning, cycle consistency, or residual constraints are used, the generated details may remain visually plausible but not necessarily supported by the LR input. This limitation is especially important in medical imaging, where plausible but incorrect structures can reduce clinical trustworthiness.

Third, the literature makes increasingly explicit the tension between perceptual quality and faithful reconstruction. SRGAN in [33] prioritizes perceptual realism and demonstrates that this can improve visual texture quality. However, this same strength becomes a weakness in fidelity-critical domains. The conditional and constrained formulations in [47] and [63] can therefore be interpreted as attempts to reduce the risk of hallucination while preserving some of the perceptual gains of adversarial learning.

Finally, [30] shows why this balance is so difficult in practice: GANs are not only conceptually powerful, but also optimization sensitive. Their impressive visual results are inseparable from nontrivial training difficulties. This makes adversarial SR fundamentally different from regression-based SR. Regression models are typically more stable but more conservative; GAN-based models are often more visually compelling but also more fragile and less strictly faithful.

Thus, the limitation of GAN-based SR is not only a matter of training difficulty, but also a matter of reconstruction control. The discriminator encourages outputs that look realistic according to the learned data distribution, but it does not guarantee that every generated high-frequency structure is justified by the LR observation. This distinction is central for medical SR, where the acceptability of a reconstruction depends not only on visual sharpness but also on whether the recovered details are anatomically reliable.

From the perspective of the present thesis, this comparison is especially important. GAN-based SR demonstrates that perceptual sharpness can be improved dramatically beyond classical pixel-wise regression. However, it also reveals a core limitation: the very mechanism that encourages realistic texture can introduce uncertainty, instability, and potentially untrustworthy detail. This is precisely why GANs are powerful but not always ideal for applications such as medical imaging, where structural fidelity is often more important than visually plausible synthesis.

3.2.7 Summary of the role of GANs

In summary, [15] provides the adversarial learning foundation, [33] establishes the canonical GAN-based perceptual SISR framework, [47] introduces a more guided conditional adversarial formulation, [63] extends adversarial SR into medically relevant CT reconstruction with added structural constraints, and [30] explains why stabilization techniques are essential in adversarial image generation. Collectively, these works show that GAN-based super-resolution is a major advance in perceptual quality, but also that it brings nontrivial risks in optimization stability and reconstruction faithfulness. This body of literature therefore forms the critical intermediate stage between distortion-oriented regression methods and the later diffusion-based alternatives.

3.3 DM-based SR

DM-based models represent a major recent development in generative SR. While GAN-based methods improved perceptual sharpness relative to regression models, they remained constrained by adversarial instability, potential mode collapse, and the risk of producing visually plausible but insufficiently controlled details. DMs emerged as an alternative generative framework that can produce high-quality samples through a gradual denoising process, while typically offering more stable optimization than adversarial learning. In the context of SR, this makes diffusion especially important: it provides a probabilistic mechanism for modeling the one-to-many nature of the LR-to-HR mapping without relying on a discriminator.

3.3.1 Foundational diffusion formulation: DDPM

The modern diffusion literature is built on the denoising diffusion probabilistic model (DDPM) introduced in [22]. This work establishes the core probabilistic formulation that later diffusion-based SR methods inherit and adapt.

The main idea in [22] is to define a forward Markov process that gradually corrupts a data sample by adding Gaussian noise over many steps, until the sample is transformed into near-isotropic noise. A neural network is then trained to approximate the reverse process, i.e., to iteratively remove noise and reconstruct a clean sample from a noisy one. In practical terms, this creates a generative model in which sampling proceeds by starting from Gaussian noise and repeatedly denoising it through a learned reverse chain [22].

From an architectural perspective, the contribution of [22] is primarily a probabilistic framework rather than a task-specific SR architecture. Its significance lies in the formulation of diffusion as a trainable latent-variable model with a Markov noising process and a learned reverse denoising process. An especially important practical detail is the simplified training objective based on noise prediction, which makes the model effective to optimize while preserving high sample quality.

The loss design in [22] is rooted in a weighted variational bound, together with a simplified denoising objective that predicts the added noise at each step [22]. This is a key point for the later SR literature, because it shows that one can train a powerful

generative model without adversarial competition and without directly regressing only a single deterministic HR estimate.

The strengths of [22] are substantial. First, it demonstrates that diffusion models can generate very high-quality images, challenging the earlier dominance of GANs in perceptual image synthesis. Second, the framework avoids the discriminator-based minimax instability of GANs. Third, the stepwise denoising process provides a natural way to model multimodal outputs, which is highly relevant for SISR because one LR image may correspond to multiple plausible HR reconstructions.

Its limitations are equally important. The most obvious drawback is computational cost, because sampling needs many iterative denoising steps, inference is much slower than standard feed-forward CNN regression and often slower than GAN-based generation. In addition, [22] is a general unconditional image generation framework, not a SR method by itself. Therefore, while it provides the conceptual foundation, it does not directly solve the LR-conditioned SR problem.

3.3.2 Improved DDPM: better likelihoods and faster sampling

A major refinement to the basic DDPM framework is introduced in [46]. This work is important because it addresses two practical weaknesses of the original DDPM formulation: limited efficiency and the gap between strong sample quality and competitive likelihood-based modeling.

The central contribution of [46] is to improve the DDPM framework through a set of relatively simple but highly effective changes. In particular, the paper introduces learned reverse-process variances and a hybrid training objective that combines the variational lower bound with the simplified objective used in [22] [46]. An important result is that learning the reverse variances allows the model to generate good samples in far fewer steps, in some cases reducing the number of sampling passes by roughly an order of magnitude relative to the earlier formulation.

[46] still operates within the DDPM family rather than proposing a new SR-specific network. Its significance is algorithmic and practical. It shows how the core diffusion framework can be made more efficient and more scalable while preserving sample quality.

The loss function in [46] is especially relevant for a rigorous literature review. Rather than relying only on the simplified denoising loss, the paper studies a hybrid objective that improves likelihood performance while maintaining strong sample quality [46]. This is important because it reveals that diffusion models can be improved not only through architectural changes, but also through more careful objective design and variance modeling.

The strengths of [46] are threefold. First, it improves the practical usability of diffusion models through faster sampling. Second, it demonstrates stronger log-likelihood behavior, reinforcing the argument that diffusion models cover the data distribution more completely than many competing generative approaches. Third, it shows that diffusion performance scales smoothly with model size and compute,

which helps explain why diffusion became increasingly influential in high-fidelity image generation.

However, important limitations remain. Even with faster sampling, diffusion is still iterative and comparatively expensive at inference time. Thus, although [46] reduces one of the major practical barriers identified in [22], it does not eliminate the fundamental computational burden of repeated denoising steps. Moreover, like [22], this paper is still not an SR method directly. It strengthens the diffusion foundation that later conditional restoration methods can build upon.

3.3.3 SRDiff: the first diffusion-based SISR model

The key paper that brings diffusion directly into SISR is [34], which proposes SRDiff. This work is particularly important because it explicitly positions diffusion as an alternative to three existing families of SISR methods: PSNR-oriented regression models, GAN-driven methods, and flow-based methods.

Architecturally, [34] formulates SISR as a conditional diffusion process. The method uses a pre-trained LR encoder to extract hidden conditions from the LR input, and then employs a conditional noise predictor to iteratively recover the HR image through the reverse diffusion process [34]. In contrast to GAN-based SR, there is no discriminator. Instead, generation is driven by a Markov denoising chain conditioned on the LR image.

A key design in [34] is the use of *residual prediction*. Rather than modeling the HR image directly in the diffusion chain, the method introduces the difference between HR and LR images as the effective reconstruction target in the early stage, so that the model focuses more directly on restoring missing high-frequency details [34]. This is especially relevant from the perspective of your thesis, because it shows that even within a generative framework, residual learning remains a useful mechanism for focusing reconstruction capacity on informative details.

In terms of optimization, SRDiff is trained with a variant of the variational bound on the data likelihood [34]. This is a key contrast with GAN-based methods. Instead of balancing adversarial and perceptual losses, SRDiff uses a single diffusion-style learning objective, which the authors present as easier to train and less prone to mode collapse than GAN-based SR. The paper argues that diffusion avoids the over-smoothing of PSNR-oriented regression, the mode collapse of GANs, and the large model footprint of flow-based approaches [34].

The strengths of [34] are significant. First, it is the first explicit diffusion-based SISR model, making it a landmark in the SR literature. Second, it demonstrates that diffusion can produce diverse and realistic SR results from a single LR input, directly addressing the ill-posed one-to-many nature of SISR. Third, it is presented as easier and more stable to train than GAN-based SR, since it does not require adversarial competition or an extra discriminator. Fourth, compared with flow-based models, it avoids strong architectural invertibility constraints and can achieve a smaller footprint.

At the same time, SRDiff also has limitations. Its iterative reverse process remains

computationally heavier than deterministic feed-forward regression models and typically heavier than single-pass GAN generators at inference time. In addition, while diffusion can reduce adversarial instability, it still introduces artifacts into the reconstruction process. This is useful for diversity, but it can also complicate strict reproducibility or consistency if a single deterministic output is preferred. Furthermore, although SRDiff improves perceptual diversity and detail, the broader question of structural trustworthiness remains important in fidelity-critical domains such as medical imaging.

3.3.4 Diffusion models for image SR

The survey in [41] is highly valuable because it places diffusion-based SR within the broader evolution of the SR field and explicitly compares diffusion with earlier generative approaches. This paper is especially useful for a thesis literature review because it synthesizes not only the strengths of diffusion, but also its open problems.

According to [41], diffusion models have substantially reshaped image SR by closing the gap between generated image quality and human perceptual preferences. The survey emphasizes that diffusion models can produce very high-quality samples and often exceed earlier generative approaches in realism, while also being easier to train than GANs in many settings [41]. At the same time, the paper carefully notes that diffusion-based SR introduces new challenges, including high computational costs, comparability issues, lack of explainability, color shifting, and other practical concerns.

From a methodological standpoint, [41] is not a single model paper but a structured review. Its contribution is therefore comparative and analytical rather than architectural. It organizes the field by discussing diffusion foundations, conditioning strategies, sampling improvements, alternative input domains, corruption spaces, and domain-specific applications, including medical imaging [41]. This makes it particularly useful for connecting the technical diffusion literature to the wider SR ecosystem.

The main strength of [41] is that it clearly explains why diffusion became so influential in SR: it offers a compelling compromise between perceptual realism and training stability, especially when compared with GANs. It also explicitly highlights that diffusion-based SR is now central to current research trends. At the same time, the survey does not present diffusion as a complete solution. Instead, it stresses that computational cost, explainability, and evaluation remain major unresolved issues [41].

For the present thesis, this perspective is important. It supports the claim that diffusion models are highly competitive in perceptual SR, but it also reinforces why they may still be sub-optimal in settings where inference cost, deterministic behavior, or strict structural control are critical.

3.3.5 Comparative discussion of diffusion SR

All together, [22], [46], [34], and [41] show a clear progression in diffusion-based super-resolution.

First, the literature moves from general diffusion generation to *practical diffusion refinement*. The DDPM framework in [22] establishes the basic generative mechanism of iterative noising and denoising, while [46] improves its practicality through learned variances, better likelihood optimization, and faster sampling. In this sense, [46] does not replace [22], but makes its core framework more usable.

Second, the literature moves from unconditional image synthesis to *conditional inverse problems*. Both [22] and [46] are foundational generative papers, but they do not directly solve SR. SRDiff in [34] is the crucial transition that adapts diffusion to the LR-conditioned SISR setting. It shows that diffusion is not only a general-purpose image generator, but also a viable framework for solving the ill-posed LR-to-HR reconstruction problem.

Third, the literature clarifies diffusion's position relative to earlier SR families. Compared with regression-based methods, diffusion can avoid the strong averaging effect of pure ℓ_1/ℓ_2 training and can generate richer high-frequency details. Compared with GAN-based methods, diffusion avoids adversarial competition and is generally less vulnerable to mode collapse. Compared with flow-based methods, diffusion avoids strict invertibility constraints and may require a smaller model footprint, as emphasized in [34]. However, compared with all of these alternatives, diffusion usually needs a heavier sampling cost because of its iterative reverse process.

Another point is that diffusion-based SR changes the nature of the reconstruction problem from deterministic estimation to conditional sampling. This is beneficial when multiple HR images are plausible for the same LR input, because the model can represent uncertainty more naturally than point-estimate regression. However, in fidelity-sensitive applications, this same sampling behavior may reduce reproducibility and make it harder to guarantee that generated high-frequency details are strictly supported by the LR measurement. Therefore, DMs improve perceptual realism and probabilistic modeling, but they also introduce challenges in computational efficiency, consistency, and clinical controllability.

Fourth, the survey perspective in [41] makes explicit that diffusion is both a major advance and a source of new challenges. In other words, diffusion does not simply solve the problems of GANs. It replaces some of them with different trade-offs, especially computational complexity, explainability concerns, and evaluation difficulties.

In summary, [22] provides the probabilistic foundation of modern diffusion models, [46] improves that foundation through more efficient and scalable sampling, [34] brings diffusion directly into SISR through the first dedicated diffusion-based SR model, and [41] provides a recent synthesis of the field's strengths, limitations, and emerging directions. Collectively, these works establish diffusion-based SR as a powerful alternative to GAN-based perceptual SR. It can generate diverse and realistic reconstructions with more stable training, but it does so at the cost of heavier computation and continued concerns about control, consistency, and explainability.

3.4 Real-world and blind SR

Although the earlier SR literature achieved strong progress under synthetic degradations such as bicubic downsampling, practical deployment requires models that operate under complex, unknown, and often highly mismatched real-world degradations. In such settings, the LR image may be affected not only by downsampling, but also by sensor limitations, optics, blur, noise, compression, demosaicing artifacts, transmission errors, and other unknown image-processing factors. As a result, the real-world and blind SR literature shifts the focus from idealized benchmark performance toward robustness under uncertain degradations, broader applicability, and more realistic restoration objectives.

3.4.1 Real-world SR as a distinct problem setting

A useful high-level reference for this shift is the review in [4], which argues that performance on synthetic data often overestimates real-world super-resolution capability. The paper frames real-world single-image super-resolution (RSISR) as a practically distinct problem rather than a simple extension of conventional SISR. In particular, it emphasizes that the domain gap between simulated low-resolution images and realistic LR observations is a central reason why many strong synthetic-data methods degrade significantly in practice [4].

From a methodological standpoint, [4] is a survey rather than a single reconstruction model. Its contribution is therefore organizational and comparative. The review summarizes datasets, evaluation metrics, and four broad categories of RSISR methods: degradation-modeling-based approaches, image-pairs-based approaches, domain-translation-based approaches, and self-learning-based approaches [4]. This taxonomy is useful because it makes clear that real-world SR is not solved by a single modeling choice; instead, different approaches attempt to address the same degradation uncertainty from different assumptions and data regimes.

The strength of [4] is that it provides a broad practical framing for modern SR research. It highlights why degradation realism, dataset construction, computational efficiency, and evaluation criteria all become more important once the field moves beyond synthetic bicubic benchmarks. It also identifies unresolved challenges, including realistic training and testing datasets, dedicated RSISR models, and appropriate performance assessment [4].

It says why real-world SR requires a broader and more practical viewpoint than the earlier synthetic-data literature.

3.4.2 Blind SR and the challenge of unknown degradation

A related but more technically focused perspective is provided by the survey in [36], which addresses *blind* image SR. While the term "blind" does not have a single universally fixed definition, the paper adopts a broad interpretation in which the degradation of the LR input is unknown or only partially specified. This is highly relevant to real-world SR because practical degradations are rarely known exactly at test time.

The central contribution of [36] is a taxonomy that categorizes blind SR methods according to how they model degradation and what data are available for solving the SR problem. The paper distinguishes among approaches based on explicit degradation modeling, degradation-aware methods that can be adapted to unknown degradations, and more challenging unsupervised or self-supervised settings in which paired supervision may be absent [36]. This is an important conceptual step because it clarifies that blind SR is not a single technique, but a family of strategies built around different assumptions about degradation knowledge and supervision.

From a modeling perspective, [36] also explains the common mathematical formulations used in blind SR. In particular, it discusses the classical degradation model involving blur, downsampling, and noise, while also arguing that real-world degradations are often too complex to be fully captured by such explicit formulations [36]. This paper connects blind SR directly to the mismatch between idealized degradation assumptions and practical image formation.

The strength of [36] lies in its analytical clarity. It not only surveys methods, but also explains why many blind SR systems still fail on truly arbitrary real images: even methods labeled "blind" often remain tied to some assumed degradation family. The paper therefore makes explicit that current blind SR methods, while more practical than non-blind bicubic SR, still do not fully solve the open-ended real-world degradation problem [36].

Its limitation, however, is similar to that of other surveys: it synthesizes the field rather than proposing one unified new model. In the present thesis, [36] is therefore best used to contextualize why robustness to unknown degradation remains difficult, and why practical SR must be evaluated beyond narrow synthetic assumptions.

3.4.3 Task diversity in real-world SR: TGSR

A more model-specific modern contribution is given in [67], which reinterprets real-world image SR from a multi-task learning perspective. This is a particularly interesting development because it moves beyond the usual emphasis on degradation simulation alone and instead analyzes the *optimization structure* of real-SR training.

The key idea in [67] is that a conventional real-SR model is effectively trained to solve a very large number of distinct degradation tasks using one shared network. Each sampled degradation can be viewed as a separate task, and training across a broad degradation space therefore becomes a large-scale multi-task learning problem [67]. Under this interpretation, the authors argue that real-SR suffers from task competition or task conflict: some degradation tasks dominate the learning process and suppress performance on others.

[67] does not primarily introduce a radically new backbone for feature extraction. Instead, its main innovation lies in the training framework. The proposed method, Task Grouping based real-SR (TGSR), first identifies "unsatisfactory" degradation tasks—that is, those on which the shared real-SR network underperforms relative to task-specific fine-tuned models. It then groups these tasks into multiple task groups and continues fine-tuning the pre-trained real-SR network using these groups in

order to reduce negative transfer and improve overall performance [67].

In terms of optimization, the paper reframes the total real-SR objective as the sum of losses over many degradation tasks, and its contribution lies in how training is reorganized rather than in proposing a new low-level pixel loss or adversarial loss. The method uses a gradient-update-based performance indicator to identify problematic tasks and a performance-improvement score to form task groups more efficiently than exhaustive pairwise task-affinity estimation [67].

The strength of [67] is that it introduces a new and convincing explanation for why real-SR remains difficult even when the degradation model is broad: the issue is not only degradation realism, but also optimization competition across many heterogeneous tasks. This is an important insight because it suggests that simply enlarging the degradation space is not always enough; one must also manage how the model allocates capacity across that space.

Its limitations should also be noted. The method still depends on sampling the degradation space adequately, and therefore its success remains tied to how well the chosen degradation generation process reflects practical conditions [67]. In addition, because it is built around grouping and fine-tuning, it adds training complexity and workflow overhead compared with a single straightforward end-to-end training pipeline. Nevertheless, [67] is valuable because it advances the real-SR literature from pure degradation modeling toward a more optimization-aware perspective.

3.4.4 Semantics-aware real-world SR with diffusion priors

A very recent generative direction is represented by [58], which proposes SeeSR, a semantics-aware framework for real-world image super-resolution. This work is particularly notable because it combines real-world SR with large pretrained text-to-image diffusion priors, while explicitly addressing the problem of semantic inconsistency under heavy degradation.

The central motivation of [58] is that severe real-world degradation can destroy local structures and make semantic interpretation ambiguous. As a result, even strong generative priors may reconstruct visually detailed outputs that contain semantic errors. To address this, the paper proposes a semantics-aware strategy that uses degradation-aware semantic prompts to better preserve semantic fidelity during real-world SR [58].

Architecturally, SeeSR is a two-stage framework. In the first stage, a degradation-aware prompt extractor (DAPE) is trained to produce reliable semantic information from degraded LR inputs. In the second stage, the extracted prompts are used to guide a pretrained text-to-image diffusion model. The method uses both hard prompts (tag-style text prompts) and soft prompts (feature representations), allowing the generative model to receive both explicit semantic cues and additional embedding-level control [58]. During inference, the LR image is also incorporated into the initial sampling noise to reduce the diffusion model's tendency to generate excessive random details.

The loss design in [58] is also worth noting. In the DAPE training stage, the model

aligns the LR branch with the HR branch using a combination of mean squared error for representation alignment and cross-entropy loss for logits alignment [58]. This is an elegant design because it explicitly trains the prompt extractor to be degradation-aware rather than relying on off-the-shelf semantic extraction from heavily corrupted LR images.

The strengths of [58] are great. First, it directly addresses a real limitation of diffusion-prior-based real-SR: strong generative power does not guarantee semantic correctness. Second, by introducing degradation-aware prompt extraction, the method improves control over the generative process and better aligns detailed synthesis with the content actually present in the LR input. Third, it represents a modern trend in which real-world SR increasingly leverages large pretrained generative models rather than relying only on task-specific CNNs or GANs.

At the same time, the limitations are also important. Because the method depends on a pretrained text-to-image diffusion model, it inherits the computational cost and inference complexity of diffusion-based generation. In addition, while semantics-aware prompting improves control, the reconstruction still remains partly generative and therefore cannot be assumed to be strictly deterministic or guaranteed perfectly faithful in all details. Thus, [58] is a strong modern example of how real-world SR is moving toward richer generative priors, but it also reflects the continued trade-off between realism, control, and computational efficiency.

3.4.5 Comparative discussion of real-world and blind SR

[4], [36], [67], and [58] show a meaningful progression in the practical super-resolution literature.

First, the field moves from recognizing the domain gap to formalizing degradation uncertainty. The review in [4] makes clear that synthetic benchmark success does not directly translate to real-world success, while [36] clarifies the blind SR problem as one of unknown degradation and organizes the methodological landscape accordingly.

Second, the literature evolves from degradation modeling alone to optimization-aware training. Much of the earlier real-SR work emphasizes more realistic degradation simulation, but [67] shows that even if the degradation space is broad, a single shared model may still underperform because different degradation tasks compete during training. This provides a deeper explanation for why real-SR remains difficult.

Third, the literature moves from restoring under unknown degradation to controlling generative restoration with semantics. The SeeSR framework in [58] demonstrates that, in highly degraded real-world scenarios, perceptual detail generation alone is insufficient; the reconstruction must also remain semantically coherent. This is a particularly modern concern, reflecting the integration of large generative priors into image restoration.

From the perspective of the present thesis, these works are highly relevant because they reinforce the distinction between benchmark-oriented SR and practically useful SR. They show that real deployment requires attention not only to architecture and loss design, but also to degradation realism, uncertainty, task diversity, semantic

correctness, and computational constraints. In other words, once SR is considered in real-world settings, the problem becomes broader than simply improving PSNR or perceptual texture on synthetic test sets.

In summary, [4] provides a practical overview of RSISR and its major methodological categories, [36] formalizes the blind SR problem and clarifies the role of degradation uncertainty, [67] introduces a multi-task interpretation of real-SR and addresses task competition through task grouping, and [58] represents a recent semantics-aware, diffusion-prior-based approach for real-world SR. Collectively, these works show that modern practical SR is increasingly shaped by unknown degradations, broader restoration objectives, and the need for stronger control over real-world reconstruction.

3.5 Medical Imaging Context, Supporting DL Foundations, and Concluding Synthesis

The final perspective needed in this literature review is the medical-imaging context in which super-resolution is ultimately expected to be useful. While the previous subsections reviewed the core methodological progression of SISR—from classical and regression-based approaches to GAN-based, diffusion-based, and real-world restoration methods—the practical motivation of the present thesis is not merely to improve benchmark image quality, but to do so in a domain where structural reliability and clinical usefulness are especially important. For this reason, it is helpful to conclude the literature review by briefly positioning recent medical CT-related deep learning work and the broader deep learning components that indirectly support this thesis.

3.5.1 Medical CT SR relevance: a task-specific example

The most directly relevant to SR is [6], which proposes a dedicated lung CT SR method. This paper is important because it demonstrates that the SR problem in medical imaging is not simply a direct copy of natural-image SR, but can motivate architectures tailored to the frequency characteristics and diagnostic demands of CT images.

Architecturally, [6] proposes a double-path network for lung CT image SR, referred to as DRIDSR. The model explicitly separates reconstruction into a low-frequency path and a high-frequency path. The low-frequency branch uses shallow convolutional processing to preserve coarse image content, while the high-frequency branch uses a residual information distillation module (RIDM) to recover finer details more effectively. In addition, the model introduces a gradient identity path based on the Sobel operator so that image-gradient information can guide restoration of edges and high-frequency structures [6]. The network then fuses the outputs of the two paths and uses pixel shuffle for upsampling.

In terms of loss design, [6] uses an ℓ_1 reconstruction loss between the generated SR image and the target HR image [6]. This keeps the optimization within the stable

deterministic regression family, while the architectural design attempts to recover sharper medical details by explicitly separating low- and high-frequency processing.

The strengths of [6] are closely aligned with the goals of this thesis. First, it is medical-domain-specific rather than purely natural-image-oriented. Second, it recognizes that high-frequency detail in CT deserves special treatment rather than being mixed uniformly with low-frequency content. Third, it remains within a non-adversarial regression framework, which is attractive in fidelity-sensitive domains. At the same time, its limitations are also clear: despite its frequency-aware design, it is still trained as a supervised deterministic reconstruction network, so it remains subject to the broader limitations of regression-based SR, including controlled degradation assumptions and potentially conservative texture recovery. However, [6] is an important supporting reference because it shows that medical SR benefits from architectures designed around domain structure rather than generic image restoration alone.

3.5.2 Medical DL context beyond SR

[17] focuses on CT-based COVID-19 detection using deep neural networks. Its value in this chapter is therefore contextual rather than methodological: it reinforces why robust CT image analysis has been a major research priority and why data diversity, efficiency, and explainability matter in medical imaging.

The paper introduces COVID-Net CT-2, an enhanced family of CNN-based models for COVID-19 detection from chest CT images, trained on large, multinational benchmark datasets. It emphasizes increased data quantity and diversity, proposes a lightweight architecture (COVID-Net CT S), and performs explainability-driven performance validation so that the model decisions can be checked against clinically relevant visual cues rather than irrelevant artifacts [17]. The training setup uses cross-entropy loss with ℓ_2 regularization and standard data augmentation, but the most important aspect for the present thesis is not the classification objective itself. Rather, it is the paper's broader message: in medical imaging, model quality is not judged only by numerical performance, but also by whether the learned decisions are based on clinically meaningful structures.

It supports the wider argument that any image-enhancement or reconstruction method intended for medical use should be evaluated not only for visual improvement, but also for reliability, data realism, and trustworthiness.

3.5.3 Residual learning

A third supporting reference is [21], which introduces deep residual learning.It is one of the most influential architectural foundations behind later CNN-based SR models.

The central contribution of [21] is the residual learning framework in which stacked layers learn a residual mapping $F(x)$ and the network output is formed as $F(x) + x$ through shortcut connections. The paper shows that such identity-based skip connections make very deep networks easier to optimize and help address the

degradation problem that arises when depth is increased in plain networks [21]. This idea became fundamental far beyond image classification.

Its importance for the present thesis is indirect but substantial. Many SR architectures discussed earlier in this chapter—including VDSR, EDSR, RCAN, and later medical SR variants such as [6]—inherit the core logic of residual learning: instead of learning an entirely new mapping from scratch, the model focuses on the missing or transformed component relative to an existing input representation. In SISR, this is especially natural because the LR image already contains most low-frequency content, while the missing high-frequency detail can often be interpreted as a residual to be recovered.

For this reason, [21] is best treated here as a foundational architectural reference rather than as a standalone SR paper. Its strength is generality and optimization relevance; its limitation is that it does not address the image reconstruction objective, perceptual quality, or the one-to-many nature of SR directly. However, it remains important because much of the SR models would be difficult to explain cleanly without the residual-learning paradigm it established.

3.5.4 Radiology-domain transfer learning

The work in [39] is best understood as a supporting reference on radiology-domain deep learning resources. Because [39] presents RadImageNet as an open radiologic deep learning dataset for effective transfer learning, it is relevant as background for radiology-specific feature extraction and domain-adapted pretrained representations [39].

3.5.5 Overall synthesis of the literature

The literature reviewed in this chapter shows a clear and coherent progression in SISR.

The early literature establishes the inverse-problem nature of SR and the role of priors, first through classical and sparse-representation approaches, and then through shallow end-to-end CNNs. Regression-based deep models subsequently improve reconstruction quality through deeper residual learning, better feature routing, and attention mechanisms, but remain fundamentally tied to deterministic fidelity-oriented objectives. This makes them stable, controllable, and particularly attractive for applications where structural correctness matters, yet it also causes the well-known tendency toward over-smoothed textures.

The GAN-based literature then introduces a major conceptual shift by prioritizing perceptual realism through adversarial learning. This substantially improves visual sharpness and texture plausibility, but it also introduces instability, training difficulty, and a higher risk of hallucinated detail. These limitations become especially important in medical imaging, where visually plausible content is not necessarily diagnostically trustworthy.

Diffusion-based methods represent a newer generative alternative. They avoid adversarial competition and model the one-to-many LR-to-HR mapping more explicitly

through probabilistic denoising, often yielding highly realistic reconstructions with more stable training than GANs. However, they do so at the cost of much heavier iterative inference, greater computational burden, and continued concerns about controllability, explainability, and strict output consistency.

The more recent real-world and blind SR literature further expands the problem beyond idealized synthetic degradations. It makes clear that practical deployment requires robustness to unknown degradations, better degradation modeling, optimization strategies that can handle diverse restoration tasks, and increasingly stronger semantic or structural control over generative restoration.

Finally, in medical imaging, the goal of SR is not simply to create sharper-looking images, but to improve image utility while preserving structural reliability. Domain-specific SR models such as [6] show that medical reconstruction benefits from architectures tailored to frequency structure, while broader CT deep learning work such as [17] highlights the importance of data realism, interpretability, and clinically grounded validation. Supporting references such as [21] and [39] further clarify the architectural and representation-learning foundations that make such systems possible.

This overall literature trajectory directly motivates the position of the present thesis. On one hand, purely regression-based methods are stable and structurally safer, but they often fail to recover sufficiently rich perceptual detail. On the other hand, GAN-based and diffusion-based methods can substantially improve perceptual realism, but they introduce important trade-offs in stability, computational cost, and trustworthiness. Real-world and medical-imaging studies make these trade-offs even more critical, because practical degradations and clinical constraints leave less room for uncontrolled hallucination.

Therefore, the review of the literature identifies a clear methodological gap. Classical and sparse-representation methods provide interpretable priors but are limited in scalability and contextual modeling. CNN-based regression models are stable, deterministic, and computationally efficient, but their pixel-wise objectives often suppress perceptually important high-frequency information. GAN-based methods improve sharpness and visual realism, but introduce adversarial instability and the risk of hallucinated structures. Diffusion-based models provide powerful probabilistic reconstruction and strong perceptual quality, but their iterative sampling increases computational cost and can reduce deterministic consistency. These limitations are especially important in medical imaging, where SR must improve visual quality without compromising structural reliability. This thesis is positioned in this gap by developing GAN-free, regression-based SR methods that aim to improve perceptual and structural quality while preserving stability, efficiency, and fidelity for medical image reconstruction.

Chapter 4

Twinned Residual Auto-Encoder (TRAE)

4.1 Chapter overview

This chapter introduces the TRAE model, a regression-based SISR architecture designed to enhance perceptual fidelity while preserving the stability and determinism that are desirable in sensitive imaging domains.

In contrast to generative approaches that may introduce hallucinated structures, the proposed model targets a faithful reconstruction of the underlying HR image by combining (i) a learned multi-stage representation based on auto-encoding and (ii) a supervised SR backbone optimized with pixel-domain fidelity and perceptual-like constraints.

At a high level, the TRAE is composed of three coupled modules:

- Master AE: it processes HR images and learns an HR-oriented latent space.

  In the revised training procedure, the Master AE is first pretrained on a large portion of the available HR data and then frozen. Once frozen, it acts as a fixed reference encoder that produces stable latent representations of HR targets.

- Follower AE: it processes the corresponding LR images and learns to encode them into a latent representation that is consistent with the Master latent space, while also maintaining its own reconstruction capability.

- SR backbone: a CNN-based upsampling module maps the Follower pathway outputs into a final super-resolved estimate for a desired upscaling factor (i.e., $\times 2$, $\times 4$, $\times 8$).

The core mechanism enabling perceptual improvement in a regression setting is the *latent guidance* provided by the pretrained and frozen Master AE. Specifically, during training, the Follower latent is regularized to match (according to a suitable distance) the corresponding Master latent extracted from the paired HR image. This

resembles the role of feature-based perceptual losses (e.g., those based on fixed deep features [29]), but it is implemented directly within the learned latent space of the pretrained Master AE. The resulting objective jointly optimizes the Follower AE and the SR backbone so that the final super-resolved estimate is simultaneously accurate in the pixel domain and consistent with an HR-informed latent representation.

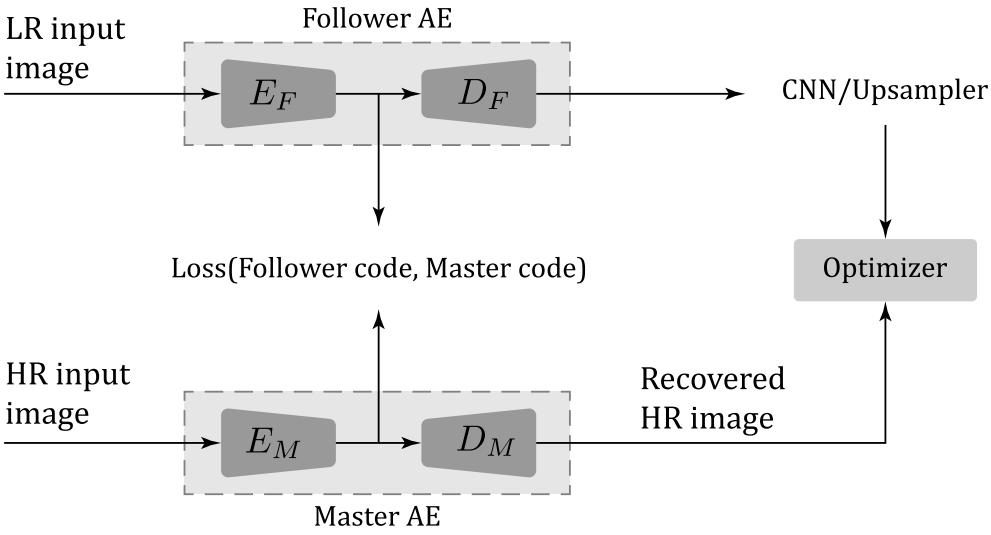

Figure 4.1. Schematic of the revised TRAE architecture. The Master AE is pretrained on HR data and then frozen. During SR training, the Follower AE and the SR backbone are optimized using a combination of SR fidelity losses and a latent guidance loss that aligns the Follower latent with the frozen Master latent.

The remainder of this chapter is organized as follows. Section 4.2 introduces the TRAE problem setting and notation. Section 4.3 details the architecture of the TRAE, including the Master/Follower AEs and the SR backbone for different upscaling factors. Section 4.4 provides a formal definition of the full training objective, including the latent guidance term and the SR fidelity losses. Section 4.5 describes the two-stage training strategy (Master pretraining followed by frozen-master joint training), while Section 4.6 reports the training algorithm and practical implementation details. Finally, Section 4.8 explains how the TRAE supports the overall thesis goals of stable regression-based SR with improved perceptual quality.

4.2 Problem setting and notation

This section specifies the notation and supervised learning setup that will be used throughout the chapter. The definitions are consistent with Chapters 1–2, and only the symbols required for describing the TRAE model are recalled here.

4.2.1 HR and LR image pairs

Let $\mathbf{X}_{\text{HR}} \in \mathbb{R}^{H \times W \times C}$ denote a ground-truth HR image, where H and W are the spatial dimensions and C is the number of channels (e.g., $C = 1$ for grayscale or $C = 3$ for RGB). For a given scale factor $s \in \mathcal{S}$, with

$$\mathcal{S} \triangleq \{2, 4, 8\}, \tag{4.1}$$

the corresponding LR image is denoted by $\mathbf{Y}_{\text{LR}}^{(s)} \in \mathbb{R}^{\frac{H}{s} \times \frac{W}{s} \times C}$.

Table 4.1. Key symbols used in Chapter 4.

Symbol	Meaning
$\mathbf{X}_{\text{HR}}$	High-resolution (HR) image
$\mathbf{Y}_{\text{LR}}^{(s)}$	Low-resolution (LR) image at scale s
s	Upscaling factor, $s \in \{2, 4, 8\}$
$\mathcal{T}_s(\cdot; \mathcal{D})$	Degradation operator for scale s
$f_\theta^{(s)}$	SR mapping for scale s
E_M, D_M	Master encoder/decoder
E_F, D_F	Follower encoder/decoder
$G^{(s)}$	SR backbone for scale s
$\mathbf{z}_M, \mathbf{z}_F$	Master and Follower latent representations

In this chapter, paired LR/HR samples are assumed to be available for supervised training. For each $\mathbf{X}_{\text{HR}}$, the associated $\mathbf{Y}_{\text{LR}}^{(s)}$ is obtained through a degradation operator that follows the same synthesis procedure adopted in the original TRAE experimental setting. Specifically, the LR image is generated by applying (i) a bicubic down-sampling with antialiasing at scale s, (ii) a Gaussian blur (sample-dependent), and (iii) additive mixed Gaussian–Laplacian noise (sample-dependent). Accordingly, we write:

$$\mathbf{Y}_{\text{LR}}^{(s)} = \mathcal{T}_s(\mathbf{X}_{\text{HR}}; \mathcal{D}), \tag{4.2}$$

where $\mathcal{D}$ collects the degradation parameters. In particular, $\mathcal{D}$ includes: (i) the scale factor s of the bicubic down-sampling with antialiasing; (ii) a spatially invariant 15×15 isotropic Gaussian blur kernel whose width is randomly drawn from the interval $[0.1, 1.5]$ (independently for each training sample); and (iii) additive i.i.d. zero-mean mixed Gaussian–Laplacian noise whose variance is randomly drawn from $[0, 0.01]$ (also independently for each sample). These degradation parameter ranges correspond to the synthesis protocol used to generate LR images from HR ones in the TRAE experimental setting.

In supervised training, the dataset is composed of paired samples:

$$\mathcal{P}^{(s)} \triangleq \left\{ \left(\mathbf{Y}_{\text{LR}}^{(s)}(i),\, \mathbf{X}_{\text{HR}}(i) \right) \right\}_{i=1}^{N}, \tag{4.3}$$

where N denotes the number of available LR/HR pairs for training at scale factor s. The same dataset cardinalities are used across the considered scaling factors (i.e., N does not depend on s), and the data are split into training, validation, and test partitions with fixed sizes.

4.2.2 Supervised SR mapping goal

For a given scale factor $s \in \mathcal{S}$, the goal of SISR is to learn a parametric mapping $f_\theta^{(s)}(\cdot)$ that predicts an HR estimate from an LR input:

$$\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)} = f_\theta^{(s)}\left(\mathbf{Y}_{\mathrm{LR}}^{(s)}\right), \tag{4.4}$$

where θ denotes the set of trainable parameters (in TRAE, θ includes the parameters of the Follower AE and the SR backbone for scale s). The supervised learning objective minimizes a task-dependent loss between $\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}$ and $\mathbf{X}_{\mathrm{HR}}$ over the paired dataset $\mathcal{P}^{(s)}$:

$$\theta^* = \arg\min_\theta \; \mathbb{E}_{(\mathbf{Y}_{\mathrm{LR}}^{(s)}, \mathbf{X}_{\mathrm{HR}}) \sim \mathcal{P}^{(s)}} \left[\mathcal{L}\left(\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}, \mathbf{X}_{\mathrm{HR}}\right)\right], \tag{4.5}$$

where $\mathcal{L}(\cdot,\cdot)$ could include multiple components (pixel-domain fidelity, reconstruction terms, and latent guidance terms). The explicit definition of $\mathcal{L}$ for TRAE is provided in Section 4.4.

4.2.3 Module Notation Used in TRAE

The TRAE model employs two auto-encoders and an SR backbone. For clarity, we denote:

- Master encoder and decoder by $E_M(\cdot)$ and $D_M(\cdot)$, respectively.
- Follower encoder and decoder by $E_F(\cdot)$ and $D_F(\cdot)$, respectively.
- The SR backbone for scale factor s by $G^{(s)}(\cdot)$.

Given a paired sample $(\mathbf{Y}_{\mathrm{LR}}^{(s)}, \mathbf{X}_{\mathrm{HR}})$, the corresponding latent representations are

$$\mathbf{z}_M = E_M(\mathbf{X}_{\mathrm{HR}}), \qquad \mathbf{z}_F = E_F(\mathbf{Y}_{\mathrm{LR}}^{(s)}), \tag{4.6}$$

and these latents will be used to define the latent guidance constraint in the TRAE loss (Section 4.4). The Master AE is pretrained before the end-to-end SR training and is kept fixed thereafter; thus, during SR training, $E_M(\cdot)$ provides a stable HR-informed latent reference.

4.3 TRAE architecture

4.3.1 High-level components

The TRAE model is composed of three coupled modules: a Master AE operating on HR images, a Follower AE operating on the corresponding LR images, and a scale-specific SR backbone that produces the final HR estimate. The role of each module is summarized below.

AE: basic formulation. An AE is an architecture that learns to commonly represent an input image $\mathbf{x}$ through a lower-dimensional latent representation, and to reconstruct the input from that latent code. Formally, an AE is the composition of an *encoder* $E(\cdot)$ and a *decoder* $D(\cdot)$:

$$\mathbf{z} = E(\mathbf{x}), \qquad \widehat{\mathbf{x}} = D(\mathbf{z}) = D\big(E(\mathbf{x})\big), \tag{4.7}$$

where $\mathbf{z}$ denotes the latent (feature) representation and $\widehat{\mathbf{x}}$ denotes the reconstructed output. The encoder and decoder are trained so that the reconstruction $\widehat{\mathbf{x}}$ is close to the input $\mathbf{x}$ under a chosen reconstruction loss (e.g., ℓ_1 or ℓ_2), while $\mathbf{z}$ captures informative content about $\mathbf{x}$. In practice, for images, both $E(\cdot)$ and $D(\cdot)$ are implemented with convolutional and residual blocks to exploit spatial locality and to stabilize optimization.

Master AE_M (HR latent space). The Master AE, denoted by $\mathrm{AE}_M = (E_M, D_M)$, processes HR images and defines an HR-oriented latent space. Given an HR image $\mathbf{X}_{\mathrm{HR}}$, the Master encoder produces the latent code

$$\mathbf{z}_M = E_M(\mathbf{X}_{\mathrm{HR}}), \tag{4.8}$$

and the Master decoder reconstructs an HR estimate

$$\widehat{\mathbf{X}_{\mathrm{HR}\,M}} = D_M(\mathbf{z}_M) = D_M(E_M(\mathbf{X}_{\mathrm{HR}})). \tag{4.9}$$

In the TRAE training procedure, the Master AE is pretrained to learn a stable representation of HR content. After pretraining, the Master AE is kept fixed and is used to produce a reference HR latent code $\mathbf{z}_M$ for each training pair. This reference latent representation provides a perceptual-like guidance signal for learning the LR-to-HR mapping without relying on adversarial training.

Follower AE (LR latent space). The Follower AE, denoted by $\mathrm{AE}_F = (E_F, D_F)$, operates on LR images. For a given LR input $\mathbf{Y}_{\mathrm{LR}}^{(s)}$, the Follower encoder produces the latent code

$$\mathbf{z}_F = E_F\Big(\mathbf{Y}_{\mathrm{LR}}^{(s)}\Big), \tag{4.10}$$

and the Follower decoder yields a reconstructed LR-domain output

$$\widehat{\mathbf{Y}_{\mathrm{LR}\,F}^{(s)}} = D_F(\mathbf{z}_F) = D_F\Big(E_F\Big(\mathbf{Y}_{\mathrm{LR}}^{(s)}\Big)\Big). \tag{4.11}$$

The Follower AE serves two purposes. First, it learns an LR representation that retains content necessary for accurate reconstruction. Second, its latent code $\mathbf{z}_F$ is encouraged to align with the HR reference latent code $\mathbf{z}_M$ (computed from the paired HR image) through a latent guidance loss.

This alignment constraint promotes HR-consistent internal representations while keeping the overall model within a regression-based framework.

SR backbone $G^{(s)}$ (upsampling CNN). The SR backbone is a convolutional network denoted by $G^{(s)}(\cdot)$, specialized for each scale factor $s \in \{2, 4, 8\}$. It maps the Follower pathway output into the final super-resolved estimate:

$$\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)} = G^{(s)}\left(\widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)}\right). \tag{4.12}$$

Intuitively, $\widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)}$ provides a content-preserving reconstruction in the LR domain, and $G^{(s)}(\cdot)$ performs the resolution enhancement by learning the upsampling and refinement required to match the HR target $\mathbf{X}_{\mathrm{HR}}$. The SR backbone is trained jointly with the Follower AE using a pixel-domain SR fidelity loss at the output, while the latent guidance term regularizes the internal representation through $\mathbf{z}_F \leftrightarrow \mathbf{z}_M$ alignment.

The subsequent subsections provide detailed architectural descriptions of AE_M, AE_F, and $G^{(s)}(\cdot)$, including their building blocks, dimensionality changes, and design choices.

4.3.2 Master AE

This subsection describes the Master auto-encoder $\mathrm{AE}_M = (E_M, D_M)$ that operates on HR images and provides a fixed latent-space reference for the TRAE framework. The Master AE is first pretrained using HR images and is then kept frozen during the training of the remaining components. Consequently, the Master encoder $E_M(\cdot)$ defines a stable HR-oriented latent space, and the latent code $\mathbf{z}_M$ serves as a consistent target for the latent guidance constraint.

Encoder–decoder factorization and latent representation

Given an HR image $\mathbf{X}_{\mathrm{HR}} \in \mathbb{R}^{H \times W \times C}$, the Master encoder produces a latent tensor:

$$\mathbf{z}_M = E_M(\mathbf{X}_{\mathrm{HR}}), \tag{4.13}$$

and the Master decoder reconstructs an HR estimate:

$$\widehat{\mathbf{X}_{\mathrm{HR}M}} = D_M(\mathbf{z}_M) = D_M(E_M(\mathbf{X}_{\mathrm{HR}})). \tag{4.14}$$

In practical convolutional AEs, the latent representation is typically a *tensor* rather than a single vector, i.e.,

$$\mathbf{z}_M \in \mathbb{R}^{H_z \times W_z \times C_z}, \tag{4.15}$$

where (H_z, W_z) are spatial dimensions after downsampling and C_z is the number of channels in the latent space. The exact values of (H_z, W_z, C_z) depend on the number of downsampling stages and the channel configuration of the encoder. We describe the architectural blocks that determine these dimensions.

Residual building block

Both the encoder and decoder are constructed using residual blocks to facilitate training stability and to enable deeper representations. A generic residual block maps an input feature tensor $\mathbf{h}$ to an output tensor of the same shape:

$$\mathrm{ResBlock}(\mathbf{h}) = \mathbf{h} + \Phi(\mathbf{h}), \tag{4.16}$$

where $\Phi(\cdot)$ denotes a sequence of learnable convolutional transformations and nonlinearities. In typical implementations, $\Phi(\cdot)$ consists of two 3×3 convolutional layers with intermediate activation (e.g., ReLU or LeakyReLU) and, if used, normalization layers. The skip connection in (4.16) improves gradient flow and mitigates degradation in deep networks.

