SAPIENZA
UNIVERSITÀ DI ROMA

Sapienza University of Rome

Department of Computer, Control and Management Engineering (DIAG)
Ph.D. in Data Science

Thesis For The Degree Of Doctor Of Philosophy

Media forensics investigations: from the origin to the authenticity of digital content

Advisor
Prof. Aris Anagnostopoulos
Prof. Irene Amerini

Candidate
Luca Maiano

Academic Year 2023-24 (XXXV cycle)

To those who believed in me.

Abstract

In recent years, we have observed a massive change in how information is exchanged. On the one hand, the explosion of social media has given birth to a new way of communicating and exchanging news, media, and ideas. On the other hand, the advancement of content manipulation and generation technologies have led to tools capable of recreating incredibly realistic artificial content. All this poses new challenges in verifying the authenticity and integrity of online content. Whenever we come across new media, we must understand its origin, whether it is real or deliberately modified, and verify its authenticity. In this thesis, we will analyze each problem, offering an overview of possible solutions.

The first challenge to solve when encountering multimedia content is reconstructing its source. This problem is as essential for verifying online news as for forensic investigations, where an image or video can represent evidence of a crime. Given a media, we wonder if it was captured with a specific offending camera model or if it was instead downloaded from a social platform. Solving this problem means analyzing the compression traces left in the file when it is captured or uploaded to a platform. To solve this challenge, we propose to train neural networks that learn to distinguish these traces, which we define as fingerprints. Specifically, we will show how these fingerprints change from camera to camera and when content is uploaded to a social network, making it possible to reconstruct the source of origin without relying on information such as metadata that can often be modified or deleted.

Another significant problem is that of verifying the authenticity of information. Recent advances in the development of artificial intelligence enable the generation of incredibly realistic content: deepfakes. On the one hand, this opens the doors to new applications in entertainment and creativity. On the other hand, it introduces a new generation of super-realistic fake content. The recognition of these contents is possible thanks to a set of factors. First, many of these techniques introduce semantic inconsistencies that are difficult to correct; furthermore, each generative technique leaves specific fingerprints similar to those left by camera models or social media. We will analyze possible strategies for recognizing fake content by exploiting these inconsistencies.

All the challenges mentioned so far have one problem in common. Data and information continually evolve, making standard detectors less and less robust as time passes. This is especially true with news, which constantly evolves as events worldwide grow. To prevent this from happening, fake news detectors must continuously learn to classify new information. The last part of this thesis will be dedicated to this topic. On the one hand, we will introduce a continuous learning strategy that allows a detector to learn to classify new news as it is published. Subsequently, we will analyze the vulnerabilities of these techniques concerning a new type of adversary attack.

Finally, we will discuss two forensic applications in the fields of ground to aerial matching and insurance.

Keywords: Media Forensics, Multimodal Content Verification, Media Source Identification, Deepfake Detection, Fake News Detection, Continual Learning.

Contents

List of Figures **v**

List of Tables **ix**

1 An introduction to Digital Forensics **1**
1.1 Contributions 3
1.2 Publications 4
1.3 Thesis outline 6

2 Multimedia forensics **7**
2.1 Source identification techniques 8
2.1.1 Platform provenance analysis 8
2.2 Content verification techniques 11
2.2.1 Deepfake detection 12
2.3 Continual learning and multimodal learning 14
2.3.1 Multimodal fake news detection 14
2.3.2 Data poisoning methods 15
2.4 Forensic applications 17
2.4.1 Reconstructing the location of images 17
2.4.2 Image similarity for the insurance domain 18

3 Platform provenance analysis **22**
3.1 Social network identification for videos 23
3.1.1 Proposed Method 23
3.1.2 Experimental Evaluation 25
3.2 Image and video source identification through the use of shared features 28
3.2.1 Proposed Method 30
3.2.2 Experimental Evaluation 32
3.2.3 Discussion 35

4 Content verification **38**
4.1 A depth-based strategy for deepfake detection 39
4.1.1 Proposed Method 40
4.1.2 Results 44
4.2 RGB-Depth Integration via Features Fusion 46
4.2.1 Proposed method 47
4.2.2 Implementation details 49
4.2.3 Results 49
4.3 From generation to detection of fake content: a comprehensive analysis of Stable Diffusion 54
4.3.1 Prompt Analysis 56
4.3.2 Frequency analysis of the generated images 58

4.3.3 Detection . . . 59
4.3.4 Human performances . . . 61
4.4 Human vs machine: a comparative analysis in detecting AI-generated images . . . 61
4.4.1 Method . . . 62
4.4.2 Prompt analysis . . . 64
4.4.3 Experiments . . . 64
4.4.4 Discussion . . . 67

5 Forensics analysis of media streams 74
5.1 A continual learning strategy for fake news detection . . . 74
5.1.1 Methodology . . . 76
5.1.2 Experiments . . . 79
5.1.3 Discussion . . . 86
5.2 Adversarial data poisoning for fake news detection . . . 87
5.2.1 Online learning framework . . . 88
5.2.2 Data poisoning attacks . . . 89
5.2.3 Results and discussion . . . 90
5.2.4 Future Work . . . 91

6 Forensic applications 92
6.1 Ground to aerial viewpoint localization . . . 92
6.1.1 Methodology . . . 93
6.1.2 Results . . . 96
6.2 Car damage reidentification . . . 97
6.2.1 Approach . . . 100
6.2.2 Implementation Details . . . 104
6.2.3 Results . . . 107
6.2.4 Discussion . . . 111

7 Conclusion and future work 113

Bibliography 116

List of Figures

1.1 "The Two Ways of Life" – Oscar Gustave Rejlander (1857) is one of the earliest examples of photo manipulation. 2

2.1 The top line shows an original MPEG encoded sequence. The next lines show the effect of deleting the three frames in the shaded area. The second line shows the reordered frames and the third line the recoded frames. The I-frame before erasing is subjected to double compression. Some of the frames following the deletion move from one GOP sequence to another. This double MPEG compression gives rise to specific statistical and temporal models that can be used to identify the platform of origin. 9

3.1 An application example of the proposed solution. An attacker records a video with illegal content and shares it on social networks or messaging apps. Subsequently, the police seize a device with this video and want to trace the source. 22

3.2 The proposed two-stream network (MultiFrame-Net) architecture. The network is constructed by concatenating the feature maps of the Ind-Net and the Pred-Net. The I-frame and P-frame streams are trained separately. Next, we concatenate the flattened output of the two-streams and train a fully connected classifier. 24

3.3 Learning approaches proposed in this study: (**a**) Method based on transfer learning; (**b**) Method based on multitask learning. In the transfer-learning approach we initially train the model on the image task. Then we reuse the feature representations learned on images to train the model on the video source platform identification task. In multitask learning we share the weights of the *constrained* and *conv1* layers of two siamese networks while learning them on images and videos in parallel. 31

3.4 Comparison of baseline single-task learning, transfer-learning–based, and multitask-learning–based models accuracy on image (in green) and video (in blue) patches. 34

3.5 Test accuracy of the multitask learner on images and video I-frames (at the image and patch level, respectively) obtained by fixing $w_{\mathcal{V}} = 1$ and varying the weight of the images $w_{\mathcal{I}}$ (x-axis) according to Equation (3.1). 36

4.1 Pipeline of the proposed method. In the fist step, we estimate the depth for each frame. Then, we extract the face and crop the frame and depth map around the face. In the last step, we train a classifier on *RGBD* input features. 40

4.2 Some example inconsistencies introduced in the depth map of manipulated faces. The first column represents the RGB patch of manipulated videos. The second and third columns show the estimated depth maps of the real and fake faces. The last column shows the difference between the real and fake depth maps. Deepfake faces tend to have fewer details than the original ones. 42

4.3 Accuracy on Deepfake (DF), Face2Face (F2F), FaceSwap (FS), NeuralTexture (NT) and all classes in the dataset (FULL) with RGB, RGBD, Gray and GrayD inputs. . 46

4.4 Our proposed pipeline. RGB and depth characteristics are analyzed separately by two MobileNet v2. We introduce an attention mechanism that masks the less important depth features based on the RGB features. Finally, we merge the features through a concatenation to proceed to the classification. 47
4.5 ROC Curve results obtained on deepfake detection task for RAW dataset when trained on all (ALL) forgeries. 52
4.6 ROC Curve results obtained on deepfake detection task for C40 dataset when trained on all (ALL) forgeries. 53
4.7 Heatmaps generated by the GradCam algorithm. The CAM RGB shows the obtained heatmaps for the RGB baseline model, while CAM MDN for the proposed method. 68
4.8 Examples of highly realistic face images (left) generated using Stable Diffusion and real (right) once extracted from FFHQ [135] dataset. 69
4.9 The taxonomy of our iterative prompt analysis. Starting from a simple base (i.e., "headshot of a"), we iteratively extend the prompt to obtain the desired output; we add some special tags to increase the realism of the image. 70
4.10 Examples of sample faces generated with three levels of realism. Figure (a) represents the less realistic sample and more like computer graphics renderings. Figure (b) shows a realistic sample with a typical shooting stage background. Figure (c) satisfies all the imposed constraints, resulting in an incredibly realistic photo immersed in a natural environment. 70
4.11 Spectra analysis of images generated with Stable Diffusion (SD) and the corresponding difference map with respect to the Real image on the top left corner. We apply two uniform color maps for a better view. 71
4.12 Different detectors trained on five classes: real, Stable Diffusion, StyleGAN, StyleGAN2, and StyleGAN3. 71
4.13 Distribution of correctly identified images. The histogram represents the number of users who get a certain number of correct answers. Most users correctly identified 17/20 images. 71
4.14 Procedure for human performance evaluation. 71
4.15 Proposed ResFormer architecture. 72
4.16 Improved generated images. 72
4.17 Common semantic errors produced by the model without negative prompts. 72
4.18 Predicted detection accuracy based on A: Confidence, B: Likeability of the photo. . . 73
4.19 Predicted detection accuracy for Ethnicity of faces (first column), Age of faces (second column), and Gender of faces (third column) for humans (first row) and machines (second row). 73

5.1 Overview of the proposed *Tri-Encoder* multimodal architecture. 76
5.2 Continual-learning strategy: For each new task T_i, we apply the knowledge-distillation loss as penalty factor. 78
5.3 F1 score of the Tri-Encoderover all tasks during time. In each of figures (a) and (b), the first plot shows the F1 score evaluated on dataset $T1$ (MediaEvalfor both of them), the second one the F1 score for $T2$ (PolitiFactfor (a), GossipCopfor (b)), and the third one for $T3$ (GossipCopfor (a), PolitiFactfor (b)). 85
5.4 Generic iteration at time t of the iterative process of online learning with data poisoning. **A**: new incoming news adds up to the already existing ones ($D^{t-1} \rightarrow D^t$). **B**: poisoned data are generated and injected into the existing data. **C**: a subset of the data is collected and added to the data d^{t-1} collected at time $(t-1)$; the aggregated data are denoted as d^t. **D**: the model f^{t-1} is updated to f^t with the addition of the newly collected data. 88

5.5 Percentage of samples required to flip the target sample label, depending on its x value and the type of poisoning used. The Most Confidence Mislabeling attack requires a lower number of samples to make the Linear LR model to misclassify the target news article. Conversely, the Quadratic LR model is resilient to it, but more significantly affected by the Target Label-Flipping attack. 89
5.6 x value of the original data, the poisoned data, and the target sample. The two Logistic Regression models trained on the poisoned data are also displayed. When subjected to the Target Label Flipping poisoning, both models slightly alter their predictions to misclassify the target sample. In contrast, the Power Model adapts to the Most Confidence Sampling poisoning while maintaining the correct classification of the target sample. 90

6.1 An example application of our method. Figures 6.1-(a) and (b) depict the aerial and ground-level images, respectively. Our proposed method extracts a graph representation of the two images and matches the graphs to localize the ground view over the aerial one. In (c), we show our predicted location (in red) and the corresponding ground truth (in green). 93
6.2 Overview of our proposed pipeline based on semantic graph for ground-to-aerial image matching. **Stage-A** depicts the semantic segmentation operation applied to the images taken from CVUSA dataset [305], both to a panoramic image and the corresponding aerial view. **Stage-B** represents the connected components algorithm and the corresponding nodes generator. **Stage-C** shows the graph generator for both panoramic and overhead images. Finally, **Stage-D** represents the final matching retrieval between the two points of view. 94
6.3 Example of connected component labeling algorithm applied to the binary mask (left side) of each label of a satellite image, i.e., building (a), pavement (b), road (c), and tree (d). On the right side of each label is illustrated the output of the connected component labeling algorithm: each color represents a connected component, i.e., different instances within a class are depicted in different colors. 95
6.4 Change in IOU in terms of mean and standard deviation when considering a subset of only three classes. As you can see, removing floors and trees improves performance. 98
6.5 Example of damage similarities. Triplets (a)-(b)-(c), (d)-(e)-(f), and (g)-(h)-(i) are real matches of the same damages. Each row shows an example type of damage: *scratch*, *crack*, and *dent*, respectively. 99
6.6 Our proposed pipeline. Images are first filtered to select images that capture the exterior of the vehicle through an EfficientNet-B5 and then, based on the view of the vehicle, that is, the sides of the car visible in the image., with an EfficientNet-B3. Zoomed images that only capture details of the damage do not allow to extract vehicle information because the zoom on the damage makes it difficult, if not impossible, to extract information about the car or the location of the damage on the bodywork; therefore they are immediately sent to the damage detector module. If the image depicts the entire vehicle, we detect the car with a RetinaNet-R50, we extract the brand and color of the vehicle with an EfficientNet-B2 and MobileNet, respectively, and we locate the damage over the car bodywork with a Mask R-CNN. Images are then filtered based on these information and are finally sent to the damage reidentification module (i.e., OSNet). 100
6.7 Distribution of the training and evaluation datasets used for the components of the pipeline. For each dataset, we show the classes and corresponding number of samples per class. For the damage reidentification dataset (Figure 6.7(g) we report several sources used to construct the dataset, and the minimum (Min) and the maximum (Max) number of matches for each image. 105

6.8 Sample queries and matching images in the public test set. The leftmost images are query images, followed by the highest similarity matches. Images framed in green are correct reidentifications, whereas those in purple indicate errors. The similarity score is also shown above each image. Even in cases of error, the model identifies images that are very similar to the query one. 108
6.9 Evaluation metrics of the damage reidentification model on the public test set. Figure 6.9(a) shows the precision–recall curve and F1 score obtained at different thresholds. Figure 6.9(b) shows the variation of the performances with respect to the top-k predictions for different values of k. 109
6.10 Embedding projection in a two-dimensional space using UMAP [201]. Points of the same color represent the same damages. We show three sample damages with their corresponding heatmap. The third column images show the heatmaps overlapped over the images for a better visualization of the activations. The model activations correctly focus on the damaged parts of the vehicle. Added to this, images of the same damages are correctly mapped to one other in the embedding space. 110
6.11 The average percentage of false positives across two European countries. Despite the good performance in an the proposed system still produces an average 90% of FPs in the experimental setting. To apply the system into production, we still need data external to the images, for instance, the VIN. 112

List of Tables

3.1 Accuracy on a subset of 10 devices from the VISION [263] dataset. The proposed method is confirmed to be more precise than the baseline at recognizing traces left by social networks and apps on frames patches. 26
3.2 Model accuracies and AUCs on a subset of 20 devices from the VISION dataset [263]. The MultiFrame-Net shows higher performance with respect to Ind-Net and Pred-Net. For this experiment, we model the problem as a binary classification task, i.e., shared or original videos. 27
3.3 Confusion matrix of the MultiFrame-Net over YT, WA and SC patches from 234 unique videos of the VISION dataset [263]. 27
3.4 Confusion matrix of the baseline single-task model on patches extracted from images. FBH and FBL represent high-quality and low-quality patches from Facebook. WA and NAT represents WhatsApp and native image patches respectively. 33
3.5 Confusion matrix of the baseline single-task model on patches extracted from video frames. YT, WA, and NAT represent YouTube, WhatsApp, and native video patches respectively. 34
3.6 Comparison of video patch classification accuracy of our transfer-learning and multitask-learning methods with the one of Amerini et al. [9] on the VISION dataset. 35
3.7 Confusion matrices on video patches of the transfer-learning (a) and multitask learning (b) models sharing the *constrained* and *conv1* layers. 36
3.8 Confusion matrices on video patches of the transfer-learning (a) and multitask learning (b) models sharing the *constrained* and *conv1* layers. 37

4.1 Patch level accuracy on Deepfake (DF), Face2Face (F2F), FaceSwap (FS), NeuralTexture (NT), and the full dataset (FULL) with RGB and RGBD inputs. Bold represents the best configuration of each backbone and underlined accuracies represent the best value over each class. In brackets, we indicate the percentage difference added by the depth. 43
4.2 Patch level accuracy of the Xception-based model on video level for Deepfake (DF), Face2Face (F2F), FaceSwap (FS), NeuralTexture (NT) and the full dataset (FULL) with RGB and RGBD inputs. In brackets we indicate the percentage difference added by the depth, while bold represents the best configuration. 43
4.3 Floating point operations (FLOPS) and average frame per second (FPS) inference frequency over different platforms. For the GrayD and RGBD, we indicate in brackets the difference with respect to the Gray and RGB models respectively. 44
4.4 Quantitative results obtained on deepfake detection task for RAW and C40 dataset settings when trained on Deepfakes (DF), Face2Face (F2F), FaceSwap (FS), NeuralTextures (NT), and all (ALL) forgeries. The best results for each configuration are reported in bold. 50
4.5 AUC values obtained on deepfake detection task for RAW and C40 dataset settings when trained on Deepfakes (DF), Face2Face (F2F), FaceSwap (FS), NeuralTextures (NT), and all (ALL) forgeries. The best results for each configuration are reported in bold. 51

4.6 AUC values by our proposed MDN compared to state-of-the-art methods. The best results for each configuration are reported in bold. 51
4.7 Accuracy results obtained on deepfake detection task for RAW dataset settings when Blur (BLR), Noise (NSE), Rescale (RSC), Translation (TRN), and all Combined (CMB) black box attacks are applied. The best results are in bold. 53
4.8 Accuracy results obtained on deepfake detection task for C40 dataset settings when Blur (BLR), Noise (NSE), Rescale (RSC), Translation (TRN), and all Combined (CMB) black box attacks are applied. The best results are in bold. 54
4.9 Accuracy results obtained on deepfake detection task for RAW dataset settings when Blur (BLR), Noise (NSE), Rescale (RSC), Translation (TRN), and all Combined (CMB) black box attacks are applied. The best results are in bold and the second best are underlined. 55
4.10 Accuracy results obtained on deepfake detection task for C40 dataset settings when Blur (BLR), Noise (NSE), Rescale (RSC), Translation (TRN), and all Combined (CMB) black box attacks are applied. The best results are in bold and the second best are underlined. 56
4.11 Deepfake detection results: binary classification between real and generated samples on the same configuration scenario. 59
4.12 Generalization performances of VGG16 model over different compression levels trained on configuration S1. 60
4.13 Generalization performances of VGG16 model over DALL-E2 images in the four considered setups. 60
4.14 Descriptive statistics of human participants. 65
4.15 Performance in terms of accuracy of state-of-the-art neural networks. 66
4.16 Generalization performance. First column: training on CDDB-s and testing on DMHD. Second column: training on DMHD and testing on CDDB-s. 66
4.17 Descriptive statistics of ResFormer trained on DMHD and tested over images employed in the human perception experiment. 67

5.1 Fake news detection performance on the MediaEval [29] dataset. 80
5.2 Model trained on each dataset in isolation from scratch 80
5.3 Multimodal methods performances on the MediaEval [29] dataset. 81
5.4 Results of the model trained from scratch on all three datasets available at once. The *Balanced* results indicate that in the training dataset, we balance the facts present in the three datasets and the *Not Balanced* case, reports the results of the model trained on the concatenation of the three unbalanced training sets. 83
5.5 Transfer learning performance in terms of F1 score of the model trained on $T1$, followed by transfer learning to $T2$, followed by transfer learning to $T3$. The rows indicate the dataset on which we perform the transfer learning (or the initial training for row $T1$) and the columns indicate the dataset on which we perform evaluation. For instance, Table (a) shows that after performing the initial training on the $T1 = MediaEval$ dataset, and then performing transfer learning to $T2 = PolitiFact$ followed by transfer learning to $T3 = GossipCop$, the F1 score on the MediaEval-dataset is 0.7218, and the F1 score on the PolitiFactdataset is 0.5224. 84
5.6 F1 score of the Tri-Encoderover all tasks after the last training session. Bold values indicate the best performance on a task. Tasks marked with $*$ indicate the learning strategy that performed best on those specific tasks. 86
5.7 Average accuracy (ACC) and forgetting (BWT) of the continual-leaning approaches on the three datasets. For ACC a higher value is better, for BWT a lower value is better. 87

6.1 Performances obtained with the proposed method varying the windows dimension. . 97

6.2 Quantitative evaluation of the proposed automated pipeline with respect to Verde et al. The best results are in bold. 97
6.3 Damage detection. model trained with different learning rate (LR) strategies. 109
6.4 Evaluation of the damage-similarity model on the public test set. 111
6.5 Evaluation of the damage-similarity model on the private test sets. 111

Chapter 1

An introduction to Digital Forensics

What is digital forensics? An accepted definition describes it as the set of scientific techniques for the *acquisition, preservation, validation, identification, analysis, interpretation, documentation, and presentation* of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events, most of the times of a criminal nature [304]. This definition can be considered very close to the definition of artificial intelligence (AI), which can be defined as a system's ability to correctly *interpret* external data, to *learn* from such data, and to *use those learnings to achieve specific goals* and tasks through *flexible adaptation* [133]. As such, artificial intelligence can be very useful in the validation, identification, and mostly in the analysis and interpretation of multimedia content.

In this thesis, we focus on the area of digital forensics called *multimedia forensics*, which deals with analyzing digital media such as images, videos, or audio files. This field combines principles and approaches from diverse research areas, such as artificial intelligence, computer vision, and signal processing, when it comes to addressing the authenticity, integrity and source of an image or a video. Although they may seem like very recent problems, this discipline has its roots far back in the past. The history of photo manipulation, in particular, has surprising antecedents that can be traced back to the early days of photography in the 19th century. One significant early development in the world of photography was the wet collodion process, which was introduced in the mid-19th century. This revolutionary technique made it possible for photographers to combine multiple images into a single negative. It involved coating glass plates with collodion, a syrupy solution of cellulose nitrate, which was then sensitized with light-sensitive chemicals and exposed to a camera. The resulting negative could be used to produce multiple prints. This process not only allowed for multiple prints of a single image but also laid the groundwork for the early forms of photo manipulation. Skilled photographers of the time could take several shots and manipulate elements within the scene by combining them in a single negative. These early manipulations were often done to correct imperfections or enhance certain elements of the photograph. For instance, retouching was used to remove blemishes or imperfections in portraits. One famous example of early photo manipulation is the work of Oscar Gustave Rejlander, a pioneering photographer in the mid-19th century. He is known for creating composite photographs by carefully cutting and pasting different images together to create a single cohesive scene. His work, "The Two Ways of Life" (depicted in Figure 1.1), is a prime example of this technique, featuring a complex tableau of various subjects and symbolic elements assembled from over thirty individual negatives. Rejlander's work and other early

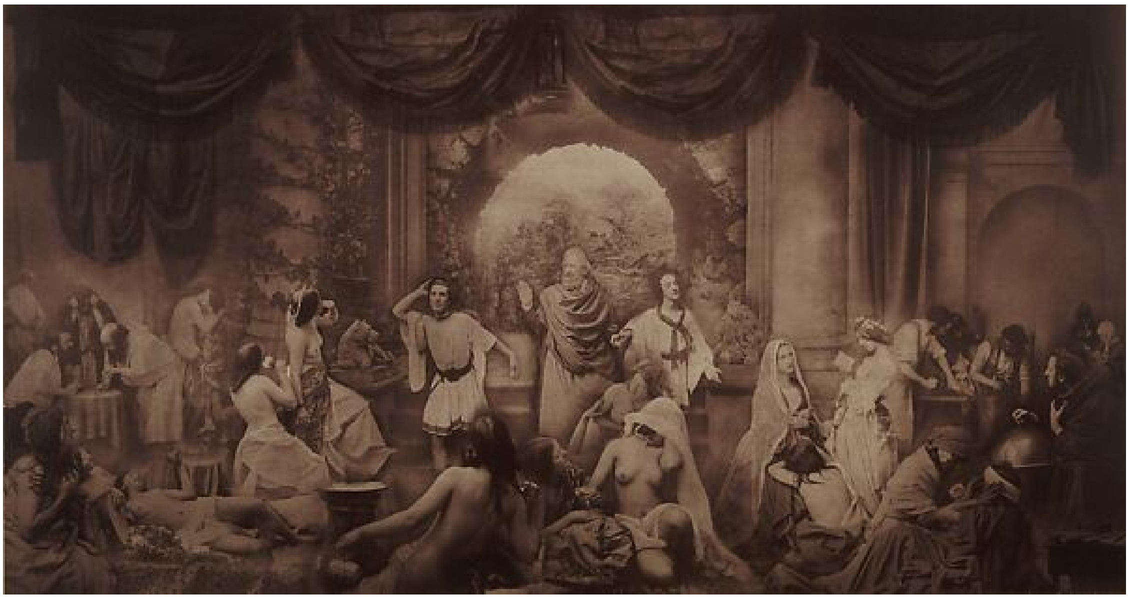


Figure 1.1: "The Two Ways of Life" – Oscar Gustave Rejlander (1857) is one of the earliest examples of photo manipulation.

experiments with photo manipulation not only demonstrated the creative potential of photography but also raised ethical and philosophical questions about the authenticity of photographs. Even in these early days of photography, the potential to manipulate and alter reality through imagery was evident. Fast-forward to the digital age and the manipulation of images has become vastly more accessible, sophisticated, and sometimes deceptive. While the methods have evolved, the core ethical and technical issues surrounding photo manipulation that began in the 19th century persist today, highlighting the enduring significance of media forensics in discerning the truth in an increasingly visual world. Fake media content such as deepfakes, hoaxes, or misleading images represents a growing concern in the digital age. Misinformation, which refers to misleading information shared either unintentionally or deliberately, has become a pervasive and concerning issue in recent years. It can take many forms, from hoaxes and urban legends to fabricated news stories, manipulated images, and viral rumors. This issue became even more important with the spread of social media platforms. The viral nature of social media means that misinformation can quickly reach a wide audience, making it challenging to contain.

In the last few years, multimedia forensics has undergone a remarkable evolution driven by the convergence of technological advancements, the increasing prevalence of digital media in our daily lives, and the growing importance of maintaining trust and credibility in a world where visual content is widely shared and manipulated. One of the most significant drivers of this evolution has been the rapid progress in machine learning and artificial intelligence. Machine learning algorithms have become increasingly sophisticated in detecting manipulated or fake media by analyzing patterns, inconsistencies, and artifacts in images and videos. This has enabled more accurate and efficient detection of media tampering. However, like any technology, AI-based methods have their limitations, and it's important to understand these constraints to make informed and effective use of these tools.

AI models, especially deep learning algorithms, require substantial amounts of labeled training data to learn and perform well. These datasets are often limited and may not adequately cover all possible manipulation techniques. The effectiveness of AI methods heavily relies on the quality and diversity of the training data. AI-based models can be susceptible to adversarial attacks, where attackers purposefully manipulate media in a way that can evade detection. Moreover, AI models trained on specific manipulation techniques may not generalize well to detect previously unseen methods or combinations of manipulations. This limitation is particularly relevant as new manipulation techniques constantly emerge. Finally, some AI models for multimedia forensics are considered "black boxes," making it challenging to understand how and why they arrive at particular conclusions. Interpretability is crucial for legal and ethical considerations.

All these problems are still unsolved, and despite numerous advances, the scenario in which forensic investigations are applied is constantly changing with the advancement of new technologies. If until a few years ago, for example, one of the significant challenges was to recognize images or videos partially modified through photo or video retouching tools such as Photoshop or Adobe Premiere Pro, today the scenario has become even more complex due to generative techniques in capable of generating realistic images from nothing. We are only at the dawn of a new generation of increasingly advanced content creation methods, and this opens up numerous challenges for the forensic community to solve. Complicating the picture is the growing ease of sharing information on the web, which makes spreading false information much easier than in the past.

1.1 Contributions

This thesis discusses possible solutions to some of the current problems. In particular, we will focus on four challenges.

- *Can we understand if an image or video has been downloaded from a social network?*

  Source identification of images is a crucial process in digital forensics and media analysis. It involves determining the origin of an image, which is vital in various contexts, such as verifying the legitimacy of a news photograph, tracking down copyright violations, or investigating criminal activities. In this thesis, we will present two studies on the topic. In particular, we extend the studies already conducted on reconstructing the platform of origin of images to videos. We will show that the process of uploading a video to a social network leaves traces in the video signal that are different for each social network, and thanks to these, it is possible to recognize content shared on a specific platform by others. Furthermore, we will analyze the differences between the traces left on images and those left on videos and show similarities that can be exploited to simultaneously train a deep learning-based detector on both media.

- *Can we verify the authenticity of the content and recognize an artificially generated video or image?*

  This problem is comprehensive and covers a large number of applications. This work will focus mainly on recognizing artificially generated images and videos. This content, commonly known as deepfakes, can be highly realistic and challenging to distinguish from natural videos. An essential part of this thesis will be dedicated to this type of content. In particular, fake content recognition techniques will be introduced that exploit semantic inconsistencies introduced during the generation process. We will also show the first studies on the human perception of these contents, demonstrating that, in some specific cases, with some limitations, AI detectors can be more accurate than humans.

- *How can we recognize new false content as it becomes public online?*

  The biggest challenge is to develop resilient solutions concerning an ever-increasing number of content generation or manipulation techniques. This problem is evident in any forensic task, but we will focus on recognizing fake news. The ongoing nature of information introduces the problem of developing fake news detectors that can update as new content becomes available. In this context, continuous learning is an increasingly important approach to detect fake news. In this thesis, we will analyze the problem from two points of view. First, we introduce a multimodal detector capable of simultaneously analyzing the text and images associated with news. We will then show how it is possible to make the detector capable of updating its knowledge through continuous learning techniques. Next, we will propose a preliminary study on a new class of adversary attacks. We will show how it is possible to attack a method of recognizing fake news online by introducing adversarial examples that allow you to manipulate the detector's behavior on pre-existing notice. The novelty of this attack is that the attacker does not have to have access to previous information, making this attack very effective and dangerous.

- *Can we apply forensic techniques to real application scenarios?*

  Applying forensic techniques in real-world scenarios can pose several challenges. In fact, outside of a controlled working environment, such as those that can be recreated in the experimental phase, forensic techniques can face further challenges. One of these is the interpretability of the results. In this sense, we will show how it is possible to verify a place depicted in a photo by matching it with satellite images. This solution is extremely useful in all scenarios, such as social media, where the image metadata may have been removed or modified. In addition, an often underestimated but essential problem in the application phase is the rate of false alarms that a forensic application can generate. In this regard, we will show how to apply false image recognition techniques in an industrial context: insurance.

1.2 Publications

Most of the works presented in this thesis have been published in international conferences and journals, and for each section, the relevant publications will be cited. The following is the list of publications I authored or co-authored during my PhD.

Chapter 2

- Amerini, I., Anagnostopoulos, A., Maiano, L. and Celsi, L.R., 2021. Deep learning for multimedia forensics. *Foundations and Trends in Computer Graphics and Vision*, 12(4), pp.309-457.

Chapter 3

- Amerini, I., Anagnostopoulos, A., Maiano, L. and Celsi, L.R., 2021, June. Learning double-compression video fingerprints left from social-media platforms. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2530-2534). IEEE.
- Maiano, L., Amerini, I., Ricciardi Celsi, L. and Anagnostopoulos, A., 2021. Identification of social-media platform of videos through the use of shared features. *Journal of Imaging*, 7(8), p.140.

Chapter 4

- Maiano, L., Papa, L., Vocaj, K., and Amerini, I., 2023. Depthfake: A depth-based strategy for detecting deepfake videos. In *J.-J. Rousseau and B. Kapralos, editors, Pattern Recognition, Computer Vision, and Image Processing.* ICPR 2022 International Workshops and Challenges, pages 17–31, Cham, 2023. Springer Nature Switzerland.
- Papa, L., Faiella, L., Corvitto, L., Maiano, L., and Amerini, I., 2023. On the use of stable diffusion for creating realistic faces: from generation to detection. In *2023 11th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6.

Under review

- Maiano, L., Benova, A., Papa, L., Stockner, M., Marchetti, M., Convertino, G., Mazzoni, G., and Amerini, A., 2023. Human vs machine: a comparative analysis in detecting AI-generated images. In *IEEE Security & Privacy*, 2023. *(Under review.)*
- Leporoni, G., Maiano, L., Papa, L., Amerini, I., 2023. A Guided-Based Approach for Deepfake Detection: RGB-Depth Integration via Features Fusion. In *Pattern Recognition Letter*, 2023. *(Under review.)*

Chapter 5

- Siciliano, F., Maiano, L., Papa, L., Baccini, F., Amerini, A., and Silvestri, F., 2023. Adversarial Data Poisoning for Fake News Detection: How to Make a Model Misclassify a Target News without Modifying It. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD) Workshops*, 2023.

Under review

- Maiano, L., Evangelisti, M., Bianchini, S., and Anagnostopoulos, A., 2024. What's Real News Today? A Multimodal, Continual-Learning Approach for Detecting Fake News Over Time. In *SIAM International Conference on Data Mining (SDM24)*, April 18 - 20, 2024. *(Under review.)*

Chapter 6

- Maiano, L., Montuschi, A., Caserio, M., Ferri, E., Kieffer, F., Germanò, C., Baiocco, L., Celsi, L.R., Amerini, I. and Anagnostopoulos, A., 2023. A deep-learning–based antifraud system for car-insurance claims. *Expert Systems with Applications*, p.120644.
- Bonaventura, T.S., Maiano, L., Papa, L. and Amerini, I., 2023, June. An Automated Ground-to-Aerial Viewpoint Localization for Content Verification. In *2023 24th International Conference on Digital Signal Processing (DSP)* (pp. 1-5). IEEE.

1.3 Thesis outline

The remainder of this thesis is structured as follows. The following chapter (i.e., Chapter 2) offers an overview of this discipline's state of the art. Next, in Chapter 3, we discuss the problem of recognizing the provenance platform of images and videos. In Chapter 4, we propose techniques for verifying the authenticity of photos and videos. Chapter 5 is dedicated to our studies on continuous learning applied to fake news. Chapter 6 discusses two forensic applications. Finally, Chapter 7 concludes and discusses possible future developments.

no

Chapter 2

Multimedia forensics

In today's interconnected digital world, multimedia content has become an integral part of our daily lives. From photos and videos shared on social media platforms to audio recordings and digital documents, multimedia data surrounds us. However, with the proliferation of digital media comes the potential for manipulation, forgery, and misuse. This is where multimedia forensics emerges as a crucial field of study and practice.

Multimedia forensics is a multidisciplinary branch of digital forensics that focuses on the analysis, authentication, and verification of multimedia data. Its primary aim is to uncover the truth behind digital media, addressing questions of authenticity, integrity, and credibility. This field plays a pivotal role in various domains, including law enforcement, journalism, legal proceedings, and cybersecurity. In the realm of multimedia forensics, experts employ a wide array of techniques and tools to examine and scrutinize multimedia content. They investigate the origins of an image or video, verify its authenticity, detect alterations, and establish a chain of custody for digital evidence. These efforts are essential for ensuring the reliability and admissibility of multimedia data in legal cases and investigations. Multimedia forensics encompasses various subdisciplines, each focusing on specific types of media, such as image forensics, video forensics, audio forensics, and document forensics. These subfields utilize specialized methodologies to uncover digital tampering, conduct source identification, and extract valuable information from multimedia artifacts.

As technology continues to advance, so too do the challenges in multimedia forensics. With the advent of deepfake technology, sophisticated image and video manipulation tools, and the spread of misinformation, the need for accurate and reliable multimedia analysis has never been greater. Experts in this field work tirelessly to develop innovative techniques to counteract these threats and maintain the trustworthiness of digital media. In this Chapter, we review the most recent techniques in this field. Specifically, we will review three main problems. (1) We start discussing the techniques for identifying the source of origin of a media (see Chapter 2.1). Such methods are helpful to understand *where* a media comes from (i.e., a social media or a camera model). (2) In Chapter 2.2, we will then move to the online content verification methods, which cover a wide range of topics from deepfake detection to ground-to-aerial matching for reconstructing the location captured by an image. These techniques can be applied to verify an online media whose *veridicity and history are unknown.* Finally, (3) in Chapter 2.3, we review the most recent and intriguing problems in the field: dealing with *continuously evolving settings* where new data, topics, manipulation, and sources emerge every day. This is a highly complex setting where new content becomes available over time (like news, for example), and we are interested in verifying the authenticity of that content.

Before proceeding to the next sections, for the reader who is interested in further investigating the state of the art with respect to this thesis, we suggest the numerous surveys covering different aspects of the matter [8, 223, 274, 287, 314].

2.1 Source identification techniques

Source identification problems, often referred to as source attribution or source tracking, are a set of challenges within the field of multimedia forensics. These problems revolve around determining the origin or source of a piece of multimedia content, such as an image, video, audio recording, or document. The primary goal of source identification is to ascertain who created, captured, or authored a specific piece of digital media. These methods can be applied in various scenarios and have important implications in areas such as cybersecurity, law enforcement, journalism, and legal proceedings.

In image forensics, source identification aims to determine the source camera or device that captured a particular image. Each camera or device has unique characteristics, such as sensor imperfections, lens distortions, and noise patterns, which can serve as digital fingerprints. Analyzing these characteristics can help identify the source of an image. Similar to image source identification, video source identification involves determining the source camera or device for a given video. Videos may contain additional information, such as temporal variations in sensor characteristics, that can aid in source attribution.

With the rise of social media and online communication platforms, source identification problems extended to digital content shared on the internet. Detecting the source of viral images, videos, or messages is crucial in combating misinformation and cybercrimes. In the following section, we review some of the most promising techniques proposed so far for source attribution in social media.

2.1.1 Platform provenance analysis

Researchers have been studying multimedia forensics for more than two decades in different experimental settings; however, the practical application of these techniques has been limited because of the high variability of real cases, which is difficult to reproduce in experiments. Today, the assessment of the authenticity and the source of multimedia content has become an essential element for building trust in images and videos shared across online platforms. When videos of military propaganda, revenge porn, cyberbullying, or other illegal content are shared on social media, they can easily go viral. While it is important to immediately identify and delete this content from social platforms, another problem to be addressed is to identify the authors of the video to proceed with any legal action. In many other cases, law enforcement may locate a device containing illegal content and to identify its source, it may be necessary to understand whether the video was recorded with the hijacked device or whether it was downloaded via messaging apps or social networks. In fact, in all these cases videos and images could be used as evidence in court, and knowing how to identify videos shared on social platforms could help identify criminal networks operating online. However, for this to be possible, it is necessary to be able to prove the origin of such content.

When uploaded and shared across social networks and messaging apps, multimedia content undergoes a processing step in which the platforms perform a set of operations on the input. Indeed, to optimize transfer bandwidth as well as display quality, most platforms apply specific compression and resizing methods. These methods, which tend to be unpublished, differ among the different social platforms [85].

To guide the reader through this section, we review some basic concepts related to this problem that will help further understand why each platform leaves different traces. In video coding, a video is represented as a sequence of *groups of pictures* (GOPs), each of which begins with an *I-frame*. I-frames are not predicted from any other frame and are independently encoded using a process similar to JPEG compression. Apart from the I-frames, the rest of each GOP consists of *P-frames* and *B-frames*. These frames are predictively encoded using motion estimation and compensation. Thus, these frames are derived from segments of an anchor I-frame and represent lower-quality frames. As shown in [228, 296], recompression operations can leave both static and temporal artifacts in the video signal when a video sequence is subjected to double MPEG compression. Statically, the I-frames of an MPEG sequence are subjected to double JPEG compression. Temporally, frames that move from one GOP to another, as a result of frame deletion, give rise to relatively larger motion estimation errors. Figure 2.1 shows an example of a short eleven-frame MPEG sequence. In this example, during the upload phase, the video is subjected to the removal of three frames and subsequent recompression. The second row shows the reordered frames, and the third line shows the re-encoded frames after recompressing the video as an MPEG video.

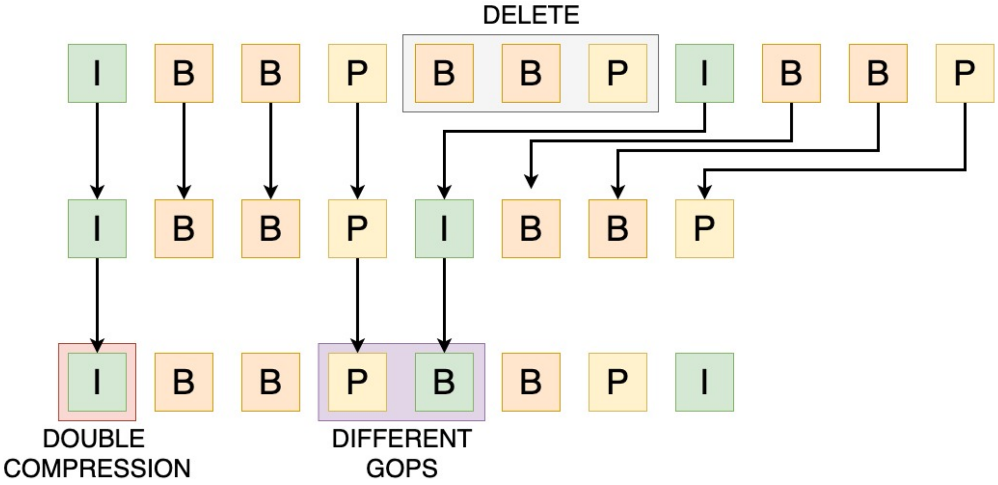


Figure 2.1: The top line shows an original MPEG encoded sequence. The next lines show the effect of deleting the three frames in the shaded area. The second line shows the reordered frames and the third line the recoded frames. The I-frame before erasing is subjected to double compression. Some of the frames following the deletion move from one GOP sequence to another. This double MPEG compression gives rise to specific statistical and temporal models that can be used to identify the platform of origin.

Statically, when an I-frame gets recompressed with different bit rates (i.e., quantization amounts), the DCT coefficients are subject to two quantization levels, leaving behind a specific statistical signature in the distribution of DCT coefficients [189, 228]. Quantization is a pointwise operation, which can be calculated as:

$$Q_k(s_1) = \left\lfloor \frac{k}{s_1} \right\rfloor,$$

where s_1 indicates the quantization step and k denotes a value in the range of the input frame.

Similarly, double quantization is also a pointwise operation given by:

$$Q_{s_1 s_2}(k) = \left\lfloor \left\lfloor \frac{k}{s_1} \right\rfloor \frac{s_1}{s_2} \right\rfloor,$$

where s_1 and s_2 are the quantization steps. From the equation above, double quantization can be described as a sequence of three operations: A quantization with step s_1, a de-quantization with step s_1, and a quantization with step s_2. As Wang and Farid show [296], the re-quantization introduces the periodicity of the artifacts into the histograms of quantized frames. As these artifacts will differ depending on the quantization step used by every platform, they can be used to distinguish differences between social media platforms.

Temporarily, deleting a few frames of the video to fit the maximum length set by some platforms can in turn leave information. For example, consider deleting three frames in Figure 2.1. Within the first GOP of this sequence, the I-frame and the first P-frame come from the first GOP of the original sequence. The third B-frame, however, is the I-frame of the second GOP of the original sequence, and the second I-frame is the first P-frame of the second GOP of the original video. When this new sequence gets re-encoded, we will observe a larger motion error between the first and second P-frames, as they originated from different GOPs. Furthermore, this increase in motion error will be periodic, occurring in each of the GOPs after the frame gets deleted. Formally, consider a six-frame sequence that is encoded as $I_1, P_2, P_3, P_4, P_5, I_6$. Because of JPEG compression and motion error, each frame can be modeled by an additive noise, that is:

$$I_i = F_i + N_i \qquad P_j = F_j + N_j$$

with $i \neq j$, where each N_i, N_j is the additional noise and F_i, F_j are the original frames. Notice that the noise for I_1 through P_5 will be correlated to each other because they belong to the same GOP, but not to that of I_6. If we denote the motion compensation as $M(\cdot)$, we can derive the motion error for a frame $i, (i > 1)$ as:

$$\begin{aligned} e_i &= P_i - M(I_{i-1}) \\ &= F_i + N_i - M(F_{i-1} + N_{i-1}) \\ &= (F_i - M(F_{i-1})) + (N_i - M(N_{i-1})). \end{aligned}$$

Suppose now that we delete frame P_4, bringing frames P_5 and I_6 to the fourth and fifth positions, respectively. I_6 will now be encoded as the new P_5'. The motion error for this new frame will be:

$$e_5' = (F_6 - M(F_5)) + (N_6 - M(N_5)).$$

Notice that for frames belonging to the same GOP, the components of the additive noise term $N_i - M(N_{i-1})$ are correlated, thus, we can expect some noise cancellation. After the deletion of frame P_4, however, the two components of the additive noise term $(N_6 - M(N_5))$ are not correlated, leading to a relatively larger motion error compared to the others. This pattern can be learned by a deep neural network with sufficient training data samples, as we will discuss below.

All these operations inevitably leave some traces on the media content itself [179, 273, 297]. The social media identification problem has been widely studied for image files with promising results

[14, 34, 85], employing machine learning classifiers. Recently, Quan et al. [232] showed that by using convolutional methods it is possible to recognize Instagram filters and attenuate the sensor pattern noise signal in images. Amerini et al. [11] introduced a CNN for learning distinctive features among social networks from the histogram of the discrete cosine transform (DCT) coefficients and the noise residual of the images. Phan et al. [227] proposed a method to track multiple image sharing on social networks by using a CNN architecture able to learn a combination of DCT and metadata features. Nevertheless, the identification of the traces left by social networks and messaging apps on video content remains an open problem. Recently, Iuliani et al. [123] presented an approach that relies on the analysis of the container structure of a video through the use of unsupervised algorithms to perform source-camera identification for shared media with high performance; their method is strictly dependent on the file structure, whereas in our work we are interested in approaches that are based on the content of a video, independently of the file type. Kiegaing and Dirik [141] showed that fingerprinting the I-frames of a flat content native video can be used to accurately identify the source of YouTube videos. Moreover, although the research community has treated video and image forensics as separate problems, a recent work from Iuliani et al. [122], demonstrates that it is possible to identify the source of a digital video by exploiting a reference sensor pattern noise generated from still images taken by the same device, suggesting that it could be possible to link social media profiles containing images and videos captured by the same sensor.

Despite numerous efforts, several challenges remain to be resolved; first of all, the adaptability of platform provenance techniques to the possible fingerprint changes that can occur over time on each platform. Periodically, social platforms may change the processing operations performed when loading content. Being able to be resilient to these changes is an absolute necessity. Related to this, we often have little training data available, thus making it necessary to update the models quickly. Understanding how to deal with these changes is fundamental. In Chapter 3, we will discuss possible solutions.

2.2 Content verification techniques

Content verification is a crucial process in multimedia forensics and information integrity. It involves the systematic examination of digital content, such as images, videos, audio recordings, and text, to confirm its authenticity, accuracy, and reliability. The primary goal of content verification is to determine whether the content has been altered, manipulated, or misrepresented in any way.

In recent years, machine learning and artificial intelligence have played a significant role in content verification. These technologies enable the development of algorithms that can detect anomalies, inconsistencies, or patterns indicative of manipulation. Machine learning models can be trained to recognize alterations in images or videos, aiding in the verification process. These technologies have gained particular importance with the rise of deepfake technology. Deepfakes are highly convincing, AI-generated multimedia content that can deceive viewers. Many efforts have been devoted to designing specialized tools and algorithms to detect deepfakes and distinguish them from authentic media. We will review these methodologies in Chapter 2.2.1.

2.2.1 Deepfake detection

Deepfakes are synthetic media or altered videos that use deep learning and computer graphics techniques to replace the likeness of one person with another in video or audio recordings. Despite their enormous potential in many creative applications, such as the world of filmmaking and video editing, deepfakes have attracted the attention of many researchers in the forensic community due to their potential for misuse, including spreading misinformation, impersonating individuals, and manipulating content for malicious purposes. Today, the term deepfake has expanded in scope to encompass a wide range of potential video alterations. This includes the ability to generate speech in the voice of any individual, modify facial expressions, interchange one person's identity with another, and even change the content of their speech.

Deepfake detection is the process of identifying and flagging multimedia content, typically videos or audio recordings, that have been manipulated or generated using deep learning and artificial intelligence techniques. Detecting deepfakes is crucial to maintaining trust and authenticity in multimedia content, as they can be used for deceptive purposes such as spreading false information, defamation, or identity theft. Several approaches have been proposed to recognize this type of content in recent years. We still do not have a definitive solution to the problem, but we can group the most common methodologies into two macro-areas. *Single-modality methods* analyze the video, image, or audio features separately. These methodologies aim to train deep learning models that can learn to identify the distinctive features or semantic inconsistencies typical of deepfakes. These methodologies work pretty well, but the most critical challenge remains generalization compared to new generative techniques. More recently, the possibility of combining the different modalities (audio and video) has been explored to intercept possible inconsistencies between various media. This type of analysis is commonly called *multimodal analysis*.

Single-modality methods

In the extensive and swiftly expanding body of research on deepfake detection, the majority of approaches depend on supervised training. They make use of extensive datasets containing both authentic and manipulated videos. Most of these methods primarily focus on analyzing video content and capitalize on low-level features, which are artifacts stemming from various imperfections in the generation process. These techniques tend to be highly effective when the video being examined exhibits a manipulation that aligns with what was seen during training [48, 71, 214]. However, they prove to be considerably less effective when facing videos that were manipulated using novel, previously unseen techniques. Given the frequent emergence of new methods for generating synthetic content, encountering the latter scenario has become increasingly common. Even when assuming the availability of examples demonstrating new manipulation techniques, the process of continually expanding the training datasets becomes unmanageable. Conversely, fine-tuning these models with new data often leads to a loss in performance, known as *catastrophic forgetting* [81],. To address this challenge, the literature has proposed specific solutions, such as few-shot learning [15, 51, 128, 153], incremental learning [139, 197], weakly-supervised learning[157], or continual learning [145] approaches. Nonetheless, the fundamental problem persists: acquiring a ready supply of new manipulation examples. Another straightforward yet effective strategy to enhance generalization is augmentation. In forensic applications, this should extend beyond the conventional operations in computer vision to encompass compression and resizing, which increase resilience against the typical degradations introduced by social networks. Additionally, certain specialized forms of cut-out techniques have shown utility in deepfake detection [57]. Employing ensembling techniques also proves advantageous in elevating performance and fortifying against potential misalignments [31, 72].

Apart from the limited performance, another notable drawback of the aforementioned methods is the absence of interpretability. This issue is partially mitigated by techniques aimed at identifying specific cues related to the generation process. One approach involves detecting low-level artifacts stemming from the up-sampling operation, which are clearly discernible in the Fourier domain. Consequently, several studies have conducted frequency-based analyses [47, 164, 175, 186]. Other studies tackle the learning of distinct artifacts introduced during blending, a necessary processing step in many manipulation techniques [165]. More broadly, attention mechanisms are employed to direct the network's focus toward low-level and/or high-level inconsistencies in both spatial and temporal domains [35, 56, 292, 328, 329, 333, 346, 347]. While these solutions offer some insights into the nature of the performed manipulation, they are susceptible to various quality degradation actions that obscure the low-level features they rely on. These actions encompass not only inadvertent image impairments but also deliberate alterations and adversarial attacks [118, 211], which are becoming increasingly prevalent. In pursuit of robustness, a method presented in Haliassos et al. [94] adopts a different approach by centering on semantic features and targeting inconsistencies in mouth movements. The spatio-temporal architecture is pre-trained on the visual speech recognition task and subsequently fine-tuned using mouth embeddings from authentic and manipulated videos.

Multimodal analysis

In recent years, a handful of pioneering research endeavors have ventured into the concurrent analysis of audio and video to detect deepfakes. Some of these studies focus on identifying disparities between audio and video components. For instance, the approach outlined in Korshunov et al. [150, 151] capitalizes on the inability of certain deepfake generation techniques to properly align the audio stream with the video content. Similarly, the fundamental concept presented in Zhou and Lim [342] revolves around learning and utilizing the inherent synchronization between video and audio. However, due to rapid technological advancements, there now exist numerous methods capable of generating highly convincing deepfakes with precise synchronization between speech and lip movements [229]. Consequently, conducting an audio-visual synchronization analysis has become an exceedingly intricate task. The method introduced in Mittal et al. [205] places its emphasis on extracting emotional features from both modalities, followed by a similarity analysis conducted within the same audio and video. In Wang et al. [293], a multi-modal and multi-scale transformer architecture is devised to harness spatial and frequency domain artifacts simultaneously, while Chugh et al. [45] pursues the concept of identifying disparities between audio and visual streams by training a modality dissonance score. Despite the promise displayed by these approaches, it's important to note that they demand access to both counterfeit and genuine videos during the training phase, a requirement that could potentially limit their capacity for broad applicability.

Other works approach the problem as a reidentification problem. Re-identification methods distinguish each individual by extracting some specific biometric traits that can be hardly reproduced by a generator [2, 2, 3]. The first work of this kind was introduced by Agarwal et al. [3] and exploits the distinct patterns of facial and head movements of an individual to detect fake videos. In another work [2], the same research group studied the inconsistencies between the mouth shape dynamics and a spoken phoneme. More recently, Cozzolino et al. [50] introduced a method that extracts facial features based on a 3D morphable model and focuses on temporal behavior through an adversarial learning strategy. Another work from Cozzolino et al. [49] introduces a contrastive method based on audio-visual features for person-of-interest deepfake detection.

Regardless of the techniques that are adopted, one challenge remains to be solved. The generalization of deepfakes and the adaptability of detection techniques to new generative models remains an unsolved problem. In addition to these, beyond the performance of these models, the interpretability of the detection techniques remains another open problem. In Chapter **??** we will discuss different solutions to these problems and compare the performance of current detection systems with human performance.

2.3 Continual learning and multimodal learning

Continual learning [60, 160], also known as incremental learning or lifelong learning, refers to the ability of a learning algorithm to learn from a potentially unlimited stream of data where all data is not available at once. In such a setting, continual learning algorithms may have to deal with imbalanced or scarce data problems [272], catastrophic forgetting [81], or data distribution shifts [84]. Online learning can be thought as a special case [131] of continual learning where updates are done on per single data point basis and therefore, the batch size is one.

In terms of strategies, we can identify five main approaches to continual learning. (1) *dynamic architectures approaches* [192, 246, 334] dynamically modify the architecture of a model to make it learn new concepts or skills without interfering with old ones; (2) *regularization approaches* [88, 105, 158] consist of constraining the weight updates during learning in order to keep the memory of previous knowledge; (3) *rehearsal approaches* [98, 187, 193] gather all methods that save raw samples as memory of past tasks to use them to maintain knowledge; (4) *generative replay approaches* [70, 159, 259] train generative models on the data distribution, therefore, making them able to sample data from past experiences afterward when learning new data; (5) *hybrid approaches* [213, 241, 271] combine the previous strategies to tackle catastrophic forgetting.

2.3.1 Multimodal fake news detection

Studies show that uninformative content spreads faster than quality information [291], particularly when accompanied by visual content [322]. The different speed and virality with which such content spreads have seen a growing interest in new methodologies that exploit the diffusion patterns of news on social networks as the main signal for the classification of contents [25, 39, 182, 215, 239, 265, 318]. Although these solutions reach state-of-the-art performances, their applicability remains limited to a few companies that have access to the entire network, and, therefore, hardly applicable for fact-checking newspapers. For these reasons, most studies on fake news recognition have focused on content-based classification techniques [7, 8]. The advantage of these approaches, in addition to easier accessibility to this information, lies in the possibility of recognizing false content starting from the semantic analysis of the news without requiring external information such as, for example, user interaction patterns with the content. If the first works in this sense focused on the analysis of the text of the news [19, 21, 22, 235, 248, 254, 288, 291], more recently, the problem has begun to be tackled from a multimodal perspective, thus considering both the text and the images associated with it.

Multiple modalities are commonly combined with three approaches: (1) *early-fusion* methods [130, 132, 269, 315, 323, 339] learn low-level features from different modalities that are immediately fused, and fed into a single prediction model, (2) *late-fusion* models [5, 41, 231] fuse unimodal decisions with some mechanisms such as averaging and voting, and (3) *hybrid-fusion* [66, 129, 143] combines early fusion and late fusion. Using VisualBERT [166], MMBT [142], and ViLBERT [181], Dimitrov et al. [66] evaluated several fusion techniques (such as early-fusion, late-fusion and self-supervised models) for propaganda identification. According to their research, self-supervised joint learning models, and in particular VisualBERT, outperform other fusion techniques. As we will discuss in Capther 5.1, our Tri-Encoder falls into this last category, and as we will see in the experiments, it allows us to obtain better results than the other approaches. In terms of supervised learning, adversarial learning and autoencoder-based [140] models proved successful. Among these, Wang et al. [302] proposed an event adversarial neural network to detect emerging and time critical fake news. The model is designed to be robust to new topic that could emerge over time. Compared to this method, we propose an incremental learning strategy that can improve on new topics over time and test its robustness on several datasets.

As for continual learning in disinformation detection, the scientific literature still needs to catch up with more contributions. Most efforts in this direction have focused on applying graph neural networks that analyze social interactions or news propagation [25, 97, 239]. Content-based methods are instead very few. Horne et al. [108] examine the impact of adversarial content manipulation by malicious news producers on unimodal text-based fake news detectors. Silva et al. [188] studied the performance of two multimodal online learning approaches: (1) updating the model only when it makes a prediction error, and (2) updating it after both error or success. Their study shows that even when the model is updated intermittently, the classifier can overcome the concept drift phenomena found in the relatively tiny variations between the performance gained by the classifiers in the immediate, uncertain, and delayed feedback. Compared to this study, in Chapter 5.1 we evaluate differently continual strategies and propose a multimodal encoder that achieves state-of-the-art performance in all our experiments.

Continual learning has only recently been applied to deepfakes [115, 144, 162]. In particular, the work of Li et al. [162] has laid the foundation to explore this direction by introducing a new benchmark dataset designed explicitly for this purpose.

2.3.2 Data poisoning methods

Data poisoning attacks [76] are a class of adversarial attacks that aim to manipulate the behavior of machine learning models by injecting malicious instances into the training data. These attacks exploit the vulnerability of models to the presence of misleading or biased data during the learning process, thus making them potentially more covert and challenging to detect. Indeed, by carefully crafting and injecting adversarial examples into the training dataset, attackers can influence the model's decision boundaries, leading to incorrect predictions or biased outcomes. Data poisoning attacks can take various forms, depending on the target model's specific characteristics and the attacker's goals [87]. For example, Biggio et al. [28] proposed a poisoning attack against an adaptive face recognition system. Some common types of data poisoning attacks include (1) *Poisoning with* *mislabelling*, where the attacker deliberately mislabels a subset of training instances to introduce incorrect or misleading information into the model, and (2) *Feature injection*, where the attacker adds carefully crafted instances with manipulated features to the training data. These instances are designed to bias the model towards specific patterns or characteristics, leading to skewed predictions. Precisely feature injection attacks can be particularly effective when the model heavily relies on specific features for decision-making. In this thesis, we are mostly interested in *backdoor attacks* [91], as it will be discussed in Capter 5.2. When a backdoor attack occurs, the attacker manipulates the training set to cause incorrect behavior at test time. However, test-time errors are only triggered in the presence of a triggering event. In this sense, the compromised network continues to act as expected for regular inputs, and the malicious behavior only occurs when the attacker decides to activate the backdoor hidden inside.

Data poisoning attacks can also be used in the context of fake news detection to manipulate the behavior of machine learning models trained to classify news articles as either *true* or *false*. The impact of these attacks on fake news detection can be severe: misclassifying true news articles as false can lead to a loss of trust in the model's predictions and potentially allow the propagation of harmful information. To the best of our knowledge, no previous research has focused on the specific problem of making a fake news detection model misclassify a true news article as false without modifying the news article itself. However, there are some works, such as the one from Price et al. [230], which focus on Twitter bots since they can mimic real-people text prompts, and the work from Campanile et al. [36], which focuses on the robustness of deep neural networks for fake news classification with respect to the poisoned world in text prompts.

Online learning methods can be vulnerable to data poisoning attacks [330], which can be designed for both *semi-online* and *fully-online* learning settings [251, 300]. The former applies when an online or streaming algorithm is used to train a classifier, which is used directly in a downstream application; therefore, the attacker seeks to modify the training data stream to maximize its objective on the classifier obtained at the end of training. In the fully online setting, the attacker seeks to modify the training data to maximize the accumulated objectives over the entire online learning window. It corresponds to adversaries in applications where an agent continually learns online, thus constantly adapting to a changing environment. Zang et al. [326] formulate the optimal online attack problem as a stochastic optimal control problem and provide a theoretical analysis of the regret suffered by the attacker for not knowing the actual data sequence. The study from Wang et al. [300] systematically analyzes data poisoning attacks for both learning strategies in typical computer vision classification tasks and proposes alternative defensive solutions. Similarly, Seetharaman et al. [251] propose a defense mechanism to minimize the degradation caused by the poisoned training data on a learner's model in an online setup. Li and Ditzler [163] focus on targeted data poisoning attacks against continual learning (incremental learning scenario) for fake news detection, which artificially forces the neural network to catastrophic forgetting. Horne et al. [109] show that poison attacks for fake news detection can harm the attacker more than the victim. According to their study, a significant decrease in performance is observed when the attacks begin and are maintained throughout time. After the attack time frame is over, the algorithm almost immediately recovers to its original performance.

2.4 Forensic applications

This section focuses on two practical forensic applications that will be discussed in Chapter 6: (1) the geolocalization of the area captured in an image and (2) the similarity analysis between images for anti-fraud applications in the insurance domain.

Journalists and fact-checkers are usually required to apply content verification techniques to confirm the truthfulness of images, videos, and claims before reporting or sharing them. In Chapter 2.4.1, we will focus our review on ground-to-aerial image matching techniques, which are techniques used to compare and verify the content of images taken from different perspectives or sources, particularly from ground-level and aerial perspectives. Ground-to-aerial image matching can be used to confirm the location and details of viral photos or videos to prevent the spread of misinformation or fake news. This overview will be helpful to the reader to understand the contribution in this field introduced in Chapter 6.1.

Another common forensic application when discussing content verification is understanding whether an image has been retouched by introducing or removing elements. This problem is also significant in some industrial sectors, such as insurance, where ideas can constitute evidence for a damage compensation claim. In Chapter 2.4.2, we present an overview of current solutions focusing, on the one hand, on the recognition of damage on a vehicle and, on the other, on the problem of finding two similar images within a database.

2.4.1 Reconstructing the location of images

Generally speaking, the matching between images taken from an overhead point of view and the ground level is a fundamental task in computer vision applications due to the high amount of available information that can be extracted. Before tackling the ground-to-aerial problem, researchers focused their attention on ground-to-ground image matching. Hays et al. [99] proposed the first data-driven method that sorted out the problem of geo-localization from ground-level images. However, this solution relied more on scene categorization rather than localization retrieval. Another typical technique to comprehend relationships among images for data collection and geolocalization is based on 3D reconstruction [4, 40, 255] and geometric constraints, both in urban and natural environments. Baatz et al. [20] focused on mountainous areas to pull out the recognition of the skyline given a digital elevation model of a country. Following this concept, Lin et al. [170] proposed the first approach using ground and aerial images to retrieve geolocalization via a data-driven approach. Satellite images are now much more widespread and cover every region of the planet, offering a conspicuous advantage over terrestrial photos, which can be more challenging to collect. This offers a substantial advantage in forensic applications.

Recently, with the advent of deep learning, AI-based algorithms have been exploited to improve the matching performances of previous methods. Three are the main architectural structures employed to handle the task: (i) the siamese-like networks, (ii) the generative adversarial networks (GANs) and (iii) transformer networks. The goal of Siamese-like architectures is to extract shared features between the ground point of view and the overhead one. Subsequently, the distance between extracted features is computed in order to understand if there is a relationship between the two or if there are some features that are immune to the significant shift in the point of view. Lin et al. [171] are the first to introduce the Where-CNN, a Siamese network that achieves superior results when compared to traditional hand-crafted features. Subsequently, Vo and Hays [289] and Shi et al. [257] focused on recovering orientation besides location using a soft-margin triplet loss on top of a Siamese CNN network and a siamese-like network relying on polar coordinate mapping, respectively. Whereas, Hu et al. [113] insert the NetVLAD [16] layer in addition to a Siamese network to identify features that are consistent with large changes in perspective. Shi et al. [258] aims at transforming the features from the ground domain to the aerial one utilizing a novel feature transport module together with a Siamese network. Other methods exploit GANs to synthesize images related to the two viewpoints and use them as additional information useful for obtaining a better understanding of the scene. Deng et al. [64] propose a GAN-based methodology in which the generator produces a ground-level image from the aerial point of view that is then compared with the real ground query image to retrieve the matching. On the other hand, Regmi et al. [237] aim at generating an aerial viewpoint of a ground-level panorama query image in a way that the transformed image has scene representations similar to the images it is matched against. It is also worth mentioning more recent deep learning-based approaches that rely on vision transformers (ViT). For example, Tian et al. [282] suggest a conditional GAN combined with a Transformer to synthesize an aerial image that appears with the same style as the ground view. Whereas, Zhang et al. [325] work towards the use of limited field-of-view images captured from more common devices such as smartphones and digital cameras instead of panoramic ground images, grasping sequential spatiotemporal features via the implementation of a VGG16 and temporal feature aggregation module inspired by the ViT architecture. Even if these deep learning-based solutions achieve excellent performance, they are still difficult to explain and, therefore, difficult to apply in forensic scenarios in which it is necessary to justify the output of the analysis. To overcome these issues, in Chapter 6.1, we proposed an algorithmic solution for this task. Moreover, state-of-the-art deep learning methods are designed in a supervised setting, while here, we propose an unsupervised approach that leverages the view-independent adjacency properties of visible landmarks to create comparable graph structures.

2.4.2 Image similarity for the insurance domain

Insurance companies receive thousands of new claims every day, each containing several images. After being taken over, insurance company experts must analyze all the images of a claim to decide how to conclude the compensation process. For the larger insurance companies, this translates into having to analyze millions of images every year. However, managing such a large amount of data is extremely expensive in terms of human resources and the cost of maintaining these processes. Furthermore, detecting fraud attempts on millions of claims is an even more complex task. As a result, many insurance companies have started developing image-analysis solutions to automate part of the claim management process. In the following sections we focus on the fundamental blocks of the pipeline proposed in this study: (1) damage recognition systems and (2) deep learning techniques for object reidentification in images.

Damage Detection

The enormous heterogeneity of damages and the lack of large labeled datasets make it difficult to train robust damage classifiers. In addition to this, being a very different task compared to traditional object detection tasks in which a certain object to be identified has a more or less homogeneous shape, it is not obvious that using pre-trained models can improve the performance of a damage classifier. In sight of this [224] consider a wide range of damages such as dent, glass shatter, broken lamp, scratch or smash, and propose a series of experiments in which they compare the effectiveness of different approaches including (1) training a CNN, (2) unsupervised pretraining of an auto-encoder followed by fine-tuning, (3) using of transfer learning from CNN trained on ImageNet [61] and (4) creating an ensemble classifier on top of the set of pre-trained classifiers. A similar approach is proposed by [155], who collect a dataset of damaged and undamaged car images from the web and fine-tune a pre-trained VGG [266] with L2 regularization to contain overfitting. The results from Patil et al., show that transfer learning combined with ensemble learning works best. However, ensemble learning can be computationally expensive, leading to the increase of maintenance cost of an automated claim management pipeline. On the contrary, in this study, we propose a simpler approach that allows us to obtain acceptable performance without requiring too many computational resources. With the same hardware available, this allows you to optimize the claim management process without increasing processing times or costs per image.

Accurately recognizing damages is not enough to automate the entire damage detection process. To use these methods in production it is first of all necessary to select only the images that may contain the damage. For this reason, in many cases, damage detection models are often preceded by other models that deal with filtering images containing vehicles and are used in parallel with another car-panel detection system that allows damage to be localized on the bodywork. In this direction, [23] propose an approach based on a pipeline made up of four models: (1) a filter that discards images that do not contain cars, (2) a classifier that identifies damages to the bodywork, and two parallel classifier estimating (3) the severity of the damage, and (4) position (side, rear, front). [138] propose a similar methodology. In terms of deep-learning architectures, the Mask R-CNN [100] is a common and accurate solution for damage and panels detection [324, 345]. [345], propose a pipeline consisting of a Mask R-CNN for identifying vehicle panels, a RetinaNet [172] used for damage recognition and an Inception-V3 [278] network that classifies the type of damage and the corresponding severity. Differently, in our work, the localization of the damage is obtained by classifying the view of the vehicle, that is back, front, left, right, back–left, and so on. However, the complexity of car-damage detection and segmentation may lead to lower detection segmentation accuracy and slower detection speed. Therefore, [324] propose a modification of the ResNet-50 [101] network. By reducing the number of layers in the residual network, and adjusting the internal structure to strengthen the regularization of the model, they enhance its generalization ability. Compared to these works, in this study, we choose to adopt the Mask RCNN for damage detection and to filter possible matches based on the view of the vehicle. Through our experiments, we show that this model performs really well in a real setting. Moreover, our pipeline is based on a filtering step that selects images containing vehicles, a vehicle detection module that retrieves the position of the vehicle in the image, and brand and color detection systems that extract information about the car. All these modules are in handy to produce an end-to-end damage detection system.

Deep-Reidentification Architectures

The lack of data and understanding of the challenges associated with insurance fraud by people outside the insurance business has not attracted the scientific community's interest in these problems.

To our knowledge, the only work dealing with damage reidentification has been proposed by [167]. The paper proposes to use the YOLO [236] network as a local damage detector and a pre-trained VGG16 as a global feature descriptor. By fusing the features extracted by the last convolutional layer of the VGG16 with a color histogram, they obtain a more discriminative global descriptor. The local and global descriptors are finally concatenated and compared with an image history via the cosine distance. Differently from [167], in this study, we cast the problem to a *reidentification* task, similarly to what has been done for person reidentification [174, 208, 310]. Whereas [167] use the color and global features to make descriptors more robust, we propose to use them to filter the possible pairs to compare. Indeed, comparing every possible damage (which we consider as a *query*) with an insurance company's database containing all the previously checked claims could require an unsustainable number of comparisons. For this, we propose to filter the images based on the color, the brand, and the panel on which the damage is located, and we use this information to retrieve possible matches containing near duplicates to the new one. [332] use a similar approach for the car-reidentification task. In their work, the car attributes are divided into two categories: special attributes and common attributes. Special attributes reflect the car's unique characteristics, such as individual paints or car damage, whereas common attributes denote the car's inherent appearance. Using specific attributes to re-rank results has been shown to increase retrieval performance. In our setting, however, we are interested in reidentifying damages, which represent one of the unique attributes of a car. Therefore, we chose to filter images based on their common characteristics to reidentify damage by only looking for it on vehicles of the same model, panel, and color as the query image.

Searching for damages is extremely challenging, as those may appear with a cluttered background and occlusion. In addition, the queried damage can appear in the gallery from different viewpoints, scales and lighting or reflection conditions, which makes this scenario very similar to that of the reidentification of objects, where an object can appear under different views and conditions. Most existing deep reidentification convolutional neural networks (CNNs) [6, 92, 168, 256, 276, 283, 301] borrow architectures designed for generic object classification problems. However, these architectures are designed to treat objects with the exact and fixed shape that characterizes them. This does not apply to damages, which are harder to be matched to a shape. Each damage has unique distinguishing characteristics because of its typical irregular shape. Consequently, using CNNs designed to recognize objects of regular shape more easily leads the models to overfit on the training set without being able to really learn useful information for our task. An interesting idea to cope with this kind of problems comes from [336], who propose the OSNet, a network that learns multiscale features explicitly at each layer of the network. This is accomplished by a residual block composed of multiple convolutional streams, each detecting features at a certain scale. Then, a unified aggregation gate fuses multi-scale features with input-dependent channel-wise weights. To efficiently learn spatial-channel correlations and avoid overfitting, the building block uses pointwise and depthwise convolutions. Thanks to this structure, the OSNet turns out to be extremely lightweight and less prone to overfitting. For these reasons, in this study, we propose a damage reidentifier based on an OSNet backbone and compare its performance with other state-of-the-art methods, specifically, [33] and [63].

In parallel with the drafting of this study, new reidentification strategies based on attention mechanisms have been proposed. [103] introduced a transformer-based object reidentification framework, which is made of a patch module that rearranges patch embeddings by shift and shuffles operations. This results in robust features with increased discriminating ability and side information embeddings that counteract feature bias towards camera-view fluctuations by including these non-visual cues into learnable embeddings. [344] propose a similar approach, which uses a dual cross-attention learning algorithm to coordinate with self-attention learning, and they show that it reduces misleading attentions and diffuses the attention response to discover more complementary parts for recognition. These works are based on very deep models, which are helpful in tasks where many training examples are available. In our case, however, we do not have enough labeled data to justify the use of these models. Therefore, we decided to propose a less complex network, such as OSNet, which helps us control model overfitting.

Finally, in this section, as well as in the ones just discussed so far, we have a labeled dataset available and we treat the problem with a supervised approach. In addition, there are unsupervised approaches based on contrastive learning [55, 173] or noncontrastive [96] learning techniques. These approaches have not yet been explored on the damage reidentification task, and we leave this possibility open as a possible extension of this study in the future.

Chapter 3

Platform provenance analysis

As explained in the previous chapters, the reconstruction of the source of multimedia content is a highly relevant task in forensic scenarios. This chapter proposes two methodologies to understand if images and videos have been shared on a social network. This task has become increasingly important in recent years as the information posted online has grown exponentially. Every day people watch over a billion hours of video on YouTube [121] and share more than a billion stories on Facebook [75]. Multimedia content, especially images and videos, attract the attention of online users more efficiently; for this reason, they have become a favorite means of disseminating content. Precisely for this reason, it becomes increasingly important to verify the source of this information, which in many cases can represent tangible evidence of events. In particular, two problems must be solved from a forensic perspective: (1) *knowing how to reconstruct the source of acquisition (camera model or device)* and (2) *understanding whether some media content found on an offending device comes from social media*, which will be the focus of this chapter. Being able to respond to the latter would allow the sharing network to be reconstructed and possible online criminal groups to be identified, as shown in Figure 3.1. This could be helpful in different applications such as, for example, cyberbullying, where we want to be able to investigate who and where this individual has shared certain content. Similarly, this tool could be helpful to trace the sharing of videos of military propaganda or other criminal activity back to the source, as well as for fact-checking and countering fake news.

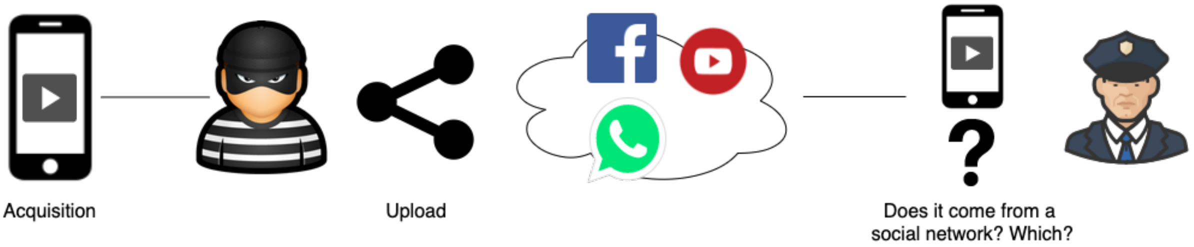


Figure 3.1: An application example of the proposed solution. An attacker records a video with illegal content and shares it on social networks or messaging apps. Subsequently, the police seize a device with this video and want to trace the source.

As discussed earlier, when we upload a video to a social-media platform, it usually goes through a series of operations, which most commonly may include recompression to reduce the bandwidth requirement for using the video on the platform, a resize, and in some cases the removal of some frames of the video to make it fit the maximum duration of the videos imposed by some platforms. While, as mentioned, these operations may vary depending on the platform, in this chapter we want to formalize as much as possible how these operations can leave information in the video.

In the rest of this chapter, we propose two methods to detect the fingerprint left by social media when content is uploaded to the platform. The first [9], presented in Chapter 3.1, is a seminal work that extends the application of these techniques from images to videos for the first time. Before this study, all the studies had focused on reconstructing the social platform of origin of pictures. This method demonstrates that it is possible to identify traces left by the platform in videos similarly to what is done in pictures. Still, it highlights a lack of large enough video datasets to allow further investigations on this topic. In the second work [190], presented in the Chapter 3.2, we propose a multi-task learning strategy that will enable us to circumvent this problem by exploiting the shared features between images and videos for this task.

3.1 Social network identification for videos

This section presents our first study [9] on social network identification for videos. To address this problem, we propose a multistream neural network architecture that can capture the double compression traces left by social networks and messaging apps on videos. According to our knowledge, this is the first work that investigates whether it is possible to recognize videos from different social networks by analyzing the traces of compression left by these websites when loading content. Our model achieves an accuracy of 95.51% in recognizing the source of three types of videos: (1) shared on YouTube, (2) shared on WhatsApp, and (3) the original equivalents. Moreover, we investigate the possibility of detecting the origin of images once the network is trained on videos. For this purpose, we use the pre-processed network on WhatsApp videos to distinguish the images shared on the platform from the original ones and show that we can still achieve 92.74% accuracy in this experiment.

3.1.1 Proposed Method

In this section, we describe the proposed architecture (see Figure 3.2) composed by a two-stream network, inspired by the work by Nam et al. [209]. However, the application of this particular network to the problem that we study is novel and it requires some important modifications to the method in [209]. First, we modified the third convolutional block of the Ind-Net removing a stack of Convolutional, Batch Normalization, and ReLU operations and we added one more convolutional block (Block 6) at the end of the CNN. This deeper configuration helps the network to capture more subtle details in the input. Next, we modified the Pred-Net by doubling the number of operations in each block and increasing the number of output channels of each block in order to learn a richer representation. Finally, we changed the dimensionality of the flattened feature maps from 128 to 256 for the P-frames stream and from 16,384 to 4,096 for the IF-stream. This helps to limit the importance of I-frames over the P-frames. We choose not to include B-frames in our analysis because of the lower quality of this kind of frames. Finally, we introduce a two-stream network (MultiFrame-Net), which learns the inter-modal relationships between features extracted from both types of frames. In the rest of this section, we use the notation $W \times H$ to denote the resolution of a video v. Each video can also be represented by N frames denoted as $f_0, \ldots, f_{N-1}$,

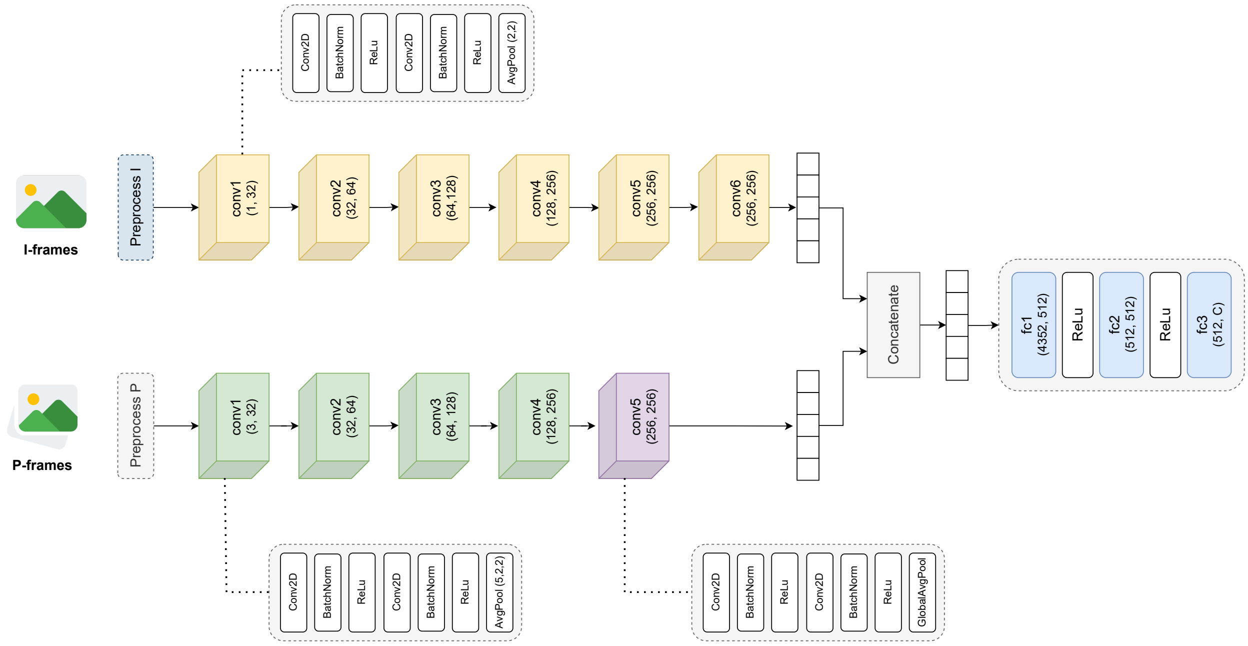


Figure 3.2: The proposed two-stream network (MultiFrame-Net) architecture. The network is constructed by concatenating the feature maps of the Ind-Net and the Pred-Net. The I-frame and P-frame streams are trained separately. Next, we concatenate the flattened output of the two-streams and train a fully connected classifier.

where $f_j \in \mathbb{Z}^{3\times W\times H}$. Moreover, we use the notation $f_{Ii}^{(v)}$ and $f_{Pi}^{(v)}$ to denote the ith I-frame or P-frame, respectively, of a video v.

Ind-Net

In this section we propose a network that analyzes the I-frames of a video. The network is depicted in the bottom part of Figure 3.2. We designed a network that consists of six convolutional blocks that act as a feature extractor and a fully connected network that takes the input feature vector and produces an output classification. The first tree convolutional blocks made of (1) two consecutive stacks of convolution (Conv2D), batch normalization (BatchNorm), and ReLU operations, and (2) a final max pooling (MaxPool) layer. The last three convolutional blocks are organized in three consecutive stacks of (1) Conv2D, BatchNorm, and ReLU operations, and (2) a final MaxPool layer. Apart from the first convolutional layer, which has a 5 × 5 kernel, all other convolutional layers have a 3 × 3 kernel. The feature extracted by the last MaxPool layer becomes eventually flattened and passed through two stacks made by a 512-dimensional fully connected layer and a ReLU, and a final 512-dimensional fully connected layer followed by a softmax one. The network outputs a $|C|$-dimensional vector, where $|C|$ is the number of output classes.

Before being fed into the network, the decompressed I-frames are initially transformed through a preprocessing module. To highlight the traces left by a double compression, we employ the high-pass filter introduced by Fridrich and Kodovsky [82, operator S5a], and used in [209] and apply it to the Y-channel of the input after RGB-to-YUV conversion. Therefore, we denote as $X_{Ii} = \{f_{Ii}^{(v)}\} \in \mathbb{Z}^{3\times W\times H}$ the input ith frame of video v and compute $X'_{Ii} = \{HPF(Y(f_{Ii}^{(v)}))\} \in \mathbb{Z}^{W\times H}$ to obtain the preprocessed input of the network, where $HPF(\cdot)$ indicates the high pass filter and $Y(\cdot)$ indicates the Y-channel of the input frame. Because we assume that each video could come from a single social media platform, we train the model using a cross-entropy loss function, thus training the model to output a probability over the $|C|$ classes for each video.

Pred-Net

Now we present Pred-Net, a new network architecture that analyzes the P-frames of a video to detect double compression fingerprints. The network (depicted in the top of Figure 3.2) is made of five convolutional blocks and a fully connected network. All the convolutional blocks consist of two stacks of (1) Conv2D, BatchNorm, and ReLU operations, and (2) a final average pooling (AvgPool) layer. The AvgPool and GlobalAvgPool levels help to preserve the statistical properties of feature maps that could otherwise be distorted with the MaxPool. All the Conv2D layers in the first two blocks have a 5×5 kernel, and the last three blocks have a a 3×3 kernel. Finally, the feature maps extracted from the last convolutional block are flattened and passed through a 256-dimensional fully connected layer that outputs a $|C|$-dimensional vector and a softmax operation that calculates the output prediction.

Similarly to the Ind-Net, we add a preprocessing step to the input frames in which a high-frequency–component extraction operation is applied to eliminate the influence of diverse video content. Further, because the P-frames represent predicted low-quality frames, we compensate for the loss of information by stacking consecutive frames. In fact, given a stack of three consecutive P-frames denoted as $X_{Pi} = \{f^{(v)}_{Pi-1}, f^{(v)}_{Pi}, f^{(v)}_{Pi+1}\} \in \mathbb{Z}^{3\times3\times W\times H}$, we compute $X'_{Pi} = \{Y(f) - G(f) | f \in X_{Pi}\} \in \mathbb{R}^{3\times W\times H}$, where the function $G(\cdot)$ denotes a Gaussian filter. Like the Ind-Net, the network is trained with a cross-entropy loss function.

MultiFrame-Net

Multistream architectures have been successfully applied by multimedia forensics researchers for both forgery detection and source identification tasks [11, 13, 200, 285]. Therefore, we combine the feature maps of both Ind-Net and Pred-Net to feed the fully connected classifier with inter-modal relationships between different types of frames. As shown in Figure 3.2, we concatenate the output features maps of the two CNNs and feed them to the classifier. The concatenated features vector is a $4,352$-dimensional vector obtained by integrating the $4,096$-dimensional output vector of the Ind-Net and the 256-dimensional output vector of the Pred-Net.

In our setting, we train the Ind-Net and Pred-Net separately and exploit the weights of the pre-trained convolutional blocks of these networks to train the fully connected classifier. As for the Ind-Net and Pred-Net, we train the model according to a cross-entropy loss function.

3.1.2 Experimental Evaluation

This section describes the experimental setup and the tests that have been carried out to evaluate the robustness of the proposed approach. We begin describing the dataset and configurations used for this study, then, in sections 3.1.2 and 3.1.2 we discuss the results that we obtained on several tests.

All the experiments discussed in this section were conducted on a Google Cloud Platform n1-standard-8 instance with 8 vCPUs, 30GB of memory, and an NVIDIA Tesla K80 GPU. The networks have been implemented using Pytorch v.1.6 [203]. We trained all the networks with the learning rate set to $1e-4$, weight decay of the L2-regularizer set to $5e-5$, and Adam optimizer with an adaptive learning rate. In our experiments, we trained the networks for 80 epochs with batches of size 32 and early stopping set to 10.

To train our model and evaluate its performance, we relied on the VISION dataset [263]. The dataset comprises of 34,427 images and 1,914 videos, both in the native format and in their social media version (i.e., Facebook, YouTube, and WhatsApp), captured by 35 portable devices of 11 major brands. The dataset has been collected recording 648 native single-compressed (SC) videos, mainly registered in landscape mode with *mov* format. For each device, the videos depict flat, indoor, and outdoor scenarios and different acquisition modes. The resolution varies from 640×480 up to 1920×1080 depending on the device. Furthermore, the dataset contains 622 videos that were uploaded on YouTube (YT), and 644 shared through WhatsApp (WA). Similarly to videos, the dataset also contains images captured in multiple orientations and scenarios and shared via Facebook and WhatsApp.

In our experiments, we previously process the dataset with the *ffprobe* [80] analyzer from the *FFmpeg* software to extract the I-frames and P-frames from a subset of 20 devices. Next, we crop each frame into nonoverlapping patches of size $H \times W$, where $H = W = 256$, obtaining 153,843 I-frame patches and 209,916 P-frame patches. Finally, we balance all classes and split the dataset for training, validation, and test with a proportion of 70%, 15%, and 15%, respectively.

Results on Shared Videos

To estimate the performance of our method, we initially compared the system with respect to a baseline model. Then, we moved forward to assess the performance of our two-stream architecture, namely to validate the increase in performance obtained combing the Ind-Net and Pred-Net.

1) Baseline comparison: In our first set of experiments we measured the performance of the single components of MultiFrame-Net (the Ind-Net and Pred-Net streams) with respect to the baseline model introduced by Nam et al. [209], for their classification efficacy when using only I-Frames and P-Frames, respectively. To limit model training time, we chose to conduct these experiments on a subset of 10 devices from the VISION dataset. In fact, in this test, we are not interested in obtaining the absolute best performances, but we limit ourselves to proving that there is a boost in performance compared to the baseline. For these experiments, we produce an 80%-10%-10% split of the dataset of the input patches for training, validation, and testing, respectively. For this first experiment, we model the problem as a binary classification task, i.e., YouTube and Whatsapp videos are considered shared videos, while single-compressed videos are treated as original ones.

	Nam et al. [209]	**Proposed method**
I-Frame	67.71%	**88.42%** (Ind-Net)
P-Frame	67.23%	**76.84%** (Pred-Net)

Table 3.1: Accuracy on a subset of 10 devices from the VISION [263] dataset. The proposed method is confirmed to be more precise than the baseline at recognizing traces left by social networks and apps on frames patches.

The results reported in Table 3.1 confirm the significantly improved performance of our method with respect to the baseline. In fact, the deeper architectures help to distinguish with higher accuracies (88.42% and 76.84% for the Ind-Net and Pred-Net, respectively) between different types of double compressions left by social media and messaging apps. Indeed, the model must be able to distinguish not only between single and double compression but also between different types of double-compression fingerprints. In this sense, a deeper architecture is capable of extracting more complex information.

2) MultiFrame-Net evaluation: In this test, we evaluate whether and to what extent our two-stream architecture (MultiFrame-Net) improves even more in terms of accuracy compared to the single streams. For this experiment, we trained and evaluated the models on a subset of 20 devices with a dataset split of 70%, 15%, and 15% for training, validation, and test, respectively. First, we train the Ind-Net and Pred-Net in an end-to-end fashion on a subset of 15 devices. Next, applying transfer learning, we froze the convolutional layers of both networks and retrained the fully connected classifier on a subset of 5 devices that have not been used on the previous training. For the first experiment, we treat the problem as a binary classification problem as we did for the previous experiment. We measure the performance of each network with respect to its accuracy and its area under the curve (AUC) score. Table 3.2 reports the results of this experiment. In our second experiment, reported in Table 3.3, we model the problem as a multiclass classification problem. Table 3.3 represents the confusion matrix of MultiFrame-Net. The experiment confirms that by combining the classification of different types of frames, the model achieves better performance, with the MultiFrame-Net gaining up to 95.51% of accuracy and 96.44% of AUC score on patches from SC, WA, and YT. Moreover, the confusion matrix (see Table 3.3) of the MultiFrame-Net on 3,749 patches from 234 unique videos from WA, YT, and SC suggests that the errors are very small and slightly more numerous in the case of SC patches.

Model	**Accuracy**	**AUC**
Ind-Net	92.32%	94.24%
Pred-Net	91.87%	93.12%
MultiFrame-Net	**95.51%**	**96.44%**

Table 3.2: Model accuracies and AUCs on a subset of 20 devices from the VISION dataset [263]. The MultiFrame-Net shows higher performance with respect to Ind-Net and Pred-Net. For this experiment, we model the problem as a binary classification task, i.e., shared or original videos.

	YouTube	**Whatsapp**	**Single-compressed**
YouTube	**1238 (96.41%)**	20(1.65%)	32(2.55%)
Whatsapp	31(2.41%)	**1161 (95.79%)**	49(3.91%)
Single-compressed	15(1.16%)	31(2.55%)	**1172 (93.53%)**

Table 3.3: Confusion matrix of the MultiFrame-Net over YT, WA and SC patches from 234 unique videos of the VISION dataset [263].

Results on Shared Images

In our last experiment, we measure the robustness of the Ind-Net with respect to images. Specifically, we moved from the intuition that I-frames are independently encoded using a process similar to JPEG compression, such that it could be possible to detect images as well as videos coming from the same social media platform. For this reason, we test the Ind-Net trained on videos, on native and WhatsApp images available on the VISION dataset. Unfortunately, the VISION dataset contains images uploaded only on WA and Facebook. Therefore, we can apply this test only on WA images. We began the experiment by training the Ind-Net on native and WA video patches obtaining 92.74% of accuracy. Next, by applying transfer learning, we froze the convolutional blocks of the network to act as feature extractors and retrained the fully connected classifier on images from the same classes. With minimal retraining of the classifier, it achieves 86.83% of accuracy. This result suggests that a mixed method to trace both kinds of media is actually possible. Therefore, we leave this problem for future research and extensive experiments.

3.2 Image and video source identification through the use of shared features

Deep learning has pushed the design of new methods that can learn forensic fingerprints automatically from data [53, 116, 199], helping us to take a new step towards applying these techniques to real problems. Despite the promising results of artificial neural networks, some limitations still remain. Single-task learning has been very successful in computer vision applications, with many models performing as well or even exceeding human performance for a large number of tasks; however, they are extremely data-dependent and poorly adaptable to new contexts. Recently, collecting data from social networks has become increasingly difficult because of data protection regulations and the most stringent policies introduced by the platforms (`https://www.facebook.com/apps/site_scraping_tos_terms.php`, `https://twitter.com/en/tos`—accessed on 4 August 2021). Indeed, it is mandatory to obtain end-user consent or the platform's written permission before acquiring data via the API or web scraping of the most common social networks like Facebook, Instagram or Twitter. Moreover, new data protection regulations, such as GDPR (`https://europa.eu/youreurope/citizens/consumers/internet-telecoms/data-protection-online-privacy/index_en.htm`—accessed on 4 August 2021), CCPA (`https://oag.ca.gov/privacy/ccpa`—accessed on 4 August 2021), or the Australian privacy act are contributing to the introduction of new limitations in some countries around the world. All these limitations make it difficult to collect enough data to train a deep-learning model. Moreover, the human ability to learn from experience and reuse what has been learned in new contexts is still difficult to reproduce in machine learning as well as in multimedia forensics. All these reasons, along with the unavailability of large training datasets containing both video and image content, have led researchers to treat the problems of social media–platform identification of images [12, 14, 227, 264] and videos [124] separately. Recently, Iuliani et al. [122] showed that it is possible to identify the source of a digital video by exploiting a reference sensor pattern noise generated by still images taken by the same device, suggesting that images and videos share some forensic traces. Based on this intuition, we build a model that classifies videos from different social-media platforms or messaging apps by taking advantage of the shared features between images and videos. More specifically, to overcome the aforementioned limitations, we try to answer the following question: *Is it possible to decide whether a video has been downloaded from a specific social media platform? If so, do images and videos have any common forensic trace that can be used to solve video social media platform identification using both media?* To answer these questions, we propose two methods: A method based on transfer learning and one based on multitask learning. Both methods offer the possibility of reusing the features learned from one media into another using fewer training data, a feature that is very useful in this domain given the difficulty of finding datasets large enough to train neural networks.

In *transfer learning*, we first train the base model on the image task, and then reuse the learned features, or *transfer* them, to videos. This process tends to work if the features are general, that is, suitable to both tasks [317]. The forensics community has adopted widely transfer learning because, as new manipulation methods are continually introduced, there is a need of detection techniques that are able to detect fakes with little to no training data [52, 321]. In *multitask learning*, a model shares weights across multiple tasks and makes multiple inferences in a forward pass. This method has proved to be more scalable and robust compared to single-task models, allowing for successful applications in several scenarios outside the forensic community [327]. Some applications of multitask learning have been even applied to multimedia forensics problems as well, for example, to solve camera model and manipulation detection tasks [198], as well as brand, model, and device-level identification, using original and manipulated images [67].

We apply both learning approaches in this study to accelerate the training of a deep-learning method for deciding whether a video has been downloaded from a social media platform. Because the collection of large datasets for this task is usually very difficult, if not impossible, in practical applications because of privacy reasons, it is worth investigating the effectiveness and the limits of transfer learning and multitasking learning on the task of social media platform identification of videos.

In this study, we show how well low-level features generalize between images and videos, demonstrating that common platform-dependent features can be learned when the training data are not large enough to train a deep learning model from scratch to estimate the traces left by social media platforms during the upload phase on videos. The sharing process can combine multiple operations that leave different traces in the video signal. These alterations can be exposed in various ways. For example, as first observed in [206], compression and resizing are usually applied by Facebook to reduce the size of uploaded images and this may happen differently on different platforms based on the resolution and size of the input data before loading. As is widely known in multimedia forensics, such operations can be detected and characterized by analyzing the video signal where distinctive patterns can be exhibited. Indeed, these operations typically distort the original video signal with some artifacts that can be detected. When the signal is used as a source of information for the provenance analysis, different choices can be made to preprocess the signal and extract an effective feature representation. After the feature representation is extracted, different kinds of machine-learning or deep-learning classifiers can then be trained to perform platform identification. To detect videos shared through social media platforms, we propose two methods that can learn to detect the traces left by different social-media platforms without any preprocessing operation on the input frames. To our knowledge, this is the first work that analyzes the similarity of the traces left by social media platforms on images and videos used in combination. Next, we show that the features learned in the task of social-media identification of images can be successfully applied on social-media identification of videos, but not vice versa, thus suggesting a *task asymmetry*, which could possibly be explained by looking at social-media identification of videos as a special case of the image task. Indeed, as discussed in the introduction of this Chapter, shared videos may have both static and temporal artifacts, whereas shared images have static features only. These findings are particularly valuable in investigative scenarios where law-enforcement agencies have to trace the origin of multimedia content without being able to refer to other sources such as metadata. This scenario is depicted in Figure 3.1.

3.2.1 Proposed Method

In this section, we propose an analysis of what could be the traces that can be left on videos by social media and we propose a framework for platform identification.

Social Media Platform Identification Framework

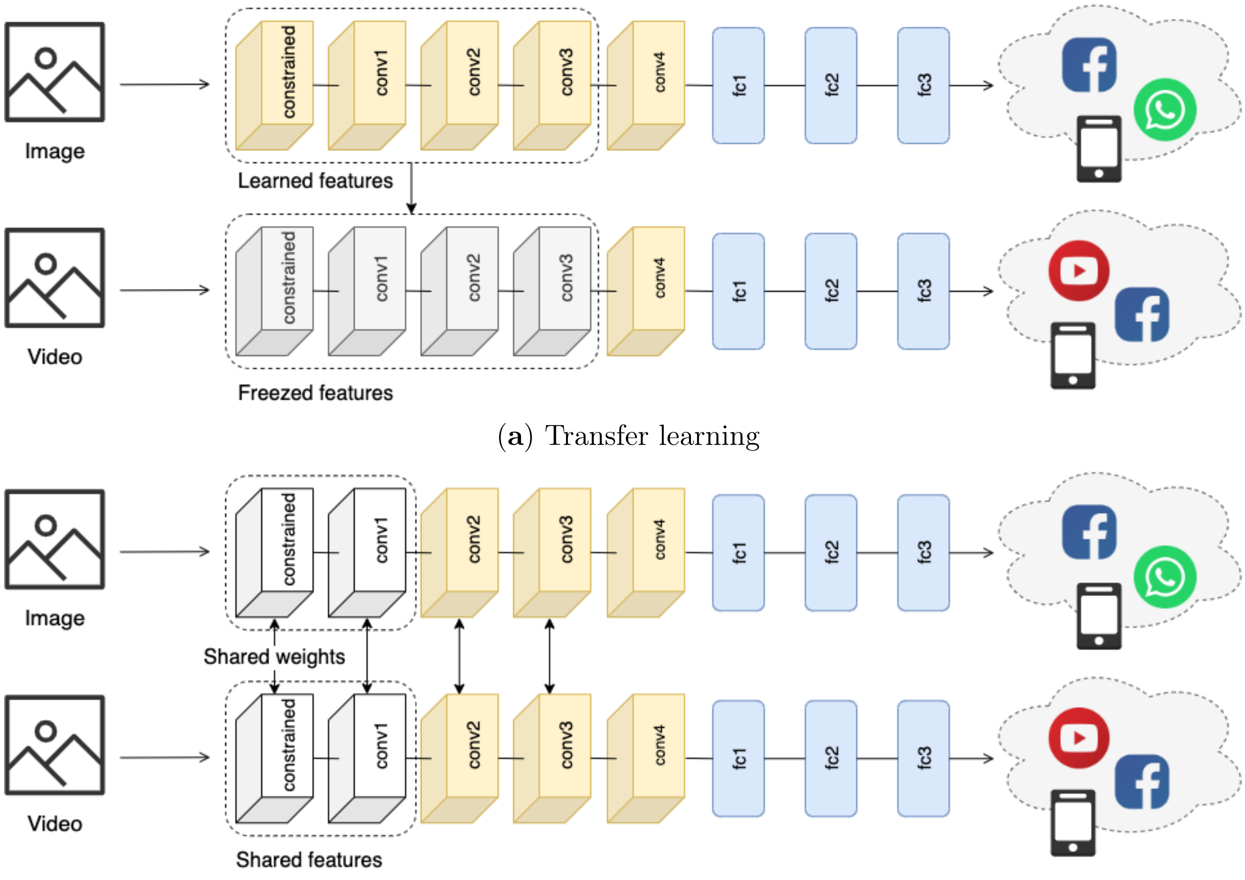
In this section, we propose two learning methods to detect and classify different static and temporal recompression fingerprints left by social media platforms on shared videos exploiting a unified set of features. Through these learning methods, the goal is to evaluate the transferability of features between the image and video tasks and to show the hierarchical relation of these two tasks. In all the following sections, we construct our methods starting from the MISL network introduced by Bayar and Stamm [27] to train it with two different learning approaches. This network has proven successful in several multimedia forensics applications [198, 199], so we decided to keep its architecture and optimize it for our setting. Because the MISL network was originally designed to work on greyscale images, we modified the initial constrained layer to work on RGB inputs, therefore, we doubled the number of kernels in the first convolutional layer from 3 to 6, to increase the expressive power of the network and match the more complex input the model is fed with. The network has 5 convolutional layers (called *constrained*, *conv1*, *conv2*, *conv3*, *conv4*) and three fully connected layers (called *fc1*, *fc2*, *fc3*). The *fc3* layer has a number of neurons corresponding to the number of output classes. The network is trained on RGB image patches for the image social media identification platform task, and on RGB I-frame and P-frame patches extracted from videos for the video source platform identification task. Differently from state-of-the-art methods, we decided to use the constrained convolutional layer to automatically learn the best input transformation instead of feeding the network with DCT histograms or reference sensor pattern noise. Therefore, we train the network with RGB input patches extracted from video frames.

In the following sections, we use $\mathcal{I}$ and $\mathcal{V}$ to refer to the image task and video task respectively. Moreover, we use $X_{\mathcal{I}}$ and $X_{\mathcal{V}}$ to refer to the input image or video patches of the network and $Y_{\mathcal{I}}$ and $Y_{\mathcal{V}}$ to refer to the corresponding output classes.

Method Based on Transfer Learning

In this section, we propose transfer learning to transfer the static features learned by a base model on images to the video domain, so as to increase the performance of the same model on this new target task. Because we want the model to learn a certain fingerprint in both image and video-sharing tasks, we adopt this technique to measure how features learned on one of the two tasks, generalize to the other, and study the hierarchical structure of features extracted at different layers of the network.

In this setting, we initially train the model with image RGB inputs $X_{\mathcal{I}}$ to predict the platform of provenance $Y_{\mathcal{I}}$ of these images. The network is initialized with a Xavier initializer [86] and trained on 256×256 input patches to predict the output classes with a cross-entropy loss function. As shown in Figure 3.3, we train this network on native single-compressed images (i.e., images that have not been shared on any platform) and images shared across social networks. Next, we perform feature transfer by freezing a number of layers from the image task and we retrain the remaining network layers on RGB patches $X_{\mathcal{V}}$ extracted from video frames. We iterate this process starting from the lower *constrained* layer up to the higher *fc2* layer of the network. At each iteration, we freeze all the middle layers in between the constrained layer and the upper layer that we want to transfer. Figure 3.3 shows an example of this iterative feature-transfer approach. We initially train the model on the image task in a single-task learning fashion to predict the corresponding platforms of provenance. Then, we freeze all the convolutional layers from the *constrained* layer up to the *conv3* layer and retrain the remaining fully connected layers on the video task to predict the actual new social media platforms. In Chapter 3.2.2, we show that, according to the generic transfer learning behavior, low-level features generalize well across the two tasks, whereas deeper levels tend to learn more task-related representations. This information will be useful in understanding how much the two tasks share with each other.



(**a**) Transfer learning

(**b**) Multitask learning

Figure 3.3: Learning approaches proposed in this study: (**a**) Method based on transfer learning; (**b**) Method based on multitask learning. In the transfer-learning approach we initially train the model on the image task. Then we reuse the feature representations learned on images to train the model on the video source platform identification task. In multitask learning we share the weights of the *constrained* and *conv1* layers of two siamese networks while learning them on images and videos in parallel.

Method Based on Multitask Learning

In multitask learning, we constrain some layers of two models to learn a unique set of parameters for different tasks. In this way, we encourage the shared layers of the network to learn a generalized representation that should help to produce more robust and flexible classifiers with respect to both static and temporal features. As we mentioned previously, the collection of large datasets of shared multimedia content is very hard because of several limitations (mostly related to privacy policies and API restrictions); this approach instead helps to train the network on smaller training datasets. Therefore, in this setting, we force the two networks to share a number of layers to learn more adaptable feature representations.

Figure 3.3 shows the multitask learning-based network used in this study. In the figure, the two proposed networks share the weights from the *constrained* layer up to the *conv1* layer to learn a common feature extractor given input images $X_{\mathcal{I}}$ and videos $X_{\mathcal{V}}$. Next, the two networks independently learn to predict the correct output classes $Y_{\mathcal{I}}$ and $Y_{\mathcal{V}}$. Clearly, as suggested by the hierarchical dependencies of features maps extracted by different layers of the network highlighted by transfer learning, for these tasks it is not helpful to share all the layers from the *constrained* layer up to the *fc2* layer (see Chapter 3.2.2). Thus, to choose which layers to share, we use what we have learned with transfer learning by selecting the layers that extract the more general representations useful for both images and videos, that is the constrained layer and *conv1* layer.

Because detecting forensics traces left by social media on videos is harder than learning such fingerprints on images [10], we train the multitask learner by taking this information into consideration and slowing down the learning process on images. More precisely, we train the model to measure the cross-entropy loss on each task and weighing the overall loss according to the following equation:

$$L = \frac{1}{N}(w_{\mathcal{I}}L_{\mathcal{I}} + w_{\mathcal{V}}L_{\mathcal{V}}) \tag{3.1}$$

where $L_{\mathcal{I}}$ and $L_{\mathcal{V}}$ are the cross-entropy losses on images and videos respectively, N is the number of tasks (2 in our setting), and $w_{\mathcal{I}}$ and $w_{\mathcal{V}}$ are the weights assigned to each task. The weights can be experimentally adjusted on each task depending on the availability of training data and task complexity. In all our experiments, we fix $w_{\mathcal{I}} = 0.25$ and $w_{\mathcal{V}} = 1$ such as to reduce the loss on the image task and accelerate the improvements on videos. As for the method based on transfer learning, at each training iteration the weights and biases of the model are updated according to the gradient descent $w^{(\ell)} = w^{(\ell)} - \alpha\frac{\partial L_t}{\partial w^{(\ell)}}$, where L_t indicates the loss measured on task $t \in \{\mathcal{I}, \mathcal{V}\}$ and $w^{(\ell)}$ represents the weights matrix at layer ℓ.

3.2.2 Experimental Evaluation

In this section, we experimentally evaluate the effectiveness of transfer learning and multitask learning with respect to a baseline single-task learning model fully trained on the target task. Specifically, (1) we measure the performance of two baseline single-task models trained on images and videos; (2) we evaluate the importance of hierarchical features with respect to images and videos, measuring the amount of information that the two tasks share at each level of depth through transfer learning; (3) we compare the results of the multitask-learning approach with those relative to transfer learning and single-task learning.

Dataset and Experimental Setting

We run our experiments on the VISION dataset [263]. In our experiment, we split the dataset for training and validation with a proportion of 80% and 10%, respectively. Moreover, we use the remaining 10% of the dataset for testing purposes. All the results reported in this section refer to this set. This ensures the robustness of the model with respect to completely unseen data. Finally, we use the *ffprobe* (`https://ffmpeg.org/ffprobe.html`—accessed on 4 August 2021) analyzer to extract the I-frames and P-frames from all videos in the dataset and crop each frame and image into non-overlapping patches of size $H \times W$, where $H = W = 256$.

All experiments were carried on a Google Cloud Platform n1-standard-8 instance with 8 vCPUs, 30 GB of memory, and an NVIDIA Tesla K80 GPU. The models have been implemented using Pytorch (`https://pytorch.org/`—accessed on 4 August 2021) v.1.6. For the first two sets of experiments, we trained all the networks with the learning rate set to 1×10^{-4}, a learning rate decay of 0.95 fixed at every epoch, weight decay set to 5×10^{-3}, and AdamW optimizer. In our experiments, we trained the networks for 100 epochs with batches of size 64 and early stopping set to 10. Finally, to train the multitask model, we set a learning rate to 1×10^{-3}, a learning rate decay of 0.99, and weight decay set to 1×10^{-2}. The model was trained for 250 epochs with a batch size of 64. All models were initialized with a Xavier initializer [86].

Evaluation of Single-Task Learning

To measure the effect of transfer learning and multitask learning, we introduce a baseline model trained on each task. We trained the network on images and videos, measuring the model's effectiveness on both tasks. In a single task, we achieved an accuracy of 97.84% for RGB image patches and 86.85% for RGB video patches extracted from frames (see Figure 3.4). Interestingly, we did not observe substantial differences when training the model with both I-frame and P-frame video versus I-frame alone. However, we decided to keep both types of frames to help generalize the model by exposing it to as many different cases as possible. Finally, to validate our choice to train the model on RGB patches without any preprocessing on the input, we compared the performance of our method with the Y-channel of the input after converting RGB to YUV, and we observed a decrease in accuracy of 1.41% for images and 4.2% for videos.

Tables 3.4 and 3.5 report the confusion matrices of the single-task detectors on both tasks. Even though we do not apply any preprocessing operation to the input patches, the model achieves state-of-the-art performance comparable to the much more complex FusionNET [12] for the image task. Indeed, the FusionNET has 98.78%, 98.37%, and 97.13% patch-level accuracy on Facebook, WhatsApp, and native images, respectively, with an average difference of +1.89% with respect to our single-task model. For videos, our method suffers a drop in accuracy compared to the image task, but it still achieves results around 86.85%.

	Facebook	**Whatsapp**	**Native**
Facebook	**98.78%**	0.05%	1.17%
Whatsapp	0.23%	**98.37%**	1.40%
Native	1.56%	1.31%	**97.13%**

Table 3.4: Confusion matrix of the baseline single-task model on patches extracted from images. FBH and FBL represent high-quality and low-quality patches from Facebook. WA and NAT represents WhatsApp and native image patches respectively.

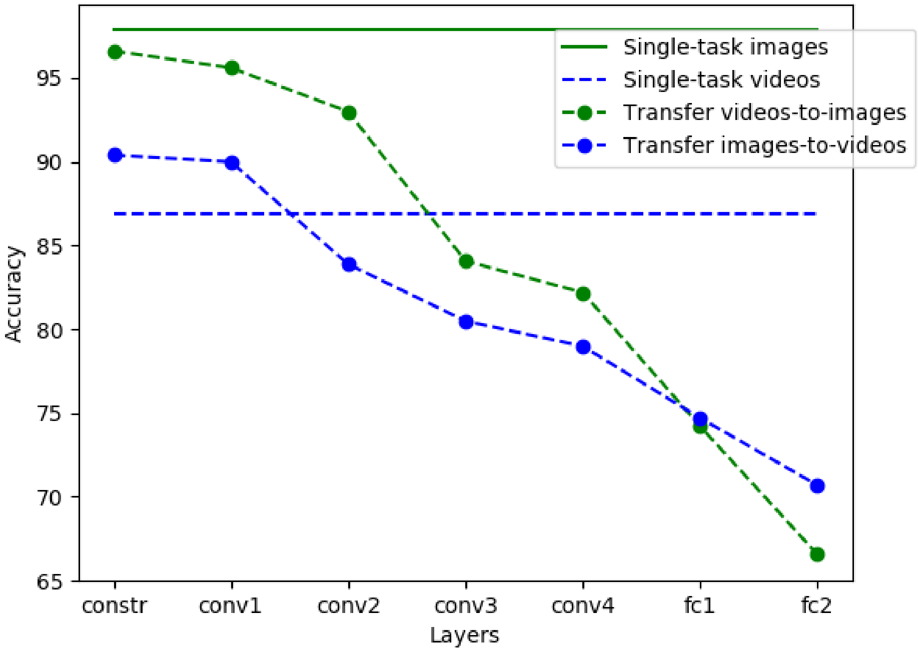


Figure 3.4: Comparison of baseline single-task learning, transfer-learning–based, and multitask-learning–based models accuracy on image (in green) and video (in blue) patches.

	YouTube	**Whatsapp**	**Native**
YouTube	**85.28%**	8.36%	6.45%
Whatsapp	11.56%	**72.35%**	16.09%
Native	2.85%	11.15%	**86.00%**

Table 3.5: Confusion matrix of the baseline single-task model on patches extracted from video frames. YT, WA, and NAT represent YouTube, WhatsApp, and native video patches respectively.

Evaluation of Transfer Learning

We performed a set of experiments to measure the robustness of methods based on transfer learning to images and videos. To perform the experiments, we froze some layers of the network with the learned parameters in one task and we retrained the remaining layers in the other task. To track the hierarchical dependencies of each task and measure the similarity of the two, we repeated this process for each level in the network from the *constrained* layer up to the *fc2* layer. As shown in Figure 3.4, the two tasks share low-level features, whereas deeper representations are mostly related to the target task with the accuracy varying from 66.56% to 96.60% for images and from 70.69% to 90.39% for videos at the patch level. On images (in green), the accuracy deteriorates as more layers are shared from the pretrained *constrained* layer up to the *fc2* layer. When knowledge is transferred from the image domain to the video domain (in blue), the network achieves 90.39% accuracy, gaining 3.54% accuracy with respect to the single-task model. This result confirms the intuition that lower-level features are shared between the two tasks, and that the *hierarchical* dependence between the two tasks can be used to train a deep-learning model on a small set of images or videos originating from social networks. In fact, the features extracted from the deeper levels turn out to be specific to the task being solved and therefore less generalizable, whereas the features extracted from the first levels of the network are more generic and, therefore, can be shared between the two tasks.

The accuracy increases when measuring the performance at the image and at the video level. Specifically, the accuracy on images varies from 80.15% to 97.87%, with maximum accuracy up to 98.37% obtained by transferring video features up to the *conv2* layer. Finally, when transferring from images to video, we can observe an increase in accuracy from 85.48% to 92.61% on the video classifier, but the same does not happen for the transfer from video to images. This result can probably be explained by considering the videos as a more specific case and then thinking of this task as a subset of the corresponding task on images, thus suggesting an *asymmetry* between the two tasks.

Evaluation of Multitask Learning

With this last experiment, we measured the performance of the proposed multitask learner. Specifically, we chose to train two networks on both tasks, by forcing them to share weights between the first two convolutional layers, namely the *constrained* and *conv1* layers. Because of the different complexity of the two tasks highlighted by transfer learning, it is not useful to share all the layers between the two networks and it becomes necessary to balance the learning speed on images with compared to the videos. Therefore, we initially ran several experiments with variable weighted loss according to Equation (3.1). To speed up the training, in this exploratory phase, we chose to train the networks on images and I-frames only for the videos. We report the results of this experiment in Figure 3.5. We have varied the image weight $w_{\mathcal{I}}$ from 0.5 down to 0.1. Then, we chose $w_{\mathcal{I}} = 0.25$ so as to maximize the accuracy of the multitask learner on the video task and we retrained the multitask-learning-based model sharing the *constrained* and *conv1* layers between the two tasks. In this configuration, the multitask-learning-based model achieved 85.91% accuracy on images and 81.70% accuracy on videos. Finally, we tested the overall accuracy of the model at the image and the video level, reaching 92.08% and 91.55% accuracy on the images and the videos respectively. In this setting the model reaches an accuracy comparable to the single-task learner for the video task.

To evaluate the performance of our method, we compared it with the state-of-the-art two-stream network introduced in Chapter 3.1 Amerini et al. [9]. To compare the performance of the transfer-learning and multitask-learning–based methods with that of Amerini et al. [9], we retrained the model of that method in this new setting. Table 3.6 shows the results of this comparison. Splitting the dataset at video level instead of frame level, the method from Amerini et al. [9] records a drop in accuracy of 15.47% compared to the configuration used in the original study.

Method	**Accuracy**
Amerini et al. [9]	80.04%
TL (ours)	**92.61%**
MT (ours)	91.55%

Table 3.6: Comparison of video patch classification accuracy of our transfer-learning and multitask-learning methods with the one of Amerini et al. [9] on the VISION dataset.

3.2.3 Discussion

While the method based on transfer learning achieves a higher overall accuracy than the one based on multitask learning, we investigated the different performances of these two approaches. To analyze and compare the results of the two methods, we kept the best configuration of the multitask

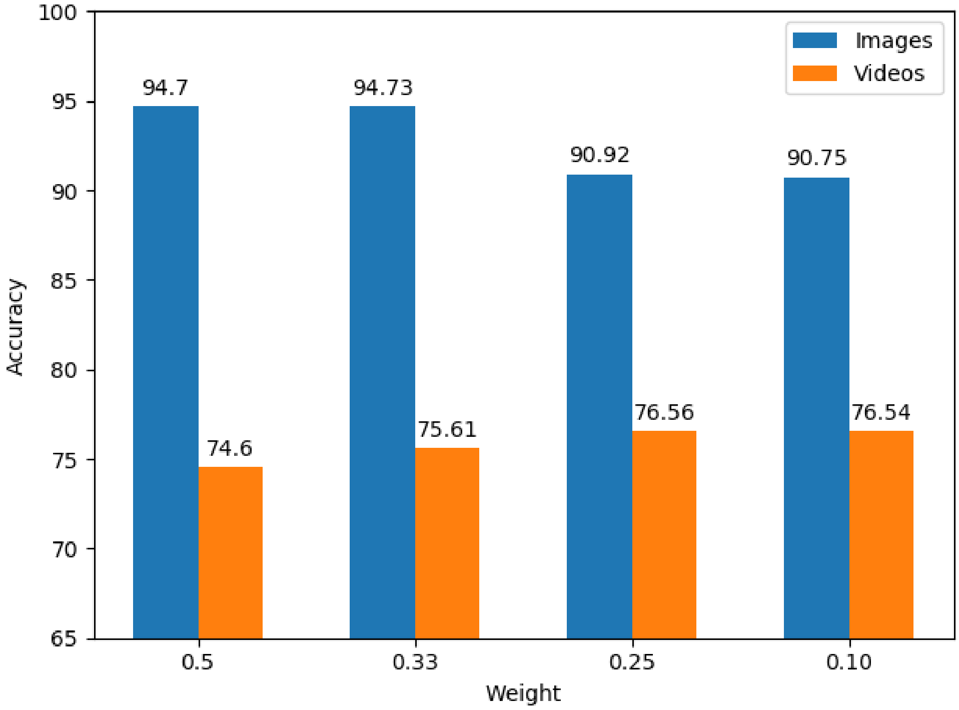


Figure 3.5: Test accuracy of the multitask learner on images and video I-frames (at the image and patch level, respectively) obtained by fixing $w_{\mathcal{V}} = 1$ and varying the weight of the images $w_{\mathcal{I}}$ (x-axis) according to Equation (3.1).

learning-based model and examined the results of the transfer learning-based model when transferring features from the *constrained* and *conv1* layers as for the multitask network. Table 3.7 shows the confusion matrices of these two methods on videos.

	YouTube	Whatsapp	Native
YouTube	**91.24%**	1.08%	7.66%
Whatsapp	13.33%	**69.50%**	17.15%
Native	6.05%	1.49%	**92.45%**

(a) Transfer Learning.

	YouTube	Whatsapp	Native
YouTube	**83.68%**	6.19%	10.04%
Whatsapp	10.04%	**80.24%**	9.72%
Native	10.58%	10.17%	**79.25%**

(b) Multitask Learning.

Table 3.7: Confusion matrices on video patches of the transfer-learning (a) and multitask learning (b) models sharing the *constrained* and *conv1* layers.

First, the transfer-learning model is able to achieve better results than the baseline model on YouTube and native videos (see Tables 3.5 and 3.7a). However, the WhatsApp class gets more easily confused with the other classes. Second, the multitask learner (Table 3.7b) tends to learn feature representations that are more equally separated, with accuracy in all classes that oscillates between 79.25% and 83.68%, making the multitask learner less biased and more robust across all the classes. Moreover, thanks to this property, the multitask approach introduces an improvement in

	Whatsapp	Native
Whatsapp	**60.12%**	39.88%
Native	28.07%	**71.93%**

(a) Single-Task Learning.

	Whatsapp	Native
Whatsapp	**63.08%**	36.92%
Native	23.69%	**76.30%**

(b) Transfer Learning.

	Whatsapp	Native
Whatsapp	**71.48%**	28.52%
Native	26.16%	**73.84%**

(c) Multitask Learning.

Table 3.8: Confusion matrices on video patches of the transfer-learning (a) and multitask learning (b) models sharing the *constrained* and *conv1* layers.

classification performance on WhatsApp compared to transfer learning (+10.74%, see Table 3.7) and the baseline model (+7.89%, see Tables 3.5 and 3.7b). Because WhatsApp is the only class shared by the image and video sets, it might suggest that training a model in a multitask setting on images and videos from the same social media platform could be even more beneficial. To evaluate this intuition we tested the model on WhatsApp with native images and videos, achieving encouraging results. The multitask-learning model achieves higher accuracy than transfer learning and single-task learning, again obtaining more stable accuracy across all classes. Most likely, images and videos shared through the same platform use very similar compression algorithms, favoring the learning of the alterations introduced when the content is recompressed when uploaded to the platform. Table 3.8b,c show the results of this experiment. However, because of the lack of publicly available datasets containing both images and videos we are not able to verify whether this is the case with more classes and leave this issue open for future research.

Chapter 4

Content verification

Online content is an integral part of the digital age, encompassing a vast set of information, entertainment, and communication. However, the rapid proliferation of fake or generated content has become a significant threat, impacting various aspects of society, information sharing, and security. In this chapter, we focus on three macro areas. The most recently generated content is undoubtedly the most studied by the community due to these technologies' enormous impact on society. The extreme realism of the content generated poses new challenges in the fight against disinformation. Next, we move on to the problem of verifying the geographic areas depicted in an image. This problem is significant for verifying information disseminated online or reconstructing events. Finally, we will see how image falsification can impact some industrial sectors, such as insurance, where photos can be modified to defraud or alter evidence.

The incredible realism of artificial intelligence-generated images has attracted interest even outside academia. The spread of these technologies has favored the emergence of many applications in the real world but has also introduced new threats and tools to spread disinformation. The latest deepfakes on the war in Ukraine[1] have clarified the need to develop solutions to detect generated videos. This is just the latest example in a long series of realistic deepfake videos that have hit public figures in the last few years. Deepfakes have become a real threat, and the techniques for generating this content are advancing at an incredible speed. Until now, access to these technologies has mostly remained confined to experts in the field, as generating realistic deepfakes still require in-depth knowledge. For instance, state-of-the-art image generation methods based on GANs [249] must be tuned and adapted to the specific scenario to generate realistic false samples reflecting the desired output. Recently, diffusion models introduced by Ho et al. [106] have further refined the barrier to the entry of these technologies. Based on this methodology, Stable Diffusion [243] and Dall-E [234] are two mainstream applications capable of generating highly realistic images by providing a textual description as input. If, on the one hand, this has contributed to the creation of new startups and the spread of new research fields, on the other, it has opened a new era in disseminating fake content. Current deepfake detectors still have several limitations to overcome. First, they tend to overfit training data, resulting in a performance drop that can be very relevant to new attacks. In addition, the more robust detectors are often difficult to interpret, which poses a reliability problem. Moreover, existing state-of-the-art deepfake detection systems rely on neural network-based classification models, which are known to be vulnerable to adversarial exam-

[1] https://www.bbc.com/news/technology-60780142

ples [119, 212, 277]. In this chapter, we propose four studies conducted on this topic. The first two start from the intuition that, however realistic, deepfakes can contain semantic inconsistencies in the depth of the scene. While difficult to spot at first glance, these inconsistencies can be automatically analyzed using a specially designed model. Consequently, in Chapter 4.1, we analyze the possibility of combining RGB and depth information for more robust detection of these inconsistencies. Next, in Chapter 4.2, we study the best strategy for merging this information to understand how to combine RGB and depth optimally. In Chapter 4.3, we propose an analysis of recent diffusion models to understand how complex it is today to obtain images of human faces that are realistic enough to be challenging to distinguish from real photos, and we propose a first comparison between human and machine performance. Finally, in Chapter 4.4, we further delve into the study of human perception of these contents, demonstrating that in some cases, automatic detectors can be more precise than human ones, but that at the same time, these detectors require specific training and are still very far from human generalization.

Satellite imagery has become an indispensable tool in investigative journalism, providing a means to verify facts, report on conflicts, and sometimes identify those responsible for human rights abuses. They provide journalists with a powerful means of fact-checking, reconstructing events, and shedding light on complex stories. Their use has proven particularly useful in regions where information is tightly controlled or inaccessible. However, matching terrestrial images with satellite images remains a complicated problem. It can become even more complex to use these technologies in the forensic field, where we need to verify and interpret the responses of a model. Chapter 6.1 proposes a study on this topic that offers an interpretable ground-to-aerial matching solution.

Content falsification can be a threat not only to public information but also to specific industry sectors. In Chapter 6.2, we propose a study for the recognition of damage to vehicles in the insurance sector. Image manipulation for insurance claims is a fraudulent practice that involves altering or falsifying images to support a fraudulent insurance claim. This form of insurance fraud can result in significant financial losses for insurance companies and increased premiums for policyholders. The application of forensic techniques in an industrial environment poses numerous challenges, including reducing the number of false alarms or automating a series of steps that are usually performed manually. In Chapter 6.2 we will therefore see how to solve these problems.

4.1 A depth-based strategy for deepfake detection

In this study, we analyze the depth inconsistencies introduced by face manipulation methods. Unlike methods that analyze either imaging pipelines (e.g., PRNU noise [184], specifications [116]), encoding approaches (e.g., JPEG compression patterns [26]), or image fingerprint methods [319], our work analyzes the alteration introduced by the manipulation on RGB and depth features. These spatial features contain semantic information that has the advantage of being more easily interpretable and robust to strong compression operations. With these strengths, *semantic features* can help solve two major challenges with deepfake detection. On the one hand, the lack of explainable detectors, which do not limit themselves to classifying the contents as true or false but allow us to understand what information led to a certain decision. On the other hand, these detectors should be robust to detect fake videos even when some low-level information gets destroyed by compression algorithms. This is particularly important when a video is disseminated on social networks and published several times

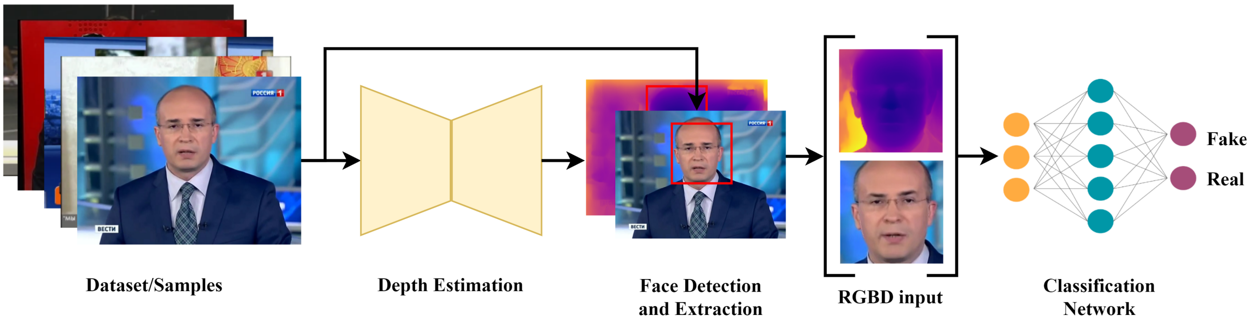


Figure 4.1: Pipeline of the proposed method. In the fist step, we estimate the depth for each frame. Then, we extract the face and crop the frame and depth map around the face. In the last step, we train a classifier on *RGBD* input features.

by different users. In fact, most platforms often reduce the quality and resolution of the video, which can weaken many deepfake detectors. To analyze these semantic features, in this thesis we propose to extract the depth of the face with a monocular-based estimation method that is concatenated to the RGB image. We then train a network to classify each video frame as *real* or *fake*. These two modalities enable the analysis of semantic inconsistencies in each frame by investigating color and spatial irregularities.

To demonstrate the effectiveness of our *DepthFake* method, we conduct extensive experiments on different classes included in the FaceForensics++ [244] dataset. In our experiments, we demonstrate the effectiveness of our method by introducing a vanilla *RGB* baseline and demonstrating that adding depth information allows us to systematically improve detection performance. In summary, the main contributions of this study are threefold as below.

- We analyze the importance of depth features and show that they consistently improve the detection rate. *To the best of our knowledge*, this is the first work that analyses the depth inconsistencies left by the deepfake creation process. Figure 4.2 shows some examples of depth inconsistencies.

- We investigate the contribution of the RGB data and show that a simple RGB-to-grayscale conversion can still lead to acceptable or even higher results in some experiments. We hypothesize that there are semantic features in this conversion that still allow good detection despite the reduction of input channels.

- We conduct preliminary experiments on inference times required by one of the most used convolutional neural networks on several hardware configurations. The increasingly massive adoption of streaming and video conferencing applications brings the need to develop deepfake detection solutions in *real-time*. With this thesis, we propose some experiments to analyze the impact of using multiple channels such as depth or grayscale features on inference times. Our aim is to analyze the impact of our multi-channel model on inference time. These first experiments are a valuable baseline for future developments and studies.

4.1.1 Proposed Method

In this section, we introduce *DepthFake*. Our system is structured in two steps. First, we estimate the depth of the entire image through the FaceDepth model proposed by Khan et al. [136]. This model is pre-trained to estimate the depth of human faces. Next, in the second phase, we extract a 224×224 patch of the subject's face from both the RGB image and depth map. This step allows extracting the face without having to resize the image, as resizing may eliminate precious details useful for classification. Finally, we train a convolutional neural network to classify whether the content is fake or real.

In Chapters 4.1.1 and 4.1.1, we delve into the two modules with further details on the system represented in Figure 4.1, while in Chapter 4.1.1 we discuss about the implementation details.

Depth Estimation

Depth estimation is at the heart of our method. In fact, we hypothesize that the process of generating deepfakes introduces depth distortions on the subject's face. Therefore, the first step in the proposed pipeline is extracting the depth of the face. We can estimate the depth of an image through a monocular depth estimation technique. However, since there are no deepfake datasets containing ground-truth depth information, we propose to use a pre-trained model.

Current deepfakes are usually created with a foreground subject. Therefore, we adopt the FaceDepth [136], a network trained to estimate the distance of the subjects captured by the camera. The model is trained on synthetic and realistic 3D data where the camera is set at a distance of thirty centimeters from the subject and the maximum distance of the background is five meters. This allows us to discriminate facial features by obtaining fine-grained information on the depth of each point of the face. The model has an encoder-decoder structure and consists of a MobileNet [110] network that works as a feature extractor, followed by a series of upsampling layers and a single pointwise layer. The network was trained to receive 480×640 input images and output a 240×320 depth map. The estimated map constitutes one of the four input channels of our deepfake discriminator.

Deepfake Detection

The second module of our system concatenates the estimated depth map to the original RGB image. Since the alterations introduced by deepfakes are usually more significant on the subject's face, we crop 224×224 pixels to extract the face from the rest of the image. The result of this concatenation generates an $RGBD$ tensor $x \in \mathbb{R}^{224 \times 224 \times 4}$, which constitutes the input of our classification network.

In the last step of our method, we train a neural network to classify real and fake video frames. In terms of architecture, we use an Xception [43] network pre-trained on ImageNet [62]. Since we are using a network pre-trained on classical RGB images, i.e. the one used for ImageNet, the addition of the depth channel as fourth input creates the need to adapt and modify the initial structure of the original network to handle this type of data while guaranteeing the correct weights initialization. Therefore, if we randomly initialized an input layer with 4 channels, this would end up heavily affecting the weights learned during the pre-training phase. To solve this problem, we decided to add an additional channel obtained by calculating the average of the three original input channels from the pre-trained model. This change makes the training more stable and allows the model to converge towards the optimum fastly. Consequently, we have chosen to use this initialization method for all the experiments. In addition to this, there is a further problem to be taken into consideration. The values contained in the depth channel range from 0 to 5000, which is the range in

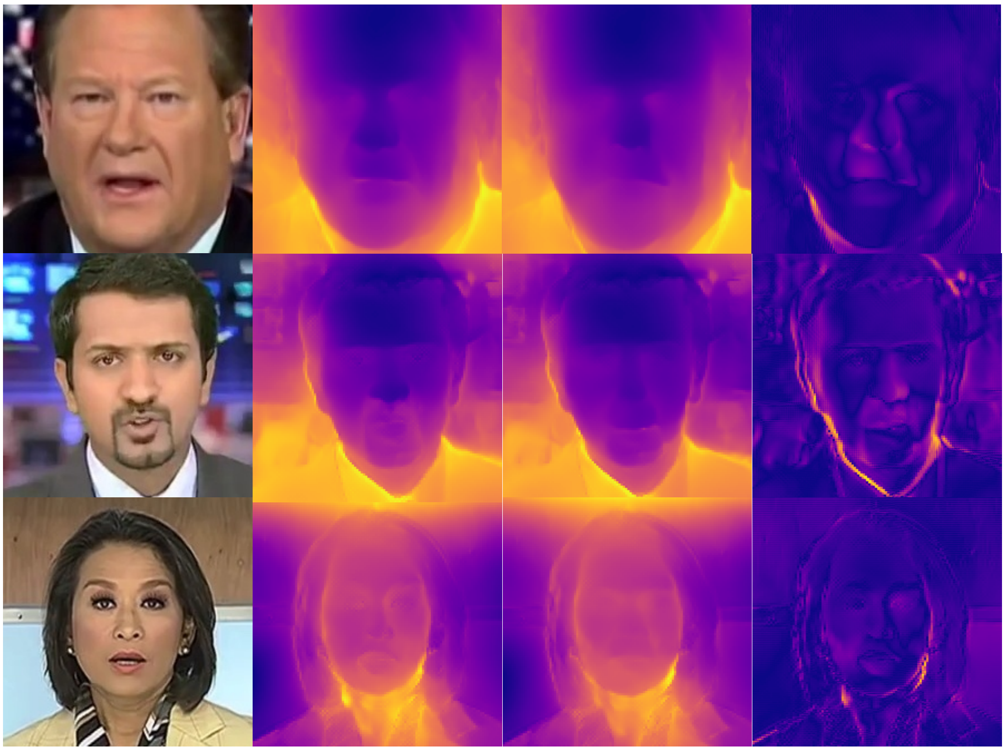


Figure 4.2: Some example inconsistencies introduced in the depth map of manipulated faces. The first column represents the RGB patch of manipulated videos. The second and third columns show the estimated depth maps of the real and fake faces. The last column shows the difference between the real and fake depth maps. Deepfake faces tend to have fewer details than the original ones.

which the depth estimation module has been pre-trained. Linking this channel to the RGB channel without normalizing these values would end up causing numerical instability and would heavily cancel the RGB contribution. To handle this problem we normalize the depth channel values in the range 0–255.

In terms of augmentations, we apply flipping and rotation. While it has been shown that the strongest boost-based on compression and blurring generally improves performance for this task, we decide to keep the augment strategy as simple as possible to avoid altering the information provided by the depth channel.

Added to this, we investigate the contribution of the RGB color model for deepfake detection when paired to the depth information. To this end, we train our system on 2-channel grayscale plus depth input data ($x \in \mathbb{R}^{224 \times 224 \times 2}$). The results reported in Chapter 4.1.2, show that a system trained on depth and grayscale features achieve acceptable or higher results than *RGBD* input data and superior results compared to standard RGB inputs. In this configuration, the network may assign a greater contribution to the depth channel, thus reducing the importance of the information contained in the RGB space. While this is not the goal of this study, it allows us to analyze the impact of a different number of input channels on model inference times, which can be extremely important for real-time applications.

	ResNet50		MobileNet–V1		XceptioNet	
	RGB	**RGBD**	**RGB**	**RGBD**	**RGB**	**RGBD**
DF	93.91%	**94.71**% (+0.8)	95.14%	**95.86**% (+0.72)	97.65%	**97.76**% (+0.11)
F2F	96.42%	**96.58**% (+0.16)	97.07%	**98.44**% (+1.37)	95.82%	**97.41**% (+1.59)
FS	97.14%	**96.95**% (−0.19)	97.12%	**97.87**% (+0.75)	97.84%	**98.80**% (+0.96)
NT	75.81%	**77.42**% (+1.61)	70.47%	**82.26**% (+11.79)	76.87%	**85.09**% (+8.22)
FULL	85.68%	**90.48**% (+4.80)	90.60%	**91.12**% (+0.52)	86.80%	**91.93**% (+5.13)

Table 4.1: Patch level accuracy on Deepfake (DF), Face2Face (F2F), FaceSwap (FS), NeuralTexture (NT), and the full dataset (FULL) with RGB and RGBD inputs. Bold represents the best configuration of each backbone and underlined accuracies represent the best value over each class. In brackets, we indicate the percentage difference added by the depth.

	RGB	**RGBD**
DF	97.25%	**98.85**% +(1.60)
F2F	97.50%	**98.75**% +(2.25)
FS	98.00%	**98.75**% +(0.75)
NT	81.25%	**88.00**% +(6.25)
FULL	87.00%	**93.00**% +(6.00)

Table 4.2: Patch level accuracy of the Xception-based model on video level for Deepfake (DF), Face2Face (F2F), FaceSwap (FS), NeuralTexture (NT) and the full dataset (FULL) with RGB and RGBD inputs. In brackets we indicate the percentage difference added by the depth, while bold represents the best configuration.

Implementation Details

We implement the proposed study using the *TensorFlow*[2] API and train the system on an NVIDIA GTX 1080 with 8GB of memory. In all the experiments we use ADAMAX as an optimizer with the following setup: a starting learning rate equal to 0.01 with 0.1 decay, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e^{-07}$. The training process is performed for 25 epochs with a batch size of 32. We set the input resolution of the architectures equal to 224 × 224 while cropping the original input image around the face. The face detection and extraction is performed with *dlib*[3]. The loss function chosen for the training process is the Binary Crossentropy, a widely employed function for classification tasks; its mathematical formulation is reported in Equation 4.1, where we indicate with $\hat{y}_i$ the predicted sample and with y_i the target one.

$$Loss = -\frac{1}{2}\sum_{i=1}^{2} y_i \cdot log\hat{y}_i + (1 - y_i) \cdot log(1 - \hat{y}_i) \tag{4.1}$$

[2] https://www.tensorflow.org/

[3] http://dlib.net/

Once the training phase has been completed, we compare the inference times between the different models and input channels; we report those values in milliseconds (ms) in Chapter 4.1.2.

4.1.2 Results

In this section, we report the experiments and evaluations that have been conducted. We propose a comparison with some well-established CNN networks and show the first results on the inference times of the model that will be deepened in future studies. We evaluate our model on the FaceForensic++ [244] dataset, which allows us to evaluate the importance of depth features on the most common strategies to create a deepfake. This dataset is composed of 1000 original video sequences that have been manipulated with four face forgeries, namely: Deepfake (DF), Face2Face (F2F), FaceSwap (FS) and NeuralTexture (NT). For all experiments, we split the dataset into a fixed training, validation, and test set, consisting of 90%, 5%, and 5% of the videos, respectively, and report the results on RAW videos.

Deepfake detection

We begin our experiments by analyzing the effectiveness of the proposed solution in identifying deepfakes. We train our system to solve a binary classification task on individual video frames. We evaluated multiple variants of our approach by using different state-of-the-art classification methods. In addition, we show that the classification based on the Xception [43] architecture outperforms all other variants in detecting fakes, as previously demonstrated in other works [90, 244].

	ARM Cortex CPU [fps]	**Nvidia GTX 1080 [fps]**	**Nvidia Titan V [fps]**	**Nvidia RTX 3090 [fps]**	**Giga FLOPS**
Gray	0.497	28.33	25.02	20.41	9.19
GrayD	0.496 (-0.001)	27.91 (-0.42)	25.66 (-0.26)	19.96 (-0.45)	9.20 (+0.01)
RGB	0.499	27.89	26.54	21.27	9.211
RGBD	0.495 (-0.005)	27.94 (-0.05)	23.77 (-2.77)	19.74 (-1.53)	9.218 (+0.007)

Table 4.3: Floating point operations (FLOPS) and average frame per second (FPS) inference frequency over different platforms. For the GrayD and RGBD, we indicate in brackets the difference with respect to the Gray and RGB models respectively.

First, we evaluate the effectiveness of the main backbones that are popular for deepfake detection: ResNet50 [102], MobileNet [110], and XceptionNet [43]. Table 4.1 compares the results of our experiments with all our configurations. As shown in other studies [48, 244], the Xception network achieves the best performance on all backbones. The results show that the depth input channel always improves the model's performance in all configurations. Added to this, it is interesting to note that the MobileNet is slightly inferior to the Xception and outperforms the deeper ResNet50. This is a notable result when considering the goal of reducing inference times for real-time applications. While this is not the main contribution of this study, we still consider it an encouraging result for future developments.

Next, to have a complete overview of the depth contribution, we compare the Xception's performances through the following four setups.

- **RGB**. The baseline on which the different backbones have been trained using only the RGB channels.
- **Gray**. The backbone is trained on grayscale images solely.
- **RGBD**. The model is trained on 4-channel inputs based on the composition of the RGB and depth channels.
- **GrayD**. The configuration is trained on 2-channel inputs composed of grayscale and depth channels.

As shown in Figure 4.3, the results reveal a consistent advantage of the *RGBD* and *GrayD* configurations over other *RGB* and *Gray* ones. In particular, this advantage is more evident in the NeuralTexture class, which is also the most difficult class to recognize among those analyzed. For the *GrayD* configuration, the results are comparable or in some cases even higher than the performance of the model trained on *RGBD* data. These results confirm our initial hypothesis that depth can make a significant contribution to the detection of deepfakes. In the *RGBD* configuration, the model learns to reduce the contribution of the information contained in the RGB channels, while in the *GrayD* configuration, a lot of irrelevant information has already been removed, allowing the model to obtain good results with fewer input channels. This result suggests that depth in this case adds a more relevant contribution to classification than color artifacts. Similar observations can be made by analyzing the results at the video level shown in Table 4.2. In this case, the performances were measured by predicting the most-voted class for each video.

Preliminary studies on inference time

We conclude our study by presenting preliminary results on the inference times of the solution we have introduced. To the best of our knowledge, we are the first to analyze this aspect in detecting deepfakes. The inference time is of fundamental importance when considering the scenarios in which it is useful to detect a fake video in real time. To do this we analyze the impact of using a different number of input channels on our system. Our aim, for this study, is to analyze the inference times of our model to understand if the different configurations we have introduced have an impact on this aspect or not. Specifically, in Table 4.3 we report the floating point operations (FLOPS) and average frame per second (fps) of the Xception-based model on four different hardware platforms. The results suggest that the higher number of channels has a minimal impact on the inference time with an average 0.68 reduction of frame per second. The depth estimation step is not included in these computations; instead, only the facial extraction and deepfake detection stages are measured. As mentioned, at this stage we are only interested in studying the differences introduced by different number of input channels. Additionally, it is worth noting that even if we do not consider depth estimation in our measurements, there are numerous approaches for real-time monocular depth estimation that might be used for this phase [42, 221, 226].

Based on these results, we can draw some considerations that can trace the right path to design a deepfake detector in real time. The first is that models like the Xception tend to be more effective at detecting fakes. This could suggest that the use of a lightweight network with layers inspired by this architecture could allow to obtain lower inference times while maintaining satisfactory performance. The second is that integrating features such as depth can improve the detection of fakes without

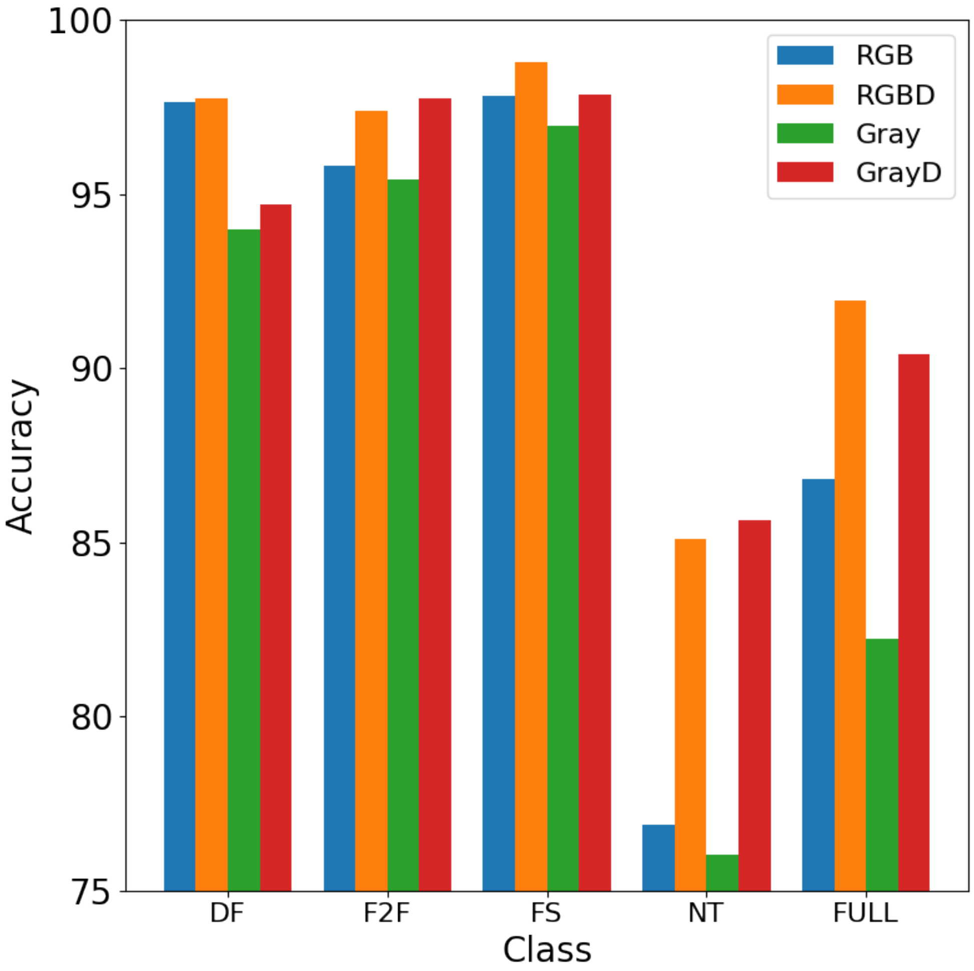


Figure 4.3: Accuracy on Deepfake (DF), Face2Face (F2F), FaceSwap (FS), NeuralTexture (NT) and all classes in the dataset (FULL) with RGB, RGBD, Gray and GrayD inputs.

affecting too much on the frames per second that the model can process. This aspect will be deepened in subsequent works.

4.2 RGB-Depth Integration via Features Fusion

This study builds on the one presented in the previous section proposing substantial improvements in terms of the robustness of the model with respect to adversarial attacks. Depth information provides valuable spatial and semantic cues that can reveal inconsistencies introduced by facial manipulation methods. However, it is unclear to what extent this additional information could contribute to developing a more robust detector than the corresponding methods based on RGB features alone. To deepen this aspect, in this study, we analyze different fusion methods of the RGB and depth channels, and we propose various experiments to understand the best way to integrate this extra information into a detector. Next, we compare the heat maps of the proposed model with a model trained only on RGB features. As shown in Chapter 4.2.3, integrating the depth with RGB helps the model learn more discriminative features concerning the RGB-only counterpart, paving the way for possible developments in model interpretability. Finally, we test the robustness of the model against the most commonly used attacks in deepfakes. In our experiments on the FaceForensics++ [245]

dataset, the model is more resistant to these attacks than its RGB counterpart.

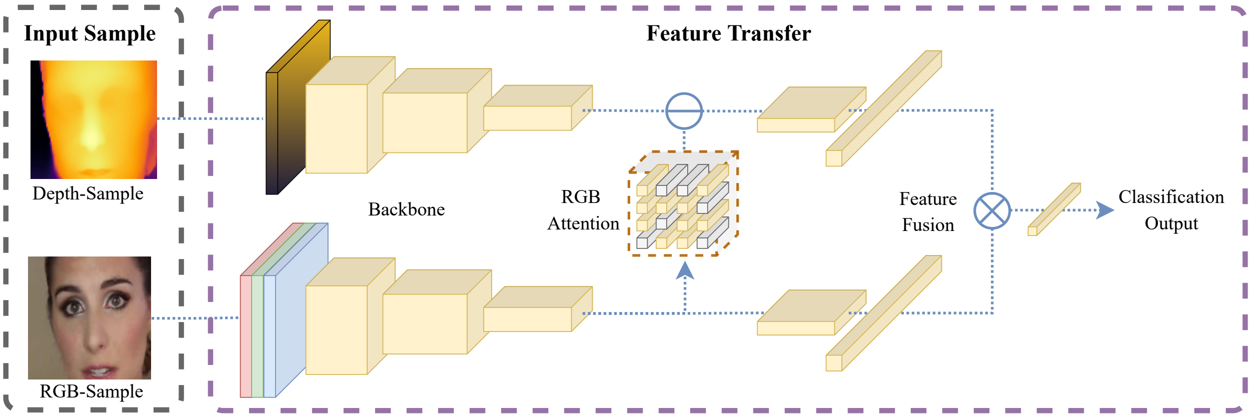


Figure 4.4: Our proposed pipeline. RGB and depth characteristics are analyzed separately by two MobileNet v2. We introduce an attention mechanism that masks the less important depth features based on the RGB features. Finally, we merge the features through a concatenation to proceed to the classification.

4.2.1 Proposed method

Our method is based on the intuition that, as shown in our previous study [191], the deepfake creation process introduces inconsistencies in the depth of the face. Consequently, combining the depth information with the RGB image will improve the learning process and make it more stable and possibly more robust against adversarial attacks. Our goal is, therefore, to understand how RGB and depth information can best be combined to obtain greater accuracy than RGB features on which the networks typically focus. Unlike our previous study, we propose a late fusion mechanism combining the RGB and depth features. Moreover, we propose an attention mechanism that guides the learning of the feature-depth network based on the most important features identified by the RGB one. This allows us to keep the two inputs in two separate streams but, at the same time, direct learning toward a common feature space. The proposed method is composed of the following two steps. (1) First, we extract depth from the whole frame using the pre-trained model introduced by Khan et al. [137]. Since we know that image resize tends to destroy fundamental traces for the recognition of fakes, and that manipulations usually focus on the face and the areas around it, we extract the person's face using a $W \times H$ crop. (2) Then, as shown in Figure 4.4, we input the RGB and depth patches to the deepfake detection model to classify the frame as *true* or *false*.

More details on both steps are provided in the remainder of this section. Specifically, Chapter 4.2.1 explains the preprocessing operations we perform to extract depth and crop the face, and Chapter 4.2.1 describes our method of detecting deepfakes.

Pre-processing

Our method revolves around the depth estimation task. We assume that the deepfake generation process introduces distortions in the depth of the subject's face, which can be crucial for the final classification. For this step, we rely on the method introduced by Maiano et al. [191].

As mentioned above, the proposed pipeline starts with estimating the frame depth. For this purpose, we use FaceDepth [137], a technique for estimating the monocular depth of faces. This network has been specifically trained to calculate the distance of faces from the camera. FaceDepth can detect details of facial features and obtain precise depth information for each facial point. This allows for accurate discrimination of facial features and allows us to estimate the differences between real and fake faces more accurately.

Image resizing can eliminate important information that can help the model in the classification task, so to avoid this kind of problem, we extract a crop centered on the subject's face. We use the Dlib[4] library for face detection and extraction.

Deepfake detection

In this section, we discuss the central part of our contribution. The proposed model takes as input the original RGB patch of size $W \times H \times 3$ and the estimated depth map of size $W \times H \times 1$. Given RGB's larger number of channels, using a single input that concatenates RGB and depth information may not be optimal because RGB information may get more network attention than depth. To avoid this, we have developed an ad hoc architecture, shown in Figure 4.4. The whole architecture consists of two different networks, one for each type of input (RGB and depth).

The proposed architecture, which we call *Masked Depth Network (MDN)*, comprises two parallel networks. The RGB network is designed to process individual RGB frames and extract relevant features for deepfake detection. In contrast, the depth network captures depth-related inconsistencies that are often difficult to eliminate in deepfake videos. The information extracted from the two networks is merged before classification by concatenating the output of the last convolutional layer of both streams. This allows us to integrate the information extracted from the RGB and depth networks while preserving the most discriminating aspects of both channels.

In addition to the fusion phase, we introduce an *attention mechanism* to guide the depth network in selecting the most essential features. This step enforces the fusion process by highlighting regions of interest for deepfake detection. The attention is introduced by masking the weights of the RGB network and integrating this mask into the depth network. Formally, given a weight matrix W, we compute the attention mask $a(w_i)$ for all $w_i \in W$ as follows.

$$a(w_i) = \begin{cases} 0, & \text{if } w_i < 0. \\ 1, & \text{otherwise.} \end{cases} \tag{4.2}$$

In our experiments, we apply this masking operation on the fourth convolutional block of the MobileNet v2 [247] architecture. This attention mechanism allows the depth network to dynamically adjust its attention to focus on the most informative regions indicated by the RGB stream and which are expected to contain critical depth-based cues for distinguishing real video from deepfakes. As shown in Chapter 4.2.3, this architectural choice helps to achieve better performance than simple feature concatenation.

In summary, our proposed architecture's fusion and mask generation steps enable the integration of RGB and depth channels while selectively highlighting informative regions. As we will discuss in Chapter 4.2.3, this approach improves the architecture's overall accuracy and robustness leverages both channels' strengths and facilitates the extraction of relevant features for effective deepfake detection. We incorporate augmentation techniques on the pre-trained model during the training process to address the overfitting problem and improve our architecture's generalization capability.

[4] `http://dlib.net/`

The implementation details are discussed in the next section.

4.2.2 Implementation details

The proposed method has been implemented using PyTorch[5] deep learning API. The trained architectures are initialized on ImageNet pre-trained weights and trained with a CrossEntropy loss function for 30 epochs with a batch size of 192 using Adam optimizer [147] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and learning rate of 0.0001. Moreover, to improve the generalization performances of trained models, data augmentation has been incrementally performed during the training epochs, i.e, by increasing the augmentation effect with the increase of the number of epochs. We examine multiple augmentation strategies and transformations as proposed in [38, 57, 90, 114]. We evaluate our method on the FaceForensics++ dataset [245]. expression manipulation. In our experiments, we report results on individual classes and the entire test set of all four classes. We refer to this last case indicating it as *ALL*. The dataset has different compression levels for each video, namely RAW (uncompressed), C23, and C40.

4.2.3 Results

This section shows the effectiveness of including depth information for deepfake detection versus a standard RGB approach. Precisely, we chose the MobileNet v2 [247] as the backbone in all experiments, which is demonstrated to perform well despite being a lightweight architecture [191]. In addition, to leverage the effectiveness of the proposed attention mechanism included in our Masked Depthfake Network and inspired by fusion strategies discussed by Ophoff et al. [217] and Zhou et al. [338], we also compare the proposed method with different architectural and input configurations. Precisely, to validate the proposed architecture, we introduce four baseline structures where we modify how the RGBD input is provided to the model. Below is a detailed description of each model.

- *RGB*: it consists of a single MobileNet v2 network that is trained on the RGB frames.
- *Depth (D)*: it consists of a single MobileNet v2 network trained only on depth maps.
- *Early fusion (EF)*: in this scenario, the RGB and depth inputs are stacked and passed to the network as a single (4-channel) input as done in the previous section (i.e., Chapter 4.1), which we refer to as Maiano et al. [191]. The addition of the depth channel beside RGB creates the need to use correct weight initialization for pre-trained models. To do this, we average the weights of the first layer of the RGB network model trained on ImageNet.
- *Late fusion (LF)*: it comprises two separate MobileNet v2 networks whose output features are concatenated into a single vector before the classification layers. The combined vector is then passed to the fully connected layers for the final classification phase.

The remainder of this section is organized as follows. We first compare the deepfake recognition performance of the proposed model with the baselines described above. Then, we examine the activation maps of the proposed model against the RGB baseline to see if the addition of depth

[5] Code and corresponding pre-trained weights are made publicly available at the following GitHub repository: https://github.com/gleporoni/rgbd-depthfake

and the proposed attention system lead the model to pay attention to more discriminating features. Finally, we conclude the section by studying the robustness of the proposed method against common *black-box* adversarial attacks for deepfakes.

Detection performance

Our first analysis aims to quantitatively demonstrate the effectiveness of the addition of the depth channel to standard RGB approaches and validate the proposed Masked Depthfake Network architecture with respect to the baselines. Through these experiments, the intent is to understand what is the most effective way to use depth for this task. Table 4.4 shows the overall accuracy obtained at testing time over the different forgeries and compressions levels of the FaceForensic++ dataset. The ROC curves and their respective AUC (area under the curve) values are reported in Figure 4.5, Figure 4.6 and Table 4.5 respectively.

Class	**Testing Set (RAW)**				
	RGB	**D**	**EF**	**LF**	**MDN**
DF	96,00%	89,23%	95,35%	96,59%	**96,86%**
F2F	95,35%	84,75%	95,38%	95,57%	**95,85%**
FS	95,32%	81,23%	95,62%	**96,33%**	96,29%
NT	92,01%	78,77%	92,30%	**92,76%**	92,65%
ALL	95,02%	83,23%	**95,09%**	94,81%	94,87%

Class	**Testing Set (C40)**				
	RGB	**D**	**EF**	**LF**	**MDN**
DF	88,15%	73,46%	88,65%	90,75%	**91,26%**
F2F	**82,57%**	66,39%	82,13%	82,25%	81,82%
FS	86,11%	67,00%	85,45%	86,73%	**87,17%**
NT	70,77%	59,26%	70,77%	**71,00%**	70,50%
ALL	82,37%	79,59%	82,09%	82,25%	**82,43%**

Table 4.4: Quantitative results obtained on deepfake detection task for RAW and C40 dataset settings when trained on Deepfakes (DF), Face2Face (F2F), FaceSwap (FS), NeuralTextures (NT), and all (ALL) forgeries. The best results for each configuration are reported in bold.

The results show that integrating the depth channel into the standard RGB approach guarantees increased detection performance. The MDN and the LF usually perform better than the other baselines for all types of forgeries and compressions. These architectures achieve an average accuracy and AUC boost of up to +1.01% and +0.42% on the RAW dataset and up to +3.11% and +3.86% on the C40 dataset respectively, demonstrating that the depth information combined with RGB one is able to improve the overall detection process. Moreover, we can notice that the RGB network outperforms the D network alone for both RAW and C40 datasets by an average percentage of 12.07% and of 15.55% on the AUC. This is perfectly explained by the fact that, besides having fewer channels, the depth information is estimated starting from the RGB. However, when we combine the two pieces of information, the results confirm the hypothesis that depth information helps the model better discriminate between fake and real examples based on inconsistencies in image depth.

Finally, in Table 4.6 we compare our results against three state-of-the-art methods: the Depth Map-guided Triplet Network [169] (DGN), the Multiple Instance Networks (MIL [299]), and Fakespot-

Class	**Testing Set (RAW)**				
	RGB	**D**	**EF**	**LF**	**MDN**
DF	99,08%	95,73%	99,22%	99,02%	**99,33%**
F2F	99,07%	90,65%	**99,19%**	99,09%	98,88%
FS	99,09%	85,20%	99,19%	99,50%	**99,51%**
NT	97,52%	85,38%	97,72%	**97,91%**	97,90%
ALL	98,29%	78,33%	98,25%	98,37%	**98,50%**

Class	**Testing Set (C40)**				
	RGB	**D**	**EF**	**LF**	**MDN**
DF	93,79%	77,62%	93,79%	96,71%	**96,83**%
F2F	89,51%	68,01%	89,01%	**90,21%**	90,15%
FS	90,48%	69,71%	90,72%	94,17%	**94,34**%
NT	74,32%	58,75%	73,06%	**77,70**%	77,55%
ALL	78,19%	54,50%	78,76%	80,16%	**81,20**%

Table 4.5: AUC values obtained on deepfake detection task for RAW and C40 dataset settings when trained on Deepfakes (DF), Face2Face (F2F), FaceSwap (FS), NeuralTextures (NT), and all (ALL) forgeries. The best results for each configuration are reported in bold.

Class	**Testing Set (RAW)**			
	MDN	DGN	MIL	FKSPT
ALL	**98,50%**	98,30%	97,73%	**98,50%**

Table 4.6: AUC values by our proposed MDN compared to state-of-the-art methods. The best results for each configuration are reported in bold.

ter (FKSPT [294]). Our proposed method achieves the best performance together with FKSPT. These results confirm the contribution introduced by depth compared to other methodologies. In the next section, we delve further into the contribution of depth to identify any limitations.

Feature analysis

We now analyze the activation maps of the proposed Masked Depthfake Network from a qualitative perspective. In particular, with this analysis, we want to understand if the attention mechanism leads the network to focus on more discriminative and, therefore, more interpretable features than the RGB counterpart. Consequently, we calculate and display the activation maps of the last convolutional layer of the Masked Depthfake Network and the RGB baseline using the GradCam [252] method. We report an example of the obtained *Real* and *Fake* output heatmaps in Figure 4.7.

The first row of the figure represents the RGB and corresponding depth inputs extracted from the FaceForensic++ dataset (RAW). The second and third rows report the heatmaps of the RGB and depth models, respectively. The heatmaps show that the RGB model produces more or less uniform activations, which does not give us particular indications on any area of the face. This could suggest that the model may have overfitted the training samples, making it less robust and interpretable. In the case of depth, the activation is most robust in the area around the nose. In the EF model, we notice that this difference between RGB and depth disappears, highlighting the problem of immediately merging the necessary features. The effect of early fusion is to reduce the depth contribution compared to other fusion methods. The advantage of late fusion becomes evident

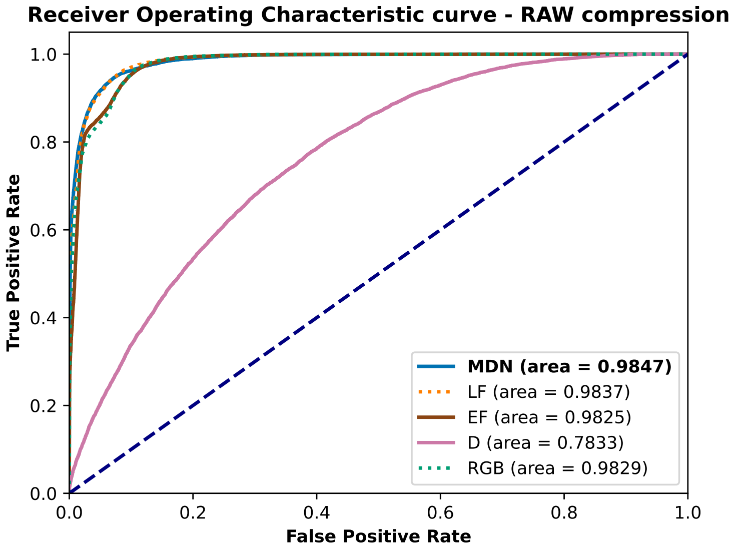


Figure 4.5: ROC Curve results obtained on deepfake detection task for RAW dataset when trained on all (ALL) forgeries.

for the LF and MDN models, where the depth and RGB components have different turn-ons. In particular, we can observe a stronger activation of the MDN in correspondence with the nose and eyes area, which shows how the attention mechanism manages to concentrate the model's attention towards the places most subject to manipulation. This also highlights a possible limitation of this approach: the model's performance is strictly linked to the accuracy of the depth estimation model. However, this problem can be easily mitigated with a more accurate depth estimation method.

Differently, we can notice that the MDN network activates on specific regions of the face. Precisely, the model focuses on the nose region for the authentic RGB image, while for the fake one, the model focuses on the nose and the eyes. Similarly, but with an opposite behavior, we notice that the depth network focuses on particular interest areas of the mouth, nose, and eyes.

Summarizing, we can conclude that although the RGB approach achieves good results in terms of accuracy, it does not pay particular attention to specific regions of the face. Indeed, it also takes into consideration the background areas, and this is why this approach is, in general, less reliable. This could suggest that instead of learning facial features, the network is overfitting the dataset, storing general information only partially related to the characteristics introduced by the generation process. Contrarily, our proposed method, which focuses on the tampered region of the image, leads to improved results in terms of accuracy and makes the whole pipeline more robust, as discussed in the next section.

Robustness to adversarial attacks

To have a complete overview of the proposed method against the RGB baseline, we study its robustness against adversarial attacks. An attacker may decide to introduce imperceptible disturbances in the fake video to bypass deepfake detectors. Specifically, we test the robustness of the models against *black-box* attacks discussed in Gandhi and Jain, and Hussain et al. [83, 117]. Since these attacks include *Guassian blur* (BLR), *Guassian noise* (NSE), *rescaling* (RSC), and *translation* (TRN), which are also used in the data augmentation strategy, for a correct comparison, we use the RGB

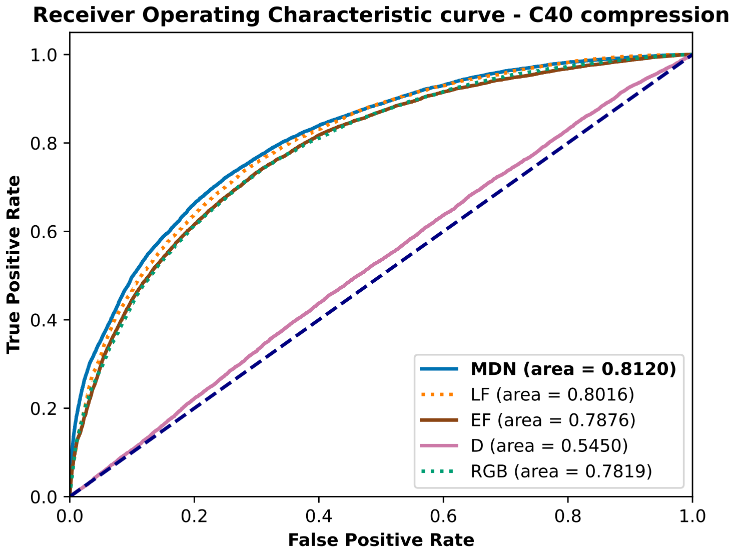


Figure 4.6: ROC Curve results obtained on deepfake detection task for C40 dataset when trained on all (ALL) forgeries.

Attack	**Model**	**Testing Set (RAW)**				
		DF	**F2F**	**FS**	**NT**	**ALL**
BLR	RGB	50,32%	54,30%	50,45%	49,00%	**79,74%**
	MDN	**50,98%**	**70,38%**	**50,62%**	**50,73%**	79,61%
NSE	RGB	85,80%	88,73%	93,83%	76,21%	92,06%
	MDN	**95,69%**	**95,43%**	**95,67%**	**91,61%**	**94,67%**
RSC	RGB	60,63%	60,60%	50,58%	53,89%	74,48%
	MDN	**61,62%**	**75,64%**	**50,62%**	**60,80%**	**78,76%**
TRN	RGB	95,87%	95,11%	95,19%	91,63%	**94,89%**
	MDN	**96,75%**	**95,49%**	**96,23%**	**92,13%**	94,51%
CMB	RGB	**50,31%**	55,22%	50,33%	49,47%	77,95%
	MDN	50,27%	**64,64%**	**50,61%**	**50,46%**	**79,80%**

Table 4.7: Accuracy results obtained on deepfake detection task for RAW dataset settings when Blur (BLR), Noise (NSE), Rescale (RSC), Translation (TRN), and all Combined (CMB) black box attacks are applied. The best results are in bold.

and MDNmethods trained without any data augmentation strategy.

Tables 4.9 and 4.10 report the performance of all the baseline models against every single attack as well as their combination (CMB). Based on the obtained results, we can notice that all the RGBD approaches are able to outperform the standard RGB one in almost all of the experiments. More in detail, in the case of the RAW dataset (see Table 4.9), the MDNachieves an averaged percentage boost of +3.89%, +2.54%, and +0.63% with respect to the RGB, EF, and LF methods respectively. Similarly, in the case of the compressed (C40) dataset, reported in Table 4.10, we can notice that the average improvement achieved by MDNover the RGB, EF, and LF methods is equal to +3.48%, +3.96% and +1.18% respectively.

Based on the reported values, we can conclude that the proposed method could be a viable solution to improve the estimation performances and the robustness against adversarial attacks in the deepfake detection task.

Attack	Model	Testing Set (C40)				
		DF	F2F	FS	NT	ALL
BLR	RGB	66,03%	67,82%	58,02%	**56,00%**	79,27%
	MDN	**75,17%**	**74,60%**	**71,92%**	49,70%	**79,90%**
NSE	RGB	87,20%	81,05%	85,57%	62,00%	81,41%
	MDN	**89,75%**	**81,69%**	**86,86%**	**70,65%**	**82,00%**
RSC	RGB	73,02%	70,19%	64,67%	**57,18%**	80,11%
	MDN	**78,37%**	**74,96%**	**76,59%**	50,46%	**80,38%**
TRN	RGB	87,63%	81,23%	84,83%	**70,13%**	81,07%
	MDN	**89,76%**	**81,46%**	**86,24%**	69,81%	**81,82%**
CMB	RGB	58,73%	66,32%	55,75%	**50,71%**	79,67%
	MDN	**71,96%**	**73,22%**	**65,96%**	49,52%	**79,80%**

Table 4.8: Accuracy results obtained on deepfake detection task for C40 dataset settings when Blur (BLR), Noise (NSE), Rescale (RSC), Translation (TRN), and all Combined (CMB) black box attacks are applied. The best results are in bold.

4.3 From generation to detection of fake content: a comprehensive analysis of Stable Diffusion

Up to now, few works have been presented to determine how difficult it is to distinguish between photorealistic synthetic images generated by diffusion models and real ones [46, 240, 253], while other studies [177, 219, 253] focus on proposing the optimal textual prompts to generate more realistic samples. In this study, we continue to examine the path traced by Wang et al. [295] relying on a different perspective, i.e., in creating realistic human faces based on Stable Diffusion for security purposes. We focus on the two main phases: the *generation*, examining the difficulties that still exist in the design of adequate prompts to produce realistic images and on the *detection*, verifying how complex it is to distinguish between generated and authentic face images. With that, this study wants to photograph all the steps needed for creating content for good or bad applications and understanding how realistic these contents can appear. Although we focus here on the generation of faces, similar considerations can also be extended to other types of images. We leave this door open as a future extension of this study.

With the advances in generative techniques, many efforts have focused on detecting deepfakes and, more recently, images created through diffusion models. Despite the incredible quality of these images, Farid [77] noticed that the lack of explicit 3D modeling of objects and surfaces causes asymmetries in shadows and reflected images. Likewise, global semantic inconsistency can, to some extent, be observed in lighting [78]. Although this may seem encouraging news, the rapid advancement of GANs in recent years teaches us that these semantic limitations will soon be overcome [8]. Since we cannot rely on the model's semantic errors alone, a good part of the state-of-the-art has focused on identifying peculiar traces that are introduced in the generation process. These traces, which we call *fingerprints*, differ from those of modern digital cameras, therefore enabling fake image detection [196]. The fingerprint of an image is commonly estimated by removing the scene content from the image using a denoising filter. Then, so-called noise residuals are averaged across a large number of images to estimate the fingerprint of the generative model. These studies tell us that GANs tend to have sharp peaks in the frequency spectra, implying the presence of quasi-periodic patterns in the synthetic images. Recent findings [46, 253] show that the same happens with some

Attack	Model	Testing Set (C40)				
		DF	F2F	FS	NT	ALL
BLR	RGB	50,32%	54,30%	50,45%	49,00%	79,64%
	D	51,47%	64,84%	**51,69%**	49,39%	**79,75%**
	EF [191]	**53,05%**	51,63%	50,60%	49,58%	77,50%
	LF	50,34%	69,00%	50,47%	**51,92%**	78,07%
	MDN	50,98%	**70,38%**	50,62%	50,73%	79,71%
NSE	RGB	85,80%	88,73%	93,83%	76,21%	92,06%
	D	50,46%	51,88%	60,41%	50,64%	29,03%
	EF [191]	95,32%	95,36%	95,61%	**92,27%**	**95,08%**
	LF	95,58%	94,96%	**95,89%**	92,11%	94,98%
	MDN	**95,69%**	**95,43%**	95,67%	91,61%	94,67%
RSC	RGB	60,63%	60,60%	50,58%	53,89%	74,48%
	D	52,23%	68,36%	**52,44%**	49,46%	**79,70%**
	EF [191]	**67,06%**	56,91%	50,58%	54,60%	70,42%
	LF	56,11%	73,91%	50,65%	**62,17%**	74,48%
	MDN	61,62%	**75,64%**	50,62%	60,80%	78,76%
TRN	RGB	95,87%	95,11%	95,19%	91,63%	94,89%
	D	88,16%	81,75%	75,49%	77,57%	82,52%
	EF [191]	95,19%	95,15%	95,42%	91,80%	**94,92%**
	LF	96,42%	95,22%	**96,29%**	**92,52%**	94,76%
	MDN	**96,75%**	**95,49%**	96,23%	92,13%	94,51%
CMB	RGB	50,31%	55,22%	50,33%	49,47%	77,95%
	D	50,80%	53,74%	**51,05%**	49,45%	79,57%
	EF [191]	**52,44%**	51,75%	50,52%	49,64%	77,83%
	LF	50,11%	62,67%	50,51%	**51,33%**	77,63%
	MDN	50,27%	**64,64%**	50,61%	50,46%	**79,80%**

Table 4.9: Accuracy results obtained on deepfake detection task for RAW dataset settings when Blur (BLR), Noise (NSE), Rescale (RSC), Translation (TRN), and all Combined (CMB) black box attacks are applied. The best results are in bold and the second best are underlined.

recent diffusion models, although these peaks are much weaker for some models, such as DALL-E 2 [218], thus making this analysis more complex. The problem may arise from the fact that little importance is given to high frequencies during training due to the choice of training target, as such frequencies are less critical to the perceived quality of the generated images than the lower ones.

Albeit these early studies have been given encouraging signs, generalization remains the biggest challenge in developing robust fake image detectors, which are very sensitive to the subtle high-frequency traces left by the generation process. Many studies agree on the fundamentally important role of augmentation [90, 195, 295] (especially image blurring and compression), and the heterogeneity of training data to ensure robustness. This helps the models to be robust to the recompression and resizing operations that images often undergo once shared on social networks. In this study, we extend this analysis also to images generated with Stable Diffusion. Finally, these tools must be robust against an ever-changing variety of models and conditions and should be capable of continuously learning from new samples that become available over time [161]. Some initial studies [46, 240, 253] extended this analysis to diffusion models showing that detectors trained only on GAN-generated images work poorly on these new images. Including diffusion models in training can help to detect images generated by similar diffusion models, but results can be unsatisfactory for others.

Attack	Model	Testing Set (C40)				
		DF	F2F	FS	NT	ALL
BLR	RGB	66,03%	67,82%	58,02%	**56,00%**	79,27%
	D	55,27%	60,35%	57,67%	50,22%	79,69%
	EF [191]	**75,94%**	67,00%	55,94%	53,17%	74,34%
	LF	63,29%	70,57%	69,09%	50,27%	**80,00%**
	MDN	75,17%	**74,60%**	**71,92%**	49,70%	79,90%
NSE	RGB	87,20%	81,05%	85,57%	62,00%	81,41%
	D	61,11%	58,82%	61,74%	59,38%	79,55%
	EF [191]	88,63%	82,15%	85,48%	**70,73%**	**82,09%**
	LF	**90,98%**	**82,38%**	86,65%	70,28%	82,02%
	MDN	89,75%	81,69%	**86,86%**	70,65%	82,05%
RSC	RGB	73,02%	70,19%	64,67%	**57,18%**	80,11%
	D	56,12%	60,25%	58,36%	51,11%	79,74%
	EF [191]	68,64%	69,23%	61,28%	55,40%	75,51%
	LF	**80,76%**	71,58%	73,67%	51,41%	**80,44%**
	MDN	78,37%	**74,96%**	**76,59%**	50,46%	80,38%
TRN	RGB	87,63%	81,23%	84,83%	70,13%	81,07%
	D	71,74%	61,76%	64,71%	57,35%	79,65%
	EF [191]	87,50%	80,80%	84,67%	69,64%	81,37%
	LF	89,72%	**81,84%**	86,14%	**70,73%**	81,68%
	MDN	**89,76%**	81,46%	**86,24%**	69,81%	**81,82%**
CMB	RGB	58,73%	66,32%	55,75%	50,01%	79,67%
	D	50,50%	56,15%	54,80%	50,13%	79,73%
	EF [191]	60,45%	63,68%	54,07%	**51,97%**	73,70%
	LF	67,00%	68,70%	64,32%	49,85%	79,59%
	MDN	**71,96%**	**73,22%**	**65,96%**	50,52%	**79,80%**

Table 4.10: Accuracy results obtained on deepfake detection task for C40 dataset settings when Blur (BLR), Noise (NSE), Rescale (RSC), Translation (TRN), and all Combined (CMB) black box attacks are applied. The best results are in bold and the second best are underlined.

In this study, we analyze Stable Diffusion [243] from two points of view: generation and detection. For the first, we investigate how complex it is to generate images of realistic faces and propose strategies that allow obtaining more natural results. Although Stable Diffusion has made it much easier to generate realistic images than in the past, careful engineering of the prompts is still necessary, and sometimes non-trivial, to obtain good results. Figure 4.8 shows some realistic examples we generated through our proposed taxonomy depicted in Figure 4.9.

4.3.1 Prompt Analysis

Designing an accurate prompt is a fundamental step for obtaining photorealistic images generated through a text-to-image model since it allows driving the generated picture in the direction of more realistic images. This practice is commonly known as *prompt engineering* or *prompt analysis* and is commonly based on creating classes of prompt modifiers that control different aspects of the image [220, 225, 303]. However, none of the previous studies focused on photorealistic human faces, which are rich in detail and in distinctive tracts. Therefore, in this section, we focus our analysis on this aspect.

To generate photorealistic samples, we focus on the following three constraints: (**C1**) the image must contain only the *subject's face* as a half-length shot, (**C2**) the image must have a realistic *background* rather than a monochrome one which is typical of professional shooting stage, and (**C3**) the overall image should look *real* meaning it should not resemble a *computer graphics* generated image that typically has unrealistic illumination. Figure 4.10(c) shows an example of a photorealistic image generated through our proposed approach, while Figures 4.10(b) and 4.10(a) show two examples that violate constraints C2 and C3, respectively. Compared to other works, which focus on engineering prompts that produce good images in general but end up generating faces more similar to Figure 4.10(a), our contribution concentrates on creating prompts that constrain the model to generate human faces in the most realistic way possible.

An automatic evaluation of these constraints, however, is a complex computational task and not error-free. Therefore we propose a *human-in-the-loop* approach to evaluate keyword sets. In particular, we follow an iterative method like the one proposed by Pavlichenko et al. [218] in which we input a set of descriptions and a set of *tags* that control the aesthetic appeal to humans. At each step, we increasingly improve the quality of the image's description and add additional tags. Through this iterative method, we found out that using keywords related to photography or photograph websites such as *"50mm"*, *"Nikon"*, *"Pexels"* or *"Unsplash"* produces a significant improvement in terms of realism, generating images comparable to the one in Figure 4.10(b). This, however, is still not enough to obtain sufficiently natural images that respect all the constraints above. To solve this problem, we introduce another set of tags that resemble realistic scenarios like *"realistic background"*, *"shot on iPhone"* and social media platform names like *"Facebook"*, and *"Instagram"*. These tags produce images like the ones reported in Figure 4.10(c) and Figure 4.8. As an example, the image in Figure 4.10(c) can be generated using the following prompt:

"headshot portrait of a young woman, real life, shot on iPhone, realistic background, HD, HDR color, 4k, natural lighting, photography, Facebook, Instagram, Pexels, Flickr, Unsplash, 50mm, 85mm, #wow, AMAZING, epic details, epic, beautiful face, fantastic, cinematic, dramatic lighting".

One of the more severe problems of models trained on large collections of uncurated data scanned from the internet is bias, as much of the internet is dominated by content created by Western people. These biases can very easily be found in Stable Diffusion as well. It is sufficient to generate images using *"man"* as a prompt to see that this will tend to create primarily white Caucasian men's faces with short brown hair. The model can also produce strongly discomforting and disturbing content for a human viewer. This second problem, however, can partially be solved through a Not Safe For Work (NSFW) filter [250] to censor the disturbing images generated from the model. To mitigate this bias and, at the same time, increase the heterogeneity of the outputs, we design an *incremental* process that builds the prompt starting from a minimal description to arrive at a more articulated description of the desired image. Consequently, the basic prompt can be extended by concatenating it with the best modifiers to obtain an ever better and more accurate result. Our prompt construction process can be seen in Figure 4.9.

Following this iterative method, we created a dataset composed of 25800 generated images, 431 of which have been filtered by the NSFW filter. Of this second group of images, 76% represent prompts of women, and in particular, 57% of these are composed of prompts containing *'young women'* in the description. The remaining 24% are prompts that generate images of men. In this second case, only 15% are prompts that create images of *"young man"* and 55% are prompts of *'blonde men'.* It is also interesting to note that no NSFW image is generated when generating images of elderly or black-haired people. In Chapter 4.3.3, we will evaluate some fake image detectors on this dataset.

All the images have been generated using the PyTorch and Stable Diffusion v1.5. We use the CLIP ViT-L/14 backbone and set the number of inference steps equal to 50 and the guidance scale to 7 to guarantee variety in the image outputs with the same prompt. Code, generated faces, and corresponding pre-trained weights are made publicly available at the following GitHub repository: https://github.com/LucaCorvitto/RealFaces_w_StableDiffusion.

4.3.2 Frequency analysis of the generated images

Frequency analysis is often used in forensic applications to visualize artifacts left by the image acquisition pipeline [185] and has similarly been applied to verify the existence of a fingerprint of images generated by GANs [196, 319] and diffusion models [46, 295]. Building on previous studies, in this section, we want to deepen the aspects related to the attenuation of the artifacts visible in spectral images as the compression applied to the image increases. As we have seen in other works [90, 195, 295], compression can significantly erode the performance of the detectors, which can be a massive problem when the images are shared on social media where images are typically subjected to a series of post-processing operations, such as compression. Specifically, we compute a frequency map (denoted as "*Freq*" in Figure 4.11) by mimicking the pipeline used to extract the PRNU pattern for device identification [185], and compare the noise residuals of our generated images with that of real images from the FFHQ dataset. Formally, for each input image x_i, we filter the input with a high-pass filter $f(x_i)$ as in Corvi et al. [46]. Next, the residual R_i is obtained as $R_i = x_i - f(x_i)$, and the desired fingerprint is extracted by applying the Fast Fourier transform (FFT) to the average of the residuals ($\hat{R}$). The complete mathematical formulation is given below.

$$FFT(\hat{R}) = FFT\Big(\frac{1}{N}\sum_{i=1}^{N} x_i - f(x_i)\Big)$$

Figure 4.11 shows the amplitude of these spectra for real and fake images under different compression levels. Each spectra has been generated by averaging 2000 image residuals. A qualitative analysis of the frequency images reveals a marked difference between the fingerprint of the authentic images and that of generated images. In the pictures created from the Stable Diffusion spectrum (*SD* in Figure4.11), we can observe the presence of prominent peaks. On the other hand, a more uniform pattern is visible in the real dataset; this behavior may suggest the presence of quasi-periodic patterns in the generated images. Furthermore, observing the spectrum samples over different compression levels ($C100$, $C90$, and $C70$), we can notice that the presence of artifacts is slightly reduced while the general pattern remains visible. This behavior could indicate that the compression has a minor impact on the performance of the detectors. At the same time, these could be more affected by the resize, which destroys the artifacts much more aggressively. This analysis will be further explored in Chapters 4.3.3 and 4.3.3

Models	**PNG**				**JPEG C100**				**JPEG C90**				**JPEG C70**			
	S1	**S2**	**S3**	**S4**	**S1**	**S2**	**S3**	**S4**	**S1**	**S2**	**S3**	**S4**	**S1**	**S2**	**S3**	**S4**
VGG16	1.0	1.0	0.9994	0.9994	0.9998	1.0	0.9994	0.9994	1.0	0.9998	0.9994	0.9990	0.9992	0.9988	0.9996	0.9984
ResNet50	1.0	0.9988	0.9981	0.9966	0.9996	0.9992	0.9971	0.9960	0.9992	0.9986	0.9977	0.9937	0.9979	0.9964	0.9968	0.9913
Mob.Net	0.9996	0.9998	0.9992	0.9983	0.9994	0.9998	0.9994	0.9983	0.9996	0.9994	0.9996	0.9979	0.9986	0.9984	0.9990	0.9975
Xception	1.0	0.9998	0.9998	0.9990	0.9998	0.9996	1.0	0.9990	0.9996	0.9998	0.9996	0.9981	0.9996	0.9986	0.9996	0.9986
ViT	0.9994	1.0	0.9994	0.9994	0.9977	0.9994	0.9996	0.9986	0.9996	0.9990	0.9998	0.9990	0.9996	0.9996	0.9998	0.9988

Table 4.11: Deepfake detection results: binary classification between real and generated samples on the same configuration scenario.

4.3.3 Detection

Following the previous section, our goal is now to verify how complex it is to recognize these images through some of the best-known deep learning architectures. This section builds upon the analysis introduced in the previous section. Therefore, in Chapter 4.3.3, we present a study on the generated dataset over different compression levels, indicated as $C100$, $C90$, and $C70$. In detail, in C100, the original image (PNG) is converted to JPEG format with the highest quality factor (100%); then $C90$ and $C70$ will have a compression quality equal to 90% and 70%, respectively, using the Python's Pillow library. Next, we move to analyze the generalization performances with respect to new test data, levels of compression, image resolutions, and new generative models. Specifically, in Chapters 4.3.3 and 4.3.3, we focus on model generalization capabilities to analyze the behavior of trained models over different compression levels, resizes, and generative techniques (i.e., StyleGAN [134] and DALL-E 2 [218] generated samples). We perform our analysis on five well-known deep learning architectures: VGG16 [267], ResNet50 [102], MobileNet [111], Xception [44], and ViT [69]. Such models have been trained at an image resolution of 224 × 224 while using a Stochastic Gradient Descent (SGD) optimizer with a momentum equal to 0.9 and an initial learning rate of 0.001. The weights of the selected models are initialized on the ImageNet dataset. We use a batch size of 32 for a total of 100 epochs and the Cross-Entropy Loss as a cost function. Once the training phase is complete, we evaluate the performances of each model as the accuracy value between predicted and true labels. Moreover, due to the different image resolutions used by the compared datasets, the following analysis is performed in four setups. In the first two, we avoid performing an image resize by extracting a central crop (*S1*) or a random crop (*S2*) of the original frame in order to keep all the original information. Differently, in the last two setups, we apply a resize of the entire image (*S3*) or a resized random crop (*S4*) even if some information is flattened. Finally, unless otherwise specified, all the experiments that we report in the following sections are conducted on the set of generated images introduced in Chapter 4.3.1, and authentic images are taken from the FFHQ dataset [135].

Deepfake detection

Our first analysis aims to measure the classification performance of the five models against the different configurations (S1, S2, S3, S4) and compressions ($C100$, $C90$, $C70$) of the dataset. Quantitative results are reported in Table 4.11. From the experiments, we can observe that all models are able to almost perfectly distinguish the distribution of real and fake samples on images generated in lossles PNG format. In particular, VGG16 outperforms the other models, which generally work very well in the S1 and S2 configurations, i.e., when the images are not scaled. On the S3 and S4 configurations, however, the scaling process flattens the image; in these setups, even though all

VGG16	**S1**				**S2**				**S3**				**S4**			
	PNG	**C100**	**C90**	**C70**	**PNG**	**C100**	**C90**	**C70**	**PNG**	**C100**	**C90**	**C70**	**PNG**	**C100**	**C90**	**C70**
PNG	1.0	0.9998	0.9989	0.9942	1.0	1.0	0.9940	0.9842	0.8649	0.8207	0.7760	0.7850	0.9983	0.9904	0.9534	0.9720
*C*100	0.9998	0.9998	0.9985	0.9977	1.0	1.0	0.9964	0.9857	0.7012	0.6734	0.6427	0.6268	0.9992	0.9966	0.9818	0.9945
*C*90	1.0	1.0	1.0	0.9991	1.0	1.0	0.9998	0.9949	0.8239	0.7889	0.7619	0.7871	0.9991	0.9983	0.9974	0.9962
*C*70	0.9992	0.9996	0.9996	0.9992	0.9976	0.9959	0.9968	0.9996	0.6724	0.6501	0.6326	0.6362	0.9944	0.9895	0.9889	0.9979

Table 4.12: Generalization performances of VGG16 model over different compression levels trained on configuration S1.

VGG16	**S1**				**S2**				**S3**				**S4**			
	PNG	**C100**	**C90**	**C70**	**PNG**	**C100**	**C90**	**C70**	**PNG**	**C100**	**C90**	**C70**	**PNG**	**C100**	**C90**	**C70**
S1	0.7365	0.8219	0.9048	0.9341	0.3304	0.2975	0.2195	0.4219	0.3182	0.4853	0.4121	0.6109	0.0634	0.0524	0.0878	0.1134
S2	0.7524	0.8329	0.9085	0.9182	0.3402	0.3207	0.2463	0.4195	0.3670	0.4743	0.4097	0.6682	0.0670	0.0548	0.0878	0.1329
S3	0.9390	0.9634	0.9804	0.9804	0.8012	0.7073	0.6804	0.9219	0.2134	0.2024	0.2487	0.3207	0.1	0.0853	0.1182	0.1317
S4	0.7317	0.8195	0.9048	0.9390	0.3182	0.3146	0.2439	0.4243	0.3439	0.4951	0.4402	0.6536	0.0658	0.0475	0.0804	0.1280

Table 4.13: Generalization performances of VGG16 model over DALL-E2 images in the four considered setups.

models can make accurate predictions, no model can predict all samples perfectly. Moreover, all the models are still able to achieve accurate estimations when trained and tested over the same compression level. However, we can observe a small decrease in the overall performance when the level of compression increases.

Generalization performances over compressions

In this subsection, we go one step further than the ideal situation reported in the previous subsection. In particular, we focus on generalization performance when the model is trained on one compression level and tested on another compression. Since, in the previous subsection, VGG16 performed best in all configurations, we use this model as a reference for subsequent analyses. In each column of Table 4.12, the model was trained on a specific setup and compression level and tested on all compressions using face-centered non-resized images (i.e., the S1 setup). The obtained values show that the models trained on uncompressed images (PNG) have more robust classification performances compared to others. Therefore, we can observe that the larger the area where scaling is applied (e.g., the S3 setting), the worse the generalization performance. Conversely, in the S4 configuration, scaling is applied to a randomly cropped portion of the original image, thus excluding a large amount of information needed for proper classification.

Generalization performances over different datasets

We conclude the generalization analysis by measuring the detectors' robustness with respect to images generated with other models and setups. Specifically, we report two studies: the first is based on the analysis of the blind classification performance of models pre-trained on Stable Diffusion and tested on images generated with DALL-E2. In contrast, in the second, we train the model on real images and fake ones generated by three different GAN models and Stable Diffusion to determine the presence of a particular fingerprint that allows distinguishing the images generated with Stable Diffusion from others.

The first analysis is reported in Table 4.13, where the columns represent the training configurations and the rows are the tested ones. The results show that the S1 configuration achieves the best overall results. In general, it can be seen that the models trained on areas with multiple facial features (setups S1 and S3) generalized better even when the scaling operation is applied compared to those trained on random regions (S2, S4). Moreover, we can observe that in most cases, the greater the compression level on which the models are trained, the better the generalization performance on the original samples. This behavior is likely owing to the more remarkable ability to identify minor artifacts in images with less information.

For the second experiment, we select 1000 faces per class by expanding the real/fake dataset with images generated by three different GAN models (StyleGAN, StyleGAN2, and StyleGAN3). The results shown in Figure 4.12 suggest that models working on non-resized images (setups S1 and S2) achieve high estimation accuracy ranging between from 95% to 99% over the compared models. On the other hand, we can notice that pre-processing operations used in S3 and S4 configurations flatten important image artifacts, resulting in less robust estimation accuracy ranging between 69% to 93%.

4.3.4 Human performances

In parallel with the development of sufficiently robust automatic detection models, some studies [152, 156] have compared the performance of these models with that of humans. In the wake of these works, we measure the human performances on the images generated with the proposed methodology. To do this, we showed a selection of 20 images (10 real and 10 synthetic images from Stable Diffusion), 16 of which are reported in Figure 4.8, to a sample of more than 600 people. The results show that in Figure 4.13 a large part of the representative is able to correctly recognize the 60% of the images with respect to the 99% achieved by a specialized deep learning model. Interestingly, the sample made slightly more errors on the real images than on the fake ones. In our opinion, this indicates that when asked to identify fake faces, many become more critical of real faces as well. However, we can conclude that this task remains complex for non-expert humans, which brings out the need for solutions to contrast the spread of disinformation. We will deepen this preliminary analysis in future works.

4.4 Human vs machine: a comparative analysis in detecting AI-generated images

Generative Artificial Intelligence (AI) dominates the main emerging technologies of 2023. The enormous advances in this field have gained the general public's interest with mainstream tools such as ChatGPT[6] or Midjourney[7]. The results of these tools are simply astonishing, opening the door to the adoption of AI in numerous new sectors. In parallel with all this, however, the interest of a large part of society is growing in the ethical and social impact that these technologies will have on our lives. In addition to the creative and industrial uses, the problem arises of understanding how to distinguish real from generated content. In this regard, despite the generally shared concerns, we still know little about how humans perceive the difference between real and artificial content. Knowing this boundary and monitoring it is essential to understand how far we are still from generating content that is indistinguishable from real.

The main aim of this study is to investigate the limits of humans and artificial intelligence in the recognition of fake images. In fact, we propose, on the one hand, an analysis of the *human perception*

[6] `https://openai.com/chatgpt`

[7] `https://www.midjourney.com/`

of artificially generated images with some of the most recent generative techniques. On the other hand, we compare these results with some *automatic detection* models. Our results demonstrate that the interaction between facial sympathy and type of photo can lead one to believe the photo is real. Moreover, we noticed that the accuracy in recognizing fake faces can be lower for some ethnicities. Finally, the sex and age of the portrayed subject also seem to impact human performance. From a machine perspective, we propose an analysis of some automatic detectors. In particular, we introduce an architecture called ResFormer that combines the benefits of convolutional networks with transformers and propose a comparison between this network and other commonly used baselines. Our experiments confirm the results reported in previous studies. Although the performance of AI systems is superior to human performance when the model is trained and tested on images generated with the same family of techniques, the generalization of these models is a problem yet to be solved.

Our intent with this study is not to pit humans against machines to decide a winner but rather to study both limits to understand how to create more robust and human-friendly deepfake detectors. As we will show later, both have limitations that can give us some suggestions on how to design automatic detectors that are more robust and more applicable in real contexts.

The remainder of this section is organized as follows. The following section provides an overview of the state of the art. We then present this study's methodology for assessing human and AI performance. Next, we introduce a new dataset and describe the design guidelines we adopted to create it. We then show the results of the experiments and discuss the difference between human and AI performance.

4.4.1 Method

Our goal for this study is to compare the performance of an automatic deep learning-based detector with human performance. The first part introduces the methodology we followed to measure human perception of fake images. Next, we present a hybrid architecture based on convolutional layers and transformers for deepfake detection.

Human perception

To evaluate human performance, we recruited 120 online participants (63 females, age: $M = 26.03$; $SD = 8.13$). For the experiments, we used 24 deepfake photos generated through the method from Papa et al. [222] and 24 real photos from the FFHQ[8] dataset. Photos were matched for gender, age, and ethnicity of the faces. After signing the informed consent and giving socio-demographic information, participants were randomly presented one at a time with 48 photos with the request to indicate if they were *real* or *fake* (presentation lasted until a response was given). After pressing the corresponding key on the keyboard, participants were asked to rate their confidence as well as the likeability of the face in the photo, both ratings on a scale from 0 (not at all) to 6 (very much), as shown in Figure 4.14. The whole procedure was constructed on PsychoPy[9] and then carried out on Pavlovia[10].

[8] `https://github.com/NVlabs/ffhq-dataset`

[9] `https://www.psychopy.org/`

[10] `https://pavlovia.org/`

Next, we performed a statistical analysis aimed to verify if our predictors could significantly determine the levels of the dependent variable (i.e., Accuracy). To do this, we chose to apply generalized linear mixed models. These statistical models allow us to take into account both fixed and random effects terms, controlling for the variability of both participants and stimuli. The accuracy of each response was computed by assigning a score to each correct (1) or incorrect answer (0). For all participants, we test if the predicted class depends on the type of photo and other predicted variables. Specifically, seven different models were run, each of them including one of the following predictive variables: confidence responses, likeability responses, age and gender of the participants, ethnicity, age, and gender of the stimuli. Formally, we use the following regression model: Accuracy $\sim$ (C $*$ V), where $\sim$ indicates the regression model, C indicates the type of the photo (i.e., real or fake) and V represents the predictive variables.

The binomial generalized linear mixed models were implemented with the *glmer()* function from the *lme4* package (R[11] version 4.1.3), with subjects and items included as random factors in all models. All post hoc tests were performed with the *emmeans()* function from the *emmeans* package, which is used to obtain the estimated marginal means of each model.

Automatic detection

CNNs have proven effective in detecting AI-generated content; however, these architectures have inherent disadvantages. CNNs operate on fixed-sized local receptive fields, which restricts their ability to understand global contexts effectively. Differently, the transformers introduced the *self-attention* mechanism, allowing models to weigh the importance of different parts of the input sequence when making predictions. Similar to how the human brain can focus attention on specific aspects of our environment selectively, the self-attention mechanism allows transformers to capture the global context of the input sequence and long-range dependencies.

Despite the superiority of transformers over CNNs in several tasks, their effectiveness was not as good in forensic studies, where the limited availability of data leads the models to overfit. To overcome this problem, we propose a hybrid architecture called ResFormer. The hybrid model (depicted in Figure 4.15) consists of two main parts. The first is the convolutional part for extracting spatial relationships. This component extracts feature maps that effectively capture all the essential parts in the images. After that, we turn these feature maps into patches that we feed into a transformer model which is expected to spot connections and context across the entire feature map. The transformer structure is based on a *multi-head self-attention* (MSA) mechanism, which is composed of several single self-attention layers running in parallel. Formally, given an input feature map, the transformer layer first computes three matrices: the query Q, the key K, and the value V, of sizes $d_q = d_k = d_v$. Then, we use the softmax dot-product self-attention operation introduced by Vaswani et al. [284], which is defined as follows.

$$Att(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4.3}$$

The multi-head attention layer allows the model to attend to information from different representation subspaces at different positions and operates by concatenating several attention heads.

[11] `https://www.r-project.org/`

Formally:

$$MSA(Q, K, V) = Concat\,(Att_1, \ldots, Att_h)\, W^O \tag{4.4}$$

where $Att_i = Att(QW_i^Q, KW_i^K, VW_i^V)$, $W_i^Q \in \mathcal{R}^{d_{model} \times d_k}$, $W_i^K \in \mathcal{R}^{d_{model} \times d_k}$ and $W_i^V \in \mathcal{R}^{d_{model} \times d_v}$.

The output of the network classifies the images as *real* or *fake*.

4.4.2 Prompt analysis

Despite the astonishing results obtained from the most recent text-to-image generative models, the choice of one textual prompt over another still makes a difference concerning the images' quality. Our goal is to position ourselves in a setting where images are presented most lifelike, enabling us to assess human capabilities in perceiving AI-generated images. To do this, we rely on the prompt engineering strategy proposed in Papa et al. [222] but propose further improvement. Although their prompt engineering method has extremely high quality, we can note that it tends to produce images with a specific, darker color palette. To solve this problem, we aimed to broaden our images' color palette by making them more heterogeneous and realistic. Figure 4.16 shows an example of the newly generated images and compares them with the ones from Papa et al. [222].

Unlike their method, which used Stable Diffusion v1.5, our solution is based on the Attend-and-Excite[12] pipeline, which allows us to guide image generation with greater flexibility thanks to the ability to specify *negative prompts*. In fact, in the image generation phase, we noticed that the model often tends to generate faces with deformed eyes and teeth, as shown in Figure 4.17. Often, these areas are the most obvious indicator of artificiality. To solve this problem, we realized that the specific words we used in negative suggestions had a big impact. Experimentally, we have noticed that applying the following negative prompt to all the generated images can obtain very good-quality results as shown in Figure 4.16.

"disfigured, ugly, bad, immature, cartoon, anime, 3d, painting, b&w.".

These negative prompts were carefully curated to discourage the generation of unrealistic and disfigured images. Following the prompt generation procedure proposed by Papa et al. and the negative prompts explained above, we generated a dataset of $10,000$ images. For all images, we set the guidance scale parameter to 7, which encourages the model to generate images closely linked to the text prompt.

Compared to Papa et al., our images are more realistic in the details of the eyes (see the first row in Figure 4.16), the mouth, and especially the teeth (see the last row in Figure 4.16) on the wrinkles in Papa et al. were excessively marked, and in the backgrounds, which in our case are very realistic (see the first two row of Figure 4.16).

4.4.3 Experiments

In this section, we report the results we obtained with humans and AI detectors.

Human detection

Table 4.14 reports the descriptive statistics on human participants' accuracy and confidence level. We evaluate the confidence interval of our measurements through t-distributions. We found no sta-

[12] `https://huggingface.co/docs/diffusers/api/pipelines/attend_and_excite`

tistical difference between accuracy on deepfake and real photos ($t(119) = -.958, p = .339$), where $t(n-1)$ indicates the equivalent to the number of standard deviations away from the mean of the t-distribution, n is the size of the population ($n = 120$ for this study) and p indicates the probability to obtain the hypothesized results. Conversely, confidence was significantly higher for deepfake compared to real photos (paired t-test: $t(119) = 2.709,\ p < .01$), which is confirmed from the *glmm* analysis. We validate this hypothesis through Pearson's chi-squared test, which determines whether there is a statistically significant difference between the expected frequencies and the observed frequencies in one or more categories of a contingency table. The analysis revealed a significant interaction between confidence and type of photo ($\chi^2(1) = 5.65,\ p < .05$) on identification accuracy, where $\chi^2(1)$ represents the chi-square distribution with one degree of freedom. Lower confidence was associated with lower accuracy for fake faces (see Figure 4.18(A)). Interestingly, the significant interaction between face likeability and type of photo ($\chi^2(1) = 204.37, p < .001$) on accuracy rates indicates that face likeability makes people believe that faces are real (see Figure 4.18(B)). In regard to the participants' characteristics, age was found to be a significant predictor of accuracy for real photos' ($\chi^2(1) = 3.89, p < .05$) but not fake photos' ($p = .91$), with higher age associated with lower accuracy of real photos. Conversely, the participants' gender was not a significant predictor of accuracy rates ($p = .06$).

Class	**Accuracy**	**Confidence**
Fake	0.694 (0.175)	4.42 (0.929)
Real	0.718 (0.158)	4.30 (0.934)
Total	0.708 (0.456)	4.36 (1.43)

Table 4.14: Descriptive statistics of human participants.

Regarding the photo characteristics impacting human detection accuracy, there was a significant interaction between type of photo and ethnicity ($\chi^2(1) = 13.24, p < .01$) with post hoc analysis confirming a significant difference in accuracy between fake and real for Caucasian ($\beta = -0.929, SE = 0.420, z = -2.211, p < 0.05$, where β indicates the estimated coefficient of the regression model, SE indicates the standard error and z represents the index of the ratio of the estimated coefficient to its standard error), and South American faces ($\beta = 1.548, SE = 0.565, z = 2.739, p < 0.01$), as illustrated in Figure 4.19(a). Furthermore, the age of faces (young vs. old) impacted detection accuracy for both real ($\beta = -0.539, SE = 0.249, z = -2.164, p < .05$) and fake faces ($\beta = 1.492, SE = 0.342, z = 4.369, p < .0001$), as illustrated in Figure 4.19(b). Finally, the gender of the faces (male vs. female) impacted detection accuracy, with fake male faces being less easily identified ($\beta = 0.755, SE = 0.317, z = 2.378, p < .05$), as illustrated in Figure 4.19(c).

AI-detection

We compare the performance of the AI model proposed in the previous section against state-of-the-art models (i.e., Resnet18, Resnet50, and Vision Transformers) on two different datasets: our proposed new dataset, which we call Diffusion Model Human Detection Dataset (DMHD), and a modified version of the CDDB [162] dataset which we refer to as CDDB-s. We chose these two datasets for a specific reason. The first was explicitly generated to have highly realistic images that were difficult for a human to recognize as fake. The second, much more heterogeneous in terms of models used to generate the images, is more complex for an automatic detection model that must learn to generalize concerning different generative techniques. The DMHD dataset is composed of 10,000 fake images generated as explained in the previous section, 10,000 fake images from Papa et al., and 20,000 real images taken from FFHQ [222]. We use an 80-10-10 split for training, validation, and testing. For the CDDB-s dataset, we select a subset of fake classes containing human faces. Specifically, we use ProGAN, StyleGAN, BigGAN, and CycleGAN from CDDB, adding Stable Diffusion from Papa et al. for training. For testing, we use NewFake, Glow, and StarGAN from CDDB and the Attend-and-Excite generated images proposed in this paper. Due to the way it is constructed, this dataset is much more complex than the first concerning the models used for generation, and it is essential to underline that the generative models used in testing are different from those used in training.

Model	**DMHD**	**CDDB-s**
ViT	0.97	0.58
Resnet18	<u>0.98</u>	**0.65**
Resnet50	0.97	0.55
ResFormer (ours)	**0.99**	<u>0.62</u>

Table 4.15: Performance in terms of accuracy of state-of-the-art neural networks.

Table 4.15 reports the performance of all models on both datasets. Our proposed model achieves the best performance on the DMHD dataset and the runner-up for the CDDB-s dataset. In general, we can see that on our dataset, which is very difficult for humans, the models' performances are, on average, high. This suggests that if appropriately trained on very realistic data, the models can learn characteristics that are less visible to us as humans. However, in a more complex and heterogeneous scenario such as that of CDDB-s, the models' performances are much lower, suggesting that the information learned during training is probably too specific and not very generalizable compared to other classes never seen in training. To confirm this assumption, in Table 4.16, we report the results in the more complex scenario. In the first column, we measure the performance of models trained on CDDB-s and tested on DMHD, while the second column reports the opposite scenario. We can see that performance on DMHD is generally lower. In particular, the deeper models (Resnet50 and ViT) record a more marked drop in accuracy, while ResFormer and Resnet18 appear to be more robust. On CDDB-s the results are slightly improved compared to the previous scenario. We find this result particularly surprising and believe it may depend on the fact that the training images on DMHD are numerically more than those seen when we train on CDDB-s, which seems to help the model generalize. We will further investigate this phenomenon in future work.

Model	**DMHD**	**CDDB-s**
ViT	0.83	0.66
Resnet18	**0.94**	**0.68**
Resnet50	0.81	0.67
ResFormer (ours)	**0.94**	0.65

Table 4.16: Generalization performance. First column: training on CDDB-s and testing on DMHD. Second column: training on DMHD and testing on CDDB-s.

Generally speaking, the results seem to align with what has been seen in the state of the art. When trained on identical data distributions, models perform even better than humans. However, generalization to new distributions (i.e., new generative models) remains an open issue. In the next section, we compare human performance with automatic performance.

4.4.4 Discussion

Table 4.17 shows the performance of ResFormer when trained on DMHD. Comparing these results with those in Table 4.14, it is clear that an AI model can outperform humans in accuracy. However, from the results presented in the previous section, it must be highlighted that AI has significant limitations. Generalization is still far behind that of humans. In fact, it must be considered that none of the human subjects involved in this study had any specific training for recognizing false images. Recorded performances depend solely on their experience. This allows us to understand the impact of fake content on the average population. In this regard, it will be interesting in the future to understand whether a change in performance can be observed when people are trained to recognize fakes.

Class	**Accuracy**	**Confidence**
Fake	0.994	4.99 (0.001)
Real	0.9976	4.99 (0.007)
Total	0.996	4.99 (0.009)

Table 4.17: Descriptive statistics of ResFormer trained on DMHD and tested over images employed in the human perception experiment.

Figure 4.19, compares the model's performance across different age groups, ethnic groups, and gender. As we can see, the model reaches 100% accuracy on the fake images taken into consideration, with some errors for the real pictures. Given the results, we do not notice any beneficial indications. This suggests that the model learns the specific features of the dataset almost perfectly and indicates that the nature of these features may be only marginally semantic. This would be in line with other works analyzing the differences between real and fake images in the frequency spectrum. However, this highlights another limitation of automatic detection models. As effective as they are, they are difficult to interpret, and justifying the predictions of these models simply on the basis of their accuracy is reductive and not very feasible in a real scenario.

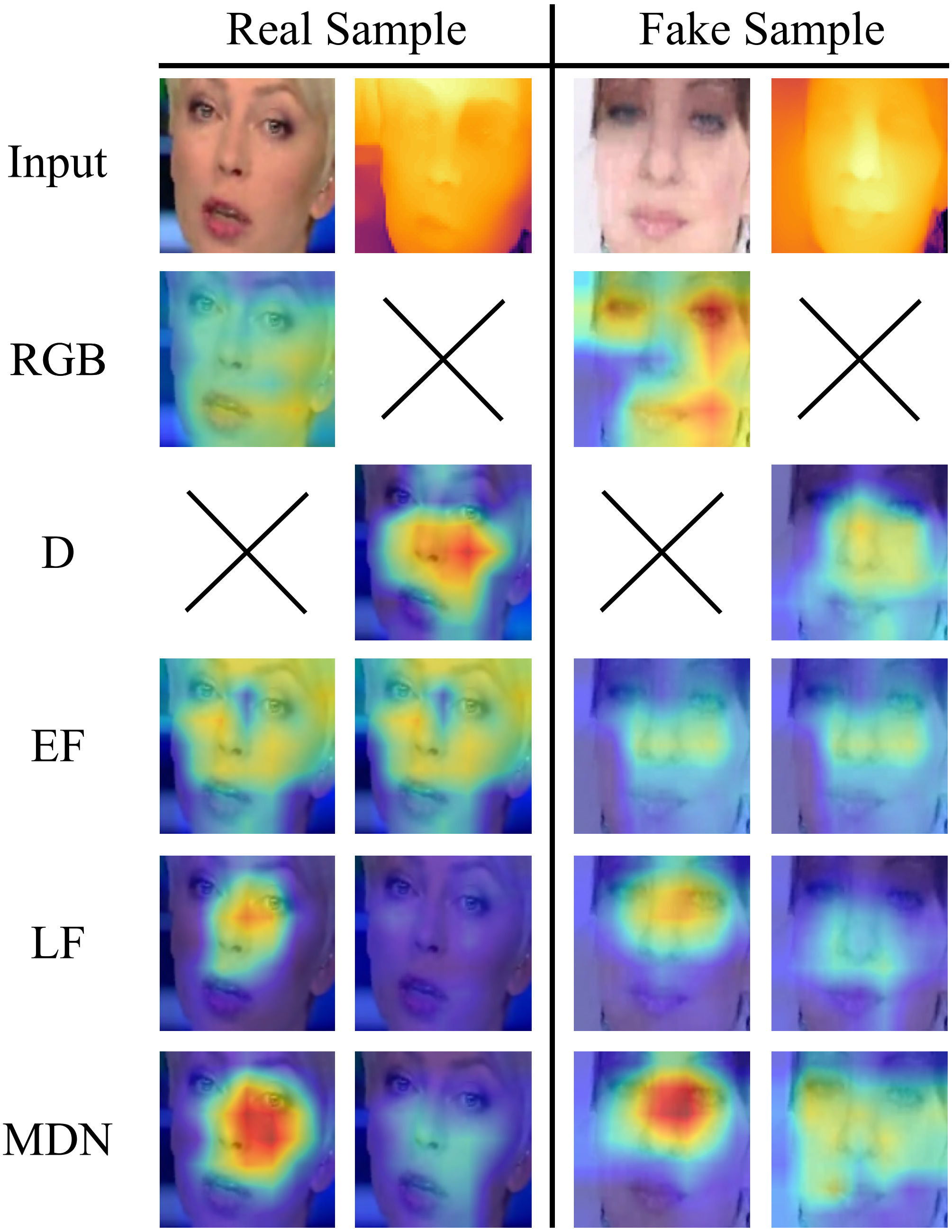


Figure 4.7: Heatmaps generated by the GradCam algorithm. The CAM RGB shows the obtained heatmaps for the RGB baseline model, while CAM MDN for the proposed method.

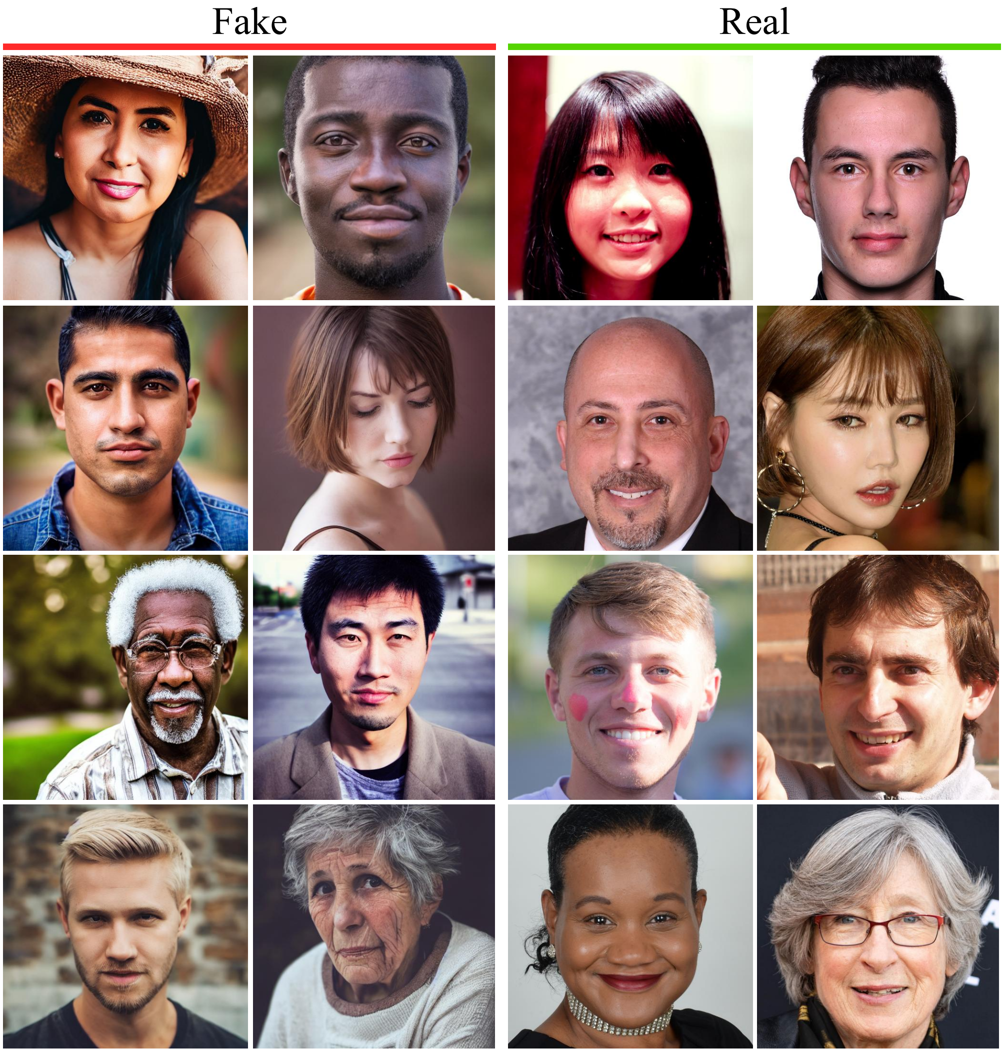
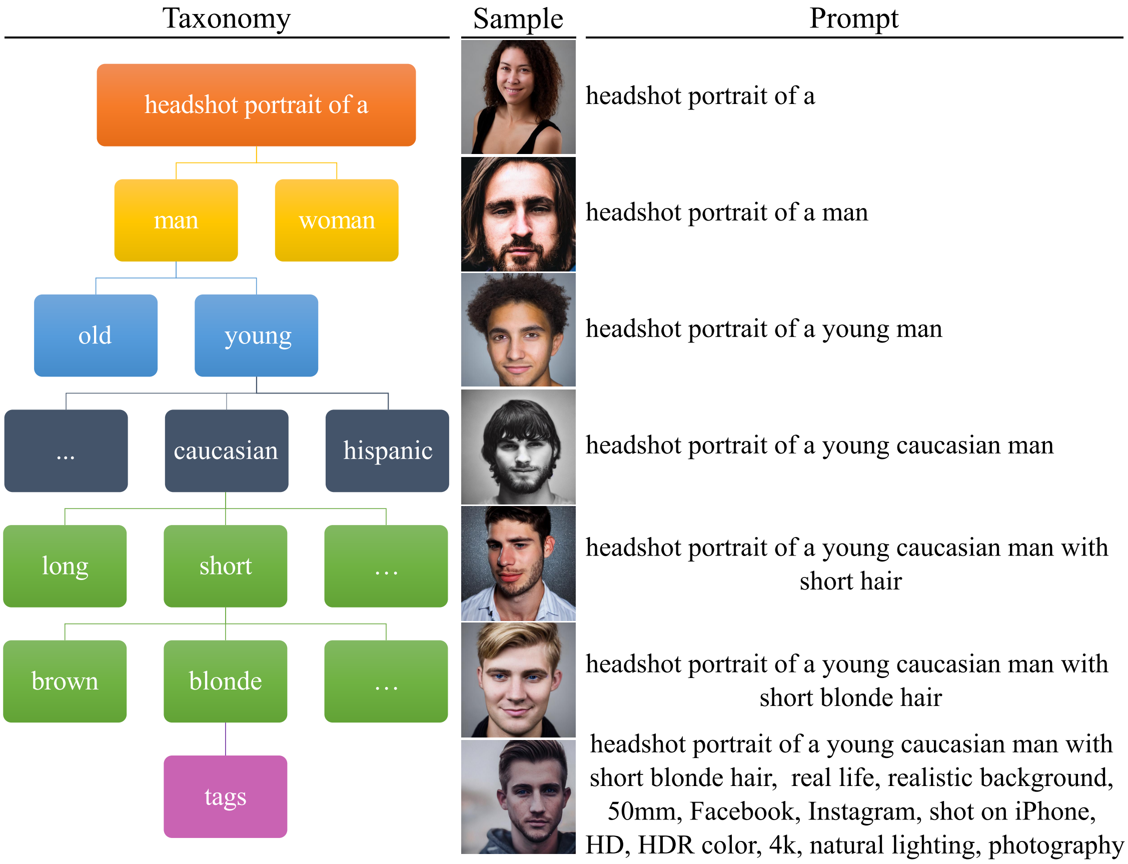


Figure 4.8: Examples of highly realistic face images (left) generated using Stable Diffusion and real (right) once extracted from FFHQ [135] dataset.



headshot portrait of a

headshot portrait of a man

headshot portrait of a young man

headshot portrait of a young caucasian man

headshot portrait of a young caucasian man with short hair

headshot portrait of a young caucasian man with short blonde hair

headshot portrait of a young caucasian man with short blonde hair, real life, realistic background, 50mm, Facebook, Instagram, shot on iPhone, HD, HDR color, 4k, natural lighting, photography

Figure 4.9: The taxonomy of our iterative prompt analysis. Starting from a simple base (i.e., "headshot of a"), we iteratively extend the prompt to obtain the desired output; we add some special tags to increase the realism of the image.

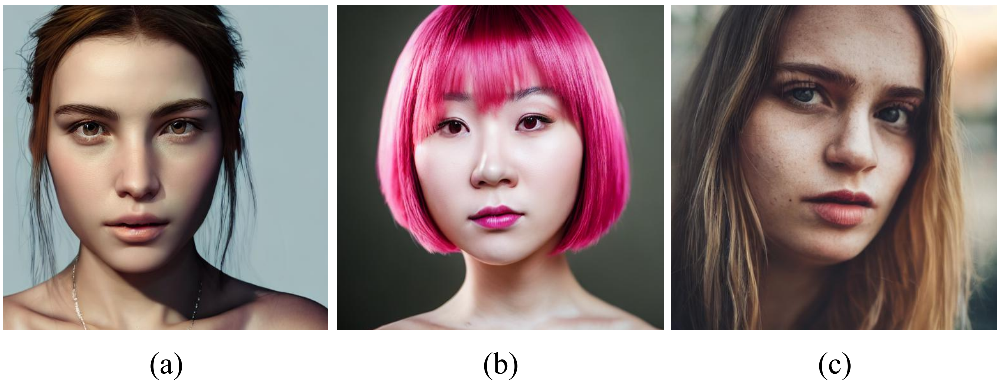


Figure 4.10: Examples of sample faces generated with three levels of realism. Figure (a) represents the less realistic sample and more like computer graphics renderings. Figure (b) shows a realistic sample with a typical shooting stage background. Figure (c) satisfies all the imposed constraints, resulting in an incredibly realistic photo immersed in a natural environment.

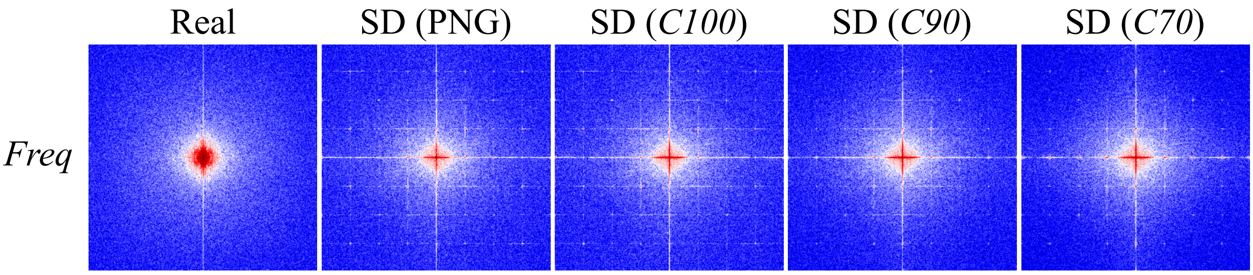


Figure 4.11: Spectra analysis of images generated with Stable Diffusion (SD) and the corresponding difference map with respect to the Real image on the top left corner. We apply two uniform color maps for a better view.

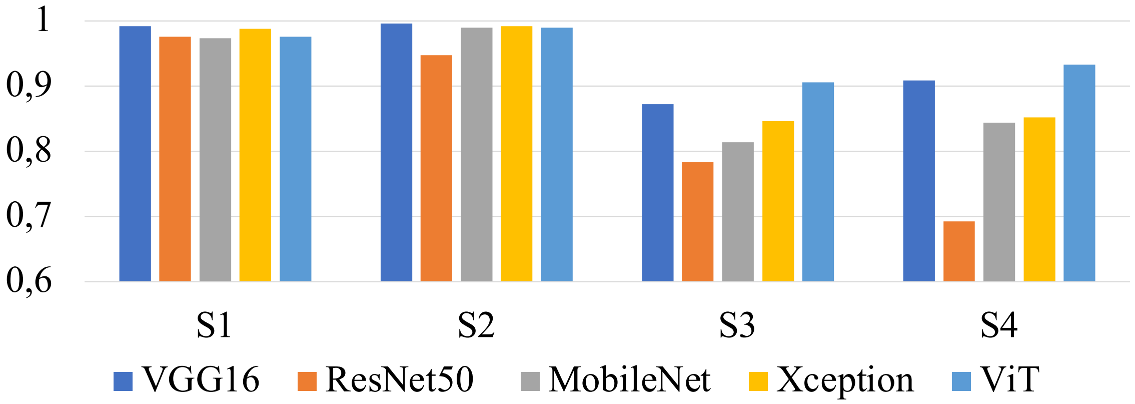


Figure 4.12: Different detectors trained on five classes: real, Stable Diffusion, StyleGAN, StyleGAN2, and StyleGAN3.

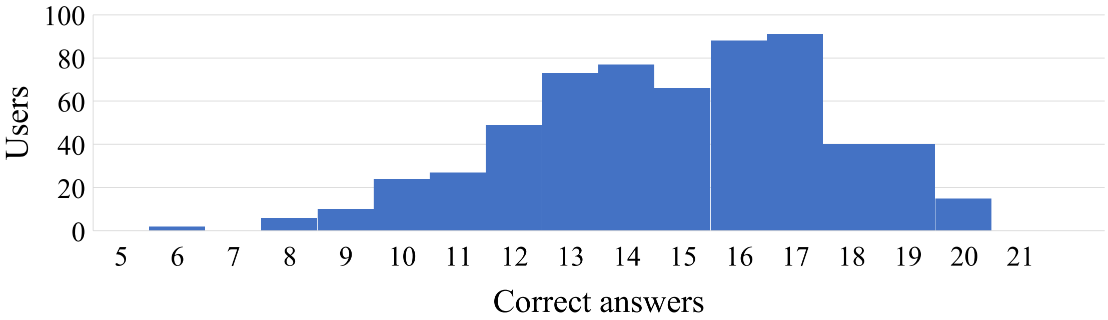


Figure 4.13: Distribution of correctly identified images. The histogram represents the number of users who get a certain number of correct answers. Most users correctly identified 17/20 images.

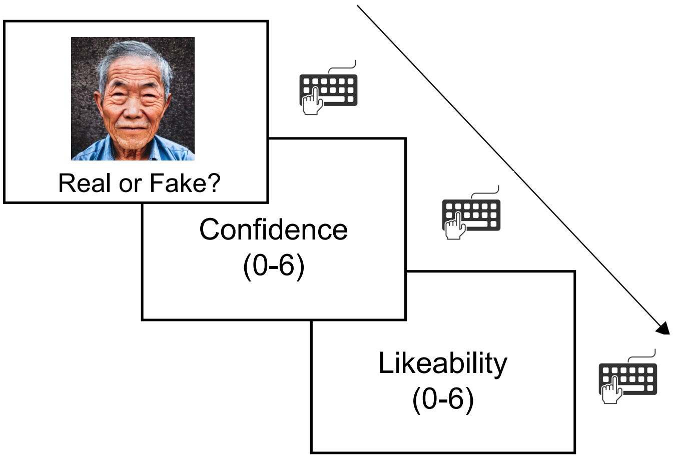


Figure 4.14: Procedure for human performance evaluation.

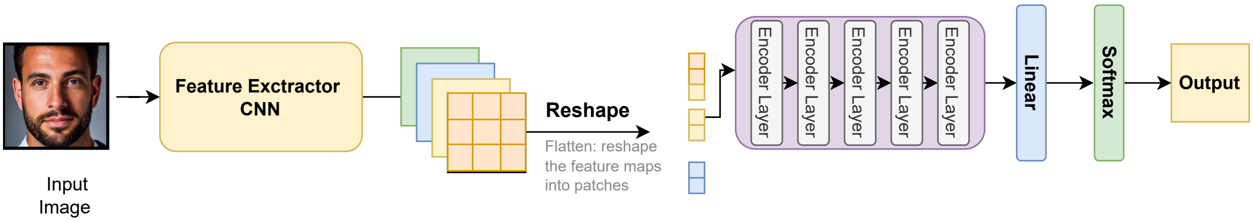


Figure 4.15: Proposed ResFormer architecture.

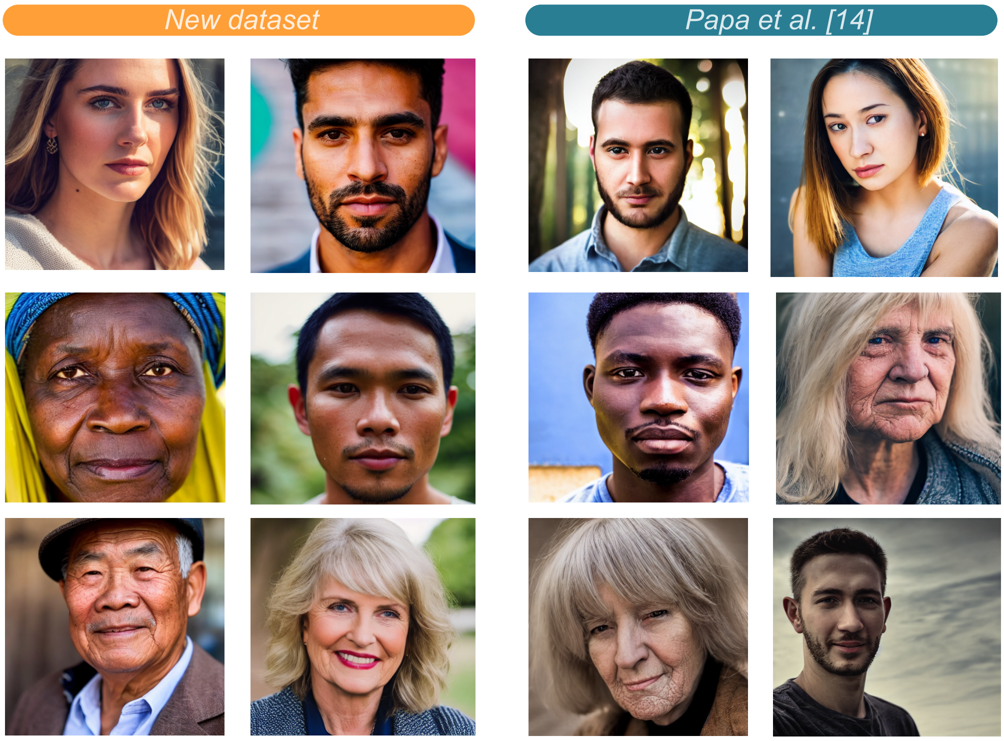


Figure 4.16: Improved generated images.

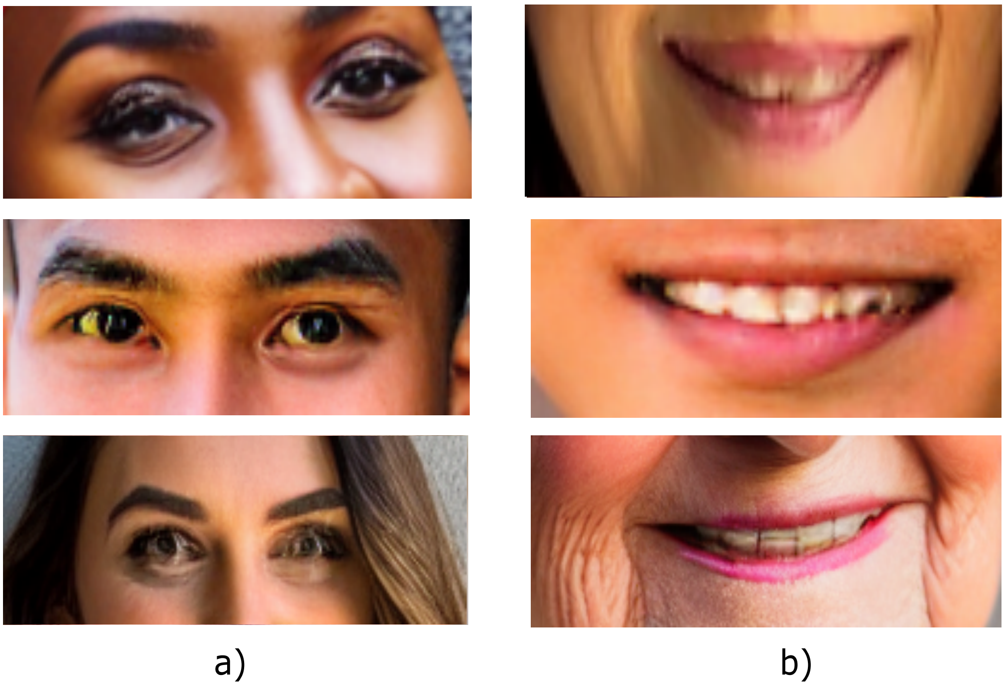


Figure 4.17: Common semantic errors produced by the model without negative prompts.

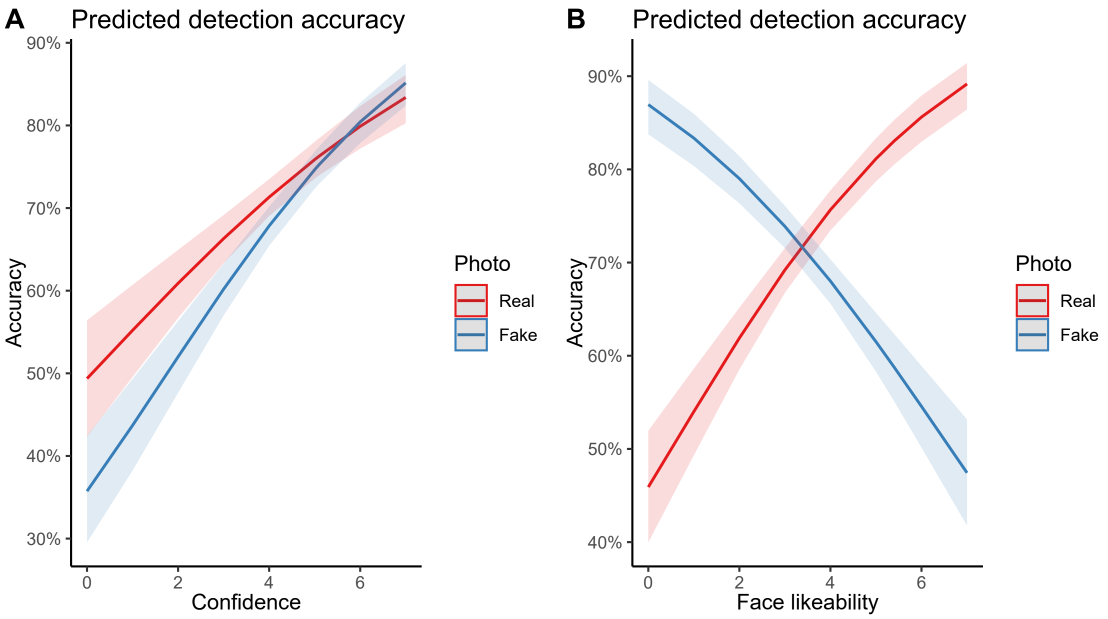
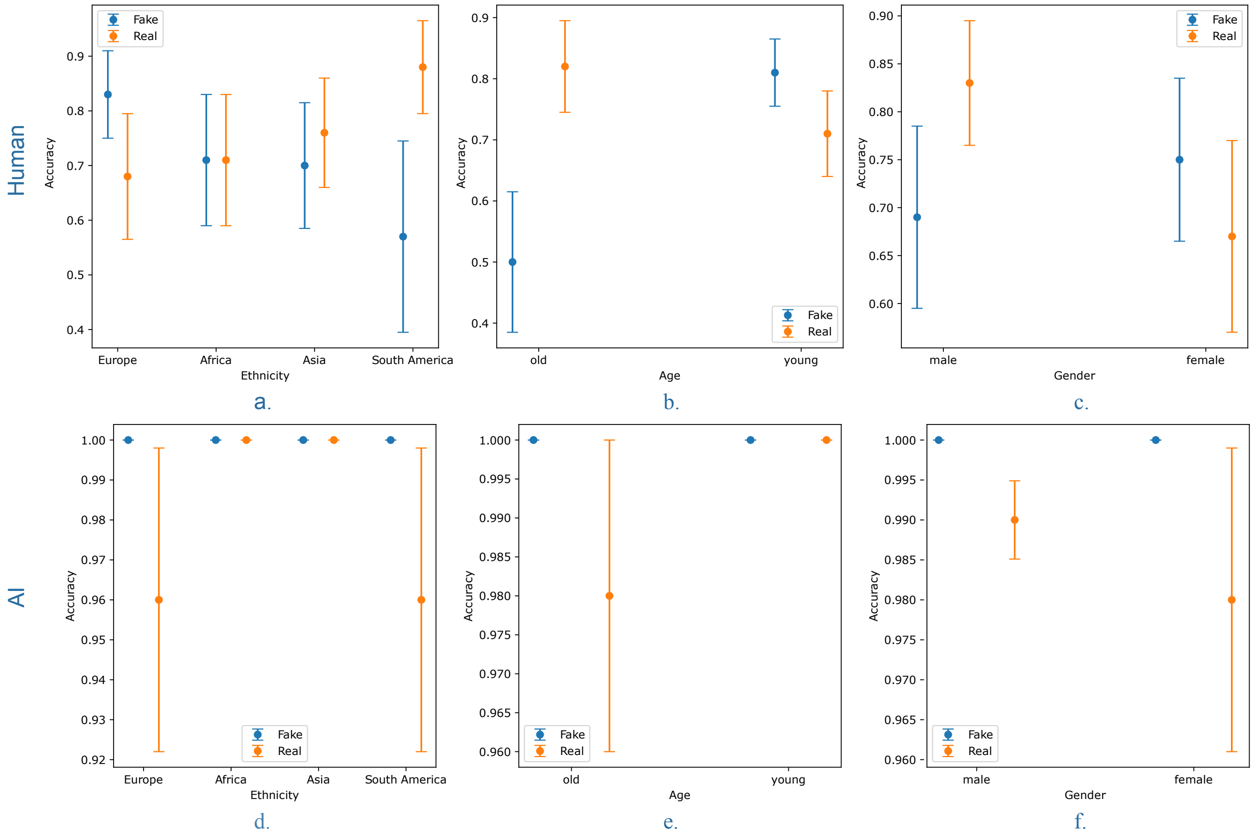


Figure 4.18: Predicted detection accuracy based on A: Confidence, B: Likeability of the photo.



a.

b.

c.

d.

e.

f.

Figure 4.19: Predicted detection accuracy for Ethnicity of faces (first column), Age of faces (second column), and Gender of faces (third column) for humans (first row) and machines (second row).

Chapter 5

Forensics analysis of media streams

In this chapter, we will analyze the information from another perspective. Forensic problems are often treated starting from an available dataset, assuming that this can represent what we will see in the future. However, this assumption, as also emerged in the previous chapters, appears to be rather limited and unrealistic, as, in reality, the data is very often dynamic. This problem becomes especially evident with online news, where knowledge and content evolve, develop, and accumulate over time. Information is constantly generated, shared, and updated across various platforms and media in the digital age. This makes static solutions too limited to apply in a real setting.

Although generalization remains an essential piece of the puzzle, it is clear that this alone is not enough in an ever-changing scenario. We, as humans, must also constantly learn and update ourselves regardless of our ability to apply our knowledge in multiple fields of our lives. Active learning allows us to select and add the necessary daily information to our knowledge base. Consequently, to apply forensic techniques in an ever-changing world, it is essential that these tools can continuously evolve and update. In this chapter, we focus precisely on this, treating the problem from two points of view. First, in Chapter 5.1, we propose a new detector capable of classifying news based on textual information and images attached to the news. This allows us to consider all the news information to look for patterns that indicate that this content could be fake. Next, we show how it is possible to make the detector capable of updating its knowledge through continual learning techniques.

Then, we analyze the robustness of such a detector from an adversary point of view. Specifically, we put ourselves in a scenario in which the attacker wants to manipulate the detector's behavior on a specific piece of news without being able to modify it. For example, suppose the attacker seeks to classify accurate information as fake. In a real-world scenario, this attacker may not be able to alter the text of the actual news story. In Chapter 5.2, we show how it is possible to create appropriately poisoned news to change the behavior of the online learning detector on the target news, which will then be misclassified. This powerful attack highlights new threats to this class of techniques.

5.1 A continual learning strategy for fake news detection

The massive adoption of social networks has made them a very effective tool for spreading false content. Fake news stories often spread faster and with a higher frequency than the real ones [291], but, more importantly, the more a user is exposed to the same content, the more she tends to perceive it as trustworthy [340]. This fact can have a more profound effect than one may expect. An example of this is the 2016 presidential election in the United States. Snopes[1] identified 529 social-media rumors about Donald Trump and Hillary Clinton that could have influenced the election outcome—the presidential elections of the first economic power in the world. Similarly, researchers have analyzed the effect of disinformation on other more recent events, such as Covid-19 [242] or the current war in Ukraine [154].

There are many challenges to face to counter this phenomenon. First, the most influential fake news contain both texts and images. For example, tweets with images obtain 18% more clicks, 89% more likes, and 150% more retweets than tweets with text-only content [322]. A similar trend takes place on Facebook, where the 87% of the posted photos have been liked, clicked, or shared [322]. Because of this fact, recent studies have analyzed the semantics of multimodal content to classify the news as real or false [41, 140, 269, 290, 302, 308, 309]. Although this type of approach seems to attain high levels of accuracy in most of the studies, its applicability in real scenarios is still somewhat limited. Indeed, most state-of-the-art approaches apply these techniques in a static setting, where the training and test data belong to a *fixed distribution*, known at design time. This assumption, however, does not reflect the ever-changing nature of news [7] being spread online based on recent events, as we have seen with Covid-19 or the war in Ukraine. Some studies proposed to tackle this problem from a different perspective, by analyzing the propagation of news or the communities and the users' reactions to such content [120, 204, 207, 262, 270]. However, these interactions can sometimes be complex to capture because they require monitoring of the entire network, something that is not always feasible. Therefore, analyzing the content stream remains the most accessible way.

Motivated by the discussion in the previous paragraph, in this chapter, we propose to model the latest news flow as an *incremental task*, where data arrive sequentially in batches, and each batch corresponds to a new fact that we want to learn to classify as real or fake. The main challenge is to learn without *catastrophic forgetting* [60, 160]: performance on a previously known task or domain should not degrade significantly over time as new tasks or domains are added.

Our contributions can be summarized as follows:

- We introduce a multimodal architecture (the *Tri-Encoder*) for fake-news classification based on the analysis of texts and images.

- We apply a *continual-learning strategy*, which allows to continually learn to classify new topics without losing the ability to classify previously known ones.

- We perform various measurements and comparisons of our approach with others. Our experiments demonstrate the robustness of the proposed solution: The Tri-Encoderachieves state-of-the-art performance in multimodal tweet classification. Furthermore, we show how the performance of a model tends to eventually degrade on older tasks without the adoption of an incremental-learning strategy. Thus, the proposed solution allows not only to maintain good performance over time, but it even improves compared to the ideal case in which all the topics are immediately available in the first training session.

[1] `https://www.snopes.com` – Fact-checking website and reference source for urban legends, folklore, myths, rumors, and misinformation.

5.1.1 Methodology

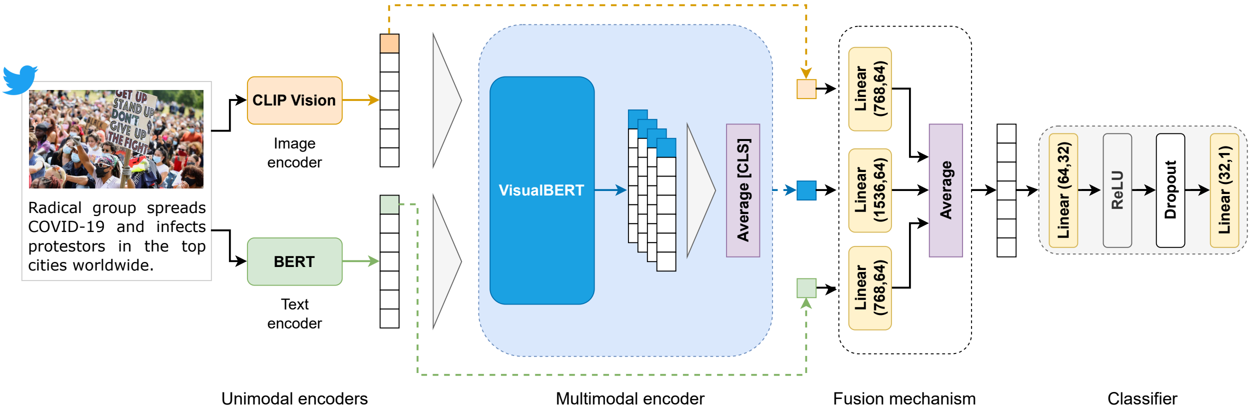


Figure 5.1: Overview of the proposed *Tri-Encoder* multimodal architecture.

This section introduces our multimodal encoder for news representation and explains the proposed continual-learning strategy for learning new topics over time. Our model aims at learning the discriminable feature representations for fake-news detection in a way that can constantly adapt to the evolving flow of the most recent facts. In this study, we focus on the news spread on Twitter, but the same framework can be extended to other social networks. Formally, given a tweet $X = \{T, V\}$ comprising textual information T and visual information V, our goal is to learn a target function $g(X, \theta) = Y$ that predicts whether the post is a fake ($Y = 0$) or true ($Y = 1$) content by examining the textual and visual information as well as the semantic relationship between the two types of information. Beyond that, we consider news a (potentially infinite) stream of unknown distributions $\mathcal{D} = \{\mathcal{D}^1, \ldots, \mathcal{D}^n\}$ over $X \times Y$, with X and Y input and output random variables, respectively. At time step i, the model learns a new function $f_i^{CL} = g(X, \theta^i)$ by updating its current parameters θ^{i-1} on a new fact $\mathcal{D}^i$ by training it on a training set $\mathcal{D}^i_{\text{train}}$ and testing it on a test set $\mathcal{D}^i_{\text{test}}$. The objective of the continual-learning algorithm is to minimize the loss $\mathcal{L}_D$ over the entire stream of data $\mathcal{D}$:

$$\mathcal{L}_D\big(f_n^{CL}, n\big) = \frac{1}{\sum_{i=1}^{n} |\mathcal{D}^i_{\text{test}}|} \sum_{i=1}^{n} \mathcal{L}_{\text{fact}}\big(f_n^{CL}, \mathcal{D}^i_{\text{test}}\big) \tag{5.1}$$

$$\mathcal{L}_{\text{fact}}\big(f_n^{CL}, \mathcal{D}^i_{\text{test}}\big) = \sum_{j=1}^{|\mathcal{D}^i_{\text{test}}|} \mathcal{L}_{\text{cls}}\big(f_n^{CL}(x_j^i), y_j^i\big) \tag{5.2}$$

where the loss $\mathcal{L}_{\text{cls}}\big(f_n^{CL}(x_j^i), y_j^i\big)$ represents the binary cross entropy loss.

In the remainder of this section, we introduce the multimodal encoder network that we use for encoding tweets and the continual learning strategy used to learn new facts.

The *Tri-Encoder* Model Architecture

The Tri-Encoder model architecture is shown in Figure 5.1. The model involves an *image encoder* and a *text encoder* to obtain unimodal image and text representations, and a *multimodal encoder* to fuse and align the image and text representations for multimodal reasoning based on transformers.

Text encoder. Given the text of a tweet, we first tokenize and embed it in a list of word vectors using WordPiece [307] with a vocabulary of 30,000 tokens and append two special characters to the input: the class token `[CLS]`, which is appended in front of each input example, and the separator token `[SEP]`. Then, we apply a transformer model over the word vectors to encode them into a list of N_T hidden state vectors $h_T \in \mathbb{R}^H$, including $h_{\text{CLS},T}$ for the text classification token. In all our experiments, we use the bidirectional BERT-base [65] model with 12 layers and 12 attention heads, which produces 768-dimensional hidden vectors. In the training phase, all weights are frozen except for the last two layers.

Image encoder. For the image encoder, we use the pre-trained CLIP's [233] visual feature extractor. Given an input image, we split it into 32×32 patches, which are then linearly embedded and fed into a ViT-B/32 [69] transformer model along with positional embeddings and an extra image classification token `[CLS]`. Similarly to the text encoder, the image-encoder output is a list of N_V image hidden state vectors $h_V \in \mathbb{R}^H$ ($H = 768$), each corresponding to an image patch, plus an additional $h_{\text{CLS},V}$ for the image classification token. Similarly to the text encoder, all weights are frozen except for the last two layers during training.

Multimodal encoder. We use an additional transformer model for learning a joint contextualized representation of the image and text hidden states. Specifically, we apply the VisualBERT [166] model, which consists of a stack of transformer layers that align the regions of the input image with the textual input through self-attention. Compared to a simple concatenation of the two unimodal embeddings, this configuration allows *cross-attention* between the projected unimodal image and text representations and fuses the two modalities. VisualBERT is pre-trained on the Visual Commonsense Reasoning dataset [320], which consists of $290K$ questions derived from $110K$ movie scenes having as focus the visual commonsense reasoning. This encoder takes as input the visual (h_V) and textual (h_T) hidden representations extracted from the unimodal models and produces $N_T + N_V + 2$ multimodal hidden state vectors $h_M \in \mathbb{R}^H$ ($H = 768$), where N_T and N_V are the numbers of text tokens and image patches, respectively, and the two additional vectors are the special `[CLS]` and `[SEP]` (between the modalities) tokens.

The output of the last layer may not always be the best representation of the input when fine-tuning for downstream tasks. Previous studies proved that for pre-trained language models, the most transferable contextualized representations of input text tend to occur in the middle layers, whereas the top layers specialize in language modeling [54, 89, 127, 176, 280, 343]. Therefore, inspired by the same considerations, we average the penultimate last three layers' output and concatenate the averaged hidden state vector with the `[CLS]` hidden state vector of the output layer, producing a 1536-dimensional output h_{MM}. We validate this choice in Chapter 5.1.2.

Fusion mechanism. In the fusion step, the visual, textual, and the multimodal `[CLS]` feature vectors $h_{T,\texttt{CLS}}$, $h_{V,\texttt{CLS}}$, and $h_{MM,\texttt{CLS}}$ are all projected onto a 64-dimensional subspace through a linear layer, producing the corresponding $h'_{T,\texttt{CLS}}$, $h'_{V,\texttt{CLS}}$, and $h'_{M,\texttt{CLS}}$ vectors. Finally, we calculate a weighted average of these vectors

$$h_{TVM} = \text{avg}(w_T h'_{T,\texttt{CLS}} + w_V h'_{V,\texttt{CLS}} + w_M h'_{M,\texttt{CLS}}) \tag{5.3}$$

where w_T and w_V are fixed to 0.25, and $w_M = 0.5$.

Classifier. The final step of the Tri-Encoder is the classification step. The classifier is composed of two linear layers generating a 32-dimensional and a 1-dimensional outputs, and are separated by the rectified linear unit (ReLU) and dropout operations. A sigmoid activation function follows the output of the last layer:

$$\sigma(x) = \frac{1}{1 + exp(-x)}$$

Values below a threshold τ are predicted *false*. Experimentally, we found $\tau = 0.46$ as the optimal value.

Continual-Learning Strategy

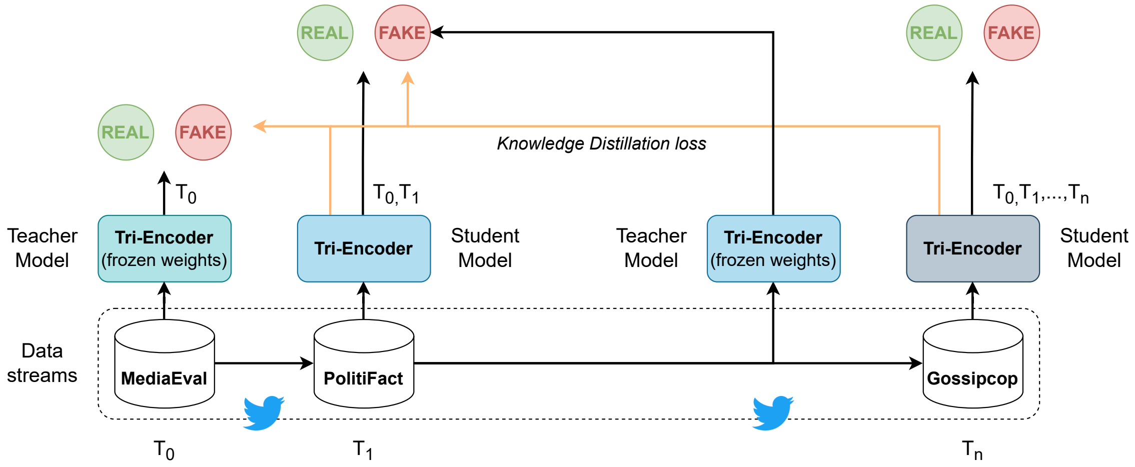


Figure 5.2: Continual-learning strategy: For each new task T_i, we apply the knowledge-distillation loss as penalty factor.

Our goal in this section is to propose a continuous learning strategy that allows the Tri-Encoder to learn to classify the latest news articles as soon as they become available. To this end, it is essential to avoid catastrophic forgetting, as we want the model to continue to classify previously known news accurately. A naive idea might be to retrain the model on an ever-growing set of training data; however, such an approach can become prohibitively expensive as the volume of data grows over time. On the other hand, in the context of continuous learning, the model should not overfit to a new topic because it would cause it to forget its previous skills. The strategy devised in this study is based on *regularization*, as it has properties well-suited to this use case. *Scalability* is an important driver as we don't want the model size to increase as the data size increases. Another big advantage of a regularization approach is that we don't need to store the previous data set in *memory*. Indeed, regularization approaches consist in modifying the updating process of weights during learning and lend themselves well to maintaining the memory of previous knowledge.

We choose to adopt a *knowledge distillation* approach as shown in Figure 5.2. Distillation techniques were introduced by Hinton et al. [105] as a means to transfer knowledge from a neural network T (the *teacher*) to a neural network S (the *student*). The key idea behind knowledge distillation is that soft probabilities predicted by a network of trained "teachers" contain much more information about a data point than a simple class label. For example, if multiple classes are assigned high probabilities for an image, this could mean that the image must be close to a decision boundary between those classes. Forcing a student to mimic these probabilities should then cause the student network to absorb some of this knowledge that the teacher discovered, above and beyond the information in training labels alone. To implement this strategy, we modify the classification loss $\mathcal{L}_{\text{cls}}$ in Equation 5.2 by adding a regularization factor

$$\mathcal{L}'_{\text{cls}} = \alpha\mathcal{L}_{kd} + \beta\mathcal{L}_{\text{cls}}\big(f_n^{CL}(x_j^i), y_j^i\big), \tag{5.4}$$

where α and β are experimentally set to 0.5 and 0.6, respectively, and $\mathcal{L}_{kd}$ is the mean squared error (MSE) loss that measures the squared L2 norm between the teacher and the student outputs.

5.1.2 Experiments

In this section, we validate the model and the proposed continual-learning solution. In Chapter 5.1.2, we begin with a brief description of the datasets. Subsequently, in Chapters 5.1.2 and 5.1.2, we evaluate the performance of the Tri-Encoder against the state of the art and propose an ablation study to validate the architectural choices. Finally, in Chapter 5.1.2, we measure learning performance in the case of ever-changing facts.

Datasets

To evaluate the performance of the proposed model and continual-learning strategy, we conduct experiments on three datasets collected by Twitter. The three datasets contain both tweets and images and were chosen because of the different topics they cover. We provide a brief description below. For all datasets, we preprocess both text and images. All images have been center cropped to 224 × 224 pixels and normalized by mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$. For texts, we translate all non-english tweets with the Google Translator's API[2], substitute all usernames with `@name`, and remove all punctuation (except for the question marks) and hashtags.

MediaEval. The dataset comes from the *MediaEval Verifying Multimedia Use benchmark* [29], a challenge launched with the aim of detecting false content on Twitter. The original dataset contains 6255 real and 9596 fake tweets related to 17 different events. The test set contains 1107 tweets. Training and test sets are built such that they don't share events. The dataset incorporates textual content, attached images/videos, and additional social context information. Given that in this study we are interested in detecting fake news by embedding text and image information. we remove tweets without text or image.

FakeNewsNet: PolitiFact and GossipCop. To test the model's ability to incrementally learn new topics, we use the FakeNewsNet dataset [260]. Similarly to MediaEval, the dataset contains multimodal information and contains two distinct subsets of posts collected from the *PolitiFact*[3] and *GossipCop*[4] websites. The original dataset includes news articles along with the related tweets, but as for MediaEval, we just take tweets into consideration. The PolitiFact dataset contains a total of 3359 tweets (2306, 226, and 827 tweets for training, validation, and testing respectively), and the GossipCop dataset contains 3002, 500, and 1339 tweets for training, validation, and testing.

[2] https://pypi.org/project/googletrans/

[3] https://www.politifact.com/

[4] https://www.gossipcop.com/

Fake-News Detection Performance

To validate the proposed Tri-Encoder, we propose a comparison with state-of-the-art methods. In particular, we validate the model's performance against (1) well-known unimodal deep-learning architectures and (2) multimodal solutions designed for fake-news detection. All the models have been trained on MediaEvalfor 10 epochs with a batch size of 32, a learning rate of 3e-05, and the Adam [147] optimizer.

Model	F1-micro	F1-macro
BERT [65]	0.6247	0.6238
ResNet50 [102]	0.7021	0.6962
VGG19 [268]	0.6275	0.6273
CLIP Vision [233]	0.7440	0.7353
VisualBERT [166]	0.7978	0.7934
MVAE [140]	0.745	0.744
EANN [302]	0.715	0.719
EANN- [302]	0.648	0.6385
SpotFake [269]	0.778	0.760
MFN [41]	0.808	0.785
MCAN [308]	0.809	0.808
CALM [309]	0.845	0.839
Tri-Encoder (ours)	**0.851**	**0.845**

Table 5.1: Fake news detection performance on the MediaEval [29] dataset.

Model	All news		Fake News			Real News		
	Acc/F1	F1-macro	Prec.	Rec.	F1	Prec.	Rec.	F1
MediaEval	0.8515	0.8446	0.8586	0.8968	0.8773	0.8399	0.7857	0.8119
PolitiFact	0.789	0.789	0.802	0.766	0.784	0.778	0.812	0.795
GossipCop	0.708	0.706	0.691	0.685	0.688	0.723	0.728	0.725

Table 5.2: Model trained on each dataset in isolation from scratch

Table 5.1 reports the results of this first experiment. We can notice that our Tri-Encoderarchitecture outperforms all the other methods, followed by CALM [309], which achieves comparable performance. In the next section we study the architectural choices that led to the proposed Tri-Encoder. We can generally observe that multimodal models perform better than unimodal ones, confirming the additive contribution of the images to an accurate classification. We can also notice that for the unimodal architectures, models that analyze images outperform BERT, a model based on text. A possible explanation for this could be that in the MediaEvaldataset, many fake images have been manipulated in a way that makes the detection of such manipulation highly accurate by the image classifiers. Finally, in Table 5.2, we report the performance of the Tri-Encoderacross PolitiFactand GossipCop. On both datasets the model obtains performance comparable to those achieved on MediaEval, confirming the robustness of the Tri-Encoder.

Ablation Study

To evaluate the design choices of our model, we now analyze several possible multimodal variants. We consider three baseline models: the *Simple-Encoder* (SE), the *Dual-Encoder* (DE), and the

VisualBERT (VB) [166]. We also compare three feature extractors for the image features: ResNet50 *(R)*, CLIP Vision *(C)*, and ViT *(V)*.

Model	**All news**		**Fake News**			**Real News**		
	Acc/F1	**F1-macro**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
SE(R)	0.6347	0.6338	0.7180	0.5984	0.6528	0.5584	0.6837	0.6147
SE(C)	0.7586	0.7520	0.7820	0.8031	0.7924	0.7250	0.6987	0.7119
DE(R)	0.7094	0.6711	0.6844	**0.9158**	0.7834	**0.7921**	0.4316	0.5587
DE(C)	0.7058	0.7024	0.7623	0.7079	0.7341	0.6413	0.7029	0.6707
VB(R)-b	0.7613	0.7564	0.7948	0.7873	0.7910	0.7172	0.7264	0.7218
VB(R)-cat	0.7513	0.7460	0.7846	0.7809	0.7828	0.7070	0.7115	0.7092
VB(R)-avg	0.7367	0.7313	0.7728	0.7666	0.7697	0.6892	0.6965	0.6928
VB(C)-b	0.7522	0.7382	0.7479	0.8571	0.7988	0.7606	0.6111	0.6777
VB(C)-cat	0.7358	0.7337	0.8003	0.7190	0.7575	0.6672	0.7585	0.71
VB(C)-avg	**0.7978**	**0.7934**	**0.8248**	0.8222	**0.8235**	0.7617	0.7649	**0.7633**
VB(V)-b	0.6766	0.6766	0.7956	0.5873	0.6757	0.5892	**0.7970**	0.6775
VB(V)-cat	0.6493	0.6485	0.7351	0.6079	0.6655	0.5719	0.7051	0.6315
VB(V)-avg	0.7167	0.7115	0.7593	0.7412	0.7502	0.6625	0.6837	0.6729

Table 5.3: Multimodal methods performances on the MediaEval [29] dataset.

Simple-Encoder(SE). This model is based on the simple concatenation of the features extracted from images and texts. For text, we use BERT, taking the [CLS] representation for the last hidden state. Each unimodal model is fed into a linear layer with an output size of 512. Then the output of these linear layers is concatenated, producing a 1024-dimensional vector. The vector is finally passed through two linear layers of 1024×32 and 32×1 dimensions, which are separated by a ReLU function, a dropout layer (set to 0.4), and a sigmoid activation function.

Dual-Encoder(DE). This architecture has been inspired by the double visual textual transformer model (DVTT) introduced by Messina et al. [202] for the task of classifying hateful memes. The model is made of two different transformer networks: one for text and one for images; however, each network is conditioned by the other, enriching the text with visual information from the text encoder and vice-versa. As for the Simple-Encoder, the textual and visual representations are projected by a linear layer of with a 512-dimensional output, before being fed into the transformer model. The textual representation is taken from the last hidden state of the BERT model. Similarly, when CLIP Vision is employed as a visual feature extractor, the image representation comes from the last hidden state of the transformer encoder. In the case of ResNet50, feature maps are extracted from the second last layer, and a $(6, 6)$ pooling is applied. Finally, we concatenate the [CLS] tokens from both transformers, obtaining a 1024-dimensional embedding.

VisualBERT (VB). This model combines image regions and language with a transformer, allowing self-attention to discover implicit alignments between language and vision. It is pre-trained on visual-reasoning tasks, thus offering a good starting point for visual commonsense reasoning. We consider the following three variants with all the backbones:

- *base (b):* the representation of the [CLS] token representation from the last hidden state is fed into the classifier;

- *concatenation (cat):* the [CLS] token representations from the last four layers are concatenated before classification;

- *average (avg):* the `[CLS]` token representations from the penultimate three layers are averaged and concatenated with the last `[CLS]` token before classification.

Table 5.3 summarizes all the experiments. The results show a consistent advantage of the *VB(C)-avg* configuration over the others, which is the same configuration used for our Tri-Encoder, introduced in Chapter 5.1.1. Besides that, we can observe that using CLIP Vision for the visual component achieves superior performance in all the configurations. As the original model is trained in a multimodal setting, we make the hypothesis that this is because of the fact that it manages to extract features more aligned with the textual component. Regarding the compared architectures, VisualBERT achieves, on average, superior performance compared to the Simple-Encoderand Dual-Encoder.

Robustness to Incremental Topics

Whereas the previous results demonstrate the effectiveness of the proposed method on a task, in this section, we evaluate the model's performance on new tasks in a continual-learning scenario. Specifically, we evaluate the performance of knowledge distillation (*KD*) compared to two other strategies: the transfer learning (*TL*) and the elastic weight consolidation (*EWC*) [149], which can be seen as an improvement of the L2-regularization.

Elastic weight consolidation (EWC)

EWC remembers old tasks by selectively slowing down learning on weights that are important for these tasks. As shown by Kirkpatrick et al. [149], in learning from a distribution $\mathcal{D}^{i-1}$ to a new distribution $\mathcal{D}^i$, there exist many configurations of the model parameters θ that lead to the same performance. Actually, the (common) over-parameterization of the models makes more likely the existence of a solution θ_i^* for task $\mathcal{D}^i$ (1) that is close to the optimal set of parameters θ_{i-1}^* and (2) that minimizes the loss function for task $\mathcal{D}^{i-1}$. Therefore, previous tasks' performance can be kept by constraining—using a quadratic penalty—the parameters to stay in a region centered in θ_{i-1}^*. Formally, the function $\mathcal{L}_{EWC}$ that we minimize in EWC is:

$$\mathcal{L}_{EWC}(\theta) = \mathcal{L}_{\mathcal{D}^i}(\theta) + \sum_j \frac{\lambda}{2} F_j(\theta_j - \theta_{i-1,j}^*)^2, \tag{5.5}$$

where $\mathcal{L}_{\mathcal{D}^i}(\theta)$ is the classification loss for task $\mathcal{D}^i$, F is the Fisher information matrix, λ controls how important the old task $\mathcal{D}^{i-1}$ is compared to the new one, and j labels each parameter. When moving to a third task (i.e., task $\mathcal{D}^{i+1}$), EWC will try to keep the network parameters close to the learned parameters of both tasks $\mathcal{D}^{i-1}$ and $\mathcal{D}^i$. This can be enforced either by using two separate penalty terms, or a single one (after noting that the sum of two quadratic penalties is itself a quadratic penalty). In all our experiments, we experimentally set $\lambda = 10^3$.

Continual-Learning Metrics. We evaluate the performance of continual learning with respect to the two most commonly used metrics [180]: the *average accuracy* and the *forgetting* (also known as *backward transfer*). For testing, we consider access to a test set for each of the t tasks. After the model finishes learning about task T_i, we evaluate its test performance on all t tasks. By doing so, we construct the evaluation matrix $E \in [0,1]^{t\times t}$, where $E_{i,j}$ is the test classification accuracy of the model on task T_j after observing the last sample from task T_i.

Average accuracy Average accuracy (ACC) is a simple mean across all tasks

$$ACC = \frac{1}{t}\sum_{i=1}^{t} E_{T,i} \tag{5.6}$$

where $a_{T,i}$ is the accuracy of the model on sample i after observing the last sample from task T.

Forgetting (backward transfer.) Backward transfer (BWT) measures the influence that learning of a task $\mathcal{D}^i$ has on the performance on a previous task $\mathcal{D}^{i-1}$. A *positive* backward transfer when learning the new task $\mathcal{D}^i$ increases the performance on $\mathcal{D}^{i-1}$, whereas a *negative* backward transfer when learning $\mathcal{D}^i$ decreases the performance on some preceding task $\mathcal{D}^{i-1}$. Large negative backward transfer is also known as *catastrophic forgetting.*

$$BWT = \frac{1}{t-1}\sum_{i=1}^{t-1} \left(E_{T,i} - E_{i,i}\right). \tag{5.7}$$

The higher the accuracy, the better is the model. If two models have comparable ACC, the most preferable one is the one with lowest BWT.

Static training sessions

Before illustrating the results on the continual-learning setting, let's evaluate the performance of the models when the Tri-Encoderis trained from scratch on *all datasets simultaneously.* This allows us to evaluate the performance of the continual learner in the ideal scenario where the data are immediately available in the first training session. In Table 5.4, we report the results in terms of F1 score when the training set is balanced or unbalanced between the three datasets. As expected, when the data are unbalanced, the F1 score on the MediaEvaldataset is higher than the others, having three times the number of samples that the other slices have. By balancing the data, the overall accuracy doesn't change much, but the distribution between the different portions is more even. We can also notice that the model's performance on MediaEvaldrops slightly compared to the case in which the model is trained only on this dataset (see Table 5.1). This could be justified by the fact that when the model is trained on all datasets simultaneously, the broader distribution of facts present in all datasets leads the model to converge into a region where it minimizes errors on all topics, but which leads to a slight performance drop on the MediaEvaltopics.

Tested dataset	**Not Balanced**		**Balanced**	
	F1-micro	**F1-macro**	**F1-micro**	**F1-macro**
MediaEval	0.7810	0.7762	0.7105	0.6855
PolitiFact	0.6251	0.6232	0.6529	0.6519
GossipCop	0.6781	0.6776	0.6855	0.6848
All	0.6984	0.6978	0.6966	0.6895

Table 5.4: Results of the model trained from scratch on all three datasets available at once. The *Balanced* results indicate that in the training dataset, we balance the facts present in the three datasets and the *Not Balanced* case, reports the results of the model trained on the concatenation of the three unbalanced training sets.

Furthermore, we evaluate the model's performance in a scenario where we apply *transfer learning.* Starting from MediaEvalas the first task ($T1$), in Table 5.5(a) we see the performance after applying transfer learning to $T2 = PolitiFact$ and to $T3 = GossipCop$, and in Table 5.5(a) we see the performance after applying transfer learning to $T2 = GossipCop$ and to $T3 = PolitiFact$. We can see that in both cases, the model suffers from catastrophic forgetting. Indeed, as we train it on new datasets, it becomes less accurate on previously seen ones. In the following section we see how our incremental-learning strategy manages to reduce this problem.

Task	**MediaEval**	**PolitiFact**	**GossipCop**
$T1$: MediaEval	0.8515	-	-
$T2$: PolitiFact	0.7312	0.7932	-
$T3$: GossipCop	0.7218	0.5224	0.7117

(a) $T1 = MediaEval, T2 = PolitiFact, T3 = GossipCop$

Task	**MediaEval**	**GossipCop**	**PolitiFact**
$T1$: MediaEval	0.8515	-	-
$T2$: GossipCop	0.6860	0.7259	-
$T3$: PolitiFact	0.6466	0.5123	0.7351

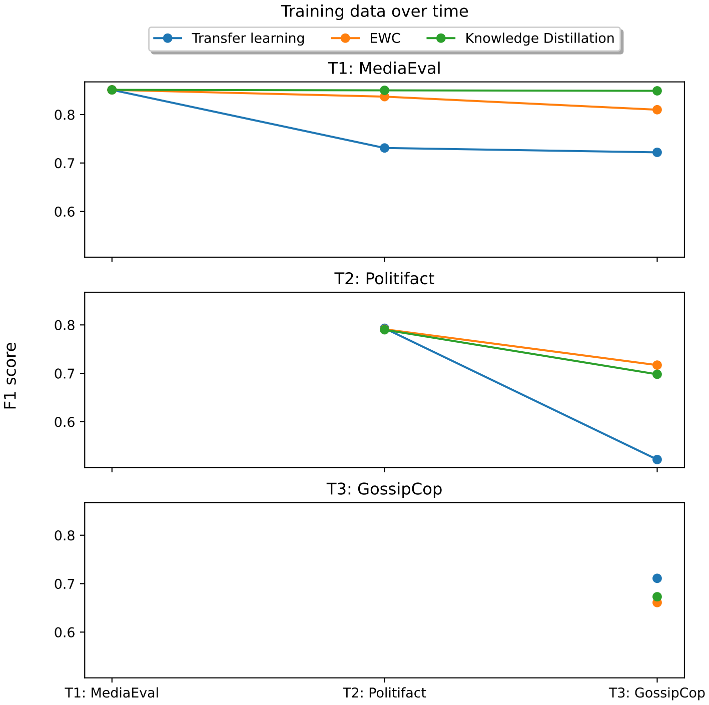
(b) $T1 = MediaEval, T2 = GossipCop, T3 = PolitiFact$

Table 5.5: Transfer learning performance in terms of F1 score of the model trained on $T1$, followed by transfer learning to $T2$, followed by transfer learning to $T3$. The rows indicate the dataset on which we perform the transfer learning (or the initial training for row $T1$) and the columns indicate the dataset on which we perform evaluation. For instance, Table (a) shows that after performing the initial training on the $T1 = MediaEval$ dataset, and then performing transfer learning to $T2 = PolitiFact$ followed by transfer learning to $T3 = GossipCop$, the F1 score on the MediaEvaldataset is 0.7218, and the F1 score on the PolitiFactdataset is 0.5224.

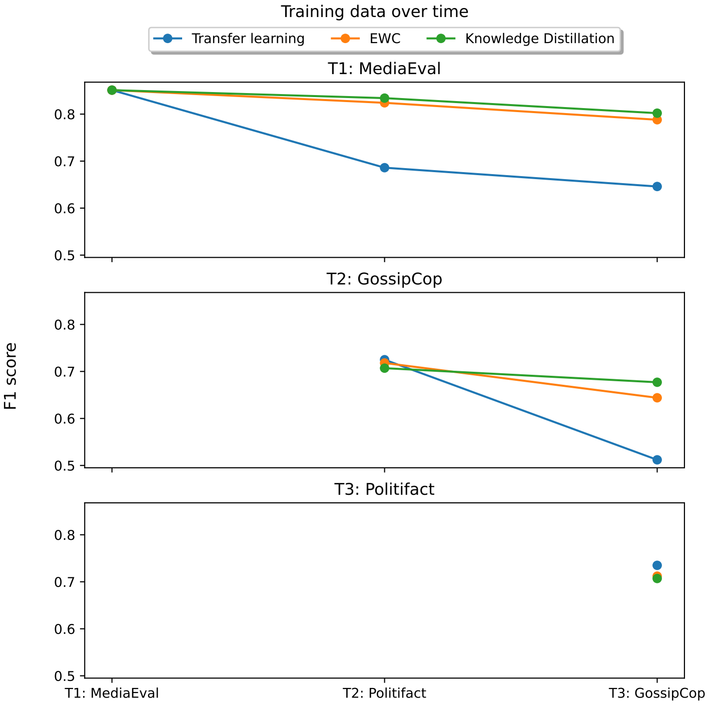
Continual-training sessions

We now evaluate the robustness of the model through several continual-learning sessions. To do this, in the first training session $T1$, we train the model on MediaEval. We chose this dataset for the first session as it is the largest among those considered and it allows a first training phase of the Tri-Encoderwithout causing overfitting. In subsequent training sessions, we expose the model to the new facts in GossipCopand PolitiFact. To this end, we introduce two more training sessions, namely $T2$ and $T3$. The goal of the continual learner is to learn new tasks without encountering catastrophic forgetting of the previous ones. To validate this approach, we compare the performance of knowledge distillation (KD) with respect to EWC and transfer learning.

In Figure 5.3, we show the performance of all the strategies in terms of F1 score. In Figure 5.3(a) we train first on MediaEval($T1$), and then on PolitiFact($T2$) and GossipCop($T3$). In Figure 5.3(b) we switch GossipCop($T2$) and PolitiFact($T3$). From both figures, we can see that the performance of the knowledge-distillation approach remains more or less constant on $T1$ during all the training sessions. For concreteness let us look at Figure 5.3(a). EWC's performance on this task is more or less comparable, although it suffers a more pronounced drop in F1 score when we switch from PolitiFact($T2$) to GossipCop($T3$). In the case of transfer learning, we notice a substantial drop in performance with the arrival of new tasks. This is absolutely justifiable because, in transfer learning, we do not impose to the model to perform well also in the previous tasks. In the second plot, where we evaluate with respect to the dataset $T2 = PolitiFact$, we can observe a similar behavior. Knowledge distillation and EWC have more or less similar performance, with a small drop (about 10%) in F1 score from $T2$ to $T3$, and with a high drop in the case of transfer learning.



(a) T_1 = MediaEval, T_2 = PolitiFact, T_3 = GossipCop



(b) T_1 = MediaEval, T_2 = GossipCop, T_3 = PolitiFact

Figure 5.3: F1 score of the Tri-Encoderover all tasks during time. In each of figures (a) and (b), the first plot shows the F1 score evaluated on dataset $T1$ (MediaEvalfor both of them), the second one the F1 score for $T2$ (PolitiFactfor (a), GossipCopfor (b)), and the third one for $T3$ (GossipCopfor (a), PolitiFactfor (b)).

Training Strategy	**Task**	**All news**		**Fake News**			**Real News**		
		Acc/F1	**F1-macro**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
TL	$T1$	0.722	0.721	0.839	0.656	0.736	0.621	**0.818**	0.706
	$T2$	0.522	0.521	0.522	0.478	0.499	0.522	0.566	0.543
	$T3^*$	**0.712**	**0.711**	**0.703**	**0.786**	**0.695**	**0.719**	**0.735**	**0.727**
EWC	$T1$	0.810	0.796	0.801	0.903	0.849	0.828	0.675	0.744
	$T2^*$	**0.717**	**0.717**	0.729	**0.687**	**0.707**	**0.706**	0.747	**0.744**
	$T3$	0.661	0.658	0.661	0.595	0.627	0.661	0.721	0.689
KD	$T1^*$	**0.849**	**0.841**	**0.845**	**0.913**	**0.878**	**0.857**	0.758	**0.804**
	$T2$	0.698	0.693	**0.758**	0.578	0.656	0.661	**0.817**	0.731
	$T3$	0.673	0.673	0.643	0.709	0.674	0.706	0.639	0.671

(a) $T1$: MediaEval, $T2$: PolitiFact, $T3$: GossipCop.

Training Strategy	**Task**	**All news**		**Fake News**			**Real News**		
		Acc/F1	**F1-macro**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
TL	$T1$	0.647	0.584	0.650	0.873	0.745	0.633	0.318	0.423
	$T2$	0.512	0.428	0.495	**0.937**	0.647	0.683	0.123	0.208
	$T3^*$	**0.735**	**0.734**	0.702	**0.813**	**0.754**	**0.78**	0.658	0.714
EWC	$T1$	0.788	0.771	0.777	**0.902**	**0.835**	**0.814**	0.624	0.707
	$T2$	0.644	0.635	0.589	0.845	**0.694**	**0.765**	0.461	0.575
	$T3$	0.712	0.708	**0.773**	0.597	0.674	0.674	**0.826**	**0.742**
KD	$T1\ast$	**0.802**	**0.791**	**0.810**	0.868	0.838	0.787	**0.705**	**0.744**
	$T2\ast$	**0.677**	**0.677**	**0.636**	0.758	0.692	0.731	**0.604**	**0.661**
	$T3$	0.707	0.704	0.757	0.607	0.674	0.674	0.807	0.735

(b) $T1$: MediaEval, $T2$: GossipCop, $T3$: PolitiFact.

Table 5.6: F1 score of the Tri-Encoderover all tasks after the last training session. Bold values indicate the best performance on a task. Tasks marked with $\ast$ indicate the learning strategy that performed best on those specific tasks.

Finally, in the last training session, transfer learning outperforms the other strategies, whereas knowledge distillation and EWC again have comparable performance. The results generally suggest greater robustness of continual learning methods compared to transfer learning. Although knowledge distillation and EWC obtain comparable performance in all training sessions, the former is more robust on the oldest task ($T1$), guaranteeing superior stability on all learning sessions.

For a more detailed report of the performance of the three strategies after the last training session, we report the results in terms of F1 score in Table 5.6. For each task, we report the best results in bold. We also mark with $\ast$ the strategy that achieves the best performance on a given task. As also mentioned in the discussion of Figure 5.3, transfer learning achieves the best performance only in the last task ($T3$). However, knowledge distillation is shown to be the most robust method on the first task ($T1$), followed by EWC. Compared to Table 5.2, we can see that although the performance degrades slightly on all tasks compared to standard training, continuous-learning strategies still achieve acceptable performance on all three tasks. Moreover, comparing the results with those of Table 5.4, we can even notice that with continual-learning strategies, we achieve higher performance compared to training all three datasets in a single session.

5.1.3 Discussion

The experiments in the previous section indicate that continuous-learning strategies can be used successfully on facts that evolve over time. The results suggest a better effectiveness of the knowledge-distillation strategy than the others; yet, to confirm these results, we analyze them with respect to the continual-learning metrics introduced in Chapter 5.1.2.

Table 5.7 shows the average accuracy and forgetting of transfer learning, EWC, and knowledge distillation on the three tasks after the last training session. In particular, Table 5.7(a) shows the results of the training session in which we train from MediaEval($T1$) to PolitiFact($T2$) and finally to GossipCop($T3$). Similarly, Table 5.7(b) reports the session in which GossipCop($T2$) arrives before PolitiFact($T3$). In both tables, transfer learning attains the worst results regarding average accuracy and forgetting. As for EWC and knowledge distillation, these achieve comparable accuracy values with a slight advantage of knowledge distillation. In terms of forgetting, however, knowledge distillation achieves the best performance, confirming the considerations made in the previous section. The only case that the continual-learning approaches give an inferior score compared to transfer learning is in the evaluation of the third training session ($T3$), but even there the difference is small (see the two bottom plots in Figure 5.3).

Method	Average Accuracy	Forgetting
Transfer learning	0.6520	0.2002
EWC	0.7294	0.0576
Knowledge Distillation	**0.7401**	**0.0475**

(a) $T1$: MediaEval, $T2$: PolitiFact, $T3$: GossipCop.

Method	Average Accuracy	Forgetting
Transfer learning	0.6314	0.2092
EWC	0.7151	0.0681
Knowledge Distillation	**0.7277**	**0.0399**

(b) $T1$: MediaEval, $T2$: GossipCop, $T3$: PolitiFact.

Table 5.7: Average accuracy (ACC) and forgetting (BWT) of the continual-leaning approaches on the three datasets. For ACC a higher value is better, for BWT a lower value is better.

It is interesting to note a detail that emerges both from the experiments presented in Table 5.7, as well as from those of Tables 5.6 and 5.5. We can observe a small difference in terms of performance in the order in which we train the model on the various tasks, which seems to suggest that it may have an effect at the model's capacity to generalize. Training on PolitiFactand then on GossipCopseems to improve performance in all experiments. This could be because the topics in GossipCopare very different from those in the other two datasets. Consequently, introducing this dataset in the second training session could have a negative effect on the third session. We leave the exploration of this phenomenon as future work.

5.2 Adversarial data poisoning for fake news detection

AI plays a crucial role in recognizing fake news online. Indeed, automated verification is indispensable in the fight against the dissemination of misleading content, especially in the context of large social platforms. In this scenario, detectors should be designed to continuously learn to classify recent news without affecting the performance obtained from previously acquired knowledge. As a consequence, online learning plays a crucial role in the design of such models [76]. In this constantly evolving framework, the doors are opened to new adversarial attacks capable of compromising detectors' performance on some news items.

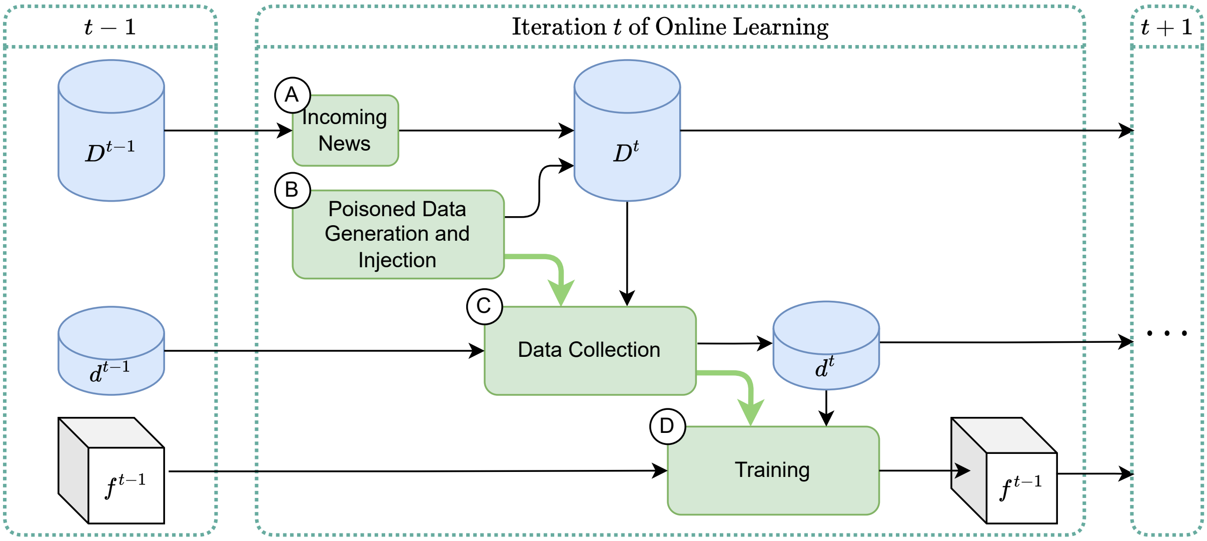


Figure 5.4: Generic iteration at time t of the iterative process of online learning with data poisoning. **A**: new incoming news adds up to the already existing ones ($D^{t-1} \rightarrow D^t$). **B**: poisoned data are generated and injected into the existing data. **C**: a subset of the data is collected and added to the data d^{t-1} collected at time $(t-1)$; the aggregated data are denoted as d^t. **D**: the model f^{t-1} is updated to f^t with the addition of the newly collected data.

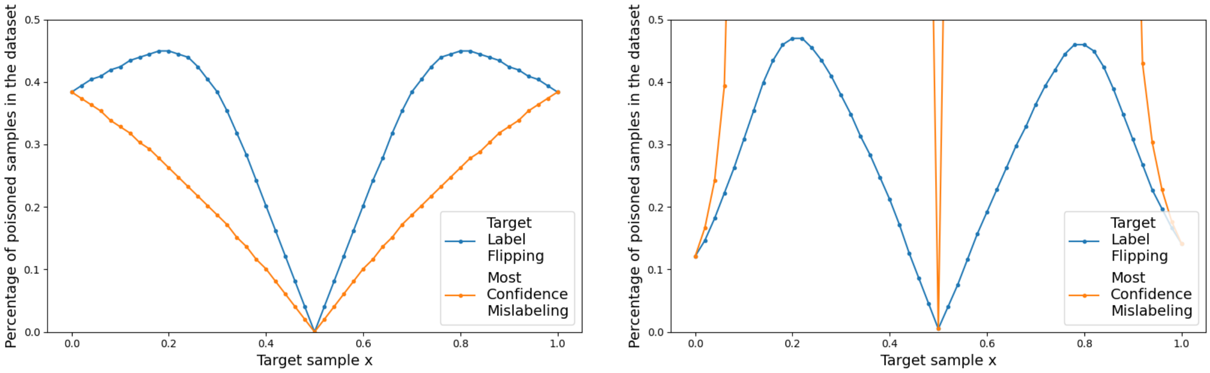
This preliminary study aims to explore the concept of adversarial data poisoning [76] in the context of online learning fake news detectors. Specifically, we investigate suitable methods to manipulate a model to ultimately misclassify a true news article as false *without* directly modifying the target article. This type of manipulation reflects the realistic scenario in which the attacker cannot control all the spreading news but can deliberately attack the detector to make it misclassify a specific news item by introducing new poisoned examples.

Existing research has primarily focused on modifying the target news articles [36, 230] to manipulate model behavior. This approach poses practical limitations and requires direct access to the articles. Moreover, while many studies concentrate on recognizing fake news in an offline scenario [341], Horne et al. [109] show that traditional content-based methods' performances slowly degrade over time, requiring periodic retraining which can be mitigated through online learning procedures. Despite the robustness of this learning method, a few studies have considered applying online learning to content-based fake news detection methods [59, 109].

Unlike previous works, by carefully selecting and incorporating adversarial examples into the training data, we seek to understand the vulnerabilities of the online learning model [107] and the potential risks associated with data poisoning attacks. By shifting the focus to modifying the training data without altering the target article, we aim to explore a more covert and scalable form of attack, highlighting the need for robust defenses against data poisoning techniques.

5.2.1 Online learning framework

In this section, we describe a realistic online learning framework in which poisoned data are injected into the training data to misguide a model's prediction. Figure 5.4 depicts the iterative process of an online learning fake news detector. As shown in the figure, at each time t, previously unseen news articles are combined with already existing ones; then, some poisoned data are generated and added to the collected data. Finally, the model is trained and updated using all the aggregated



(a) Linear LR

(b) Quadratic LR

Figure 5.5: Percentage of samples required to flip the target sample label, depending on its x value and the type of poisoning used. The Most Confidence Mislabeling attack requires a lower number of samples to make the Linear LR model to misclassify the target news article. Conversely, the Quadratic LR model is resilient to it, but more significantly affected by the Target Label-Flipping attack.

data. Since online learning involves the fake news detection model actively collecting news articles at different time instants, the model remains up-to-date and adapts to the ever-changing landscape of online information. Moreover, the framework not only encompasses the new data but also integrates historical data previously encountered.

Within this dynamic scenario, it is possible to analyze the effects of the deliberate introduction of poisoned samples aimed at challenging the model's ability to distinguish between genuine and fake news. To support this observation, in the next section, we report experimental results related to two types of data poisoning attacks against a logistic regression.

5.2.2 Data poisoning attacks

Within the online learning framework introduced in the previous section, we conduct preliminary experiments to analyze the effect of data poisoning attacks on Logistic Regression (LR) models, which are employed by many advanced techniques as the primary classifiers after feature extraction. Two LR models are considered:

- **Linear LR**: $logit(\hat{p}) = \beta_0 + \beta_1 x$
- **Quadratic LR**: $logit(\hat{p}) = \beta_0 + \beta_1 x + \beta_2 x^2$,

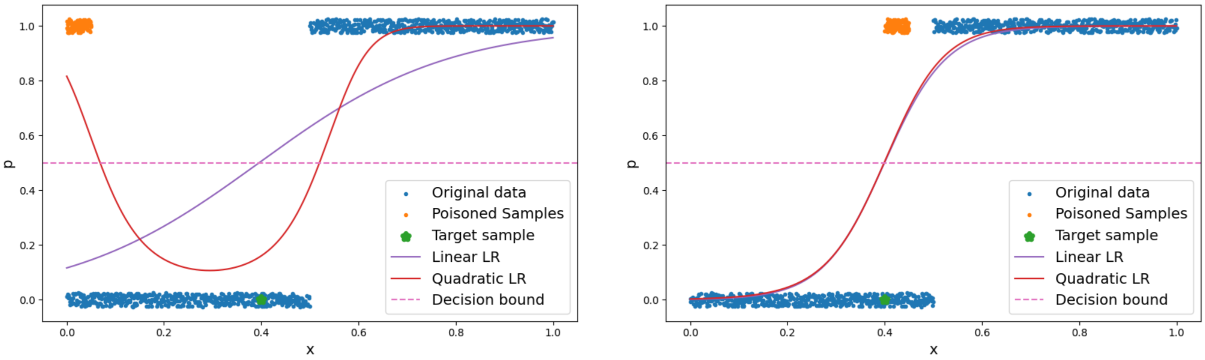
where $\hat{p}$ represents the probability of classifying sample x as fake news, $\beta_i \in \mathbb{R}\ \forall i \in \{0,1,2\}$ are the coefficients of the LR, and $logit(\hat{p}) = ln\left(\frac{\hat{p}}{1-\hat{p}}\right)$.

The examined attack strategy consists in the following data poisoning methods:

- *Most Confidence Mislabeling*: a sample that is confidently classified by the model is added to the training data with the flipped label.
- *Target Label Flipping*: a sample identical to the target sample except for the label, which is flipped, is added to the training data.

5.2.3 Results and discussion

To validate our theoretical framework, we propose experiments on synthetic data that allow us to show how these attacks can be performed on an online detector. In particular, these synthetic data will enable us to show how it is possible to manipulate the detector's behavior by appropriately modifying new training samples. The synthetic data used consists of real values $x \in [0, 1]$, with a binary class determined by the separation value $p = 0.5$. Figure 5.5 reports the comparison between the two poisoning attacks based on the number of samples required to misclassify the target sample. Figure 5.5(a) reveals that in the linear model, the Most Confidence Mislabeling attack requires a lower number of samples to misclassify the target news article. Conversely, Figure 5.5(b) shows that the Quadratic LR model is resilient to Most Confidence Mislabeling, but more significantly affected by the Target Label-Flipping attack.



(a) Most Confidence Mislabeling

(b) Target Label Flipping

Figure 5.6: x value of the original data, the poisoned data, and the target sample. The two Logistic Regression models trained on the poisoned data are also displayed. When subjected to the Target Label Flipping poisoning, both models slightly alter their predictions to misclassify the target sample. In contrast, the Power Model adapts to the Most Confidence Sampling poisoning while maintaining the correct classification of the target sample.

Figure 5.6 shows the x value of the original data distribution, the chosen target sample, and the poisoned data. The two LR models trained on the poisoned data are depicted as red and purple curves. Figure 5.6(a) shows that with the Most Confidence Mislabeling, the Quadratic LR model (the one with the most parameters) is able to follow the poisoned data and still correctly predict the target sample. This phenomenon occurs because the increased complexity of the Quadratic LR model allows it to adapt and capture patterns in the poisoned data, enabling accurate predictions of the target sample even in the presence of adversarial manipulation. In contrast, in Figure 5.6(b), both models shift their decision bound and incorrectly classify the target sample when applying Target Label Flipping.

These initial findings highlight the importance of model architecture and complexity in determining their vulnerability to specific types of adversarial attacks. The Linear LR model exhibits greater susceptibility to Most Confidence Mislabeling attacks, while the Quadratic LR model demonstrates resistance to this attack type but remains vulnerable to Target Label Flipping. These results provide valuable insights into the behavior of LR models under adversarial data poisoning attacks, laying the foundation for further exploration of more sophisticated models and defense mechanisms.

5.2.4 Future Work

In this study, we delve into the uncharted territory of adversarial data poisoning attacks within the context of fake news detection. The proposed method formalizes an online learning framework where an online learner is pushed toward misclassifying a true news article as false without any direct modification of the target article. Specifically, we introduced two types of data poisoning attacks, namely Most Confidence Mislabeling and Target Label Flipping, and we evaluated their impact on the performance of two logistic regression (LR). Results indicate that the susceptibility of the models varies on the basis of their complexity. It is important to remark that the effectiveness of data poisoning attacks may vary depending on the model architecture, the dataset used for training, and the robustness of the model.

This study represents only preliminary efforts so far. In the future, we plan to train and evaluate several fake news detection models (including deep neural networks) on real-world datasets which are commonly analyzed in the field (e.g., PolitiFact[5], Gossipcop[6], FakeNewsNet [261], Weibo21 [210], FbMultiLingMisinfo [24]). Furthermore, the inclusion of various sources and types of misinformation will enable us to assess the robustness and generalizability of the proposed method across different contexts and sources of information. Another possible research direction is the accurate analysis of a model's performance is subject to other possible data poisoning attacks. For example, if the attacker has access to the information about the model gradient, a sample that maximizes the gradient on the target sample might be added to the training (e.g. Gradient Maximization).

Finally, a range of traditional and deep learning models, which have shown promising performance in identifying fake news articles, can be considered. Possible examples of models include, among others, Support Vector Machines [178], Convolutional Neural Networks [146], Transformer-based Models, Graph neural networks [216]. A thorough examination of these models will be necessary to assess their susceptibility to data poisoning attacks.

[5] `https://www.politifact.com/`

[6] `https://www.gossipcop.com/`

Chapter 6

Forensic applications

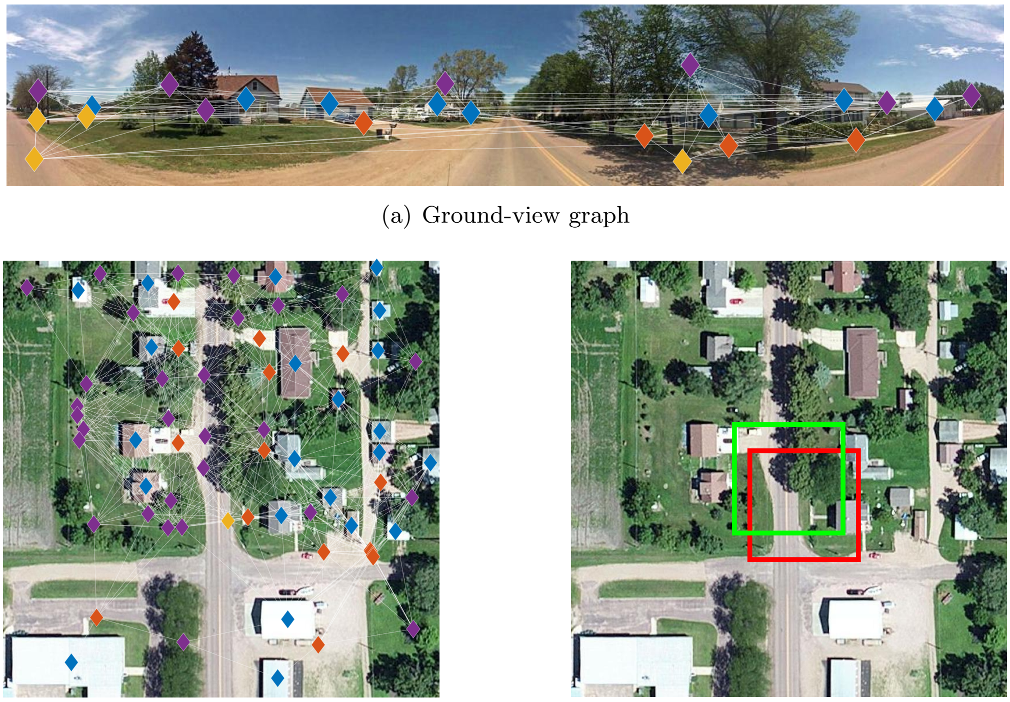
In the ever-evolving landscape of forensic applications, the marriage of cutting-edge technologies and investigative methodologies has paved the way for unprecedented advancements. This chapter delves into two pivotal forensic applications that have revolutionized journalistic investigations and insurance claim assessments: (1) ground-to-aerial matching and (2) an anti-fraud system designed to estimate image similarity of damaged cars within the insurance domain.

The application of forensic techniques in diverse environments, from crime scenes to journalistic investigations, requires careful consideration of interpretability. As investigators rely on the results of forensic tools, the ability to interpret and trust the results becomes critical. In this sense, in Chapter 6.1 we will introduce a ground-to-aerial matching technique based on an interpretable approach.

Simultaneously, the journey towards a deployment system introduces us to the delicate balance between sensitivity and specificity. The anti-fraud system's effectiveness is measured not only by its capacity to identify genuine instances of image similarity in damaged cars but also by its resilience against false alarms. A low rate of false positives is indispensable in preventing unwarranted accusations and ensuring the integrity of the investigative process. In Chapter 6.2 we will introduce a system that was developed for the automatic claim management process. Every time we report an accident to an insurance company, this process involves a series of checks aimed at verifying the vehicle's insurance coverage and finally identifying possible fraud attempts through which damage that has already been previously compensated is reported.

6.1 Ground to aerial viewpoint localization

Satellite images have become an essential investigative tool in many journalistic analyses. Whether verifying the authenticity of facts, reporting on conflict zones, or reconstructing the location of a specific event due to partial video evidence leaked online, satellite imagery is widely adopted in newsrooms. The BBC was one of the first to use satellite images to report on the internment camp system used to imprison Muslim minorities in China's Xinjiang region in 2017 [58]. As soon as the first evidence of these events became public, the regime immediately censored all relevant documents like social media posts. The story would have been quickly suppressed if it were not for satellite imagery. Since then, many other analyses have made it possible to reconstruct where several events like brutal murders took place [17] thanks to the *cross-view* matching between ground evidence and



(a) Ground-view graph

(b) Aerial-view graph

(c) Final matching

Figure 6.1: An example application of our method. Figures 6.1-(a) and (b) depict the aerial and ground-level images, respectively. Our proposed method extracts a graph representation of the two images and matches the graphs to localize the ground view over the aerial one. In (c), we show our predicted location (in red) and the corresponding ground truth (in green).

aerial images. Similar considerations could be made on the current war in Ukraine, where several crimes and political stories have been spread. Similar problems can be solved through forensics analysis of remote sensing images [1].

In this study, we examine the problem of automatic ground-to-aerial viewpoint localization from a *forensic perspective*. Although it is a very active area of research, many state-of-the-art techniques involve end-to-end deep learning-based methodologies that lack explainability. The lack of these characteristics represents a problem for media verification and authentication, as the matches obtained must be supported by clear and justifiable evidence [8]. Given these requirements, in this study, we propose to address the problem based on graph matching as shown in Figure 6.1. First, the analyzed images are segmented to identify the objects of interest present within them. Subsequently, each entity becomes a node in a graph connected to the others based on a covisibility window. This approach offers complete automation of the matching process without sacrificing the interpretation of the results, thus offering an advantage over deep learning-based models, which, although they have proven to be very accurate, are challenging to unfold and, therefore, hardly usable in investigations [95]. Differently from the previous methods that require high computational costs for explicit feature extraction, we model this problem via a probabilistic framework that matches graph representations of the image obtained from connecting the salient objects of the image.

6.1.1 Methodology

In this section, we present our pipeline for ground-to-aerial viewpoint localization. Our pipeline, shown in Figure 6.2, automates the entire task without sacrificing model explicability, making the

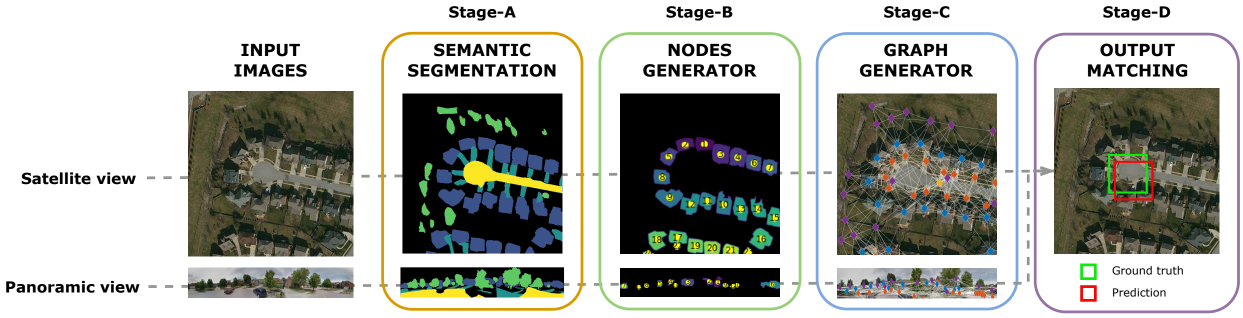
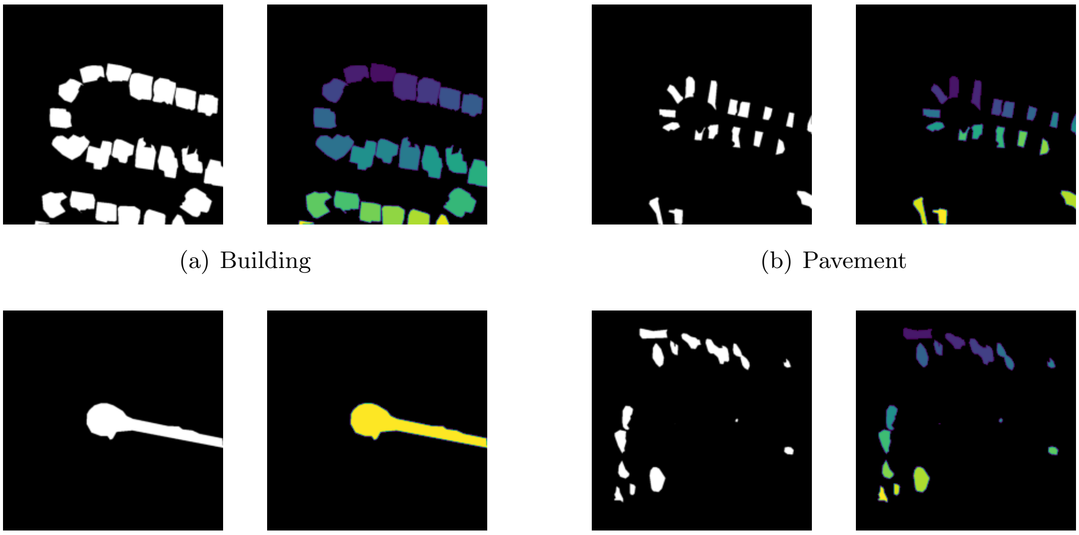


Figure 6.2: Overview of our proposed pipeline based on semantic graph for ground-to-aerial image matching. **Stage-A** depicts the semantic segmentation operation applied to the images taken from CVUSA dataset [305], both to a panoramic image and the corresponding aerial view. **Stage-B** represents the connected components algorithm and the corresponding nodes generator. **Stage-C** shows the graph generator for both panoramic and overhead images. Finally, **Stage-D** represents the final matching retrieval between the two points of view.

solution applicable in forensic investigations. We achieve this goal by building upon the previous landmarks graph matching technique introduced in Verde et al. [286]. Differently from this study, which required an extensive human labeling process, our proposed methodology automatically extracts the graph representation of the image. This graph representation has the advantage of being able to take into account image features that are not dependent on the viewpoint, therefore preserving the adjacency relationships among objects no matter the angle or point of view from where the image is taken. From here on, we introduce the stages of our pipeline, which is composed of four stages: *Stage-A* segments the image extracting the objects of interest in the satellite and panoramic images, *Stage-B* identifies the nodes that will compose the graph, *Stage-C* is used to generate the graph connecting all the nodes, and *Stage-D* computes the matches between the two-view images.

Stage-A: Semantic Segmentation. Our pipeline takes as input a panoramic ground viewpoint image and a wide aerial perspective from a satellite photograph. These images are initially processed with common AI-based semantic segmentation [311] techniques in order to extract a labeled image where each pixel corresponds to a given class. We select this approach with the aim of building a pipeline that is automated without the need to extract the significant landmarks a priori. To demonstrate the effectiveness of the proposed pipeline, we segment the image with respect to generic entities that can appear in any viewpoint image, such as *buildings*, *pavements*, *roads*, and *trees*. Figure 6.2 reports an example of the output of the semantic segmentation task, both for the satellite and the panoramic ground image. We report each label in the image with a different color, that is, (1) buildings are colored in blue, (2) pavements are depicted in light blue, (3) roads are yellow, and (4) trees are pictured in green. All the irrelevant elements are blacked out in the segmented image.

Stage-B: Node Generator. Once the objects in the images have been labeled, we want to convert the segmented image into a graph. In this stage, we represent each object in the image as a separate labeled node. To do so, the segmented image obtained from the previous step is filtered to handle one class at a time. Then, we convert the segmented objects into a binary mask where white pixels represent the relevant class, and black pixels depict the background, as illustrated in Figure 6.3. The binary images are finally analyzed by the Spaghetti labeling algorithm introduced in Bolelli et al. [30] using 4-way connectivity. An example of this process is shown in Figures 6.3(a)-(d). Each color represents a group of pixels belonging to the same connected component, i.e., different objects



(a) Building

(b) Pavement

(c) Road

(d) Tree

Figure 6.3: Example of connected component labeling algorithm applied to the binary mask (left side) of each label of a satellite image, i.e., building (a), pavement (b), road (c), and tree (d). On the right side of each label is illustrated the output of the connected component labeling algorithm: each color represents a connected component, i.e., different instances within a class are depicted in different colors.

within a class, are depicted in different colors. The same framework is applied to both aerial and ground images. The last step of this phase is the extraction of the centroid of each of the connected components. This point will represent the coordinates of the node.

Stage-C: Graph Generator. Now that nodes have been created, this stage connects them based on a *covisibility window* using the strategy proposed by Verde et al. [286]. The landmark graph is obtained by sliding a window through the image with a stride of one pixel. If two or more nodes are detected within the covisibility window, they are connected and stored as a *clique*. The covisibility window size is variable and adjusted based on the distance of the nodes with the aim of obtaining a connected graph, both for the aerial and the ground image. The output of this stage will be a labeled graph that represents all the spatial connections between the objects in the image.

Stage-D: Graph Matching. At this stage, we can finally compare the satellite image with the ground image by matching their graphs. Once the pictures are matched, the result gives an estimate of the location of the query ground panoramic image in the aerial view. The matching of the graphs is conducted by considering just cliques of the previous Stage-C that are considered *relevant* [286]; that is, the same clique in the aerial view must at least once occur on the query image. Given the drastic viewpoint change and possible element occlusions in the query image, traditional keypoint matching techniques would not work in this scenario, as some nodes visible in the satellite image may not be present in the ground image. Consequently, we cast the problem to a probabilistic framework by obtaining a collection of candidate locations $\mathcal{L}_x$ that represent different subgraphs of the satellite image and corresponding possible matches in the ground image $\mathcal{Z}$. Following Stumm et al. [275], we assume that the sparse normalized cross-correlation between location adjacency matrices represents the observation likelihood $P(\mathcal{Z}|\mathcal{L})$ of the ground query $\mathcal{Z}$ given a location $\mathcal{L}$ in the satellite image. Denoting the class adjacency matrix between classes i and j in $\mathcal{Z}$ and $\mathcal{L}$ by $W_{ij}^{\mathcal{Z}}$

and $W^{\mathcal{L}}_{ij}$, the likelihood is computed as shown in Equation 6.1.

$$P(\mathcal{Z}|\mathcal{L}) = \frac{\sum_{ij} W^{\mathcal{Z}}_{ij} \cdot W^{\mathcal{L}}_{ij}}{\sqrt{\sum_{ij} (W^{\mathcal{Z}}_{ij})^2 \cdot \sum_{ij} (W^{\mathcal{L}}_{ij})^2}} \tag{6.1}$$

At this point, we can apply Bayes' rule to derive the posterior probability of being in a location given the observation as follows in Equation 6.2.

$$P(\mathcal{L}_x|\mathcal{Z}) = \frac{P(\mathcal{Z}|\mathcal{L}_x)P(\mathcal{L}_x)}{P(\mathcal{Z}|\mathcal{L}_x)P(\mathcal{L}_x) + P(\mathcal{Z}|\overline{\mathcal{L}}_x)P(\overline{\mathcal{L}}_x)} \tag{6.2}$$

In conclusion, the candidate location that satisfies the maximum a posteriori (MAP) criterion in Equation 6.3 is the best possible matching for the present ground query.

$$\mathcal{L}_{MAP} = argmax_{\mathcal{L}_x} P(\mathcal{L}_x|\mathcal{Z}) \tag{6.3}$$

The output of the prediction of the matching pipeline is the covisibility window corresponding to the best candidate location $\mathcal{L}_{MAP}$ (i.e., the green square in Stage-D of Figure 6.2) to be compared to the ground truth (i.e., the red square in Stage-D of Figure 6.2). As said, the dimension of the covisibility window is variable, and as a consequence, the dimension of the bounding box of the prediction will vary accordingly. This aspect will be deepened in Chapter 6.1.2.

6.1.2 Results

In this section, we report our experiments and implementation details. All the pipeline has been implemented in Python. In particular, we use the OpenCV[1] and NetworkX [93] libraries for extracting the connected components and generating the nodes corresponding to each object as described in Chapter 6.1.1. We evaluate our pipeline on the CVUSA [305] dataset, which contains more than $44k$ image pairs taken from the aerial viewpoint at a resolution of 750×750 and from ground level at a resolution of 1232×224. The dataset is assembled by downloading images depicting locations in the United States from Google Street View and Flickr. In all our experiments, we evaluate viewpoint localization in terms of Intersection Over Union (IOU), which measures the alignment between the ground truth and predicted bounding boxes.

As mentioned in Chapter 6.1.1, we use a variable size covisibility window as in Verde et al. [286]; we evaluate our method with two other window sizes: 128×128, and 256×256. The ground truth bounding boxes are always placed at the center of the satellite image because of how the dataset was assembled.

Experiment 1. The first experiment investigates the trade-off between the variable window and a fixed strategy. We report, in Table 6.1, the top-1 and top-3 mean IOU with respect to the estimated windows. As can be seen from the reported values, the proposed solution introduces a 96.56% gain in terms of mean top-1 IOU when compared to the 256×256 fixed window and 390.37% when compared to the 128×128 one. Moreover, on the same settings, we achieve a top-3 accuracy improvement of 105.33% with respect to the fixed window 256×256 and 483.17% for the 128×128 window. Based on those findings, we assume that the use of a variable size window allows the

[1] `https://opencv.org/`

pipeline to better generalize over different scenarios and obtain more accurate matching.

Ground truth size	IOU top-1	IOU top-3
128x128	0.0948	0.0766
256x256	0.2366	0.2175
variable window	**0.4650**	**0.4467**

Table 6.1: Performances obtained with the proposed method varying the windows dimension.

Experiment 2. In the second experiment, we compare our proposed automated pipeline with respect to the interpretable method introduced by Verte et al. over the same test set considered in their paper, which is composed of 15 images extracted from the CVUSA dataset. We report in Table 6.2 the obtained results over different methodologies and window sizes. Based on the reported values, the proposed pipeline achieves an improvement on the top-1 and top-3 accuracy equal to 17.84% and 32.71%, respectively, with the variable window configuration. We can also notice that the proposed method obtains slightly worse results with respect to Verde et al. in the case of fixed windows. However, our method is fully automated and probably extracts better graph representations that lead to the best matching accuracy than the other method when it is considered a variable configuration for the covisibility window.

Ground truth size	Verde et al. [286]		Proposed method	
	IOU top-1	IOU top-3	IOU top-1	IOU top-3
128x128	0.2215	0.1058	0.1503	0.0527
256x256	0.2789	0.2057	0.2535	0.1147
variable window	0.2724	0.2305	**0.3210**	**0.3059**

Table 6.2: Quantitative evaluation of the proposed automated pipeline with respect to Verde et al. The best results are in bold.

Experiment 3. Our final experiment analyzes the robustness of the proposed pipeline with respect to every single object class for estimating their importance when constructing and matching the graphs. We report a graphical comparison in Figure 6.4, where the blue line shows the accuracy trend of our method while the dotted one reports the performances of Verde et al. [286] while evaluating the 15 image subset from CVUSA with the variable window setting. Figure 6.4 also reports the standard deviation of our method (in purple) compared to the one from Verde et al. (in red). Based on the reported values and previous findings, we can notice that our method outperforms the baseline when all four classes are present. Moreover, this experiment measures the impact of removing one single class at a time. From the figure, we can observe that removing the *pavement* or *tree* classes can lead to a performance gain with respect to operating on all four classes. Therefore, when using a set of three classes, our proposed pipeline achieves an average IOU improvement equal to 22.15% with respect to the method from Verde et al.

6.2 Car damage reidentification

The insurance industry consists of thousands of companies, which collect over one trillion dollars in premiums every year. The massive size of the industry contributes significantly to the cost of insurance fraud by providing more opportunities and more incentives to commit illegal activities.

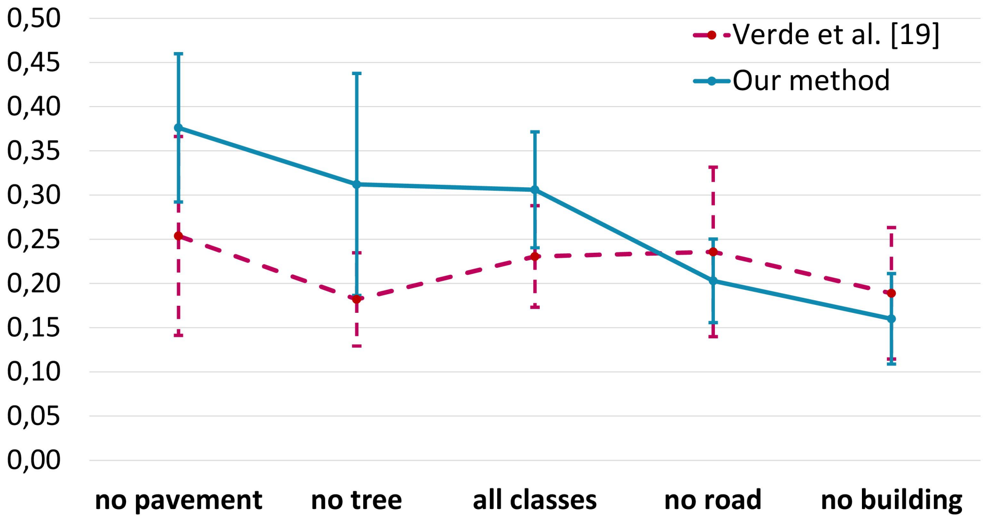
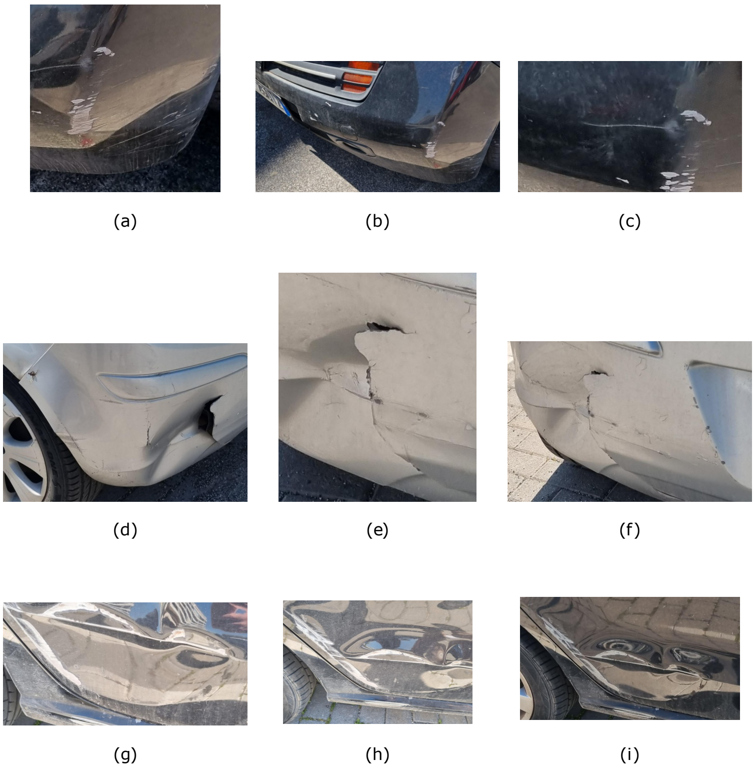


Figure 6.4: Change in IOU in terms of mean and standard deviation when considering a subset of only three classes. As you can see, removing floors and trees improves performance.

Indeed, the annual cost of vehicle insurance fraud is estimated to exceed 40 billion dollars [79]. Added to this, insurance fraud is often used to fund the wider activities of criminal gangs, which may be linked to serious organized crime such as drug dealing, burglary, or terrorism [18]. For this reason, insurance companies have developed processes to detect, disrupt, and prosecute people who try to fabricate a claim. Advanced analytics software helps insurers proactively identify cross-industry patterns and alert the industry to fraudulent networks. This work is in this direction: We introduce an end-to-end pipeline designed to detect automotive damage fraud.

Insurance companies process a very large amount of images every day. Customers who make a claim for car damage are required to upload several photographs of the involved vehicle, which allow the insurance company to examine the damage as well as the vehicle as a whole. These images include photos of the exterior or interior of the vehicle, the insured's documents, the vehicle registration document, details of the damage, the license plate, and the car's vehicle identification number (VIN). These images flow into the claim management process, through which the insurance experts manually inspect the correspondence between the claim reported and the information present in the images. This process requires extracting a significant number of information from the images, such as the correspondence of the vehicle in the image with the insured one, the verification of the license plate number [37, 68, 74, 125, 126, 183, 194, 313], the VIN, the color of the car, or the presence of damages on the bodywork [23, 73, 155, 224]. Extracting manually all these data requires a very large amount of time and effort and has a significant effect on the costs incurred by the insurance companies. Added to this is the need to deal with increasingly sophisticated attempts at fraud [18, 79]. In some cases, previously reported damages are reproposed to the same insurance company to obtain new compensation. A first idea to address this problem could be to verify that the vehicle analyzed has not suffered the exact same damage in the past; yet this is not always sufficient: In fact, in the most sophisticated cases, the damaged bodywork component is removed from the vehicle and reassembled on another car! Thus, to identify this type of fraud, requires to inspect the damages and compare damages among different vehicles. Of course, the adversary can even change the damage by scratching more or hitting the already damaged part, which makes this



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6.5: Example of damage similarities. Triplets (a)-(b)-(c), (d)-(e)-(f), and (g)-(h)-(i) are real matches of the same damages. Each row shows an example type of damage: *scratch*, *crack*, and *dent*, respectively.

problem even harder to solve. Identifying these cases among the millions of images processed every year is an extremely complex task to automate because of the enormous heterogeneity of the collected data. Different damages of the same type can have different shapes, sizes, and colors. Added to this, reflections or dirt on the bodywork can make it even more difficult to identify them. In this study, we introduce a new pipeline, which is designed to support the experts to automatically identifying possible fraud attempts. Figure 6.5 shows some examples of this problem for three types of damage.

Bodywork damage can be classified in various ways according to its severity. In the worst cases, an accident can lead to the destruction and deformation of a substantial part of the bodywork. In less severe cases, the damage may simply be limited to a *scratch*, *dent*, or *crack*. This second category of damages is certainly the most widespread and the most easy to apply insurance fraud; for this reason in this study we narrow the attention to these categories of damage. However, recognizing them can be very complex. Each damage can have a very different shape and size from any other. An additional complication arises from the necessity to be able to recognize these damages in spite of reflections, light conditions, partial occlusions, zoom-level, blurring, or dirt on the bodywork. The problem becomes even harder because of the need to find the same damage among millions of images, in which (even worse) the photograph of the damage may have been taken from a different angle and under different environmental conditions (lighting, background, etc.). Unfortunately, unlike other tasks such as person reidentification, this problem has been little addressed on cars because of the scarce availability of open data available to explore new possible solutions.

The main contributions of this study can be summarized as follows:

- We introduce a new benchmark dataset for the recognition of similar damages; with this, we hope to stimulate discussion on this type of problem and to make a common dataset available to the community to evaluate the proposed solutions.
- We propose an end-to-end pipeline for *damage similarity* detection. As far as we know, this is the first work that proposes to investigate the possibility of recognizing this type of fraud with a pipeline that manages the entire process from image acquisition to signaling of possible similar damage.
- We discuss the difficulties encountered in scaling these solutions in a real-world setting.

In detail, our proposed system is structured in the following different phases: images sent by policyholders are initially filtered to select only those containing the exterior of the vehicle. The car is then detected within the image. At this point, the system classifies the color and brand of the vehicle and localizes the damages present on the car. Finally, the identified damages are mapped within an embedding and compared with those of the claims already analyzed in the past, filtering the possible matches with respect to the color, brand and view of the vehicle. This filtering is intended to reduce the number of comparisons to be made and reduce the possible number of incorrect matches.

6.2.1 Approach

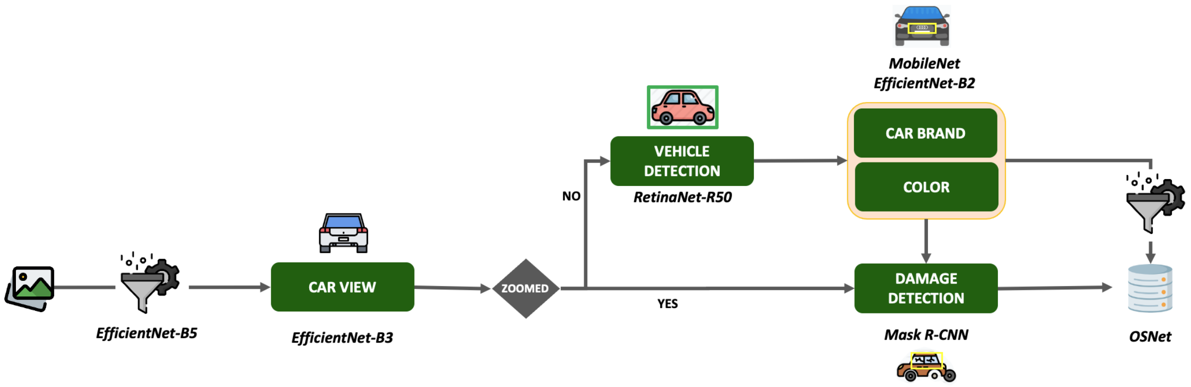


Figure 6.6: Our proposed pipeline. Images are first filtered to select images that capture the exterior of the vehicle through an EfficientNet-B5 and then, based on the view of the vehicle, that is, the sides of the car visible in the image., with an EfficientNet-B3. Zoomed images that only capture details of the damage do not allow to extract vehicle information because the zoom on the damage makes it difficult, if not impossible, to extract information about the car or the location of the damage on the bodywork; therefore they are immediately sent to the damage detector module. If the image depicts the entire vehicle, we detect the car with a RetinaNet-R50, we extract the brand and color of the vehicle with an EfficientNet-B2 and MobileNet, respectively, and we locate the damage over the car bodywork with a Mask R-CNN. Images are then filtered based on these information and are finally sent to the damage reidentification module (i.e., OSNet).

The images sent to insurance can be very different from each other. Some of these represent documents, other details of the bodywork or mechanical parts, and others could portray images of the interior or exterior of the vehicle. In addition to this, the data collected in different countries may have biases that differentiate them from those of others. To manage this complexity in a real system, we introduce an end-to-end pipeline for recognizing similar damages in a large gallery of collected images. Our system, illustrated in Figure 6.6, is based on five main steps. Initially, images entering the pipeline are filtered to select only images of the exterior of the car. Then, the car view classification module classifies the sides of the car visible in the image, that is, front, back, front–left, back–right, and so on. Based on this classification, the zoomed images are directly sent to the damage recognition module, whereas the other images are used to extract the branding and color of the car. This information is useful for filtering the matches to be verified within the image database. The final module of the pipeline selects images of vehicles with damage similar to that of the newly uploaded images. Damages recovered from the system that exceed a certain similarity threshold are selected as potential copies of the damage and are then reported to the claim experts.

In the remainder of this section, we describe these components. We begin with vehicle information extraction, which is used to reduce the number of damages to compare. Then, we discuss the damage detection and localization module, and, finally, we introduce our damage reidentification system.

Damage Localization and Claim-Level Feature Aggregation

We integrate our damage reidentification system into a pipeline that deals with the identification of damages and the extraction of basic information on the claim under analysis. The first part of the pipeline deals with filtering the images of the claims, selecting only those that portray the vehicle from the outside, as the damages that we analyze in this study are damages that can only be found on the vehicle bodywork. To do this we use an EfficientNet-B5 [279] trained on 11 classes (Documents, Odometer, Exterior, Interior, Mechanical Parts, Disassembled Parts, Display, three VIN classes, and Others) to select only images of the *exterior* (i.e., the Exterior class) of the vehicle. Next, we classify the vehicle view with an EfficientNet-B3 [279] trained in 9 classes (Back, Back–Left, Back–Right, Left, Right, Front, Front–Left, Front–Right, and Zoomed). These two steps allow us to select only images of the exterior of the vehicle and distinguish different views of the bodywork of the car. These two steps are then followed by the extraction of the vehicle information and the recognition of the damage.

Vehicle information. With millions of new claims open every year, insurance companies collect a significant number of images. All this translates into having to compare each new damage with millions of other damages present in the database. In one year, this would mean making millions or billions of comparisons, which would be computationally prohibitively expensive. To reduce the complexity of these comparisons, we propose a pipeline that extracts various information about the vehicle, which we use to reduce the search space: Although it is possible to perform fraud by reassembling a panel of one car on another one, or to claim damage on the same vehicle, these require that the target vehicle has the same *model* and *color* with the original one; thus the research can be limited to vehicles of the same brand, model, and color. of the same *model* and *color*. To automate this process, we propose the pipeline in Figure 6.6. After the images have been filtered and classified according to the view of the vehicle, a RetinaNet-R50 [172] extracts the bounding boxes of the vehicle and the car's brand logo. These two pieces of information are then used to classify the brand and color of the car. For these two steps, we employ an EfficientNet-B2 [279] for brand classification and a MobileNet [112] for color classification. The aforementioned models are all trained with *cross entropy* loss except for RetinaNet which is trained with the *focal loss* function introduced in [172].

Damage detection and localization. Pictures of a claim must typically include both close-up shots of the damage and images with a distant view that contains the entire body of the car. These different perspectives allow on one hand the identification of the insured car and on the other hand, precise localization of the damage that has been reported. Identifying damage in an image is a key part of our pipeline as the entire reidentification system needs accurate damage identification to find any fraud attempts. However, it is important to note that this task is more challenging than traditional object detection problems, as the damage can have very different characteristics. In this study, we focus on the three most common damages: *scratches*, *dents*, and *cracks*. We treat this problem as a segmentation problem, in which we want to reconstruct the segmentation mask of the damage and its corresponding class. Mask R-CNN has proved extremely robust and accurate for this type of applications; therefore, we adopted it in our system with a ResNext-101 [312] *backbone* and we train this model to detect damages. Formally, we optimize the model parameters on each sampled *region of interest* with respect to the multitask loss introduced in [100].

$$\mathcal{L} = L_{cls} + L_{bbox} + L_{mask}$$

The classification loss L_{cls} and the bounding box regression loss L_{bbox} are the same from the Faster R-CNN [238] architecture, whereas the mask component is the average binary cross-entropy loss used in the standard Mask R-CNN implementation [100].

Accurate damage detection is not enough in a production environment. Once the damage of a vehicle has been identified, claim experts usually need to verify that the damaged area corresponds to the one reported. Having this information not only allows us to have a more precise location of the damage but also allows us to exclude false matches with damages located in other positions of the car bodywork. Indeed, the same damage, even if disguised or reassembled on a new vehicle, will always be found on the same panel (component), which allows to exclude many other possible pairs. Thanks to the vehicle-view classification module, we can identify the location of the damage on one of the sides of the vehicle and thus reduce the possible matches to be identified in the database.

Claim-level aggregation. Although it is possible to train very robust deep-learning models, these will still be subject to some, albeit small, error rate. However, an error in the first part of the pipeline risks affecting the subsequent damage reidentification module. To reduce such errors, after an insured opens a claim for compensation and sends the images of the vehicle, we perform a *claim-level* refinement: Given two or more images belonging to the same claim, we select the brand that is predicted with higher confidence by the brand model across all images. Formally, given a set of brand predictions $B = \{f_b(x_1), \ldots, f_b(x_m)\}$ for colored images $x_i \in \mathbb{R}^{H \times W \times 3}$, we take:

$$\arg\max_{f} |\{f_b(x_i) \in B \mid x_i \in \mathit{claim}\}| \tag{6.4}$$

for any $m \geq 2$, where m represents the cardinality of the claim, and $f_b(x_i)$ represents the output confidence of the brand model for image i. The $\arg\max$ operation selects the brand model that obtained the higher confidence across all the m images of the claim.

Furthermore, we apply the claim-level logic for the classification of the color of the car as well. At the image level, we apply a threshold τ to select the color predicted by the model. Below this confidence, we also consider the second class with the highest score, that is:

$$f_c(x_i) = \begin{cases} C_k, & \text{if } C_k \geq \tau. \\ (C_k, C_{k-1}), & \text{otherwise.} \end{cases} \tag{6.5}$$

for any ordered prediction $\{C_1, \ldots, C_k\}$ where C_k is the highest value from the softmax function. Finally, given a set of color predictions $C = \{f_c(x_1), \ldots, f_c(x_m)\}$ on the images of the claim, the most voted color is chosen as the color of the car. Formally:

$$\arg\max_f |\{f_c(x_i) \in C \mid x_i \in \mathit{claim}\}| \ . \tag{6.6}$$

This phase of aggregation of the claims allows for the extraction of color and model information directly from the images, thus verifying that the information on the reported vehicle is consistent with the data of the insured vehicle and, finally, allows, as explained, to reduce the number of necessary comparisons within the database.

Damage Reidentification

Damages identified by the damage localization and vehicle information modules are ready to be reidentified via the damage-similarity module. Different shots of the same damage can be very different. The same damage can be captured in different lighting conditions, reflections, zoom levels, partial obstructions, and perspectives. The damage reidentification module must, therefore be robust to all these variables. We chose to build our reidentification module on top of the OSNet of Zhou et al. [335–337], which proved to be very effective for people's reidentification task. In our setup, the OSNet works as a feature extractor that maps input damage into an *embedding* space. Hence, the damages are directly compared in this space through *cosine similarity*. Unlike [167] who propose to add global image features that represent the vehicle's color and view, in our case we use the car view, color, and brand information to reduce the number of required comparisons. This allows us on the one hand to significantly reduce the time required to reidentify the damage and on the other hand to eliminate some biases such as the vehicle's view or color from the vector representation of the damage. Considering the heterogeneity of the photos of the same damage, many photos may contain information that is not useful for the reidentification of the damage. Therefore, we have chosen to crop the image around the damage to only include this information when comparing two damages. This allows us to eliminate unwanted noise and focus solely on the areas of interest.

The goal of our damage reidentification module is to learn an embedding function $f_\theta(x) : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^E$ that maps semantically similar points from the data manifold in $\mathbb{R}^{H\times W\times 3}$ to close points in $\mathbb{R}^E$ and different points in $\mathbb{R}^{H\times W\times 3}$ to distant points in $\mathbb{R}^E$. Formally, let $D(f_\theta(x_i), f_\theta(x_j)) : \mathbb{R}^E \times \mathbb{R}^E \to \mathbb{R}$ be a metric function measuring distances in the embedding space. We train our model to minimize the *hard triplet loss* [104] function, which, for each sample in a batch, selects the hardest positive and the hardest negative samples *within the batch.* We create the batches by randomly sampling D different damages, and again randomly sampling K images of the same damage, thus obtaining a batch of DK images. Then, we train the model to minimize the following loss function for each batch X:

$$\mathcal{L}_B(\theta, X) = \sum_{i=1}^{D}\sum_{a=1}^{K}\Big[m + \overbrace{\max_{p=1\dots K} D(f_\theta(x_a^i), f_\theta(x_p^i))}^{\text{hardest positive}} - \underbrace{\min_{\substack{j=1\dots D\\ n=1\dots K\\ i\neq j}} D(f_\theta(x_a^i), f_\theta(x_n^j))}_{\text{hardest negative}}\Big]_+, \tag{6.7}$$

where a data point x_j^i corresponds to the ith image of the jth damage in the batch. This loss ensures that, given an *anchor* point x_a, the projection of a positive point x_p representing the same damage j is closer to the anchor's projection than that of a negative point x_n belonging to another class d, by at least a margin m. The margin guarantees that in the end, points that are sufficiently close to each other will end up belonging to the same cluster, representing multiple copies of the same damage.

We choose the cosine similarity as our distance metric:

$$D(f_\theta(x^i), f_\theta(x^j)) = \frac{f_\theta(x^i) \cdot f_\theta(x^j)}{\|f_\theta(x^i)\|\|f_\theta(x^j)\|}$$

and use the following similarity score:

$$S(f_\theta(x^i), f_\theta(x^j)) = \begin{cases} 0, & \text{if } D(f_\theta(x^i), f_\theta(x^j)) < \zeta, \\ 1, & \text{otherwise.} \end{cases} \tag{6.8}$$

Then two damages x^i and x^j are considered the same damage if $S(f_\theta(x^i), f_\theta(x^j)) = 0$.

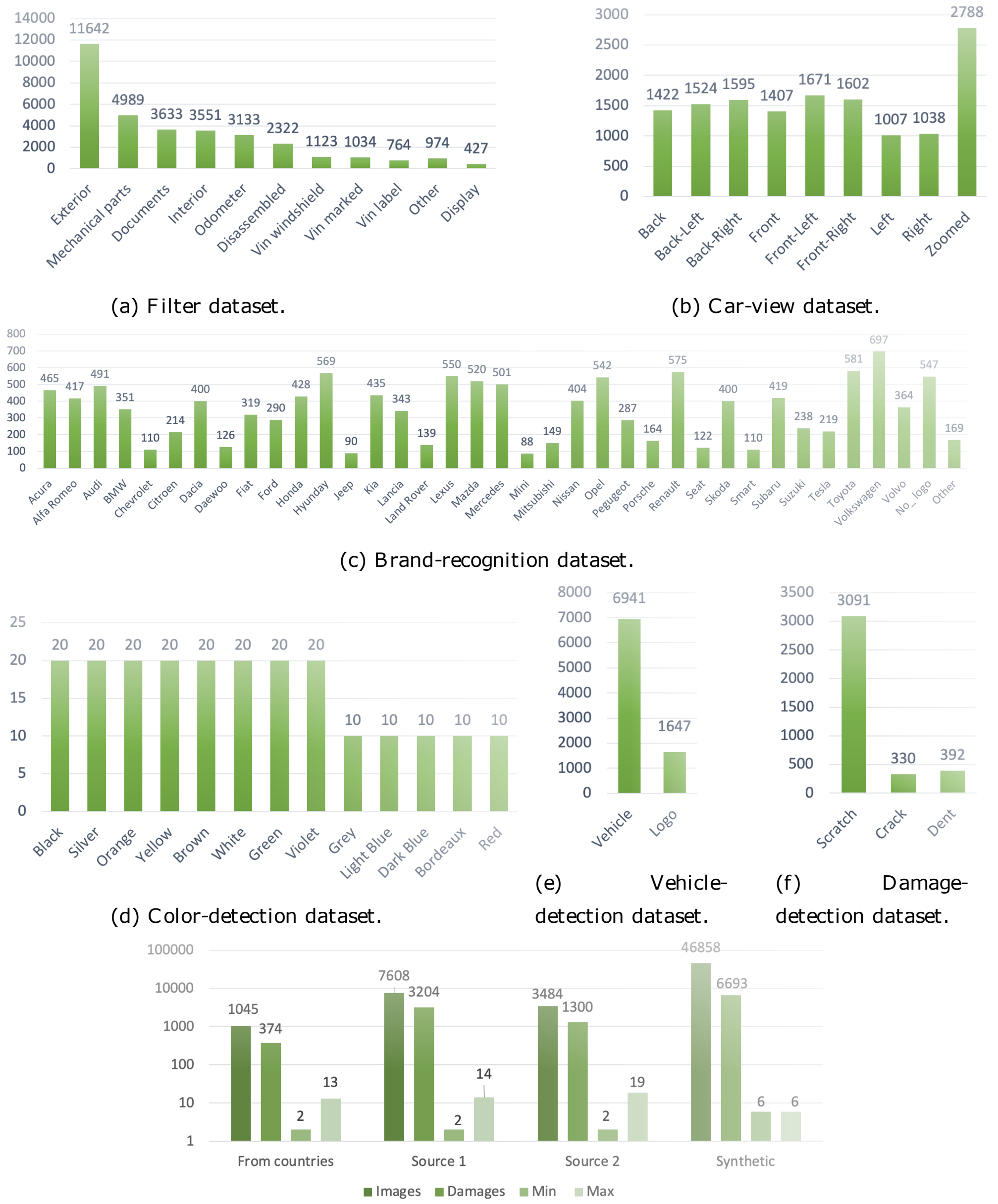
6.2.2 Implementation Details

All the experiments that we present in the next section were conducted on an Azure Standard NCasT4-v3 series virtual machine with a 16GB NVIDIA T4. Next, we provide some implementation details on the models, datasets, and our evaluation metrics.

We trained the Filter, the car-view and the vehicle-detection models on 684 × 684 images with the Adam [148] optimizer and a learning rate set to 0.0001. We trained the models to minimize the cross-entropy loss on batches of 4 images. Instead, we trained the brand and the color classification models with the same configuration but on batches of 32 images of size 224 × 224 pixels. We trained the damage modules on batches of 4 images with basic Detectron2 [306] configuration. Finally, we trained the damage similarity module on 256 × 256 images with a larger batch of 64 images and a basic learning rate set to 0.0003. We projected the images onto a 512–dimensional embedding space and, after performing experiments, we chose $m = 0.3$, $\tau = 0.5$, and $\zeta = 0.5$ for Equations (6.4), (6.5) and (6.8) respectively.

Datasets

The training and testing of all pipeline components that we have introduced so far require tackling several tasks. In industrial applications, data preparation requires a major effort to be able to



(a) Filter dataset.

(b) Car-view dataset.

(c) Brand-recognition dataset.

(d) Color-detection dataset.

(e) Vehicle-detection dataset.

(f) Damage-detection dataset.

(g) Damage-reidentification dataset.

Figure 6.7: Distribution of the training and evaluation datasets used for the components of the pipeline. For each dataset, we show the classes and corresponding number of samples per class. For the damage reidentification dataset (Figure 6.7(g) we report several sources used to construct the dataset, and the minimum (Min) and the maximum (Max) number of matches for each image.

structure datasets useful for model training. For this reason, we have decided to dedicate this section of the study to the description of the datasets built to train the various components of the pipeline introduced so far. Unfortunately, we cannot release all the data used for each component as they are subject to privacy and covered by trade secrets, but we will describe them in detail and report their characteristics summarized in Figure 6.7. In addition, to facilitate reproducibility of results, we also introduce a new set of tests for the damage reidentification task. The test set and the proposed models will be released publicly. For all datasets, unless otherwise specified, we use 90% of the data for training and the remaining 10% for testing. Finally, because we propose a supervised method for damage reidentification, we use the COCO Annotator [32] tool to annotate the damages and components of the car in the images.

Damage detection and localization. To identify and locate the damage to the vehicle we have introduced two components. The first deals with classifying the vehicle view and the second identifies the damage. For the former, we built a dataset of 14,054 images. Each of these has been annotated to classify the following 9 views: back, back–left, back–right, left, right, front, front–left, front–right, and zoomed. Figure 6.7(b) shows the distribution of each element. For the second task, we created a dataset of 3,818 examples of scratches, cracks, and dents. As shown in Figure 6.7(f), scratches are numerically much more frequent. Consequently, to balance the number of samples across all the classes, in the training phase, we increase the crack and dent by applying a data augmentation strategy by introducing random flipping, rotation, saturation, contrast, and brightness. Because the damage can only be present in photos of vehicle exteriors in Figure 6.7(a) we report the distribution of the dataset used to train the filter. The dataset contains many other classes than External, which are used for other purposes that are beyond the scope of this study.

Vehicle information. The detection and extraction of the vehicle and its basic information such as the brand and color is another fundamental step of the reidentification system. The first stage, therefore, requires vehicle detection. To this end, the model was trained on a dataset of 2,036 images. Instead, the color detection model is trained on a dataset of 210 images containing 13 colors. Finally, a dataset of 36 brands with a total of 37,000 images was created for the brand identification task. For this last dataset, we use 80% of the data for training and the remaining 20% for testing. Figures 6.7(e), 6.7(d) and 6.7(c) show the statistics of these datasets.

Damage reidentification. To train our damage reidentification model, we built a dataset, shown in Figure 6.7(g), of 57,950 images. Of these, we use 90% for training and the remaining 10% for validation. This dataset contains 11,571 possible matches and is constructed from several sources that include images from real claims (marked as *From countries* in Figure 6.7(g)), two internal datasets of damages (labeled as Source 1 and 2), and a synthetic dataset that was constructed as follows: First, we manually extract 74 real damages from images of damaged vehicles using the GIMP software [281]. The damages were extracted with a transparent background to remove details of the original bodywork. Then, for each damage, we automatically paste it on a car identified through our vehicle detection system. For each vehicle image, we create 7 different versions of the damage by applying it in 5 different positions and creating a perspective change and an affine transformation of the damage.

We evaluate the model on two test sets. The first contains 385 images with a total of 139 unique damages with at least 2 matches per damage and up to 7 matches. The second,[2] contains 567 images with 420 possible matches with at least 38 matches for each damage. The public test set was collected by acquiring images of 42 vehicles with five different smartphones. The images contain zoomed and non zoomed views of the vehicle and were captured in different lighting, dirt, and reflection conditions to simulate the images sent by policyholders as realistically as possible.

Evaluation Metrics for Damage Reidentification

We conclude this section with a brief discussion on the evaluation metrics adopted to evaluate the reidentification system. The task is a reidentification problem, and as such, we use the most commonly adopted metrics to evaluate these tasks [316]. In addition, we add two metrics that we used to evaluate the system and which proved useful for scaling the system into production. We call them *recall-at-top-k* ($recall_k$) and *F1-at-top-k* ($F1_k$).

The first reference metric is the *cumulative matching characteristics* (CMC). CMC-k (also known as Rank-k matching accuracy [298]) denotes the likelihood that a correct match will appear in the top-k ranked retrieved results. Because it only examines the first match in the assessment process, CMC is accurate when only one ground truth exists for each query. However, in a real setting, the image database typically comprises many ground truths, so CMC cannot completely reflect a model's discriminability across numerous matches. Therefore, we use the Mean Average Precision (mAP, [331]). It assesses average retrieval performance with numerous ground facts. Originally, it was frequently used in image retrieval. It can address the issue of two systems doing equally well in searching for the first ground truth but having varied retrieval abilities for additional challenging matches in reidentification evaluation.

Finally, we use $recall_k$. Because we are interested in finding the highest number of matches, the recall allows us to measure the number of matches correctly identified against the total number of possible matches. However, wanting to calculate this value with respect to the top-k, we apply a change to the recall.

$$recall_k = \frac{\text{TP in the top-}k\text{ results for each query}}{\text{Max TP}}$$

where

$$\text{Max TP} = Q - \sum_{q=1}^{Q} max\{TP_q - k, 0\}$$

with TP_q representing the true positives (TP) of the actual query q and Q representing the number of queries. From this definition, we can finally define the $F1_k$ as follows.

$$F1_k = 2\frac{precision \cdot recall_k}{precision + recall_k} \tag{6.9}$$

6.2.3 Results

In this section, we evaluate the performance of the proposed pipeline. Before discussing the performance of the reidentification model, shown in Tables 6.4 and 6.5, we begin by reporting the performance of the vehicle information and damage detection and localization components that

[2] URL will appear here after publication.

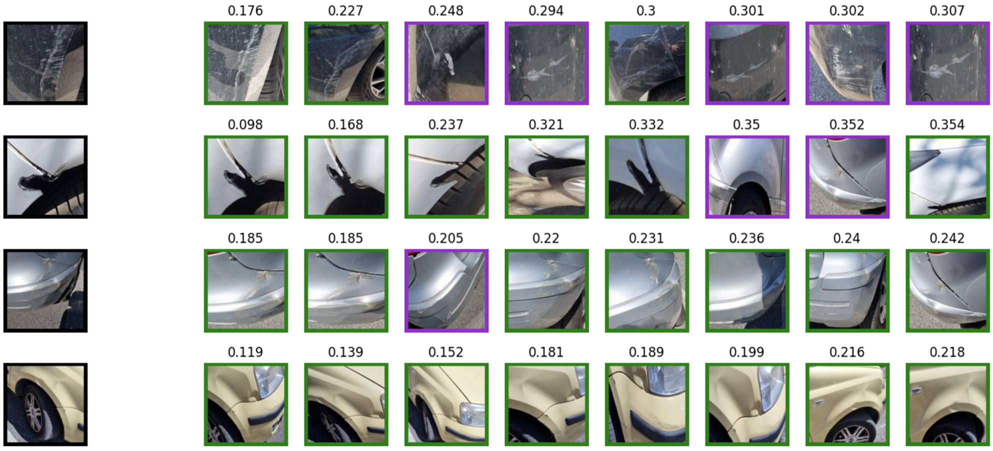


Figure 6.8: Sample queries and matching images in the public test set. The leftmost images are query images, followed by the highest similarity matches. Images framed in green are correct reidentifications, whereas those in purple indicate errors. The similarity score is also shown above each image. Even in cases of error, the model identifies images that are very similar to the query one.

contribute to the functioning of the damage reidentification system. Finally, we conclude by reporting the performance of the damage reidentification model and presenting the challenges faced in scaling the model in a production pipeline.

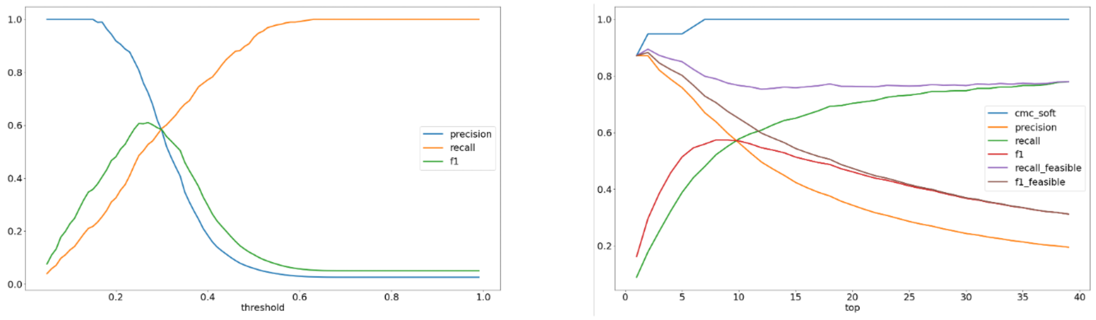
Vehicle Detection and Localization

Extracting basic vehicle information allows us to reidentify damage more accurately. Although it is interesting to reduce the error rate of the reidentification system as much as possible, *an error rate of 1% on 2 million images still implies a very high number of false alarms.* In addition, it leads to a very high number of images to compare. As explained, however, it is possible to reduce the complexity simply by removing all possible unnecessary pairs by extracting vehicle information. First, the vehicle identification pattern introduced in Section 6.2.1 achieves 85.0% accuracy. Once the vehicle has been localized, we identify the color and brand of the car. The brand recognition model accurately classifies 98.0% of the images analyzed, and the color model achieves 87.0% accuracy.

Damage Detection

The damage detector is a key component of our pipeline, as damage reidentification would not be effective without accurate damage recognition. As explained previously, in this study we focus on *cracks*, *dents*, and *scratches*. However, we are not interested in the classification of damage, that is, the correct identification and classification of the damage as a crack, dent, or scratch, but we limit the analysis to recognizing each of these classes as damage. In spite of this, we still consider the classification task to be very interesting and at the same time complex; we, therefore, leave the possibility to investigate this problem in the future.

Unlike many other object detection tasks, damage can be very heterogeneous, with different shapes, colors and sizes. Especially in some conditions of light and reflection on the bodywork (see Figure 6.5), recognizing some types of damage (especially the dents) can be very complex even for the most experienced claim experts. For this, proper tuning of the model hyperparameters can help to significantly improve performance. In our experiments, we compared three different update



(a) Precision–recall curve and F1 at different thresholds.

(b) The different evaluated metrics at top-k.

Figure 6.9: Evaluation metrics of the damage reidentification model on the public test set. Figure 6.9(a) shows the precision–recall curve and F1 score obtained at different thresholds. Figure 6.9(b) shows the variation of the performances with respect to the top-k predictions for different values of k.

configurations of learning rate. Table 6.3 shows the comparison in terms of IoU and mAP between three learning rate strategies: (1) *Step LR*, the learning speed of each group of gamma parameters decays at each step size epoch, (2) *Cosine LR*, setting the learning speed of each parameter group using a cosine annealing program, and (3) *Aug. Cosine LR*, the Cosine LR scheduler with data augmentation. From these experiments, the Cosine LR obtains better performances than the Step LR, and the data augmentation further contributes to improving the overall robustness of the model.

Ultimately, recognizing the damage is not sufficient to understand where it is located on the vehicle's body. For this, we also report the performance of the filter and car view models. The first means that the damage detection model receives in input only images of the exterior of the vehicle which can therefore contain damages. This model achieves 96.0% of accuracy. The car view model allows to identify the location of the damage on the vehicle body and to reduce the comparisons necessary to identify possible matches. In this case, the model achieves 90.0% of accuracy.

LR strategy	**IoU**	**mAP**
Step LR	10.0%	64.7%
	25.0%	45.0%
Cosine LR	10.0%	67.6%
	25.0%	51.9%
Aug. Cosine LR	10.0%	**69.6%**
	25.0%	51.9%

Table 6.3: Damage detection. model trained with different learning rate (LR) strategies.

Damage Reidentification

The inspection of the damage similarity is the final step in our pipeline. Damages and information extracted from previous modules can be used to identify possible fraud attempts. In this section, we present the experiments that we conducted on the model, and in Section 6.2.4 we show the performance of the model when aggregated with information extracted from the other components of the pipeline. Figure 6.8 shows some reidentification examples of our system.

To evaluate the performance of the proposed solution, we compare it with two models commonly

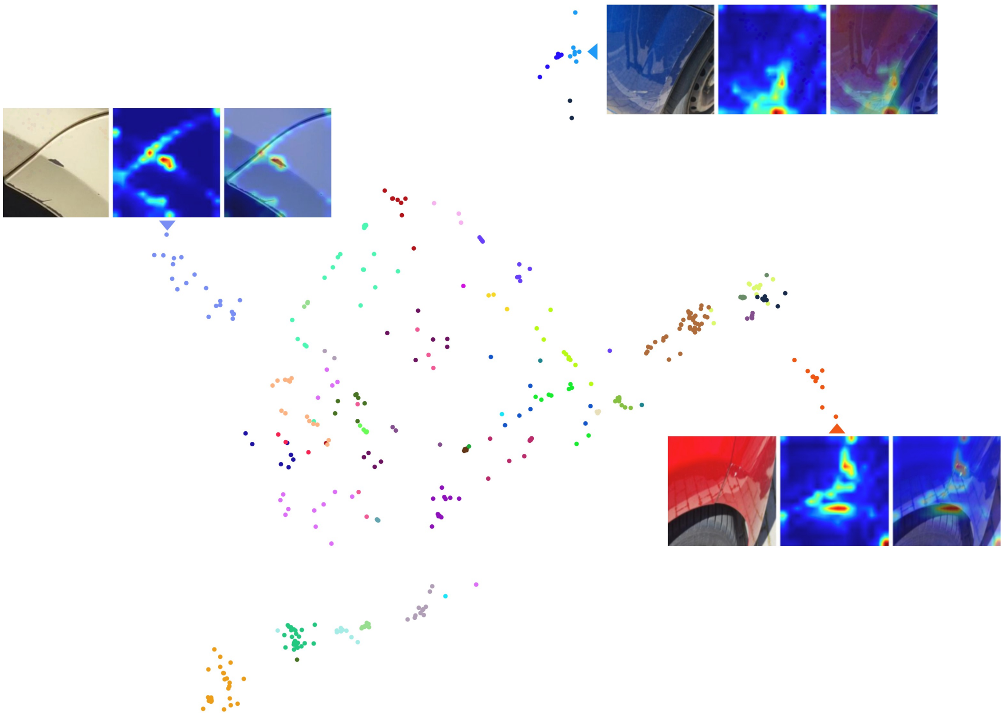


Figure 6.10: Embedding projection in a two-dimensional space using UMAP [201]. Points of the same color represent the same damages. We show three sample damages with their corresponding heatmap. The third column images show the heatmaps overlapped over the images for a better visualization of the activations. The model activations correctly focus on the damaged parts of the vehicle. Added to this, images of the same damages are correctly mapped to one other in the embedding space.

used for image similarity tasks: Tensorflow Similarity [33] and Arcface [63]. For all models, we report the results on the public and private test sets introduced in Section 6.2.2. As shown in Tables 6.4 and 6.5, our similarity model achieves the best performance across both datasets. Arcface achieves higher recall on the public test set, but our proposed model still outperforms all the others in terms of mAP, CMC and top-k $F1_k$. The F1 score is the most important metric for us to take into account. If the recall indicates the number of correctly reidentified damages, to apply the system in a real scenario, we must be sure that the ratio between these and the number of false positives is not too high; otherwise, we would provide a system that correctly identifies an increased number of matches together with an excessive number of false alarms, making our system inapplicable in practice.

Figure 6.9(a) reports the precision–recall curve obtained at different threshold values of the model's predicted class scores. This plot is essential to deploy such a model into production because it allows us to measure the tradeoff between these two metrics. In a real setting, the balance between these two metrics is very important. A system with higher precision is preferred over one with higher recall. In fact, it is very important to have a small number of false positives with respect to the total number of alerts: Because each of the alerts is verified by a claim expert, a high number of false positives would require the verification of too many alarms, thus raising the costs of maintaining the process. However, retrieving all potential fraud attempts is also very important.

Figure 6.9(b) adds another important ingredient to scale into production. The system maintains a very high $recall_k$ within the top-5 and beyond. This is an encouraging result, as it suggests that within five possible similarity alerts, there will be a very high probability of encountering a correct

Model	**mAP**	**CMC**	**Top 5** $F1_k$	**Top 10** $F1_k$	$recall_k$ **95**	$recall_k$ **90**	$recall_k$ **50**
TF similarity [33]	50.8%	76.9%	56.9	45.2%	12.1%	19.7%	43.6%
Arcface [63]	63.7%	84.6%	72.1	58.2%	**33.1**%	**36.7**%	**60.6**%
Ours	**71.5**%	**87.2**%	**80.2**	**65.0**%	29.1%	35.4%	**60.6**%

Table 6.4: Evaluation of the damage-similarity model on the public test set.

Model	**mAP**	**CMC**	**Top 5** $F1_k$	**Top 10** $F1_k$	$recall_k$ **95**	$recall_k$ **90**	$recall_k$ **50**
TF similarity [33]	61.4%	61.4%	33.9	22.5%	-	15.4%	39.1%
Arcface [63]	64.4%	64.4%	35.6	24.2%	32.0%	35.2%	49.0%
Ours	**79.2**%	**79.5**%	**43.8**	**28.1**%	**45.5**%	**49.8**%	**72.3**%

Table 6.5: Evaluation of the damage-similarity model on the private test sets.

match.

Embedding Visualization

There is another key element to consider in evaluating the applicability of such a model in an industrial setting. The end users will be anti-fraud experts, but, as such, ignore how to interpret deep-learning models. What is crucial is that users do not perceive alarms as completely random. Interpretability is very important. As mentioned, a limited number of errors is acceptable, but for the system to be really used it is necessary that even the errors are somehow interpretable. Figure 6.8 shows some examples of matches produced by our system. Green framed images indicate correct matches and purple frames indicate errors. Although some recovered images are incorrect, the errors are still acceptable as they include images that are very similar to query images.

Figure 6.10 shows an embedding obtained through a projection of the features through UMAP [201]. Dots of the same color represent the real matches. Interestingly, the model learns to correctly map similar damage that is very close to each other. What's further interesting is that many false alarms consist of examples that are visually very similar to the input one. In fact, the model maps nearby images of cars of the same model or very similar models and of the same color. Even though the damage is therefore a key component of learning, it is not the only feature used by the model. Added to this, Figure 6.10 shows the attention maps of three images. The activations are mostly concentrated around the damage, which confirms that the model is correctly looking at the damaged area of the picture to make a decision. The analysis of the attention maps suggests that there are some problems that still need to be solved. In many cases, the activations are stronger around the vehicle's escape lines. These are in fact very similar to damage, especially with respect to scratches. We leave the solution of this issue for future development.

6.2.4 Discussion

The proposed solution allows for the identification of possible duplicate damages with acceptable performance in the test phase. However, as mentioned, it is important to be able to apply these solutions in a real scenario. The largest insurance companies operate in several countries around the world. This means being exposed to a huge number of possible variations in the image acquisition processes as well as in the characteristics of the insured vehicles. This implies that the performance of the proposed solution may vary depending on the countries where it is applied. In general, it

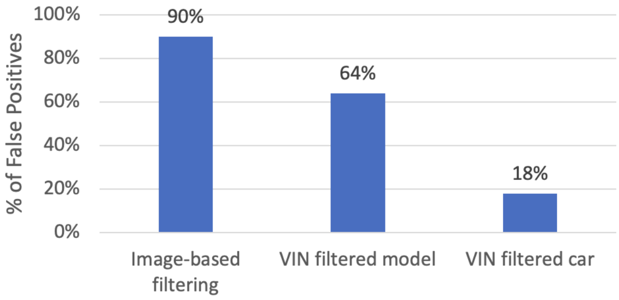


Figure 6.11: The average percentage of false positives across two European countries. Despite the good performance in an the proposed system still produces an average 90% of FPs in the experimental setting. To apply the system into production, we still need data external to the images, for instance, the VIN.

is possible to identify the two most frequent causes of the variation in performance: (1) the *image quality*, which is higher in some countries and very low in others, and (2) the *average number of images* acquired for each claim, which can vary depending on the regulations applied in the various countries. Therefore, the main idea behind this study is to support the damage reidentification module with the vehicle information extracted by the other components of the pipeline. Figure 6.11 shows the average effect of filtering across several countries. Despite the good performances in an experimental setting, the proposed damage reidentification system would still produce an average 90% of *false positives* (FP). Obviously, this is not acceptable. To reduce the alarms further, it is necessary to integrate the information extracted from the images with the Vehicle Identification Number (VIN) data. This is an identification number that acts as the car's fingerprint, as there are no two vehicles on the road with the same VIN. A VIN consists of 17 characters (digits and uppercase letters) which serve as a unique identifier for the vehicle. A VIN shows the car's unique features, specifications, and manufacturer. The VIN can be used to track recalls, registrations, warranty claims, theft, and insurance coverage. By integrating our proposed method with the car model filtering extracted through the VIN leads to 65% of false alarms. By further refining the filter by restricting the search to the single vehicle, the FP is further reduced by up to 18%, which represents a 72% reduction of possible alerts.

We hope that this analysis will stimulate the interest of the scientific community in this type of problem. The results show that despite good performance in the experimental settings, it is still difficult to use a system based solely on image analysis.

Chapter 7

Conclusion and future work

Disinformation is a multifaceted issue that affects various aspects of society, communication, and the dissemination of information. It encompasses various forms and impacts, and understanding its complexity is critical to addressing and mitigating its negative consequences. Countering this phenomenon has become increasingly difficult due to the proliferation of new technologies and the simplicity and speed with which information is propagated.

Misinformation comes in various forms, including disinformation (deliberate false information), misinformation (inaccurate information spread unintentionally), and misinformation (sharing true but harmful information), and it can spread in many forms through social media, traditional media, word of mouth, and various online platforms, which can facilitate rapid dissemination. Disinformation can serve political, financial, or ideological motivations, and perpetrators often seek to influence public opinion or gain an advantage. Disinformation can undermine public trust, polarize societies, and have serious consequences, such as influencing elections or public health behavior during crises. This phenomenon originates from a complex ecosystem including creators, spreaders, and consumers of disinformation who operate through decentralized networks. Tackling misinformation raises technical, legal, and ethical challenges, including balancing free speech and content moderation on online platforms.

In this thesis, we have analyzed the numerous technical challenges still to be solved. We discussed possible solutions to reconstruct the source of origin of images and videos. This is a topic of fundamental importance in today's world. Content is generated at a very high rate, and it is crucial to reconstruct its origin to counter disinformation and use this content for investigations linked to criminal actions. We have shown that it is possible to exploit the traces left by social media platforms during the content upload process and how these traces are unique to each platform. We also showed that images and videos share, albeit up to a certain point, similar traces that can be exploited to train detectors on both media simultaneously. However, many open questions still remain. What makes the traces of the passage of media on a platform unique are the operations, such as compression, that these platforms perform on the media during the upload phase. These algorithms, most often proprietary, are constantly updated by modifying the fingerprint that the platform leaves on the media. Understanding how to make source identification tools resilient to these changes and the multiple devices used to capture photos and videos remains an open question.

We then dealt with the problem of verifying the authenticity of the contents. This is a massive problem in our information society. The technical challenges are multiple and constantly evolving.

While forensic investigation tools are continually under study, at the same time, the proliferation of new, super-advanced content manipulation and generation tools has been even more critical. In just a few years, we have witnessed the birth of generative techniques that until recently seemed like science fiction but have become a reality. Today, it is possible to generate images or videos by providing an AI model with a simple text prompt. While these tools may still seem imperfect, we are not far from making these models capable of generating content that is indistinguishable from the real thing. This is a significant problem that the forensic community immediately embraced but which we are far from solving. In Chapter 4, we discussed several solutions. The analysis of semantic inconsistencies introduced by generative models seems promising, but the biggest stumbling block remains the problem of generalization compared to previously unseen generative techniques. Community efforts have focused on different directions, but we still do not have a definitive solution to the problem. In Chapter 4, we also analyzed human perception of these contents and showed that in some cases, despite the limitations just discussed, automatic detectors can be more accurate than humans.

Chapter 5 analyzed the problem of recognizing fake news, proposing a more accurate vision of this type of content. Information and facts are constantly evolving, and to identify fake news in a real context, it is necessary that the detectors themselves can update automatically. This changes our perspective from a static scenario to a more dynamic one, considering information as a data stream. We have shown that applying transfer learning in this context severely limits the effectiveness of these systems, and we have demonstrated the robustness of continuous learning techniques to these problems. In doing so, we also considered the multimodal nature of news, proposing a detector that simultaneously analyzes images and text of the news and records performances that exceed the state of the art. Continuous learning seems to be a very promising solution, which could also have multiple applications in other media forensics problems. At the same time, however, we have shown a possible vulnerability of these systems. In fact, we have proposed a new adversarial machine learning attack that allows the attacker to manipulate an online learning system's behavior by creating deliberately poisoned content. The potential of the attack is that it can lead the detector to misclassify news to which the attacker does not have direct access, making this attack much more dangerous than previously seen.

Finally, in Chapter 6, we showed two forensic applications related to the verification of contents: (1) matching satellite images with ground images and (2) detecting similar damages to deploy an antifraud system in the insurance context. Both problems give us important insights into the difficulties encountered in the deployment phase of a forensic method outside of an experimental environment.

The future of media forensics is promising and presents a dynamic landscape shaped by technological advances, evolving challenges, and innovative solutions. Looking ahead, several key trends and developments are likely to influence the trajectory of this field. Artificial intelligence and machine learning are ready to play a fundamental role in this sense. These technologies will continue to improve the accuracy and efficiency of detecting manipulated content, including deepfakes and other forms of media deception. In this sense, we believe that multimodal analysis (such as the analysis of images and texts or video and audio) will become increasingly important. Therefore, multimodal methods like the ones presented in Chapters 4 and 5, will be further improved. In parallel with this development, we see a growing need for explainable AI models to provide insights into how decisions are made, improving their transparency and interoperability, especially for content verification systems.

Real-time detection and monitoring of misinformation and manipulated content will become increasingly important, especially for social media platforms and news organizations. Early intervention can help mitigate the impact of false information. Closer collaboration between media forensics experts, fact-checking organizations, and investigative journalists will be essential in the fight against disinformation. Combining technical expertise with journalistic rigor can lead to more practical exposure to fake news. In this context, we believe that continual learning can become a leading player in making automatic tools robust to changes. We plan to extend the studies discussed in Chapter 5 to artificially generated content. Given the continuous evolution of generative techniques, an active learning approach can help readjust the robustness of detectors to these new techniques. Moreover, continual learning could also positively affect the performance of platform identification methods over time. Social media can change the preprocessing operations they do on the uploaded content, and this can change the digital footprints left by each platform. Through constant updating of the detectors, we believe it is possible to improve the performance of these techniques further.

This thesis has explored the complexities of image and video analysis as well as the detection of fake news, delving into the broad field of multimedia forensic investigations. As we traverse the evolving landscape of digital media, the methodologies and insights presented here contribute to a deeper understanding of forensic challenges and solutions. With technology continuing to progress, multimedia forensics has an exciting future ahead of it. This field is expected to be crucial in tackling new difficulties as it anticipates the incorporation of AI, machine learning, and other advanced technologies. By fostering continued research and innovation, we pave the way for a more resilient and adaptive approach to multimedia forensic investigations, ensuring the integrity of evidence and upholding the principles of justice in our continually evolving and technologically advancing world.

Bibliography

[1] L. Abady, E. Cannas, P. Bestagini, B. Tondi, S. Tubaro, M. Barni, et al. An overview on the generation and detection of synthetic and manipulated satellite images. *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.

[2] S. Agarwal, H. Farid, O. Fried, and M. Agrawala. Detecting deep-fake videos from phoneme-viseme mismatches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 660–661, 2020.

[3] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li. Protecting world leaders against deep fakes. In *CVPR workshops*, volume 1, 2019.

[4] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *2009 IEEE 12th International Conference on Computer Vision*, pages 72–79, 2009.

[5] T. Agrawal, R. Gupta, and S. Narayanan. Multimodal detection of fake social media use through a fusion of classification and pairwise ranking systems. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1045–1049, 2017.

[6] E. Ahmed, M. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[7] F. Alam, S. Cresci, T. Chakraborty, F. Silvestri, D. Dimitrov, G. D. S. Martino, S. Shaar, H. Firooz, and P. Nakov. A survey on multimodal disinformation detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6625–6643, Gyeongju, Republic of Korea, Oct. 2022. International Committee on Computational Linguistics.

[8] I. Amerini, A. Anagnostopoulos, L. Maiano, and L. R. Celsi. Deep learning for multimedia forensics. *Foundations and Trends® in Computer Graphics and Vision*, 12(4):309–457, 2021.

[9] I. Amerini, A. Anagnostopoulos, L. Maiano, and L. R. Celsi. Learning double-compression video fingerprints left from social-media platforms. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2530–2534, 2021.

[10] I. Amerini, R. Caldelli, A. D. Mastio, A. D. Fuccia, C. Molinari, and A. P. Rizzo. Dealing with video source identification in social networks. *Signal Processing: Image Communication*, 57:1 – 7, 2017.

[11] I. Amerini, C. Li, and R. Caldelli. Social network identification through image classification with cnn. *IEEE Access*, 7:35264–35273, 2019.

[12] I. Amerini, C. Li, and R. Caldelli. Social network identification through image classification with cnn. *IEEE Access*, 7:35264–35273, 2019.

[13] I. Amerini, T. Uricchio, L. Ballan, and R. Caldelli. Localization of JPEG double compression through multi-domain convolutional neural networks. *CoRR*, abs/1706.01788, 2017.

[14] I. Amerini, T. Uricchio, and R. Caldelli. Tracing images back to their social network of origin: A cnn-based approach. In *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2017.

[15] S. Aneja and M. Nießner. Generalized zero and few-shot transfer for facial forgery detection. *arXiv preprint arXiv:2006.11863*, 2020.

[16] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[17] F. Architecture. Geolocation: Forensic architecture, 2023.

[18] Association of British Insurers (ABI). 8 myths about insurance fraud busted. `https://www.abi.org.uk/products-and-issues/topics-and-issues/fraud/8-myths-about-insurance-fraud/`, 2022. Accessed: 2022-03-10.

[19] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein. Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Online, July 2020. Association for Computational Linguistics.

[20] G. Baatz, O. Saurer, K. Köser, and M. Pollefeys. Large scale visual geo-localization of images in mountainous terrain. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012*, pages 517–530, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[21] R. Baly, G. Karadzhov, J. An, H. Kwak, Y. Dinkov, A. Ali, J. Glass, and P. Nakov. What was written vs. who read it: News media profiling using text analysis and social media context. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3364–3374, Online, July 2020. Association for Computational Linguistics.

[22] R. Baly, M. Mohtarami, J. Glass, L. Màrquez, A. Moschitti, and P. Nakov. Integrating stance detection and fact checking in a unified corpus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 21–27, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[23] H. Bandi, S. Joshi, S. Bhagat, and A. Deshpande. Assessing car damage with convolutional neural networks. In *2021 International Conference on Communication information and Computing Technology (ICCICT)*, pages 1–5. IEEE, 2021.

[24] G. Barnabò, F. Siciliano, C. Castillo, S. Leonardi, P. Nakov, G. Da San Martino, and F. Silvestri. Fbmultilingmisinfo: Challenging large-scale multilingual benchmark for misinformation

detection. In *International Joint Conference on Neural Networks (IJCNN) - to appear*. IEEE, 2022.

[25] G. Barnabò, F. Siciliano, C. Castillo, S. Leonardi, P. Nakov, G. Da San Martino, and F. Silvestri. Deep active learning for misinformation detection using geometric deep learning. *Online Social Networks and Media*, 33:100244, 2023.

[26] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, and S. Tubaro. Aligned and non-aligned double jpeg detection using convolutional neural networks. *Journal of Visual Communication and Image Representation*, 49:153–163, 2017.

[27] B. Bayar and M. C. Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 13(11):2691–2706, 2018.

[28] B. Biggio, P. Russu, L. Didaci, F. Roli, et al. Adversarial biometric recognition: A review on biometric system security from the adversarial machine-learning perspective. *IEEE Signal Processing Magazine*, 32(5):31–41, 2015.

[29] C. Boididou, K. Andreadou, S. Papadopoulos, D. T. Dang Nguyen, G. Boato, M. Riegler, M. Larson, and I. Kompatsiaris. Verifying multimedia use at mediaeval 2015 in mediaeval benchmarking initiative for multimedia evaluation, 09 2015.

[30] F. Bolelli, S. Allegretti, L. Baraldi, and C. Grana. Spaghetti labeling: Directed acyclic graphs for block-based connected components labeling. *IEEE Transactions on Image Processing*, 29:1999–2012, 2020.

[31] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro. Video face manipulation detection through ensemble of cnns. In *2020 25th international conference on pattern recognition (ICPR)*, pages 5012–5019. IEEE, 2021.

[32] J. Brooks. COCO Annotator. `https://github.com/jsbroks/coco-annotator/`, 2019.

[33] E. Bursztein, J. Long, S. Lin, O. Vallis, and F. Chollet. Tensorflow similarity: A usable high-performance metric learning library. *Fixme*, 2021.

[34] R. Caldelli, R. Becarelli, and I. Amerini. Image origin classification based on social network provenance. *Trans. Info. For. Sec.*, 12(6):1299–1308, June 2017.

[35] R. Caldelli, L. Galteri, I. Amerini, and A. Del Bimbo. Optical flow based cnn for detection of unlearnt deepfake manipulations. *Pattern Recognition Letters*, 146:31–37, 2021.

[36] L. Campanile, P. Cantiello, M. Iacono, F. Marulli, and M. Mastroianni. Vulnerabilities assessment of deep learning-based fake news checker under poisoning attacks. *Computational Data and Social Networks*, page 385, 2021.

[37] G. Cantarini, N. Noceti, and F. Odone. Boosting car plate recognition systems performances with agile re-training. In *2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS)*, pages 102–107. IEEE, 2020.

[38] L. Chai, D. Bau, S.-N. Lim, and P. Isola. What makes fake images detectable? understanding properties that generalize, 2020.

[39] S. Chandra, P. Mishra, H. Yannakoudakis, and E. Shutova. Graph-based modeling of online communities for fake news detection. *ArXiv*, abs/2008.06274, 2020.

[40] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. In *CVPR 2011*, pages 737–744, 2011.

[41] J. CHEN, Z. WU, Z. YANG, H. XIE, F. WANG, and W. LIU. Multimodal fusion network with latent topic memory for rumor detection. In *2021 IEEE International Conference on Multimedia and Expo, ICME 2021*, Proceedings - IEEE International Conference on Multimedia and Expo, pages 1–6, United States, June 2021. IEEE Computer Society.

[42] M.-J. Chiu, W.-C. Chiu, H.-T. Chen, and J.-H. Chuang. Real-time monocular depth estimation with extremely light-weight neural network. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7050–7057, 2021.

[43] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[44] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[45] K. Chugh, P. Gupta, A. Dhall, and R. Subramanian. Not made for each other-audio-visual dissonance-based deepfake detection and localization. In *Proceedings of the 28th ACM international conference on multimedia*, pages 439–447, 2020.

[46] R. Corvi, D. Cozzolino, G. Zingarini, G. Poggi, K. Nagano, and L. Verdoliva. On the detection of synthetic images generated by diffusion models. *arXiv preprint arXiv:2211.00680*, 2022.

[47] R. Corvi, D. Cozzolino, G. Zingarini, G. Poggi, K. Nagano, and L. Verdoliva. On the detection of synthetic images generated by diffusion models. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.

[48] D. Cozzolino, D. Gragnaniello, G. Poggi, and L. Verdoliva. Towards universal gan image detection. In *2021 International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5, 2021.

[49] D. Cozzolino, A. Pianese, M. Nießner, and L. Verdoliva. Audio-visual person-of-interest deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 943–952, 2023.

[50] D. Cozzolino, A. Rössler, J. Thies, M. Nießner, and L. Verdoliva. Id-reveal: Identity-aware deepfake video detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15108–15117, 2021.

[51] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv preprint arXiv:1812.02510*, 2018.

[52] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection, 2019.

[53] D. Cozzolino and L. Verdoliva. Noiseprint: A cnn-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security*, 15:144–159, 2020.

[54] Z. Dai, G. Lai, Y. Yang, and Q. Le. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33 of *NIPS'20*, pages 4271–4282, Red Hook, NY, USA, 2020. Curran Associates, Inc.

[55] Z. Dai, G. Wang, W. Yuan, S. Zhu, and P. Tan. Cluster contrast for unsupervised person re-identification. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 1142–1160, December 2022.

[56] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. Jain. On the detection of digital face manipulation, 2020.

[57] S. Das, S. Seferbekov, A. Datta, M. S. Islam, and M. R. Amin. Towards solving the deepfake problem : An analysis on improving deepfake detection using dynamic face augmentation, 2021.

[58] A. Datta. Satellite imagery is taking journalism to new orbits, 2023.

[59] A. De, D. Bandyopadhyay, B. Gain, and A. Ekbal. A transformer-based approach to multilingual fake news detection in low-resource languages. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 21(1), nov 2021.

[60] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022.

[61] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[62] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[63] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition, 2018.

[64] X. Deng, Y. Zhu, and S. Newsam. Using conditional generative adversarial networks to generate ground-level views from overhead imagery. *ArXiv*, abs/1902.06923, 2019.

[65] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[66] D. Dimitrov, B. Bin Ali, S. Shaar, F. Alam, F. Silvestri, H. Firooz, P. Nakov, and G. Da San Martino. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online, Aug. 2021. Association for Computational Linguistics.

[67] X. Ding, Y. Chen, Z. Tang, and Y. Huang. Camera identification based on domain knowledge-driven deep multi-task learning. *IEEE Access*, 7:25878–25890, 2019.

[68] T. DJARA, M. K. ASSOGBA, and A. VIANOU. An approach for benin automatic licence plate recognition. *International Journal of Image Processing (IJIP)*, 11(2):25, 2017.

[69] 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 16x16 words: Transformers for image recognition at scale, 2020.

[70] T. J. Draelos, N. E. Miner, C. C. Lamb, J. A. Cox, C. M. Vineyard, K. D. Carlson, W. M. Severa, C. D. James, and J. B. Aimone. Neurogenesis deep learning: Extending deep networks to accommodate new classes. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 526–533, 2017.

[71] M. Du, S. Pentyala, Y. Li, and X. Hu. Towards generalizable forgery detection with locality-aware autoencoder. *arXiv preprint arXiv:1909.05999*, 1(2):3, 2019.

[72] M. Du, S. Pentyala, Y. Li, and X. Hu. Towards generalizable deepfake detection with locality-aware autoencoder. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 325–334, 2020.

[73] M. Dwivedi, H. S. Malik, S. N. Omkar, E. B. Monis, B. Khanna, S. R. Samal, A. Tiwari, and A. Rathi. Deep learning-based car damage classification and detection. In N. N. Chiplunkar and T. Fukao, editors, *Advances in Artificial Intelligence and Data Engineering*, pages 207–221, Singapore, 2021. Springer Nature Singapore.

[74] E. E. Etomi and D. U. Onyishi. Automated number plate recognition system. *Tropical Journal of Science and Technology*, 2(1):38–48, 2021.

[75] Facebook. Facebook, company info. `https://about.fb.com/company-info/`.

[76] J. Fan, Q. Yan, M. Li, G. Qu, and Y. Xiao. A survey on data poisoning attacks and defenses. In *2022 7th IEEE International Conference on Data Science in Cyberspace (DSC)*, pages 48–55, 2022.

[77] H. Farid. Lighting (in)consistency of paint by text, 2022.

[78] H. Farid. Perspective (in)consistency of paint by text, 2022.

[79] FBI. FBI report on insurance fraud. `https://www.fbi.gov/stats-services/publications/insurance-fraud`, 2022. Accessed: 2022-03-10.

[80] FFmpeg. Ffmpeg, a complete, cross-platform solution to record, convert and stream audio and video. `https://ffmpeg.org/ffprobe.html`.

[81] R. M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999.

[82] J. Fridrich and J. Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.

[83] A. Gandhi and S. Jain. Adversarial perturbations fool deepfake detectors, 2020.

[84] A. R. T. Gepperth and B. Hammer. Incremental learning algorithms and applications. In *The European Symposium on Artificial Neural Networks*, 2016.

[85] O. Giudice, A. Paratore, M. Moltisanti, and S. Battiato. A classification engine for image ballistics of social data. *Lecture Notes in Computer Science*, page 625–636, 2017.

[86] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[87] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Mądry, B. Li, and T. Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1563–1580, 2023.

[88] I. J. Goodfellow, M. Mirza, X. Da, A. C. Courville, and Y. Bengio. An empirical investigation of catastrophic forgeting in gradient-based neural networks. *CoRR*, abs/1312.6211, 2013.

[89] S. Goyal, A. R. Choudhury, S. Raje, V. Chakaravarthy, Y. Sabharwal, and A. Verma. PoWER-BERT: Accelerating BERT inference via progressive word-vector elimination. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3690–3699. PMLR, 13–18 Jul 2020.

[90] D. Gragnaniello, D. Cozzolino, F. Marra, G. Poggi, and L. Verdoliva. Are gan generated images easy to detect? a critical analysis of the state-of-the-art. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021.

[91] W. Guo, B. Tondi, and M. Barni. An overview of backdoor attacks against deep neural networks and possible defences. *IEEE Open Journal of Signal Processing*, 2022.

[92] Y. Guo and N.-M. Cheung. Efficient and deep person re-identification using multi-level similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[93] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using networkx. In G. Varoquaux, T. Vaught, and J. Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[94] A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5039–5049, 2021.

[95] S. W. Hall, A. Sakzad, and K.-K. R. Choo. Explainable artificial intelligence for digital forensics. *WIREs Forensic Science*, 4(2):e1434, 2022.

[96] D. Han, W. Liu, M. Zou, and B. Liu. Non-contrastive nearest neighbor identity-guided method for unsupervised object re-identification. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2022.

[97] Y. Han, S. Karunasekera, and C. Leckie. Continual learning for fake news detection from social media. In I. Farkaš, P. Masulli, S. Otte, and S. Wermter, editors, *Artificial Neural Networks and Machine Learning – ICANN 2021*, pages 372–384, Cham, 2021. Springer International Publishing.

[98] T. L. Hayes, N. D. Cahill, and C. Kanan. Memory efficient experience replay for streaming learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9769–9776, 2019.

[99] J. Hays and A. A. Efros. im2gps: estimating geographic information from a single image. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[100] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn, 2018.

[101] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[102] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[103] S. He, H. Luo, P. Wang, F. Wang, H. Li, and W. Jiang. Transreid: Transformer-based object re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15013–15022, October 2021.

[104] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification, 2017.

[105] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network, 2015.

[106] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[107] S. C. Hoi, D. Sahoo, J. Lu, and P. Zhao. Online learning: A comprehensive survey. *Neurocomputing*, 459:249–289, 2021.

[108] B. D. Horne, J. Nørregaard, and S. Adali. Robust fake news detection over time and attack. *ACM Trans. Intell. Syst. Technol.*, 11(1), dec 2019.

[109] B. D. Horne, J. Nørregaard, and S. Adali. Robust fake news detection over time and attack. *ACM Trans. Intell. Syst. Technol.*, 11(1), dec 2019.

[110] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[111] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[112] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[113] S. Hu, M. Feng, R. M. H. Nguyen, and G. H. Lee. Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7258–7267, 2018.

[114] H. Huang, A. Geiger, and D. Zhang. Good: Exploring geometric cues for detecting objects in an open world, 2023.

[115] H. Huang, N. Sun, X. Lin, and N. Moustafa. Towards generalized deepfake detection with continual learning on limited new data. In *2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–7. IEEE, 2022.

[116] M. Huh, A. Liu, A. Owens, and A. A. Efros. Fighting fake news: Image splice detection via learned self-consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 101–117, 2018.

[117] S. Hussain, P. Neekhara, M. Jere, F. Koushanfar, and J. McAuley. Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples, 2020.

[118] S. Hussain, P. Neekhara, M. Jere, F. Koushanfar, and J. McAuley. Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3348–3357, 2021.

[119] S. Hussain, P. Neekhara, M. Jere, F. Koushanfar, and J. McAuley. Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3348–3357, January 2021.

[120] S. Imaduwage, P. Kumara, and W. Samaraweera. Importance of user representation in propagation network-based fake news detection: A critical review and potential improvements. In *2022 2nd International Conference on Advanced Research in Computing (ICARC)*, pages 90–95, 2022.

[121] G. Inc. Youtube for press. `https://www.youtube.com/about/press/`.

[122] M. Iuliani, M. Fontani, D. Shullani, and A. Piva. Hybrid reference-based video source identification. *Sensors*, 19:649, 02 2019.

[123] M. Iuliani, D. Shullani, M. Fontani, S. Meucci, and A. Piva. A video forensic framework for the unsupervised analysis of mp4-like file container. *IEEE Transactions on Information Forensics and Security*, 14(3):635–645, 2019.

[124] M. Iuliani, D. Shullani, M. Fontani, S. Meucci, and A. Piva. A video forensic framework for the unsupervised analysis of mp4-like file container. *IEEE Transactions on Information Forensics and Security*, 14(3):635–645, 2019.

[125] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks, 2016.

[126] V. Jain, Z. Sasindran, A. Rajagopal, S. Biswas, H. S. Bharadwaj, and K. Ramakrishnan. Deep automatic license plate recognition system. In *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing*, pages 1–8, 2016.

[127] G. Jawahar, B. Sagot, and D. Seddah. What does BERT learn about the structure of language? In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, July 2019.

[128] H. Jeon, Y. Bang, J. Kim, and S. S. Woo. T-gd: Transferable gan-generated images detection framework. *arXiv preprint arXiv:2008.04115*, 2020.

[129] Z. Jin, J. Cao, H. Guo, Y. Zhang, and J. Luo. Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 795–816, New York, NY, USA, 2017. Association for Computing Machinery.

[130] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian. Novel visual and statistical image features for microblogs news verification. *IEEE Transactions on Multimedia*, 19(3):598–608, 2017.

[131] C. Käding, E. Rodner, A. Freytag, and J. Denzler. Fine-tuning deep neural networks in continuous learning scenarios. In C.-S. Chen, J. Lu, and K.-K. Ma, editors, *Computer Vision – ACCV 2016 Workshops*, pages 588–605, Cham, 2017. Springer International Publishing.

[132] S. Kang, J. Hwang, and H. Yu. Multi-modal component embedding for fake news detection. In *2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, pages 1–6, 2020.

[133] A. Kaplan and M. Haenlein. Siri, siri, in my hand: Who's the fairest in the land? on the interpretations, illustrations, and implications of artificial intelligence. *Business horizons*, 62(1):15–25, 2019.

[134] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021.

[135] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[136] F. Khan, S. Hussain, S. Basak, J. Lemley, and P. Corcoran. An efficient encoder–decoder model for portrait depth estimation from single images trained on pixel-accurate synthetic data. *Neural Netw.*, 142(C):479–491, oct 2021.

[137] F. Khan, S. Hussain, S. Basak, J. Lemley, and P. Corcoran. An efficient encoder–decoder model for portrait depth estimation from single images trained on pixel-accurate synthetic data. *Neural Networks*, 142:479–491, 2021.

[138] M. H. Khan, M. Z. Hussein Sk Heerah, and Z. Basgeeth. Automated detection of multi-class vehicle exterior damages using deep learning. In *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pages 01–06, 2021.

[139] S. A. Khan and H. Dai. Video transformer for deepfake detection with incremental learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1821–1828, 2021.

[140] D. Khattar, J. S. Goud, M. Gupta, and V. Varma. Mvae: Multimodal variational autoencoder for fake news detection. In *The World Wide Web Conference*, WWW '19, page 2915–2921, New York, NY, USA, 2019. Association for Computing Machinery.

[141] E. Kiegaing and A. E. Dirik. Prnu-based source device attribution for youtube videos. *Digital Investigation*, 29, 03 2019.

[142] D. Kiela, S. Bhooshan, H. Firooz, and D. Testuggine. Supervised multimodal bitransformers for classifying images and text. *ArXiv*, abs/1909.02950, 2019.

[143] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2611–2624. Curran Associates, Inc., 2020.

[144] M. Kim, S. Tariq, and S. S. Woo. Cored: Generalizing fake media detection with continual representation using distillation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 337–346, 2021.

[145] M. Kim, S. Tariq, and S. S. Woo. Fretal: Generalizing deepfake detection using knowledge distillation and representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1001–1012, 2021.

[146] Y. Kim. Convolutional neural networks for sentence classification, 2014.

[147] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014.

[148] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014.

[149] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 03 2017.

[150] P. Korshunov, M. Halstead, D. Castan, M. Graciarena, M. McLaren, B. Burns, A. Lawson, and S. Marcel. Tampered speaker inconsistency detection with phonetically aware audio-visual features. In *International conference on machine learning*, number CONF in Synthetic Realities: Deep Learning for Detecting AudioVisual Fakes, 2019.

[151] P. Korshunov and S. Marcel. Speaker inconsistency detection in tampered video. In *2018 26th European signal processing conference (EUSIPCO)*, pages 2375–2379. IEEE, 2018.

[152] P. Korshunov and S. Marcel. Subjective and objective evaluation of deepfake videos. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2510–2514, 2021.

[153] P. Korshunov and S. Marcel. Improving generalization of deepfake detection with data farming and few-shot learning. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(3):386–397, 2022.

[154] J. Kreft, M. Boguszewicz-Kreft, and D. Hliebova. Under the fire of disinformation. attitudes towards fake news in the ukrainian frozen war. *Journalism Practice*, pages 1–21, 2023.

[155] P. M. Kyu and K. Woraratpanya. Car damage detection and classification. In *Proceedings of the 11th International Conference on Advances in Information Technology*, pages 1–6, 2020.

[156] F. Lago, C. Pasquini, R. Böhme, H. Dumont, V. Goffaux, and G. Boato. More real than real: A study on human visual perception of synthetic faces [applications corner]. *IEEE Signal Processing Magazine*, 39(1):109–116, 2022.

[157] S. Lee, S. Tariq, J. Kim, and S. S. Woo. Tar: Generalized forensic framework to detect deepfakes using weakly supervised learning. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 351–366. Springer, 2021.

[158] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4655–4665, Red Hook, NY, USA, 2017. Curran Associates Inc.

[159] T. Lesort, H. Caselles-Dupré, M. Garcia-Ortiz, A. Stoian, and D. Filliat. Generative models from the perspective of continual learning. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

[160] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information Fusion*, 58:52–68, 2020.

[161] C. Li, Z. Huang, D. P. Paudel, Y. Wang, M. Shahbazi, X. Hong, and L. Van Gool. A continual deepfake detection benchmark: Dataset, methods, and essentials. *arXiv preprint arXiv:2205.05467*, 2022.

[162] C. Li, Z. Huang, D. P. Paudel, Y. Wang, M. Shahbazi, X. Hong, and L. Van Gool. A continual deepfake detection benchmark: Dataset, methods, and essentials. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1339–1349, 2023.

[163] H. Li and G. Ditzler. Targeted data poisoning attacks against continual learning neural networks. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022.

[164] J. Li, H. Xie, J. Li, Z. Wang, and Y. Zhang. Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6458–6467, 2021.

[165] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5001–5010, 2020.

[166] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang. Visualbert: A simple and performant baseline for vision and language, 2019.

[167] P. Li, B. Shen, and W. Dong. An anti-fraud system for car insurance claim based on visual evidence. *arXiv preprint arXiv:1804.11207*, 2018.

[168] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 152–159, 2014.

[169] B. Liang, Z. Wang, B. Huang, Q. Zou, Q. Wang, and J. Liang. Depth map guided triplet network for deepfake face detection. *Neural Networks*, 159:34–42, 2023.

[170] T.-Y. Lin, S. Belongie, and J. Hays. Cross-view image geolocalization. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2013.

[171] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays. Learning deep representations for ground-to-aerial geolocalization. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5007–5015, 2015.

[172] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection, 2018.

[173] Y. Lin, L. Xie, Y. Wu, C. Yan, and Q. Tian. Unsupervised person re-identification via softened similarity learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[174] H. Liu, J. Feng, Z. Jie, K. Jayashree, B. Zhao, M. Qi, J. Jiang, and S. Yan. Neural person search machines. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 493–501, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society.

[175] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, and N. Yu. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 772–781, 2021.

[176] N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and N. A. Smith. Linguistic knowledge and transferability of contextual representations, 2019.

[177] V. Liu and L. B. Chilton. Design guidelines for prompt engineering text-to-image generative models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.

[178] Y. Liu and Y.-F. B. Wu. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[179] C. Long, E. Smith, A. Basharat, and A. Hoogs. A c3d-based convolutional neural network for frame dropping detection in a single video shot. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1898–1906, 2017.

[180] D. Lopez-Paz and M. A. Ranzato. Gradient episodic memory for continual learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[181] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[182] Y.-J. Lu and C.-T. Li. GCAN: Graph-aware Co-Attention Networks for Explainable Fake News Detection on Social Media. *arXiv e-prints*, page arXiv:2004.11648, Apr. 2020.

[183] Lubna, N. Mufti, and S. A. A. Shah. Automatic number plate recognition:a detailed survey of relevant algorithms. *Sensors*, 21(9), 2021.

[184] J. Lukas, J. Fridrich, and M. Goljan. Determining digital image origin using sensor imperfections. In *Image and Video Communications and Processing 2005*, volume 5685, pages 249–260. International Society for Optics and Photonics, 2005.

[185] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, 2006.

[186] Y. Luo, Y. Zhang, J. Yan, and W. Liu. Generalizing face forgery detection with high-frequency features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16317–16326, 2021.

[187] N. Lyubova, S. Ivaldi, and D. Filliat. From passive to interactive object learning and recognition through self-identification on a humanoid robot. *Autonomous Robots*, 40:33–57, 2016.

[188] R. M. Silva, P. R. Pires, and T. A. Almeida. Incremental learning for fake news detection. *Journal of Information and Data Management*, 13(6), Jan. 2023.

[189] B. Mahdian, S. Saic, and R. Nedbal. Jpeg quantization tables forensics: A statistical approach. In H. Sako, K. Y. Franke, and S. Saitoh, editors, *Computational Forensics*, pages 150–159, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[190] L. Maiano, I. Amerini, L. Ricciardi Celsi, and A. Anagnostopoulos. Identification of social-media platform of videos through the use of shared features. *Journal of Imaging*, 7(8), 2021.

[191] L. Maiano, L. Papa, K. Vocaj, and I. Amerini. Depthfake: A depth-based strategy for detecting deepfake videos. In J.-J. Rousseau and B. Kapralos, editors, *Pattern Recognition, Computer Vision, and Image Processing. ICPR 2022 International Workshops and Challenges*, pages 17–31, Cham, 2023. Springer Nature Switzerland.

[192] A. Mallya and S. Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2017.

[193] D. Maltoni and V. Lomonaco. Continuous learning in single-incremental-task scenarios. *Neural Networks*, 116:56–73, 2019.

[194] M. Manana, C. Tu, and P. A. Owolawi. Edge-based licence-plate template matching for identifying similar vehicles. *Vehicles*, 3(4):646–660, 2021.

[195] S. Mandelli, N. Bonettini, P. Bestagini, and S. Tubaro. Detecting gan-generated images by orthogonal training of multiple cnns. *arXiv preprint arXiv:2203.02246*, 2022.

[196] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi. Do gans leave artificial fingerprints? In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 506–511, 2019.

[197] F. Marra, C. Saltori, G. Boato, and L. Verdoliva. Incremental learning for the detection and classification of gan-generated images. In *2019 IEEE international workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2019.

[198] O. Mayer, B. Bayar, and M. C. Stamm. Learning unified deep-features for multiple forensic tasks. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '18, page 79–84, New York, NY, USA, 2018. Association for Computing Machinery.

[199] O. Mayer and M. C. Stamm. Forensic similarity for digital images. *IEEE Transactions on Information Forensics and Security*, 15:1331–1346, 2020.

[200] G. Mazaheri, N. Chowdhury Mithun, J. H. Bappy, and A. K. Roy-Chowdhury. A skip connection architecture for localization of image manipulations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[201] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018.

[202] N. Messina, F. Falchi, C. Gennaro, and G. Amato. AIMH at SemEval-2021 task 6: Multimodal classification using an ensemble of transformer models. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1020–1026, Online, Aug. 2021. Association for Computational Linguistics.

[203] Meta. Pytorch, an open source machine learning framework that accelerates the path from research prototyping to production deployment. `https://pytorch.org/`.

[204] M. Meyers, G. Weiss, and G. Spanakis. Fake news detection on twitter using propagation structures. In M. van Duijn, M. Preuss, V. Spaiser, F. Takes, and S. Verberne, editors, *Disinformation in Open Online Media*, pages 138–158, Cham, 2020. Springer International Publishing.

[205] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha. Emotions don't lie: An audio-visual deepfake detection method using affective cues. In *Proceedings of the 28th ACM international conference on multimedia*, pages 2823–2832, 2020.

[206] M. Moltisanti, A. Paratore, S. Battiato, and L. Saravo. Image manipulation on facebook for forensics evidence. In V. Murino and E. Puppo, editors, *Image Analysis and Processing — ICIAP 2015*, pages 506–517, Cham, 2015. Springer International Publishing.

[207] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein. Fake news detection on social media using geometric deep learning, 2019.

[208] B. Munjal, S. Amin, F. Tombari, and F. Galasso. Query-guided end-to-end person search. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[209] S. Nam, J. Park, D. Kim, I. Yu, T. Kim, and H. Lee. Two-stream network for detecting double compression of h.264 videos. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 111–115, 2019.

[210] Q. Nan, J. Cao, Y. Zhu, Y. Wang, and J. Li. Mdfend: Multi-domain fake news detection. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3343–3347, 2021.

[211] P. Neekhara, B. Dolhansky, J. Bitton, and C. C. Ferrer. Adversarial threats to deepfake detection: A practical perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 923–932, 2021.

[212] P. Neekhara, B. Dolhansky, J. Bitton, and C. C. Ferrer. Adversarial threats to deepfake detection: A practical perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 923–932, June 2021.

[213] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018.

[214] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. In *2019 IEEE 10th international conference on biometrics theory, applications and systems (BTAS)*, pages 1–8. IEEE, 2019.

[215] V.-H. Nguyen, K. Sugiyama, P. Nakov, and M.-Y. Kan. Fang: Leveraging social context for fake news detection using graph representation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 1165–1174, New York, NY, USA, 2020. Association for Computing Machinery.

[216] V.-H. Nguyen, K. Sugiyama, P. Nakov, and M.-Y. Kan. Fang: Leveraging social context for fake news detection using graph representation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1165–1174, 2020.

[217] T. Ophoff, K. V. Beeck, and T. Goedemé. Exploring RGBDepth fusion for real-time object detection. *Sensors*, 19(4):866, Feb. 2019.

[218] J. Oppenlaender. The creativity of text-to-image generation. In *Proceedings of the 25th International Academic Mindtrek Conference*, Academic Mindtrek '22, page 192–202, New York, NY, USA, 2022. Association for Computing Machinery.

[219] J. Oppenlaender. Prompt engineering for text-based generative art. *arXiv preprint arXiv:2204.13988*, 2022.

[220] J. Oppenlaender. A taxonomy of prompt modifiers for text-to-image generation, 2022.

[221] L. Papa, E. Alati, P. Russo, and I. Amerini. Speed: Separable pyramidal pooling encoder-decoder for real-time monocular depth estimation on low-resource settings. *IEEE Access*, pages 44881–44890, 2022.

[222] L. Papa, L. Faiella, L. Corvitto, L. Maiano, and I. Amerini. On the use of stable diffusion for creating realistic faces: from generation to detection. In *2023 11th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6, 2023.

[223] C. Pasquini, I. Amerini, and G. Boato. Media forensics on social media platforms: a survey. *EURASIP Journal on Information Security*, 2021(1):4, May 2021.

[224] K. Patil, M. Kulkarni, A. Sriraman, and S. Karande. Deep learning based car damage classification. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)*, pages 50–54. IEEE, 2017.

[225] N. Pavlichenko, F. Zhdanov, and D. Ustalov. Best prompts for text-to-image models and how to find them, 2022.

[226] V. Peluso, A. Cipolletta, A. Calimera, M. Poggi, F. Tosi, F. Aleotti, and S. Mattoccia. Monocular depth perception on microcontrollers for edge applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(3):1524–1536, 2022.

[227] Q. Phan, G. Boato, R. Caldelli, and I. Amerini. Tracking multiple image sharing on social networks. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8266–8270, 2019.

[228] A. C. Popescu and H. Farid. Statistical tools for digital forensics. In J. Fridrich, editor, *Information Hiding*, pages 128–147, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[229] K. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *Proceedings of the 28th ACM international conference on multimedia*, pages 484–492, 2020.

[230] K. R. Price, J. Priisalu, and S. Nomm. Analysis of the impact of poisoned data within twitter classification models. *IFAC-PapersOnLine*, 52(19):175–180, 2019. 14th IFAC Symposium on Analysis, Design, and Evaluation of Human Machine Systems HMS 2019.

[231] P. Qi, J. Cao, T. Yang, J. Guo, and J. Li. Exploiting multi-domain visual information for fake news detection. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 518–527, Los Alamitos, CA, USA, nov 2019. IEEE Computer Society.

[232] Y. Quan, X. Lin, and C.-T. Li. Provenance analysis for instagram photos. In R. Islam, Y. S. Koh, Y. Zhao, G. Warwick, D. Stirling, C.-T. Li, and Z. Islam, editors, *Data Mining*, pages 372–383, Singapore, 2019. Springer Singapore.

[233] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.

[234] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[235] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics.

[236] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection, 2016.

[237] K. Regmi and M. Shah. Bridging the domain gap for ground-to-aerial image matching. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 470–479, 2019.

[238] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[239] Y. Ren, B. Wang, J. Zhang, and Y. Chang. Adversarial active learning based heterogeneous graph neural network for fake news detection. *2020 IEEE International Conference on Data Mining (ICDM)*, pages 452–461, 2020.

[240] J. Ricker, S. Damm, T. Holz, and A. Fischer. Towards the detection of diffusion model deepfakes. *arXiv preprint arXiv:2210.14571*, 2022.

[241] A. Rios and L. Itti. Closed-loop memory gan for continual learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, page 3332–3338. AAAI Press, 2019.

[242] Y. M. Rocha, G. A. de Moura, G. A. Desidério, C. H. de Oliveira, F. D. Lourenço, and L. D. de Figueiredo Nicolete. The impact of fake news on social media and its influence on health during the covid-19 pandemic: A systematic review. *Journal of Public Health*, pages 1–10, 2021.

[243] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[244] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[245] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images, 2019.

[246] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *ArXiv*, abs/1606.04671, 2016.

[247] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[248] A. Sathe, S. Ather, T. M. Le, N. Perry, and J. Park. Automated fact-checking of claims from Wikipedia. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6874–6882, