IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 16, 2022, accepted April 28, 2022, date of publication May 5, 2022, date of current version May 12, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3172939

Image Classification With Small Datasets: Overview and Benchmark

LORENZO BRIGATO[1], BJÖRN BARZ[2], LUCA IOCCHI[1], AND JOACHIM DENZLER[2], (Member, IEEE)
[1]Department of Computer, Control and Management Engineering, Sapienza University of Rome, 00185 Rome, Italy
[2]Department of Mathematics and Computer Science, Friedrich Schiller University Jena, 07743 Jena, Germany

Corresponding author: Lorenzo Brigato (brigato@diag.uniroma1.it)

ABSTRACT Image classification with small datasets has been an active research area in the recent past. However, as research in this scope is still in its infancy, two key ingredients are missing for ensuring reliable and truthful progress: a systematic and extensive overview of the state of the art, and a common benchmark to allow for objective comparisons between published methods. This article addresses both issues. First, we systematically organize and connect past studies to consolidate a community that is currently fragmented and scattered. Second, we propose a common benchmark that allows for an objective comparison of approaches. It consists of five datasets spanning various domains (*e.g.*, natural images, medical imagery, satellite data) and data types (RGB, grayscale, multispectral). We use this benchmark to re-evaluate the standard cross-entropy baseline and ten existing methods published between 2017 and 2021 at renowned venues. Surprisingly, we find that thorough hyper-parameter tuning on held-out validation data results in a highly competitive baseline and highlights a stunted growth of performance over the years. Indeed, only a single specialized method dating back to 2019 clearly wins our benchmark and outperforms the baseline classifier.

INDEX TERMS Data-efficiency, image classification, benchmark, neural networks, small datasets.

I. INTRODUCTION

Many recent advances in computer vision and machine learning in general have been achieved by large-scale pre-training on massive datasets [15], [16], [45]. For instance, the most popular dataset for image classification, ImageNet-1k [48], contains one thousand classes, each comprised of several hundred or over one thousand training examples. However, reaching high recognition performance by training on large-scale datasets is strictly connected to the laborious process of collecting and labeling large quantities of samples. Application scenarios in which the usual pre-training on large web-sourced image datasets is useless due to a strong domain shift (*e.g.*, document style classification [51]) or even impossible due to different data modalities (*e.g.*, multi-channel spectral data from satellites [21]) strictly depend on methods for learning directly from the limited amounts of data available.

Deep learning from small datasets is a research area that has been receiving increasing interest in the past couple of years [3], [8], [9], [41]. The distinctive feature that makes this field of research differ from other learning problems is that, the use of additional, external datasets, *e.g.*, for pre-training neural networks, is not available. While nowadays popular datasets contain hundreds or thousands of training examples per class, *deep learning from small data* trains neural networks on tens or hundreds of samples per category. These extreme settings exacerbate the well-known weaknesses of neural networks *i.e.*, being prone to memorizing spurious correlations among training features instead of actually learning a general function for the requested task [18]. Therefore, deploying a performant classifier in this scenario remains an important challenge.

To keep focus, we limit the scope of our study to image classification with few examples, excluding other computer vision domains such as object detection or semantic segmentation that have also recently gained increasing attention [9], [38]. Despite their invaluable importance, such domains are still lacking a sufficiently large body of literature for an extensive overview. In contrast, our literature analysis on image classification with small datasets led to a substantial

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

number of existing works, probably because image classification remains one of the most established tasks for artificial neural networks.

A key missing piece of the current literature is an objective comparison of proposed methods due to the lack of a common benchmark. Fortunately, there recently have been activities to establish common benchmarks and organize challenges to foster direct competition between proposed methods [9]. Still, they are often limited to a single dataset, *e.g.*, ImageNet [48], which comprises a different type of data than usually encountered in a small-data scenario. Moreover, most existing works compare their proposed method against insufficiently tuned baselines [3] or baselines trained with default hyper-parameters [26], [29], [41], [52], [56], which makes it easy to outperform them. Careful hyper-parameter optimization (HPO) [6] is not only crucial for applying deep learning techniques in practice but also for a fair comparison between different methods so that each can obtain its optimal or near-to-optimal performance. Comparing against an untuned baseline with default hyper-parameters does not provide clear evidence of improvements. Additionally, due to the fragmentation of the relevant literature, approaches are rarely compared to the existing state of the art, obfuscating the progress of the field.

The contributions of this paper enrich the literature on image classification with small datasets with two fundamental building blocks that are currently missing: 1) a review of the recent literature and 2) a dedicated benchmark. The former represents the first comprehensive collection of works on image classification with small datasets. We provide a clear overview of the current literature and existing approaches. The second building block is a dedicated benchmark allowing for a direct, objective, and informative comparison of existing and future methods. The benchmark consists of five datasets from different domains: natural images of everyday objects, fine-grained classification, medical imagery, satellite images, and handwritten documents. Two datasets consist of non-RGB data, where the common large-scale pre-training and the fine-tuning procedure is not straightforward, emphasizing the need for methods that can learn from limited amounts of data from scratch. Our dataset splits, implementations of all compared methods, and code for reproducing our experiments is publicly available under https://github.com/lorenzobrigato/gem.

For our benchmark, we carefully optimize the hyper-parameters of all methods for each dataset individually on held-out validation data, while evaluating the final performance on a separate test split. Surprisingly and somewhat disillusioning, we discover two key findings that are summarized in Fig. 1: 1) hyper-parameter optimization makes the categorical cross-entropy loss a strong baseline that is outperformed by only one of the ten specialized methods evaluated; 2) there is no clear performance progress considering the approaches published in the recent literature. The untuned baseline (red dashed line), *i.e.*, a classifier trained with default hyper-parameters, underperforms the baseline trained with HPO (green dashed line) by ~13 percentage points. In more detail, such a strong baseline is obtainable by combining small batch sizes (*i.e.*, as small as 8), strong regularization (*i.e.*, weight decays up to $\sim 10^{-2}$), and small learning rates confined to a limited range (*i.e.*, $\sim[10^{-4}, 10^{-3}]$). To show the lack of performance progress, we order the evaluated methods on the proposed benchmark along the x-axis of Fig. 1 according to the original publication year. We notice that the accuracy does not monotonically increase throughout the years, as it would be desirable.

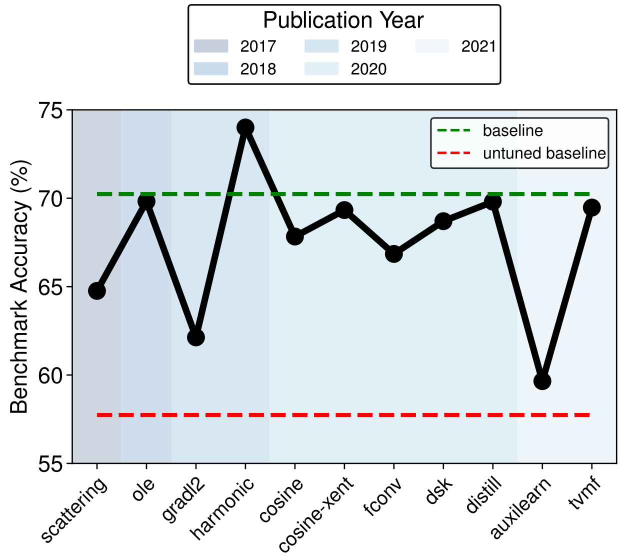


FIGURE 1. Accuracy of state-of-the-art methods and baselines on the proposed benchmark. The untuned baseline (red dashed line) is trained with default hyper-parameters *i.e.*, a learning rate of 0.1, and weight decay of 10^{-4}. Conversely, for all other methods, including the baseline (green dashed line), we performed hyper-parameter optimization. Methods are ordered on the x-axis according to their publication year.

Hence, we learn that default hyper-parameters found in the image classification literature are inadequate in data-deficient scenarios and should be substituted with properly tuned configurations, *e.g.*, more aggressive regularization for baseline classifiers. We identify the lack of such hyper-parameter tuning and comprehensive comparisons in the existing literature as the cause for the illusion of frequent progress. In contrast, only a single method analyzed in our study is able to outperform the baseline consistently in a realistic setting.

First, we differentiate the faced learning settings from related research areas in Section II. We then dive deeper into image classification from small datasets and review the existing literature in this field in Section III. Afterward, we describe our proposed benchmark, starting with the methods selected for the comparison in Section IV and datasets in Section V. Our experimental setup and training procedure are detailed in Section VI and the results are presented in Section VII. In Section VIII, we discuss potential limitations of our comparison. Section IX summarizes the conclusions from our study.

II. RELATED RESEARCH AREAS

To avoid potential sources of confusion, we will first give a brief description of research areas that are related to *learning from small datasets* but different in crucial aspects.

Transfer learning is a well-established approach that uses knowledge from a previous task to solve a secondary, generally related task [43], [60]. Often, the target task has a much smaller number of training examples, hence, fine-tuning a pre-trained network has benefits in terms of recognition performance. Note that this area should not be included in the literature on image classification with small datasets, as the pre-training is performed on large image databases such as ImageNet. This is not possible in some scenarios, especially when the type of the input data (*e.g.*, multispectral images from satellites) is different from RGB.

Domain adaptation is a subfield of *transfer learning* that assumes related source and target tasks, with the latter undergoing a distributional shift. Typically, the target task has only unlabeled data or a few annotated pairs. We refer the reader to [58] for a survey on this topic. Similar to the previous paradigm, *domain adaptation* also uses knowledge extraction from a data-rich source task.

Few-shot learning is a domain that has received considerable attention over the past years [23], [59]. The goal of this approach is to train a model that learns to recognize similarities and in turn perform tasks in data-poor target domains, including scenarios with only 1 or 5 samples per class. While the goal of *few-shot learning* overlaps with that of *learning from small datasets*, their implementations practically differ. *Few-shot learning* relies on a qualitatively rich base set of annotated pairs, from which it can *meta-learn* more general representations that are then used to solve the few-shot task. On the other hand, *learning from small datasets* uses a very modest training set to learn from that is slightly larger than a few shots, but does not have access to any large-scale pre-training data.

Weakly supervised learning deals with training datasets with few, noisy or inconsistent labels [69]. Moreover, in this domain, there are no assumptions about the dimension of the training dataset. There could be samples for which there is no training label, but which are still available to a semi-supervised algorithm. In contrast, *learning from small datasets* assumes that the correct label is available for each member of the small training dataset.

Long-tailed recognition, also known as *unbalanced classification*, is a largely researched area, as high-class imbalance naturally occurs in many classification problems [25]. In this scenario, the learner is regularized to not only learn effective representations to classify the majority classes but also correctly recognize the minority classes [14]. The number of samples in the tail of the training distribution is comparable in size to the dimensions of the datasets used in *classification with small data*. However, the latter excludes the existence of other classes with a large number of samples, which would contribute to the learning of general representations.

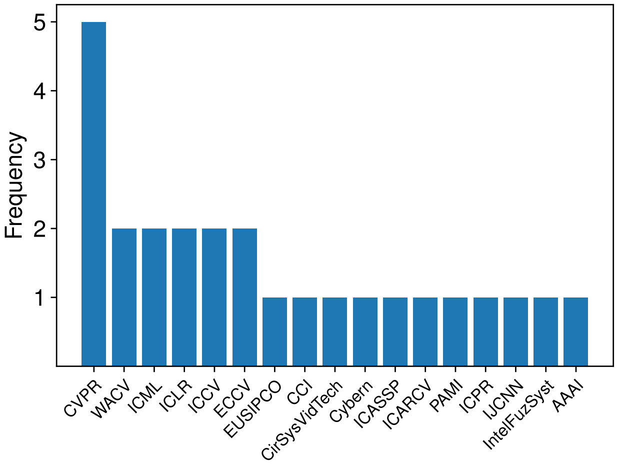


FIGURE 2. Distribution of publication venues concerning the reviewed body of literature.

III. LITERATURE REVIEW

In this section, we present the body of literature that we found after a careful search for image classification methods with small datasets. Moreover, we propose a classification taxonomy to organize the relevant research directions that have been explored so far.

A. SEARCH

We included in our collection mainly articles that fulfill two criteria: First, they should have been peer-reviewed at renowned conferences and journals. Second, they propose an approach for specifically tackling the problem of learning from a small sample. To verify the latter, we checked whether the experiments described in each candidate paper were executed on small or sub-sampled versions of popular image classification datasets.

To find relevant papers, we searched popular search engines and archives using keyword arguments that strongly match the features of this research domain such as *data efficiency*, *small data*, *small datasets* and *data-efficient*. Along with that, we also used direct paper references as a channel to find additional connections.

As a result of our search, we found 26 articles published between 2015 and 2021. Five of these works are published in journals while the rest have been presented at conferences or workshops. Figure 2 shows the distribution of articles among venues. From the figure, it stands out that computer vision conferences (*e.g.*, CVPR, ICCV, ECCV, WACV) are the venues where the community is more inclined to publish. Instead, machine learning (*e.g.*, ICLR and ICML) and signal processing conferences are slightly behind in terms of preferences.

B. MAP

To provide a clear visual picture of the literature and extract additional insights on the current status of the research

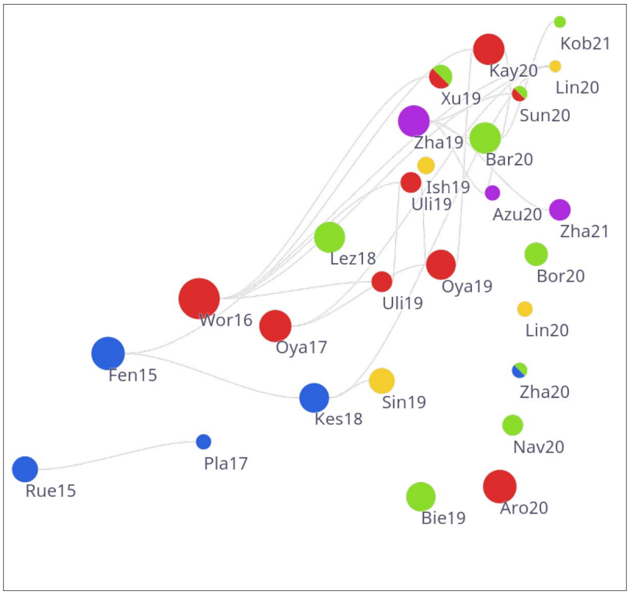


FIGURE 3. **Visualization of relative connections within the reported body of literature. Works are shown chronologically along the x-axis and distributed along the y-axis for better readability. Colors represent the taxonomic classes *architecture* (🔴), *cost function* (🟢), *data augmentation* (🟣), *latent augmentation* (🟡), and *warm-starting* (🔵).**

domain, we employ the online tool Litmaps[1] to generate a literature map of the state-of-the-art methods for image classification with small datasets (Fig. 3).

Each circle on this map represents a paper, denoted with a short keyword composed of part of the first-author name and the year of publication. The radius of the circle is proportional to the total number of citations and the colors represent the taxonomic classes to which the paper has been assigned by our breakdown. Therefore, circles of the same color represent articles that proposed methods belonging to the same methodological approach. More details about the taxonomy will follow in the next sub-section. Furthermore, articles are ordered along the x-axis according to publication time in ascending order. Connections between circles indicate references.

From the literature map, two things seem evident: the domain has gained increasing interest in the recent past and exhibits low connectivity in terms of references. The first fact, undoubtedly positive, is evincible from the growth of circles in the rightmost part of the map. While only 7 papers appeared between 2015 and 2018, the number of publications has grown to 19 in the subsequent three years. The second factor, which is instead negative, regards the fragmentation of the current state of the art. We notice that different papers published in 2020 or 2021 do not reference any of the older papers (bottom right corner of Fig. 3) indicating that a comparison between newly proposed approaches and the existing state of the art is often missing. Moreover, articles that are graphed to the upper part of the map, despite having higher connectivity, still did not reference multiple previous works. For instance, the most cited work by other members of the map, denoted with *Wor16* and corresponding to [61], has only been referenced by 5 subsequent papers. In response to the aforementioned connectivity issue of the current literature, we propose and organize the largest collection of works related to image classification with small datasets.

C. TAXONOMY

In this section, we categorize current methods into a methodological taxonomy that well fits the existing literature. Precisely, we distinguish five families of approaches, depending on how the model is regularized:

- *architecture* (🔴)
- *cost function* (🟢)
- *data augmentation* (🟣)
- *latent augmentation* (🟡)
- *warm-starting* (🔵)

We briefly illustrate the intrinsic characteristics of each taxonomic category and describe the papers belonging to each one. Notice that our proposed taxonomy classifies methods along independent axes, but it is also possible that some papers proposed a combination of multiple components falling into different families. A trivial example could be the proposal of a novel architecture trained with a new loss function.

1) ARCHITECTURE (🔴)

This category includes all possible modifications to standard network architectures at all topological levels. This covers changes applied to network layers, blocks, stages, etc. Reasonably, this is a quite general class that is composed of multiple sub-categories and contains a significant number of works in our collection (7).

An important contribution to this family derives from those methods relying on *geometric priors i.e.*, techniques that use intrinsic and geometric approaches coming from the signal-processing literature. We include the use of pre-set, fixed filters based on wavelet transformations [41], [42] or discrete cosine transform [55], [56]. Also, invariance to rotation and translation are desirable properties that have been embedded into CNNs through the use of steerable filters or circular harmonics [61].

Other architectural changes aim for invariance with respect to input transformations without employing mathematics from signal processing. For instance, F-Conv [26] uses an alternative padding strategy to decrease image-boundary effects and improve translation invariance. Xu *et al.* [62] introduced an alternative convolution block to favor the learning of scale-invariant representations. Similarly, Sun *et al.* [52] modified the standard residual block and proposed a multi-branch structure with anti-aliasing modules and selective kernels. Finally, Arora *et al.* [1] employ neural tangent kernels, which are equivalent to infinitely wide networks.

[1] https://www.litmaps.co/

2) COST FUNCTION (●)

Networks are optimized to minimize a so-called *cost function* to learn the target task. Cost functions are generally composed of multiple items, including different regularization or penalty terms (*e.g.*, weight decay). We prefer to employ the term *cost* instead of *loss* since, colloquially, the latter is widely used to exclusively indicate the error between predictions and targets without considering other terms. We notice that this family of regularization is the most popular within this community with 9 different contributions.

Two works proposed losses based on cosine similarity to regularize network training [3], [29], while Sun *et al.* [52] employed a three-term cost function including a change of the standard cross-entropy to a combination of itself and the cosine loss. Xu *et al.* [62] hand-crafted a rotation-invariant regularizer based on prior knowledge, which is added to the cost function as a penalty term. Bietti *et al.* [5] tested multiple existing regularization principles (*i.e.*, gradient penalties, spectral norms) and also proposed new regularization penalties for learning from small datasets. Moreover, Lezama *et al.* [32] conceived a geometric loss term based on an orthogonal low-rank embedding that can be plugged into any cost function and encourages embeddings for different classes to be orthogonal. Bornschein *et al.* [7] calibrate the cross-entropy loss using a scalar temperature parameter, which is optimized alternatingly on a validation set. Navon *et al.* [40] use implicit differentiation to either learn how to combine multiple losses or predict auxiliary targets. Also overlapping with this category is the work of Zhao and Wen [66], who employ a two-stage training using a modified version of the contrastive loss for self-supervised representation learning and a distillation loss to regularize the features learned by the final classifier.

3) DATA AUGMENTATION (●)

All methods that increase the size of the training dataset reside in this category. For instance, standard data augmentation (*i.e.*, cropping, flipping transformations, etc.) is a classical strategy belonging to this family. We also include in this category the coupling of generative models with classifiers since, the latter, can be used to synthesize additional examples and improve classification accuracy. Data augmentation techniques have received plenty of attention in the deep learning literature [49] and are based on simple tricks [68] or more complex automated strategies [12], [13].

Surprisingly, in our literature collection, we only found 3 works proposing data augmentation approaches specifically designed for the small-sample regime. One common approach consists in deep adversarial augmentation [64], [65] and the other one on generative latent implicit conditional optimization [2].

4) LATENT AUGMENTATION (●)

Stochastic or adversarial transformations applied to features inside networks constitute the main building block of this family of regularizers.

Ishii and Sato [24] adversarially augment features at randomly selected hidden layers by adding small perturbations to the original features extracted from training data. Lin *et al.* [35], [36] propose a framework that regularizes the classifier by sampling latent variables encoded in Gaussian distributions. Finally, Keshari *et al.* [27] introduce a more advanced dropout policy by measuring the strength of each node.

5) WARM-STARTING (●)

This class of approaches employs algorithmic schemes to initialize the classifier with weights that favor better learning on small datasets. We assume that warm-starting may happen a single time or multiple times in the training process. For instance, the *self-supervised* paradigm belongs to this family since encoders are first pre-trained on unsupervised tasks and then used to warm-start the final classifier.

The oldest work of our collection [47], and its successive implementation [44], trained networks in a layer-wise greedy manner, analogous to that used in unsupervised deep networks. In other words, each layer is initialized with the weights obtained from the precedent run. Similarly, Feng and Darrell [17] proposed a multi-step initialization algorithm that adapts the model complexity to the available training data and learns the structure of filters. Subsequent research decouples the structure and strength of convolutional filters to reduce the overall number of parameters by using a dictionary-based filter learning algorithm and, subsequently, standard training [28]. More recently, Zhao and Wen [66] used self-supervised learning to learn a general encoder followed by self-distillation coupled with the standard classification objective.

IV. BENCHMARK METHODS

In this section, we first present in more detail the approaches that we evaluated on our benchmark (Section IV-A). Then, we name the methods that we discarded and motivate such exclusions by describing their limiting factors (Section IV-B).

A. EVALUATED

Along with a baseline cross-entropy classifier, we evaluated ten specialized methods of our literature review. Out of the ten approaches, four belong to the taxonomic category *architecture* (●), five to *cost function* (●), and one to both *warm-starting* and *cost function* (●●). We describe the evaluated approaches in more detail in the following.

1) CROSS-ENTROPY LOSS

This is the widely used standard loss function for classification. We use it as a baseline with standard network architectures and optimization algorithms.

2) DEEP HYBRID NETWORKS (DHN) (●)

This approach represents one of the first attempts to incorporate pre-defined geometric priors via a hybrid approach of combining pre-defined and learned representations [41], [42]. According to the authors, decreasing the number of parameters to learn could make deep networks more data-efficient, especially in settings where the scarcity of data would not allow the learning of low-level feature extractors. Deep hybrid networks first perform a scattering transform on the input image generating feature maps and then apply standard convolutional blocks. The spatial scale of the scattering transform is controlled by the parameter $J \in \mathbb{N}$.

3) ORTHOGONAL LOW-RANK EMBEDDING (OLÉ) (●)

Lezama *et al.* [32] proposed this geometric loss that is intended to reduce intra-class variance and enforce inter-class margins for deep networks. This method collapses deep features into a learned linear subspace, or union of them, and inter-class subspaces are pushed to be as orthogonal as possible. The contribution of the low-rank embedding to the overall loss is weighted by the hyper-parameter λ_{ole}.

4) GRAD-ℓ_2 PENALTY (●)

This is a regularization strategy tested in the context of improving generalization on small datasets [5]. The ℓ_2 (squared) gradient norm is computed for the input samples and used as a penalty in the loss weighted by parameter λ_{grad}. Among many regularization approaches evaluated in by Bietti *et al.* [5], we have chosen the grad-ℓ_2 penalty because it was among the best-performing methods in the experiments with ResNet and sub-sampled versions of CIFAR-10.

5) COSINE LOSS (●)

Barz and Denzler [3] proposed this loss to decrease overfitting in problems with scarce data. Thanks to an ℓ_2 normalization of the learned feature space, the cosine loss is invariant against scaling of the network output and solely focuses on the directions of feature vectors instead of their magnitude. In contrast to the softmax function used with the cross-entropy loss, the cosine loss does not push the activations of the true class towards infinity, which is commonly considered a cause of overfitting [20], [53]. A further increase in performance was obtained by combining the cosine with the cross-entropy loss after an additional layer on top of the embeddings learned with the cosine loss.

6) HARMONIC NETWORKS (HN) (●)

HN uses a set of preset filters based on windowed cosine transform at several frequencies which are combined by learnable weights [55], [56]. Similar to hybrid networks, the idea of the harmonic block is to have a useful geometric prior that can help to avoid overfitting. Harmonic networks use Discrete Cosine Transform filters which have excellent energy compaction properties and are widely used for image compression.

7) FULL CONVOLUTION (F-CONV) (●)

Kayhan and Gemert [26] proposed F-Conv to improve the translation invariance of convolutional filters. Standard CNNs exploit image boundary effects and learn filters that can exploit the absolute spatial locations of objects in images. In contrast, full convolution applies each value in the filter to all values in the image. According to Kayhan and Gemert [26], improving translation invariance strengthens the visual inductive prior of convolution, leading to increased data efficiency in the small-data setting.

8) DUAL SELECTIVE KERNEL NETWORKS (DSKN) (●)

In this approach [52], the standard residual block is modified, keeping the skip connection, with two forward branches that use 1 × 1 convolutions, selective kernels [34] and an anti-aliasing module. To further regularize training, only one of the two branches is randomly selected in the forward and backward passes, while at inference, the two paths are weighted equally.

Besides the specialized network architecture, the original work uses a combination of three custom loss functions [52]. Despite best efforts, we were unable to derive the correct implementation from the ambiguous description of these loss functions in the paper. Therefore, we only use the DSKN architecture with the standard cross-entropy loss.

9) DISTILLING VISUAL PRIORS (DVP) (●●)

Zhao and Wen [66] introduce this two-stage framework that, firstly, learns a teacher model via self-supervised learning using the popular MoCo approach [10], and secondly, distills the representations into a student classifier using self-distillation [22]. A contribution of this work regards the novel margin loss implemented to better learn general representations under the data-deficient scenario. Hence, in addition to the hyper-parameters for MoCo, a margin λ_m needs to be set. In addition, for the second stage of training, λ_{dist} weights the contribution of the distillation loss to the overall cost function.

10) AUXILIARY LEARNING (AuxiLearn) (●)

AuxiLearn is a method for generating meaningful and novel auxiliary tasks [40]. This is achieved by training an auxiliary network to generate auxiliary labels while training another, primary network to learn both the original task and the auxiliary task. The objective is to push the representation of the primary network to generalize better on the main task by exploiting multi-task learning as a regularizer. To train this method, multiple hyper-parameters need to be set. First, the dimension of the auxiliary set which is a small percentage of the training data. Then, the strength of the auxiliary loss component along with the update-frequency of auxiliary gradients. Finally, the type of the auxiliary network *i.e.*, linear or not linear, and, for the latter case, its depth and number of units per layer.

11) T-vMF SIMILARITY (●)

This similarity [29] is a generalization of the cosine similarity and was proposed to make modern CNNs more robust to some realistic learning situations such as class imbalance, few training samples, and noisy labels. As the name suggests, T-vMF Similarity is mainly based on the von Mises-Fisher distribution of directional statistics and built on top of the heavy-tailed student-t distribution.

The combination of these two ingredients provides high compactness in high-similarity regions and low similarity in heavy-tailed ones. The degree of compactness/dispersion of the similarity is controlled by the parameter κ.

B. DISCARDED

We now describe the approaches belonging to the literature overview that we discarded and provide the related reasons.

A group of papers does not provide the original implementation of the proposed methods. To avoid unreliable or wrong re-evaluations, we restricted our final choice to approaches for which source code was available. For this reason, we discarded two contributions from *architecture* (●) [1], [62], three from *cost function* (●) [7], [52], [62], two from *latent augmentation* (●) [24], [27], and four from *warm-starting* (●) [17], [28], [44], [47]. Only for DSKN [52], we were able to implement the proposed architecture by ourselves but unable to correctly derive the proposed loss function.

In a few cases, despite best efforts and the availability of source code, we were unable to reproduce a properly working implementation. Faced obstacles included divergence issues [35], [36] or very poor results [61], [64], [65]. Finally, one *data augmentation* method employs a pre-trained VGG network to compute a perceptual loss embedded in the proposed framework [2]. This approach does not fully respect the assumptions of deep learning from small data, which does not allow external data or pre-trained models. Furthermore, it can not be adapted straightforwardly to different data types (e.g., grayscale or multi-spectral images).

V. BENCHMARK DATASETS

Most works on deep learning from small datasets use custom sub-sampled versions of popular standard image classification benchmarks such as ImageNet [48] or CIFAR [31]. This is visible from Fig. 4, where we show the frequency of use of all datasets employed in the reviewed body of literature. CIFAR-10, ImageNet, and CIFAR-100 were utilized 14, 7, and 5 times, respectively, accounting for an overall fraction of ~40% of datasets together. It is also striking that ~50% of the articles carried out experiments on CIFAR-10, making it the most frequently used dataset in this community.

This limited variety bears the risk of overfitting research progress to individual datasets and the domain covered by them, in this case, photographs of natural scenes and everyday objects. In particular, this is not the domain typically dealt with in a small-data scenario, where specialized data that is difficult to obtain or annotate is in the focus. Additionally, very recent work showed that high performance on ImageNet does not necessarily correlate with high performance on other vision datasets [54].

Therefore, we compile a diverse benchmark consisting of five datasets from a variety of domains and with different data types and numbers of classes. We sub-sampled all datasets to fit the small-data regime, except for CUB [57], which was already small enough. By default, we aimed for 50 training images per class. To account for variance stemming from the sub-sampling operation, we employ 3 different sets of dataset splits. In other words, we sub-sample the original datasets three times, when possible, and train and evaluate methods on each sub-portion independently. Full training splits are only used for the final training and split into training (~ 60%) and validation sets (~ 40%) for hyper-parameter optimization. For testing the final models trained on the *trainval{i}* splits, with $i \in \{0, 1, 2\}$, we used official standard test datasets where they existed. Only for two datasets, namely EuroSAT [21] and ISIC 2018 [11], we had to create own test splits, *i.e.*, *test{i}*. Given the already small size of CUB, we could not vary the training splits but only the *train* and *val* ones. A summary of the dataset statistics is given in Table 1 and Fig. 5 shows examples from all datasets. In the following, we briefly describe each dataset used for our benchmark.

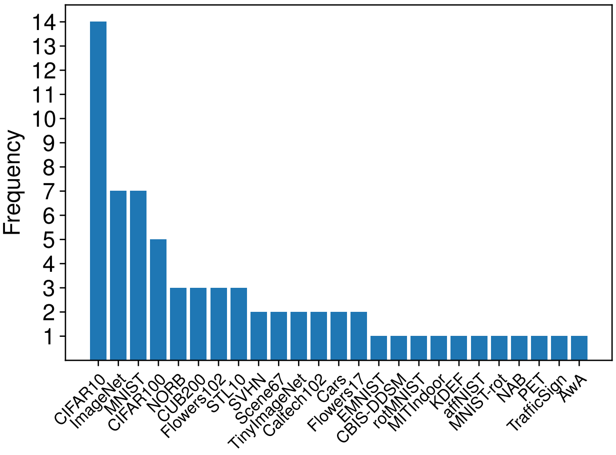


FIGURE 4. Frequency of datasets used used in the reviewed literature.

1) ciFAIR-10

Barz and Denzler proposed a variant [4] of the popular CIFAR-10 dataset [31], which comprises low-resolution images of size 32 × 32 from 10 different classes of everyday objects. To a large part, its popularity stems from the fact that the low image resolution allows for fast training of neural networks and hence rapid experimentation. However, the test dataset of CIFAR-10 contains about 3.3% duplicates from the training set [4], which can potentially bias the evaluation. The ciFAIR-10 dataset [4] provides a variant of the test set, where these duplicates have been replaced with new images from the same domain.

2) CALTECH-UCSD BIRDS-200-2011 (CUB)

CUB is a fine-grained dataset of 200 bird species [57]. Annotating this kind of images typically requires a domain expert and is hence costly. Therefore, the dataset is rather small and only comprises 30 training images per class. Pre-training on related large-scale datasets is hence the de-facto standard for CUB [15], [37], [50], [67], which makes it particularly interesting for research on sample-efficient methods closing the gap between training from scratch and pre-training.

3) ISIC 2018

ISIC 2018 is a medical dataset consisting of dermoscopic skin lesion images, annotated with one of seven possible skin disease types [11]. Since medical data usually requires costly expert annotations, this domain is important to be covered by a benchmark on data-efficient deep learning. Due to the small number of classes, we increase the number of images per class to 80 for this dataset, so that the size of the training set is more similar to our other datasets.

4) EuroSAT

This is a multispectral image dataset based on Sentinel-2 satellite images of size 64 × 64 covering 13 spectral bands [21]. Each image is annotated with one of ten land cover classes. This dataset does not only exhibit a substantial domain shift compared to standard pre-training datasets such as ImageNet but also a different number of input channels. This scenario renders the standard pre-training and fine-tuning procedure impossible.

Nevertheless, Helber *et al.* [21] adhere to this procedure by fine-tuning a CNN pre-trained on RGB images using different combinations of three out of the 13 channels of EuroSAT. Unsurprisingly, they find that the combination of the R, G, and B channels provides the best performance in this setting. This limitation to three channels due to pre-training is a waste of data and potential. In our experiments on a smaller subset of EuroSAT, we found that using all 13 channels increases the classification accuracy by 9.5% compared to the three RGB channels when training from scratch.

5) CLaMM

CLaMM is a dataset for **C**lassification of **La**tin **M**edieval **M**anuscripts [51]. It was originally used in the ICFHR 2016 Competition for Script Classification, where the task was to classify grayscale images of Latin scripts from handwritten books dated 500 C.E. to 1600 C.E. into one of twelve script style classes such as *Humanistic Cursive*, *Praegothica* etc. This domain is quite different from that of typical pre-training datasets such as ImageNet and one can barely expect any useful knowledge to be extracted from ImageNet about medieval documents. In addition, the standard pre-training and fine-tuning procedure would require converting the grayscale images to RGB for passing them through the pre-trained network, which incurs a waste of parameters.

VI. EXPERIMENTAL SETUP

In this section, we give an overview of the experimental pipeline that we followed for a fair evaluation of the aforementioned methods on the five datasets that constitute our benchmark.

A. EVALUATION METRICS

In our benchmark, we evaluate each method on each dataset with the widely used balanced classification accuracy. This metric is defined as the average per-class accuracy, *i.e.*, the average of the diagonal in the row-normalized confusion matrix. We turned our attention toward this metric since some datasets in our benchmark do not have balanced test sets. In any case, for balanced test sets, the balanced accuracy equals the standard classification accuracy.

Since our benchmark contains multiple datasets it is hard to directly make a comparison between two methods without computing an overall ranking. Therefore, for each method, we also compute the average balanced accuracy across all datasets to provide a simple and intuitive way to rank methods. Additionally, in this manner, future methods will be easily comparable with those already evaluated.

B. DATA PRE-PROCESSING AND AUGMENTATION

All input images were normalized by subtracting the channel-wise mean and dividing by the standard deviation computed on the *trainval* splits. We applied standard data augmentation policies with slightly varying configurations, adapted to the specific characteristics of each dataset and problem domain. Note that none of the currently re-evaluated methods in our benchmark had as original contribution a specialized data augmentation technique. Nothing prevents the use of a data-augmentation-based method from partaking in the benchmark.

For datasets with a small, fixed image resolution, *i.e.*, ciFAIR-10 and EuroSAT, we perform random shifting by 12.5% of the image size and horizontal flipping in 50% of the cases. For all other datasets, we apply scale augmentation using the `RandomResizedCrop` transform from PyTorch[2] as follows: A crop with a random aspect ratio drawn from $[\frac{3}{4}, \frac{4}{3}]$ and an area between $A_{\min}$ and 100% of the original image area is extracted from the image and then resized to 224×224 pixels. The minimum fraction $A_{\min}$ of the area was determined based on preliminary experiments to ensure that a sufficient part of the image remains visible. It therefore varies depending on the dataset: we use $A_{\min} = 20\%$ for CLaMM and $A_{\min} = 40\%$ for CUB and ISIC 2018.

For ISIC 2018 and EuroSAT, we furthermore perform random vertical flipping in addition to horizontal flipping, since these datasets are completely rotation-invariant and vertical reflection augments the training sets without drifting them away from the test distributions. On CLaMM, in contrast, we do not perform any flipping, since handwritten scripts are not invariant even against horizontal flipping.

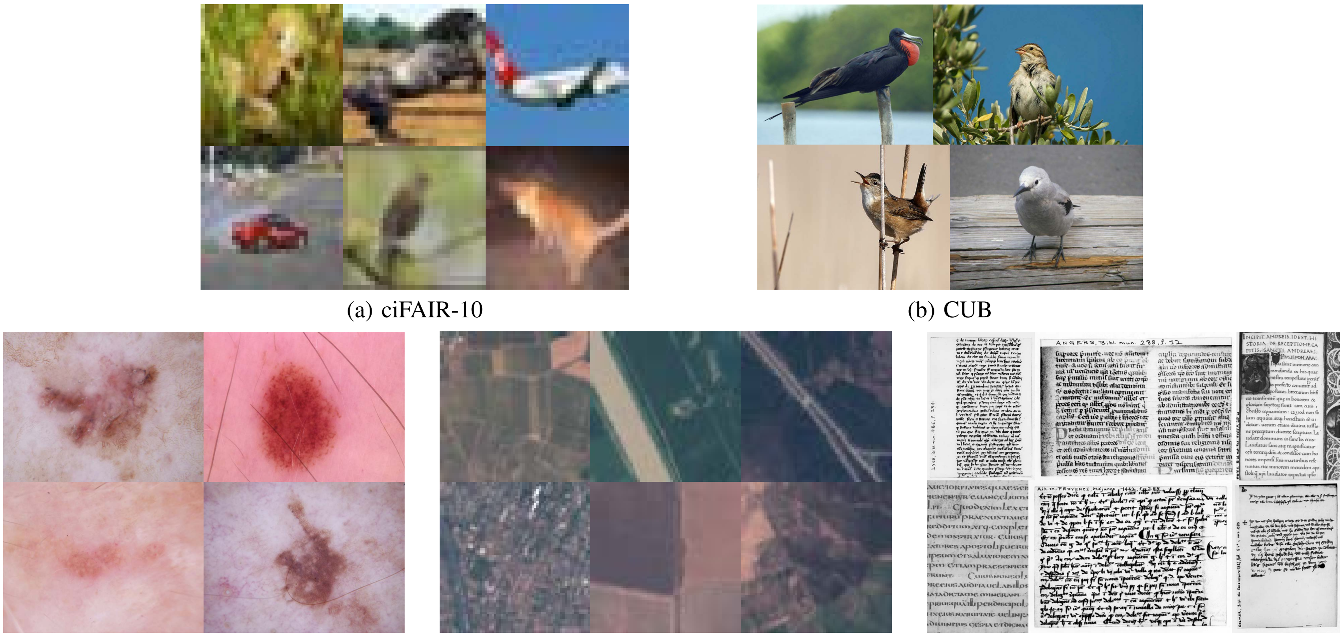
C. ARCHITECTURE AND OPTIMIZER

To perform a fair comparison, we use the same backbone CNN architecture for all methods. For ciFAIR-10, we employ a Wide Residual Network (WRN) [63], precisely

[2] https://pytorch.org/vision/stable/transforms.html#torchvision.transforms.RandomResizedCrop

TABLE 1. Datasets constituting our benchmark. To account for variance stemming from the sub-sampling operation, we employ three different training splits (except for CUB). On ISIC 2018 and EuroSAT, given the lack of a fixed testing set, we also vary such splits. The value of *i* refers to the identifier of the three splits, *i.e.*, $i \in \{0, 1, 2\}$.

Dataset	Classes	Imgs/Class	#Trainval	#Test	Train Splits	Test Splits	Problem Domain	Data Type
ciFAIR-10 [31, 4]	10	50	500	10,000	*trainval{i}*	*test*	Natural Images	RGB (32x32)
CUB [57]	200	30	5,994	5,794	*trainval*	*test*	Fine-Grained	RGB
ISIC 2018 [11]	7	80	560	1,944	*trainval{i}*	*test{i}*	Medical	RGB
EuroSAT [21]	10	50	500	19,500	*trainval{i}*	*test{i}*	Remote Sensing	Multispectral
CLaMM [51]	12	50	600	2,000	*trainval{i}*	*test*	Handwriting	Grayscale



(a) ciFAIR-10

(b) CUB

(c) ISIC 2018

(d) EuroSAT

(e) CLaMM

FIGURE 5. Example images from the datasets included in our benchmark. For EuroSAT, we only show the RGB bands.

WRN-16-8, which is widely used in the existing literature for data-efficient classification on CIFAR. For all other cases, the popular and well-established ResNet-50 (RN50) architecture [19] is used. Note that we made changes to the architecture when that was an original contribution of the paper, but all those changes were applied to the selected base architecture. Due to the high popularity of residual networks, the majority of the selected approaches were originally tested with a RN/WRN backbone. This fact allowed us to perform a straightforward porting of the network setup, when necessary.

We furthermore employ a common optimizer and training schedule across all methods and datasets to avoid any kind of optimization bias. We use standard stochastic gradient descent (SGD) with a momentum of 0.9, weight decay, and a cosine annealing learning rate schedule [39], which reduces the learning rate smoothly during the training process. The initial learning rate and the weight decay factor are optimized for each method and dataset individually, together with any method-specific hyper-parameters as detailed in the next subsection. The total number of training epochs for each dataset was chosen according to preliminary experiments.

D. HYPER-PARAMETER OPTIMIZATION

Careful tuning of hyper-parameters, as one would do in practice, is crucial and can have a considerable impact on the final performance [6] (empirical proofs in Section VII).

For our benchmark, we hence first tune the hyper-parameters of each method on each individual dataset using the training and validation splits, which are disjoint from the test sets used for final performance evaluation (see Section V). Since for each dataset we have three different sets of training splits, we perform three hyper-parameter optimization runs. Only on CUB, we perform the three searches on the unique available training split. For any method, we tune the initial learning rate and weight decay, sampled from a log-uniform space, as well as the batch size, chosen from a pre-defined set. Details about the search space are provided in Table 2. In addition to these general hyper-parameters, any method-specific hyper-parameters are tuned as well simultaneously, considering the boundaries used in the original paper, if applicable, or lower and upper bounds estimated by ourselves.

TABLE 2. Summary of hyper-parameters searched/used with ASHA [33]. Method specific hyper-parameters were included in the search space but not included in this table due to space limitations. An epoch number in parentheses means that a higher number of epochs was used for the final training than for the hyper-parameter optimization.

Hyper-Parameter	ciFAIR-10	CUB	ISIC 2018	EuroSAT	CLaMM
Learning Rate	`loguniform`(1e-4, 0.1)				
Weight Decay	`loguniform`(1e-5, 0.1)				
Batch Size	{10, 25, 50}	{8, 16, 32}	{8, 16, 32}	{10, 25, 50}	{8, 16, 32}
Epochs	500	200	500	500	500
HPO Trials	250	100	100	250	100
Grace Period	50	10	25	25	25

For selecting hyper-parameter configurations to be tested and scheduling experiments, we employ Asynchronous HyperBand with Successive Halving (ASHA) [33] as implemented in the Ray library.[3] This search algorithm exploits parallelism and aggressive early-stopping to tackle large-scale hyper-parameter optimization problems. Trials are evaluated and stopped based on their accuracy on the validation split.

Two main parameters need to be configured for the ASHA algorithm: the number of trials and the grace period. The former controls the number of hyper-parameter configurations tried in total while the latter the minimum time after which a trial can be stopped. Since the number of trials corresponds to the time budget available for HPO, we choose larger values for smaller datasets, where training is faster. The grace period, on the other hand, should be large enough to allow for a sufficient number of training iterations before comparing trials. Therefore, we choose larger grace periods for smaller datasets, where a single epoch comprises fewer training iterations. The exact values for each dataset as well as the total number of training epochs can be found in Table 2. These values were determined based on preliminary experiments with the cross-entropy baseline.

E. FINAL TRAINING AND EVALUATION

After having completed HPO for each split using the procedure described above, we train the classifiers with the three determined configurations on the *trainval{i}* splits and evaluate the balanced classification accuracy on the test splits. To account for the effect of random initialization, this training is repeated ten times. Therefore, the final performance is the balanced average accuracy over 30 repetitions with each set of 10 repetitions initialized with the parameters found through HPO runs on the three sets of splits.

F. METHOD-SPECIFIC IMPLEMENTATION DETAILS

Two methods required individual modifications to the general training and evaluation pipeline described above, which we describe in the following.

1) GRAD-ℓ_2 PENALTY

We disabled weight decay because this method is an alternative regularizer and considered as mutually exclusive with weight decay in the original paper [5]. Moreover, in contrast to the original implementation, we enabled the use of batch normalization since, without this component, we obtained extremely low results in preliminary experiments.

2) DISTILLING VISUAL PRIORS

DVP [66] required several adaptations due to its two-step training process.

First, task-agnostic features are learned using self-supervision. Then, the resulting model is used as a teacher for a student model trained for classification. Thus, not only the training of the final classifier needs tuned hyper-parameters but the pre-training step as well. For evaluating the quality of the pre-trained models during HPO, we attach a linear classification head on top of the learned representations but do not back-propagate through it. The contrastive pre-training criterion furthermore requires larger batch sizes than we usually use and longer training schedules. We hence select batch sizes from {64, 128, 256} and increase the number of epochs until the accuracy of the additional classification head converges. This resulted in a training schedule of 2,000 epochs for ciFAIR-10, 6,400 epochs for CUB, and 16,000 epochs for ISIC 2018, CLaMM, and EuroSAT. These numbers are one to two orders of magnitude larger than our usual training durations used for all other methods.

After we found hyper-parameters for the self-supervised pre-training, we trained a single model on the training split, which served as a basis for the subsequent HPO for distilling the learned knowledge into the student classifier. For this step, we used the same batch sizes and numbers of epochs as usual. Finally, we trained 30 self-supervised models, on the combined training and validation data and subsequently used each of them as a teacher for 30 other student models performing the classification task. As for the other methods, HPO and the final training of each group of 10 are performed on a different set of dataset splits.

[3] https://docs.ray.io/en/master/tune/

TABLE 3. Average balanced classification accuracy in % over 30 repetitions for each task and across all tasks. The best value per dataset is highlighted in bold font. Numbers in italic font indicate that the result is not significantly worse than the best one on a significance level of 5%. Colored dots represent the taxonomic classes to which the approaches belong, *i.e., architecture* (●), *cost function* (●), and *warm-starting* (●).

Method	ciFAIR-10	CUB	ISIC 2018	EuroSAT	CLaMM	Average
Cross-Entropy Loss (untuned baseline)	46.52	53.24	57.33	83.48	48.12	57.74
Cross-Entropy Loss (baseline)	55.18	70.79	64.49	90.58	70.15	70.24
Deep Hybrid Networks (●) [41, 42]	53.84	55.37	62.06	88.77	63.75	64.76
OLÉ (●) [32]	55.19	66.55	62.80	90.29	74.28	69.82
Grad-ℓ_2 Penalty (●) [5]	51.90	51.94	60.21	81.50	65.10	62.13
Cosine Loss (●) [3]	52.39	66.94	62.42	88.53	68.89	67.83
Cosine + Cross-Entropy Loss (●) [3]	52.77	70.43	63.17	89.65	70.64	69.33
Harmonic Networks (●) [55, 56]	**58.00**	**73.07**	**69.70**	**91.98**	**77.25**	**74.00**
Full Convolution (●) [26]	54.64	63.74	57.34	89.47	69.06	66.85
Dual Selective Kernel Networks (●) [52]	53.84	69.75	63.41	91.09	65.43	68.70
Distilling Visual Priors (●●) [66]	*57.80*	70.81	62.39	88.96	69.07	69.81
Auxiliary Learning (●) [40]	51.84	43.57	61.70	80.92	60.24	59.65
T-vMF Similarity (●) [29]	56.75	68.19	64.60	88.50	69.33	69.47

VII. RESULTS

In the following, we first present the main results of our benchmark in Section VII-A, followed by comparisons in terms of training speed and memory requirements (SectionVII-B), and between taxonomic classes (Section VII-C). We then show an analysis concerning an additional high-resolution fine-tuning step in Section VII-D. In Section VII-E, we compare benchmarked methods with transfer learning. Next, in Section VII-F, we provide evidence that published baselines are underperforming. The importance of hyper-parameter optimization is discussed in Section VII-G. Finally, in Section VII-H we provide additional insights on the tuned hyper-parameters of the cross-entropy baseline.

A. BENCHMARK FOR IMAGE CLASSIFICATION WITH SMALL DATASETS

Table 3 presents the average balanced classification accuracy over 30 repetitions for all methods and datasets. We performed Welch's t-test to assess the significance of the advantage of the best method per dataset in comparison to all others. All results but one are significantly worse on a level of 5% than the best method on the respective task.

Harmonic Networks [55], [56] clearly win the benchmark by providing top performance on all datasets. However, an even more interesting finding of this study is that the default cross-entropy loss is highly competitive when tuned carefully. In terms of average balanced accuracy across all datasets, the baseline scores 70.24%, which is only clearly below the accuracy scored by Harmonic Networks (74%) but superior than or on par with the results of the remaining methods. To better characterize the impact of hyper-parameter optimization (HPO) on the baseline, we also perform experiments with the cross-entropy loss and default hyper-parameters (untuned baseline in Table 3). By default hyper-parameters, we mean the learning rate and weight decay that are usually employed in data-rich scenarios, *i.e.*, 0.1 for the first and 10^{-4} for the latter. Without HPO there is a substantial degradation of the baseline performance resulting in a very low average balanced accuracy across all tasks. The untuned baseline only scores 57.74% which is ∼13 percentage points below the tuned baseline and clearly outperformed by all other specialized methods.

A relatively large group of state-of-the-art approaches, *i.e.*, OLÉ [32], Cosine + Cross-Entropy Loss [3], Dual Selective Kernel Networks [52], Distilling Visual Priors [66], and T-vMF Similarity [29], obtains an overall recognition performance comparable to the one of the baseline. However, the finding that the vast majority of recent methods for image classification with small datasets does not exceed the performance of the baseline is sobering. We attribute this to the fact that the importance of hyper-parameter optimization is immensely underestimated, resulting in misleading comparisons of novel approaches with weak and underperforming baselines. We will investigate this hypothesis further in later sub-sections.

B. TIME AND MEMORY REQUIREMENTS

In the previous sub-section, we compared the benchmarked methods in terms of accuracy. Further important factors for choosing a method from a practical point of view are the training speed and memory requirements. We measured these two metrics on an Nvidia V100 GPU with 16 GB of memory using PyTorch 1.7.

For measuring training throughput, we run each training method for 10 epochs on the CUB dataset and count

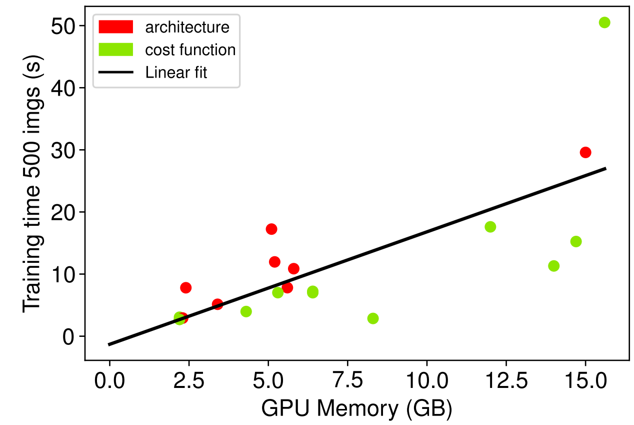


FIGURE 6. Comparison between *architecture* (●) and *cost function* (●) methods in terms of GPU memory usage and training time (500 images). Training times are derived from the training speeds shown in Table 4.

TABLE 4. Comparison between methods in terms of training speed (images/second) and GPU memory requirements (GB) for input resolutions of 224 × 224 and 448 × 448 on an NVIDIA V100. The best value per column is reported in bold font. Colored dots represent the taxonomic classes to which the approaches belong, *i.e.*, *architecture* (●), *cost function* (●), and *warm-starting* (●).

Method	Images/sec (↑)		GPU Mem. (↓)	
	224	448	224	448
Cross-Entropy Loss (baseline)	**189.0**	**72.2**	**2.2**	6.4
Deep Hybrid Networks (●)	64.1	29.0	2.4	**5.1**
OLÉ (●)	163.7	69.2	**2.2**	6.4
Grad-ℓ_2 Penalty (●)	28.4	9.9	12.0	15.6
Cosine + Cross-Entropy Loss (●)	183.8	71.4	**2.2**	6.4
Harmonic Networks (●)	171.0	64.0	2.3	5.6
Full Convolution (●)	97.3	46.0	3.4	5.8
Dual Selective Kernel Networks (●)	41.8	16.9	5.2	15.0
Distilling Visual Priors (●●)				
Pre-Training	175.0	–	8.3	–
Fine-Tuning	125.9	44.2	4.3	14.0
Auxiliary Learning (●)	32.8	–	14.7	30.3
T-vMF Similarity (●)	184.8	71.3	**2.2**	5.3

the number of images processed per second during the last 5 epochs. For a fair comparison, we used a constant batch size of 8 for all methods, which was the most common batch size found by HPO. The only exception is the self-supervised pre-training step of Distilling Visual Priors, which depends on sufficiently large batch sizes. For this method, we used a batch size of 64, as determined by HPO.

In Table 4, we report training throughput and GPU memory requirements for input resolutions of 224 × 224 and 448 × 448. We also report the latter resolution to gain additional insights regarding how computational requirements scale with larger inputs. As we will see in Section VII-D, some tasks benefit from larger resolutions.

Five among the top methods concerning recognition accuracy, namely Harmonic Networks, the baseline, OLÉ, Cosine + Cross-Entropy Loss, and T-vMF Similarity, are also among the fastest and most memory-efficient ones. The slowest and most memory-consuming methods are Auxiliary Learning and Grad-ℓ_2 Penalty, which take 5-7 times longer to train than the baseline and consume 5-7 times more memory. In addition, they are the worst-performing methods in terms of classification accuracy. For Auxiliary Learning, the memory of the V100 was insufficient for the higher resolution of 448 × 448, wherefore we measured the memory consumption in this case on an A100 GPU with 40 GB of memory. Due to the different compute hardware, we do not report the training throughput in this case.

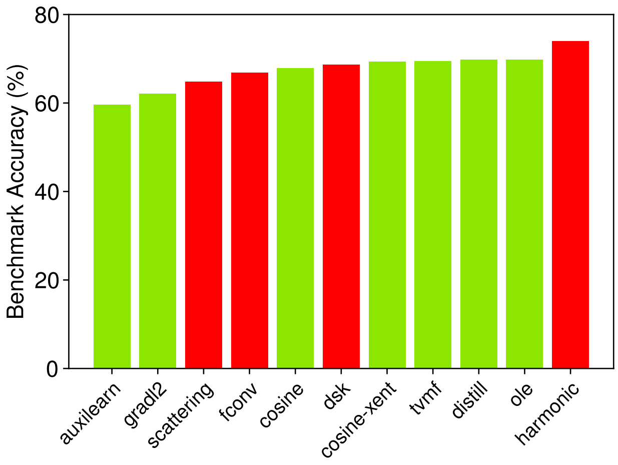


FIGURE 7. Comparison between *architecture* (●) and *cost function* (●) methods in terms of recognition performance. We order methods along the x-axis from the least to the best performing one.

C. COMPARISON BETWEEN TAXONOMIC CLASSES

In this paragraph, we briefly discuss the performance difference among the two largest taxonomic groups evaluated on our benchmark, *i.e.*, the *architecture* (●) and *cost function* (●) classes.

First, we compare these two classes considering the recognition performance over the benchmark. In Fig. 7, we show the results of each method on the full benchmark in ascending order. We notice that colored bars follow an alternating pattern, i.e., the worst two methods belong to the *cost function* class, then *architecture*, and so on. Hence, there is no clear winner considering this evaluation metric. Harmonic Networks [55], [56], which belong to the *architecture* class, however, remain the winners of our benchmark.

Second, we discuss the differences concerning computational requirements and show the comparison in Fig. 6. We plot the training time needed to process 500 images vs the GPU memory usage. Training times are derived from the speeds shown in Table 4. We opted for 500 images because they well represent the time to perform a single epoch on a small dataset (e.g., the size of our ciFAIR-10 training splits). We notice that the two taxonomic classes have similar computational requirements but lie on opposite sides of a linear fit. The majority of red dots are above, while the green ones are below. Hence, *architecture* methods seem to be slower to train while *cost function* ones to require more GPU memory. This is reasonable since the introduction of modified modules in the architecture may induce more complex processing and hence lower training speeds. Modified losses may instead require gradient updates that consume more memory.

D. HIGH-RESOLUTION FINE-TUNING

In the field of fine-grained visual recognition, it is common practice to increase the input image resolution from 224 × 224 pixels usually used for pre-training to 448 × 448 pixels [15], [37]. Therefore, in these experiments, we test such an additional step for the CUB and CLaMM datasets. Initially, we also experimented with high-resolution fine-tuning for ISIC 2018 but did not observe any substantial advantage. We run the test on ten models that have been trained on the training set with the standard resolution. Such networks are fine-tuned with the double input image size of 448 × 448, which is, consequently, also used for evaluation. For this high-resolution fine-tuning step, we use the same hyper-parameters and number of epochs as for the initial standard-resolution training.

OLÉ [32], Full Convolution [26], and T-vMF Similarity [29] benefit the most from the high-resolution fine-tuning step on CUB and CLaMM, which improves the balanced accuracy by 18%-23% on CLaMM and 15%-18% on CUB for these methods. Deep Hybrid Networks [41], [42], in contrast, take the least advantage from the higher resolution. On CLaMM, they only gain ∼3 percentage points. The baseline is in-between with an improvement of 8% on CLaMM and 13% on CUB. For AuxiLearn [40], we omitted the high-resolution fine-tuning step since the computational cost for this method becomes too high with the increase of resolution (see Section VII-B). Training would have taken several months to complete, which is by no means advantageous.

We believe that this analysis turns out to be useful for practitioners facing problems with data scarcity in fine-grained scopes. High-resolution fine-tuning can be performed to raise the recognition performance of the classifier through a relatively straightforward additional training step.

E. STRENGTHS AND LIMITS OF TRANSFER LEARNING

In scenarios where it is possible, so-called *transfer learning* by pre-training on a large related dataset and fine-tuning on the target data is a popular technique. It does not qualify for our benchmark due to the use of external data, but a comparison with this approach allows us to understand its benefits and limitations. ImageNet pre-training particularly benefits down-stream tasks whose labels are well-represented in ImageNet (*e.g.*, CUB) [30]. Yet, we show that the outcome changes as the target domain moves away from the one of natural images.

In this set of experiments, we fine-tune a ResNet-50 pre-trained on ImageNet-1k on the datasets of our benchmark which do not contain natural images, *i.e.*, ISIC 2018, EuroSAT, and CLaMM using the standard cross-entropy loss and compare it with the best small-data method in Table 6. The hyper-parameters for the fine-tuning step are tuned in the same manner as for our benchmark (see Section VI-D). For EuroSAT, which is a multispectral dataset, we have to restrict the fine-tuning to the R, G, and B channels to be feasible.

TABLE 5. Impact of high-resolution fine-tuning step on two fine-grained datasets, *i.e.*, CUB and CLaMM. We report the average balanced classification accuracy in % over 10 repetitions. Colored dots represent the taxonomic classes to which the approaches belong, *i.e.*, *architecture* (🔴), *cost function* (🟢), and *warm-starting* (🔵).

Method	CUB		CLaMM	
448 × 448 fine-tuning	✗	✓	✗	✓
Cross-Entropy Loss (baseline)	71.44	80.57	75.34	81.53
Deep Hybrid Networks (🔴)	52.54	58.57	65.74	68.00
OLÉ (🟢)	63.32	73.33	71.42	83.72
Grad-ℓ_2 Penalty (🟢)	51.94	61.04	65.10	76.95
Cosine Loss (🟢)	66.94	73.66	68.89	79.03
Cosine + Cross-Entropy Loss (🟢)	70.80	78.08	69.29	79.90
Harmonic Networks (🔴)	72.26	80.49	74.59	83.41
Full Convolution (🔴)	64.90	76.37	63.33	77.91
Dual Selective Kernel Networks (🔴)	71.02	78.84	61.51	63.45
Distilling Visual Priors (🔵🟢)	71.27	77.59	67.89	75.92
Auxiliary Learning (🟢)	59.68	—	43.61	—
T-vMF Similarity (🟢)	67.43	77.26	66.40	78.16

TABLE 6. Comparison between the best performance when training from scratch according to Table 3 and fine-tuning from weights pre-trained on ImageNet-1k. Numbers are the average balanced classification accuracy in % over 30 repetitions.

	ISIC 2018	EuroSAT	CLaMM
Best From Scratch	69.70	91.98	77.25
ImageNet Pre-Training	75.92	94.22	75.21
Absolute Gap	-6.22	-2.24	+2.04

Transfer learning has a clear advantage on the ISIC dataset. The latter, despite being from a different domain (medical), shares low-level textures and colors with natural images. On EuroSAT, which comprises satellite images, we note a smaller accuracy difference (2.24 percent points). Finally, on CLaMM (manuscript imagery) the domain shift is detrimental: training from scratch outperforms fine-tuning by 2 percent points. Here, an additional factor that might play a role besides the domain shift is the data modality, since CLaMM contains only grayscale images.

We learn from this analysis that *transfer learning* may only be applicable in data-deficient scenarios that share low-level features with natural images, but fail as the domain shift becomes more significant.

F. PUBLISHED BASELINES ARE UNDERPERFORMING

We show further evidence of why tuning the hyper-parameters and not neglecting the baseline in a small-data setting is fundamental for performing a fair comparison between different methods.

We analyzed the original results reported for the methods considered in our study and compare those that share a similar setup with the performance of our re-evaluation. Furthermore, we compare the performance of the baseline published in those works with ours. Note that due to the lack of a

TABLE 7. Summary of published/our results of the cross-entropy loss (left) and other methods (right) on similar setups.

Cross-Entropy Loss (baseline)					Other Methods				
Publication	Dataset	Network	Accuracy	Abs. Gap	Method	Dataset	Network	Accuracy	Abs. Gap
[41]	CIFAR-10	WRN-16-8	46.50		DHN [41]	CIFAR-10	WRN-16-8	54.70	
Ours	ciFAIR-10	WRN-16-8	55.18	+8.68	DHN (Ours)	ciFAIR-10	WRN-16-8	53.84	-0.86
[56]	CIFAR-10	WRN-16-8	52.20		HN [56]	CIFAR-10	WRN-16-8	58.40	
Ours	ciFAIR-10	WRN-16-8	55.18	+2.98	HN (Ours)	ciFAIR-10	WRN-16-8	58.00	-0.40
[40]	CUB	RN18	37.20		AuxiLearn [40]	CUB	RN18	44.50	
Ours	CUB	RN18	65.80	+28.60	AuxiLearn (Ours)	CUB	RN18	56.00	+11.50
[3]	CUB	RN50	51.92		Cosine Loss [3]	CUB	RN50	67.60	
Ours	CUB	RN50	70.79	+18.87	Cosine Loss (Ours)	CUB	RN50	66.94	-0.66

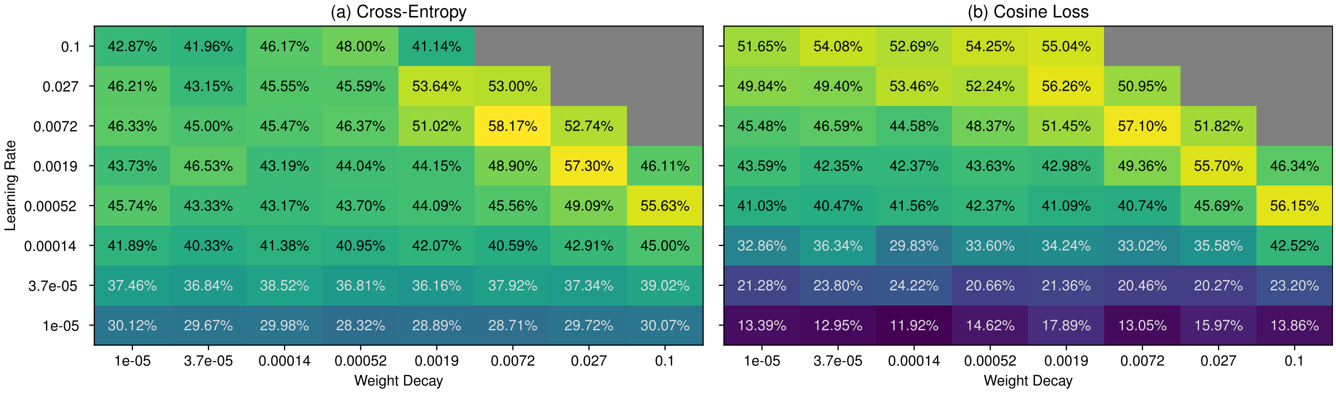


FIGURE 8. Classification accuracy obtained with standard cross-entropy and cosine loss [3] on ciFAIR-10 with 1% of the training data for different combinations of learning rate and weight decay. Gray configurations led to divergence.

standard benchmark and the common practice of randomly sub-sampling large datasets, we are unable to conduct a fair comparison with the same dataset splits, training procedure, etc. Still, our benchmark shares the base dataset and network architecture with the selected cases. Therefore, we believe that this analysis is suitable for supporting our point regarding the common practice of comparing tuned proposed methods with underperforming baselines. The results of this analysis are shown in Table 7.

Deep Hybrid Networks and Harmonic Networks were originally tested with a WRN-16-8 on CIFAR-10. Full Convolution and Cosine Loss employed RN50 on CUB. Since in the original Auxiliary Learning paper the authors trained a ResNet-18 (RN18) on CUB [40], we also perform an additional experiment with this smaller architecture to perform a fair comparison. The training set of CIFAR-10 was comprised of 50 images per class. Differently, experiments on CUB employed the full training set (*i.e.*, 30 images per class).

Our baseline clearly outperforms the published baselines by large margins (Table 7, left part). More precisely, our models surpass the published ones by ~ 9, ~ 3, ~ 29, and ~ 19 percentage points on the CIFAR and CUB setups. Recall also that the ciFAIR-10 test set is slightly harder than the CIFAR-10 one due to the removal of duplicates [4].

The picture looks different in the case of the proposed methods (Table 7, right part), where the difference between ours and the original results is sharply less evident. Our DHN and HN slightly underperform the original ones by ~ 1.0 and ~ 0.50 percentage points, respectively. However, this was expected due to the higher difficulty of ciFAIR-10. On CUB, our re-evaluation of the cosine loss scores an average balanced accuracy of 66.94% which is very close to the original 67.60%. Finally, our AuxiLearn model employing RN18 gains ~ 11 percentage points confirming once again that careful HPO can further boost the performance.

From this analysis, it seems clear that proposed methods are usually tuned to obtain an optimal or near-optimal result while baselines are trained with default hyper-parameters that have been found useful for large datasets but do not necessarily generalize to smaller ones.

G. IMPORTANCE OF HYPER-PARAMETER OPTIMIZATION

To further underpin the importance of hyper-parameter optimization, especially in a data-deficient setting, we perform a full grid search for combinations of learning rate and weight decay with a Wide ResNet architecture [63] trained on as few as 1% of the CIFAR-10 training data [31] and evaluated on the ciFAIR-10 test set [4]. We conduct this experiment with the standard cross-entropy loss and with the cosine loss [3], which was proposed as a loss function with a regularizing effect for better performance on small datasets.

TABLE 8. Hyper-parameters found with ASHA [33] for the cross-entropy baseline on one of the training splits.

Dataset	Batch Size	Learning Rate	Weight Decay
ciFAIR-10	10	4.55×10^{-3}	5.29×10^{-3}
CUB	8	2.44×10^{-3}	2.22×10^{-3}
ISIC 2018	8	0.69×10^{-3}	4.16×10^{-2}
EuroSAT	25	4.82×10^{-3}	6.31×10^{-2}
CLaMM	8	1.81×10^{-3}	1.81×10^{-2}

First, the results for standard cross-entropy shown in Fig. 8a illustrate that this baseline can substantially benefit from suitable HPO. Typical default hyper-parameters such as a learning rate of 0.1 and weight decay of 1×10^{-4} as used by [9] would achieve $\sim 46\%$ accuracy in this scenario, which is entire 12 percentage points below the optimal performance of $\sim 58\%$.

In comparison with the results for cosine loss shown in Fig. 8b, we observe that the cosine loss is less sensitive to changes of the weight decay factor but more sensitive to the learning rate. For the experiments in the original cosine-loss paper [3], the authors did perform HPO for both methods but only took the learning rate into account while keeping the weight decay fixed to a small constant. In this setting, the cosine loss can easily outperform cross-entropy because it has better chances with arbitrary weight decays. For 6 out of the 8 weight decay values we tested, cosine loss achieves better performance than cross-entropy. Only when both hyper-parameters are tuned can the cross-entropy baseline demonstrate its strength.

It is furthermore worth noting that the optimal weight decay in this data-deficient setting is rather large compared to usual defaults, which range between 1×10^{-5} and 1×10^{-4}. Such small training datasets apparently require much stronger regularization.

Moreover, we observe that the best performing hyper-parameter combinations are close to an area of the search space that results in divergence of the training procedure. This makes hyper-parameter optimization a particularly delicate endeavor.

H. TUNED HYPER-PARAMETERS

For reproducibility, but also to gain further insights into hyper-parameter optimization for small datasets, we show one of the three hyper-parameter combinations found during our searches for the cross-entropy baseline in Table 8.

We can observe that small batch sizes seem to be beneficial, despite the use of batch normalization. While the learning rate exhibits a rather small range of values from 0.7×10^{-3} to 7.4×10^{-3} across datasets and spans only one order of magnitude, weight decay varies within a range of two orders of magnitude from 4.1×10^{-4} to 1.8×10^{-2}.

Furthermore, learning rate and weight decay appear to be negatively correlated. Higher learning rates are usually accompanied by smaller weight decay factors. The same correlation can be observed in Fig. 8.

VIII. DISCUSSION

We designed a rigorous evaluation protocol for each method based on a common experimental setup in terms of base architecture, dataset splits, and optimization pipeline. However, we acknowledge that our study is not inclusive with respect to all possible aspects. In the following, we provide a list of the possible limitations of our benchmark along with explanations and arguments concerning each of them.

A. DATASETS

An extensive focus on individual benchmark datasets and even certain dataset splits bear the risk of adapting methods specifically to the test sets of these few datasets. To account for random variations caused, for instance, by data subsampling, we run our experiments on three independent dataset splits. To ensure the generalization of the tested methods across domains, our benchmark transcends the common datasets, *e.g.*, CIFAR, and incorporates four additional datasets with widely varying characteristics. These additions augment the generality of our benchmark, yet keep a balance between the spectrum of covered domains and the overall computation time needed to evaluate a method. However, given the ''living'' nature of our benchmark, we plan in the future to introduce domains spanning an even broader range of fields, data types, and applications to drive further progress toward small-sample learning methods.

B. BASE ARCHITECTURE

Concerning the base architecture employed, ResNet is not only a quite popular architecture in the image classification literature, but also in the image classification with small datasets community, as we saw from our literature analysis. We employed this network class to remain consistent with previous literature but we would not exclude an architectural bias for ResNets a priori. However, including multi-architecture evaluations in such a large benchmark like ours would incur high computational costs.

C. HYPER-PARAMETER SEARCH

The strong performance of our baseline and the comparison with published baselines in TAble 7 demonstrate the importance of thorough hyper-parameter optimization. Concerning this aspect, our benchmark is fair in the sense that all methods had the same budget (in terms of the number of HPO trials) and we tuned all their hyper-parameters jointly. However, comparing the grid-search results for the cosine loss on the ciFAIR-10 test set (see Fig. 8b) with the respective performance reported in Table 3 exhibits a gap of about 5 percentage points. Obviously, in this case, our HPO procedure did not find hyper-parameters on the validation sets that provide optimal performance on the test set. We conjecture two main reasons that may have caused this failure: the search algorithm could not find the optimal solutions or the evaluation performance obtained on the validation sets does not directly translate into optimal performance on the test set. Our correct practice of performing HPO on the few held-out samples allows us to gain useful insights into the realistic performance of this method in a practical setting. A generalization gap in the range between 3% and 15% is to be expected on the domain of CIFAR when replacing the test set [46]. In general, we should not consider the best accuracy in Fig. 8b as the optimal performance since we would be optimizing hyper-parameters on the test set.

To avoid the possibility of having methods "luckier" than others concerning HPO, for each training split, we run an independent hyper-parameter search. Due to the comparatively small size of the validation sets in our benchmark, unstable HPO is not completely unlikely. However, the validation sets cannot be much larger, since we operate in data-deficient settings and a sufficient number of samples needs to be available for the actual training. We hence argue that our solution to this issue, *i.e.*, averaging the results stemming from three groups of found hyper-parameters, improves the robustness of the results by considering the randomness of the search process. Clearly, the more precision is requested for the estimate, the more the cost for evaluating a method on our benchmark. We believe that increasing the number of HPO runs would exclude research groups without access to large clusters. We have shown in Table 7 that our results for four methods originally evaluated in similar settings are either on par with the performance reported in the original publication or even outperform it. This indicates that our searches found hyper-parameters as effective as the ones in the current literature.

IX. CONCLUSION

We presented the first comprehensive overview and dedicated benchmark for *deep learning from small datasets* in the context of image classification.

First, we carefully searched the literature for specialized methods applied to small-data tasks. We categorized this collection into a constructive taxonomy and provided an overview to consolidate this field, which is currently very fragmented.

Second, analyzing our literature review, we found that a common evaluation benchmark with fixed datasets, architectures, and training pipelines was lacking in the research domain. In addition, we found experimental evidence of weak baseline evaluations due to a lack of careful tuning. To address the urge for a fair comparison, we developed a benchmark consisting of five datasets from different domains and data types. Re-evaluating ten selected state-of-the-art methods led us to the surprising and sobering finding that standard cross-entropy loss is only surpassed by Harmonic Networks, and that performance growth is currently lacking in the literature.

In light of these results, we conclude that the importance of hyper-parameter optimization is immensely underestimated and should be considered in future studies to avoid misleading comparisons of new approaches with weak and underperforming baselines. We also provide the largest collection of approaches to enable extensive comparisons with the state of the art. We hope that our benchmark and training procedure will provide a fruitful basis for future developments and accelerate the progress in the field of image classification with small datasets.

REFERENCES

[1] S. Arora, S. S. Du, Z. Li, R. Salakhutdinov, R. Wang, and D. Yu, "Harnessing the power of infinitely wide deep nets on small-data tasks," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[2] I. Azuri and D. Weinshall, "Generative latent implicit conditional optimization when learning from small sample," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 8584–8591.

[3] B. Barz and J. Denzler, "Deep learning on small datasets without pre-training using cosine loss," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1371–1380.

[4] B. Barz and J. Denzler, "Do we train on test data? Purging CIFAR of near-duplicates," *J. Imag.*, vol. 6, no. 6, p. 41, Jun. 2020, doi: 10.3390/jimaging6060041. [Online]. Available: https://www.mdpi.com/2313-433X/6/6/41

[5] A. Bietti, G. Mialon, D. Chen, and J. Mairal, "A kernel perspective for regularizing deep neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 664–674.

[6] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A.-L. Boulesteix, D. Deng, and M. Lindauer, "Hyperparameter optimization: Foundations, algorithms, best practices and open challenges," 2021, *arXiv:2107.05847*.

[7] J. Bornschein, F. Visin, and S. Osindero, "Small data, big decisions: Model selection in the small-data regime," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1035–1044.

[8] L. Brigato and L. Iocchi, "A close look at deep learning with small data," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 2490–2497.

[9] R.-J. Bruintjes, A. Lengyel, M. B. Rios, O. S. Kayhan, and J. van Gemert, "VIPriors 1: Visual inductive priors for data-efficient deep learning challenges," 2021, *arXiv:2103.03768*.

[10] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.

[11] N. Codella, V. Rotemberg, P. Tschandl, M. Emre Celebi, S. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M. Marchetti, H. Kittler, and A. Halpern, "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC)," 2019, *arXiv:1902.03368*.

[12] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation strategies from data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 113–123.

[13] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 702–703.

[14] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9268–9277.

[15] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4109–4118.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–21.

[17] J. Feng and T. Darrell, "Learning the structure of deep convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2749–2757.

[18] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Mach. Intell.*, vol. 2, no. 11, pp. 665–673, Nov. 2020.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[20] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 558–567.

[21] P. Helber, B. Bischke, A. Dengel, and D. Borth, "EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 7, pp. 2217–2226, Jul. 2019.

[22] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1921–1930.

[23] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, "Metalearning in neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, May 11, 2021, doi: 10.1109/TPAMI.2021.3079209.

[24] M. Ishii and A. Sato, "Training deep neural networks with adversarially augmented features for small-scale training datasets," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[25] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, pp. 1–54, Dec. 2019.

[26] O. S. Kayhan and J. C. van Gemert, "On translation invariance in CNNs: Convolutional layers can exploit absolute spatial location," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14274–14285.

[27] R. Keshari, R. Singh, and M. Vatsa, "Guided dropout," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 4065–4072.

[28] R. Keshari, M. Vatsa, R. Singh, and A. Noore, "Learning structure and strength of CNN filters for small sample size training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9348–9358.

[29] T. Kobayashi, "T-vMF similarity for regularizing intra-class feature distribution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6616–6625.

[30] S. Kornblith, J. Shlens, and Q. V. Le, "Do better ImageNet models transfer better?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2661–2671.

[31] A. Krizhevsky, "Learning multiple layers of features from tiny images," M.S. thesis, 2009.

[32] J. Lezama, Q. Qiu, P. Muse, and G. Sapiro, "OLE: Orthogonal low-rank embedding, a plug and play geometric loss for deep learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8019–8118.

[33] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, J. Ben-tzur, M. Hardt, B. Recht, and A. Talwalkar, "A system for massively parallel hyperparameter tuning," in *Proc. Conf. Mach. Learn. Syst.*, 2020, pp. 230–246.

[34] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.

[35] L. Lin, D. Liu, B. Liu, and Y. Xiao, "A latent variables augmentation method based on adversarial training for image categorization with insufficient training samples," in *Proc. 16th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Dec. 2020, pp. 969–975.

[36] L. Lin, B. Liu, X. Zheng, and Y. Xiao, "An efficient image categorization method with insufficient training samples," *IEEE Trans. Cybern.*, early access, Aug. 11, 2020, doi: 10.1109/TCYB.2020.3011165.

[37] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1449–1457.

[38] L. Liu, M. Muelly, J. Deng, T. Pfister, and L.-J. Li, "Generative modeling for small-data object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6073–6081.

[39] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–16.

[40] A. Navon, I. Achituve, H. Maron, G. Chechik, and E. Fetaya, "Auxiliary learning by implicit differentiation," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[41] E. Oyallon, E. Belilovsky, and S. Zagoruyko, "Scaling the scattering transform: Deep hybrid networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5618–5627.

[42] E. Oyallon, S. Zagoruyko, G. Huang, N. Komodakis, S. Lacoste-Julien, M. Blaschko, and E. Belilovsky, "Scattering networks for hybrid representation learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2208–2221, Sep. 2019.

[43] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[44] D. R. Plata, R. Ramos-Pollán, and F. A. González, "Effective training of convolutional neural networks with small, specialized datasets," *J. Intell. Fuzzy Syst.*, vol. 32, no. 2, pp. 1333–1342, Jan. 2017.

[45] A. Radford, "Language models are unsupervised multitask learners," OpenAI blog, Tech. Rep., 2019.

[46] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet classifiers generalize to ImageNet," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds., Long Beach, CA, USA, Jun. 2019, pp. 5389–5400.

[47] D. Rueda-Plata, R. Ramos-Pollán, and F. A. González, "Supervised greedy layer-wise training for deep convolutional networks with small datasets," in *Computational Collective Intelligence*, 2015.

[48] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and A. C. Berg, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[49] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[50] M. Simon, E. Rodner, T. Darrell, and J. Denzler, "The whole is more than its parts? From explicit to implicit pose normalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 749–763, Mar. 2020.

[51] D. Stutzmann, "Clustering of medieval scripts through computer image analysis: Towards an evaluation protocol," *Digit. Medievalist*, vol. 10, Jun. 2016.

[52] P. Sun, X. Jin, W. Su, Y. He, H. Xue, and Q. Lu, "A visual inductive priors framework for data-efficient image classification," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*. Cham, Switzerland: Springer, 2020, pp. 511–520.

[53] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[54] L. Tuggener, J. Schmidhuber, and T. Stadelmann, "Is it enough to optimize CNN architectures on ImageNet," 2021, *arXiv:2103.09108*.

[55] M. Ulicny, V. A. Krylov, and R. Dahyot, "Harmonic networks for image classification," in *Proc. BMVC*, 2019.

[56] M. Ulicny, V. A. Krylov, and R. Dahyot, "Harmonic networks with limited training samples," in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2019, pp. 1–5.

[57] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The CALTECH-UCSD birds-200–2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.

[58] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018.

[59] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, May 2021.

[60] K. Weiss, T. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, pp. 1–40, May 2016.

[61] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5028–5037.

[62] W. Xu, G. Wang, A. Sullivan, and Z. Zhang, "Towards learning affine-invariant representations via data-efficient CNNs," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 904–913.

[63] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, R. C. Wilson, E. R. Hancock, and W. A. P. Smith, Eds., Sep. 2016, pp. 87.1–87.12.

[64] X. Zhang, Z. Wang, D. Liu, and Q. Ling, "DADA: Deep adversarial data augmentation for extremely low data regime classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2807–2811.

[65] X. Zhang, Z. Wang, D. Liu, Q. Lin, and Q. Ling, "Deep adversarial data augmentation for extremely low data regimes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 15–28, Jan. 2021.

[66] B. Zhao and X. Wen, "Distilling visual priors from self-supervised learning," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 422–429.

[67] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5209–5217.

[68] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 13001–13008.

[69] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, Jan. 2018.

LORENZO BRIGATO received the M.Sc. degree (Hons.) in artificial intelligence and robotics from the RoCoCo Laboratory, Department of Computer, Control and Management Engineering, Sapienza University of Rome, Italy, in 2018, where he is currently pursuing the Ph.D. degree in engineering in computer science. He is a member of the RoCoCo Laboratory, Department of Computer, Control and Management Engineering, Sapienza University of Rome. His main research interests include data-efficient deep learning, anomaly detection, and robot learning.

BJÖRN BARZ received the M.Sc. degree (Hons.) in computer science from Friedrich Schiller University Jena, Germany, in 2016, and the Ph.D. degree *(summa cum laude)*, in 2021, with a focus on semantic and interactive content-based image retrieval, under the supervision of Joachim Denzler. He was leading the Knowledge Integration Team, as a Senior Researcher at the Computer Vision Group Jena, from 2020 to 2021. He is currently a Machine Learning Scientist at Carl Zeiss AG. His research interests include data-efficient deep learning, content-based image retrieval, and multimodal learning.

LUCA IOCCHI has been a Principal Investigator of several international, EU, national and industrial projects in artificial intelligence and robotics. He is currently the Vice President of RoboCup Federation and organized several international scientific robot competitions, as well as student competitions focusing on service robots and human–robot interaction. He is also a Full Professor at the Sapienza University of Rome, Italy, mainly teaching in the master's in artificial intelligence and robotics, and the Coordinator of the Ph.D. Program in Engineering in Computer Science. He has authored over 175 refereed papers (H-index 43 [Google Scholar]) in journals and conferences in artificial intelligence and robotics. His main research interests include cognitive robotics, task planning, multirobot coordination, robot perception, robot learning, human–robot interaction, and social robotics.

JOACHIM DENZLER (Member, IEEE) received the Diplom-Informatiker, Dr.-Ing., and Habilitation degrees from the University of Erlangen, Germany, in 1992, 1997, and 2003, respectively. He is currently a Full Professor in computer science and the Head of the Computer Vision Group, Friedrich Schiller University Jena, Germany. He is also the Director of the German Aerospace Center (DLR), Data Science Institute, Jena. He has authored and coauthored over 300 journal articles and conference papers as well as technical articles. His research interests include automatic analysis, fusion, and understanding of sensor data, especially the development of methods for visual recognition tasks and dynamic scene analysis. He has contributed in the area of active vision, 3-D reconstruction, and object recognition and tracking. He is a member of IEEE Computer Society, DAGM, and GI.