Master encoder $E_M(\cdot)$: downsampling path

The Master encoder extracts hierarchical HR features through a sequence of convolutional layers and downsampling stages. Conceptually, $E_M(\cdot)$ is composed of:

- Input projection: a convolutional layer maps the input image $\mathbf{X}_{\mathrm{HR}}$ into an initial feature space with C_0 channels.
- Residual feature extraction: a stack of residual blocks refines features at the current resolution.
- Downsampling stages: spatial resolution is reduced through a set of strided convolutions (or equivalent downsampling operators), typically halving (H, W) at each stage, while increasing the number of channels to preserve information capacity.

Let L denote the number of downsampling stages in the encoder. After L stages, the spatial size is reduced by a factor of 2^L, yielding:

$$H_z = \frac{H}{2^L}, \qquad W_z = \frac{W}{2^L}. \tag{4.17}$$

The number of latent channels is denoted by C_z, typically obtained by increasing channel width progressively across stages (e.g., $C_0 \to 2C_0 \to 4C_0 \to \cdots$). The final encoder layers output the latent tensor $\mathbf{z}_M \in \mathbb{R}^{H_z \times W_z \times C_z}$.

Master decoder $D_M(\cdot)$: upsampling path

The Master decoder mirrors the encoder structure to reconstruct an HR image from the latent tensor. It is composed of:

- Latent feature refinement: residual blocks operating at the latent resolution (H_z, W_z) refine $\mathbf{z}_M$ before upsampling.
- Upsampling stages: a sequence of learnable upsampling blocks progressively restores spatial resolution, typically doubling size at each stage until reaching (H, W). Upsampling can be implemented using transposed convolutions, subpixel convolution (pixel-shuffle), or interpolation followed by convolution; the

specific choice is fixed for the TRAE implementation and is described in Section 4.7.

- Output projection: a final convolution maps the last feature tensor back to C channels to produce the reconstruction $\widehat{\mathbf{X}_{\mathrm{HR}\,M}}$.

Pretraining and freezing of the Master AE

The Master AE is pretrained on HR images to learn an HR-consistent latent space. Denote by $\mathcal{X}_{\mathrm{train}}$ the set of HR training images used for Master pretraining. The Master AE parameters θ_M (i.e., the parameters of E_M and D_M) are obtained by minimizing a reconstruction objective of the form:

$$\begin{aligned} \theta_M^* &= \arg\min_{\theta_M} \mathbb{E}_{\mathbf{X}_{\mathrm{HR}}\sim\mathcal{X}_{\mathrm{train}}} \left[\mathcal{L}_M \left(\widehat{\mathbf{X}_{\mathrm{HR}\,M}}, \mathbf{X}_{\mathrm{HR}} \right) \right], \\ \widehat{\mathbf{X}_{\mathrm{HR}\,M}} &= D_M(E_M(\mathbf{X}_{\mathrm{HR}})), \end{aligned} \tag{4.18}$$

where $\mathcal{L}_M(\cdot,\cdot)$ is a reconstruction loss (the explicit form is provided in Section 4.4).

After pretraining, the Master AE is *frozen*: its parameters θ_M^* are kept fixed and are not updated during the training of the Follower AE and the SR backbone. In other words, during SR training, the Master encoder produces a deterministic latent reference

$$\mathbf{z}_M = E_M(\mathbf{X}_{\mathrm{HR}}; \theta_M^*), \tag{4.19}$$

which is used to guide the Follower latent representation through a latent alignment term. This fixed-latent reference prevents the latent target from drifting during optimization and provides a stable, HR-informed constraint that complements the pixel-domain SR fidelity loss.

4.3.3 Follower AE

The Follower AE, $\mathrm{AE}_F = (E_F, D_F)$, is a trainable module that processes the LR input images and produces (i) an LR-oriented reconstruction and (ii) a latent representation that is constrained to be consistent with the HR latent space induced by the frozen Master encoder.

In the TRAE pipeline, the Follower AE is therefore responsible for extracting from the LR input a compact but informative representation that will be subsequently exploited by the SR backbone.

Encoder–decoder factorization and latent representation

Given an LR input $\mathbf{Y}_{\mathrm{LR}}^{(s)} \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times C}$, the Follower encoder produces the latent representation:

$$\mathbf{z}_F = E_F\left(\mathbf{Y}_{\mathrm{LR}}^{(s)}\right), \tag{4.20}$$

and the Follower decoder reconstructs the LR image:

$$\widehat{\mathbf{Y}_{\mathrm{LR}\,F}^{(s)}} = D_F(\mathbf{z}_F) = D_F\Big(E_F\left(\mathbf{Y}_{\mathrm{LR}}^{(s)}\right)\Big). \tag{4.21}$$

As for the Master AE, the latent representation produced by a convolutional encoder is generally a tensor. In TRAE, the latent representation is handled in a way that enables a well-defined latent guidance loss between the Master and Follower encodings. Specifically, we consider the latent codes $\mathbf{z}_M$ and $\mathbf{z}_F$ as elements of a common feature space:

$$\mathbf{z}_M, \mathbf{z}_F \in \mathbb{R}^{d_z}, \tag{4.22}$$

where d_z is a fixed latent dimensionality. This can be achieved, for example, by applying a final feature aggregation step at the output of the encoder (e.g., global average pooling followed by a 1×1 convolution and flattening), so that the produced latent vector has the same dimension d_z independently of the spatial size of the input image. In practice, any deterministic projection that maps the last encoder feature tensor into a fixed-size vector may be adopted, provided that it is applied consistently in both $E_M(\cdot)$ and $E_F(\cdot)$.

Architecture of the follower encoder $E_F(\cdot)$

The Follower encoder follows the same design principles as the Master encoder, relying on convolutional layers, residual blocks, and downsampling stages to progressively extract hierarchical representations from the LR input. $E_F(\cdot)$ consists of:

- Input projection: an initial convolution that maps $\mathbf{Y}_{\text{LR}}^{(s)}$ into a feature tensor with C_0 channels;
- Residual feature extraction: stacks of residual blocks that refine features at each resolution level;
- Downsampling stages: strided convolutions that reduce the spatial resolution while increasing the number of feature channels;
- Latent aggregation: a deterministic mapping from the final feature tensor to a fixed-dimensional latent vector $\mathbf{z}_F \in \mathbb{R}^{d_z}$, as in (4.22).

The residual blocks within the Follower encoder follow the generic form:

$$\text{ResBlock}(\mathbf{h}) = \mathbf{h} + \Phi(\mathbf{h}), \tag{4.23}$$

where $\Phi(\cdot)$ denotes the composition of convolutional transformations and nonlinearities. This residual formulation improves the conditioning of the optimization problem and supports deeper feature extractors.

Architecture of the Follower decoder $D_F(\cdot)$

The Follower decoder maps $\mathbf{z}_F$ back to the LR image space. It mirrors the encoder by progressively restoring the LR spatial resolution through a sequence of learnable upsampling stages. Overall, $D_F(\cdot)$ consists of:

- Latent expansion: a mapping from the latent vector $\mathbf{z}_F$ to an initial low-resolution feature tensor;
- Upsampling stages: learnable upsampling blocks that increase the spatial resolution until reaching $\frac{H}{s} \times \frac{W}{s}$;

- Residual refinement: residual blocks that refine intermediate feature maps;
- Output projection: a final convolution producing the reconstructed LR output $\widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)} \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times C}$.

The output $\widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)}$ plays a dual role in TRAE: it is (i) directly supervised through an LR reconstruction term (Section 4.4) and (ii) used as the input of the SR backbone $G^{(s)}(\cdot)$ to generate the final super-resolved estimate $\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}$ (see Section 4.3.4). The architectures of the Master and Follower AEs are presented in Table 4.2. Three separate Follower AE are considered and trained for each scale factor $s \in \mathcal{S}$.

Table 4.2. Layer-wise specifications of Master AE, and Follower AEs for $s \in \mathcal{S}$. The kernel size for all layers is 3×3. $\mathrm{Conv}(F, K, S)$ $\mathrm{TrC}(F, K, S)$ denote Conv2D layer and Transpose Conv2D layers with F filters of the size $K \times K$ and S stride. The input images are grayscale. TP denotes the number of trainable parameters.

Layer	AE_M	$\mathrm{AE}_F, s = 2$	$\mathrm{AE}_F, s = 4$	$\mathrm{AE}_F, s = 8$
Input size	512×512	256×256	128×128	64×64
Layer #1	$\mathrm{Conv}(32, 3, 2)$	$\mathrm{Conv}(32, 3, 2)$	$\mathrm{Conv}(64, 3, 2)$	$\mathrm{Conv}(64, 3, 2)$
Layer #2	$\mathrm{Conv}(64, 3, 2)$	$\mathrm{Conv}(64, 3, 2)$	$\mathrm{Conv}(128, 3, 2)$	$\mathrm{Conv}(128, 3, 2)$
Layer #3	$\mathrm{Conv}(128, 3, 2)$	$\mathrm{Conv}(128, 3, 2)$	$\mathrm{Conv}(256, 3, 2)$	$\mathrm{Conv}(256, 3, 2)$
Layer #4	$\mathrm{Conv}(256, 3, 2)$	$\mathrm{Conv}(256, 3, 2)$	$\mathrm{Conv}(512, 3, 2)$	$\mathrm{Conv}(512, 3, 1)$
Layer #5	$\mathrm{Conv}(512, 3, 2)$	$\mathrm{Conv}(512, 3, 2)$	$\mathrm{Conv}(512, 3, 1)$	$\mathrm{Conv}(512, 3, 1)$
Layer #6	$\mathrm{Conv}(512, 3, 2)$	$\mathrm{Conv}(512, 3, 1)$	$\mathrm{Conv}(512, 3, 1)$	$\mathrm{Conv}(512, 3, 1)$
Layer #7	$\mathrm{Conv}(512, 3, 1)$	$\mathrm{Conv}(512, 3, 1)$	$\mathrm{TrC}(256, 3, 2)$	$\mathrm{TrC}(256, 3, 1)$
Layer #8	$\mathrm{Conv}(512, 3, 1)$	$\mathrm{TrC}(256, 3, 2)$	$\mathrm{TrC}(128, 3, 2)$	$\mathrm{TrC}(128, 3, 2)$
Layer #9	$\mathrm{TrC}(512, 3, 2)$	$\mathrm{TrC}(128, 3, 2)$	$\mathrm{TrC}(64, 3, 2)$	$\mathrm{TrC}(64, 3, 2)$
Layer #10	$\mathrm{TrC}(256, 3, 2)$	$\mathrm{TrC}(64, 3, 2)$	$\mathrm{TrC}(1, 3, 2)$	$\mathrm{TrC}(1, 3, 2)$
Layer #11	$\mathrm{TrC}(128, 3, 2)$	$\mathrm{TrC}(32, 3, 2)$	–	–
Layer #12	$\mathrm{TrC}(64, 3, 2)$	$\mathrm{TrC}(1, 3, 2)$	–	–
Layer #13	$\mathrm{TrC}(32, 3, 2)$	–	–	–
Layer #14	$\mathrm{TrC}(1, 3, 2)$	–	–	–
Output size	512×512	256×256	128×128	64×64
TP	$20,733,953$	$9,853,954$	$8,670,722$	$8,031,746$

Role of the Follower latent code in latent guidance

During training, $\mathbf{z}_F$ is encouraged to be close to the reference latent code $\mathbf{z}_M = E_M(\mathbf{X}_{\mathrm{HR}})$ computed from the paired HR image. This is enforced by a latent guidance loss of the form:

$$\mathcal{L}_{\mathrm{lat}} = d(\mathbf{z}_F, \mathbf{z}_M)\,, \tag{4.24}$$

where $d(\cdot, \cdot)$ is a suitable distance (e.g., ℓ_1 or ℓ_2 norm). Since the Master encoder is frozen, the latent target $\mathbf{z}_M$ provides a stable HR-informed reference, while the Follower encoder adapts its representation so that LR inputs are mapped into a latent space consistent with HR structure and content.

4.3.4 SR Backbone: upsampling CNN (×2, ×4, ×8)

In TRAE, the SR backbone is an *upsampling CNN* that transforms the output of the Follower AE into the final HR estimate. For each scale factor $s \in \{2, 4, 8\}$, a separate TRAE model is trained; therefore, the upsampling module $G^{(s)}(\cdot)$ is instantiated with a scale-specific architecture and a scale-specific parameter set.

Let $\widehat{\mathbf{Y}_{\mathrm{LR}}}_F^{(s)}$ denote the output of the Follower AE (Section 4.3.3). The SR backbone produces the super-resolved output as:

$$\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)} = G^{(s)}\left(\widehat{\mathbf{Y}_{\mathrm{LR}}}_F^{(s)}\right), \tag{4.25}$$

where $\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}$ has the same spatial size and number of channels as the HR target $\mathbf{X}_{\mathrm{HR}}$.

Importantly, the upsampling CNN in TRAE is a feed-forward stack of convolutional layers and Depth-to-Space operators.

General input–output dimensions and scale-specific training

Let the HR image be $\mathbf{X}_{\mathrm{HR}} \in \mathbb{R}^{H\times W\times C}$ and the corresponding LR image at scale s be $\mathbf{Y}_{\mathrm{LR}}^{(s)} \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times C}$. In the patch-based training setting adopted for TRAE, HR patches of size $H \times W = 512 \times 512$ are used, yielding:

$$\mathbf{Y}_{\mathrm{LR}}^{(2)} \in \mathbb{R}^{256\times256\times C}, \qquad \mathbf{Y}_{\mathrm{LR}}^{(4)} \in \mathbb{R}^{128\times128\times C}, \qquad \mathbf{Y}_{\mathrm{LR}}^{(8)} \in \mathbb{R}^{64\times64\times C}, \tag{4.26}$$

and the upsampler output for all scales satisfies:

$$\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)} \in \mathbb{R}^{512\times512\times C}. \tag{4.27}$$

This formulation is valid for arbitrary channel count C (e.g., $C = 1$ for grayscale images and $C = 3$ for RGB images). When reporting experiments on a grayscale dataset, we explicitly set $C = 1$ in the corresponding implementation details and tables.

Layer notation

We describe the scale-specific upsamplers using the following layer notation.

- $\mathrm{Conv}(F, K, S)$ denotes a 2D convolutional layer with F output filters (feature maps), kernel size $K \times K$, and stride S.
- $\mathrm{D2S}(B)$ denotes a Depth-to-Space operator (pixel shuffle) with block size B.

The Depth-to-Space operator rearranges data from depth into spatial blocks. Formally:

$$\mathrm{D2S}(B):\ \mathbb{R}^{h\times w\times (c\,B^2)} \ \to\ \mathbb{R}^{(Bh)\times(Bw)\times c}. \tag{4.28}$$

Thus, D2S(2) increases each spatial dimension by a factor of 2 while reducing the channel dimension by a factor of 4. In the TRAE upsamplers, D2S(2) is used one, two, or three times to realize ×2, ×4, or ×8 upsampling, respectively.

Scale-specific architectures of $G^{(2)}$, $G^{(4)}$, and $G^{(8)}$

The architecture of $G^{(s)}(\cdot)$ is scale-specific and follows the configuration summarized in Table 4.3. The input to the upsampler is $\widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)} \in \mathbb{R}^{\frac{512}{s}\times\frac{512}{s}\times C}$. Since each D2S(2) operation requires its input to have a channel dimension divisible by 4, the preceding convolutional layers are designed to provide a suitable number of feature maps. The final convolution layer projects the feature tensor back to C channels to produce $\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)} \in \mathbb{R}^{512\times512\times C}$.

Table 4.3. Scale-specific TRAE upsampler architectures. Here, $\mathrm{Conv}(F, K, S)$ denotes a convolutional layer with F filters of size $K \times K$ and stride S, and D2S(2) denotes Depth-to-Space (pixel shuffle) with block size 2. For the grayscale datasets used in the TRAE experiments, $C = 1$.

	Upsampler (×2)	Upsampler (×4)	Upsampler (×8)
Input size	$256 \times 256 \times C$	$128 \times 128 \times C$	$64 \times 64 \times C$
Layer #1	$\mathrm{Conv}(64, 5, 1)$	$\mathrm{Conv}(64, 3, 1)$	$\mathrm{Conv}(64, 3, 1)$
Layer #2	D2S(2)	D2S(2)	D2S(2)
Layer #3	$\mathrm{Conv}(C, 5, 1)$	$\mathrm{Conv}(64, 5, 1)$	$\mathrm{Conv}(64, 5, 1)$
Layer #4	–	D2S(2)	D2S(2)
Layer #5	–	$\mathrm{Conv}(C, 5, 1)$	$\mathrm{Conv}(64, 5, 1)$
Layer #6	–	–	D2S(2)
Layer #7	–	–	$\mathrm{Conv}(C, 5, 1)$
Output size	$512 \times 512 \times C$	$512 \times 512 \times C$	$512 \times 512 \times C$

Remarks on parameterization and the grayscale experimental setting

The table above expresses the architectures in a channel-general form by using $\mathrm{Conv}(C, 5, 1)$ as the final projection layer. When the training data are grayscale, $C = 1$ and the upsampler trainable parameters count 2,065, 26,705, and 52,369 for for $s = 2$, $s = 4$, and $s = 8$ respectively.

For multi-channel images (RGB with $C = 3$), the only modification required is to set the final output-channel dimension to C (and, correspondingly, to interpret the input and output tensors as having C channels). The overall computational pattern of the upsampler remains unchanged: feature extraction at LR resolution, followed by one or more D2S(2) operations and a final channel projection to obtain the HR estimate.

4.3.5 Data flow through the network

This subsection describes the end-to-end data flow in TRAE for a single paired training sample and a fixed scale factor $s \in \{2, 4, 8\}$. The goal is to explicitly identify all intermediate variables that participate in the forward computation and, consequently, in the loss functions used during training.

The description is kept general with respect to the number of channels C; for

grayscale datasets we set $C = 1$.

Paired inputs at scale s

For a given scale factor s, each supervised training sample consists of a paired LR and HR image:

$$(\mathbf{Y}_{\mathrm{LR}}^{(s)}, \mathbf{X}_{\mathrm{HR}}), \tag{4.29}$$

where

$$\begin{aligned} \mathbf{X}_{\mathrm{HR}} &\in \mathbb{R}^{H\times W\times C}, \\ \mathbf{Y}_{\mathrm{LR}}^{(s)} &\in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times C}. \end{aligned} \tag{4.30}$$

Master pathway: HR encoding and reconstruction

The Master pathway operates on the HR image and provides two quantities: an HR-informed latent code and an HR reconstruction. Given $\mathbf{X}_{\mathrm{HR}}$, the Master encoder computes the latent representation

$$\mathbf{z}_M = E_M(\mathbf{X}_{\mathrm{HR}}), \tag{4.31}$$

and the Master decoder produces the reconstructed HR image

$$\widehat{\mathbf{X}_{\mathrm{HR}\,M}} = D_M(\mathbf{z}_M) = D_M(E_M(\mathbf{X}_{\mathrm{HR}})). \tag{4.32}$$

The Master AE is pretrained and then kept frozen. Therefore, during the training of the Follower AE pathway and the SR backbone, both $\mathbf{z}_M$ and $\widehat{\mathbf{X}_{\mathrm{HR}\,M}}$ are deterministic outputs that serve as fixed references.

Follower pathway: LR encoding and LR reconstruction

The Follower pathway processes the LR input and produces a latent representation as well as an LR reconstruction. Given $\mathbf{Y}_{\mathrm{LR}}^{(s)}$, the Follower encoder computes

$$\mathbf{z}_F = E_F\left(\mathbf{Y}_{\mathrm{LR}}^{(s)}\right), \tag{4.33}$$

and the Follower decoder reconstructs the LR image

$$\widehat{\mathbf{Y}_{\mathrm{LR}\,F}}^{(s)} = D_F(\mathbf{z}_F) = D_F\Big(E_F\left(\mathbf{Y}_{\mathrm{LR}}^{(s)}\right)\Big). \tag{4.34}$$

The reconstruction $\widehat{\mathbf{Y}_{\mathrm{LR}\,F}}^{(s)}$ is explicitly supervised during training because it is the direct input to the SR backbone.

Latent guidance connection: aligning LR and HR representations

For each paired sample, TRAE computes a latent code from the HR target (Master AE) and a latent code from the LR input (Follower AE):

$$\mathbf{z}_M = E_M(\mathbf{X}_{\mathrm{HR}}), \qquad \mathbf{z}_F = E_F\left(\mathbf{Y}_{\mathrm{LR}}^{(s)}\right). \tag{4.35}$$

A latent alignment distance $d(\cdot,\cdot)$ is applied to enforce consistency between these representations:

$$\mathcal{L}_{\text{lat}} = d(\mathbf{z}_F, \mathbf{z}_M)\,. \tag{4.36}$$

Since the Master encoder is frozen, $\mathbf{z}_M$ provides a stable HR-informed reference, while the Follower encoder adapts such that LR inputs are mapped into a representation space consistent with HR content and structure.

SR backbone: LR reconstruction to HR estimate

The SR backbone takes the Follower reconstruction $\widehat{\mathbf{Y}_{\text{LR}F}}^{(s)}$ and produces the final SR estimate:

$$\widehat{\mathbf{X}_{\text{HR}}}^{(s)} = G^{(s)}\left(\widehat{\mathbf{Y}_{\text{LR}F}}^{(s)}\right), \tag{4.37}$$

where $G^{(s)}(\cdot)$ is the scale-specific upsampling CNN defined in Section 4.3.4.

Dual-target SR supervision at the network output

TRAE training objective supervises the SR $\widehat{\mathbf{X}_{\text{HR}}}^{(s)}$ using two targets: the ground-truth HR image $\mathbf{X}_{\text{HR}}$ and the frozen Master reconstruction $\widehat{\mathbf{X}_{\text{HR}M}}$. This dual supervision achieves two complementary effects:

- the loss against $\mathbf{X}_{\text{HR}}$ anchors the SR output to the true HR target;
- the loss against $\widehat{\mathbf{X}_{\text{HR}M}}$ encourages the SR output to remain consistent with the HR manifold captured by the pretrained Master AE.
- stability against artifacts: incorporating the fidelity term $\ell\left(\widehat{\mathbf{X}_{\text{HR}}}^{(s)}, \widehat{\mathbf{X}_{\text{HR}M}}\right)$ regularizes the SR output toward a structurally coherent HR reference produced by the pretrained Master AE. This additional constraint can reduce the tendency of the upsampling CNN to introduce unstable high-frequency artifacts (e.g., ringing-like patterns) that may arise when optimizing only a pixel-domain loss against $\mathbf{X}_{\text{HR}}$.
- noise and irrelevant-detail suppression: if the HR targets contain acquisition noise or sample-specific irrelevant fluctuations, the Master AE reconstruction $\widehat{\mathbf{X}_{\text{HR}M}}$ often preserves the dominant anatomical/structural content while attenuating such nuisance components. Encouraging $\widehat{\mathbf{X}_{\text{HR}}}^{(s)}$ to match $\widehat{\mathbf{X}_{\text{HR}M}}$ therefore promotes SR outputs that emphasize stable content and are less sensitive to noise-like variations.
- consistent appearance across outputs: since the Master AE is fixed after pre-training, its reconstruction $\widehat{\mathbf{X}_{\text{HR}M}}$ provides a deterministic and consistent HR reference across all training samples. The corresponding loss term encourages the SR outputs to follow a coherent appearance profile (e.g., in terms of texture/edge behavior) across different inputs and scale factors, improving the uniformity of the reconstructed images.

Accordingly, the SR output is optimized with a composite fidelity term of the form:

$$\mathcal{L}_{\mathrm{sr}} = \lambda_{\mathrm{gt}}\, \ell\Big(\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}, \mathbf{X}_{\mathrm{HR}}\Big) + \lambda_{\mathrm{m}}\, \ell\Big(\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}, \widehat{\mathbf{X}_{\mathrm{HR}M}}\Big), \tag{4.38}$$

where $\ell(\cdot,\cdot)$ denotes a pixel-domain fidelity loss (e.g., ℓ_1 or Charbonnier), and $\lambda_{\mathrm{gt}}, \lambda_{\mathrm{m}} \geq 0$ are scalar weights controlling the relative influence of the two targets. The explicit choice of $\ell(\cdot,\cdot)$ and the complete TRAE objective (including LR reconstruction and latent guidance terms) are provided in Section 4.4.

End-to-end forward summary for one scale

The complete forward pass of TRAE for a given scale factor s can be summarized as:

$$\mathbf{z}_M = E_M(\mathbf{X}_{\mathrm{HR}}), \tag{4.39}$$

$$\widehat{\mathbf{X}_{\mathrm{HR}M}} = D_M(\mathbf{z}_M), \tag{4.40}$$

$$\mathbf{z}_F = E_F\Big(\mathbf{Y}_{\mathrm{LR}}^{(s)}\Big), \tag{4.41}$$

$$\widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)} = D_F(\mathbf{z}_F), \tag{4.42}$$

$$\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)} = G^{(s)}\Big(\widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)}\Big). \tag{4.43}$$

During SR training, the Master parameters are fixed, and only the parameters of the Follower AE and the SR backbone are updated. The learning objective combines (i) an LR reconstruction loss on $\widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)}$, (ii) the latent guidance loss between $\mathbf{z}_F$ and $\mathbf{z}_M$, and (iii) the dual-target SR fidelity loss in (4.38). This coupling between LR supervision, latent guidance, and dual-target SR supervision constitutes the core training mechanism of TRAE.

4.4 Objective functions and loss design

4.4.1 Reconstruction losses

This subsection defines the reconstruction loss terms used in TRAE. These terms supervise (i) the pretraining of the Master auto-encoder and (ii) the LR reconstruction of the Follower AE during the training of the overall SR model. Throughout, we consider a generic pixel-domain fidelity function $\ell(\cdot,\cdot)$, and then instantiate it with a specific choice (e.g., ℓ_1 or Charbonnier).

Pixel-domain fidelity functions

Two common and robust choices for pixel-domain fidelity are the ℓ_1 loss and the Charbonnier loss. For tensors $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{H\times W\times C}$, the ℓ_1 loss is

$$\ell_1(\mathbf{A},\mathbf{B}) \triangleq \frac{1}{HWC} \sum_{h=1}^{H}\sum_{w=1}^{W}\sum_{c=1}^{C} |A_{h,w,c} - B_{h,w,c}|\,. \tag{4.44}$$

The Charbonnier loss is a smooth approximation of ℓ_1 defined as

$$\ell_{\text{char}}(\mathbf{A}, \mathbf{B}) \triangleq \frac{1}{HWC} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{c=1}^{C} \sqrt{(A_{h,w,c} - B_{h,w,c})^2 + \epsilon^2}, \tag{4.45}$$

where $\epsilon > 0$ is a small constant (e.g., $\epsilon = 10^{-3}$) that prevents numerical instability near zero. We write $\ell(\cdot, \cdot)$ to denote the chosen pixel-domain fidelity function (either ℓ_1 or ℓ_{char}), and specify the selected option in the implementation details section.

Master AE reconstruction loss (pretraining stage)

The Master auto-encoder $\text{AE}_M = (E_M, D_M)$ is pretrained using HR images so that it learns an HR-consistent latent representation space. For an HR input $\mathbf{X}_{\text{HR}}$, the Master produces:

$$\mathbf{z}_M = E_M(\mathbf{X}_{\text{HR}}), \qquad \widehat{\mathbf{X}_{\text{HR}M}} = D_M(\mathbf{z}_M) = D_M(E_M(\mathbf{X}_{\text{HR}})). \tag{4.46}$$

The Master pretraining objective is defined by the HR reconstruction loss:

$$\mathcal{L}_M^{\text{rec}} \triangleq \ell\left(\widehat{\mathbf{X}_{\text{HR}M}}, \mathbf{X}_{\text{HR}}\right). \tag{4.47}$$

The Master parameters are optimized to minimize the expected reconstruction loss over the HR training set. After this stage, the Master parameters are kept fixed (frozen) during the training of the SR model, and the Master AE is used to compute deterministic reference quantities $\mathbf{z}_M$ and $\widehat{\mathbf{X}_{\text{HR}M}}$ for each paired sample.

Follower AE reconstruction loss (SR training stage)

During SR training, the Follower AE $\text{AE}_F = (E_F, D_F)$ processes the LR input $\mathbf{Y}_{\text{LR}}^{(s)}$ and produces an LR-domain reconstruction:

$$\mathbf{z}_F = E_F\left(\mathbf{Y}_{\text{LR}}^{(s)}\right), \qquad \widehat{\mathbf{Y}_{\text{LR}F}}^{(s)} = D_F(\mathbf{z}_F) = D_F\left(E_F\left(\mathbf{Y}_{\text{LR}}^{(s)}\right)\right). \tag{4.48}$$

Since $\widehat{\mathbf{Y}_{\text{LR}F}}^{(s)}$ is the input to the upsampling CNN, it is explicitly supervised to remain consistent with the LR observation. The Follower AE reconstruction loss is defined as:

$$\mathcal{L}_F^{\text{rec}} \triangleq \ell\left(\widehat{\mathbf{Y}_{\text{LR}F}}^{(s)}, \mathbf{Y}_{\text{LR}}^{(s)}\right). \tag{4.49}$$

This term encourages the Follower AE to preserve the content of the LR input and to avoid producing an LR representation that is overly distorted for the subsequent upsampling stage.

Batch-wise empirical objectives

In practice, losses are computed over mini-batches. Let $\{(\mathbf{Y}_{\text{LR}}^{(s)}(i), \mathbf{X}_{\text{HR}}(i))\}_{i=1}^{B}$ denote a batch of size B for scale factor s. The empirical (batch-averaged) reconstruction

losses are:

$$\mathcal{L}^{\text{rec}}_{M,\text{batch}} \triangleq \frac{1}{B}\sum_{i=1}^{B} \ell\Big(\widehat{\mathbf{X}_{\text{HR}\,M}}(i), \mathbf{X}_{\text{HR}}(i)\Big), \tag{4.50}$$

$$\mathcal{L}^{\text{rec}}_{F,\text{batch}} \triangleq \frac{1}{B}\sum_{i=1}^{B} \ell\Big(\widehat{\mathbf{Y}_{\text{LR}\,F}}^{(s)}(i), \mathbf{Y}^{(s)}_{\text{LR}}(i)\Big), \tag{4.51}$$

where $\widehat{\mathbf{X}_{\text{HR}\,M}}(i) = D_M(E_M(\mathbf{X}_{\text{HR}}(i)))$ and $\widehat{\mathbf{Y}_{\text{LR}\,F}}^{(s)}(i) = D_F(E_F(\mathbf{Y}^{(s)}_{\text{LR}}(i)))$.

The reconstruction terms in this subsection will be combined with the latent guidance loss and the dual-target SR fidelity loss (defined in the next subsections) to form the complete TRAE training objective.

4.4.2 SR fidelity loss (pixel domain)

This subsection defines the SR fidelity loss applied at the output of the upsampling CNN. In TRAE, the SR output is supervised using a *dual-target* strategy: the predicted HR image is encouraged to match both the ground-truth HR image and a deterministic HR reference obtained from the pretrained (and frozen) Master AE. This design anchors the SR output to the true HR target while regularizing it toward an HR manifold captured by the Master AE.

SR output and reference targets

For a given scale factor $s \in \{2, 4, 8\}$, the trainable SR pipeline produces:

$$\begin{aligned}\widehat{\mathbf{X}_{\text{HR}}}^{(s)} &= G^{(s)}\left(\widehat{\mathbf{Y}_{\text{LR}\,F}}^{(s)}\right),\\ \widehat{\mathbf{Y}_{\text{LR}\,F}}^{(s)} &= D_F\Big(E_F\Big(\mathbf{Y}^{(s)}_{\text{LR}}\Big)\Big),\end{aligned} \tag{4.52}$$

where $\widehat{\mathbf{X}_{\text{HR}}}^{(s)} \in \mathbb{R}^{H\times W\times C}$.

In addition to the ground-truth HR target $\mathbf{X}_{\text{HR}}$, TRAE computes a deterministic HR reference through the frozen Master AE:

$$\widehat{\mathbf{X}_{\text{HR}\,M}} = D_M(E_M(\mathbf{X}_{\text{HR}})). \tag{4.53}$$

Since the Master AE is fixed after pretraining, $\widehat{\mathbf{X}_{\text{HR}\,M}}$ acts as a stable HR-oriented regularizer during SR training.

Dual-target SR fidelity loss

The SR fidelity loss is defined as a weighted sum of two pixel-domain fidelity terms:

$$\mathcal{L}_{\text{sr}} \triangleq \lambda_{\text{gt}}\,\ell\Big(\widehat{\mathbf{X}_{\text{HR}}}^{(s)}, \mathbf{X}_{\text{HR}}\Big) + \lambda_{\text{m}}\,\ell\Big(\widehat{\mathbf{X}_{\text{HR}}}^{(s)}, \widehat{\mathbf{X}_{\text{HR}\,M}}\Big), \tag{4.54}$$

where $\ell(\cdot,\cdot)$ denotes the chosen pixel-domain fidelity function (Section 4.4.1), and $\lambda_{\text{gt}} \geq 0$ and $\lambda_{\text{m}} \geq 0$ control the influence of the two targets.

The first term in (4.54) enforces agreement with the ground-truth HR image, ensuring that the SR output remains anchored to the true target. The second term encourages the SR output to remain consistent with the HR manifold captured by the pretrained Master AE, thereby acting as a deterministic regularizer that can promote stable reconstructions and suppress spurious high-frequency artifacts.

Batch-wise empirical form

For a mini-batch $\{(\mathbf{Y}_{\mathrm{LR}}^{(s)}(i), \mathbf{X}_{\mathrm{HR}}(i))\}_{i=1}^{B}$, the empirical SR fidelity loss is computed as:

$$\mathcal{L}_{\mathrm{sr,batch}} = \frac{1}{B}\sum_{i=1}^{B}\left[\lambda_{\mathrm{gt}}\,\ell\Big(\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}(i),\, \mathbf{X}_{\mathrm{HR}}(i)\Big) + \lambda_{\mathrm{m}}\,\ell\Big(\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}(i),\, \widehat{\mathbf{X}_{\mathrm{HR}\,M}}(i)\Big)\right], \tag{4.55}$$

where $\widehat{\mathbf{X}_{\mathrm{HR}\,M}}(i) = D_M(E_M(\mathbf{X}_{\mathrm{HR}}(i)))$ and $\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}(i) = G^{(s)}(\widehat{\mathbf{Y}_{\mathrm{LR}\,F}}^{(s)}(i))$.

The next subsection introduces the latent guidance loss term, which regularizes the Follower latent representation $\mathbf{z}_F$ using the Master latent reference $\mathbf{z}_M$.

4.4.3 Latent guidance loss

This subsection formalizes the latent guidance mechanism used in TRAE. The key idea is to encourage the latent representation extracted from the LR input by the Follower encoder to be consistent with an HR-informed latent reference produced by the pretrained Master encoder from the paired HR image. This constraint provides a perceptual-like regularization directly in the learned feature space of the Master AE.

Master and Follower latent codes

For each paired sample $(\mathbf{Y}_{\mathrm{LR}}^{(s)}, \mathbf{X}_{\mathrm{HR}})$, the Master encoder and the Follower encoder produce:

$$\begin{aligned}\mathbf{z}_M &= E_M(\mathbf{X}_{\mathrm{HR}}),\\ \mathbf{z}_F &= E_F\left(\mathbf{Y}_{\mathrm{LR}}^{(s)}\right),\end{aligned} \tag{4.56}$$

where $\mathbf{z}_M$ and $\mathbf{z}_F$ denote the Master and Follower latent representations, respectively. The Master encoder is fixed after pretraining; therefore, $\mathbf{z}_M$ acts as a deterministic latent reference during SR training, and only the Follower encoder parameters are updated to reduce the latent discrepancy.

Distance metric in latent space

Let $d(\cdot,\cdot)$ denote a distance measure in the latent space. Two standard choices are:

$$d_{\ell_1}(\mathbf{a},\mathbf{b}) \triangleq \frac{1}{d_z}\sum_{k=1}^{d_z}|a_k - b_k|\,, \tag{4.57}$$

$$d_{\ell_2}(\mathbf{a},\mathbf{b}) \triangleq \frac{1}{d_z}\sum_{k=1}^{d_z}(a_k - b_k)^2\,, \tag{4.58}$$

where $\mathbf{a},\mathbf{b}\in\mathbb{R}^{d_z}$ and d_z is the latent dimensionality. We use the generic notation $d(\cdot,\cdot)$; the specific choice used in the experiments is stated in the implementation details section.

Latent guidance loss definition

The latent guidance loss is defined by comparing the Follower latent code to the Master latent code:

$$\mathcal{L}_{\text{lat}} \triangleq d(\mathbf{z}_F,\mathbf{z}_M) = d\Big(E_F\Big(\mathbf{Y}_{\text{LR}}^{(s)}\Big)\,,\,E_M(\mathbf{X}_{\text{HR}})\Big)\,. \tag{4.59}$$

Because the Master encoder is frozen, $\mathbf{z}_M$ provides a stable HR-informed reference for each training pair. Minimizing $\mathcal{L}_{\text{lat}}$ therefore encourages the Follower encoder to produce representations that are aligned with the HR latent structure captured by the pretrained Master encoder.

Batch-wise empirical form

For a mini-batch $\{(\mathbf{Y}_{\text{LR}}^{(s)}(i),\mathbf{X}_{\text{HR}}(i))\}_{i=1}^{B}$, the empirical latent guidance loss is:

$$\mathcal{L}_{\text{lat,batch}} = \frac{1}{B}\sum_{i=1}^{B} d(\mathbf{z}_F(i),\,\mathbf{z}_M(i))\,, \tag{4.60}$$

where $\mathbf{z}_F(i) = E_F(\mathbf{Y}_{\text{LR}}^{(s)}(i))$ and $\mathbf{z}_M(i) = E_M(\mathbf{X}_{\text{HR}}(i))$.

In the next subsection, we combine the LR reconstruction loss, the latent guidance loss, and the dual-target SR fidelity loss into a single composite objective for training the TRAE model.

4.4.4 Full multi-term training objective

This subsection combines all loss components defined previously into a single composite objective for training the TRAE model at a fixed scale factor $s\in\{2,4,8\}$. TRAE does not employ additional feature-based perceptual losses. Instead, perceptual regularization is realized through the latent guidance term defined in Section 4.4.3, together with the dual-target SR fidelity supervision in the pixel domain.

Loss components

For a paired sample $(\mathbf{Y}_{\mathrm{LR}}^{(s)}, \mathbf{X}_{\mathrm{HR}})$, the following quantities are computed:

$$\begin{aligned} \mathbf{z}_M &= E_M(\mathbf{X}_{\mathrm{HR}}), \\ \widehat{\mathbf{X}_{\mathrm{HR}\,M}} &= D_M(\mathbf{z}_M), \\ \mathbf{z}_F &= E_F\left(\mathbf{Y}_{\mathrm{LR}}^{(s)}\right), \\ \widehat{\mathbf{Y}_{\mathrm{LR}\,F}}^{(s)} &= D_F(\mathbf{z}_F), \\ \widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)} &= G^{(s)}\left(\widehat{\mathbf{Y}_{\mathrm{LR}\,F}}^{(s)}\right). \end{aligned} \tag{4.61}$$

Using these variables, the individual loss terms are:

$$\mathcal{L}_F^{\mathrm{rec}} \triangleq \ell\left(\widehat{\mathbf{Y}_{\mathrm{LR}\,F}}^{(s)}, \mathbf{Y}_{\mathrm{LR}}^{(s)}\right), \tag{4.62}$$

$$\mathcal{L}_{\mathrm{lat}} \triangleq d(\mathbf{z}_F, \mathbf{z}_M), \tag{4.63}$$

$$\mathcal{L}_{\mathrm{sr}} \triangleq \lambda_{\mathrm{gt}}\, \ell\left(\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}, \mathbf{X}_{\mathrm{HR}}\right) + \lambda_{\mathrm{m}}\, \ell\left(\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}, \widehat{\mathbf{X}_{\mathrm{HR}\,M}}\right). \tag{4.64}$$

Composite objective for SR training

During SR training, the Master AE is kept fixed after pretraining; therefore, only the parameters of the Follower AE and the SR backbone are optimized. The full training objective for scale factor s is defined as the weighted sum:

$$\mathcal{L}_{\mathrm{total}}^{(s)} \triangleq \lambda_F^{\mathrm{rec}}\, \mathcal{L}_F^{\mathrm{rec}} + \lambda_{\mathrm{lat}}\, \mathcal{L}_{\mathrm{lat}} + \lambda_{\mathrm{sr}}\, \mathcal{L}_{\mathrm{sr}}, \tag{4.65}$$

where $\lambda_F^{\mathrm{rec}} \geq 0$, $\lambda_{\mathrm{lat}} \geq 0$, and $\lambda_{\mathrm{sr}} \geq 0$ control the influence of the LR reconstruction, latent guidance, and SR fidelity objectives respectively. Substituting (4.62)–(4.64) into (4.65) yields:

$$\begin{aligned} \mathcal{L}_{\mathrm{total}}^{(s)} = \lambda_F^{\mathrm{rec}}\, \ell\left(\widehat{\mathbf{Y}_{\mathrm{LR}\,F}}^{(s)}, \mathbf{Y}_{\mathrm{LR}}^{(s)}\right) + \lambda_{\mathrm{lat}}\, d(\mathbf{z}_F, \mathbf{z}_M) \\ + \lambda_{\mathrm{sr}}\left[\lambda_{\mathrm{gt}}\, \ell\left(\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}, \mathbf{X}_{\mathrm{HR}}\right) + \lambda_{\mathrm{m}}\, \ell\left(\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}, \widehat{\mathbf{X}_{\mathrm{HR}\,M}}\right)\right]. \end{aligned} \tag{4.66}$$

In (4.66), λ_{sr} controls the overall contribution of the SR output supervision, while λ_{gt} and λ_{m} distribute that contribution between the ground-truth HR target and the deterministic HR reference produced by the frozen Master AE. In practice, it is convenient to absorb λ_{sr} into λ_{gt} and λ_{m} (i.e., redefine effective weights), but we keep the explicit factorization in (4.66) for clarity.

Empirical risk minimization form

Let $\{(\mathbf{Y}_{\mathrm{LR}}^{(s)}(i), \mathbf{X}_{\mathrm{HR}}(i))\}_{i=1}^{B}$ be a mini-batch of size B for scale factor s. The batch-averaged objective is:

$$\mathcal{L}_{\mathrm{total,batch}}^{(s)} = \frac{1}{B} \sum_{i=1}^{B} \mathcal{L}_{\mathrm{total}}^{(s)}(i), \tag{4.67}$$

where $\mathcal{L}_{\text{total}}^{(s)}(i)$ denotes the per-sample loss computed using (4.66). The overall training procedure minimizes the expected value of this objective over the training distribution. Table 4.4 shows the loss terms, coefficients plus which stage each term is active (pretraining vs joint training).

Table 4.4. Loss terms used in TRAE, their coefficients, and the training stages in which they are active. The Master AE is optimized only during the pretraining stage and is frozen during SR training. During SR training, only the Follower AE and the upsampler $G^{(s)}(\cdot)$ are updated.

Loss term	Definition	Weight	Active stage(s)
Master reconstruction	$\mathcal{L}_M^{\text{rec}}$	–	Pretraining only
Follower reconstruction	$\mathcal{L}_F^{\text{rec}}$	λ_F^{rec}	SR training
Latent guidance	$\mathcal{L}_{\text{lat}}$	λ_{lat}	SR training
SR fidelity (dual-target)	$\mathcal{L}_{\text{sr}}$	λ_{sr}, λ_{gt}, λ_{m}	SR training
Total SR-training objective	$\mathcal{L}_{\text{total}}^{(s)}$	—	SR training

Role of the loss weights. The coefficients in Table 4.4 control the relative influence of the training objectives. The weight λ_F^{rec} enforces that the Follower output remains a content-preserving LR reconstruction, providing a reliable input for the upsampling CNN. The weight λ_{lat} determines the strength of the latent guidance constraint, i.e., how strongly the LR representation is regularized toward the HR-informed latent space defined by the frozen Master encoder. The SR fidelity term is controlled by λ_{gt} and λ_{m}, which balance adherence to the ground-truth HR target $\mathbf{X}_{\text{HR}}$ and consistency with the deterministic HR reference $\widehat{\mathbf{X}_{\text{HR}\,M}}$ produced by the Master AE. Increasing λ_{gt} prioritizes pixel-level agreement with the ground truth, whereas increasing λ_{m} strengthens the regularization toward stable HR-consistent reconstructions.

The subsequent sections detail the training strategy and provide the explicit training algorithm for optimizing (4.65) for each scale factor $s \in \{2, 4, 8\}$.

4.5 Training Strategy

This section describes the training procedure used for TRAE. The training is organized into two stages: (i) pretraining the Master AE on HR images and (ii) training the SR model for each scale factor with the Master kept fixed. Since the scale factors $s \in \{2, 4, 8\}$ correspond to different input resolutions, a separate TRAE model is trained for each scale.

4.5.1 Stage I: Master AE pretraining

The goal of Stage I is to learn an HR-consistent latent representation space using the Master AE $\text{AE}_M = (E_M, D_M)$. Let $\mathcal{X}_{\text{train}}$ denote the set of HR training images used for Master pretraining. For each HR sample $\mathbf{X}_{\text{HR}} \sim \mathcal{X}_{\text{train}}$, the Master AE

computes:

$$\begin{aligned} \mathbf{z}_M &= E_M(\mathbf{X}_{\mathrm{HR}}), \\ \widehat{\mathbf{X}_{\mathrm{HR}M}} &= D_M(\mathbf{z}_M), \end{aligned} \tag{4.68}$$

and the Master AE parameters are optimized by minimizing the reconstruction loss:

$$\mathcal{L}_M^{\mathrm{rec}} = \ell\left(\widehat{\mathbf{X}_{\mathrm{HR}M}}, \mathbf{X}_{\mathrm{HR}}\right). \tag{4.69}$$

The output of Stage I is a set of pretrained Master parameters, denoted by θ_M^*. These parameters are subsequently used to compute deterministic HR reference quantities during SR training.

4.5.2 Stage II: SR training

In Stage II, the Master AE is kept fixed and is used as a stable HR reference. Only the Follower AE $\mathrm{AE}_F = (E_F, D_F)$ and the scale-specific upsampling CNN $G^{(s)}(\cdot)$ are optimized. Let $s \in \{2, 4, 8\}$ be a fixed scale factor, and consider a paired sample $(\mathbf{Y}_{\mathrm{LR}}^{(s)}, \mathbf{X}_{\mathrm{HR}})$. The forward computations are:

$$\mathbf{z}_M = E_M(\mathbf{X}_{\mathrm{HR}}; \theta_M^*), \qquad \widehat{\mathbf{X}_{\mathrm{HR}M}} = D_M(\mathbf{z}_M; \theta_M^*), \tag{4.70}$$

$$\mathbf{z}_F = E_F\left(\mathbf{Y}_{\mathrm{LR}}^{(s)}\right), \qquad \widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)} = D_F(\mathbf{z}_F), \tag{4.71}$$

$$\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)} = G^{(s)}\left(\widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)}\right). \tag{4.72}$$

The trainable parameters in Stage II are those of the Follower AE and the upsampling CNN. For a given s, we denote the complete set of trainable parameters by $\theta^{(s)} \triangleq \{\theta_F, \theta_G^{(s)}\}$.

Losses used in Stage II

Stage II minimizes the composite objective defined in Section 4.4.4. For clarity, the principal loss components are recalled here:

- Follower reconstruction: $\mathcal{L}_F^{\mathrm{rec}} = \ell(\widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)}, \mathbf{Y}_{\mathrm{LR}}^{(s)})$. This term is included because $\widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)}$ is the direct input to the upsampling CNN.
- Latent guidance: $\mathcal{L}_{\mathrm{lat}} = d(\mathbf{z}_F, \mathbf{z}_M)$. This term regularizes the LR latent representation toward the HR-informed latent reference provided by the frozen Master encoder.
- SR fidelity (dual-target): $\mathcal{L}_{\mathrm{sr}} = \lambda_{\mathrm{gt}}\ell(\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}, \mathbf{X}_{\mathrm{HR}}) + \lambda_{\mathrm{m}}\ell(\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}, \widehat{\mathbf{X}_{\mathrm{HR}M}})$. This term anchors the SR output to the ground-truth HR target while regularizing it toward the deterministic HR reference produced by the Master AE.

The resulting objective is minimized with respect to $\theta^{(s)}$ while keeping θ_M^* fixed.

4.5.3 Separate training per scale factor

TRAE is trained independently for each scale factor $s \in \{2, 4, 8\}$. The scale-specific training setup differs in the LR input resolution and in the upsampling CNN architecture $G^{(s)}(\cdot)$ (Section 4.3.4), while the Master AE architecture remains the same and Follower AE and the upsampler are trained separately per scale (with scale-specific input size, and scale-specific layer configuration as per Table 4.2).

For each s, a dedicated training set of paired samples $\mathcal{P}^{(s)} = \{(\mathbf{Y}_{\mathrm{LR}}^{(s)}(i), \mathbf{X}_{\mathrm{HR}}(i))\}_{i=1}^{N}$ is constructed, and the corresponding model parameters $\theta^{(s)}$ are optimized by minimizing $\mathcal{L}_{\mathrm{total}}^{(s)}$.

Training separate models per scale offers two practical advantages in this framework. First, it allows the upsampling CNN to be tailored to the specific enlargement factor via the appropriate number of Depth-to-Space operations. Second, it avoids coupling gradients across different LR input sizes, thereby simplifying optimization and implementation. Once trained, each scale-specific TRAE model provides a deterministic SR mapping for its corresponding scale factor.

4.5.4 Summary of the training strategy

The TRAE training strategy can be summarized as follows:

1. Pretrain the Master AE on HR images using $\mathcal{L}_M^{\mathrm{rec}}$.
2. Freeze the Master parameters θ_M^*.
3. For each scale factor $s \in \{2, 4, 8\}$, train a separate SR model by optimizing the Follower AE and the scale-specific upsampler using the composite loss $\mathcal{L}_{\mathrm{total}}^{(s)}$, which combines LR reconstruction, latent guidance, and dual-target SR fidelity.

The next section provides explicit pseudo-code algorithms for Stage I and Stage II training and details the implementation choices used to optimize the TRAE objectives.

4.6 Training algorithms and implementation details

This section provides explicit training algorithms for (i) pretraining the Master auto-encoder and (ii) training the scale-specific TRAE models with a frozen Master. Unless otherwise stated, all losses use the Charbonnier penalty (a smooth and robust approximation of ℓ_1), and latent alignment is enforced using an ℓ_1 distance in the latent space.

4.6.1 Choice of loss norms and default settings

Pixel-domain loss (Charbonnier). For image-domain fidelity terms, we use the Charbonnier loss $\ell_{\mathrm{char}}(\cdot, \cdot)$ defined in (4.45), with a small constant $\epsilon = 10^{-3}$.

Latent-space distance (ℓ_1). For latent alignment, we use an ℓ_1 distance:

$$d(\mathbf{a}, \mathbf{b}) \triangleq \frac{1}{d_z} \sum_{k=1}^{d_z} |a_k - b_k| , \tag{4.73}$$

where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{d_z}$.

4.6.2 Algorithm 1: Master AE pretraining

The Master auto-encoder $\text{AE}_M = (E_M, D_M)$ is pretrained using HR images only. The objective is to minimize the HR reconstruction loss $\mathcal{L}_M^{\text{rec}} = \ell_{\text{char}}(\widehat{\mathbf{X}_{\text{HR}\,M}}, \mathbf{X}_{\text{HR}})$.

Algorithm 1 Master AE pretraining

Require: HR training set $\mathcal{X}_{\text{train}}$, batch size B, number of epochs E_M
Require: Optimizer Adam with learning rate η_M
Initialize Master parameters $\theta_M \leftarrow \{\theta_{E_M}, \theta_{D_M}\}$
for epoch $= 1$ to E_M **do**
  for each mini-batch $\{\mathbf{X}_{\text{HR}}(i)\}_{i=1}^{B}$ sampled from $\mathcal{X}_{\text{train}}$ **do**
    $\mathbf{z}_M(i) \leftarrow E_M(\mathbf{X}_{\text{HR}}(i); \theta_{E_M})$
    $\widehat{\mathbf{X}_{\text{HR}\,M}}(i) \leftarrow D_M(\mathbf{z}_M(i); \theta_{D_M})$
    $\mathcal{L}_M^{\text{rec}} \leftarrow \frac{1}{B} \sum_{i=1}^{B} \ell_{\text{char}}\Big(\widehat{\mathbf{X}_{\text{HR}\,M}}(i), \mathbf{X}_{\text{HR}}(i)\Big)$
    Update θ_M by one step of Adam on $\nabla_{\theta_M} \mathcal{L}_M^{\text{rec}}$
  end for
end for
Output: pretrained Master parameters $\theta_M^* \leftarrow \theta_M$

After Algorithm 1, the Master parameters are *frozen*, i.e., θ_M^* is kept fixed during SR training and used to compute deterministic references $\mathbf{z}_M$ and $\widehat{\mathbf{X}_{\text{HR}\,M}}$ for each paired sample.

4.6.3 Algorithm 2: scale-specific SR training

For each scale factor $s \in \{2, 4, 8\}$, TRAE trains a separate SR model composed of a scale-specific Follower AE: $\text{AE}_F^{(s)} = (E_F^{(s)}, D_F^{(s)})$ and the scale-specific upsampler $G^{(s)}(\cdot)$. The Master AE remains frozen and provides (i) the latent reference $\mathbf{z}_M = E_M(\mathbf{X}_{\text{HR}}; \theta_M^*)$ and (ii) the deterministic HR reference $\widehat{\mathbf{X}_{\text{HR}\,M}} = D_M(\mathbf{z}_M; \theta_M^*)$.

4.6.4 Optimization details

The following default settings are used: Adam optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$; η_M learning rate for Master pretraining and η_{SR} for SR training ($\eta_M = \eta_{SR} = 10^{-4}$, with decay scheduling); Charbonnier loss in the pixel domain with $\epsilon = 10^{-3}$ and ℓ_1 distance in latent space as in (4.73); Algorithm 1 is executed once to obtain the pretrained Master parameters θ_M^*. Then, Algorithm 2 is executed independently for each $s \in \{2, 4, 8\}$, training the scale-specific Follower AE $\text{AE}_F^{(s)}$ and the scale-specific upsampler $G^{(s)}(\cdot)$ to obtain three scale-specific SR models.

Algorithm 2 Scale-specific TRAE training

Require: Paired training set $\mathcal{P}^{(s)} = \{(\mathbf{Y}_{\mathrm{LR}}^{(s)}(i), \mathbf{X}_{\mathrm{HR}}(i))\}_{i=1}^{N}$ for a fixed scale s
Require: Master parameters θ_M^*, Batch size B, epochs $E_{SR}^{(s)}$, Adam with η_{SR}
Require: Weights $\lambda_F^{\mathrm{rec}}, \lambda_{\mathrm{lat}}, \lambda_{\mathrm{sr}}, \lambda_{\mathrm{gt}}, \lambda_{\mathrm{m}}$
Initialize trainable parameters $\theta_F^{(s)} \leftarrow \{\theta_{E_F^{(s)}}, \theta_{D_F^{(s)}}\}$ and $\theta_G^{(s)}$
for epoch = 1 to $E_{SR}^{(s)}$ **do**
  for each mini-batch $\{(\mathbf{Y}_{\mathrm{LR}}^{(s)}(i), \mathbf{X}_{\mathrm{HR}}(i))\}_{i=1}^{B}$ sampled from $\mathcal{P}^{(s)}$ **do**
    Master references (no gradient):
      $\mathbf{z}_M(i) \leftarrow E_M(\mathbf{X}_{\mathrm{HR}}(i); \theta_M^*)$
      $\widehat{\mathbf{X}_{\mathrm{HR}M}}(i) \leftarrow D_M(\mathbf{z}_M(i); \theta_M^*)$
    Follower forward (scale-specific):
      $\mathbf{z}_F(i) \leftarrow E_F^{(s)}(\mathbf{Y}_{\mathrm{LR}}^{(s)}(i); \theta_F^{(s)})$
      $\widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)}(i) \leftarrow D_F^{(s)}(\mathbf{z}_F(i); \theta_F^{(s)})$
    Upsampler forward (scale-specific):
      $\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}(i) \leftarrow G^{(s)}(\widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)}(i); \theta_G^{(s)})$
    Compute losses:
      $\mathcal{L}_F^{\mathrm{rec}} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \ell_{\mathrm{char}}\left(\widehat{\mathbf{Y}_{\mathrm{LR}F}}^{(s)}(i), \mathbf{Y}_{\mathrm{LR}}^{(s)}(i)\right)$
      $\mathcal{L}_{\mathrm{lat}} \leftarrow \frac{1}{B}\sum_{i=1}^{B} d(\mathbf{z}_F(i), \mathbf{z}_M(i))$
      $\mathcal{L}_{\mathrm{sr}} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \Big[\lambda_{\mathrm{gt}}\ell_{\mathrm{char}}\left(\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}(i), \mathbf{X}_{\mathrm{HR}}(i)\right) + \lambda_{\mathrm{m}}\ell_{\mathrm{char}}\left(\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}(i), \widehat{\mathbf{X}_{\mathrm{HR}M}}(i)\right)\Big]$
      $\mathcal{L}_{\mathrm{total}}^{(s)} \leftarrow \lambda_F^{\mathrm{rec}}\mathcal{L}_F^{\mathrm{rec}} + \lambda_{\mathrm{lat}}\mathcal{L}_{\mathrm{lat}} + \lambda_{\mathrm{sr}}\mathcal{L}_{\mathrm{sr}}$
    Update $(\theta_F^{(s)}, \theta_G^{(s)})$ by one step of Adam on $\nabla_{(\theta_F^{(s)}, \theta_G^{(s)})}\mathcal{L}_{\mathrm{total}}^{(s)}$
  end for
end for
Output: trained parameters $\theta_F^{(s)*}$ and $\theta_G^{(s)*}$ for scale s

4.7 Implementation details

4.7.1 Training dataset description

Dataset source. For training and evaluation of the TRAE on lung CT, we use the "COVIDx CT-2A" benchmark dataset introduced in [17].

COVIDx CT-2A contains 194,922 gray-scale chest CT slices acquired from a multinational cohort of 3,745 patients (age range: 0–93 years; median age: 51).

The dataset provides three clinically categories: normal controls, Common Pneumonia (CP), and Novel Coronavirus Pneumonia (NCP).

Although these labels are intended for classification, they are useful in SR to ensure that training data includes diverse lung appearance patterns.

An advantage of COVIDx CT-2A is its diversity: the cohort aggregates data collected from multiple organizations and initiatives across at least 16 countries, captured with different CT equipment and protocols, which increases the variability of anatomical appearance, noise characteristics, and reconstruction style.

Such diversity is desirable for SISR training because SR models can otherwise overfit to a narrow scanner/domain distribution.

COVIDx CT-2A is partitioned at the patient level, meaning each patient appears in only one split (training/validation/test), which prevents data leakage across splits.

Although COVIDx CT-2A provides more than 190k slices, training the proposed SR model does not require using all slices because (i) adjacent slices in a CT volume are highly correlated, and (ii) patch-based sampling can generate many distinct training examples from each slice. Instead, we select a subset of 50,000 CT slices with preserving patient-level split to maximize patient diversity and to limit redundancy.

The subset selection protocol is designed to satisfy two goals:

- generalization: prioritize many distinct patients/scanners rather than many nearly-identical neighboring slices.
- coverage of appearance variability: include slices spanning the dataset's three types to expose the SR model to a broad range of lung textures and pathologies, while keeping SR itself class-agnostic.

We build our dataset of size 50,000 to:

1. include as many distinct patients as possible.
2. to reduce redundancy.
3. maintain approximately the same class proportions (Normal/CP/NCP).

From the COVIDx CT-2A training split, the slice distribution is reported as Normal: 35,996, CP: 25,496, NCP: 82,286.

If we preserve these proportions in a 50,000 subset, we have approximately:

$$N_{\text{Normal}} \approx 12{,}500, \qquad N_{\text{CP}} \approx 9{,}000, \qquad N_{\text{NCP}} \approx 28{,}500.$$

This subset size is a compromise between computational feasibility and data diversity: with patient-level sampling and a per-patient cap, 50k slices can still represent a large fraction of the training-patient group, which improves generalization.

Consistency across SR models. To ensure a consistent comparison among the SR methods presented throughout this thesis, we use the same COVIDx CT-2A data source and the same patient-level training/validation/test partitioning strategy when training the other SR models introduced in other chapters. So, observed performance differences can be attributed mainly to architectural and loss-function design choices rather than differences in training data.

4.7.2 Data preparation and degradation model

This subsection describes how the paired LR/HR training samples are constructed.

Let $\mathbf{X}_{\mathrm{HR}} \in \mathbb{R}^{H\times W\times C}$ denote an HR image (or HR patch). For a given scale factor $s \in \{2, 4, 8\}$, the corresponding LR image $\mathbf{Y}_{\mathrm{LR}}^{(s)} \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times C}$ is generated synthetically by applying a degradation operator $\mathcal{T}_s(\cdot;\mathcal{D})$:

$$\mathbf{Y}_{\mathrm{LR}}^{(s)} = \mathcal{T}_s(\mathbf{X}_{\mathrm{HR}};\mathcal{D}). \tag{4.74}$$

Bicubic downsampling with antialiasing

The first component of the degradation operator is bicubic downsampling with antialiasing. Denote by $\mathcal{B}_s(\cdot)$ the bicubic downsampling operator at scale factor s, which maps an HR image to an LR image of spatial size $(H/s)\times(W/s)$:

$$\widetilde{\mathbf{Y}_{\mathrm{LR}}}^{(s)} = \mathcal{B}_s(\mathbf{X}_{\mathrm{HR}}). \tag{4.75}$$

Gaussian blur

A sample-dependent Gaussian blur is then applied to $\widetilde{\mathbf{Y}_{\mathrm{LR}}}^{(s)}$. Let $\mathcal{G}(\cdot;\sigma)$ denote a spatially invariant convolution with an isotropic Gaussian kernel of size 15×15 and standard deviation σ. For each training sample, σ is drawn uniformly at random from the interval [0.1, 1.5]:

$$\sigma \sim \mathcal{U}(0.1,\, 1.5), \qquad \overline{\mathbf{Y}_{\mathrm{LR}}}^{(s)} = \mathcal{G}\left(\widetilde{\mathbf{Y}_{\mathrm{LR}}}^{(s)};\sigma\right). \tag{4.76}$$

Additive mixed Gaussian–Laplacian noise

Finally, additive mixed Gaussian–Laplacian noise is injected. Let $\mathbf{n}$ denote i.i.d. noise with zero mean and variance ν. For each sample, the variance is drawn uniformly at random from [0, 0.01]:

$$\nu \sim \mathcal{U}(0,\, 0.01), \qquad \mathbf{Y}_{\mathrm{LR}}^{(s)} = \overline{\mathbf{Y}_{\mathrm{LR}}}^{(s)} + \mathbf{n}. \tag{4.77}$$

The mixed Gaussian–Laplacian noise model can be expressed as a convex mixture:

$$\mathbf{n} = \alpha\,\mathbf{n}_{\mathcal{N}} + (1-\alpha)\,\mathbf{n}_{\mathcal{L}}, \tag{4.78}$$

where $\mathbf{n}_{\mathcal{N}}$ is i.i.d. Gaussian noise, $\mathbf{n}_{\mathcal{L}}$ is i.i.d. Laplacian noise (scaled to the same variance ν), and $\alpha \in [0, 1]$ controls the mixture proportion. In the implementation, α is fixed, and the overall noise energy is controlled through ν.

Summary of the degradation operator

Combining (4.75)–(4.77), the degradation operator $\mathcal{T}_s(\cdot; \mathcal{D})$ can be summarized as:

$$\mathbf{Y}_{\mathrm{LR}}^{(s)} = \mathcal{T}_s(\mathbf{X}_{\mathrm{HR}}; \mathcal{D}) = \mathcal{G}(\mathcal{B}_s(\mathbf{X}_{\mathrm{HR}}); \sigma) + \mathbf{n}, \tag{4.79}$$

where $\mathcal{D} = \{s, \sigma, \nu, \alpha\}$ collects the degradation parameters.

The above construction yields paired samples $(\mathbf{Y}_{\mathrm{LR}}^{(s)}, \mathbf{X}_{\mathrm{HR}})$ for supervised SR training at scale s. When experiments are performed on grayscale datasets, the channel count is set to $C = 1$; the same procedure applies to multi-channel images by applying the degradation independently per channel.

4.7.3 Network configuration

This subsection specifies the exact architectural configuration of the TRAE components. TRAE consists of: (i) a pretrained and frozen Master AE AE_M, (ii) a scale-specific Follower AE $\mathrm{AE}_F^{(s)}$ for each $s \in \{2, 4, 8\}$, and (iii) a scale-specific, non-residual upsampling CNN $G^{(s)}(\cdot)$ for each scale factor.

Common layer notation and conventions

All convolutions and transpose convolutions in the auto-encoders use kernel size 3×3. We use the following notation:

- $\mathrm{Conv}(F, K, S)$: a 2D convolution with F output feature maps, kernel $K \times K$, and stride S.
- $\mathrm{TrC}(F, K, S)$: a 2D transpose convolution with F output feature maps, kernel $K \times K$, and stride S.
- D2S(2): Depth-to-Space (pixel shuffle) with block size 2.

A point-wise activation is applied after each convolution/transpose convolution layer, except for the final output layers of the decoders and upsamplers.

Activation function. We employ Parametric ReLU (PReLU) as the default activation, since it provides a learnable negative slope and typically improves convergence stability in SR-related CNNs. Concretely, for a scalar input x, $\mathrm{PReLU}(x) = \max(0, x) + a \min(0, x)$, where a is a learned parameter (per-channel).

Normalization. No batch normalization is used in the AEs or the upsampling CNN. Avoiding normalization in SR models is a common design choice to prevent undesirable shifts in image statistics and to preserve fine details. Consequently, the networks are composed only of convolution/transpose-convolution layers, nonlinearities, and Depth-to-Space operators.

Residual design of the AEs

Both the Master and Follower AEs follow a symmetric encoder–decoder design: the encoder progressively reduces spatial resolution using strided convolutions and increases the number of channels; the decoder mirrors this process using transpose convolutions to restore the original spatial resolution. In addition, the AE are *residual* in the sense that their decoders predict a residual correction that is added to the corresponding input:

$$\widehat{\mathbf{x}} = \mathbf{x} + R(\mathbf{x}), \tag{4.80}$$

where $R(\cdot)$ denotes the learnable encoder–decoder mapping (implemented by the AE). For the Master AE, $\mathbf{x} = \mathbf{X}_{\mathrm{HR}}$ and $\widehat{\mathbf{x}} = \widehat{\mathbf{X}_{\mathrm{HR}\,M}}$; for the Follower AE, $\mathbf{x} = \mathbf{Y}_{\mathrm{LR}}^{(s)}$ and $\widehat{\mathbf{x}} = \widehat{\mathbf{Y}_{\mathrm{LR}\,F}}^{(s)}$. This residual formulation encourages the AE to focus on reconstructing missing/incorrect components while retaining a direct information path from input to output. If the implementation uses direct prediction without an explicit outer residual addition, the mapping in (4.80) reduces to $\widehat{\mathbf{x}} = R(\mathbf{x})$. The layer-wise configurations remain valid in either case; the residual connection simply provides an additional identity path at the AE output.

Pretrained Master AE

The Master AE operates on HR patches of size $512 \times 512 \times C$ (with $C = 1$ for grayscale experiments). Its layer-wise encoder–decoder specification is reported in Table 4.2. The encoder consists of strided Conv$(\cdot)$ layers with increasing channel widths up to 512, followed by additional Conv$(512, 3, 1)$ layers at the bottleneck. The decoder mirrors the encoder using TrC$(\cdot)$ layers to recover the original spatial resolution.

Latent representation and latent dimensionality. Let the output of the Master encoder (after the last encoder layer) be a feature tensor $\mathbf{h}_M \in \mathbb{R}^{H_M \times W_M \times 512}$. To define a fixed-dimensional latent vector for latent guidance, we apply global average pooling over the spatial dimensions:

$$\mathbf{z}_M \triangleq \mathrm{GAP}(\mathbf{h}_M) \in \mathbb{R}^{512}, \tag{4.81}$$

where GAP$(\cdot)$ denotes global average pooling. Hence, the latent dimensionality is

$$d_z = 512. \tag{4.82}$$

The same pooling operator is used for the Follower encoder to ensure that $\mathbf{z}_F$ and $\mathbf{z}_M$ lie in the same vector space.

Scale-specific Follower AE

For each scale factor $s \in \{2, 4, 8\}$, the Follower AE processes LR patches of size $\frac{512}{s} \times \frac{512}{s} \times C$ and produces LR reconstructions of the same size. The Follower AEs are *scale-specific*: their layer-wise configurations (channel widths and stride patterns) depend on s and are presented in Table 4.2.

Follower latent representation. Let $\mathbf{h}_F^{(s)}$ denote the final encoder feature tensor of the scale-specific Follower AE. As for the Master, we define the latent vector by global average pooling:

$$\mathbf{z}_F \triangleq \mathrm{GAP}(\mathbf{h}_F^{(s)}) \in \mathbb{R}^{512}, \tag{4.83}$$

which ensures a consistent latent dimensionality $d_z = 512$ for all scales, enabling the latent guidance loss $d(\mathbf{z}_F, \mathbf{z}_M)$.

Scale-specific upsampling CNNs $G^{(s)}(\cdot)$

The SR backbone is implemented by a lightweight, feed-forward upsampling CNN without residual blocks or skip connections. A separate upsampler is trained for each scale factor $s \in \{2, 4, 8\}$, and its layer-wise configuration is shown in Table 4.3. The upsamplers are constructed from convolution layers followed by one or more D2S(2) operations:

- for $\times 2$: one D2S(2) stage,
- for $\times 4$: two cascaded D2S(2) stages,
- for $\times 8$: three cascaded D2S(2) stages.

The output layer projects to C channels using $\mathrm{Conv}(C, 5, 1)$, maintaining a channel-general formulation. In grayscale experiments, one sets $C = 1$.

Summary of configuration

The key architectural hyperparameters are:

- Residual blocks: none in $G^{(s)}(\cdot)$; the AEs are residual at the output level as in (4.80) and follow a symmetric encoder–decoder design (Table 4.2).
- Channel widths: as specified layer-by-layer in Table 4.2 for AE_M and $\mathrm{AE}_F^{(s)}$, and in Table 4.3 for $G^{(s)}(\cdot)$.
- Latent dimensionality: $d_z = 512$ obtained by global average pooling (4.82).
- Upsampling method: Depth-to-Space D2S(2) (pixel shuffle), cascaded to realize $\times 2$, $\times 4$, and $\times 8$ upsampling (Table 4.3).
- Activation: PReLU after each convolution/transpose convolution, except at the final output layers of the decoders/upsamplers.
- **Normalization:** none.

4.7.4 Optimization Details

This subsection summarizes the practical optimization settings used to train TRAE, including the optimizer choice, learning-rate scheduling, numerical stability measures, and regularization.

The Master AE is optimized only during Stage I (pretraining), while Stage II optimizes the scale-specific SR models (Follower AE and upsampler) with the Master kept frozen.

Optimizer

All trainable components are optimized using Adam. Let θ denote the set of trainable parameters in a given stage: $\theta = \theta_M$ during Master pretraining, and $\theta = \{\theta_F^{(s)}, \theta_G^{(s)}\}$ during SR training at scale s. Adam updates are applied with:

$$(\beta_1, \beta_2) = (0.9,\, 0.999), \tag{4.84}$$

and a small numerical constant ϵ_{adam} (default $\epsilon_{\text{adam}} = 10^{-8}$).

Learning rates and scheduling

We use a constant initial learning rate for each training stage and apply an adaptive scheduling strategy based on the validation set. Denote the initial learning rates by η_M for Master AE pretraining and η_{SR} for SR training. A practical default setting is:

$$\eta_M = 10^{-4}, \qquad \eta_{SR} = 10^{-4}. \tag{4.85}$$

Since validation data are available, the learning rate is reduced when the validation objective stops making progress. We monitor the validation loss and apply a Reduce-on-Plateau rule: if the validation loss does not improve for P consecutive epochs,

$$\eta \leftarrow \max(\eta_{\min},\, \gamma\, \eta), \tag{4.86}$$

where $0 < \gamma < 1$ is the reduction factor (e.g., $\gamma = 0.5$), P is the patience parameter (e.g., $P = 10$ epochs), and $\eta_{\min}$ is a lower bound (e.g., $\eta_{\min} = 10^{-6}$). This adaptive schedule is applied independently for each scale-specific SR model ($s = 2, 4, 8$), since the convergence behavior differs across scales.

Gradient clipping and numerical stability

To improve numerical stability and to prevent rare exploding-gradient events, global norm clipping is applied. Let $\mathbf{g} = \nabla_\theta \mathcal{L}$ denote the gradient vector (stacked over all trainable parameters). The clipped gradient is:

$$\mathbf{g} \leftarrow \mathbf{g} \cdot \min\left(1, \frac{\tau}{\|\mathbf{g}\|_2}\right), \tag{4.87}$$

where $\tau > 0$ is the clipping threshold. In our implementation, we use $\tau = 1.0$.

Regularization

Weight decay. A small ℓ_2 weight decay is used to improve generalization. We employ an AdamW-style [37] update with weight decay coefficient λ_{wd}:

$$\theta \leftarrow \theta - \eta\, \nabla_\theta \mathcal{L} - \eta\, \lambda_{\text{wd}}\, \theta, \tag{4.88}$$

with $\lambda_{\text{wd}} = 10^{-6}$ as a conservative default. Weight decay is applied to convolutional weights and excluded for bias terms and PReLU parameters.

Dropout and pooling. No dropout or pooling layers are used in the TRAE architectures. This choice avoids injecting stochasticity into the feature representations and preserves spatial detail, which is critical for SR.

The Master AE is pretrained first and then frozen; SR training updates only the scale-specific Follower AE $\mathrm{AE}_F^{(s)}$ and the scale-specific upsampler $G^{(s)}(\cdot)$. The complete SR optimization process is repeated independently for $s = 2$, $s = 4$, and $s = 8$, including validation-based learning-rate scheduling and checkpoint selection.

Hyperparameter summary (per scale)

Table 4.5 summarizes the key hyperparameters used for the TRAE training pipeline. Values that are scale-specific are indicated explicitly.

Table 4.5. Summary of TRAE hyperparameters (reported per scale factor $s \in \{2, 4, 8\}$). Symbols follow Sections 4.4–4.6. Grayscale experiments use $C = 1$, while the formulation remains channel-general.

Category	Hyperparameter	Value / Note
Data	HR patch size ($H \times W$)	512×512
Data	Scale factors $\mathcal{S}$	$\{2, 4, 8\}$
Data	Channels C	General; grayscale experiments: $C = 1$
Loss	Pixel fidelity $\ell(\cdot, \cdot)$	Charbonnier, $\epsilon = 10^{-3}$
Loss	Latent distance $d(\cdot, \cdot)$	ℓ_1 in latent space
Loss	Loss weights	$\lambda_F^{\mathrm{rec}}, \lambda_{\mathrm{lat}}, \lambda_{\mathrm{sr}}, \lambda_{\mathrm{gt}}, \lambda_{\mathrm{m}}$
Optimizer	Optimizer	Adam
Optimizer	Adam parameters	$(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon_{\mathrm{adam}} = 10^{-8}$
Stability	Gradient clipping	Global norm, $\tau = 1.0$
Regularization	Weight decay (AdamW-style)	$\lambda_{\mathrm{wd}} = 10^{-6}$ (conservative default)
Regularization	Dropout / pooling	None
Validation	Checkpoint selection	Best validation loss per scale

4.8 How TRAE supports the thesis goals

This section connects the TRAE design choices to the central goals of this thesis. The intent is to clarify how the proposed architecture and training strategy align with a regression-based SR formulation that emphasizes stable optimization and perceptual

fidelity, without turning this chapter into an extended discussion of experimental outcomes.

4.8.1 Regression-Based SR without generative hallucination

TRAE performs SR through a deterministic regression mapping trained with supervised losses. For a given scale factor $s \in \{2, 4, 8\}$, the SR estimate is produced by:

$$\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)} = G^{(s)}\Big(D_F^{(s)}\Big(E_F^{(s)}(\mathbf{Y}_{\mathrm{LR}}^{(s)})\Big)\Big), \tag{4.89}$$

where $\mathbf{Y}_{\mathrm{LR}}^{(s)}$ is the LR input and $\widehat{\mathbf{X}_{\mathrm{HR}}}^{(s)}$ is the predicted HR output. The mapping in (4.89) is fully deterministic at inference time, and its parameters are learned by minimizing a composite supervised objective $\mathcal{L}_{\mathrm{total}}^{(s)}$ (Section 4.4.4). Consequently, the SR output is not produced by a stochastic generative process and is not driven by adversarial training.

In addition, TRAE incorporates a fixed HR reference through the pretrained and frozen Master AE. The Master encoder provides a deterministic latent reference $\mathbf{z}_M = E_M(\mathbf{X}_{\mathrm{HR}})$ for each training pair, which is used to regularize the LR latent representation. Since the Master AE is fixed during SR training, the reference latent space does not drift during optimization, and the overall learning remains within a supervised regression paradigm.

4.8.2 Perceptual quality via stable latent guidance

A central motivation of TRAE is to improve perceptual fidelity while maintaining training stability.

Rather than employing an external feature extractor (e.g., VGG) or adversarial training, TRAE introduces a latent guidance mechanism based on a pretrained autoencoder. For each paired sample, the frozen Master encoder extracts an HR-informed latent code $\mathbf{z}_M = E_M(\mathbf{X}_{\mathrm{HR}})$, while the scale-specific Follower encoder extracts an LR latent code $\mathbf{z}_F = E_F^{(s)}(\mathbf{Y}_{\mathrm{LR}}^{(s)})$. The latent guidance loss

$$\mathcal{L}_{\mathrm{lat}} = d(\mathbf{z}_F, \mathbf{z}_M) \tag{4.90}$$

encourages the LR representation to align with the HR-informed structure captured by the Master.

Intuitively, the Master AE defines a representation space that is tuned to the statistics of HR images in the training domain. By pulling the Follower latent toward this fixed HR reference, TRAE shapes the internal representation of LR inputs toward HR-consistent structure. This guidance acts as a perceptual-like regularizer. It constrains the learned representation beyond pixel-level matching, while remaining stable because the Master AE reference is deterministic and frozen. As a result, the SR pipeline can improve the coherence of reconstructed structures while preserving the optimization advantages of regression-based learning.

4.8.3 Expected behavior and design trade-offs

TRAE combines multiple loss terms (LR reconstruction, latent guidance, and dual-target SR fidelity) to balance fidelity and stability. This design introduces several high-level trade-offs:

- Fidelity versus regularization: increasing the weight of the latent guidance term λ_{lat} and the Master-reference SR term λ_{m} strengthens regularization toward HR-consistent representations and outputs, which can improve stability and suppress spurious artifacts. However, overly strong regularization may bias the SR output toward smoother reconstructions, potentially limiting very fine high-frequency detail recovery.
- LR consistency versus SR flexibility: The Follower AE reconstruction term $\mathcal{L}_F^{\text{rec}}$ ensures that the LR reconstruction fed to the upsampler remains faithful to the LR observation. While this improves robustness and interpretability of the pipeline, excessively emphasizing LR reconstruction may restrict the Follower representation in a way that reduces flexibility for the downstream upsampler.
- Ground-truth anchoring versus teacher consistency: The dual-target SR fidelity loss balances agreement with the ground-truth HR image $\mathbf{X}_{\text{HR}}$ and consistency with the deterministic Master reconstruction $\widehat{\mathbf{X}_{\text{HR}\,M}}$. This encourages stable, HR-plausible outputs while keeping the SR estimate anchored to the true target. The relative weights λ_{gt} and λ_{m} control this balance.

4.8.4 Limitations

Although TRAE provides a stable GAN-free mechanism for introducing HR-informed latent guidance, the approach has some practical limitations.

First, the effectiveness of the latent guidance depends on the quality of the pretrained Master AE. Since the Master AE is trained using a pixel-domain reconstruction objective, its latent space may emphasize dominant low-frequency and structurally stable content while under-representing very fine or ambiguous high-frequency details. So, the Master latent representation should be considered as an HR-informed regularizer rather than a perfect perceptual or semantic representation.

Second, the latent alignment term improves internal consistency between the LR and HR pathways, but the semantic meaning of each latent dimension is not explicitly interpretable. The alignment loss enforces closeness between the Follower and Master latent codes, but it does not identify which anatomical structures, edges, or texture components are encoded in each latent component. A more detailed analysis, for example through latent-space visualization, would be future work.

Third, TRAE requires scale-specific training. Separate Follower AEs and upsamplers are trained for different scale factors, which simplifies the optimization and allows each model to specialize to a fixed enlargement factor, but it limits scalability when many scale factors must be supported. This limitation motivates the multi-resolution extension presented in Chapter 5, where multiple output resolutions are handled

within a single framework.

Finally, the dual AE design increases the training complexity relative to a single CNN-based SR model. The Master AE must be pretrained first, frozen, and then used as a reference during SR training. This introduces additional parameters and training stages. However, at inference time the HR input and the Master AE are not required; the prediction is produced only from the LR input through the Follower AE and the scale-specific upsampler. Thus, the main complexity overhead is concentrated in the training phase, while inference remains deterministic and feed-forward. These trade-offs are acceptable in the present thesis because the goal is not only maximum compactness, but also stable regression-based SR with improved perceptual and structural behavior in medical imaging.

4.9 Chapter Summary

This chapter presented the TRAE framework as a regression-based approach to SISR that combines a twinned AE design with a lightweight, scale-specific upsampling CNN. The model is composed of three main modules: a pretrained and frozen Master AE $\mathrm{AE}_M = (E_M, D_M)$ operating on HR images, a scale-specific Follower AE $\mathrm{AE}_F^{(s)} = (E_F^{(s)}, D_F^{(s)})$ operating on LR inputs, and a non-residual scale-specific upsampler $G^{(s)}(\cdot)$ that maps the Follower AE reconstruction to the final SR output.

The training objective was defined as a multi-term loss that integrates: (i) an LR reconstruction loss for the Follower auto-encoder, (ii) a latent guidance loss that aligns the Follower latent representation with the deterministic HR-informed latent reference extracted by the frozen Master encoder, and (iii) a dual-target SR fidelity loss that anchors the SR output to the ground-truth HR image while encouraging consistency with the deterministic HR reference produced by the Master decoder. Importantly, TRAE does not rely on adversarial learning or external feature-based perceptual losses; perceptual regularization is realized through the stable latent guidance mechanism and the dual-target supervision at the SR output.

A two-stage training strategy was described. In Stage I, the Master auto-encoder is pretrained using HR reconstruction to learn an HR-consistent representation space. In Stage II, the Master is frozen and used as a fixed reference, while separate SR models are trained independently for each scale factor $s \in \{2, 4, 8\}$ by optimizing the corresponding Follower AE and upsampler under the full composite objective.

Chapter 5

Multi-resolution Twinned Residual Auto-Encoders (MR-TRAE)

5.1 Overview

This chapter introduces the MR-TRAE model, a multi-resolution extension of the TRAE framework (Chapter 4) for SISR in medical imaging. The key idea of MR-TRAE is to produce *multiple super-resolved outputs at different spatial resolutions* in a *single* end-to-end model, trained once, while retaining the stability and determinism of regression-based learning.

5.1.1 From TRAE to MR-TRAE

The TRAE framework introduced the idea of using two AEs: a *Master* AE trained on HR data and a *Follower* AE trained on LR data, where the Follower AE is guided by the Master AE through a latent-alignment constraint.

This design is particularly attractive for SISR because it provides a principled way to inject HR structural knowledge into the LR domain without relying on external perceptual networks. Nevertheless, the original TRAE setting is naturally aligned with producing an SR output at a single resolution (per model or per training configuration).

The proposed MR-TRAE extends this philosophy to a *cascaded* SR setting where a single training run produces multiple SR outputs. Concretely, MR-TRAE fixes the LR input spatial size to 64×64 and employs a sequence of three $\times 2$ SR CNN backbones, yielding:

$$64 \times 64 \xrightarrow{\times 2} 128 \times 128 \xrightarrow{\times 2} 256 \times 256 \xrightarrow{\times 2} 512 \times 512. \tag{5.1}$$

The key point is that MR-TRAE *does not* treat the first two SR stages as merely intermediate computational steps. Instead, each stage produces an explicit super-resolved image that is *directly supervised* during training.

This multi-resolution objective is central: it transforms the cascade into a *multi-task* learner where each resolution constitutes a task, and the tasks are coupled through the sequential architecture.

5.1.2 Motivation: why multi-resolution SR?

Many medical imaging workflows involve interpretation at multiple zoom levels and resolutions. From a learning perspective, explicitly supervising intermediate SR outputs can encourage *scale-consistent reconstruction*, meaning that the anatomical structure reconstructed at 128×128 remains compatible with the refinements at 256×256 and the final 512×512 output. This multi-scale supervision principle has been explored in the SR literature to improve robustness and generalization, particularly when a model must support multiple reconstruction scales [64]. In MR-TRAE, we incorporate multi-resolution supervision directly into the architecture and the objective, while preserving paired, deterministic training.

The choice of three target scales in MR-TRAE is mainly determined by the experimental setting used in this thesis. The LR input is fixed at 64×64, while the final HR target is 512×512. So, the overall enlargement factor is $\times 8 = 2^3$. A cascade of three $\times 2$ SR modules therefore provides a natural decomposition of the reconstruction path, producing outputs at 128×128, 256×256, and 512×512. These resolutions are also useful in practice because they correspond to progressively finer levels of image interpretation. If a different number of target scales were required, the same principle could be extended by adding or removing $\times 2$ SR stages and their corresponding supervision losses. Fewer stages would reduce computational cost but provide weaker intermediate supervision, whereas more stages could support finer multi-scale outputs but would increase memory usage, parameter count, and the risk of accumulated reconstruction errors across the cascade.

The key contributions of MR-TRAE can be summarized as follows:

- A multi-resolution SR formulation within a single forward pass: unlike conventional SISR models that are optimized to output a single HR image for a fixed scale factor, MR-TRAE produces a *set* of SR outputs from a single LR input by cascading three $\times 2$ CNN backbones. Importantly, the intermediate outputs are not treated as auxiliary or implicit representations; they are *explicit predictions* that are directly supervised during training. This multi-resolution objective is beneficial in practice because it (i) encourages the early SR stages to learn meaningful reconstructions at their own scale, and (ii) enforces cross-scale coherence between the three outputs.
- A two-stage teacher–student strategy based on TRAE: MR-TRAE adopts a staged training procedure: (Stage I) a *Master* AE is trained on HR images and then frozen; (Stage II) a *Follower* is trained on LR images while being guided by the frozen Master. The guidance is enforced by a latent-alignment loss between the Follower latent and the Master latent. This mechanism transfers HR structural priors into the LR representation space without introducing external perceptual networks, which is especially attractive in medical imaging scenarios where domain mismatch can be a concern.

- SR from an HR-informed reconstructed LR input: in MR-TRAE, the SR cascade is fed with the reconstructed output of the Follower AE, rather than the raw LR input. This design provides a principled form of regularization: the Follower AE is trained to reconstruct plausible LR images while its latent is aligned with the HR-informed latent of the Master AE. Consequently, the SR backbones receive a stabilized and structured input signal, which can reduce the amplification of LR noise/artifacts and improve training robustness.
- A Master-guided perceptual loss without external feature extractors: MR-TRAE uses the frozen Master encoder as a feature extractor to define perceptual losses. Since both predictions and targets are passed through the same frozen encoder, the resulting feature comparisons are naturally aligned in channel dimensions and capture structural information that is learned from the HR data distribution. This provides a perceptual regularization effect, avoiding reliance on generic networks pretrained on unrelated natural-image datasets.
- Improved optimization via intermediate supervision and better gradient flow: MR-TRAE improves gradient propagation to the earlier SR modules. This reduces the risk of error accumulation in deep cascades and encourages each stage to be individually accurate. In turn, this tends to yield more stable convergence and outputs that remain coherent across resolutions.

5.1.3 Problem setting and notation (high-level).

Let an HR image be denoted by $\mathbf{X}_{\mathrm{HR}} \in \mathbb{R}^{H\times W\times C}$, and its associated LR observation by $\mathbf{Y}_{\mathrm{LR}} \in \mathbb{R}^{h\times w\times C}$, where C is the number of channels.

In our implementation, images are grayscale lung CTs (thus $C = 1$), but all mathematical definitions in this chapter are written for a general C. In the MR-TRAE setting considered in this thesis, $(H, W) = (512, 512)$ and $(h, w) = (64, 64)$, hence the overall super-resolution scale is $s = 8$.

Unlike conventional SR pipelines that only aim to reconstruct a single HR output, MR-TRAE is trained to predict *three explicit super-resolved images*:

$$\widehat{\mathbf{X}}_2 \in \mathbb{R}^{128\times 128\times C}, \qquad \widehat{\mathbf{X}}_4 \in \mathbb{R}^{256\times 256\times C}, \qquad \widehat{\mathbf{X}}_8 \in \mathbb{R}^{512\times 512\times C}, \tag{5.2}$$

where $\widehat{\mathbf{X}}_2$, $\widehat{\mathbf{X}}_4$, and $\widehat{\mathbf{X}}_8$ denote the $\times 2$, $\times 4$, and $\times 8$ super-resolved predictions, respectively. A crucial point is that $\widehat{\mathbf{X}}_2$ and $\widehat{\mathbf{X}}_4$ are *not* merely intermediate tensors used only to reach the final $\times 8$ output; rather, each $\widehat{\mathbf{X}}_2$, $\widehat{\mathbf{X}}_4$, and $\widehat{\mathbf{X}}_8$ is treated as a valid SR image and is supervised by resolution-matched losses.

Architectural idea: twinned residual AEs with sequential $\times 2$ SR CNNs. At a high level, MR-TRAE comprises three components:

- Master AE: trained first on $\mathbf{X}_{\mathrm{HR}}$ (Stage I) to learn an HR-oriented latent space. During the main training stage (Stage II), the Master AE is frozen and used as a fixed reference for latent guidance and perceptual losses.
- Follower AE: trained in Stage II with $\mathbf{Y}_{\mathrm{LR}}$ as input, producing a reconstructed LR image (denoted by $\widehat{\mathbf{Y}}_F \in \mathbb{R}^{h\times w\times C}$). The Follower AE is optimized using

(i) an LR reconstruction loss and (ii) a latent alignment loss that encourages its latent representation to match the latent representation produced by the frozen Master AE when processing the paired $\mathbf{X}_{\text{HR}}$.

- Three sequential $\times 2$ SR CNN backbones, called: CNN1, CNN2, and CNN3: CNN1 takes $\widehat{\mathbf{Y}}_F$ (not the raw $\mathbf{Y}_{\text{LR}}$) and outputs $\widehat{\mathbf{X}}_2$; CNN2 refines $\widehat{\mathbf{X}}_2$ to $\widehat{\mathbf{X}}_4$; and CNN3 produces the final $\widehat{\mathbf{X}}_8$. Each CNN is supervised at its own output resolution using both pixel-domain and perceptual losses.

Two-stage training strategy. Training is performed in two stages:

1. Stage I (pretraining): the Master AE is trained on HR images to minimize an HR reconstruction objective.
2. Stage II (MR-TRAE training): the Master AE is frozen. The Follower AE and CNN1–CNN3 are trained jointly using paired $(\mathbf{Y}_{\text{LR}}, \mathbf{X}_{\text{HR}})$ samples and a composite objective consisting of: (i) Follower LR reconstruction, (ii) latent guidance toward the frozen Master latent, and (iii) multi-resolution SR losses (pixel and perceptual) for $\widehat{\mathbf{X}}_2$, $\widehat{\mathbf{X}}_4$, and $\widehat{\mathbf{X}}_8$.

By freezing the Master AE in Stage II, the perceptual and latent reference does not drift during optimization, which contributes to stable and reproducible training behavior compared to adversarial formulations.

Summary of significance and expected benefits. The MR-TRAE strategy provides the following benefits:

- Multi-resolution usability: the model provides three SR outputs which can be beneficial when different resolutions are required for visualization, storage, or downstream analysis.
- Scale-consistent learning: explicit supervision at intermediate resolutions constrains the model and encourages structural consistency across scales.
- Stable perceptual guidance without adversarial training: perceptual constraints are derived from a frozen, domain-trained AE, reducing the risk of hallucinated structures and improving optimization stability.
- Single unified training run: all three SR outputs are learned jointly within one model, instead of training separate models for each scale.

Chapter organization. The remainder of this chapter is organized as follows. Section 5.2 formalizes the problem definition, notation, and the construction of intermediate-resolution supervision targets. Section 5.3 details the MR-TRAE architecture, including the Master/Follower AEs and the three sequential $\times 2$ SR CNN backbones. Section 5.4 defines the complete training objective, including latent guidance and multi-resolution pixel/perceptual losses. Section 5.5 presents the two-stage training strategy and optimization procedure. Section 5.7 reports the implementation details required for reproducibility. Finally, Section 5.8 discusses how MR-TRAE supports the overall goals of the thesis.

5.2 Problem setting and notation

This section formalizes the learning problem addressed by MR-TRAE and introduces the notation used throughout Chapter 5.

5.2.1 Paired HR–LR training samples

Let the ground-truth HR image be denoted by $\mathbf{X}_{\mathrm{HR}} \in \mathbb{R}^{H\times W\times C}$, and the corresponding LR observation by $\mathbf{Y}_{\mathrm{LR}} \in \mathbb{R}^{h\times w\times C}$, where C denotes the number of channels.

Although our experiments use grayscale lung CT images (thus $C = 1$ in implementation), all mathematical definitions in this chapter are written for general C.

In this thesis, the MR-TRAE model is trained using paired samples

$$\mathcal{D} = \left\{ \left(\mathbf{Y}_{\mathrm{LR}}^{(i)}, \mathbf{X}_{\mathrm{HR}}^{(i)} \right) \right\}_{i=1}^{N}, \tag{5.3}$$

where $\mathbf{X}_{\mathrm{HR}}^{(i)} \in \mathbb{R}^{512\times 512\times C}$ and $\mathbf{Y}_{\mathrm{LR}}^{(i)} \in \mathbb{R}^{64\times 64\times C}$. Hence, the overall scale factor is fixed to

$$s = \frac{H}{h} = \frac{W}{w} = 8. \tag{5.4}$$

For some derivations, it is convenient to use vectorized forms $\mathbf{x}_{\mathrm{HR}} \in \mathbb{R}^{HWC}$ and $\mathbf{y}_{\mathrm{LR}} \in \mathbb{R}^{hwC}$, which are obtained by reshaping $\mathbf{X}_{\mathrm{HR}}$ and $\mathbf{Y}_{\mathrm{LR}}$, respectively, without changing the ordering of elements. Unless explicitly stated, we primarily work with tensor forms because CNN-based models are naturally defined on spatial grids.

5.2.2 LR degradation model and HR-to-LR operator

The LR observation $\mathbf{Y}_{\mathrm{LR}}$ is assumed to be generated from $\mathbf{X}_{\mathrm{HR}}$ through a degradation operator. In general form, we denote this process as

$$\mathbf{Y}_{\mathrm{LR}} = \mathcal{G}_s(\mathbf{X}_{\mathrm{HR}}), \tag{5.5}$$

where $\mathcal{G}_s(\cdot)$ maps an HR image to an LR image at overall scale s. In real-world medical imaging, the mapping from HR to LR is influenced by multiple factors (e.g., scanner physics, reconstruction kernel, noise, and sampling), and it is common to model $\mathcal{G}_s$ as a composition of operations.

Following the TRAE setting used earlier in Chapter 4, we adopt a realistic degradation model that can be expressed as

$$\mathbf{Y}_{\mathrm{LR}} = \mathcal{D}_s\Big(\mathcal{N}(\mathcal{B}(\mathbf{X}_{\mathrm{HR}}))\Big), \tag{5.6}$$

where: (i) $\mathcal{B}(\cdot)$ denotes a blur operator (e.g., a Gaussian kernel or a small set of plausible blur kernels), (ii) $\mathcal{N}(\cdot)$ denotes a noise operator (e.g., additive Gaussian noise or signal-dependent noise), and (iii) $\mathcal{D}_s(\cdot)$ denotes downsampling by factor s. The above decomposition is intentionally generic; in Section 5.7 we specify the exact

choices (kernel size, noise distribution, and the downsampling implementation). This formulation allows $\mathbf{Y}_{\text{LR}}$ to reflect realistic acquisition artifacts rather than being a purely bicubic downsampled version of $\mathbf{X}_{\text{HR}}$, which is important for *realistic* SISR settings [4].

5.2.3 Intermediate-resolution supervision targets

A distinctive aspect of MR-TRAE is that the model produces three super-resolved outputs at different spatial sizes. To train these outputs, we construct resolution-matched supervision targets derived from the same ground-truth $\mathbf{X}_{\text{HR}}$.

Let $\mathcal{S}_r(\cdot)$ denote a *deterministic antialiased downsampling* operator that downsamples an image by factor r (with appropriate low-pass filtering to prevent aliasing). Using $\mathcal{S}_r$, we define the intermediate ground-truth targets:

$$\mathbf{X}_{256} = \mathcal{S}_2(\mathbf{X}_{\text{HR}}) \in \mathbb{R}^{256\times256\times C}, \qquad \mathbf{X}_{128} = \mathcal{S}_4(\mathbf{X}_{\text{HR}}) \in \mathbb{R}^{128\times128\times C}. \tag{5.7}$$

In other words, $\mathbf{X}_{256}$ and $\mathbf{X}_{128}$ are derived directly from $\mathbf{X}_{\text{HR}}$ and represent the "ideal" images at those resolutions. This choice is deliberate: the intermediate outputs of MR-TRAE are intended to be *valid reconstructions at their native resolution*, rather than additional degraded observations. Thus, $\mathcal{S}_r$ is chosen as a high-quality, deterministic operator (e.g., bicubic with antialiasing, or area-based resampling), while the operator $\mathcal{G}_s$ in (5.5) may include blur/noise to model realistic LR acquisition.

We provide further justification and implementation details in Section 5.7.

5.2.4 Multi-resolution SR outputs and model mapping

Given $\mathbf{Y}_{\text{LR}} \in \mathbb{R}^{64\times64\times C}$, the MR-TRAE model predicts three super-resolved images:

$$\widehat{\mathbf{X}}_2 \in \mathbb{R}^{128\times128\times C}, \qquad \widehat{\mathbf{X}}_4 \in \mathbb{R}^{256\times256\times C}, \qquad \widehat{\mathbf{X}}_8 \in \mathbb{R}^{512\times512\times C}, \tag{5.8}$$

corresponding to scale factors $\times2$, $\times4$, and $\times8$ with respect to the LR input.

We denote the overall model by a parameterized mapping $f_{\boldsymbol{\theta}}(\cdot)$ such that

$$(\widehat{\mathbf{X}}_2, \widehat{\mathbf{X}}_4, \widehat{\mathbf{X}}_8) = f_{\boldsymbol{\theta}}(\mathbf{Y}_{\text{LR}}). \tag{5.9}$$

A central principle of MR-TRAE is that each output in (5.8) is a *final* prediction at its corresponding resolution and is explicitly supervised by resolution-matched objectives: $\widehat{\mathbf{X}}_2$ is supervised against $\mathbf{X}_{128}$, $\widehat{\mathbf{X}}_4$ is supervised against $\mathbf{X}_{256}$, and $\widehat{\mathbf{X}}_8$ is supervised against $\mathbf{X}_{\text{HR}}$. This differs from cascaded SR pipelines in which intermediate stages exist only to improve the final 512×512 output.

5.2.5 Summary of symbols introduced in this section

For convenience, Table 5.1 summarizes the main symbols introduced in Section 5.2.

Table 5.1. Summary of symbols and notation used in Chapter 5.

Symbol	Meaning	Typical size
$\mathbf{X}_{\mathrm{HR}}$	Ground-truth high-resolution (HR) image tensor.	$H \times W \times C$
$\mathbf{Y}_{\mathrm{LR}}$	Observed low-resolution (LR) image tensor paired with $\mathbf{X}_{\mathrm{HR}}$.	$h \times w \times C$
$\mathbf{x}_{\mathrm{HR}}$	Vectorized form of $\mathbf{X}_{\mathrm{HR}}$ (obtained by reshaping).	$HWC \times 1$
$\mathbf{y}_{\mathrm{LR}}$	Vectorized form of $\mathbf{Y}_{\mathrm{LR}}$ (obtained by reshaping).	$hwC \times 1$
C	Number of channels (kept general; $C = 1$ for grayscale CT implementation).	—
s	Overall scale factor between HR and LR.	$s = \frac{H}{h} = \frac{W}{w}$
$\mathcal{D}$	Paired dataset of training samples $\{(\mathbf{Y}_{\mathrm{LR}}^{(i)}, \mathbf{X}_{\mathrm{HR}}^{(i)})\}_{i=1}^{N}$.	—
$\mathcal{G}_s(\cdot)$	HR-to-LR degradation operator mapping $\mathbf{X}_{\mathrm{HR}} \mapsto \mathbf{Y}_{\mathrm{LR}}$ at overall scale s (may include blur/noise + downsampling).	—
$\mathcal{B}(\cdot)$	Blur operator used in a realistic degradation model.	—
$\mathcal{N}(\cdot)$	Noise operator used in a realistic degradation model.	—
$\mathcal{D}_s(\cdot)$	Downsampling operator by factor s used inside degradation.	—
$\mathcal{S}_r(\cdot)$	Deterministic antialiased downsampling operator by factor r used to create intermediate supervision targets.	—
$\mathbf{X}_{256}$	Intermediate ground-truth image at 256×256, constructed from $\mathbf{X}_{\mathrm{HR}}$ via $\mathcal{S}_2(\cdot)$.	$256 \times 256 \times C$
$\mathbf{X}_{128}$	Intermediate ground-truth image at 128×128, constructed from $\mathbf{X}_{\mathrm{HR}}$ via $\mathcal{S}_4(\cdot)$.	$128 \times 128 \times C$
$\widehat{\mathbf{X}}_2$	Predicted SR output at $\times 2$ w.r.t. LR input (explicit supervised output).	$128 \times 128 \times C$
$\widehat{\mathbf{X}}_4$	Predicted SR output at $\times 4$ w.r.t. LR input (explicit supervised output).	$256 \times 256 \times C$
$\widehat{\mathbf{X}}_8$	Predicted SR output at $\times 8$ w.r.t. LR input (final SR output).	$512 \times 512 \times C$
$f_{\boldsymbol{\theta}}(\cdot)$	MR-TRAE mapping producing $(\widehat{\mathbf{X}}_2, \widehat{\mathbf{X}}_4, \widehat{\mathbf{X}}_8)$ from $\mathbf{Y}_{\mathrm{LR}}$ with parameters $\boldsymbol{\theta}$.	—

5.3 MR-TRAE architecture

This section describes the architecture of the proposed MR-TRAE model. The design is composed of three main components: (i) a pretrained and frozen Master AE, (ii) a trainable Follower AE, and (iii) three sequential $\times 2$ SR CNN backbones (CNN1–CNN3).

The most important architectural characteristic of MR-TRAE is that it produces *three explicit super-resolved outputs* $\widehat{\mathbf{X}}_2$, $\widehat{\mathbf{X}}_4$, and $\widehat{\mathbf{X}}_8$, each supervised at its corresponding resolution.

5.3.1 High-level components and forward dataflow

Model inputs and outputs. Given an LR input image $\mathbf{Y}_{\mathrm{LR}} \in \mathbb{R}^{64\times 64\times C}$, the MR-TRAE model produces three super-resolved outputs $\widehat{\mathbf{X}}_2 \in \mathbb{R}^{128\times 128\times C}$, $\widehat{\mathbf{X}}_4 \in \mathbb{R}^{256\times 256\times C}$, and $\widehat{\mathbf{X}}_8 \in \mathbb{R}^{512\times 512\times C}$. These outputs correspond to scale factors $\times 2$, $\times 4$, and $\times 8$ with respect to the LR input, and are *all* treated as valid SR images rather than intermediate tensors.

Follower AE as the only LR entry point. Unlike many SR pipelines where the first SR backbone consumes the raw LR observation, in MR-TRAE the *only* direct input LR image is the one provided to the Follower AE.

Let $F_{\boldsymbol{\theta}_F}(\cdot)$ denote the Follower AE, parameterized by $\boldsymbol{\theta}_F$. It maps the LR input to a reconstructed LR image and an internal latent representation:

$$(\widehat{\mathbf{Y}}_F, \mathbf{Z}_F) = F_{\boldsymbol{\theta}_F}(\mathbf{Y}_{\mathrm{LR}}), \tag{5.10}$$

where $\widehat{\mathbf{Y}}_F \in \mathbb{R}^{64\times 64\times C}$ is the Follower reconstruction and $\mathbf{Z}_F$ denotes the Follower latent tensor (its precise dimensionality is specified in Section 5.3.3). Feeding the SR backbones with $\widehat{\mathbf{Y}}_F$ instead of $\mathbf{Y}_{\mathrm{LR}}$ is motivated by the fact that the Follower AE can suppress part of the acquisition noise and produce a more "SR-friendly" LR representation, while still being trained end-to-end with the SR objectives.

Sequential $\times 2$ SR backbones with three supervised outputs. Let $\mathcal{U}_{\boldsymbol{\theta}_k}(\cdot)$ denote the $\times 2$ SR backbone used at stage $k \in \{1, 2, 3\}$ with parameters $\boldsymbol{\theta}_k$. The three SR modules are applied sequentially:

$$\widehat{\mathbf{X}}_2 = \mathcal{U}_{\boldsymbol{\theta}_1}\left(\widehat{\mathbf{Y}}_F\right), \tag{5.11}$$

$$\widehat{\mathbf{X}}_4 = \mathcal{U}_{\boldsymbol{\theta}_2}\left(\widehat{\mathbf{X}}_2\right), \tag{5.12}$$

$$\widehat{\mathbf{X}}_8 = \mathcal{U}_{\boldsymbol{\theta}_3}\left(\widehat{\mathbf{X}}_4\right). \tag{5.13}$$

Each module performs an upscaling by a factor of 2 (implemented via a learned feature refinement followed by a sub-pixel upsampling operation, detailed in Section 5.3.4). Crucially, the outputs $\widehat{\mathbf{X}}_2$ and $\widehat{\mathbf{X}}_4$ are not simply internal activations used only to reach $\widehat{\mathbf{X}}_8$. Instead, they are explicitly supervised during training (Section 5.4) using resolution-matched ground truths $\mathbf{X}_{128}$ and $\mathbf{X}_{256}$, and they can be used independently at inference time.

Master AE as a frozen latent and perceptual reference. Let $M_{\boldsymbol{\theta}_M}(\cdot)$ denote the Master AE, pretrained on HR images in Stage I and then frozen in Stage II (hence $\boldsymbol{\theta}_M$ is fixed during MR-TRAE training).

During the main training stage, the Master AE plays two roles:

- Latent guidance: the Master encoder extracts a reference latent representation from the paired HR ground truth $\mathbf{X}_{\mathrm{HR}}$, which is used to guide the Follower latent $\mathbf{Z}_F$ (formalized in Section 5.4).
- Perceptual feature extraction: the Master encoder provides multi-level feature maps for perceptual losses at the three output resolutions (also defined in Section 5.4).

Using a frozen, domain-trained AE as the perceptual reference preserves determinism and avoids the training instabilities often introduced by adversarial discriminators.

End-to-end forward summary. Combining (5.10)–(5.13), the complete forward mapping of MR-TRAE can be summarized as

$$\begin{aligned}(\widehat{\mathbf{X}}_2, \widehat{\mathbf{X}}_4, \widehat{\mathbf{X}}_8) &= \Big(\mathcal{U}_{\boldsymbol{\theta}_1}(\widehat{\mathbf{Y}}_F),\ \mathcal{U}_{\boldsymbol{\theta}_2}(\mathcal{U}_{\boldsymbol{\theta}_1}(\widehat{\mathbf{Y}}_F)),\ \mathcal{U}_{\boldsymbol{\theta}_3}(\mathcal{U}_{\boldsymbol{\theta}_2}(\mathcal{U}_{\boldsymbol{\theta}_1}(\widehat{\mathbf{Y}}_F)))\Big), \\ (\widehat{\mathbf{Y}}_F, \mathbf{Z}_F) &= F_{\boldsymbol{\theta}_F}(\mathbf{Y}_{\mathrm{LR}}).\end{aligned} \tag{5.14}$$

Figure 5.1 shows a schematic of MR-TRAE model.

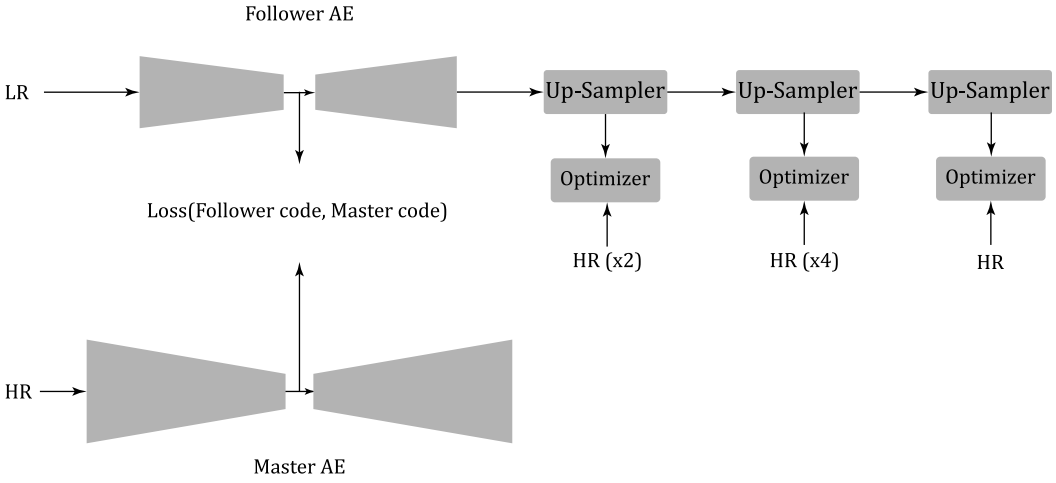

Figure 5.1. Overall architecture of the proposed MR-TRAE model. The LR input $\mathbf{Y}_{\mathrm{LR}}$ is first processed by the Follower AE to produce a reconstructed LR image $\widehat{\mathbf{Y}}_F$ and a latent representation $\mathbf{Z}_F$. The reconstructed LR image is then passed through three sequential $\times 2$ SR backbones (CNN1-CNN3), producing three explicitly supervised SR outputs $\widehat{\mathbf{X}}_2$, $\widehat{\mathbf{X}}_4$, and $\widehat{\mathbf{X}}_8$. The pretrained Master AE is frozen during Stage II and is used to (i) extract a reference latent from the paired HR target $\mathbf{X}_{\mathrm{HR}}$ for latent guidance, and (ii) compute perceptual features for output/target paired.

Residual nature of Master and Follower AEs. Both the Master AE and the Follower AE are residual architectures: they are composed of residual blocks (e.g., Residual Channel Attention Block (RCAB)-style units) and residual groups, and they also include long skip/fusion connections between encoder-side and decoder-side feature streams. These residual pathways improve gradient flow and help preserve anatomical structures during reconstruction. The detailed layer-by-layer configurations of the Master AE and Follower AE are provided in Sections 5.3.2 and 5.3.3, respectively.

5.3.2 Master AE

Role of the Master AE. The Master AE is trained in Stage I using only HR images $\mathbf{X}_{\text{HR}} \in \mathbb{R}^{512\times512\times C}$, and it is *frozen* during Stage II. Its purpose in MR-TRAE is twofold:

1. Latent reference for guidance. During Stage II, the Master encoder extracts an HR-oriented latent tensor from the paired ground truth $\mathbf{X}_{\text{HR}}$, denoted by $\mathbf{Z}_M$. This latent is used to guide the Follower latent $\mathbf{Z}_F$ via a latent alignment loss (formalized in Section 5.4).

2. Perceptual feature extractor. The Master encoder provides multi-level feature maps (at different depths) that define a *domain-specific perceptual space*. These features are used to compute perceptual losses between each SR output $(\widehat{\mathbf{X}}_2, \widehat{\mathbf{X}}_4, \widehat{\mathbf{X}}_8)$ and its corresponding ground truth $(\mathbf{X}_{128}, \mathbf{X}_{256}, \mathbf{X}_{\text{HR}})$.

Freezing the Master AE in Stage II ensures that the reference representation does not drift during optimization, yielding stable and reproducible training behavior compared to adversarial perceptual guidance.

Residual and skip-connected design. Architecturally, the Master is a residual AE. It relies on three types of skip/residual pathways:

- Local residual learning: the core of the encoder and decoder consists of stacked RCABs, where each block learns a residual correction to its input.
- Residual groups: multiple RCABs are organized into higher-level residual groups (a group-level skip connection around a stack of RCABs), improving gradient flow and enabling deeper feature extraction.
- Long skip fusions: encoder-side feature tensors are fused into the decoder via skip connections (e.g., concatenation followed by 1×1 fusion), improving structural preservation and detail recovery.

This residual structure is well suited for medical image reconstruction, because it facilitates learning high-frequency residuals while preserving low-frequency anatomical content.

Encoder–decoder mapping and latent tensor. Let the Master encoder be denoted by $E_M(\cdot)$ and the Master decoder by $D_M(\cdot)$. Given an HR input $\mathbf{X}_{\text{HR}}$, the

Master AE produces the reconstruction $\widehat{\mathbf{X}_{\mathrm{HR}\,M}}$ as

$$\mathbf{Z}_M = E_M(\mathbf{X}_{\mathrm{HR}}), \qquad \widehat{\mathbf{X}_{\mathrm{HR}\,M}} = D_M(\mathbf{Z}_M). \tag{5.15}$$

In Stage I, the Master AE is trained to minimize an HR reconstruction loss (defined later in Section 5.4), learning an HR-aware latent space encoded by $\mathbf{Z}_M$.

Multi-level feature taps for perceptual supervision. In addition to the latent tensor $\mathbf{Z}_M$, the Master encoder exposes intermediate feature tensors at multiple depths. We denote the set of selected feature maps by

$$\Phi_M(\mathbf{X}) = \left\{\phi_M^{(l)}(\mathbf{X})\right\}_{l\in\mathcal{L}}, \tag{5.16}$$

where $\mathcal{L}$ indexes the chosen layers/stages in the encoder (e.g., early, middle, and late encoder blocks). These features are used to define perceptual losses at the three output resolutions. Specifically, for each SR output $\widehat{\mathbf{X}}_2$, $\widehat{\mathbf{X}}_4$, and $\widehat{\mathbf{X}}_8$, we compare Master features of the prediction with Master features of the corresponding resolution-matched ground truth: $\mathbf{X}_{128}$, $\mathbf{X}_{256}$, and $\mathbf{X}_{\mathrm{HR}}$, respectively. Importantly, since the Master AE is frozen in Stage II, gradients from perceptual losses flow only to the trainable components (Follower AE and CNN1–CNN3), not to the Master parameters.

Compatibility across spatial resolutions. The Master AE is pretrained on $\mathbf{X}_{\mathrm{HR}} \in \mathbb{R}^{512\times512\times C}$, but during Stage II it is also applied to images at 128×128 and 256×256 to compute perceptual features. This is feasible because the Master encoder and decoder are fully convolutional, and its downsampling/upsampling operations preserve consistent feature semantics across input sizes, as long as spatial dimensions remain compatible with the network strides. Consequently, $\Phi_M(\cdot)$ can be evaluated on $\mathbf{X}_{128}$, $\mathbf{X}_{256}$, $\mathbf{X}_{\mathrm{HR}}$ as well as on $\widehat{\mathbf{X}}_2$, $\widehat{\mathbf{X}}_4$, $\widehat{\mathbf{X}}_8$.

Table 5.2 gives the detailed architecture for the Master AE.

In Section 5.7, we will also report the final hyperparameters used to train the Master AE in Stage I.

5.3.3 Follower AE

Role of the Follower AE in MR-TRAE. The Follower AE is the *only* component that directly receives the LR input $\mathbf{Y}_{\mathrm{LR}} \in \mathbb{R}^{64\times64\times C}$. Its function is twofold:

1. LR reconstruction (denoising/regularization): it reconstructs an LR estimate $\widehat{\mathbf{Y}}_F \in \mathbb{R}^{64\times64\times C}$ from $\mathbf{Y}_{\mathrm{LR}}$. This reconstruction serves as a stabilized, SR-friendly input to CNN1 (rather than feeding CNN1 with the raw $\mathbf{Y}_{\mathrm{LR}}$).
2. Latent representation aligned with HR knowledge: it produces a latent tensor $\mathbf{Z}_F$ that is encouraged (via a latent alignment loss) to match a reference latent $\mathbf{Z}_M$ extracted by the frozen Master AE from the paired HR target $\mathbf{X}_{\mathrm{HR}}$.

This design injects HR-oriented structural information into the LR processing stream while keeping the learning deterministic and paired.

Table 5.2. Layer-wise specification of the Master AE.

Stage	Module	Channels	Output size
S0	Stem conv (3×3)	$C \to 64$	512×512
S1	Residual group (stack of RCABs)	64	512×512
S2	Downsample + residual group	128	256×256
S3	Downsample + residual group	256	128×128
Latent	Latent feature tensor $\mathbf{Z}_M$	256	128×128
D3	Upsample + residual group + fusion	256	256×256
D2	Upsample + residual group + fusion	128	512×512
D1	Residual group + head conv	$64 \to C$	512×512

Residual and skip-connected design. Similar to the Master AE, the Follower AE is a residual AE. It includes:

- Local residual learning via RCABs: the main body comprises stacked RCABs, enabling efficient learning of residual corrections and stable optimization.
- Residual groups: RCABs are arranged into residual groups, improving representational power while maintaining favorable gradient flow.
- Long skip fusions: encoder-side features are fused into the decoder via skip connections (e.g., concatenation with 1×1 fusion), which helps preserve anatomical structures and fine details.

Overall, the Follower AE is not a plain encoder–decoder; it is a residual reconstruction network designed to be robust to degradations in $\mathbf{Y}_{\text{LR}}$.

Encoder–decoder mapping and latent tensor. Let the Follower encoder be denoted by $E_F(\cdot)$ and the Follower decoder by $D_F(\cdot)$. For an LR input $\mathbf{Y}_{\text{LR}}$, the Follower AE produces

$$\begin{aligned} \mathbf{Z}_F &= E_F(\mathbf{Y}_{\text{LR}}), \\ \widehat{\mathbf{Y}}_F &= D_F(\mathbf{Z}_F), \end{aligned} \tag{5.17}$$

where $\widehat{\mathbf{Y}}_F \in \mathbb{R}^{64\times64\times C}$ is the reconstructed LR image and $\mathbf{Z}_F$ is the latent tensor.

A key architectural choice in the proposed Follower AE is that $\mathbf{Z}_F$ is a *tensor (feature map)* rather than a single vector embedding. That is, $\mathbf{Z}_F$ preserves spatial organization and can be written as

$$\mathbf{Z}_F \in \mathbb{R}^{h\times w\times C_z}, \tag{5.18}$$

where C_z is the number of latent channels (reported explicitly in Table 5.3 and in Section 5.7). In our implementation, the LR spatial resolution is $h \times w = 64 \times 64$, and C_z is selected to match the Master latent channel capacity (e.g., $C_z = 256$), enabling a meaningful tensor-to-tensor latent alignment.

Coupling with the frozen Master AE (latent guidance). During Stage II, the paired HR target $\mathbf{X}_{\text{HR}}$ is fed into the frozen Master encoder to obtain the reference latent

$$\mathbf{Z}_M = E_M(\mathbf{X}_{\text{HR}}), \tag{5.19}$$

which guides the Follower latent $\mathbf{Z}_F$ through a latent alignment loss (defined formally in Section 5.4). Intuitively, this encourages the Follower AE to organize LR information in a latent space that is compatible with the HR-trained Master representation, thereby injecting HR structure into the LR-to-SR pipeline without adversarial training.

Interface to the SR backbones. The reconstructed LR image $\widehat{\mathbf{Y}}_F$ (not $\mathbf{Y}_{\text{LR}}$) is the input to the first SR backbone:

$$\widehat{\mathbf{X}}_2 = \mathcal{U}_{\boldsymbol{\theta}_1}(\widehat{\mathbf{Y}}_F), \tag{5.20}$$

followed by the sequential refinement to obtain $\widehat{\mathbf{X}}_4$ and $\widehat{\mathbf{X}}_8$ as given in (5.12)–(5.13). This choice ensures that the SR backbones operate on an LR estimate that has already been regularized by the residual auto-encoding process.

Table 5.3 show the detailed architecture for the Follower AE.

Table 5.3. Layer-wise specification of the Follower AE.

Stage	Module	Channels	Output size
S0	Stem conv (3×3)	$C \to 64$	64×64
S1	Residual group (stack of RCABs)	64	64×64
Latent	Channel expansion to latent $\mathbf{Z}_F$	$64 \to C_z$	64×64
D1	Residual group + fusion(s)	64	64×64
Head	Output conv (3×3)	$64 \to C$	64×64

All final implementation settings will be summarized in Section 5.7.

5.3.4 SR CNN backbone (CNN1–CNN3)

Role in the MR-TRAE pipeline. The SR component of MR-TRAE consists of three sequential $\times 2$ CNN backbones, denoted by CNN1–CNN3. These networks progressively upscale the reconstructed LR output of the Follower AE:

$$\widehat{\mathbf{X}}_2 = \mathcal{U}_{\boldsymbol{\theta}_1}(\widehat{\mathbf{Y}}_F), \qquad \widehat{\mathbf{X}}_4 = \mathcal{U}_{\boldsymbol{\theta}_2}(\widehat{\mathbf{X}}_2), \qquad \widehat{\mathbf{X}}_8 = \mathcal{U}_{\boldsymbol{\theta}_3}(\widehat{\mathbf{X}}_4), \tag{5.21}$$

where each $\mathcal{U}_{\boldsymbol{\theta}_k}(\cdot)$ performs upscaling by a factor of 2. Importantly, the outputs $\widehat{\mathbf{X}}_2$ and $\widehat{\mathbf{X}}_4$ are *explicitly supervised* (Sections 5.2 and 5.4) and are not merely intermediate tensors used only to obtain $\widehat{\mathbf{X}}_8$.

Backbone design: residual refinement around bicubic upsampling. Each $\times 2$ backbone follows a common and SR-effective design: (i) a *non-learned* bicubic upsampling path provides a strong low-frequency baseline, and (ii) a *learned* residual path predicts high-frequency corrections that are added to the baseline. Concretely, for an input tensor $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, define the bicubic baseline

$$\mathbf{X}_{\text{base}} = \text{Bicubic}_{\times 2}(\mathbf{X}) \in \mathbb{R}^{2H\times 2W\times C}. \tag{5.22}$$

The backbone then learns a residual mapping $\mathcal{R}_{\boldsymbol{\theta}}(\cdot)$ such that

$$\mathcal{U}_{\boldsymbol{\theta}}(\mathbf{X}) = \mathbf{X}_{\text{base}} + \mathcal{R}_{\boldsymbol{\theta}}(\mathbf{X}). \tag{5.23}$$

This global residual formulation is advantageous in medical SR because it preserves the coarse anatomical structure via $\mathbf{X}_{\text{base}}$ while allowing the network to focus on learning fine-detail corrections.

Internal structure: head, residual body, and pixel-shuffle tail. Let $\mathcal{U}_{\boldsymbol{\theta}}$ denote one $\times 2$ backbone. Its internal structure consists of:

- Head convolution: a 3×3 convolution projects the C-channel input into a feature space with F channels.
- Residual body (no normalization): a stack of N residual blocks refines features at the original resolution. Each residual block has two 3×3 convolutions with a pointwise nonlinearity (LeakyReLU) and an internal skip connection. No batch normalization is used, consistent with common SR practice to preserve image statistics and avoid over-smoothing.
- Tail upsampling: a 3×3 convolution expands channels to $4F$, followed by a Depth-to-Space (pixel-shuffle) operator with block size 2 to upscale spatial resolution by 2. A final 3×3 convolution maps features back to C output channels.

Residual learning at multiple levels. The backbone is residual in three complementary ways:

1. Residual blocks: each block learns a scaled residual correction that is added to its input feature tensor.
2. Body skip: the refined features are added to the head features (a long skip across the stack of residual blocks), which further stabilizes optimization.
3. Global image skip: the network output is added to the bicubic baseline as in (5.23).

These residual pathways improve gradient flow and encourage the model to focus on predicting missing high-frequency components.

Default configuration used in this thesis. Each $\times 2$ backbone uses $F = 64$ feature channels and $N = 8$ residual blocks. The activation function inside the backbone is LeakyReLU with negative slope 0.1, and the residual blocks employ a residual scaling factor (e.g., 0.1) for stability. While the implementation is instantiated for grayscale inputs ($C = 1$), the architecture generalizes directly to arbitrary channel count C by setting the head input channels and tail output channels to C.

Table 5.4 shows the detailed layer-wise specification of the $\times 2$ SR backbone. Since CNN1–CNN3 share the same architecture but have independent parameter sets, one table is sufficient.

Table 5.4. Layer-wise specification of the $\times 2$ SR backbone used for CNN1-CNN3. Here F denotes the feature width and N denotes the number of residual blocks.

Stage	Module	Channels	Output size
Input	Input image	C	$H \times W$
Base	Bicubic upsample $\text{Bicubic}_{\times 2}(\mathbf{X})$	C	$2H \times 2W$
Head	Conv(3×3, stride 1)	$C \to F$	$H \times W$
Body	$N \times$ ResBlockNoBN (two Conv3×3 + LeakyReLU + skip)	F	$H \times W$
Skip	Body skip addition (body output + head features)	F	$H \times W$
Up	Conv(3×3) + D2S(2) + LeakyReLU	$F \to F$	$2H \times 2W$
Out	Conv(3×3)	$F \to C$	$2H \times 2W$
Sum	Output + bicubic base	C	$2H \times 2W$

5.4 Objective functions and loss design

This section defines the complete training objective of MR-TRAE. The loss design follows the two-stage training strategy described earlier: (i) Stage I pretrains the Master AE on HR reconstruction, and (ii) Stage II trains the Follower AE and the three SR CNNs jointly while freezing the Master AE. All losses are defined for a general channel count C, although our implementation uses grayscale images ($C = 1$).

5.4.1 Basic fidelity loss in the pixel domain

We employ a robust pixel-domain fidelity loss. Unless otherwise stated, we use the ℓ_1 loss due to its favorable behavior for SR (less tendency to over-smooth compared to ℓ_2):

$$\mathcal{L}_{\ell_1}(\mathbf{A}, \mathbf{B}) = \|\mathbf{A} - \mathbf{B}\|_1 . \tag{5.24}$$

Here $\|\cdot\|_1$ denotes the sum of absolute differences over all spatial locations and channels. In Section 5.7, we also report the exact reduction convention (mean vs. sum); in practice we use the mean to keep magnitudes invariant to resolution.

5.4.2 Stage I: Master AE reconstruction loss

In Stage I, the Master AE is trained on HR images. Let the Master reconstruction be denoted by $\widehat{\mathbf{X}_{\mathrm{HR}\,M}}$. The Stage I objective is

$$\mathcal{L}^{(\mathrm{I})}_{\mathrm{Master}} = \mathbb{E}_{\mathbf{X}_{\mathrm{HR}}\sim p_{\mathrm{HR}}}\left[\mathcal{L}_{\ell_1}\left(\widehat{\mathbf{X}_{\mathrm{HR}\,M}}, \mathbf{X}_{\mathrm{HR}}\right)\right], \tag{5.25}$$

where $\widehat{\mathbf{X}_{\mathrm{HR}\,M}}$ is obtained from the Master encoder–decoder as in (5.15). After convergence, the Master AE parameters are frozen and used as a fixed reference in Stage II.

5.4.3 Stage II: Follower AE losses (LR reconstruction and latent alignment)

During Stage II, the Follower AE processes the LR input $\mathbf{Y}_{\mathrm{LR}}$ and outputs the reconstructed LR image $\widehat{\mathbf{Y}}_F$ and latent tensor $\mathbf{Z}_F$ (see (5.17)).

We define two losses for the Follower AE.

(1) LR reconstruction loss. The Follower reconstruction loss encourages faithful reconstruction of the LR observation:

$$\mathcal{L}^{F}_{\mathrm{rec}} = \mathbb{E}_{(\mathbf{Y}_{\mathrm{LR}},\mathbf{X}_{\mathrm{HR}})\sim\mathcal{D}}\left[\mathcal{L}_{\ell_1}\left(\widehat{\mathbf{Y}}_F, \mathbf{Y}_{\mathrm{LR}}\right)\right]. \tag{5.26}$$

(2) Latent alignment loss (Master-guided). Let the frozen Master encoder extract the reference latent tensor from the paired HR target:

$$\mathbf{Z}_M = E_M(\mathbf{X}_{\mathrm{HR}}). \tag{5.27}$$

The latent alignment loss encourages the Follower latent $\mathbf{Z}_F$ to match $\mathbf{Z}_M$:

$$\mathcal{L}_{\mathrm{lat}} = \mathbb{E}_{(\mathbf{Y}_{\mathrm{LR}},\mathbf{X}_{\mathrm{HR}})\sim\mathcal{D}}\left[\|\mathbf{Z}_F - \mathbf{Z}_M\|_1\right]. \tag{5.28}$$

Because $\mathbf{Z}_M$ is produced by a frozen network trained on HR images, (5.28) injects HR-consistent structural information into the Follower representation while preserving deterministic optimization.

5.4.4 Multi-resolution SR losses for CNN1–CNN3

The SR backbones produce the outputs $\widehat{\mathbf{X}}_2$, $\widehat{\mathbf{X}}_4$, and $\widehat{\mathbf{X}}_8$. Each output is supervised using (i) a pixel-domain loss, and (ii) a perceptual loss computed using features from the frozen Master encoder.

Pixel-domain multi-resolution SR losses

As defined in Section 5.2.3, the intermediate ground truths are $\mathbf{X}_{128} = \mathcal{S}_4(\mathbf{X}_{\mathrm{HR}})$ and $\mathbf{X}_{256} = \mathcal{S}_2(\mathbf{X}_{\mathrm{HR}})$. The pixel-domain SR losses are:

$$\mathcal{L}^{(2)}_{\mathrm{pix}} = \mathbb{E}_{(\mathbf{Y}_{\mathrm{LR}},\mathbf{X}_{\mathrm{HR}})\sim\mathcal{D}}\left[\mathcal{L}_{\ell_1}\left(\widehat{\mathbf{X}}_2, \mathbf{X}_{128}\right)\right], \tag{5.29}$$

$$\mathcal{L}^{(4)}_{\mathrm{pix}} = \mathbb{E}_{(\mathbf{Y}_{\mathrm{LR}},\mathbf{X}_{\mathrm{HR}})\sim\mathcal{D}}\left[\mathcal{L}_{\ell_1}\left(\widehat{\mathbf{X}}_4, \mathbf{X}_{256}\right)\right], \tag{5.30}$$

$$\mathcal{L}^{(8)}_{\mathrm{pix}} = \mathbb{E}_{(\mathbf{Y}_{\mathrm{LR}},\mathbf{X}_{\mathrm{HR}})\sim\mathcal{D}}\left[\mathcal{L}_{\ell_1}\left(\widehat{\mathbf{X}}_8, \mathbf{X}_{\mathrm{HR}}\right)\right]. \tag{5.31}$$

Perceptual losses using frozen Master features

We define a domain-specific perceptual loss using intermediate feature maps extracted from the frozen Master encoder. Let $\phi_M^{(l)}(\cdot)$ denote the feature map at layer/stage l of the Master encoder (as in (5.16)), and let $\mathcal{L}$ denote the index set of selected layers. For an image $\mathbf{X}$, the feature extractor returns $\{\phi_M^{(l)}(\mathbf{X})\}_{l\in\mathcal{L}}$.

The perceptual loss between a prediction $\widehat{\mathbf{X}}$ and a target $\mathbf{X}^\star$ is defined as

$$\mathcal{L}_{\text{per}}(\widehat{\mathbf{X}}, \mathbf{X}^\star) = \sum_{l\in\mathcal{L}} \alpha_l \left\| \phi_M^{(l)}(\widehat{\mathbf{X}}) - \phi_M^{(l)}(\mathbf{X}^\star) \right\|_1, \tag{5.32}$$

where $\alpha_l \geq 0$ are scalar weights controlling the contribution of each feature level. Since the Master network is frozen, the terms $\phi_M^{(l)}(\cdot)$ provide a fixed perceptual embedding, and gradients flow only through the trainable networks that produce $\widehat{\mathbf{X}}$.

We then define the three multi-resolution perceptual losses:

$$\mathcal{L}_{\text{per}}^{(2)} = \mathbb{E}_{(\mathbf{Y}_{\text{LR}}, \mathbf{X}_{\text{HR}})\sim\mathcal{D}} \left[\mathcal{L}_{\text{per}}(\widehat{\mathbf{X}}_2, \mathbf{X}_{128}) \right], \tag{5.33}$$

$$\mathcal{L}_{\text{per}}^{(4)} = \mathbb{E}_{(\mathbf{Y}_{\text{LR}}, \mathbf{X}_{\text{HR}})\sim\mathcal{D}} \left[\mathcal{L}_{\text{per}}(\widehat{\mathbf{X}}_4, \mathbf{X}_{256}) \right], \tag{5.34}$$

$$\mathcal{L}_{\text{per}}^{(8)} = \mathbb{E}_{(\mathbf{Y}_{\text{LR}}, \mathbf{X}_{\text{HR}})\sim\mathcal{D}} \left[\mathcal{L}_{\text{per}}(\widehat{\mathbf{X}}_8, \mathbf{X}_{\text{HR}}) \right]. \tag{5.35}$$

5.4.5 Full Stage II MR-TRAE objective

The complete Stage II objective is a weighted combination of the Follower losses and the multi-resolution SR losses:

$$\mathcal{L}_{\text{MR-TRAE}}^{(\text{II})} = \lambda_F \, \mathcal{L}_{\text{rec}}^{F} + \lambda_{\text{lat}} \, \mathcal{L}_{\text{lat}} + \sum_{k\in\{2,4,8\}} \left(\lambda_{\text{pix}}^{(k)} \, \mathcal{L}_{\text{pix}}^{(k)} + \lambda_{\text{per}}^{(k)} \, \mathcal{L}_{\text{per}}^{(k)} \right). \tag{5.36}$$

The coefficients λ_F, λ_{lat}, $\lambda_{\text{pix}}^{(k)}$, and $\lambda_{\text{per}}^{(k)}$ control the trade-off between (i) LR reconstruction, (ii) HR-informed latent guidance, and (iii) fidelity/perceptual quality at each output resolution. In Section 5.7, we provide recommended default values for these weights and practical guidelines for balancing the gradients across the three output resolutions.

Table 5.5 summarizes each loss term and its coefficient.

The perceptual losses in (5.32)–(5.35) rely on a *frozen* Master encoder trained on domain data. This avoids adversarial training and provides a deterministic perceptual reference, which is particularly important in medical imaging where hallucinated structures are undesirable.

5.5 Training strategy

This section describes the two-stage training strategy used for MR-TRAE. Stage I pretrains the Master AE on HR images. Stage II freezes the Master AE and jointly trains the Follower AE and the three sequential $\times 2$ SR CNN backbones (CNN1–CNN3). Consistent with Section 5.4, all pixel-domain fidelity terms in this section use the ℓ_1 loss.

Table 5.5. Summary of loss terms used in MR-TRAE training (Stage II) and their weights.

Loss term	Definition	Weight
$\mathcal{L}_{\text{rec}}^{F}$	Follower LR reconstruction (5.26)	λ_F
$\mathcal{L}_{\text{lat}}$	Latent alignment (5.28)	λ_{lat}
$\mathcal{L}_{\text{pix}}^{(2)}$	Pixel SR loss at 128^2 (5.29)	$\lambda_{\text{pix}}^{(2)}$
$\mathcal{L}_{\text{per}}^{(2)}$	Perceptual SR loss at 128^2 (5.33)	$\lambda_{\text{per}}^{(2)}$
$\mathcal{L}_{\text{pix}}^{(4)}$	Pixel SR loss at 256^2 (5.30)	$\lambda_{\text{pix}}^{(4)}$
$\mathcal{L}_{\text{per}}^{(4)}$	Perceptual SR loss at 256^2 (5.34)	$\lambda_{\text{per}}^{(4)}$
$\mathcal{L}_{\text{pix}}^{(8)}$	Pixel SR loss at 512^2 (5.31)	$\lambda_{\text{pix}}^{(8)}$
$\mathcal{L}_{\text{per}}^{(8)}$	Perceptual SR loss at 512^2 (5.35)	$\lambda_{\text{per}}^{(8)}$

5.5.1 Stage I: Master AE pretraining

Dataset. Stage I uses only HR images $\mathbf{X}_{\text{HR}} \in \mathbb{R}^{512\times512\times C}$ to learn a domain-specific latent space and feature hierarchy. If paired LR–HR samples are available (as in (5.3)), Stage I simply uses the HR component $\{\mathbf{X}_{\text{HR}}^{(i)}\}_{i=1}^{N}$. We construct a standard train/validation split to enable early stopping and checkpoint selection. Although our experiments are conducted on grayscale lung CT (thus $C = 1$), the training procedure is valid for general C.

Objective. The Master AE is trained as an HR reconstruction model. Let $\widehat{\mathbf{X}_{\text{HR}\,M}}$ denote the Master reconstruction of $\mathbf{X}_{\text{HR}}$. The Stage I objective is

$$\mathcal{L}_{\text{Master}}^{(\text{I})} = \mathbb{E}_{\mathbf{X}_{\text{HR}}\sim p_{\text{HR}}}\left[\mathcal{L}_{\ell_1}\left(\widehat{\mathbf{X}_{\text{HR}\,M}}, \mathbf{X}_{\text{HR}}\right)\right], \tag{5.37}$$

where $\mathcal{L}_{\ell_1}(\cdot,\cdot)$ is defined in Section 5.4.1. In practice, training is performed on randomly cropped HR patches (e.g., 512×512 or smaller crops if memory-constrained) with standard data augmentation (random flips and 90° rotations) to improve generalization.

Stopping criteria. We monitor the validation reconstruction loss $\mathbb{E}[\mathcal{L}_{\ell_1}(\widehat{\mathbf{X}_{\text{HR}\,M}}, \mathbf{X}_{\text{HR}})]$ and optionally report validation PSNR/SSIM for interpretability. Training is stopped when one of the following conditions is met:

- a maximum number of epochs is reached (fixed training budget), or
- early stopping is triggered (no improvement in validation loss for a specified patience).

Early stopping is recommended to prevent overfitting and to ensure that the frozen Master features used in Stage II remain representative of the target distribution.

Model checkpointing. We save:

- **Best checkpoint:** the Master AE parameters with the lowest validation reconstruction loss.

- **Periodic checkpoints (optional):** snapshots every K epochs for debugging/ablation.

The *best* checkpoint is used as the frozen Master AE in Stage II. In Section 5.7, we provide the optimizer choice and learning-rate schedule used for Stage I.

5.5.2 Stage II: frozen Master + joint training of Follower AE and CNN1-CNN3

Unified end-to-end training with paired data. Stage II trains *one* unified model that outputs $\widehat{\mathbf{X}}_2$, $\widehat{\mathbf{X}}_4$, and $\widehat{\mathbf{X}}_8$ simultaneously. Each mini-batch contains paired samples $(\mathbf{Y}_{\mathrm{LR}}, \mathbf{X}_{\mathrm{HR}})$. The forward pass follows the MR-TRAE dataflow:

$$(\widehat{\mathbf{Y}}_F, \mathbf{Z}_F) = F_{\boldsymbol{\theta}_F}(\mathbf{Y}_{\mathrm{LR}}), \qquad \widehat{\mathbf{X}}_2 = \mathcal{U}_{\boldsymbol{\theta}_1}(\widehat{\mathbf{Y}}_F), \qquad \widehat{\mathbf{X}}_4 = \mathcal{U}_{\boldsymbol{\theta}_2}(\widehat{\mathbf{X}}_2), \qquad \widehat{\mathbf{X}}_8 = \mathcal{U}_{\boldsymbol{\theta}_3}(\widehat{\mathbf{X}}_4). \tag{5.38}$$

In parallel, the frozen Master encoder processes the paired HR target to provide the reference latent and perceptual features:

$$\mathbf{Z}_M = E_M(\mathbf{X}_{\mathrm{HR}}), \qquad \Phi_M(\cdot) = \{\phi_M^{(l)}(\cdot)\}_{l \in \mathcal{L}}. \tag{5.39}$$

We also generate the intermediate supervision targets $\mathbf{X}_{128} = \mathcal{S}_4(\mathbf{X}_{\mathrm{HR}})$ and $\mathbf{X}_{256} = \mathcal{S}_2(\mathbf{X}_{\mathrm{HR}})$ (Section 5.2.3).

Stage II objective. Stage II minimizes the composite MR-TRAE loss defined in (5.36), where all pixel-domain fidelity terms use ℓ_1: (i) the Follower LR reconstruction loss $\mathcal{L}_{\mathrm{rec}}^{F}$, (ii) the latent alignment loss $\mathcal{L}_{\mathrm{lat}}$, and (iii) the multi-resolution SR pixel losses $\mathcal{L}_{\mathrm{pix}}^{(2)}$, $\mathcal{L}_{\mathrm{pix}}^{(4)}$, and $\mathcal{L}_{\mathrm{pix}}^{(8)}$. Perceptual losses are computed from frozen Master features as defined in Section 5.4.4.

Gradient flow and what is updated. A central design principle of MR-TRAE is that the Master AE is *frozen* in Stage II. Therefore:

- **Frozen parameters:** Master parameters $\boldsymbol{\theta}_M$ are not updated; gradients do not propagate into the Master network.
- **Trainable parameters:** the Follower AE parameters $\boldsymbol{\theta}_F$ and the SR backbone parameters $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3$ are updated jointly.

Operationally, the Master encoder acts as a deterministic function that maps images to a latent/perceptual embedding, while all optimization steps update only the networks that produce $\widehat{\mathbf{Y}}_F$ and $(\widehat{\mathbf{X}}_2, \widehat{\mathbf{X}}_4, \widehat{\mathbf{X}}_8)$.

Contrast with training per scale (TRAE-style). In the TRAE setting (Chapter 4), scale-specific training can be required when learning distinct SR models for different upscaling factors (e.g., $\times 2$, $\times 4$, $\times 8$), or when a single pipeline is trained separately for each scale. In MR-TRAE, the three scales are trained *simultaneously*: CNN1, CNN2, and CNN3 form a single sequential pipeline, and each stage is supervised at its own output resolution. As a result, Stage II consists of a single end-to-end training procedure rather than multiple independent scale-specific trainings.

5.5.3 Why MR-TRAE trains only once (vs. scale-specific runs)

Computational efficiency. Training a separate SR model per scale increases computational cost and tuning complexity. By training CNN1–CNN3 jointly in a single Stage II optimization, MR-TRAE amortizes the training process across three outputs:

- **One optimizer run, one schedule:** a single training loop learns all three scales.
- **Shared supervision signals:** gradients from the $\times 2$, $\times 4$, and $\times 8$ objectives jointly shape the representation learned by the earlier stages of the pipeline.

Improved cross-scale consistency. Scale-specific training may yield inconsistencies where a model optimized for $\times 2$ reconstructs structures differently than a model optimized for $\times 8$. In MR-TRAE, the sequential structure and multi-output supervision enforce that:

- $\widehat{\mathbf{X}}_2$ matches $\mathbf{X}_{128}$,
- $\widehat{\mathbf{X}}_4$ matches $\mathbf{X}_{256}$,
- $\widehat{\mathbf{X}}_8$ matches $\mathbf{X}_{\mathrm{HR}}$,

within *the same* end-to-end computation graph. These explicit constraints encourage anatomical and textural consistency across scales.

Multi-output supervision replaces "train per scale." The main reason MR-TRAE can be trained once is that it does not treat intermediate stages as purely auxiliary. Instead, each stage has its own explicit target and loss, so each CNN is directly optimized to produce a usable SR image at its resolution. Consequently, MR-TRAE yields three deployable outputs after a single training run, while maintaining stable perceptual guidance via the frozen Master AE.

5.6 Training algorithms and implementation details

This section provides reproducible training algorithms and implementation details for MR-TRAE. We present pseudocode for Stage I (Master AE pretraining) and Stage II (joint training of the Follower AE and CNN1–CNN3 with multi-resolution supervision). All fidelity terms in this chapter use the ℓ_1 loss, consistent with Section 5.4.

Algorithm 3 show Stage I where the Master is trained, and Algorithm 4

5.6.1 Optimization details

This subsection specifies recommended optimization choices for reproducible training. All values given below are the default settings adopted in our implementation unless otherwise noted.

Algorithm 3 Stage I: Pretraining the Master AE on HR images

Require: HR training set $\mathcal{H}_{\text{train}} = \{\mathbf{X}_{\text{HR}}^{(i)}\}_{i=1}^{N}$ and validation set $\mathcal{H}_{\text{val}} = \{\mathbf{X}_{\text{HR}}^{(j)}\}_{j=1}^{N_{\text{val}}}$
Require: Batch size B, maximum epochs E_M, early-stopping patience P
Require: Adam optimizer with learning rate schedule $\eta_M(t)$
Initialize Master parameters $\theta_M \leftarrow \{\theta_{E_M}, \theta_{D_M}\}$
$L_{\min} \leftarrow +\infty$ ▷ best validation loss so far
$p \leftarrow 0$ ▷ patience counter
for epoch $= 1$ to E_M **do**
  for each mini-batch $\{\mathbf{X}_{\text{HR}}^{(i)}\}_{i=1}^{B}$ sampled from $\mathcal{H}_{\text{train}}$ **do**
    Forward:
      $\mathbf{Z}_M^{(i)} \leftarrow E_M(\mathbf{X}_{\text{HR}}^{(i)}; \theta_{E_M})$
      $\widehat{\mathbf{X}_{\text{HR}}}_M^{(i)} \leftarrow D_M(\mathbf{Z}_M^{(i)}; \theta_{D_M})$
    Loss:
      $\mathcal{L}_M \leftarrow \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\ell_1}\left(\widehat{\mathbf{X}_{\text{HR}}}_M^{(i)}, \mathbf{X}_{\text{HR}}^{(i)}\right)$
    Update θ_M by one step of Adam on $\nabla_{\theta_M}\mathcal{L}_M$ using $\eta_M(t)$
  end for
  Validation (no parameter update):
  Compute $L_{\text{val}} \leftarrow \frac{1}{N_{\text{val}}}\sum_{j=1}^{N_{\text{val}}} \mathcal{L}_{\ell_1}\left(\widehat{\mathbf{X}_{\text{HR}}}_M^{(j)}, \mathbf{X}_{\text{HR}}^{(j)}\right)$ on $\mathcal{H}_{\text{val}}$
  if $L_{\text{val}} < L_{\min}$ **then**
    $L_{\min} \leftarrow L_{\text{val}}$; save best checkpoint $\theta_M^* \leftarrow \theta_M$; $p \leftarrow 0$
  else
    $p \leftarrow p + 1$
  end if
  if $p \geq P$ **then**
    break ▷ early stopping
  end if
end for
Output: pretrained Master parameters θ_M^* (used and frozen in Stage II)

Algorithm 4 Stage II: Joint MR-TRAE training with frozen Master AE and three supervised SR outputs

Require: Paired training set $\mathcal{D} = \{(\mathbf{Y}_{\mathrm{LR}}^{(i)}, \mathbf{X}_{\mathrm{HR}}^{(i)})\}_{i=1}^{N}$, batch size B, epochs E
Require: Frozen Master parameters $\theta_M^* = \{\theta_{E_M}^*, \theta_{D_M}^*\}$
Require: Adam optimizer with learning rate schedule $\eta(t)$
Require: Loss weights $\lambda_F, \lambda_{\mathrm{lat}}, \lambda_{\mathrm{pix}}^{(k)}, \lambda_{\mathrm{per}}^{(k)}$ for $k \in \{2,4,8\}$; feature weights $\{\alpha_l\}_{l\in\mathcal{L}}$
Initialize trainable parameters $\theta_F \leftarrow \{\theta_{E_F}, \theta_{D_F}\}$ and $\theta_1, \theta_2, \theta_3$ for CNN1–CNN3
Load Master networks (E_M, D_M) with θ_M^* and **freeze** all Master parameters
for epoch $= 1$ to E **do**
  for each mini-batch $\{(\mathbf{Y}_{\mathrm{LR}}^{(i)}, \mathbf{X}_{\mathrm{HR}}^{(i)})\}_{i=1}^{B}$ sampled from $\mathcal{D}$ **do**
    (Optional) LR synthesis:
      If $\mathbf{Y}_{\mathrm{LR}}$ is generated on-the-fly, set $\mathbf{Y}_{\mathrm{LR}}^{(i)} \leftarrow \mathcal{G}_s(\mathbf{X}_{\mathrm{HR}}^{(i)})$
    Intermediate target synthesis:
      $\mathbf{X}_{128}^{(i)} \leftarrow \mathcal{S}_4(\mathbf{X}_{\mathrm{HR}}^{(i)}), \quad \mathbf{X}_{256}^{(i)} \leftarrow \mathcal{S}_2(\mathbf{X}_{\mathrm{HR}}^{(i)})$
    Master references (no gradient):
      $\mathbf{Z}_M^{(i)} \leftarrow E_M(\mathbf{X}_{\mathrm{HR}}^{(i)}; \theta_{E_M}^*)$
      Extract features $\{\phi_M^{(l)}(\mathbf{X}_{128}^{(i)})\}_{l\in\mathcal{L}}$, $\{\phi_M^{(l)}(\mathbf{X}_{256}^{(i)})\}_{l\in\mathcal{L}}$, $\{\phi_M^{(l)}(\mathbf{X}_{\mathrm{HR}}^{(i)})\}_{l\in\mathcal{L}}$
    Follower forward:
      $\mathbf{Z}_F^{(i)} \leftarrow E_F(\mathbf{Y}_{\mathrm{LR}}^{(i)}; \theta_{E_F})$
      $\widehat{\mathbf{Y}}_F^{(i)} \leftarrow D_F(\mathbf{Z}_F^{(i)}; \theta_{D_F})$
    Sequential SR forward (CNN1–CNN3):
      $\widehat{\mathbf{X}}_2^{(i)} \leftarrow \mathcal{U}_{\theta_1}(\widehat{\mathbf{Y}}_F^{(i)})$
      $\widehat{\mathbf{X}}_4^{(i)} \leftarrow \mathcal{U}_{\theta_2}(\widehat{\mathbf{X}}_2^{(i)})$
      $\widehat{\mathbf{X}}_8^{(i)} \leftarrow \mathcal{U}_{\theta_3}(\widehat{\mathbf{X}}_4^{(i)})$
    Master features for predictions (no gradient through Master):
      Extract features $\{\phi_M^{(l)}(\widehat{\mathbf{X}}_2^{(i)})\}_{l\in\mathcal{L}}$, $\{\phi_M^{(l)}(\widehat{\mathbf{X}}_4^{(i)})\}_{l\in\mathcal{L}}$, $\{\phi_M^{(l)}(\widehat{\mathbf{X}}_8^{(i)})\}_{l\in\mathcal{L}}$
    Compute losses:
      $\mathcal{L}_{\mathrm{rec}}^{F} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\ell_1}\left(\widehat{\mathbf{Y}}_F^{(i)}, \mathbf{Y}_{\mathrm{LR}}^{(i)}\right)$
      $\mathcal{L}_{\mathrm{lat}} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \left\|\mathbf{Z}_F^{(i)} - \mathbf{Z}_M^{(i)}\right\|_1$
      $\mathcal{L}_{\mathrm{pix}}^{(2)} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\ell_1}\left(\widehat{\mathbf{X}}_2^{(i)}, \mathbf{X}_{128}^{(i)}\right)$
      $\mathcal{L}_{\mathrm{pix}}^{(4)} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\ell_1}\left(\widehat{\mathbf{X}}_4^{(i)}, \mathbf{X}_{256}^{(i)}\right)$
      $\mathcal{L}_{\mathrm{pix}}^{(8)} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\ell_1}\left(\widehat{\mathbf{X}}_8^{(i)}, \mathbf{X}_{\mathrm{HR}}^{(i)}\right)$
      $\mathcal{L}_{\mathrm{per}}^{(2)} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \sum_{l\in\mathcal{L}} \alpha_l \left\|\phi_M^{(l)}(\widehat{\mathbf{X}}_2^{(i)}) - \phi_M^{(l)}(\mathbf{X}_{128}^{(i)})\right\|_1$
      $\mathcal{L}_{\mathrm{per}}^{(4)} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \sum_{l\in\mathcal{L}} \alpha_l \left\|\phi_M^{(l)}(\widehat{\mathbf{X}}_4^{(i)}) - \phi_M^{(l)}(\mathbf{X}_{256}^{(i)})\right\|_1$
      $\mathcal{L}_{\mathrm{per}}^{(8)} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \sum_{l\in\mathcal{L}} \alpha_l \left\|\phi_M^{(l)}(\widehat{\mathbf{X}}_8^{(i)}) - \phi_M^{(l)}(\mathbf{X}_{\mathrm{HR}}^{(i)})\right\|_1$
    Total loss:
      $\mathcal{L} \leftarrow \lambda_F \mathcal{L}_{\mathrm{rec}}^{F} + \lambda_{\mathrm{lat}}\mathcal{L}_{\mathrm{lat}} + \sum_{k\in\{2,4,8\}} \left(\lambda_{\mathrm{pix}}^{(k)}\mathcal{L}_{\mathrm{pix}}^{(k)} + \lambda_{\mathrm{per}}^{(k)}\mathcal{L}_{\mathrm{per}}^{(k)}\right)$
    Update $(\theta_F, \theta_1, \theta_2, \theta_3)$ by one step of Adam on $\nabla_{(\theta_F,\theta_1,\theta_2,\theta_3)}\mathcal{L}$ using $\eta(t)$
  end for
end for
Output: trained parameters $\theta_F^*, \theta_1^*, \theta_2^*, \theta_3^*$

Optimizer. We use Adam for both stages with $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$. We set weight decay to 0 (or 10^{-6} if mild regularization is desired). This choice is standard for SR-style regression training and works well for both the AEs and the SR backbones.

Learning-rate schedule. For both stages, we use an initial learning rate

$$\eta_0 = 2 \times 10^{-4}, \tag{5.40}$$

with cosine annealing down to $\eta_{\min} = 10^{-6}$ over the full training budget. Cosine annealing provides smooth decay and typically stabilizes convergence for SR networks. If cosine annealing is not used, a step decay by a factor of 0.5 at $\{50\%, 75\%, 90\%\}$ of epochs is a reasonable alternative.

Loss weights. An important item is to balance the three supervised outputs so that no single resolution dominates training. The following settings were evaluated as good setting:

$$\lambda_F = 1.0, \qquad \lambda_{\text{lat}} = 0.1, \qquad \lambda_{\text{pix}}^{(2)} = \lambda_{\text{pix}}^{(4)} = \lambda_{\text{pix}}^{(8)} = 1.0, \tag{5.41}$$

and for perceptual terms,

$$\lambda_{\text{per}}^{(2)} = \lambda_{\text{per}}^{(4)} = \lambda_{\text{per}}^{(8)} = 0.05. \tag{5.42}$$

5.7 Implementation settings

We describe data preparation (including LR synthesis and intermediate ground truths), exact network configurations, and computational cost considerations. The implementation uses grayscale lung CT images ($C = 1$), while the mathematical formulation in previous sections remains valid for arbitrary C.

5.7.1 Data preparation

LR synthesis for 64×64 (fixed scale $s = 8$)

To align with the TRAE degradation setting while fixing the scale to $s = 8$, we model the LR observation as a degraded version of the HR target:

$$\mathbf{Y}_{\text{LR}} = \mathcal{G}_8(\mathbf{X}_{\text{HR}}) = \mathcal{D}_8\Big(\mathcal{N}(\mathcal{B}(\mathbf{X}_{\text{HR}}))\Big), \tag{5.43}$$

where $\mathcal{B}(\cdot)$ is a blur operator, $\mathcal{N}(\cdot)$ is a noise operator, and $\mathcal{D}_8(\cdot)$ downsamples by factor 8. This choice produces LR inputs that better reflect realistic acquisition artifacts than pure bicubic downsampling.

Exact degradation operators. In our implementation, we use the following default configuration:

- Blur $\mathcal{B}$: a 2D Gaussian blur with kernel size $k = 7$ and standard deviation $\sigma \sim \mathcal{U}(0.2, 1.8)$ sampled independently for each training patch.

- Noise $\mathcal{N}$: additive white Gaussian noise with standard deviation $\sigma_n \sim \mathcal{U}(0, 0.01)$ (assuming intensities normalized to $[0, 1]$); sampled independently per patch.
- Downsampling $\mathcal{D}_8$: bicubic downsampling by factor 8 with antialiasing enabled.

At validation/test time, we fix σ and σ_n to their mean values (or disable randomization) to reduce evaluation variance.

Normalization. All CT images are converted to a consistent dynamic range: (i) clip intensities to a fixed Hounsfield Units (HU) window, (ii) linearly map the clipped range to $[0, 1]$. The same normalization is applied before generating $\mathbf{Y}_{\text{LR}}$.

Intermediate ground truths at 128×128 and 256×256

The intermediate supervision targets are derived *deterministically* from $\mathbf{X}_{\text{HR}}$:

$$\begin{aligned} \mathbf{X}_{256} &= \mathcal{S}_2(\mathbf{X}_{\text{HR}}), \\ \mathbf{X}_{128} &= \mathcal{S}_4(\mathbf{X}_{\text{HR}}), \end{aligned} \tag{5.44}$$

where $\mathcal{S}_r$ denotes antialiased downsampling by factor r. Unlike $\mathcal{G}_8$ used to produce the LR input, $\mathcal{S}_r$ is not intended to simulate acquisition artifacts; rather, it produces the *ideal* targets at each intermediate resolution so that $\widehat{\mathbf{X}}_2$ and $\widehat{\mathbf{X}}_4$ correspond to meaningful SR outputs.

Operator choice. We use **bicubic downsampling with antialiasing enabled** for $\mathcal{S}_2$ and $\mathcal{S}_4$. This choice is: (i) deterministic, (ii) widely used as a high-quality resampling operator, and (iii) consistent across resolutions, ensuring that the intermediate targets represent true downscaled versions of the HR anatomy without additional blur/noise. Using antialiasing is important because it suppresses aliasing artifacts that would otherwise corrupt supervision at lower resolutions.

Why $\mathcal{S}_r$ can differ from $\mathcal{G}_8$. It is not necessary (and often not desirable) for $\mathcal{S}_r$ to match $\mathcal{G}_8$ exactly. $\mathcal{G}_8$ is used to model LR acquisition (potentially noisy/blurred), whereas $\mathcal{S}_r$ is used to define clean supervision targets for multi-resolution SR outputs.

This separation allows MR-TRAE to learn to invert realistic degradations at the LR input while being guided toward anatomically consistent targets at all three output scales.

Patching/cropping strategy and augmentation

Patch extraction. To ensure that the final output loss at 512×512 is directly supervised, we adopt an HR patch size of 512×512 during Stage II.

For each training iteration, we sample a random crop $\mathbf{X}_{\text{HR}}^{(i)} \in \mathbb{R}^{512 \times 512 \times C}$ from the

full HR image, and then generate:

$$\begin{aligned}\mathbf{Y}_{\mathrm{LR}}^{(i)} &= \mathcal{G}_8(\mathbf{X}_{\mathrm{HR}}^{(i)}),\\ \mathbf{X}_{128}^{(i)} &= \mathcal{S}_4(\mathbf{X}_{\mathrm{HR}}^{(i)}),\\ \mathbf{X}_{256}^{(i)} &= \mathcal{S}_2(\mathbf{X}_{\mathrm{HR}}^{(i)}).\end{aligned} \tag{5.45}$$

This ensures perfect spatial alignment among all supervision signals. If the HR images are already exactly 512×512, then "random crop" is just the whole image (no cropping).

5.7.2 Network configuration

This subsection presents the hyperparameters used in our MR-TRAE implementation. All networks are convolutional and operate on tensors with C channels; in our experiments, $C = 1$ (grayscale).

Master AE (Stage I)

The Master AE configuration is:

- base_channels = 64
- latent_channels = 256
- n_rcab = 6 (per residual group, as implemented)

The Master AE is pretrained on 512×512 HR inputs and is frozen during Stage II.

Follower AE (Stage II trainable)

During stage II, the Master AE is frozen. The Follower AE configuration is:

- base_channels = 64
- latent_channels = 256
- n_rcab = 6

The Follower processes $\mathbf{Y}_{\mathrm{LR}} \in \mathbb{R}^{64\times64\times C}$ and outputs the reconstructed LR $\widehat{\mathbf{Y}}_F \in \mathbb{R}^{64\times64\times C}$ together with latent $\mathbf{Z}_F$.

SR backbone (CNN1-CNN3)

Each SR backbone (CNN1, CNN2, CNN3) uses the same $\times 2$ architecture but has independent parameters:

- feat = 64 (feature width)
- n_blocks = 8 residual blocks (no batch normalization)
- Upsampling: pixel-shuffle (Depth-to-Space) with scale 2
- Global skip: bicubic upsampling skip connection added to the network prediction

This configuration is to have a trade-off between capacity, stability, and computational cost, and supports arbitrary spatial sizes.

5.7.3 Computational cost

This subsection presents parameter counts and Video Random Access Memory (VRAM) considerations. Parameter counts are reported for both grayscale ($C = 1$) and RGB ($C = 3$) to highlight channel-general scalability.

Trainable parameters

Master AE. For input size $512 \times 512 \times 1$, the Master AE has 32,964,401 trainable parameters. For input size $512 \times 512 \times 3$, it has 32,965,553 trainable parameters.

Follower AE. For input size $64 \times 64 \times 1$, the Follower AE has 21,054,353 trainable parameters. For input size $64 \times 64 \times 3$, it has 21,055,505 trainable parameters.

$\times 2$ SR backbone. For grayscale input ($C = 1$) with any spatial size, one $\times 2$ SR backbone has 739,777 trainable parameters. For RGB input ($C = 3$), it has 740,929 trainable parameters.

Total trainable parameters in Stage II. In Stage II, the Master AE is frozen; thus the trainable parameters are those of the Follower AE plus the three SR backbones. For grayscale ($C = 1$), the total number of trainable parameters in Stage II is:

$$\#\text{Params}_{\text{train}} = 21{,}054{,}353 + 3 \times 739{,}777 = 23{,}273{,}684. \tag{5.46}$$

For completeness, the total number of parameters present in memory during Stage II (including the frozen Master) is:

$$\#\text{Params}_{\text{total}} = 32{,}964{,}401 + 23{,}273{,}684 = 56{,}238{,}085, \tag{5.47}$$

where the Master parameters do not receive gradients.

VRAM considerations

The dominant VRAM cost in Stage II arises from storing activations for the 512×512 output path (CNN3) and from computing perceptual features through the frozen Master encoder at multiple resolutions. For training with 512×512 HR patches, we need at least one Graphics Processing Unit (GPU) with ≥ 24 GB VRAM. We use automatic mixed precision to reduce VRAM usage without degrading SR quality.

5.8 How MR-TRAE supports the SISR for medical imaging

MR-TRAE is designed to be (i) stable and deterministic during training, (ii) perceptually effective without relying on adversarial hallucination-prone mechanisms, and (iii) practically useful across multiple spatial resolutions through its three explicitly supervised outputs.

5.8.1 Stability and determinism

A central goal in medical SR is to obtain reliable reconstructions that do not depend on unstable training dynamics.

MR-TRAE explicitly prioritizes stability and determinism in the following ways.

No adversarial components. MR-TRAE does not use a discriminator network and does not employ adversarial objectives. This removes a major source of training instability commonly encountered in GAN-based SR methods, where the minimax optimization may oscillate, collapse, or become sensitive to hyperparameters.

Deterministic and paired supervision. All supervision signals in MR-TRAE are derived from paired samples $(\mathbf{Y}_{\mathrm{LR}}, \mathbf{X}_{\mathrm{HR}})$. The fidelity losses are computed directly between predictions and their resolution-matched targets: $\widehat{\mathbf{X}}_2 \leftrightarrow \mathbf{X}_{128}$, $\widehat{\mathbf{X}}_4 \leftrightarrow \mathbf{X}_{256}$, and $\widehat{\mathbf{X}}_8 \leftrightarrow \mathbf{X}_{\mathrm{HR}}$.

The perceptual losses are also deterministic because they are computed through a *frozen* Master encoder, which is a fixed function during Stage II. So, the loss is consistent across training runs, and variability is largely restricted to common stochastic factors such as mini-batch sampling and data augmentation.

Structured guidance without instability. The Master-guided latent alignment term encourages the Follower latent $\mathbf{Z}_F$ to be consistent with the HR-trained latent $\mathbf{Z}_M$.

This guidance does not introduce an additional competing objective (as in adversarial learning), but instead provides a stable reference representation learned in Stage I.

This improves convergence behavior while keeping the overall optimization a standard minimization problem.

5.8.2 Perceptual quality without hallucination

In medical imaging, perceptual realism must be balanced against the risk of creating structures that are not present in the underlying anatomy.

MR-TRAE aims to improve perceptual quality while reducing hallucination risk through a domain-specific, frozen perceptual space.

Frozen Master features as a domain-specific perceptual prior. The Master AE is trained on a large set of HR medical images and therefore learns feature representations that are adapted to the domain (e.g., lung CT texture statistics and anatomical structure). During Stage II, the Master encoder is frozen and used to compute perceptual losses between SR outputs and their targets in this learned feature space:

$$\mathcal{L}_{\mathrm{per}}(\widehat{\mathbf{X}}, \mathbf{X}^{\star}) = \sum_{l \in \mathcal{L}} \alpha_l \left\| \phi_M^{(l)}(\widehat{\mathbf{X}}) - \phi_M^{(l)}(\mathbf{X}^{\star}) \right\|_1 . \tag{5.48}$$

Because the target $\mathbf{X}^\star$ is always the paired, resolution-matched ground truth, the perceptual term encourages the SR output to be close to the *true* anatomy in a semantically meaningful space, rather than merely appearing plausible to a discriminator.

Adversarial losses can reward outputs that look "realistic" in aggregate, even when they introduce fine-scale textures that do not correspond to the patient-specific ground truth. In contrast, MR-TRAE compares the prediction and its paired target through a fixed, domain-trained encoder.

This encourages perceptual similarity *anchored* to the ground truth rather than realism in a distributional sense. Moreover, the pixel-domain ℓ_1 losses at all three resolutions constrain the solution to remain faithful to the ground truth intensities, further reducing the likelihood of hallucinated patterns.

Practical interpretation. The frozen Master feature space can be viewed as a learned, domain-specific perceptual prior. It emphasizes structures and textures that matter in the HR medical domain while remaining deterministic and target-referenced.

This makes it particularly attractive for clinical SR settings where interpretability and reliability are critical.

5.8.3 Multi-scale usability and expected trade-offs

A distinctive aspect of MR-TRAE is that it outputs three explicitly supervised SR images: $\widehat{\mathbf{X}}_2 \in \mathbb{R}^{128\times128\times C}$, $\widehat{\mathbf{X}}_4 \in \mathbb{R}^{256\times256\times C}$, and $\widehat{\mathbf{X}}_8 \in \mathbb{R}^{512\times512\times C}$. This provides practical flexibility that is not available in single-output SR pipelines.

$\widehat{\mathbf{X}}_2$ (128×128) is useful for rapid preview, coarse screening, and applications where a modest resolution increase is sufficient. In clinical workflows, this can support quick navigation or triage-style viewing where speed is prioritized. It is also useful in bandwidth-limited settings as a compact SR representation.

$\widehat{\mathbf{X}}_4$ (256×256) is a balanced operating point between detail and robustness. It is suitable for routine visualizations and can provide improved edge definition without being as sensitive as the final $\times 8$ output to challenging degradations.

$\widehat{\mathbf{X}}_8$ (512×512) is the highest-detail output intended for full-resolution interpretation, quantitative post-processing, or archival-quality reconstruction.

Having three supervised outputs enables deployment modes such as:

- Progressive transmission: transmit or display $\widehat{\mathbf{X}}_2$ first, then refine to $\widehat{\mathbf{X}}_4$ and $\widehat{\mathbf{X}}_8$ if needed.
- Adaptive compute: select an output based on runtime constraints (e.g., low-latency systems may stop at $\widehat{\mathbf{X}}_2$).
- Multi-zoom clinical viewing: use lower-resolution SR for global context and higher-resolution SR for local inspection, while staying within a consistent reconstruction framework.

Multi-resolution SR naturally comes with trade-offs that depend on scale. As the upscaling factor increases, small LR ambiguities can lead to larger uncertainty at the pixel level. Thus, $\widehat{\mathbf{X}}_8$ is expected to be the most sensitive to strong degradations (noise/blur) in $\mathbf{Y}_{\mathrm{LR}}$. $\widehat{\mathbf{X}}_2$ is typically more robust but can appear smoother because fewer high-frequency details need to be synthesized.

Perceptual losses can improve perceived sharpness, but if overly weighted they may introduce subtle texture biases. In MR-TRAE, this risk is mitigated by the paired, frozen-Master perceptual space and the presence of strong pixel-domain supervision at all scales.

5.8.4 Limitations

The Master AE is trained using an HR reconstruction objective. To reduce this limitation, the Master AE could ideally be pretrained with a richer objective rather than relying only on a pixel-domain reconstruction loss.

The next limitation is that the training-time cost is higher than that of a simple single-output CNN-based SR model. However, this cost is justified by the fact that one training procedure yields three supervised SR outputs and encourages cross-scale consistency. Moreover, at inference time, the paired HR image and Master AE are not required. The LR input is processed only through the Follower AE and the three sequential SR backbones.

Chapter 6

CNN-Based super-resolution with advanced loss functions

6.1 Overview and motivation

SISR aims to recover an HR image from a single LR observation. The LR observation is denoted by $\mathbf{Y}_{\text{LR}}$ (or, more generally, an LR image with spatial size $H \times W$ and C channels), and the HR ground truth is denoted by $\mathbf{X}_{\text{HR}}$ at the desired scale factor. The SR estimate is denoted by $\widehat{\mathbf{X}}$ (or $\widehat{\mathbf{X}}^{(s)}$ when the scale factor s is emphasized).

A widely used baseline strategy for training SISR models is *regression-based learning*, where the network is optimized using a pixel-wise loss (e.g., ℓ_1 or ℓ_2) between $\widehat{\mathbf{X}}$ and $\mathbf{X}_{\text{HR}}$. Although such objectives often yield good distortion metrics (e.g., PSNR), they are known to encourage averaged solutions when the mapping from LR to HR is ambiguous. As a result, regression-only models frequently produce over-smoothed images: sharp edges become softened and fine textures are suppressed. This limitation is particularly important in high-frequency regions, where multiple plausible HR explanations can correspond to the same LR input.

The main goal of this chapter is to develop a CNN-based SR model that improves the perceptual quality of the reconstructed images compared to regression-only baselines, while maintaining stable training and preserving visually (and, in medical applications, clinically) plausible structures.

To achieve this goal, the chapter proceeds step by step:

1. Weight scaling: we first explain why common variance-preserving initializations (e.g., He/Kaiming initialization) are not always sufficient for deep SR architectures. We then derive a runtime weight-scaling strategy (inspired by StyleGAN) that keeps activation magnitudes predictable across depth.

2. Custom scaled layers: the weight-scaling strategy is embedded into dedicated building blocks used throughout the model, including *ScaledConv2D* and *ScaledLeakyReLU*. In the upsampling stages, we further employ a scaled transposed-convolution upsampler, denoted as *UpsamplerConv2D*.

3. Architecture with progressive upsampling and refinement: we then present the proposed network architecture, which maps an LR input of size $64 \times 64 \times C$ to an HR output of size $512 \times 512 \times C$ using a representation learning block (residual feature extraction at 64×64) followed by three $\times 2$ upsampling blocks (to 128×128, 256×256, and 512×512). At each scale, an intermediate image estimate is produced and progressively refined via additive skip-image connections from coarser scales.

4. Advanced loss functions for perceptual quality: to address over-smoothing and encourage perceptually faithful reconstructions, we introduce loss functions beyond pixel regression, such as perceptual losses. These losses guide the network toward sharper, more realistic high-frequency structures while still respecting the underlying LR observation.

5. Training on lung CT images: finally, we describe the full training pipeline on lung CT images, including preprocessing, LR generation, optimization settings, and validation methodology.

6.2 Problem setup and notation

We denote the HR image by $\mathbf{X}_{\mathrm{HR}}$ and the LR image by $\mathbf{Y}_{\mathrm{LR}}$. The SR output predicted by a model is denoted by $\widehat{\mathbf{X}}$ (or $\widehat{\mathbf{X}}^{(s)}$ when emphasizing the scale factor s).

6.2.1 SISR mapping and scale factors

Let s denote the upscaling factor. In this chapter, we primarily focus on an end-to-end mapping from an LR input of size $64 \times 64 \times C$ to an HR output of size $512 \times 512 \times C$, which corresponds to $s = 8$. Nevertheless, the formulation is general and applies to any scale factor (e.g., $s \in \{2, 4, 8\}$).

We model SISR as learning a parametric function $f_\theta(\cdot)$, where θ denotes the learnable parameters:

$$\widehat{\mathbf{X}}^{(s)} = f_\theta(\mathbf{Y}_{\mathrm{LR}}). \tag{6.1}$$

In our architecture (introduced later in this chapter), the reconstruction is performed progressively across intermediate resolutions. Therefore, the model may produce intermediate SR outputs (e.g., at $\times 2$ and $\times 4$) in addition to the final $\times 8$ output. We denote these intermediate predictions consistently as:

$$\widehat{\mathbf{X}}_2,\ \widehat{\mathbf{X}}_4,\ \widehat{\mathbf{X}}_8, \tag{6.2}$$

where $\widehat{\mathbf{X}}_8$ is the final output at the target resolution.

6.2.2 LR synthesis via a degradation model

In supervised training, we require paired examples $(\mathbf{Y}_{\mathrm{LR}}, \mathbf{X}_{\mathrm{HR}})$. While some datasets provide native LR–HR pairs, in many imaging pipelines LR images are synthesized from HR images using a controlled degradation operator. We represent the LR synthesis process as:

$$\mathbf{Y}_{\mathrm{LR}} = \mathcal{D}(\mathbf{X}_{\mathrm{HR}}), \tag{6.3}$$

where $\mathcal{D}(\cdot)$ denotes the degradation pipeline.

$\mathcal{D}(\cdot)$ consists of the following components:

- Stochastic blur with probability p_{blur}: randomly convolve $\mathbf{X}_{\text{HR}}$ with an isotropic Gaussian kernel k_σ with kernel size k and standard deviation σ:

$$\mathbf{X}_{\text{b}} = \mathbf{X}_{\text{HR}} * k_\sigma. \tag{6.4}$$

- Downsampling: downsample the (blurred or unblurred) image using bicubic interpolation to the LR resolution:

$$\widehat{\mathbf{Y}} = \text{BicubicDown}(\mathbf{X}_{\text{b}}, s). \tag{6.5}$$

- Additive noise: add zero-mean Gaussian noise n:

$$\begin{aligned} \mathbf{Y}_{\text{tmp}} &= \widehat{\mathbf{Y}} + n, \\ n &\sim \mathcal{N}(0, \sigma_n^2), \end{aligned} \tag{6.6}$$

  where σ_n denotes the standard deviation of the additive noise used in the LR synthesis process.

- Clipping to valid intensity range: clip the LR output to the valid range:

$$\mathbf{Y}_{\text{LR}} = \text{clip}(\mathbf{Y}_{\text{tmp}}, 0, 1). \tag{6.7}$$

The model is trained not only to invert a single deterministic downsampling operator, but also to handle mild blur and realistic noise.

6.2.3 Training objective

Given a training set of paired samples $\{(\mathbf{Y}_{\text{LR}}^{(i)}, \mathbf{X}_{\text{HR}}^{(i)})\}_{i=1}^{N}$, the model parameters θ are learned by minimizing a total loss:

$$\theta^* = \arg\min_\theta \; \mathbb{E}_{(\mathbf{Y}_{\text{LR}}, \mathbf{X}_{\text{HR}})} \Big[\mathcal{L}_{\text{total}}(f_\theta(\mathbf{Y}_{\text{LR}}), \mathbf{X}_{\text{HR}}) \Big]. \tag{6.8}$$

$\mathcal{L}_{\text{total}}(\cdot)$ is not restricted to a pixel-wise regression term. Instead, it is constructed from multiple complementary components which will be detailed later: (i) robust regression (e.g., Charbonnier), (ii) perceptual fidelity, (iii) edge/gradient preservation, (iv) frequency-domain alignment, and (v) style/texture statistics.

Although the architecture may produce intermediate reconstructions (e.g., at $\times 2$ and $\times 4$) as part of a progressive refinement pipeline, the training objective is defined only on the final SR output.

6.2.4 Perceptual quality

The term perceptual quality refers to reconstructions that exhibit high visual fidelity, meaning that edges, contrast transitions, and fine-scale textures appear natural and consistent with the underlying image content. It also preserve structural integrity, which is especially important for CT imagery where unrealistic artifacts can misrepresent clinically meaningful patterns.

Although the experiments are conducted on grayscale data ($C = 1$), the model description and formulation are kept general.

6.3 Weight scaling

In deep CNNs, a central practical goal is to keep the *signal magnitude* (and therefore gradients) in a reasonable range as we propagate through many layers. If the activations gradually shrink, the network becomes hard to optimize (vanishing signal/gradients). If they grow, training becomes unstable (exploding signal/gradients). In this section, we build the weight-scaling strategy step by step, starting from a plain convolution, then moving to variance-preserving initialization (He/Kaiming), then to runtime weight scaling (ProGAN) [30], and finally to the activation-scaled formulation, which we adopt in this thesis.

6.3.1 Convolution operator, the origin of fan_in, and why variance grows

Figure 6.1 illustrates a standard 2D convolution operator. A single output pixel (at some spatial location) in one output channel is produced by taking a dot product between: (i) a local input patch of size $k \times k$ across all c_{in} input channels, and (ii) the corresponding kernel weights (one kernel per output channel).

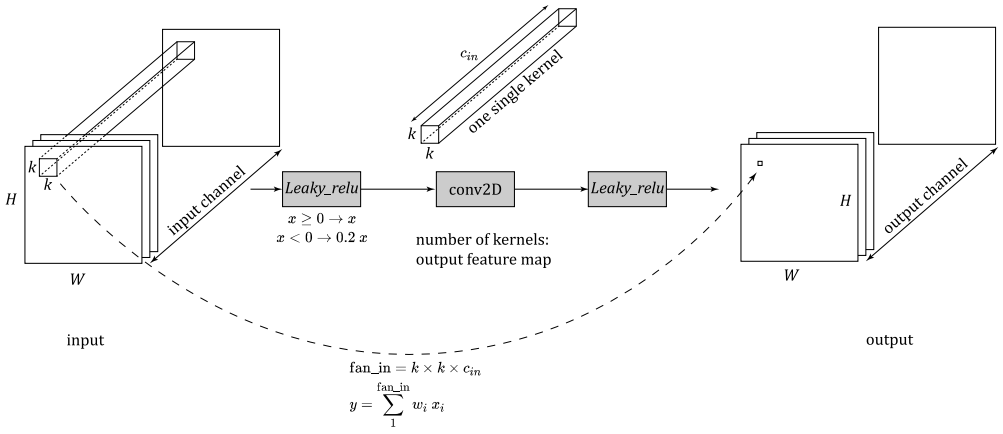

Figure 6.1. Schematic diagram of the convolution operation. For one output value, the convolution sums $k \times k \times c_{\text{in}}$ products between the input patch entries and kernel weights; this quantity is the *fan-in* of the convolution.

For one output element (one pixel, one output channel), the number of multiplicative

terms in the sum equals:

$$\text{fan\_in} = k \times k \times c_{\text{in}}. \tag{6.9}$$

This is exactly where *fan-in* comes from: it counts how many inputs feed into one scalar output of the convolution.

To see why fan-in matters, consider the pre-activation output of a convolution at a fixed location and output channel:

$$y = \sum_{i=1}^{\text{fan\_in}} w_i x_i, \tag{6.10}$$

where x_i are the elements of the input patch (across spatial positions and channels), and w_i are the corresponding kernel weights.

Assume that x_i and w_i are independent, zero-mean random variables:

$$\begin{aligned} x_i &\sim \mathcal{N}(0,1), \\ w_i &\sim \mathcal{N}(0,1). \end{aligned} \tag{6.11}$$

Then each product has variance:

$$\text{Var}(w_i x_i) = \text{Var}(w_i)\text{Var}(x_i) = 1, \tag{6.12}$$

and because y is a sum of fan_in such terms, its variance grows approximately linearly with fan_in:

$$\begin{aligned} \text{Var}(y) &\approx \text{fan\_in}, \\ \text{Std}(y) &\approx \sqrt{\text{fan\_in}}. \end{aligned} \tag{6.13}$$

For example, with $k = 5$ and $c_{\text{in}} = 256$,

$$\text{fan\_in} = 5 \times 5 \times 256 = 6400, \quad \Rightarrow \quad \text{Std}(y) \approx \sqrt{6400} = 80. \tag{6.14}$$

This simple calculation predicts a key practical issue: if we keep weights with standard deviation 1 regardless of layer shape, then deeper layers (with large fan-in) can produce very large activations.

6.3.2 A controlled experiment: tracking input/weight/output statistics

To turn the above intuition into something we can directly observe, we define a small experiment that prints summary statistics (mean, standard deviation, and a few sample values) of the input activations, convolution weights, and output activations.

```
def tensor_stats_summary(x: tf.Tensor, n_examples: int = 5) ->
    str:
    # return a summary: mean, std, and the first elements of
       the input tensor
```

```
    x = tf.convert_to_tensor(x)
    mean = tf.reduce_mean(x)
    std = tf.math.reduce_std(x)
    flat = tf.reshape(x, [-1])
    sample = flat[:n_examples]
    sample_str = " ".join([f"{v.numpy():6.2f}" for v in sample
        ])
    return f"{mean.numpy():6.2f} +/- {std.numpy():6.2f} [{
        sample_str}]"
```

Next, we simulate a single convolutional layer and optionally apply an activation function (we use LeakyReLU). The experiment allows us to control: (i) the weight initialization standard deviation, and (ii) an explicit multiplicative *weight scaling* factor applied at runtime.

```
def demo_weight_scaling_conv(
    weight_init_std: float = 1.0,
    weight_scale: float = 1.0,
    in_channels: int = 256,
    out_channels: int = 256,
    kernel_size: int = 5,
    activation=tf.nn.leaky_relu,
    spatial_size: int = 64,
    batch_size: int = 1,
) -> None:
    """
    experiment: sample a random input, optionally apply
        activation,
    sample random conv weights, apply weight scaling, run
        conv2d, optionally activate,
    and print summary statistics for input/weights/output.
    """
    # Random input feature map: [N, H, W, C] (NHWC)
    x = tf.random.normal([batch_size, spatial_size,
        spatial_size, in_channels])
    if activation is not None:
        x = activation(x)
    print(f"  Input: {tensor_stats_summary(x)}")

    # Random conv kernel: [kH, kW, Cin, Cout]
    w = tf.random.normal(
        [kernel_size, kernel_size, in_channels, out_channels],
        stddev=weight_init_std,
    )
    print(f"Weights: {tensor_stats_summary(w)}")

    # Convolution with scaled weights
    y = tf.nn.conv2d(
```

```
        x,
        w * weight_scale,
        strides=[1, 1, 1, 1],
        padding="SAME",
    )
    if activation is not None:
        y = activation(y)
    print(f" Output: {tensor_stats_summary(y)}")
```

In each run below, we interpret the printed statistics as follows:

- **Mean** should remain close to zero in the purely linear case (zero-mean inputs and weights).
- **Standard deviation (std)** tells us whether the layer amplifies or attenuates the signal.
- A good operating regime is typically one in which the input and output std remain in a comparable range across layers, so that optimization hyperparameters (especially the learning rate) behave consistently throughout the network.

6.3.3 Case 1: no activation, no scaling (why outputs explode)

We begin with the most direct setting: no activation function and weights sampled from $\mathcal{N}(0,1)$ with no additional scaling.

```
demo_weight_scaling_conv(activation=None)
#results:
Input:   0.00 +/-   1.00 [ -0.27  -2.00   1.12   1.02  -0.08]
Weight:  0.00 +/-   1.00 [ -1.25  -0.29  -0.74   1.62   0.12]
Output:  0.00 +/-  78.67 [ 29.96 -72.20 -50.93 -44.70 -33.66]
```

The input is standard normal: mean ≈ 0 and std ≈ 1. The weights are also standard normal: mean ≈ 0 and std ≈ 1. However, the output std becomes ≈ 78.67, which is extremely large compared to the input. This is exactly what Eq. (6.13) predicts.

Here, $k = 5$ and $c_{\text{in}} = 256$, so fan_in $= 6400$ and $\sqrt{\text{fan\_in}} \approx 80$. Therefore, even if each product term has variance ≈ 1, the sum of 6400 terms produces a large output variance. If a network stacks many such layers without compensating for fan-in, the activations can quickly explode and destabilize training.

6.3.4 Case 2: variance-preserving scaling for the linear case

A natural fix is to reduce the weight variance so that the output variance stays comparable to the input variance. From:

$$\text{Var}(y) \approx \text{fan\_in}\,\text{Var}(w)\,\text{Var}(x), \tag{6.15}$$

if $\mathrm{Var}(x) \approx 1$ and we want $\mathrm{Var}(y) \approx 1$, we can choose:

$$\mathrm{Var}(w) = \frac{1}{\mathrm{fan\_in}} \quad \Rightarrow \quad \mathrm{Std}(w) = \sqrt{\frac{1}{\mathrm{fan\_in}}}. \tag{6.16}$$

We test this by setting $\mathrm{Std}(w) = \sqrt{1/\mathrm{fan\_in}}$.

```
fan_in = 5 * 5 * 256
demo_weight_scaling_conv(weight_init_std=tf.sqrt(1.0/fan_in),
    activation=None)

#results:
Input:  -0.00 +/-  1.00 [  1.47  -0.19  -1.62  -1.44  -0.43]
Weight: -0.00 +/-  0.01 [  0.02  -0.00   0.01   0.01  -0.01]
Output:  0.00 +/-  0.98 [  0.12  -0.56  -0.75  -0.04   0.63]
```

Now the output std is ≈ 0.98, very close to 1. This resolves the explosion observed in Case 1. The trade-off is visible in the weights: because $\mathrm{fan\_in} = 6400$, we have $\sqrt{1/\mathrm{fan\_in}} \approx 0.0125$, so the weight samples are very small (std ≈ 0.01).

At this point, things look good if the layer is purely linear. But deep CNNs rely on non-linear activations; the next case shows why this is not yet sufficient.

6.3.5 Case 3: adding an activation shrinks the signal

We repeat Case 2, but now we apply a LeakyReLU activation (both to the input feature map and the output of convolution, as in the helper function). In practice, this resembles what happens in a real CNN block: activation-conv-activation patterns are common.

```
fan_in = 5 * 5 * 256
demo_weight_scaling_conv(weight_init_std=tf.sqrt(1.0/fan_in))

#results:
Input:   0.32 +/-  0.65 [ -0.32   0.12   1.09  -0.15  -0.14]
Weight: -0.00 +/-  0.01 [  0.00  -0.00  -0.01   0.03   0.01]
Output:  0.22 +/-  0.45 [ -0.16  -0.03   0.67   0.51   0.32]
```

Two important changes occur:

1. The input distribution is no longer symmetric. LeakyReLU passes positive values unchanged and scales negative values by a factor (e.g., 0.2). As a result, the mean becomes positive (here ≈ 0.32).
2. The standard deviation shrinks. The input std becomes ≈ 0.65 (instead of ≈ 1), and the output std becomes even smaller (≈ 0.45).

This is the key issue: if we keep using the linear variance rule in Eq. (6.16) while repeatedly applying a non-linearity, the signal tends to gradually shrink as depth

increases. In deep networks, repeated shrinkage can lead to vanishing activations and weak gradient flow.

6.3.6 Case 4: He/Kaiming initialization

A widely used fix is He (Kaiming) initialization, which increases the weight variance by roughly a factor of 2 to compensate for the variance reduction caused by ReLU-like activations:

$$\text{Std}(w) = \sqrt{\frac{2}{\text{fan\_in}}}. \tag{6.17}$$

We test this directly.

```
fan_in = 5 * 5 * 256
demo_weight_scaling_conv(weight_init_std=tf.sqrt(2.0/fan_in))

#results:
Input:   0.32 +/-   0.65 [  0.26  -0.04   0.19  -0.04  -0.08]
Weight: -0.00 +/-   0.02 [ -0.01  -0.03  -0.02  -0.02  -0.03]
Output:  0.31 +/-   0.64 [ -0.02  -0.12  -0.02   0.45  -0.05]
```

Now the output std (≈ 0.64) matches the input std (≈ 0.65) very well. This indicates that the layer approximately preserves signal magnitude when combined with a ReLU-like non-linearity, which is exactly the intent of He initialization.

However, there is still a practical concern that becomes apparent when we change the convolution shape. Consider two different layers:

1. A small fan_in layer (1x1 conv, 64 channels):

```
fan_in = 1 * 1 * 64
demo_weight_scaling_conv(
    weight_init_std=tf.sqrt(2.0/fan_in),
    kernel_size=1, in_channels=64, out_channels=64
)

#results:
Input:   0.32 +/-   0.65 [  0.70   0.26  -0.07  -0.19   0.74]
Weight:  0.00 +/-   0.18 [  0.30   0.27   0.05  -0.03   0.14]
Output:  0.33 +/-   0.67 [ -0.03   2.59  -0.29   0.02   0.05]
```

2. A large fan_in layer (7x7 conv, 512 channels):

```
fan_in = 7 * 7 * 512
demo_weight_scaling_conv(
    weight_init_std=tf.sqrt(2.0/fan_in),
    kernel_size=7, in_channels=512, out_channels=512
)
```

```
#results:
Input:    0.32 +/-    0.65 [ -0.02    0.20   -0.07   -0.12   -0.25]
Weight:  -0.00 +/-    0.01 [  0.02   -0.00   -0.02   -0.02   -0.01]
Output:   0.31 +/-    0.64 [  0.50    0.17   -0.18   -0.13    0.70]
```

Both layers preserve the activation scale reasonably well, but the "weight standard deviation" depends on fan_in. In the 1×1 case, weights have std ≈ 0.18, while in the 7×7 case, weights have std ≈ 0.01.

This fan_in dependence is not necessarily wrong, but it creates an optimization nuisance: different layers live on different parameter scales. When weight magnitudes differ substantially across layers, a single global learning rate can effectively behave differently from one layer to another (some layers update "too fast", others "too slow"). This motivates the next step: keep the *stored* weight distribution consistent across layers, while applying the fan_in correction as a deterministic scaling during the forward pass.

6.3.7 ProGAN-style runtime weight scaling: stable activations with uniform weight storage

The idea is that:

1. initialize all convolution weights using the *same* standard distribution, e.g., $w \sim \mathcal{N}(0, 1)$.
2. during the forward pass, multiply weights by a deterministic scaling factor that depends on fan_in.

In the ProGAN formulation, the scaling factor is chosen similar to He:

$$w_{\text{effective}} = w \cdot \underbrace{\sqrt{\frac{2}{\text{fan\_in}}}}_{\text{runtime scale}}, \tag{6.18}$$

so the stored weights remain $\mathcal{N}(0, 1)$ regardless of layer shape, but the effective weights match a He-scaled magnitude at runtime.

We test this for two different layers:

1. ProGAN weight scaling: `w ~ N(0,1)`, `w_scale = sqrt(2/fan_in)` (1x1 conv, 64 channels):

```
fan_in = 1 * 1 * 64
demo_weight_scaling_conv(
    weight_init_std=1.0,
    weight_scale=tf.sqrt(2.0/fan_in),
    kernel_size=1, in_channels=64, out_channels=64
)

#results:
#Input:    0.32 +/-    0.65 [ -0.01   -0.01   -0.23   -0.45    1.60]
```

```
#Weight:  0.04 +/-   1.00 [ -0.69  -0.42  -0.91  -0.72  -1.03]
#Output:  0.41 +/-   0.71 [  1.21   0.81  -0.09  -0.06   0.71]
```

2. ProGAN weight scaling: `w ~ N(0,1)`, `w_scale = sqrt(2/fan_in)` (3x3 conv, 512 channels):

```
fan_in = 3 * 3 * 512
demo_weight_scaling_conv(
    weight_init_std=1.0,
    weight_scale=tf.sqrt(2.0/fan_in),
    kernel_size=3, in_channels=512, out_channels=512
)

#results:
Input:   0.32 +/-  0.65 [ -0.12  -0.38  -0.13   0.67  -0.12]
Weight:  0.00 +/-  1.00 [ -0.34   1.22  -0.91   1.04  -0.86]
Output:  0.33 +/-  0.66 [ -0.23   0.43  -0.12   0.48   0.39]
```

In both cases, the outputs remain well-behaved (std close to the input std). Meanwhile, the weights are consistently stored with std ≈ 1 for both layers. This is the main benefit of runtime weight scaling: we decouple the statistical scale of stored parameters from the layer shape, which makes optimization more uniform across the network.

6.3.8 A remaining issue: the "factor 2" depends on whether a non-linearity exists

At first glance, ProGAN scaling seems to solve everything. However, the factor 2 (from He) is fundamentally tied to the presence of a ReLU-like non-linearity. If we use the same $\sqrt{2/\text{fan\_in}}$ in a *linear* setting (no activation), the layer output becomes too large.

We can see this by removing activation:

```
fan_in = 3 * 3 * 512
demo_weight_scaling_conv(
    weight_init_std=1.0,
    weight_scale=tf.sqrt(2.0/fan_in),
    kernel_size=3, in_channels=512, out_channels=512,
    activation=None
)

#results:
Input:   0.00 +/-   1.00 [ -1.35   0.51  -0.63   2.58   0.22]
Weight:  0.00 +/-   1.00 [  0.51  -1.01  -1.03   2.42   0.65]
Output: -0.00 +/-   1.40 [  0.58  -1.46  -0.57   0.96   0.28]
```

The output std is ≈ 1.40 instead of ≈ 1.00. This is expected: when the layer is linear, the variance-preserving scaling should match Eq. (6.16), i.e., $\sqrt{1/\text{fan\_in}}$ rather than

$\sqrt{2/\text{fan\_in}}$.

Indeed, if we switch to the linear scaling factor:

```
fan_in = 3 * 3 * 512
demo_weight_scaling_conv(
    weight_init_std=1.0,
    weight_scale=tf.sqrt(1.0/fan_in),
    kernel_size=3, in_channels=512, out_channels=512,
    activation=None
)

#results:
Input:    0.00 +/-    1.00 [  0.76    0.22    0.36    0.56   -1.39]
Weight:   0.00 +/-    1.00 [ -0.64    1.84    1.79   -0.65   -0.52]
Output:  -0.00 +/-    0.99 [  0.70   -0.34    0.68    0.14    0.61]
```

Now the output std is ≈ 0.99, which matches the input std, so the linear case is fixed. But this introduces a design tension:

- If we set $w_{\text{scale}} = \sqrt{2/\text{fan\_in}}$, the non-linear case behaves well, but the linear case is too large.
- If we set $w_{\text{scale}} = \sqrt{1/\text{fan\_in}}$, the linear case behaves well, but we must still compensate for the non-linearity's variance reduction.

To resolves this issue, we use a "fixed" weight scaling rule and we move the "factor 2" compensation into the activation itself.

6.3.9 Fixed weight scaling and activation scaling

The StyleGAN approach adopts:

$$w_{\text{effective}} = w \cdot \sqrt{\frac{1}{\text{fan\_in}}}, \tag{6.19}$$

which is always independent of activation, and then scales the activation output by $\sqrt{2}$:

$$\phi_{\text{scaled}}(x) = \sqrt{2}\,\phi(x), \tag{6.20}$$

where $\phi(\cdot)$ is a LeakyReLU (or ReLU-like) function.

Concretely:

```
def scaled_leaky_relu(x):
    return leaky_relu(x) * sqrt(2.0)
```

Now we test this combined rule:

- $w_{\text{scale}} = \sqrt{1/\text{fan\_in}}$, and
- activation `scaled_leaky_relu`.

```
fan_in = 3 * 3 * 512
demo_weight_scaling_conv(
    weight_init_std=1.0,
    weight_scale=tf.sqrt(1.0/fan_in),
    kernel_size=3, in_channels=512, out_channels=512,
    activation=scaled_leaky_relu
)

#results:
#Input:   0.45 +/-   0.91 [  0.67   0.57  -0.56   1.92   0.04]
#Weight: -0.00 +/-   1.00 [ -1.10   0.37   1.05   0.84   1.46]
#Output:  0.43 +/-   0.91 [  1.87   0.81  -0.05   0.99  -0.18]
```

This is the desired behavior:

- The stored weights remain $\mathcal{N}(0, 1)$ (std ≈ 1.00), independent of layer shape.
- The input and output activation std are well matched (≈ 0.91 in this run), indicating stable signal propagation through a non-linear conv block.

Finally, if we remove the activation entirely while keeping $w_{\text{scale}} = \sqrt{1/\text{fan\_in}}$, the linear case remains correct:

```
fan_in = 3 * 3 * 512
demo_weight_scaling_conv(
    weight_init_std=1.0,
    weight_scale=tf.sqrt(1.0/fan_in),
    kernel_size=3, in_channels=512, out_channels=512,
    activation=None
)

#results:
Input:  -0.00 +/-   1.00 [ -1.56   0.51   0.08   0.53  -0.57]
Weight: -0.00 +/-   1.00 [ -0.15  -0.40   1.32   1.91   0.03]
Output: -0.00 +/-   0.99 [  0.64  -0.10   0.77   0.49  -0.49]
```

Therefore, the final strategy we adopt is:

- Initialize convolution weights as $w \sim \mathcal{N}(0, 1)$.
- Compute $\text{fan\_in} = k \times k \times c_{\text{in}}$ for each convolution kernel.
- Apply runtime scaling $w_{\text{effective}} = w \cdot \sqrt{1/\text{fan\_in}}$.
- Use activation scaling $\phi_{\text{scaled}}(x) = \sqrt{2}\,\phi(x)$ (e.g., scaled LeakyReLU).

This formulation simultaneously achieves: (i) stable activation magnitudes across layers, (ii) consistent stored weight statistics across different kernel sizes and channel counts, and (iii) a clean separation between linear variance preservation (handled by $\sqrt{1/\text{fan\_in}}$) and non-linear compensation (handled by the activation scaling).

These properties are particularly useful in deep super-resolution architectures, where training stability and consistent optimization behavior across many blocks are essential. We use these strategies to design our customized convolutional layer (named: "Scaled Conv2D" and customized LeakyReLU activation (named: "Scaled LeakyReLU").

6.4 Custom layers derived from weight scaling

Section 6.3 established the scaling strategy we adopt: we store convolution weights with a layer-independent distribution (e.g., zero-mean unit-variance), and we apply a deterministic runtime scaling factor based on fan_in to stabilize activation magnitudes across depth. This section turns that strategy into three practical building blocks used throughout the proposed SR network: (i) a scaled convolution layer, (ii) a scaled LeakyReLU activation, and (iii) a fused upsampling–convolution block for efficiency.

We present the concepts on how scaling is defined, where it is applied, how parameters are stored, and why these blocks improve stability and efficiency.

6.4.1 ScaledConv2D layer

Definition and operation

A standard 2D convolution computes (for each output channel and spatial location) a dot product between a local input patch and a learnable kernel. In the scaled formulation, we separate the kernel into:

- a stored kernel $\mathbf{W}$ whose entries are maintained in a layer-independent statistical range (conceptually, $\mathbf{W}$ is drawn from $\mathcal{N}(0,1)$ at initialization and then updated by learning),
- a fixed scale constant c (not trainable), computed from the convolution's fan_in,
- and a standard bias vector $\mathbf{b}$.

Let the convolution kernel have spatial size $k \times k$, and let the number of input channels be c_{in}. Then:

$$\text{fan\_in} = k \times k \times c_{\text{in}}, \qquad c = \sqrt{\frac{1}{\text{fan\_in}}}. \tag{6.21}$$

The forward pass of ScaledConv2D is defined by:

$$\mathbf{Y} = \text{Conv2D}(\mathbf{X},\ c\mathbf{W}) + \mathbf{b}, \tag{6.22}$$

where $\mathbf{X}$ is the input feature map and $\mathbf{Y}$ is the output feature map.

Where scaling is applied. The scaling factor c multiplies the kernel *during the forward computation*. The trainable parameter stored in memory is $\mathbf{W}$, while c is a deterministic constant computed once from the kernel shape (and therefore fixed for that layer). This matches the principle from Section 6.3: keep stored weights in a uniform range; enforce variance control at runtime using fan_in.

The ScaledConv2D layer stores:

- Kernel weights $\mathbf{W}$ (trainable), initialized from a distribution that does not depend on fan_in (commonly unit-variance).
- Bias $\mathbf{b}$ (trainable), typically initialized to zero.
- Scale constant c (non-trainable), computed from (6.21).

Compared to applying fan_in scaling only through initialization (He initialization), ScaledConv2D gives three advantages:

1. Stable magnitudes across depth and across layer shapes: because c explicitly corrects for fan_in at runtime, outputs remain in a predictable scale even when kernel sizes and channel counts vary across the architecture.
2. Uniform parameter scale across layers: the stored weights $\mathbf{W}$ live in a similar numeric range in every layer, instead of some layers having very small weights (large fan_in) and others having large weights (small fan_in). This makes the effect of a global learning rate more consistent across layers.
3. Cleaner design separation: the linear variance-preserving part is handled by $c = \sqrt{1/\text{fan\_in}}$ (independent of activation choice), while any non-linearity compensation is delegated to the activation block (Section 6.4.2).

Algorithm 5 shows a schematic form of the implementation of the ScaledConv2D layer.

Algorithm: ScaledConv2D

Algorithm 5 Forward pass of ScaledConv2D

Require: Input feature map $\mathbf{X}$ with c_{in} channels; kernel size k; trainable kernel $\mathbf{W}$; trainable bias $\mathbf{b}$.

1: Compute fan_in $\leftarrow k \times k \times c_{\text{in}}$.
2: Compute scale constant $c \leftarrow \sqrt{1/\text{fan\_in}}$.
3: Compute effective kernel $\mathbf{W}_{\text{eff}} \leftarrow c\mathbf{W}$.
4: Compute convolution output $\mathbf{Y} \leftarrow \text{Conv2D}(\mathbf{X}, \mathbf{W}_{\text{eff}})$.
5: Add bias: $\mathbf{Y} \leftarrow \mathbf{Y} + \mathbf{b}$.
6: **return** $\mathbf{Y}$.

6.4.2 ScaledLeakyReLU activation

Definition and how it complements ScaledConv2D

A LeakyReLU activation is defined as:

$$\mathrm{LReLU}(x;\alpha) = \begin{cases} x, & x \geq 0, \\ \alpha x, & x < 0, \end{cases} \tag{6.23}$$

where $\alpha \in (0,1)$ is the negative slope (commonly $\alpha = 0.2$).

As shown in Section 6.3, ReLU-like activations reduce the variance of their inputs (negative values are down-scaled). If ScaledConv2D uses the purely linear variance-preserving scaling $c = \sqrt{1/\mathrm{fan\_in}}$, then without any additional correction, repeated application of non-linearities can gradually shrink the activation magnitude through depth.

The "ScaledLeakyReLU" resolves this by multiplying the activation output by a gain factor g:

$$\mathrm{ScaledLReLU}(x;\alpha,g) = g \cdot \mathrm{LReLU}(x;\alpha). \tag{6.24}$$

Optional gain factor and why it is used

In the StyleGAN-style strategy adopted here, the conventional choice is:

$$g = \sqrt{2}. \tag{6.25}$$

This choice compensates for the variance reduction introduced by ReLU-like nonlinearities and makes the overall block ScaledConv2D → ScaledLeakyReLU which is approximately variance-preserving across layers.

Placing this correction in the activation (instead of embedding the factor 2 inside the weight scaling) keeps the convolution scaling rule independent of whether the convolution is followed by an activation. This makes the design modular and avoids the mismatch between linear and non-linear cases discussed in Section 6.3.

Algorithm 6 shows a schematic form of the implementation of the ScaledLeakyReLU layer.

Algorithm: ScaledLeakyReLU

6.4.3 Scaled transposed convolution for upsampling

The SR networks must increase spatial resolution, usually by a factor of 2 per upsampling stage. A common pattern is: "Upsample → Convolution → Activation".

This pattern performs an explicit upsampling operation that produces a larger intermediate feature map: $(H, W) \mapsto (2H, 2W)$. This large intermediate tensor must be written to memory and be processed by convolution at high resolution.

This is inefficient because the explicit upsampled tensor can dominate memory bandwidth and cache usage. In practice, for many CNN workloads, memory reading/writing becomes a major bottleneck.

Algorithm 6 Forward pass of ScaledLeakyReLU

Require: Input tensor $\mathbf{X}$; negative slope α; gain g (default $g = \sqrt{2}$).

for each element x in $\mathbf{X}$ **do**
  if $x \geq 0$ **then**
    $y \leftarrow x$
  else
    $y \leftarrow \alpha x$
  end if
  $y \leftarrow g \cdot y$
end for
return $\mathbf{Y}$ (same shape as $\mathbf{X}$).

Our approach avoids constructing the explicit upsampled tensor. Instead, it directly produces the HR output in one operator that combines both steps.

There are two conceptually equivalent ways to define a scaled transposed convolution for upsampling:

View A: convolution on an upsampled signal (conceptual).

$$\mathbf{Y} = \text{Conv2D}(\text{Upsample}(\mathbf{X}), \mathbf{W}_{\text{eff}}) + \mathbf{b}. \tag{6.26}$$

View B: transposed convolution/fractionally-strided convolution. We define an operator that maps $\mathbf{X}$ directly to a higher-resolution $\mathbf{Y}$ without explicitly creating Upsample($\mathbf{X}$):

$$\mathbf{Y} = \text{UpConv2D}(\mathbf{X}, \mathbf{W}_{\text{eff}}) + \mathbf{b}, \tag{6.27}$$

where UpConv2D$(\cdot)$ denotes a learnable upsampling convolution often implemented as a transposed convolution. The important point is that the upsampling and convolution are executed as one single mapping. We apply runtime weight scaling:

$$\mathbf{W}_{\text{eff}} = c\mathbf{W}, \qquad c = \sqrt{\frac{1}{\text{fan\_in}}}, \tag{6.28}$$

where fan_in is computed from the number of contributing inputs per output element of the operator. It corresponds to $k \times k \times c_{\text{in}}$ up to a constant factor depending on how channels are arranged; conceptually, it is still "how many inputs feed one output".

Why scaled transposed convolution for upsampling is more efficient

The efficiency benefit comes from reducing intermediate memory. The upsampling step creates a large intermediate tensor of size proportional to $(2H) \times (2W) \times c$. This tensor is then consumed by convolution. The cost is not only the arithmetic but also writing and reading a large amount of data. In contrast, with scaled transposed convolution, the computation produces the final HR output directly, so that we use fewer memory writes/reads.

In SR, where HR feature maps are large and multiple upsampling stages exist, this saving can be significant. Moreover, this formulation can be implemented to reuse computations and to streamline the data flow, improving throughput without changing the mathematical mapping described by (6.26).

Algorithm 7 shows a schematic form of the implementation of the "scaled transposed convolution for upsampling" layer which is denoted simply by "UpsamplerConv2D".

Algorithm: scaled transposed convolution for upsampling (UpsamplerConv2D)

Algorithm 7 Forward pass of UpsamplerConv2D

Require: Input feature map $\mathbf{X}$; upsampling factor s (typically $s = 2$); kernel size k; stored trainable kernel $\mathbf{W}$; trainable bias $\mathbf{b}$; optional post-filtering operator $\mathcal{B}(\cdot)$.

1: Compute a fan_in value for the upsampling convolution.
2: Compute scale constant $c \leftarrow \sqrt{1/\text{fan\_in}}$.
3: Compute effective kernel $\mathbf{W}_{\text{eff}} \leftarrow c\mathbf{W}$.
4: Compute HR output directly:
$\mathbf{Y} \leftarrow \text{UpConv2D}(\mathbf{X}, \mathbf{W}_{\text{eff}}, s)$ ▷ produces resolution (sH, sW)
5: Add bias: $\mathbf{Y} \leftarrow \mathbf{Y} + \mathbf{b}$.
6: **if** a post-filtering / smoothing step is used **then**
7: $\mathbf{Y} \leftarrow \mathcal{B}(\mathbf{Y})$.
8: **end if**
9: **return** $\mathbf{Y}$.

6.4.4 Summary of custom modules

Table 6.1 gives a summary of the three custom modules which we introduced.

Table 6.1. Summary of custom layers.

Module	Input → Output	Key parameters
ScaledConv2D	$\mathbf{X} \in \mathbb{R}^{H\times W\times c_{\text{in}}} \rightarrow \mathbf{Y} \in \mathbb{R}^{H\times W\times c_{\text{out}}}$	Kernel size k; channels $(c_{\text{in}}, c_{\text{out}})$; stored $\mathbf{W}$ (trainable); bias $\mathbf{b}$ (trainable); scale $c = \sqrt{1/\text{fan\_in}}$ (fixed)
ScaledLeakyReLU	$\mathbf{X} \rightarrow \mathbf{Y}$ (same shape)	Negative slope α; gain g (often $\sqrt{2}$)
UpsamplerConv2D	$\mathbf{X} \in \mathbb{R}^{H\times W\times c_{\text{in}}} \rightarrow \mathbf{Y} \in \mathbb{R}^{(sH)\times(sW)\times c_{\text{out}}}$	Upscale factor s (usually 2); kernel size k; stored $\mathbf{W}$ (trainable); bias $\mathbf{b}$ (trainable); scale c (fixed); optional post-filter $\mathcal{B}$

6.5 Proposed CNN architecture

This section presents the proposed CNN for SISR, built directly from the scaling-aware components introduced in Sections 6.3 and 6.4. The model takes an LR input of size $64 \times 64 \times C$ and produces an SR output of size $512 \times 512 \times C$ (i.e., scale factor $s = 8$). Although our experiments focus on grayscale CT images ($C = 1$), the architecture description is kept general and applies identically to multi-channel images (e.g., RGB with $C = 3$).

6.5.1 Design goals

Super-resolving medical CT images puts some practical constraints. Edges and fine structural boundaries must remain geometrically consistent; excessive smoothing can erase subtle but meaningful details.

While perceptual losses can improve sharpness, the SR model should not introduce structures that are not supported by the LR evidence. In medical imaging, visually plausible but incorrect details are undesirable.

CT images have physically meaningful intensity patterns (after normalization). The SR output should remain consistent with the intensity distribution and dynamic range of the HR target.

For these reasons, the proposed SR CNN architecture is convenient for three reasons: (i) it employs scaled layers (ScaledConv2D and ScaledLeakyReLU) to guarantee stable optimization, (ii) it extracts strong LR representations through a residual backbone before upsampling to ease the workload of the HR layers, and (iii) it performs "progressive reconstruction" with "image-level" skip connections, so each finer scale focuses on refining details rather than re-creating the whole image from scratch.

6.5.2 Model overview

Figure 6.2 provides a schematic diagram of the model's full pipeline. This network is divided into the following components.

1. A "representation learning block" designed for feature extraction at 64×64 spatial size (the input image spatial size).
2. Three "upsampling plus refinement blocks" that increase resolution by $\times 2$ each time (to 128×128, 256×256, and 512×512).
3. "Progressive image skip connections" that upsample a coarser image estimate and add it to a finer one.

The weight scaling concept, which was explained in Section 6.3 appears throughout the network. All convolutional layers in the backbone and refinement stages are implemented as "ScaledConv2D". The activation functions (non-linearities) are implemented as "ScaledLeakyReLU". Each spatial upscaling step (i.e.: 128×128 block, 256×256 block, and 512×512 block) uses "UpsamplerConv2D", which denotes the "Scaled Transposed Convolution for Upsampling", as defined in Section 6.4.3.

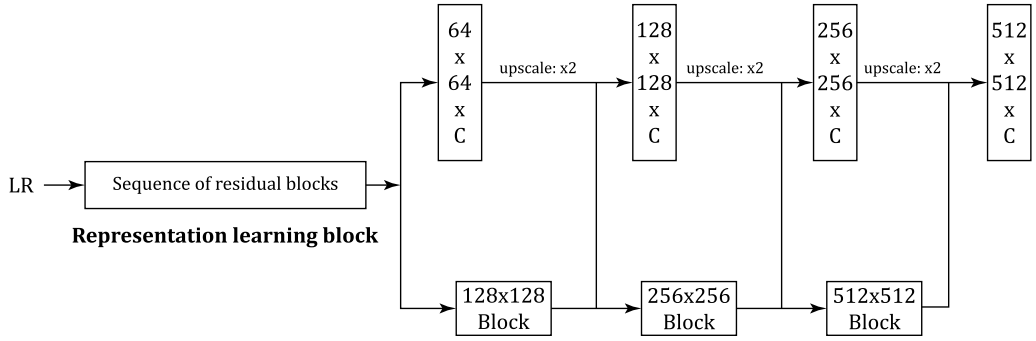

Figure 6.2. Overall architecture of the proposed SR CNN. The "representation learning block" extracts deep features at 64×64. Then, three successive upsampling blocks increase spatial resolution by $\times 2$ each time (to 128, 256, and 512). At each scale, an image estimate with C channels is formed, and a skip-image pathway upsamples and adds the previous estimate to enable progressive refinement. $C = 1$ denotes that in the implemented experiments, the LR and ground-truth HR images are grayscale)

6.5.3 Backbone and feature extraction blocks

The representation learning block operates at the LR resolution (64×64) and aims to learn rich features before any upsampling is performed. As shown in Figure 6.3, it is implemented as a sequence of 5 residual blocks (shown as "RB"). Each residual block (RB) increases the channel depth while preserving spatial size. So, during the "representation learning" stage, we do not increase the spatial size of the LR input images. We only increase the depth of channels progressively over 5 residual blocks.

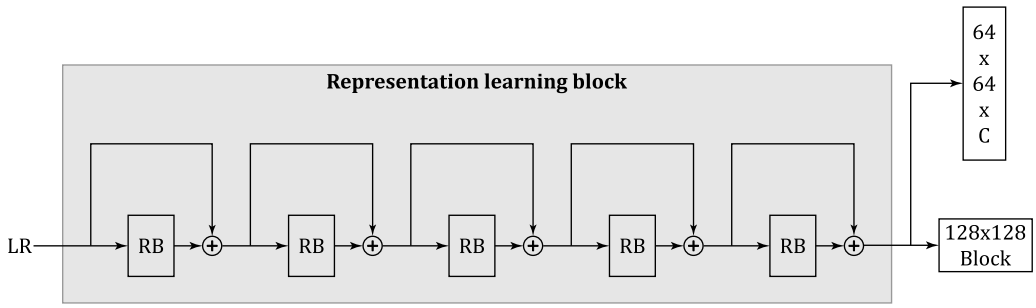

Figure 6.3. Representation learning block. A sequence of 5 residual blocks extracts features at fixed resolution (64×64). The output is forwarded to the first upsampling block (i.e. 128×128 block), and simultaneously an image prediction branch produces an intermediate $64 \times 64 \times C$ estimate. In the implemented experiments, $C = 1$.

Residual blocks and channel progression

Each residual block (denoted by RB in Fig. 6.3)contains: (i) a "main path" with two ScaledConv2D with 3×3 kernel size, where each of them are followed by ScaledLeakyReLU, and (ii) a projection skip path that uses a ScaledConv2D with 1×1 kernel size followed by ScaledLeakyReLU to match the channel dimension before addition.

Figures 6.4 and 6.5 show the first two residual blocks as examples.

The first block (Fig. 6.4) maps $64 \times 64 \times C \rightarrow 64 \times 64 \times 32$ (where $C = 1$ for grayscale input images and $C = 3$ for RGB inputs). The second residual block (Fig. 6.5) maps $64 \times 64 \times 32 \rightarrow 64 \times 64 \times 64$.

The third and fourth residual blocks follow the same pattern, doubling channels each time: $64 \rightarrow 128 \rightarrow 256$.

The fifth (last) residual block (shown in Fig. 6.6) maps $64 \times 64 \times 256 \rightarrow 64 \times 64 \times 512$, and in addition, it makes a branch to produce an initial image estimate at LR resolution 64×64.

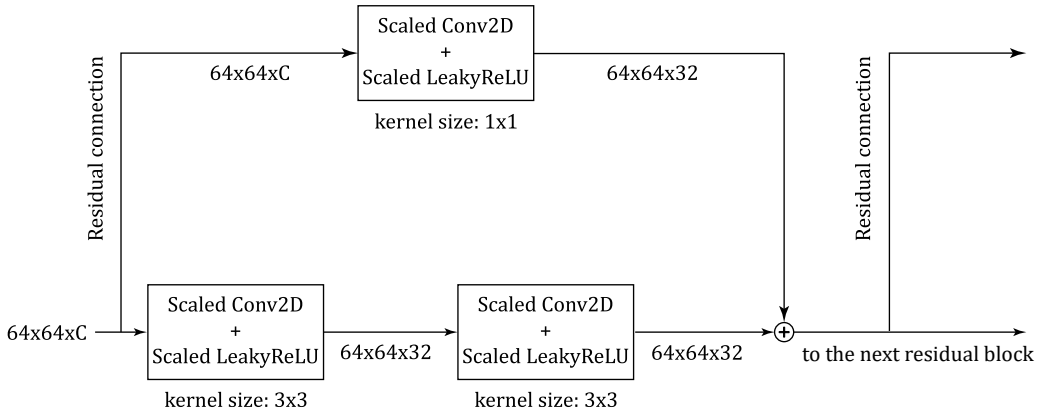

Figure 6.4. The first residual block (Block 1): $64 \times 64 \times C \rightarrow 64 \times 64 \times 32$. The main path uses two 3×3 ScaledConv2D layers (each followed by ScaledLeakyReLU). The residual path uses a 1×1 ScaledConv2D (followed by ScaledLeakyReLU) to match channel dimensions for summation.

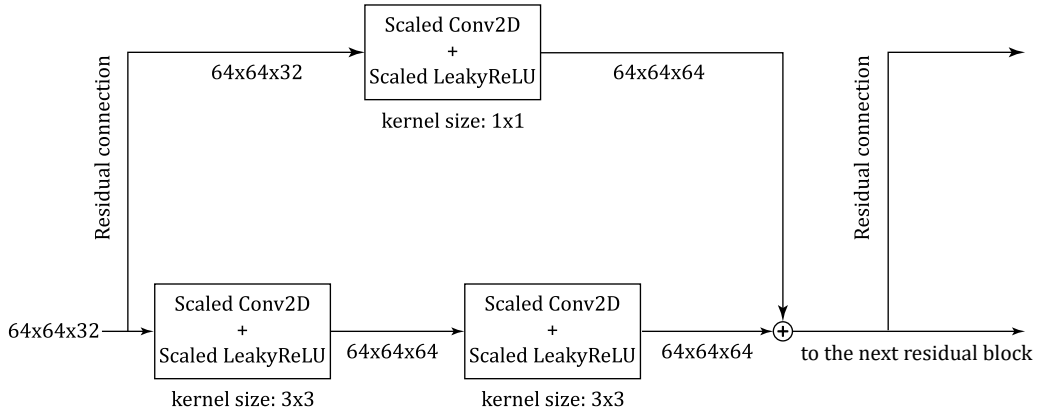

Figure 6.5. The second residual block (Block 2): $64 \times 64 \times 32 \rightarrow 64 \times 64 \times 64$. The structure is identical to Block 1 and only the channel sizes differ.

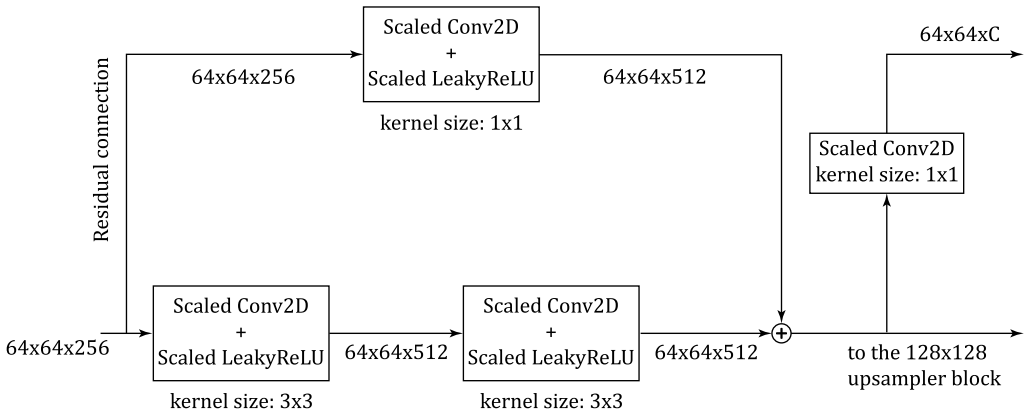

Figure 6.6. The fifth (last) residual block (Block 5): $64 \times 64 \times 256 \rightarrow 64 \times 64 \times 512$. It forwards features to the first upsampling stage (128×128 upsampler block). In addition, a 1×1 ScaledConv2D branch produces an intermediate image estimate of size $64 \times 64 \times C$.

Receptive field and kernel-size rationale

The backbone uses repeated 3×3 convolutions because they provide a good trade-off between expressive power and efficiency, while increasing the receptive field gradually through depth.

Since the backbone includes 5 residual blocks and each block contains two 3×3 convolutions on the main path, the network can aggregate contextual information from a neighborhood substantially larger than 3×3 while still operating at LR. This is useful in SR for medical imaging, where local anatomical patterns are often ambiguous in isolation and require broader context to resolve reliably. The 1×1 convolutions in skip paths serve a different purpose: they align channel dimensions for residual addition and allow controlled feature re-weighting without changing spatial support.

6.5.4 Upsampling strategy for scale factors $\times 2$, $\times 4$, and $\times 8$

The overall scale factor is $s = 8$, achieved through three consecutive $\times 2$ stages: $64 \rightarrow 128 \rightarrow 256 \rightarrow 512$.

Each upsampling stage consists of an upsampling plus refinement block that:

1. increases spatial resolution by $\times 2$ using "UpsamplerConv2D" (scaled transposed convolution for upsampling),
2. reduces the channel depth by a factor of 2 (e.g., $512 \rightarrow 256$ at the first upsampling stage),
3. refines features with a 3×3 ScaledConv2D + ScaledLeakyReLU,
4. produces an image estimate at that scale using a 1×1 ScaledConv2D that maps features to C channels,

5. and adds a skip-image term, which is created by upsampling the previous (coarser) image estimate.

Figure 6.7 shows the 128×128 block in detail. The 256×256 and 512×512 blocks follow the "same" topology and operations; only spatial sizes and channel counts differ.

The procedure continues in the same manner from 128 to 512, where each upsampler block doubles spatial resolution and reduces the number of feature channels by a half.

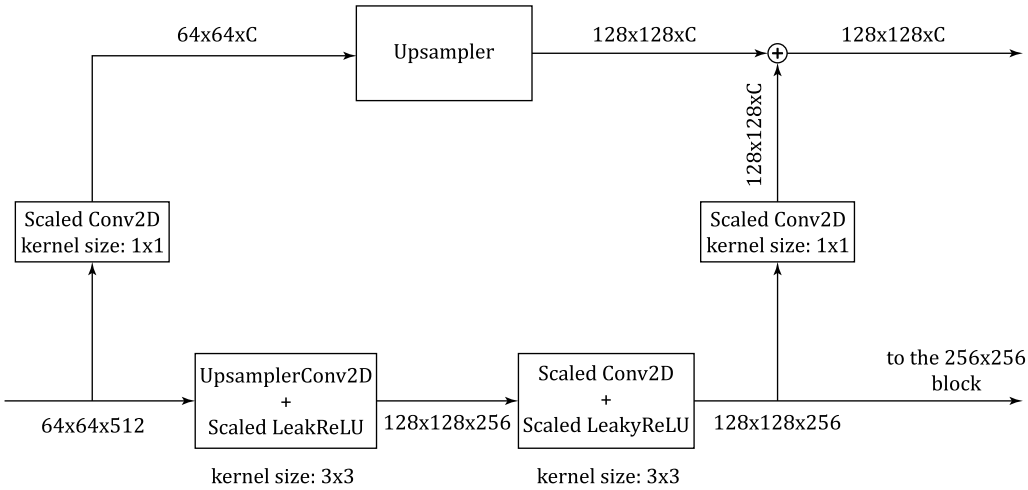

Figure 6.7. 128×128 upsampling/refinement block. The main feature stream upsamples $64 \times 64 \times 512 \to 128 \times 128 \times 256$ via UpsamplerConv2D (scaled transposed convolution for upsampling), followed by ScaledLeakyReLU and a 3×3 ScaledConv2D refinement. A 1×1 ScaledConv2D maps refined features to an image estimate $128 \times 128 \times C$. In parallel, the previous image estimate ($64 \times 64 \times C$) is upsampled and added which results in progressive image refinement.

Progressive image refinement (image-level skip connections). At each scale, the network produces an image estimate and then adds the upsampled image estimate from the previous scale. Conceptually, if $\widehat{\mathbf{X}}_{64}$ denotes the first $64 \times 64 \times C$ estimate produced after the backbone, then:

$$\begin{aligned}
\widehat{\mathbf{X}}_{128} &= g_{128}(\mathbf{F}_{128}) + \mathrm{Up}(\widehat{\mathbf{X}}_{64}), \\
\widehat{\mathbf{X}}_{256} &= g_{256}(\mathbf{F}_{256}) + \mathrm{Up}(\widehat{\mathbf{X}}_{128}), \\
\widehat{\mathbf{X}}_{512} &= g_{512}(\mathbf{F}_{512}) + \mathrm{Up}(\widehat{\mathbf{X}}_{256}),
\end{aligned} \tag{6.29}$$

where $\mathbf{F}_{128}, \mathbf{F}_{256}, \mathbf{F}_{512}$ denote the feature maps at each scale, $g_\cdot(\cdot)$ is the 1×1 image projection to C channels, and $\mathrm{Up}(\cdot)$ denotes $\times 2$ upsampling in the image branch. This design encourages coarse-to-fine reconstruction: coarse structures are formed early and then progressively refined with higher-frequency details at later stages.

6.5.5 Output layer and intensity handling

The final SR output is the highest-resolution image estimate:

$$\widehat{\mathbf{X}} \equiv \widehat{\mathbf{X}}_{512} \in \mathbb{R}^{512\times512\times C}. \tag{6.30}$$

In the last upsampling stage, the model uses a 1×1 ScaledConv2D projection from HR features to C output channels, followed by the progressive skip-image addition described above. This final projection plays the role of the reconstruction layer: it converts learned features into pixel intensities in the target image space.

To remain consistent with typical preprocessing used for CT (and with common normalized image pipelines), inputs and targets are assumed to lie in a bounded intensity range (e.g., after normalization). Therefore, the network output can be constrained to the valid range by applying a final clipping (or equivalently, a bounded activation), for example:

$$\widehat{\mathbf{X}} \leftarrow \mathrm{clip}(\widehat{\mathbf{X}}, 0, 1). \tag{6.31}$$

This prevents unrealistic out-of-range values and promotes stable intensity behavior. In practice, the exact normalization/windowing choice is determined by the dataset preprocessing described later in the training section; the architectural design remains unchanged for grayscale or multi-channel images.

6.6 Loss functions

The main goal is to improve the perceptual quality of super-resolved images beyond regression-only objectives while preserving structural integrity (especially important in CT). In our setting, the network produces a single final SR output

$$\widehat{\mathbf{X}} = f_\theta(\mathbf{Y}_{\mathrm{LR}}) \in \mathbb{R}^{512\times512\times C}, \tag{6.32}$$

and the training objective is defined "only" between $\widehat{\mathbf{X}}$ and the HR ground truth $\mathbf{X}_{\mathrm{HR}}$, which means we do not perfrom multi-scale supervision.

We define the overall training loss as a weighted combination of complementary terms:

$$\begin{aligned}\mathcal{L}_{\mathrm{total}}(\widehat{\mathbf{X}}, \mathbf{X}_{\mathrm{HR}}) = {} & \lambda_{\mathrm{pix}}\, \mathcal{L}_{\mathrm{pix}}(\widehat{\mathbf{X}}, \mathbf{X}_{\mathrm{HR}}) \\ & + \lambda_{\mathrm{per}}\, \mathcal{L}_{\mathrm{per}}(\widehat{\mathbf{X}}, \mathbf{X}_{\mathrm{HR}}) \\ & + \lambda_{\mathrm{grad}}\, \mathcal{L}_{\mathrm{grad}}(\widehat{\mathbf{X}}, \mathbf{X}_{\mathrm{HR}}) \\ & + \lambda_{\mathrm{freq}}\, \mathcal{L}_{\mathrm{freq}}(\widehat{\mathbf{X}}, \mathbf{X}_{\mathrm{HR}}) \\ & + \lambda_{\mathrm{style}}\, \mathcal{L}_{\mathrm{style}}(\widehat{\mathbf{X}}, \mathbf{X}_{\mathrm{HR}}).\end{aligned} \tag{6.33}$$

In the remainder of this section, each term is defined in detail.

6.6.1 Robust regression loss (Charbonnier)

Pixel-wise regression encourages faithful reconstruction of the HR target, but standard ℓ_1 or ℓ_2 losses can be sensitive to outliers (e.g., occasional extreme residuals caused by noise, sharp edges, or imperfect LR synthesis). We therefore use the *Charbonnier loss*, which is a smooth, robust alternative to ℓ_1:

$$\rho_\epsilon(r) = \sqrt{r^2 + \epsilon^2}, \tag{6.34}$$

where $\epsilon > 0$ is a small constant.

Let $N = HWC$ denote the number of scalar intensity values (with $H = W = 512$ at training time). The pixel loss is:

$$\mathcal{L}_{\text{pix}}(\widehat{\mathbf{X}}, \mathbf{X}_{\text{HR}}) = \frac{1}{N} \sum_{i=1}^{N} \rho_\epsilon(\widehat{x}_i - x_i)\,, \tag{6.35}$$

where $\widehat{x}_i$ and x_i are the i-th scalar entries of $\widehat{\mathbf{X}}$ and $\mathbf{X}_{\text{HR}}$, respectively.

Choice of ϵ. For images normalized to $[0, 1]$, a typical and effective choice is:

$$\epsilon = 10^{-3}. \tag{6.36}$$

This keeps the loss close to ℓ_1 for moderate residuals while remaining differentiable at zero.

The Charbonnier penalty in (6.34) can be interpreted as a smooth approximation of the absolute value $|r|$. Its behavior becomes clear by analyzing two cases:

Small residuals: $|r| \ll \epsilon$ (quadratic / ℓ_2-like). Using the Taylor expansion $\sqrt{1+t} \approx 1 + \frac{t}{2}$ for $|t| \ll 1$, we write

$$\rho_\epsilon(r) = \epsilon\sqrt{1 + \left(\frac{r}{\epsilon}\right)^2} \approx \epsilon\left(1 + \frac{1}{2}\left(\frac{r}{\epsilon}\right)^2\right) = \epsilon + \frac{r^2}{2\epsilon}.$$

Up to the additive constant ϵ (which does not affect optimization), the penalty behaves like a scaled ℓ_2 term:

$$\rho_\epsilon(r) \approx \frac{r^2}{2\epsilon} \quad \text{for } |r| \ll \epsilon.$$

Therefore, for very small errors, Charbonnier acts *quadratically* and encourages precise fitting.

Large residuals: $|r| \gg \epsilon$ (linear / ℓ_1-like). Factor out $|r|$:

$$\rho_\epsilon(r) = |r|\sqrt{1 + \left(\frac{\epsilon}{r}\right)^2}.$$

When $|r| \gg \epsilon$, we have $\left(\frac{\epsilon}{r}\right)^2 \approx 0$, hence

$$\rho_\epsilon(r) \approx |r| \quad \text{for } |r| \gg \epsilon.$$

Thus, for large errors, Charbonnier behaves *linearly* like an ℓ_1 loss, which is why it is more robust to outliers than ℓ_2.

Unlike ℓ_1 (which is not differentiable at $r = 0$), the Charbonnier penalty is differentiable everywhere because $\epsilon > 0$. Its derivative is

$$\frac{d}{dr}\rho_\epsilon(r) = \frac{r}{\sqrt{r^2 + \epsilon^2}}.$$

At $r = 0$,

$$\frac{d}{dr}\rho_\epsilon(r)\Big|_{r=0} = 0,$$

so the non-differentiability of $|r|$ at the origin is removed. Moreover, the derivative smoothly transitions:

$$\frac{d}{dr}\rho_\epsilon(r) \approx \frac{r}{\epsilon} \quad (|r| \ll \epsilon), \qquad \frac{d}{dr}\rho_\epsilon(r) \approx \text{sign}(r) \quad (|r| \gg \epsilon),$$

showing that Charbonnier interpolates between ℓ_2-like behavior near zero and ℓ_1-like behavior for large residuals.

6.6.2 Perceptual loss

Pixel-wise losses (including robust variants such as Charbonnier) encourage accurate intensity reconstruction, but they often under-penalize perceptually important differences, especially in high-frequency structures. To reduce over-smoothing and promote visually faithful details while preserving structural integrity, we introduce a "perceptual loss" computed in the feature space of a "*fixed pretrained network*".

We choose pretrained RadImageNet [39] model as the feature extractor and we keep it "*frozen*" during SR training. It means that the weights of the RadImageNet should not be updated during the process of feature extraction.

RadImageNet is a large-scale image database created to support transfer learning in medical imaging. It contains about 1.35 million radiology images including CT and MRI images.

The RadImageNet provides pretrained weights for known CNN architectures including ResNet-50, trained only on RadImageNet dataset for transfer learning use.

Feature extractor. Let $\phi(\cdot)$ denote a RadImageNet-pretrained "ResNet-50" feature network. We choose ResNet-50 [21] because it is widely used, stable, and provides a hierarchy of feature representations (from low-level edge/contrast features to higher-level structural features).

The network $\phi(\cdot)$ is used only for feature computation inside the loss and does not affect the forward inference of the SR model.

For an input image $\mathbf{X}$, let $\phi_\ell(\mathbf{X})$ denote the activation (feature map) extracted at layer ℓ. We write its shape as

$$\phi_\ell(\mathbf{X}) \in \mathbb{R}^{C_\ell \times H_\ell \times W_\ell},$$

where C_ℓ is the number of channels and (H_ℓ, W_ℓ) are the spatial dimensions at layer ℓ.

Input channel handling for grayscale CT. RadImageNet backbones are typically trained with three-channel inputs. Lung CT slices are single-channel ($C = 1$). We therefore define a deterministic channel adaptation operator $\mathcal{R}(\cdot)$ that replicates the grayscale image across three channels:

$$\begin{aligned} \mathcal{R} : \mathbb{R}^{H \times W \times 1} &\to \mathbb{R}^{H \times W \times 3}, \\ \mathcal{R}(\mathbf{X}) &= [\mathbf{X}, \mathbf{X}, \mathbf{X}], \end{aligned} \tag{6.37}$$

where $[\cdot, \cdot, \cdot]$ denotes concatenation along the channel dimension. This operation preserves all spatial structures exactly and only adapts the tensor shape to the expected input format of $\phi(\cdot)$. For multi-channel images (e.g., RGB with $C = 3$), $\mathcal{R}(\cdot)$ is simply the identity mapping.

Preprocessing and normalization. Because pretrained feature networks expect a specific intensity normalization, the perceptual loss must specify a deterministic preprocessing transform applied identically to the SR output and the HR target.

Let images be normalized to the range $[0, 1]$ before loss computation (as in our data pipeline). We define the complete RadImageNet preprocessing transform as:

$$T(\mathbf{X}) = \mathcal{N}(\mathcal{R}(\mathbf{X})), \tag{6.38}$$

where $\mathcal{R}(\cdot)$ is the channel replication in (6.37), and $\mathcal{N}(\cdot)$ is channel-wise mean–standard-deviation normalization:

$$\mathcal{N}(\mathbf{U})_c = \frac{\mathbf{U}_c - \mu_c}{\sigma_c}, \qquad c \in \{1, 2, 3\}, \tag{6.39}$$

with fixed constants $\{\mu_c, \sigma_c\}$ determined by the RadImageNet model training protocol. In practice, these constants are taken from the released RadImageNet feature model configuration and must remain fixed during SR training.

Using this notation, we define the pre-processed HR and SR images for the feature extractor as:

$$\mathbf{Z} = T(\mathbf{X}_{\mathrm{HR}}), \qquad \widehat{\mathbf{Z}} = T(\widehat{\mathbf{X}}). \tag{6.40}$$

This ensures that both $\widehat{\mathbf{X}}$ and $\mathbf{X}_{\mathrm{HR}}$ undergo the same channel handling and the same normalization before feature extraction.

Choice of perceptual layers. ResNet-50 provides a natural set of feature stages corresponding to increasing receptive fields. To emphasize structural fidelity while avoiding excessive encouragement of high-level "texture hallucination", we select a set of intermediate layers:

$$\mathcal{P} = \{\texttt{conv1}, \texttt{layer1}, \texttt{layer2}, \texttt{layer3}\}. \tag{6.41}$$

`conv1` and `layer1` focus on low-level features such as edges, contrast transitions, local patterns. `layer2` and `layer3` incorporate broader context and mid-level structures, while the final stage (`layer4`) is intentionally excluded to reduce sensitivity to very high-level semantics that may be less appropriate for CT slice texture realism.

This choice provides a balanced perceptual constraint that improves sharpness and visual fidelity while remaining conservative with respect to hallucination risk.

Perceptual distance. For each selected layer $\ell \in \mathcal{P}$, we compare feature maps of the SR and HR images using a normalized ℓ_1 distance.

The perceptual loss is defined as:

$$\mathcal{L}_{\text{per}}(\widehat{\mathbf{X}}, \mathbf{X}_{\text{HR}}) = \sum_{\ell \in \mathcal{P}} w_\ell \frac{1}{C_\ell H_\ell W_\ell} \left\| \phi_\ell(\widehat{\mathbf{Z}}) - \phi_\ell(\mathbf{Z}) \right\|_1, \tag{6.42}$$

where $\|\cdot\|_1$ denotes the sum of absolute values over all entries of the feature tensor.

To avoid introducing unnecessary tuning complexity and to keep the scale of $\mathcal{L}_{\text{per}}$ stable, we use equal weighting:

$$w_\ell = \frac{1}{|\mathcal{P}|}, \qquad \ell \in \mathcal{P}. \tag{6.43}$$

This means that each selected feature level contributes the same proportion to the perceptual loss, i.e., no layer is explicitly favored over another. Since feature maps at different depths capture complementary information (from lower-level edges/contrast to higher-level structure), equal weighting provides a simple and robust default that avoids introducing additional hyperparameters. Moreover, dividing by $|\mathcal{P}|$ keeps the overall magnitude of $\mathcal{L}_{\text{per}}$ approximately invariant to the number of selected layers; therefore, if $\mathcal{P}$ is modified (e.g., one layer added or removed), the loss scale does not change abruptly and the same value of λ_{per} remains meaningful.

Why RadImageNet? Standard perceptual losses based on natural-image feature extractors (e.g., ImageNet-trained networks) may not align well with radiological appearance. Using RadImageNet features biases the comparison toward representations learned from medical images, which helps the SR model preserve meaningful structures and textures in terms of radiology.

Perceptual supervision can increase sharpness but may also encourage texture-like patterns when over-weighted. So, the training objective is designed so that fidelity and LR-consistency remain primary, while RadImageNet perceptual features guide the reconstruction toward sharper, more visually faithful detail without destroying the reconstruction process with unrealistic structures.

6.6.3 Edge and gradient preservation loss

To explicitly preserve sharp boundaries and reduce edge blurring, we penalize mismatches in image gradients. Let ∇_x and ∇_y denote discrete horizontal and vertical gradient operators. A standard choice is the Sobel operator. The Sobel kernels are:

$$\mathbf{S}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \qquad \mathbf{S}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \tag{6.44}$$

We define:

$$\nabla_x(\mathbf{X}) = \mathbf{X} * \mathbf{S}_x, \qquad \nabla_y(\mathbf{X}) = \mathbf{X} * \mathbf{S}_y, \tag{6.45}$$

where $*$ denotes 2D convolution applied channel-wise.

The gradient loss is then:

$$\mathcal{L}_{\text{grad}}(\widehat{\mathbf{X}}, \mathbf{X}_{\text{HR}}) = \frac{1}{N}\left(\left\|\nabla_x(\widehat{\mathbf{X}}) - \nabla_x(\mathbf{X}_{\text{HR}})\right\|_1 + \left\|\nabla_y(\widehat{\mathbf{X}}) - \nabla_y(\mathbf{X}_{\text{HR}})\right\|_1\right). \tag{6.46}$$

Pixel losses can fit intensities but still blur edges because many blurred solutions have similar pixel-wise cost. Enforcing gradient similarity constrains the model to reproduce correct transitions and boundaries, which is important for CT structures.

6.6.4 Frequency-domain alignment loss

Pixel/gradient losses operate in the spatial domain. However, SR artifacts and over-smoothing also have clear frequency signatures: regression-only models often exhibit "high-frequency roll-off", i.e., attenuated mid-to-high frequency content. To directly correct spectral mismatches without amplifying pure noise, we introduce a band-limited frequency-domain.

Let $\mathcal{F}(\cdot)$ denote the 2D discrete Fourier transform applied channel-wise, and let $|\cdot|$ denote complex magnitude. Let $\mathbf{M}$ be a band-pass mask emphasizing mid-to-high frequencies. We define:

$$\mathcal{L}_{\text{freq}}(\widehat{\mathbf{X}}, \mathbf{X}_{\text{HR}}) = \frac{1}{N}\left\|\mathbf{M} \odot \left(\left|\mathcal{F}(\widehat{\mathbf{X}})\right| - |\mathcal{F}(\mathbf{X}_{\text{HR}})|\right)\right\|_1, \tag{6.47}$$

where $\odot$ denotes element-wise multiplication in the frequency domain (broadcast over channels).

Let (u, v) index discrete spatial frequencies, and let $r(u, v)$ be the radial frequency distance from the DC component (zero-frequency component of the Fourier transform) after appropriate frequency shifting. Let $r_{\max}$ be the maximum possible radius. A simple hard band-pass mask is:

$$\mathbf{M}(u, v) = \begin{cases} 1, & r_{\text{low}} \le r(u, v) \le r_{\text{high}}, \\ 0, & \text{otherwise}, \end{cases} \tag{6.48}$$

with:

$$r_{\text{low}} = 0.10\, r_{\max}, \qquad r_{\text{high}} = 0.60\, r_{\max}. \tag{6.49}$$

A smooth mask (cosine ramps near boundaries) may further reduce ringing. However, the core idea is the same: focus the penalty on frequency bands where over-smoothing typically removes content, while avoiding over-penalizing the highest frequencies that are most noise-sensitive.

6.6.5 Style/texture matching loss

Perceptual losses align deep features point-wise, but they do not explicitly match "texture statistics". To encourage realistic local texture patterns, we include a style/texture loss based on Gram matrices of selected feature maps.

Let $\phi_\ell(\cdot) \in \mathbb{R}^{C_\ell \times H_\ell \times W_\ell}$ be the feature map at layer ℓ (from pretrained RadImageNet feature extractor). Define the reshaping:

$$\Phi_\ell(\mathbf{X}) \in \mathbb{R}^{C_\ell \times (H_\ell W_\ell)}, \tag{6.50}$$

obtained by flattening spatial dimensions. The Gram matrix at layer ℓ is:

$$\mathbf{G}_\ell(\mathbf{X}) = \frac{1}{H_\ell W_\ell} \Phi_\ell(\mathbf{X})\, \Phi_\ell(\mathbf{X})^\top \in \mathbb{R}^{C_\ell \times C_\ell}. \tag{6.51}$$

Let $\mathcal{S}$ denote the set of layers used for style statistics. The style loss is:

$$\mathcal{L}_{\text{style}}(\widehat{\mathbf{X}}, \mathbf{X}_{\text{HR}}) = \sum_{\ell \in \mathcal{S}} \alpha_\ell \left\| \mathbf{G}_\ell(\widehat{\mathbf{Z}}) - \mathbf{G}_\ell(\mathbf{Z}) \right\|_1, \tag{6.52}$$

where $\widehat{\mathbf{Z}} = T(\widehat{\mathbf{X}})$ and $\mathbf{Z} = T(\mathbf{X}_{\text{HR}})$ are the preprocessed inputs to the feature extractor, and α_ℓ are non-negative weights.

We choose the layer weighting as follows:

$$\alpha_\ell = \frac{1}{|\mathcal{S}|}, \qquad \ell \in \mathcal{S}. \tag{6.53}$$

The Gram matrix captures correlations between feature channels, which act as a proxy for texture statistics. Minimizing Gram differences encourages the SR output to reproduce texture-like patterns present in the HR target and helps perceptual realism without relying on adversarial training.

6.6.6 Unified objective

The complete objective is given in (6.33). In our implementation, we use the following weight setting. It should be noted that the images are normalized to $[0, 1]$ and the perceptual/style networks are frozen:

$$\lambda_{\text{pix}} = 1.0, \quad \lambda_{\text{per}} = 0.10, \quad \lambda_{\text{grad}} = 0.05, \quad \lambda_{\text{freq}} = 0.01, \quad \lambda_{\text{style}} = 0.001. \tag{6.54}$$

These values support the main goal:

- a dominant robust regression term (λ_{pix}) to maintain intensity fidelity and reduce hallucinations,
- moderate perceptual and gradient terms to improve sharpness and preserve boundaries,
- a small frequency term to correct spectral roll-off while avoiding noise amplification,
- and a very small style term to enhance texture statistics without overpowering structural accuracy.

Finally, the training problem can be written as:

$$\theta^* = \arg\min_{\theta} \; \mathbb{E}_{(\mathbf{Y}_{\text{LR}}, \mathbf{X}_{\text{HR}})} \left[\mathcal{L}_{\text{total}} \left(f_\theta(\mathbf{Y}_{\text{LR}}), \mathbf{X}_{\text{HR}} \right) \right]. \tag{6.55}$$

6.7 Training

The emphasis is on a stable training, consistent with the degradation model in Section 6.2.2 and the loss design in Section 6.6.

Only the final SR output at $512 \times 512 \times C$ is supervised during training and intermediate image estimates are not directly supervised by the loss.

6.7.1 Training configuration

Data preparation and normalization

HR target preparation. Each HR CT slice is treated as an HR image $\mathbf{X}_{\text{HR}} \in \mathbb{R}^{512\times512\times1}$ (for grayscale lung CT images). A consistent intensity normalization is applied so that values lie in a bounded range (e.g., $[0, 1]$). This is typically achieved by applying a fixed CT window (window level/width) followed by linear scaling to $[0, 1]$ and clipping.

LR synthesis. For each HR sample, the LR input $\mathbf{Y}_{\text{LR}} \in \mathbb{R}^{64\times64\times1}$ is synthesized using the degradation pipeline:

$$\mathbf{Y}_{\text{LR}} = \mathcal{D}(\mathbf{X}_{\text{HR}}),$$

which includes probabilistic Gaussian blur, bicubic downsampling by $s = 8$, additive Gaussian noise, and clipping to $[0, 1]$ (Section 6.2.2). For concrete training values, we use:

- blur probability: $p_{\text{blur}} = 0.5$,
- Gaussian blur kernel size: $k = 7$,
- blur standard deviation: $\sigma \sim \mathcal{U}(0.2, 1.2)$,
- additive noise: $n \sim \mathcal{N}(0, \sigma_n^2)$ with $\sigma_n \sim \mathcal{U}(0, 0.01)$.

These values are commonly used to improve robustness without severely corrupting medical structure.

Patch-based training

Training on full 512×512 images is very memory-intensive. We therefore train on paired patches:

- HR patch size: $P_{\text{HR}} = 256$ (i.e., 256×256),
- LR patch size: $P_{\text{LR}} = P_{\text{HR}}/s = 32$ (i.e., 32×32 for $s = 8$).

Patches are randomly sampled from the HR image, and the corresponding LR patch is obtained by applying the degradation model consistently. Patch-based training increases the number of distinct samples per epoch and improves generalization.

Data augmentation

To improve invariance to orientation and to reduce overfitting, we use light geometric augmentations that preserve medical plausibility:

- random horizontal flip with probability 0.5,
- random vertical flip with probability 0.5,
- random rotation by multiples of 90° (i.e., $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$).

Optimization settings

Optimizer. We use the Adam optimizer (first-order adaptive optimization), with:

$$\beta_1 = 0.9, \qquad \beta_2 = 0.999, \qquad \epsilon_{\text{adam}} = 10^{-8}.$$

Adam is a robust default for deep SR training, especially when multiple loss terms are combined.

Learning rate schedule. We use an initial learning rate

$$\eta_0 = 2 \times 10^{-4},$$

with cosine decay to a small final value (10^{-6}) over training. Cosine decay reduces oscillations near convergence and tends to work well for perceptual objectives.

As we stated before, just for recalling, the total loss is defined as in (6.33), with:

$$\lambda_{\text{pix}} = 1.0, \quad \lambda_{\text{per}} = 0.10, \quad \lambda_{\text{grad}} = 0.05, \quad \lambda_{\text{freq}} = 0.01, \quad \lambda_{\text{style}} = 0.001.$$

These values prioritize intensity fidelity (important to avoid hallucinations) while still improving perceived sharpness and texture realism.

6.7.2 Training algorithm (final-output supervision only)

Algorithm 8 summarizes the complete training procedure.

Algorithm 8 Training procedure for the proposed SR model (final output supervision only)

Require: Training set of HR images $\{\mathbf{X}_{\mathrm{HR}}^{(i)}\}_{i=1}^{N}$; scale factor $s = 8$; degradation operator $\mathcal{D}(\cdot)$; SR model $f_\theta(\cdot)$; frozen RadImageNet feature extractor $\phi(\cdot)$; loss weights $\lambda_{\mathrm{pix}}, \lambda_{\mathrm{per}}, \lambda_{\mathrm{grad}}, \lambda_{\mathrm{freq}}, \lambda_{\mathrm{style}}$; optimizer parameters.

Initialize SR model parameters θ (scaling-aware layers as in Sections 6.3–6.4).
for training step $t = 1$ to T **do**
  Sample a mini-batch of HR images $\{\mathbf{X}_{\mathrm{HR}}^{(i)}\}_{i=1}^{B}$.
  Extract random HR patches $\{\mathbf{X}_{\mathrm{HR,patch}}^{(i)}\}_{i=1}^{B}$ of size $P_{\mathrm{HR}} \times P_{\mathrm{HR}}$.
  Apply geometric augmentations (random flips and 90° rotations) to each $\mathbf{X}_{\mathrm{HR,patch}}^{(i)}$.
  Synthesize LR inputs by degradation:
  $\mathbf{Y}_{\mathrm{LR}}^{(i)} \leftarrow \mathcal{D}\left(\mathbf{X}_{\mathrm{HR,patch}}^{(i)}\right)$, producing LR patches of size $P_{\mathrm{LR}} \times P_{\mathrm{LR}}$.
  Forward pass through SR model:
  $\widehat{\mathbf{X}}^{(i)} \leftarrow f_\theta\left(\mathbf{Y}_{\mathrm{LR}}^{(i)}\right)$ ▷ $\widehat{\mathbf{X}}^{(i)}$ has size $P_{\mathrm{HR}} \times P_{\mathrm{HR}}$
  Compute pixel loss:
  $\mathcal{L}_{\mathrm{pix}} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\mathrm{pix}}\left(\widehat{\mathbf{X}}^{(i)}, \mathbf{X}_{\mathrm{HR,patch}}^{(i)}\right)$
  Compute perceptual loss (RadImageNet features, frozen ϕ):
  $\mathcal{L}_{\mathrm{per}} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\mathrm{per}}\left(\widehat{\mathbf{X}}^{(i)}, \mathbf{X}_{\mathrm{HR,patch}}^{(i)}\right)$
  Compute gradient loss:
  $\mathcal{L}_{\mathrm{grad}} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\mathrm{grad}}\left(\widehat{\mathbf{X}}^{(i)}, \mathbf{X}_{\mathrm{HR,patch}}^{(i)}\right)$
  Compute frequency loss:
  $\mathcal{L}_{\mathrm{freq}} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\mathrm{freq}}\left(\widehat{\mathbf{X}}^{(i)}, \mathbf{X}_{\mathrm{HR,patch}}^{(i)}\right)$
  Compute style loss (Gram statistics from frozen ϕ):
  $\mathcal{L}_{\mathrm{style}} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\mathrm{style}}\left(\widehat{\mathbf{X}}^{(i)}, \mathbf{X}_{\mathrm{HR,patch}}^{(i)}\right)$
  Combine into total loss:
  $\mathcal{L}_{\mathrm{total}} \leftarrow \lambda_{\mathrm{pix}}\mathcal{L}_{\mathrm{pix}} + \lambda_{\mathrm{per}}\mathcal{L}_{\mathrm{per}} + \lambda_{\mathrm{grad}}\mathcal{L}_{\mathrm{grad}} + \lambda_{\mathrm{freq}}\mathcal{L}_{\mathrm{freq}} + \lambda_{\mathrm{style}}\mathcal{L}_{\mathrm{style}}$
  Compute gradients of $\mathcal{L}_{\mathrm{total}}$ w.r.t. θ.
  Update parameters θ using Adam.
end for
return trained parameters θ^*.

6.7.3 Practical trade-offs

The proposed SR-CNN provides a deterministic GAN-free alternative for improving perceptual quality through complementary loss terms. The main strength of this design is that the Charbonnier, perceptual, gradient, frequency-domain, and style losses target different reconstruction properties: pixel fidelity, structural similarity, edge preservation, spectral consistency, and texture statistics. This allows the model to reduce the over-smoothing typical of purely pixel-wise regression while avoiding adversarial training instability. However, the use of multiple losses also introduces additional training complexity, because the relative weights of the loss terms must be chosen carefully. If perceptual, frequency, or style losses are over-weighted, they may compete with the pixel-domain fidelity term and encourage visually sharper but less faithful reconstructions. For this reason, the implementation keeps the pixel loss as the dominant term and uses the other losses as auxiliary regularizers.

It should also be noted that the model remains a deterministic regression-based SR method. Therefore, it produces a single HR estimate for each LR input and does not explicitly sample from the full multi-modal posterior of possible HR reconstructions. Nevertheless, in medical imaging this deterministic behavior is also a strength, because it reduces the risk of stochastic hallucinated structures. Finally, the model depends on a pretrained RadImageNet feature extractor. Although RadImageNet features are more domain-relevant than generic natural-image features, and the degradation model includes blur and noise to improve robustness, generalization to other scanners may require further validation or domain-specific tuning.

Chapter 7

Discussion

7.1 Chapter overview and scope

This chapter provides the comparative discussion of the SR models developed and evaluated throughout this thesis. In particular, the discussion aims to explain how the observed differences between models arise from their underlying design choices, training objectives, and why these differences matter in the setting of medical image SR.

The central problem addressed by this thesis is SISR in a fidelity-sensitive domain like medical imaging. The main problem is to improve image detail and perceptual quality without sacrificing structural reliability, deterministic behavior, or training stability. This is especially important in medical imaging, where visually attractive reconstructions are not sufficient if they hallucinate.

The thesis investigates whether regression-based SR models can be pushed beyond the usual limitations of over-smoothing, while avoiding the instability and artifact risks that are commonly associated with adversarial generative approaches.

This discussion presents the models considered: the strong residual regression baseline EDSR [35], the adversarial baseline SRGAN [33], the proposed TRAE model, its multi-resolution extension MR-TRAE, and the SR-CNN developed in Chapter 6. These models are representatives of different design philosophies within modern SR.

This chapter examines quantitative performance, including the behavior of PSNR, SSIM, and LPIPS across different models and upscaling factors. Second, it considers qualitative reconstruction behavior. Third, it analyzes robustness to degradation assumptions, because model behavior changes when evaluation moves beyond idealized bicubic downsampling to more diverse and realistic LR generation. Fourth, it addresses practical considerations, such as parameter efficiency, training stability, and the trade-off between perceptual and structural considerations.

In SR, higher PSNR does not imply better perceptual realism. Visually sharper outputs do not necessarily indicate more faithful reconstruction. Similarly, improvements in LPIPS may reflect perceptual similarity in feature space, but they must

still be judged against the possibility of introduced textures or distortions. These items are particularly significant in medical imaging, where a model that produces aggressive sharpening may appear visually appealing while still being less desirable than a slightly smoother model that preserves anatomy more consistently.

At the same time, the chapter also demonstrates that no single model dominates every criterion, and that the most appropriate method depends on which trade-offs are prioritized.

7.2 Comparative context of the evaluated SR models

The purpose of the comparison is not simply to rank the models by numerical performance, but to interpret them as representatives of different design strategies in SISR.

These proposed models and the baselines were not chosen arbitrarily. Rather, each of them represents a distinct methodological position.

Among these models, EDSR [35] is considered as the "strongest" conventional regression-based baseline. It is one of the most influential residual convolutional networks in the SR literature and is a highly relevant reference because it was explicitly designed to optimize pixel-wise reconstruction performance. By simplifying the residual network formulation for low-level vision, especially through the removal of BN layers and the use of a deep residual feature transformation pipeline, EDSR became a "benchmark architecture" for high-PSNR SR. EDSR is therefore used as the reference model for a high-capacity, distortion-oriented, and deterministic CNN. If a proposed method aims to remain regression-based while improving perceptual behavior, then comparison with EDSR is very important, because EDSR represents the standard in terms of reconstruction accuracy.

However, EDSR also represents a limitation of regression-based training. Its optimization is driven by a pixel-level objective, which is highly effective for maximizing PSNR and usually SSIM. However, this can also encourage smooth reconstructions.

This smoothing tendency is well known in SR. When several plausible high-frequency solutions correspond to the same LR input, a purely regression-based model trained for point-wise fidelity tends to average them. Therefore, EDSR is a crucial baseline not only because of its strength, but also because it clearly shows the central limitation that motivates the developments in the present thesis.

The opposite comparison is SRGAN [33], which represents the well-known adversarial alternative to regression-based SR.

SRGAN is included because it fundamentally changed the SR literature by showing that "visually sharper" and "more realistic" textures could be generated when the objective is shifted from pixel-wise reconstruction toward perceptual one. Instead of optimizing only for signal fidelity, SRGAN combines a content term with an adversarial loss. This encourages its generator to produce outputs that lie on the manifold of realistic SR images. SRGAN provides a reference for a perceptual

strategy that can produce visually sharper results than conventional regression models, especially at larger upscaling factors.

SRGAN also comes with the main concern that motivates caution in medical SR. Adversarial learning highly improves sharpness for sure, but it can also introduce synthetic textures, or visually plausible but inaccurate structures. For natural images, such behavior may be acceptable and desirable. However, in medical imaging, this trade-off is critical, because any introduced artifact may interfere with structural interpretation. For this reason, SRGAN is a perceptual baseline in this chapter. In addition, it is also a conceptual contrast that helps us explain why we emphasize deterministic alternatives even when we explore perceptual enhancement.

Between these two baselines are the main contributions of the thesis. The proposed TRAE model was developed as an attempt to improve the perceptual and structural behavior of regression-based SR without relying on adversarial optimization. Its comparative role in this chapter is therefore particularly important. TRAE is not intended to compete with EDSR only by increasing raw capacity. It is also not for catching GANs by introducing stochastic image synthesis. Instead, its key idea is to enrich deterministic reconstruction with latent guidance and multi-objective supervision.

By using a pretrained and frozen HR-informed representation through the Master branch, TRAE introduces a form of feature-level regularization that is absent in conventional single-loss regression models. This makes TRAE the primary example of a guided regression framework that attempts to improve structural consistency and perceptual plausibility while preserving the determinism.

TRAE tests whether an explicitly guided latent-space constraint can make a regression-based model more structurally coherent than a purely pixel-driven baseline such as EDSR. In addition, it allows us to evaluate whether such guidance can be achieved with fewer trainable parameters than a very large residual model like EDSR, while still remaining competitive in a medical-image setting. TRAE is not simply another architecture to compare; it is one of the central methodological arguments of the thesis made concrete.

MR-TRAE extends this idea into the multi-resolution setting and is included because it addresses another practical limitation of conventional sSR: the need to train separate models for separate scales. Rather than treating each upscaling factor as a fully independent problem, MR-TRAE is designed to produce super-resolved outputs at multiple resolutions within a unified framework. Its significance is therefore not only architectural, but conceptual. It allows us to examine whether multi-scale supervision and progressive resolution modeling can further improve perceptual behavior, structural consistency, or practical efficiency when compared with single-scale approaches. MR-TRAE provides a useful comparison point for interpreting how explicit intermediate supervision may change the balance between distortion metrics and perceptual quality. Since the model is intended as a structured multi-output extension of the TRAE, it occupies a middle position between pure regression baselines and more heavily perception-oriented objectives.

The SR-CNN introduced in Chapter 6 provides another key point. It is still a

regression-based model, but unlike EDSR, it is trained with a rich collection of losses to improve perceptual fidelity and structural realism. This model represents the second strategy for narrowing the gap between conventional regression and perceptual SR.

TRAE moves towards this problem through latent guidance and dual-target supervision, and the SR-CNN in Chapter 6 approaches it through a constructed objective that combines robust content, perceptual, gradient, frequency, and style terms.

TRAE introduces guided latent alignment. MR-TRAE extends this logic to multi-resolution prediction, and the SR-CNN emphasizes perceptual enhancement through a composite objective. As a result, comparing these models against EDSR and SRGAN allows us to examine not just whether one model is better than another, but which design strategy is most effective for balancing sharpness and fidelity.

Our experiments consider both standard bicubic downsampling and more realistic degradation settings that include noise and blur with related perturbations. This is important because some models, such as EDSR, are very strong when the test degradation closely matches the bicubic degradation seen during training. On the other hand, our proposed methods were developed in part to improve robustness under more realistic and clinically relevant degradation.

EDSR is largely aligned with distortion minimization. SRGAN is aligned with perceptual realism, even at the expense of lower PSNR. TRAE and MR-TRAE seek structurally guided deterministic reconstruction. The SR-CNN goes for perceptual improvement without adversarial learning.

These objectives are not identical, so direct numerical comparison must always be interpreted in light of each model's design goal.

A lower PSNR does not necessarily imply a worse super-resolved image if the model was optimized to preserve more perceptually meaningful edges or textures. On the other hand, a sharper output is not superior if that sharpness comes with hallucinated detail.

The question is not simply which model achieves the best score under one metric, but rather which class of design is most suitable for fidelity-sensitive SISR, especially in CT imaging. We should compare a strong conventional regression model (EDSR), an adversarial model (SRGAN), and multiple deterministic models that seek to recover some of the perceptual advantages of modern SR without sacrificing structural trustworthiness.

For this reason, the discussion should be read as an analysis of different SR approaches. EDSR and SRGAN define two reference extremes: one prioritizes stable signal reconstruction, and the other goes for perceptual realism through adversarial learning. TRAE, MR-TRAE, and SR-CNN are in between, where this thesis makes its main contribution. The importance of the comparative results, therefore, lies not only in identifying strengths and weaknesses of individual models, but in clarifying whether the design can offer a more suitable compromise for medical SR.

7.3 EDSR as a strong regression baseline

EDSR [35] provides the most appropriate reference for a strong conventional regression-based baseline. It represents the mature form of distortion-oriented convolutional SR: a deep residual architecture with high representational capacity, stable optimization behavior, and a training objective focused primarily on pixel-level reconstruction accuracy. For this reason, EDSR is a useful benchmark, particularly when the discussion concerns the trade-off between signal fidelity and the more structured or perceptually informed forms of supervision.

A major reason for the strong performance of EDSR is that it is highly specialized for the reconstruction of clean LR inputs under a known degradation model (bicubic downsampling). As discussed earlier, EDSR was originally designed to optimize the residual-network paradigm specifically for low-level vision tasks, and one of its key strengths is that a very large proportion of its capacity is devoted directly to the SR mapping itself.

This gives EDSR a strong ability to learn the inverse mapping from a LR image to an HR estimate when the training and testing conditions are closely matched. In the present thesis, this makes EDSR particularly competitive on the bicubic low-resolution sets, where the degradation process is relatively deterministic, and consistent with the assumptions under which such models are typically trained.

This behavior is reflected most clearly in the distortion-oriented metrics. On bicubic LR inputs, EDSR typically achieves the highest PSNR and also very strong SSIM among the compared methods. This is an obvious expectation, because its optimization objective is closely aligned with minimizing average pixel-wise reconstruction error.

The network can devote most of its representational power to learning a direct and highly optimized signal reconstruction function, rather than splitting its capacity across robustness mechanisms, perceptual constraints, or feature-space regularization. As a result, EDSR performs as expected: it becomes a very strong upper reference for distortion-based reconstruction.

Another important strength of EDSR is its optimization stability. Since it is trained using a conventional regression loss rather than an adversarial objective, the training process is considerably more predictable than in GAN-based methods. The absence of adversarial loss avoids issues such as generator–discriminator imbalance, mode instability, or oscillatory convergence. In a medical imaging context, deterministic behavior is highly desirable, because it helps ensure that the learned mapping remains consistent across training runs. In this sense, EDSR shows one of the core strengths of classical regression-based SR.

The strength of EDSR also derives from the fact that it is a direct end-to-end SR model with no auxiliary branches, no teacher network, and no multi-term feature alignment objective. This architectural and objective simplicity allows the model to focus entirely on the mapping from the LR input to the final super-resolved output.In other words, EDSR is strong partly because the problem setting is favorable to the type of solution it was designed to learn.

HEDSR is trained using LR images generated only by bicubic downsampling. When the test input follows that same process, the model performance is excellent. But when the test LR image contains deviations from that assumption (for example, additional blur and mild noise), the model is less well prepared to handle the mismatch. Its learned inverse mapping is closely tuned to one degradation regime, and therefore its performance can drop when the input moves outside that regime.

In practical CT imaging, LR appearance is not always equivalent to a clean bicubic resampling of an underlying HR image. Blur, noise, and scanner-dependent variations can alter the LR observation in ways that make the inverse problem more complex.

A second major limitation of EDSR is the absence of explicit perceptual or structural regularization. Although the network is very powerful and can learn useful feature representations, its training objective itself remains pixel-driven. It does not contain an explicit feature-space consistency term, a frozen teacher representation, a perceptual constraint, or any mechanism designed specifically to align the reconstruction with a more stable HR manifold beyond pixel-wise similarity.

As a result, EDSR can achieve very strong distortion metrics while still tending toward the well-known smoothing behavior of regression-based SR.

Its high PSNR should not be misunderstood as evidence that it is always perceptually superior. Rather, its high PSNR indicates that it is very effective at producing reconstructions that are numerically close to the target under distortion-based criteria.

This optimization preference can lead to outputs that are smoother than those produced by models incorporating perceptual losses. In a numerical benchmark, this may be considered advantageous. In visual assessment, especially in anatomical regions, it may mean that EDSR preserves global structure well while under-representing subtle local contrast or high-frequency detail.

EDSR can therefore be understood as a model that performs best when the task is well specified, and the evaluation emphasizes distortion fidelity.

Under these conditions, it remains extremely competitive and serves as a strong baseline. For this reason, EDSR should be interpreted as both a powerful benchmark and a conceptual boundary.

7.4 TRAE and the effect of latent guidance

TRAE is best understood as a response to the two main limitations of a strong conventional regression baseline such as EDSR: first, the tendency of purely pixel-driven supervision to favor smooth average reconstructions; and second, the sensitivity of a narrowly trained model to degradation mismatch.

Rather than trying to solve these issues through adversarial learning, TRAE addresses them within a fully supervised and deterministic setting.

Its significance therefore lies not only in its numerical performance, but in the fact that it introduces a different kind of inductive bias into the SR problem. Instead

of relying solely on a direct pixel-level mapping from LR to HR, it constrains the learning process through latent guidance and dual-target supervision.

The central distinguishing feature of TRAE is the use of a pretrained and frozen Master AE as a stable HR-oriented reference during training. This design changes the nature of supervision in an important way. In a conventional regression model such as EDSR, the network is optimized only by comparing its output to the target image in the pixel domain. By contrast, TRAE also encourages the latent representation extracted from the LR input to align with the latent representation of the corresponding HR target produced by the frozen Master encoder. This means that the model is not only being taught what final output to produce, but also how to organize its internal representation so that it becomes more consistent with HR structure.

This latent guidance mechanism is important because it introduces an explicit form of feature-space regularization that a purely pixel-domain loss cannot provide. The frozen Master branch defines a stable latent space that reflects the dominant structural statistics of HR images in the training domain. Since the Master is fixed after pretraining, this reference does not drift during optimization. The Follower branch is therefore encouraged to map degraded LR inputs into a representation space that remains consistent with HR anatomy and image structure.

As a result, TRAE is designed to learn not only a mapping, but also a more constrained internal representation of the input.

The latent code of the LR input is pulled toward a stable HR-informed latent target, so the network is less free to rely only on pixel-wise minimization. This can help the model preserve more coherent anatomical boundaries, reduce structurally inconsistent local variations, and produce outputs that better respect the global organization of the image.

In medical imaging, this is valuable. CT images are not dominated by highly diverse stochastic textures in the same way as natural images. Instead, they contain recurring anatomical patterns. A supervision strategy that favors HR-consistent latent structure is therefore well aligned with the characteristics of the domain.

Another important component of TRAE is its dual-target supervision at the SR output. The final SR estimate is not trained only against the ground-truth HR image. It is also trained against the reconstruction produced by the frozen Master AE. This dual-target formulation is significant because it balances two complementary constraints. The direct ground-truth target restricts the model to the actual HR image and prevents it from drifting away from the real reconstruction task. The Master reconstruction acts as a deterministic regularizer that reflects the HR manifold learned by the pretrained auto-encoder. In practical terms, this means that the SR output is encouraged to remain close not only to the target image itself, but also to a stable, domain-informed HR reference.

This target can be interpreted as a mechanism for suppressing unstable or undesirable details. Because the Master reconstruction is generated by a pretrained and fixed model, it tends to emphasize the dominant, repeatable structure of the HR image

while being less sensitive to fluctuations.

So, the latent guidance and dual-target supervision make TRAE a multi-objective regression framework. Instead of placing the SR on a single pixel loss, TRAE distributes the learning objective across three complementary goals: LR reconstruction consistency, latent alignment, and SR fidelity under dual supervision. The practical implication is that TRAE may not always maximize distortion-oriented metrics to the same degree as a dedicated model like EDSR, but it is designed to produce a reconstruction process that is more constrained, more robust, and more structurally guided.

On clean bicubic test sets, EDSR can remain extremely strong, and in many cases it is expected to achieve the best PSNR. This is consistent with its design, because it is trained specifically to invert bicubic downsampling with a single, highly focused objective. By contrast, TRAE is not optimized exclusively for this narrow benchmark setting. Its additional regularization terms and more complex training objective mean that some of its capacity is intentionally devoted to structural guidance and generalization rather than to maximizing PSNR or SSIM on one idealized degradation. Therefore, if TRAE leads to slightly lower PSNR than EDSR. This should not be considered simply as worse learning. Rather, it reflects a different allocation of modeling capacity and a different optimization target.

TRAE's training protocol is more robustness-oriented than that of EDSR.

In the revised setting used in this thesis, the LR inputs for TRAE are not generated only by standard bicubic downsampling. Instead, the training introduces a more diverse degradation process, including blur-related variation, with the intention of better reflecting the kinds of softness effects that can occur in practical CT imaging. This has a consequence for evaluation: a model trained under a broader degradation distribution may sacrifice some degree of specialization on clean bicubic test images, but it is generally better prepared for deviations from the idealized benchmark. In other words, TRAE explicitly trades some narrow benchmark optimization for improved robustness and generalization.

Another important aspect of TRAE is its parameter efficiency. The trainable parameter count of the TRAE configuration is lower than that of EDSR. This is an important result because it means that TRAE is not attempting to compete with EDSR by increasing raw network size. Instead, it relies on architectural design to use its trainable capacity more efficiently.

If a smaller model can achieve competitive behavior through better structured supervision, that suggests the gain can come from the quality of the constraints imposed during learning.

At the same time, since TRAE balances multiple loss terms, its training is more delicate than the training of a single-loss regression model. Good performance depends on appropriate weighting between LR reconstruction, latent alignment, and dual-target SR fidelity. If these terms are not balanced properly, the model may become over-regularized or under-constrained.

This means that the stability of TRAE is not identical to the simplicity of EDSR's

optimization. It remains more stable than adversarial training, but it still requires loss design and tuning.

Another trade-off is the two-stage nature of the TRAE. Because the Master AE must first be pretrained and then frozen, the quality of the final TRAE model depends in part on the quality of the Master representation.

This is an inherent cost of relying on a learned internal reference: the guidance becomes informative only to the extent that the reference itself is meaningful. But, once the Master is well trained, this same dependency becomes one of TRAE's strengths, because it transforms that pretrained representation into a stable supervision source.

In addition, the richer objective can require longer convergence than a simpler regression baseline. This is again a natural consequence of the model's design.

EDSR can focus directly on minimizing a single output-domain objective, while TRAE must satisfy multiple coupled constraints. Therefore, even when TRAE is conceptually more guided, it is not necessarily faster to optimize. This should not be read as inefficiency in a negative sense. It shows that the model is solving a broader training problem in exchange for potentially better structural regularity and degradation robustness.

In relation to EDSR, the behavior of TRAE can be summarized as a shift from pure distortion optimization toward structured deterministic reconstruction. EDSR remains stronger when the task exactly matches its assumptions and when PSNR is the dominant criterion. TRAE, on the other hand, introduces explicit mechanisms that aim to improve structural coherence, regularize the reconstruction toward a stable HR manifold, and increase robustness to more realistic LR conditions.

For this reason, TRAE should be seen as a model that intentionally sacrifices some degree of fine details in order to achieve a more balanced and medically appropriate reconstruction behavior.

This is why TRAE is one of the central contributions of the thesis. It shows that the limitations of standard regression-based SR do not necessarily require a move to adversarial or stochastic generation. Instead, they can be addressed through carefully designed deterministic guidance mechanisms that remain supervised and stable.

7.5 MR-TRAE as a multi-resolution extension

MR-TRAE should be considered as the extension of TRAE from a single-scale SR framework to a unified multi-resolution reconstruction framework.

MR-TRAE generalizes TRAE's idea so that multiple super-resolved outputs are produced and supervised within a single end-to-end learning process. The significance of MR-TRAE is not merely that it adds more outputs, but that it redefines the SR task itself as a coupled multi-scale learning problem rather than a collection of isolated single-scale mappings.

MR-TRAE investigates whether the structural guidance of TRAE can be preserved when the model is required to operate across several output scales at once. It also examines whether explicit supervision at intermediate resolutions can improve scale consistency, training behavior, and perceptual reconstruction quality compared with single-output pipelines. This makes MR-TRAE important. It is not only a larger or more complex variant of TRAE. It is rather a test of whether the same deterministic and guided learning principles can support a more flexible and potentially more robust SR strategy.

MR-TRAE's intermediate outputs are not treated as hidden computational by-products whose only purpose is to reach the final highest-resolution estimate. Instead, each resolution level is treated as a valid SR target.

This is a key difference from conventional cascaded SR pipelines, where intermediate stages often serve as internal transformations. In MR-TRAE, the intermediate resolutions become explicit learning objectives, and this changes the optimization behavior. The model treats early stages carefully. Each stage must become individually meaningful and reconstruction-aware. This encourages the entire model to behave as a coordinated sequence of refined but valid predictions.

When intermediate outputs are directly supervised, each stage is forced to reconstruct structure at its own spatial scale. This tends to improve cross-scale coherence, because the model must preserve anatomically meaningful content in a way that remains compatible as resolution increases.

Another advantage of this design is its enhanced gradient flow through the cascaded SR pathway. In deep sequential models, one difficulty is that early modules may receive weaker or less informative training signals, especially if supervision is applied only at the end of the pipeline. Under such conditions, errors may accumulate across stages, and the earlier blocks may not be encouraged strongly enough to produce accurate or stable intermediate representations.

In MR-TRAE, the earlier SR components are trained not only as pre-processing steps for later refinement, but as accountable reconstruction modules. This can improve convergence behavior, reduce the tendency for errors to compound through the cascade, and produce outputs that are more coherent across resolutions.

Like TRAE, MR-TRAE remains a deterministic, paired, and non-adversarial framework. It relies on the pretrained, and frozen Master AE. This reference supports both latent guidance and perceptual regularization.

The frozen Master encoder can act as a domain-specific feature extractor for perceptual comparison that allows the model to measure structural similarity in a learned HR feature space. This is especially important because it avoids dependence on external perceptual networks trained on natural images (like VGG), which may not provide feature spaces well matched to grayscale CT structure. The perceptual behavior of MR-TRAE is driven by domain-consistent guidance.

This makes the perceptual character of MR-TRAE different from both EDSR and SRGAN. Compared with EDSR, MR-TRAE includes explicit feature-level and scale-level regularization that can encourage more structured and perceptually coherent

reconstructions. Compared with SRGAN, however, this perceptual improvement remains tightly constrained by deterministic supervision and a frozen domain-specific reference, which reduces the likelihood of hallucinated patterns.

MR-TRAE seeks greater sharpness and consistency than a purely pixel-driven model, but it pursues these goals without the instability and artifact risk that often come with adversarial generation.

MR-TRAE does not allocate all of its representational and optimization capacity to one final output alone. In a single-output model such as EDSR or scale-specific TRAE, the training objective is concentrated on maximizing performance at one target scale. By contrast, MR-TRAE distributes supervision across multiple resolutions simultaneously. This introduces an inherent multi-task trade-off: the model must balance performance at $\times 2$, $\times 4$, and $\times 8$ rather than specializing exclusively for one of them. As a result, even if MR-TRAE performs very well overall, its score at a particular scale may not surpass that of a model trained specifically and exclusively for that same scale.

This is not a weakness. It is an expected consequence of the model's objective. If MR-TRAE produces quantitative results that are competitive but not always dominant relative to a single-scale baseline, this should not be interpreted as evidence that the multi-resolution design is ineffective. Instead, it reflects the fact that the model is solving a broader and more demanding problem. It is optimizing for a coordinated family of reconstructions, not merely for a single benchmark endpoint.

Another reason for quantitative differences is that MR-TRAE includes the latent guidance and an additional perceptual terms derived from the frozen Master representation. These mean that the model is not optimized purely for minimizing pixel error at the highest scale. In distortion-oriented metrics such as PSNR, a model that is fully dedicated to direct final-scale regression sometimes retains a numerical advantage under idealized conditions.

MR-TRAE balances distortion fidelity against cross-scale consistency, feature-level guidance, and perceptual quality. Consequently, a slight reduction in a single distortion metric may exist but with improved visual continuity across scales.

The highest-output scale in MR-TRAE naturally becomes the most challenging one. As the effective upscaling factor increases, ambiguity in the LR input becomes larger, and the reconstruction problem becomes more difficult. This means that the final output is also the most sensitive to degradation complexity, blur, or information loss in the input. So, the model shows an important truth about SR: reconstruction quality is not uniform across scales, and the practical value of a model may lie partly in offering multiple valid operating points rather than only one maximally enlarged output.

In summary, MR-TRAE is best understood as a multi-resolution generalization of the TRAE in which explicit supervision at several output scales becomes a central learning principle. Its main strengths is scale-consistent reconstruction, improved gradient propagation in the cascade, and domain-specific perceptual regularization without adversarial learning.

7.6 SR-CNN design

The SR-CNN tries to improve perceptual quality within a fully deterministic and non-adversarial SR setting. SR-CNN is explicitly designed to recover sharper and more perceptually convincing reconstructions through a richer training objective. At the same time, unlike SRGAN [33], it does not rely on adversarial competition to generate sharper details. Its significance therefore lies in showing that perceptual realism and artifact control can be improved through loss design, without introducing the instability and hallucination risks.

As described in Section 6.2.3, the total objective of the SR-CNN is constructed from multiple terms rather than from a single distortion measure. A low pixel error may still produce blurred edges; a perceptually sharp image may still contain artificial textures. The SR-CNN therefore combines several forms of supervision so that each one addresses a different failure mode of regression-only super-resolution.

The first component is the robust content Charbonnier loss. This term shows that the most important item is fidelity. However, if the content term were used alone, the model would produce the same smooth output. For this reason, the SR-CNN also considers a perceptual loss, which is one of the key elements responsible for its improved visual realism. Perceptual supervision compares the super-resolved output and the target image in a deep feature space rather than only at the pixel level. This changes the optimization target in a fundamental way. Instead of asking only whether the output matches the target intensity value at each pixel, the model is also asked whether the output matches the target in terms of higher-level feature activations.

In the context of CT SR, this means that fine anatomical contrasts can appear more clearly than in a model trained only with a pixel-wise objective. The result is not necessarily a higher PSNR, but often a more perceptual reconstruction.

Another component of the SR-CNN is the gradient-based loss, which constrains edge and boundary behavior. This term is especially relevant in medical imaging because many important structures are defined by transitions in intensity rather than by rich natural-image textures. A model can achieve a low pixel error while still producing softened edges if it fails to reproduce the correct local gradients. By directly penalizing differences in image gradients, the SR-CNN encourages the preservation of edge sharpness and local structural transitions.

The frequency-domain alignment loss adds another layer of control that is suited to addressing over-smoothing. This term targets the spectral mismatch between the super-resolved output and the HR target, especially in the mid-to-high frequency bands. Over-smoothing is not only a spatial-domain phenomenon; it is also a spectral one. When a model suppresses high-frequency content, the resulting image may preserve coarse structure but lose fine detail and local contrast.

The style or texture loss extends this perceptual framework by encouraging the output to match the statistical structure of the target image in feature space as described in Section 6.6.5. While perceptual losses compare features point-wise, style losses compare second-order feature correlations, which are more closely related to

texture statistics and local structural patterns.

When these loss terms are combined, the resulting model behaves differently from both standard regression baselines and adversarial models. Relative to purely regression-based methods such as EDSR, the SR-CNN is much less dependent on a single distortion criterion and therefore less prone to averaging away important detail. As a result, the output can appear sharper, more detailed, and more visually faithful even when PSNR or SSIM are not maximized. This is an important point for numerical comparisons: a slight reduction in distortion-oriented metrics does not imply that the SR-CNN has failed. Rather, it often reflects the fact that the model has moved toward a different notion of image quality.

At the same time, relative to SRGAN, the SR-CNN remains fundamentally more constrained and more predictable. SRGAN can often produce outputs that appear sharper at first glance because adversarial training rewards realism and local texture synthesis [33]. However, that same mechanism can also introduce synthetic structures and artifacts that are not well supported by the input image. In medical SR, such artifacts are undesirable.

The SR-CNN addresses perceptual quality in a different way. It improves sharpness only through supervised losses that remain in the ground-truth target. So, its outputs may be less aggressive in sharpness than those of SRGAN, but they are generally more controlled and less prone to artifacts.

This richer objective also introduces trade-offs. The first is that training becomes more complex, since the relative weights of the loss terms must be chosen carefully. If the perceptual or style terms are weighted too strongly, the model may become overly texture-seeking. If the content term dominates too strongly, the output may remain too smooth. If the frequency term is over-emphasized, ringing or exaggerated high-frequency responses could appear. Thus, the strength of the SR-CNN lies not in any one loss alone, but in the balance among them.

7.7 SRGAN and the trade-off between sharpness and artifacts

SRGAN [33] has a unique and highly important position in the comparative framework because it represents the canonical shift from distortion-oriented SR to perceptually oriented SR.

The key contribution of SRGAN is that it re-frames SR as not only a reconstruction problem, but also a realism problem. The adversarial pressure encourages the generator to restore sharper edges, richer local contrast, and more realistic high-frequency patterns than those typically produced by regression-only models [33]. As a result, SRGAN became the first widely recognized framework to demonstrate that lower PSNR can still coincide with perceptually superior image quality.

One of the most influential ideas introduced by SRGAN is its perceptual loss, which combines two distinct components: a content term and an adversarial term [33]. The content term is based on VGG feature-space similarity rather than raw pixel-wise

error. In practical terms, this means that the reconstructed image and the target image are compared through high-level feature maps extracted by a pretrained VGG network. The adversarial term, meanwhile, encourages the generator to produce outputs that the discriminator classifies as real. The content term helps preserve correspondence with the target image, while the adversarial term pushes the output toward the natural image manifold.

Table 7.1. A view to the discussed models' specifications.

Aspect	EDSR	SRGAN	TRAE	MRTRAE
PSNR	Best	Week	Moderate	Moderate
SSIM	Good	Week	Better	Best
Stability	Very stable	Unstable	Stable	Stable
Inference speed	Fast	Fast	Fast	Fast
trained parameters	$\approx$ 43.0 M	$\approx$ 8.7 M	$\approx$ 28.0 M	$\approx$ 53.0 M

However, the same mechanism that gives SRGAN its perceptual sharpness also creates its central limitation: the possibility of generating visually plausible but inaccurate details. In medical imaging, this trade-off becomes much more serious than it is in natural-image enhancement.

Another issue is that the discriminator in the original SRGAN formulation evaluates realism from the generated image relative to the distribution of real HR images, but it does not explicitly assess whether the generated details are fully consistent with the specific LR observation that produced them. This means the adversarial criterion is powerful as a general realism prior, but weaker as a sample-specific faithfulness constraint.

The optimization process of SRGAN also contributes to its instability relative to deterministic models. Unlike a standard regression network trained under a single supervised loss, SRGAN must maintain a dynamic balance between two networks: the generator and the discriminator. If the discriminator becomes too strong, the generator may receive poor gradients. If the generator becomes too strong very quickly, the discriminator may fail to provide a useful learning signal. This setup can produce oscillatory behavior and unstable convergence.

Within the comparative results of this thesis, SRGAN should be interpreted as the sharpest but least constrained model among the main baselines. Its outputs can appear more detailed and visually stronger than those of EDSR or even the SR-CNN.

SRGAN sharpens through adversarial pressure, which can generate highly realistic-looking local detail but also introduces a risk of hallucination. SR-CNN, by contrast, sharpens through supervised perceptual, gradient, spectral, and consistency-based losses. As a result, SR-CNN can appear more conservative in local sharpness, but it remains tied to the ground-truth target.

Table 7.1 shows a glance to the specifications of the discussed models.

7.8 Quantitative analysis

This section presents the main quantitative comparison across the three evaluated upscaling factors, namely ×2, ×4, and ×8. In contrast to the later metric-focused discussion, the purpose here is to present the numerical results themselves and interpret how the relative ranking of the models changes with scale and with the assumed LR degradation model. For all three scales, the results are reported under two different LR formation scenarios: "Bicubic", where the LR image is generated only by bicubic downsampling with antialiasing, and "Diverse", where the LR image is generated using the broader degradation operator described in the earlier chapters, including blur, noise, and downsampling. As discussed previously, this is essential because EDSR and SRGAN were trained on bicubic-generated LR data, whereas TRAE, MR-TRAE, and SR-CNN were trained under the broader degradation setting. Consequently, the tables below should be read not only as a ranking of models, but also as evidence of how strongly each model depends on degradation matching.

Table 7.2. Quantitative comparison of the evaluated models for ×2 SR, under "Bicubic" and "Diverse" LR degradation. Higher PSNR and SSIM are better, while lower LPIPS is better.

Model	**Degradation operator**					
	Bicubic			Diverse		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
EDSR	34.12	0.9602	–	33.69	0.9561	–
TRAE	31.98	0.9302	–	32.17	0.9319	–
MR-TRAE	30.60	0.9112	0.208	30.74	0.9143	0.201
SR-CNN	30.21	0.9009	0.207	30.48	0.9034	0.196
SRGAN	28.64	0.8213	0.141	28.15	0.7793	0.154

The results in Table 7.2 show the clearest example of how powerful a strong regression baseline remains under mild enlargement. Under bicubic evaluation, EDSR is the strongest model by a large margin in both PSNR and SSIM, reaching 34.12 dB and 0.9602, respectively. This is exactly what would be expected from a high-capacity regression model trained on bicubic pairs, because the test setting closely matches the degradation model used during training. Even under the diverse degradation setting, EDSR remains numerically strongest in distortion-oriented terms, although both values decrease to 33.69 dB and 0.9561. This drop is itself informative: even a strong model such as EDSR loses performance when the test degradation moves away from its training assumptions.

At the same scale, TRAE remains the closest deterministic alternative to EDSR in terms of PSNR and SSIM, although a noticeable gap remains. Its performance is slightly better under the diverse setting than under bicubic evaluation, which

is consistent with the fact that its training pipeline is more closely aligned with diverse degradation. This supports the interpretation that TRAE is more robust when the test condition reflects the broader LR formation model. MR-TRAE and SR-CNN yield lower PSNR and SSIM than both EDSR and TRAE, but this should be interpreted in the context of their more perceptually oriented supervision.

The SRGAN reaches the lowest LPIPS in the bicubic setting (0.141), which is consistent with its strong perceptual sharpness. But this occurs alongside much lower PSNR and SSIM that confirms the distortion–perception trade-off already discussed in earlier sections.

Table 7.3. Quantitative comparison of the evaluated models for ×4 SR under Bicubic and Diverse LR degradation. Higher PSNR and SSIM are better, while lower LPIPS is better.

Model	**Degradation operator**					
	Bicubic			Diverse		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
EDSR	31.93	0.9297	–	31.24	0.9183	–
TRAE	30.34	0.8712	–	30.57	0.8739	–
MR-TRAE	29.12	0.8537	0.212	29.33	0.8599	0.209
SR-CNN	28.93	0.8509	0.207	29.01	0.8517	0.200
SRGAN	28.64	0.8213	0.163	28.15	0.7793	0.179

As the scale increases to ×4, the inverse problem becomes more difficult, and this is reflected in the lower values across all models in Table 7.3. Once again, EDSR achieves the highest PSNR and SSIM under bicubic evaluation, with 31.93 dB and 0.9297. However, the margin over TRAE becomes smaller than at ×2, especially in PSNR. EDSR still maintains the stronger SSIM.

The ×4 table also shows the increasingly important role of perceptual design. MR-TRAE and SR-CNN remain below EDSR and TRAE in PSNR and SSIM, but their relative differences are not large enough to justify reading them as simply weaker models. Their objectives are broader than direct distortion minimization, and this is reflected in their LPIPS values. Under the "Diverse" setting, SR-CNN reaches the best reported non-adversarial LPIPS value of 0.200, slightly better than MR-TRAE at 0.209, which supports the argument that its composite loss successfully improves perceptual behavior. SRGAN again reaches the most aggressive perceptual score in the bicubic setting (0.163), but it also has the lowest SSIM and the weakest overall distortion performance. Thus, Table 7.3 makes the trade-off more visible than at ×2: the models begin to separate more clearly according to whether they prioritize distortion fidelity, structural guidance, or perceptual enhancement.

Table 7.4. Quantitative comparison of the evaluated models for ×8 super-resolution under "Bicubic" and "Diverse" LR degradation. Higher PSNR and SSIM are better, while lower LPIPS is better.

Model	Degradation operator					
	Bicubic			Diverse		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
EDSR	27.75	0.8050	–	27.01	0.7901	–
TRAE	26.06	0.7742	–	26.37	0.7763	–
MR-TRAE	25.87	0.7610	0.283	26.18	0.7628	0.271
SR-CNN	25.73	0.7601	0.271	25.84	0.7602	0.265
SRGAN	24.90	0.7309	0.240	24.56	0.7093	0.251

At ×8, shown in Table 7.4, the SR task becomes extremely challenging, and all models show substantially reduced performance under the bicubic setting. Under these conditions, EDSR again yields the highest PSNR and SSIM, which is consistent with its strong specialization for direct regression under a known degradation operator. TRAE, MR-TRAE, and SR-CNN all fall below EDSR in bicubic PSNR and SSIM, while SRGAN has the lowest distortion metrics but the best reported bicubic LPIPS at 0.240. This pattern remains consistent with the earlier sections. Adversarial sharpening tends to improve perceptual metrics while reducing distortion fidelity, whereas strong regression still dominates in PSNR under matched synthetic conditions.

Under the "Diverse setting", however, the overall trend still suggests that the broader-degradation training benefits the models designed for such inputs.

Tables 7.2–7.4 show several trends. First, EDSR remains the strongest model in classical distortion metrics when the evaluation is aligned with bicubic training conditions. Second, TRAE remains the closest deterministic alternative in PSNR/SSIM, and its relative competitiveness improves when the degradation model better matches its training. Third, MR-TRAE and SR-CNN do not dominate in distortion terms, but their more perceptually oriented training objectives leads to more favorable LPIPS behavior among the non-adversarial models, especially under the diverse setting. Finally, SRGAN continues to define the perceptual extreme: it typically provides the strongest reported LPIPS, but at the cost of weaker PSNR and SSIM.

7.8.1 Interpretation and scope of the experimental validation

The experimental validation in this thesis should be interpreted primarily as a comparative analysis of reconstruction behavior under controlled medical-image SR settings, rather than as an exhaustive clinical validation study. The experiments

evaluate the main proposed models and representative baselines across three upscaling factors (×2, ×4, and ×8), using both idealized bicubic degradation and a more diverse degradation setting involving blur and noise. This design makes it possible to examine not only absolute numerical performance, but also how each method reacts when the LR formation process becomes less idealized.

The results show that EDSR remains the strongest model in classical distortion metrics, especially when the evaluation degradation matches the bicubic setting for which this type of regression model is well suited. This confirms that high-capacity deterministic regression remains highly competitive when PSNR and SSIM are the main objectives. However, the results also show that perceptually guided deterministic models such as MR-TRAE and SR-CNN can provide more visually detailed reconstructions and more favorable perceptual behavior, even when they do not dominate PSNR or SSIM. This is particularly relevant in medical SR, where purely numerical fidelity does not fully describe the usefulness of a reconstruction.

The comparison with SRGAN further clarifies the main motivation of the thesis. SRGAN often provides stronger perceptual sharpness and lower LPIPS, but this improvement is associated with weaker distortion metrics and a higher risk of visually plausible artifacts. In contrast, the proposed deterministic methods aim to improve perceptual quality while remaining more controlled and reproducible. Therefore, the experiments support the central thesis argument: the proposed models do not simply seek to maximize a single metric, but rather to provide a more balanced compromise among distortion fidelity, perceptual quality, stability, and artifact control.

At the same time, the experimental validation has limitations. The evaluation is based on full-reference metrics and visual inspection of representative CT images, and does not include radiologist scoring, task-based diagnostic evaluation, or testing across multiple independent clinical acquisition protocols. A broader experimental study with multiple datasets, scanner protocols, and clinical expert assessment would be a valuable direction for future work.

7.9 Qualitative visual analysis of reconstruction behavior

While the quantitative metrics provide an important summary of reconstruction fidelity, the visual comparison of the generated SR images shows several behaviors that are not fully captured by PSNR, SSIM, and LPIPS alone.

This is especially true in medical-image SR, where smoothness, edge clarity, residual blur, and artifact visibility all have practical significance.

For this reason, the qualitative figures included in this chapter are essential for interpreting the numerical results in a more complete way. They show how the compared models differ not only in distortion performance, but also in the visual character of the reconstructions they produce.

A first general observation from the qualitative comparisons is that the regression-oriented models and the perceptually guided models exhibit different reconstruction

tendencies. The more distortion-driven methods, especially EDSR, tend to produce clean and structurally stable reconstructions, but they may have a degree of smoothness that decrease fine local contrast.

The perceptually guided deterministic models, namely MR-TRAE and SR-CNN produce visually better local detail and reduced smoothness, even when their PSNR and SSIM values are lower than those of EDSR.

This is important, because it confirms that a reduction in distortion-oriented metrics does not necessarily correspond to an inferior perceptual reconstruction. It often reflects the fact that these models prioritize sharper structural presentation over pixel-wise averaging.

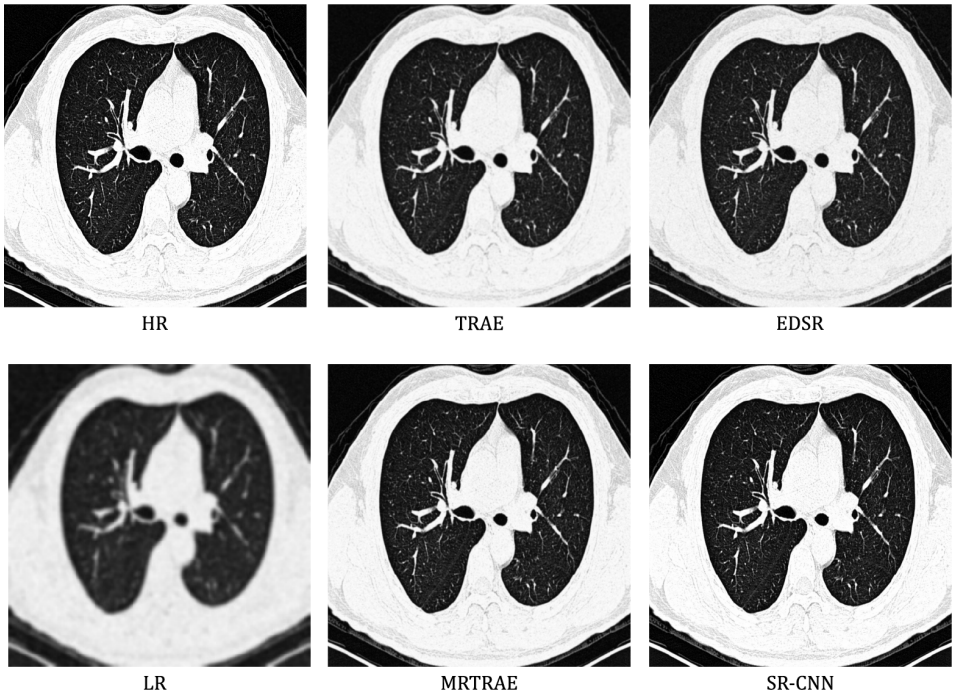

Figure 7.1. Qualitative comparison of the regression-based models at the $\times 2$ scale. The figure compares the super-resolved outputs of EDSR, TRAE, MR-TRAE, and SR-CNN. While EDSR remains strong and visually clean, MR-TRAE and SR-CNN generally reduce the smoothness associated with purely distortion-oriented reconstruction and produce perceptually stronger local detail.

Figure 7.1 provides the visual comparison at the $\times 2$ scale between EDSR, TRAE, MR-TRAE, and SR-CNN. At this upscaling factor, all four models remain capable of preserving the global anatomical structure, and the differences are therefore more subtle than at larger scales. Even so, the distinction in reconstruction style is already visible. EDSR typically produces the cleanest and most numerically faithful image, but its output can appear slightly smoother in fine local regions. TRAE remains visually close to EDSR and generally does not show a severe quality drop, which is consistent with the fact that its latent guidance helps preserve structural organization.

However, because TRAE is not directly optimized with an explicit perceptual loss in the same way as MR-TRAE or SR-CNN, it may still appear somewhat less visually enhanced than those more perceptually guided variants.

By contrast, MR-TRAE and especially SR-CNN tend to reduce the smooth appearance more effectively that produce outputs that look sharper and more visually refined despite their lower distortion metrics.

Figure 7.2 shows a "zoomed in" version for better visualizations.

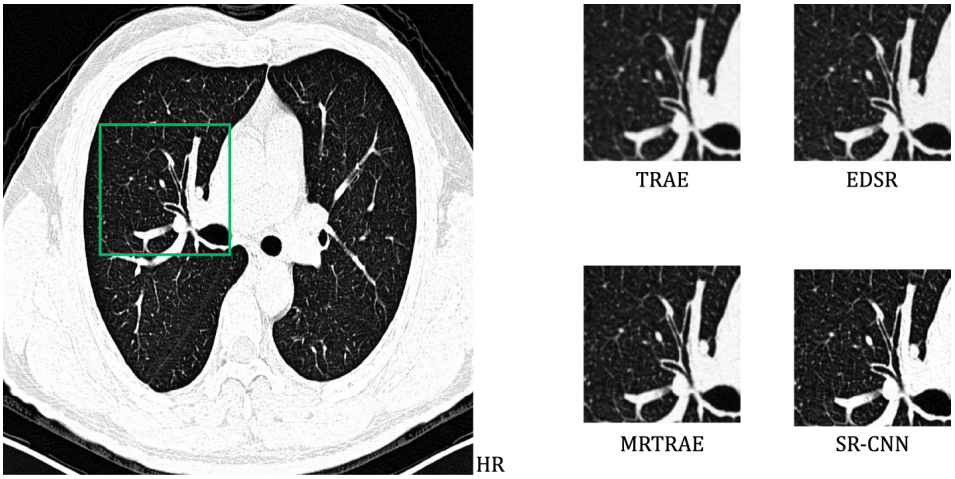

Figure 7.2. Qualitative comparison of the regression-based models at the ×2 scale (zoomed in). The figure compares the super-resolved outputs of EDSR, TRAE, MR-TRAE, and SR-CNN.

The same general pattern continues in the ×4 comparison shown in Figure 7.3, but the differences become more visible because the reconstruction task is more difficult. At this scale, the ambiguity in the LR input is substantially greater, and the limitations of simple regression become easier to observe. EDSR still produces a stable and generally reliable reconstruction, but the residual blur becomes more noticeable in local structures. TRAE remains structurally consistent, and its latent guidance still helps maintain a reasonable reconstruction, but the lack of an explicitly stronger perceptual objective means that its output may remain more conservative than the perceptual models.

MR-TRAE and SR-CNN preserve stronger local transitions and a more visually distinct edge structure.

The challenge becomes larger in the ×8 comparison shown in Figure 7.4. At this scale, the inverse problem is more ill-posed, since a very small LR image must be mapped back to a much larger HR image. All models show visible limitations here, and some degree of residual blur remains unavoidable. This is an important observation. Even the stronger models cannot fully escape the difficulty of such a large magnification factor. The figure still shows meaningful differences in reconstruction behavior. EDSR remains stable and clean, but its smoothness can become more evident. TRAE keeps overall structure reasonably well, although it may not introduce the same level of perceptual enhancement as the more explicitly perceptual models.

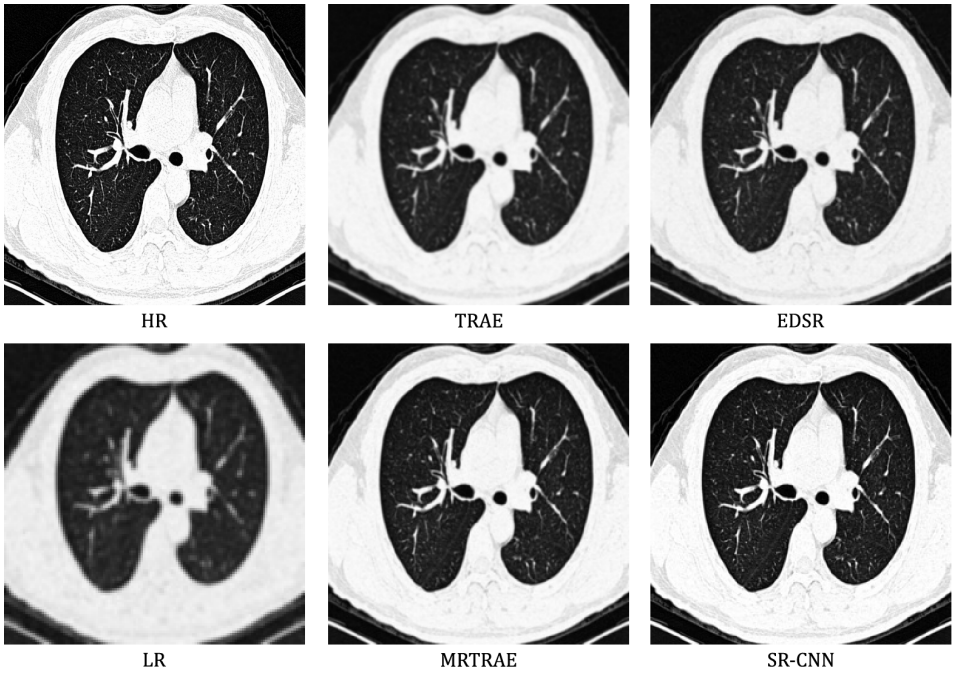

Figure 7.3. Qualitative comparison of the regression-based models at the ×4 scale. As the magnification increases, residual blur becomes more visible in all models, but MR-TRAE and SR-CNN generally retain stronger local perceptual detail than purely distortion-oriented reconstruction.

MR-TRAE and SR-CNN continue to show the advantage of reduced over-smoothing, but the final outputs are still challenged by the severe scale factor, and some blur remains present.

Next qualitative comparison is the visual contrast between SR-CNN and SRGAN, shown in Figure 7.5. This figure is particularly informative because it illustrates the difference between two very different strategies for improving perceptual quality. SR-CNN enhances visual realism through a composite supervised loss, while SRGAN enhances visual realism through adversarial learning. The resulting outputs reflect this difference clearly. The SRGAN reconstructions often appear sharper at first glance and can show stronger local edge contrast than the corresponding SR-CNN results. This is visually consistent with the adversarial objective, which rewards the generator for producing HR outputs that look realistic to the discriminator. By comparison, the SR-CNN outputs are slightly more conservative and have a degree of blur. However, they also remain visually stable, without the same degree of aggressive texture synthesis.

Figure 7.6 give a better representation of the context discussed for 7.5, by zooming in a small patch of the HR and the corresponding patches in super-resolved outputs.

The most important visual insight from this comparison is that greater sharpness is not always equivalent to greater reliability. Although SRGAN can produce images that look more detailed, its outputs may also contain visible artifacts or synthetic high-frequency patterns that are not supported by the true HR structure.

Figure 7.4. Qualitative comparison of the regression-based models at the ×8 scale. This is the most difficult setting, and all methods retain some residual blur. Even so, perceptually guided models still tend to reduce the smooth appearance compared with purely distortion-driven reconstruction.

Overall, the qualitative figures support important conclusions.

First, they confirm that distortion-oriented models such as EDSR remain strong and stable, but they often produce smoother reconstructions.

Second, they show that MR-TRAE and SR-CNN improve visual quality mainly by reducing the smoothness typical of regression-based reconstruction, which helps explain why they can look better despite lower PSNR and SSIM. Finally, they make it clear that SRGAN achieves the greatest apparent sharpness, but that this sharpness comes with a meaningful risk of visible artifacts.

7.10 Metric trade-offs

The quantitative results presented in Tables 7.2–7.4 show that the evaluation of SR models cannot be reduced to a single metric. Although PSNR, SSIM, and LPIPS are all reported for the same reconstructions, they measure different aspects of image quality.

PSNR is the most classical distortion-oriented metric and is directly linked to pixel-domain reconstruction error.

However, the same property that makes PSNR useful also limits its scope. Because it rewards the minimization of average pixel error, it favors reconstructions that

Figure 7.5. Qualitative comparison between SR-CNN and SRGAN for scaling factor x2. SRGAN produces sharper outputs. SR-CNN remains more conservative but visually more controlled.

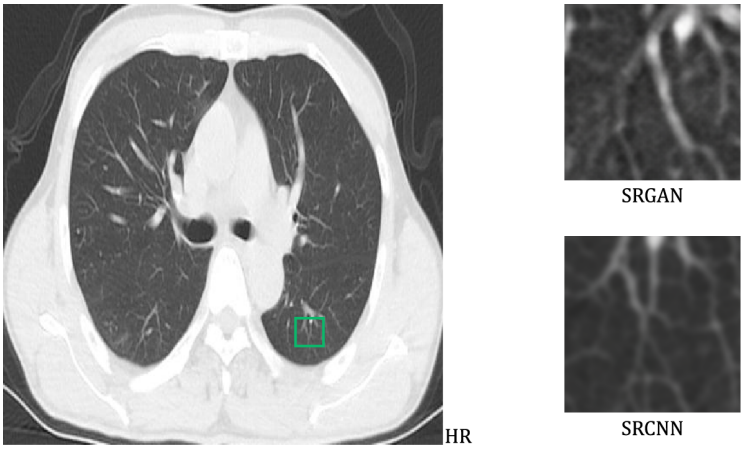

Figure 7.6. Zoomed-in comparison between SR-CNN, SRGAN, and the ground-truth HR patch. The SRGAN output appears sharper but also reveals artifact-prone local structures, whereas SR-CNN remains slightly smoother but more visually controlled.

remain close to the conditional mean of plausible solutions. In an ill-posed inverse problem such as SR, this often leads to over-smoothed outputs.

Figure 7.7. Qualitative comparison between SR-CNN and SRGAN for scaling factor x4.

SSIM provides a broader perspective by evaluating similarity in terms of local luminance, contrast, and structural organization rather than only raw pixel differences. For this reason, it is often more sensitive than PSNR to the preservation of meaningful structural relationships.

This makes SSIM especially relevant in medical imaging, where the visibility of anatomical boundaries and internal structure is often more important than exact pixel agreement alone. In the reported results, SSIM broadly follows the same general ranking trends as PSNR, with EDSR frequently remaining strongest and TRAE often emerging as the closest deterministic alternative. This is visible, for example, in Tables 7.2 and 7.3, where the models that are strongest in distortion terms also remain strong in SSIM.

Even so, SSIM should not be treated as a full perceptual metric. Although it is more structure-aware than PSNR, it still remains an image-domain comparison that favors relatively conservative reconstructions. It does not directly measure whether the output looks perceptually sharper, whether feature-level similarity is improved, or whether the reconstruction is visually more realistic.

LPIPS reflects a different aspect of evaluation. Instead of comparing two images directly in the pixel domain, it compares them in the feature space of a pretrained

convolutional network. As a result, LPIPS is more sensitive to perceptual similarity in a learned representation space than to exact pixel-wise agreement. A lower LPIPS therefore indicates that the reconstructed image is closer to the target in a feature-level sense.

The role of LPIPS is visible in all three quantitative tables. In Table 7.2, for example, SRGAN achieves the lowest reported bicubic LPIPS, while SR-CNN and MR-TRAE also show competitive perceptual scores under the diverse setting. A similar pattern appears in Table 7.3, where SRGAN again leads in perceptual score under bicubic evaluation, while SR-CNN achieves the strongest reported non-adversarial LPIPS under diverse degradation. These results show that the models designed to improve perceptual quality do in fact gain an advantage when evaluated in a perceptual feature space, even when they do not lead in PSNR or SSIM. This is why LPIPS is good in the present evaluation framework.

In summary, the metric trade-offs observed in this thesis are an expected and meaningful consequence of the fact that PSNR, SSIM, and LPIPS reflect different definitions of reconstruction quality.

From a practical perspective, the compared models differ not only in reconstruction quality, but also in how suitable they are for stable and efficient use cases. One of the clearest distinctions concerns training stability. Purely regression-based models such as EDSR, TRAE, MR-TRAE, and SR-CNN are generally more stable to optimize than SRGAN, because they do not rely on adversarial competition between a generator and a discriminator. In practice, this means they are less sensitive to optimization imbalance and to oscillatory training behavior, and more predictable across training.

The findings of this chapter are more important when interpreted in the context of CT SR. In this application domain, SR is not merely a visual enhancement task. The reconstructed image must preserve anatomically meaningful structure, maintain consistent grayscale transitions, and avoid introducing unsupported detail that could affect interpretation.

This is why structural fidelity and artifact suppression are very important. Models such as EDSR remain valuable because they are stable, deterministic, and strong in distortion-oriented reconstruction, which supports reliable structural recovery under matched conditions. At the same time, the proposed TRAE and MR-TRAE frameworks are relevant because they aim to improve reconstruction through structured guidance rather than through unconstrained perceptual synthesis. Their deterministic latent guidance and domain-oriented supervision make them better aligned with the need for trustworthy image recovery in CT. Similarly, SR-CNN is important because it improves perceptual realism through supervised losses and the assumed degradation process that reduce the risk of uncontrolled hallucination.

Despite the usefulness of the comparative results, the conclusions of this chapter should be interpreted within the limits of the present study. A first limitation is that the evaluation remains dependent on the selected metrics.

Although PSNR, SSIM, and LPIPS together provide a broader view than any

single metric alone, they still cannot fully capture all clinically relevant aspects of reconstruction quality. In particular, no full-reference image metric can by itself determine whether a reconstruction is clinically safer or more diagnostically useful.

A second limitation concerns the degradation assumptions used to generate the LR inputs. While the inclusion of both bicubic and diverse degradation settings improves the realism of the comparison, these are still synthetic approximations of real image formation. Actual CT degradation can be influenced by scanner properties, reconstruction kernels, dose conditions, and additional acquisition-specific factors that are not fully reproduced by the training pipelines used here.

A third limitation is that the compared models were not all optimized for exactly the same objective. EDSR was designed primarily for distortion minimization, SRGAN for perceptual realism through adversarial learning, TRAE and MR-TRAE for guided deterministic reconstruction plus perceptual loss, and SR-CNN for perceptual improvement through composite supervised loss design. As a result, direct numerical comparison is informative, but it is not a perfectly symmetric competition in which all models are pursuing the same target. Some models are expected to perform best in distortion metrics, while others are expected to sacrifice some distortion performance in exchange for better perceptual behavior or improved robustness.

This chapter has examined the comparative behavior of the evaluated SR models from quantitative, qualitative, practical, and application-oriented perspectives.

The results support the conclusion that *no single model is universally best under every criterion*. The comparative evidence clarifies the trade-offs that define the SR problem in a medical imaging context. In particular, the findings show that the balance among distortion fidelity, perceptual quality, robustness to degradation, and artifact control is central to determining which model is most suitable for CT super-resolution.

Chapter 8

Conclusion

This thesis addressed the problem of SISR in a medical setting, with a particular focus on CT imaging. The central motivation was that conventional SR methods, although often strong in terms of PSNR and SSIM, produce over-smoothed reconstructions.

More perceptually aggressive generative approaches can introduce visually convincing but clinically undesirable artifacts. In medical imaging, neither of these extremes is fully satisfactory. The main objective of this thesis was therefore to investigate whether perceptual quality can be improved within a deterministic and supervised framework, while preserving structural fidelity, training stability, and resistance to artifact generation.

To address this objective, the thesis developed and studied several complementary strategies for improving SR beyond the limitations of standard regression-only reconstruction. The first major contribution was the TRAE framework, which incorporated latent guidance into a deterministic SR model. By using a pretrained and frozen Master AE as an HR-informed reference, TRAE introduced feature-level structural guidance that is absent in conventional single-loss regression models.

The second contribution was the extension of this concept into the multi-resolution setting through MR-TRAE. Rather than treating each upscaling factor as a separate problem requiring a separate model, MR-TRAE reformulated SR as a coordinated multi-scale learning task with explicit supervision at several output resolutions.

The third contribution was the development of the SR-CNN, which approached the perceptual-quality problem from a different but equally important direction. Instead of relying on a latent-guidance architecture, SR-CNN improved perceptual realism through a designed composite training objective. This contribution is significant because it shows that perceptual enhancement in medical SR does not necessarily require adversarial training. Instead, much of the perceptual benefit can be recovered through better objective design while remaining more tightly constrained by the target image and the assumed degradation model.

The thesis also made an important comparative contribution by evaluating these models against established baselines such as EDSR [35] and SRGAN [33]. This

comparison made it possible to interpret the proposed methods not in isolation, but within the broader landscape of modern SR. The results showed that EDSR remains a very strong distortion-oriented baseline, especially under matched bicubic degradation, and that SRGAN remains a strong perceptual baseline when judged by visual sharpness and LPIPS.

This supports a conclusion: the gap between classical deterministic SR and perceptually enhanced reconstruction, specially in the context of medical imaging, can be narrowed without relying on adversarial or stochastic generation.

The experiments with the degradation modeling showed that SR performance cannot be interpreted independently of the LR formation process used during training and evaluation. Models trained only on bicubic downsampling naturally perform best when test images are generated by the same operator, but this advantage can weaken when the evaluation reflects broader and more realistic degradation.

The joint use of PSNR, SSIM, and LPIPS throughout the thesis showed clearly that no single metric is sufficient to characterize SR quality in a fidelity-sensitive domain. PSNR captures distortion fidelity, SSIM reflects structural similarity in the image domain, and LPIPS captures perceptual similarity in a learned feature space. Since these metrics measure different aspects of reconstruction, their divergence is expected and meaningful.

The work does not merely propose alternative SR models. It also argues for a more appropriate design strategy for medical image SR. In CT imaging, the most useful reconstruction is not necessarily the one with the highest distortion metric, nor the one with the sharpest visual appearance in isolation. Rather, the most suitable model is the one that preserves anatomical structure and improves visual clarity in a controlled manner.

The evaluation relied on synthetic LR generation, even when the degradation process was broadened beyond bicubic downsampling. Although this provides a more realistic testing framework than idealized benchmarks alone, it does not fully capture all scanner-dependent and acquisition-dependent sources of variation in real clinical CT data.

These limitations naturally suggest several directions for future work. One important extension would be to validate the proposed approaches on broader clinical datasets and on LR data that reflect real acquisition conditions more directly, including scanner-dependent variability and reconstruction-kernel effects.

A second important direction would be to further investigate clinically constrained perceptual objectives that improve visual clarity while preserving strict anatomical faithfulness.

A third direction would be to explore domain-informed supervision strategies that incorporate additional prior knowledge, such as acquisition physics, anatomical consistency, or downstream clinical tasks, into the SR process.

In conclusion, this thesis has shown that deterministic and supervised SR can be improved beyond regression formulations only. Through the development of TRAE,

MR-TRAE, and SR-CNN, and through their comparative evaluation against strong baselines, the thesis showed that guided and perceptually informed reconstruction gives a promising path for medical image SR.

Bibliography

[1] Ahmadi, K. and Salari, E. Single-image super resolution using evolutionary sparse coding technique. *IET Image Processing*, **11** (2017), 13. doi:10.1049/iet-ipr.2016.0273.

[2] Ahmed, A., Kun, S., Ahmed, J., Hayat, S., and Khan, A. A. Multimodal image enhancement using convolutional sparse coding. *Multimedia Systems*, **29** (2023), 2099. doi:10.1007/s00530-023-01074-1.

[3] Baccarelli, E., Scarpiniti, M., and Momenzadeh, A. Twinned residual auto-encoder (TRAE)–A new DL architecture for denoising super-resolution and task-aware feature learning from COVID-19 CT images. *Expert Syst. Appl.*, **225** (2023), 120104. doi:j.eswa.2023.120104.

[4] Chen, H., He, X., et al. Real-world single image super-resolution: A brief review. *Inf. Fusion*, **79** (2022), 124. doi:10.1016/j.inffus.2021.09.005.

[5] Chen, Y., Wang, Y., et al. Lctc: Lightweight convolutional thresholding sparse coding network prior for compressive hyperspectral imaging. *IEEE Transactions on Image Processing*, **34** (2025), 4286. doi:10.1109/TIP.2025.3583951.

[6] Chen, Y., Zheng, Q., and Chen, J. Double paths network with residual information distillation for improving lung CT image super resolution. *Biomed. Signal Process. Control*, **73** (2022), 103412. doi:10.1016/j.bspc.2021.103412.

[7] Dabov, K., Foi, A., et al. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.*, **16** (2007), 2080. doi:10.1109/TIP.2007.901238.

[8] Deka, B., Datta, S., Mullah, H. U., and Hazarika, S. Diffusion-weighted and spectroscopic MRI super-resolution using sparse representations. *Biomedical Signal Processing and Control*, **60** (2020), 101941. doi:10.1016/j.bspc.2020.101941.

[9] Dong, C., Loy, C. C., He, K., and Tang, X. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38** (2016), 295. https://doi.org/10.1109/TPAMI.2015.2439281.

[10] Dong, C., Loy, C. C., et al. Learning a deep convolutional network for image super-resolution. In *Computer Vision – ECCV 2014*, pp. 184–199. Springer International Publishing, Zurich, Switzerland (2014). doi:10.1007/978-3-319-10593-2_13.

[11] Dong, C., Loy, C. C., et al. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, **38** (2016), 295. doi:10.1109/TPAMI.2015.2439281.

[12] Dong, J., Zhang, H., et al. Fast video super-resolution via sparse coding. In *Sixth International Conference on Graphic and Image Processing (ICGIP 2014)* (edited by Y. Wang, X. Jiang, and D. Zhang), vol. 9443, p. 94432A. International Society for Optics and Photonics, SPIE (2015). doi:10.1117/12.2179397.

[13] Gan, H., Shen, M., et al. From patch to pixel: A transformer-based hierarchical framework for compressive image sensing. *IEEE Transactions on Computational Imaging*, **9** (2023), 133. doi:10.1109/TCI.2023.3244396.

[14] Geng, T., Liu, X.-Y., et al. Deep shearlet residual learning network for single image super-resolution. *IEEE Transactions on Image Processing*, **30** (2021), 4129. doi:10.1109/TIP.2021.3069317.

[15] Goodfellow, I., Pouget-Abadie, J., et al. Generative adversarial networks. *Communications of the ACM*, **63** (2020), 139. doi:10.1145/3422622.

[16] Greenspan, H. Super-resolution in medical imaging. *The Computer Journal*, **52** (2009), 43. https://doi.org/10.1093/comjnl/bxm075.

[17] Gunraj, H., Sabri, A., et al. COVID-Net CT-2: enhanced deep neural networks for detection of COVID-19 from chest CT images through bigger, more diverse learning. *Front. Med.*, **8** (2022). doi:10.3389/fmed.2021.729287.

[18] Guo, C., Li, C., et al. Hierarchical features driven residual learning for depth map super-resolution. *IEEE Transactions on Image Processing*, **28** (2019), 2545. doi:10.1109/TIP.2018.2887029.

[19] Hassan, E., El-Rashidy, N., et al. Exploring the frontiers of image super-resolution: a review of modern techniques and emerging applications. *Neural Computing and Applications*, **37** (2025), 17913. doi:10.1007/s00521-025-11331-1.

[20] Hassan, M., Illanko, K., and Fernando, X. N. Single image super resolution using deep residual learning. *AI*, **5** (2024), 426. doi:10.3390/ai5010021.

[21] He, K., Zhang, X., et al. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016).

[22] Ho, J., Jain, A. N., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, vol. 33 of *NeurIPS 2020*, pp. 6840–6851. Curran Associates, Inc. (2020).

[23] Hou, J., Si, Y., and Yu, X. A novel and effective image super-resolution reconstruction technique via fast global and local residual learning model. *Applied Sciences*, **10** (2020), 1856. doi:0.3390/app10051856.

[24] Huang, G., Liu, Z., et al. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR'17, pp. 4700–4708. IEEE, Honolulu, HI, USA (2017). doi:10.1109/CVPR.2017.243.

[25] Huang, J.-J. and Dragotti, P. L. A deep dictionary model for image super-resolution. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6777–6781 (2018). doi:10.1109/ICASSP.2018.8461651.

[26] Huang, J.-J. and Dragotti, P. L. Learning deep analysis dictionaries for image super-resolution. *IEEE Transactions on Signal Processing*, **68** (2020), 6633. doi:10.1109/TSP.2020.3036902.

[27] Jiang, J., Ma, J., et al. Noise robust face image super-resolution through smooth sparse representation. *IEEE Transactions on Cybernetics*, **47** (2017), 3991. doi:10.1109/TCYB.2016.2594184.

[28] Jiang, J., Ma, X., et al. Sparse support regression for image super-resolution. *IEEE Photonics Journal*, **7** (2015), 1. doi:10.1109/JPHOT.2015.2484287.

[29] Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV 2016)*, pp. 694–711. Amsterdam, The Netherlands (2016). doi:10.1007/978-3-319-46475-6_43.

[30] Karras, T., Aila, T., et al. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, (2017). Available from: https://arxiv.org/abs/1710.10196.

[31] Kasiri, S. and Ezoji, M. Single mr-image super-resolution based on convolutional sparse representation. *Signal, Image and Video Processing*, **14** (2020), 1525. doi:10.1007/s11760-020-01698-0.

[32] Kim, J., Lee, J. K., and Lee, K. M. Accurate image super-resolution using very deep convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pp. 1646–1654. Las Vegas, NV, USA (2016). https://doi.org/10.1109/CVPR.2016.182.

[33] Ledig, C., Theis, L., et al. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pp. 105–114. IEEE, Honolulu, HI, USA (2017). doi:10.1109/CVPR.2017.19.

[34] Li, H., Yang, Y., et al. SRDiff: single image super-resolution with diffusion probabilistic models. *Neurocomputing*, **479** (2022), 47. doi:10.1016/j.neucom.2022.01.029.

[35] Lim, B., Son, S., et al. Enhanced deep residual networks for single image super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, CVPRW'17, pp. 1132–1140. IEEE, Honolulu, HI, USA (2017). doi:10.1109/CVPRW.2017.151.

[36] Liu, A., Liu, Y., et al. Blind image super-resolution: A survey and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.*, **45** (2023), 5461. doi:10.1109/TPAMI.2022.3203009.

[37] Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR 2019)* (2019). Available from: https://openreview.net/forum?id=Bkg6RiCqY7.

[38] Maeda, S. Image super-resolution with deep dictionary. In *Computer Vision – ECCV 2022*, pp. 464–480. Springer Nature Switzerland, Cham (2022). doi:10.1007/978-3-031-19800-7_27.

[39] Mei, X., Liu, Z., et al. RadImageNet: an open radiologic deep learning research dataset for effective transfer learning. *Radiology: Artificial Intelligence*, **4** (2022), e210315. doi:10.1148/ryai.210315.

[40] Momenzadeh, A., Baccarelli, E., et al. Multi-resolution twinned residual auto-encoders (mr-trae)-a novel DL model for image multi-resolution. *Cognitive Computation*, **16** (2024), 1447. doi:10.1007/s12559-024-10293-1.

[41] Moser, B. B., Shanbhag, A. S., et al. Diffusion models, image super-resolution, and everything: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, **36** (2025), 11793. doi:10.1109/TNNLS.2024.3476671.

[42] Mu, G., Gao, X., Zhang, K., Li, X., and Tao, D. Single image super resolution with high resolution dictionary. In *2011 18th IEEE International Conference on Image Processing*, pp. 1141–1144 (2011). doi:10.1109/ICIP.2011.6115630.

[43] Mullah, H. U., Deka, B., and Prasad, A. Fast multi-spectral image super-resolution via sparse representation. *IET Image Processing*, **14** (2020), 2833. doi:10.1049/iet-ipr.2019.0714.

[44] Nakashima, R., Takahashi, K., and Naemura, T. Super-resolved free-viewpoint image synthesis combined with sparse-representation-based super-resolution. In *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 1–6 (2013). doi:10.1109/APSIPA.2013.6694248.

[45] Nayak, R., Patra, D., and Harshavardhan, S. Sparse representation based image super resolution reconstruction. In *TENCON 2015 - 2015 IEEE Region 10 Conference*, pp. 1–6 (2015). doi:10.1109/TENCON.2015.7373149.

[46] Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *proceedings of the 38th International Conference on Machine Learning*, pp. 8162–8171 (2021).

[47] Qiao, J., Song, H., et al. Image super-resolution using conditional generative adversarial network. *IET Image Processing*, **13** (2019), 2673. doi:10.1049/iet-ipr.2018.6570.

[48] Qin, F., Shen, Z., et al. Infraffn: A feature fusion network leveraging dual-path convolution and self-attention for infrared image super-resolution. *Knowledge-Based Systems*, **310** (2025), 112960. doi:10.1016/j.knosys.2025.112960.

[49] Ren, J., Liu, J., and Guo, Z. Context-aware sparse decomposition for image denoising and super-resolution. *IEEE transactions on Image Processing*, **22** (2013), 1456. doi:10.1109/TIP.2012.2231690.

[50] Ren, J., Liu, J., et al. Image super-resolution by structural sparse coding. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 1936–1939 (2012).

[51] Song, P., Mota, J. F. C., et al. Coupled dictionary learning for multimodal image super-resolution. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 162–166 (2016). doi:10.1109/GlobalSIP.2016.7905824.

[52] Tian, C., Song, M., et al. A tree-guided cnn for image super-resolution. *IEEE Transactions on Consumer Electronics*, **71** (2025), 3631. doi:10.1109/TCE.2025.3572732.

[53] Tian, C., Xu, Y., et al. Coarse-to-fine CNN for image super-resolution. *IEEE Trans. Multimedia*, **23** (2021), 1489. doi:10.1109/TMM.2020.2999182.

[54] Tian, C., Zhuge, R., et al. Lightweight image super-resolution with enhanced cnn. *Knowledge-Based Systems*, **205** (2020), 106235. doi:10.1016/j.knosys.2020.106235.

[55] Wang, Z., Yang, J., et al. *Sparse coding and its applications in computer vision*. World Scientific (2015).

[56] Wei, S., Zhou, X., et al. Medical image super-resolution by using multi-dictionary and random forest. *Sustainable Cities and Society*, **37** (2018), 358. doi:10.1016/j.scs.2017.11.012.

[57] Wei, Z., Xiaofeng, B., Fang, H., Jun, W., and Abidi, M. A. Fast image super-resolution algorithm based on multi-resolution dictionary learning and sparse representation. *Journal of Systems Engineering and Electronics*, **29** (2018), 471. doi:10.21629/JSEE.2018.03.04.

[58] Wu, R., Yang, T., et al. Seesr: Towards semantics-aware real-world image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25456–25467 (2024).

[59] Yang, J., Wang, Z., et al. Bilevel sparse coding for coupled feature spaces. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2360–2367 (2012). doi:10.1109/CVPR.2012.6247948.

[60] Yang, J., Wang, Z., et al. Coupled dictionary training for image super-resolution. *IEEE Transactions on Image Processing*, **21** (2012), 3467. doi:10.1109/TIP.2012.2192127.

[61] Yang, J., Wright, J., Huang, T. S., and Ma, Y. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, **19** (2010), 2861. https://doi.org/10.1109/TIP.2010.2050625.

[62] Yang, W., Feng, J., et al. Deep edge guided recurrent residual learning for image super-resolution. *IEEE Transactions on Image Processing*, **26** (2017), 5895. doi:10.1109/TIP.2017.2750403.

[63] You, C., Li, G., et al. CT super-resolution GAN constrained by the identical, residual, and cycle learning ensemble (GAN-CIRCLE). *IEEE Transactions on Medical Imaging*, **39** (2020), 188. https://doi.org/10.1109/TMI.2019.2922960.

[64] Zhang, H., Xiao, J., and Jin, Z. Multi-scale image super-resolution via a single extendable deep network. *IEEE J. Sel. Top. Signal Process.*, **15** (2021), 253. doi:10.1109/JSTSP.2020.3045282.

[65] Zhang, H., Zhou, W., et al. Cascade residual learning based adaptive feature aggregation for light field super-resolution. *Pattern Recognition*, **165** (2025), 111616. doi:10.1016/j.patcog.2025.111616.

[66] Zhang, K., Gao, X., et al. Multi-scale dictionary for single image super-resolution. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1114–1121 (2012). doi:10.1109/CVPR.2012.6247791.

[67] Zhang, W., Li, X., et al. Real-world image super-resolution as multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)* (2023).

[68] Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., and Fu, Y. Image super-resolution using very deep residual channel attention networks. In *15th European Conference on Computer Vision (ECCV 2018)*, pp. 294–310. Munich, Germany (2018). https://doi.org/10.1007/978-3-030-01234-2_18.

[69] Zhang, Y., Tian, Y., et al. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

[70] Zhao, L., Sun, Q., and Zhang, Z. Single image super-resolution based on deep learning features and dictionary model. *IEEE Access*, **5** (2017), 17126. doi:10.1109/ACCESS.2017.2736058.

[71] Zheng, H., Bouzerdoum, A., and Phung, S. L. Depth image super-resolution using multi-dictionary sparse representation. In *2013 IEEE International Conference on Image Processing*, pp. 957–961 (2013). doi:10.1109/ICIP.2013.6738198.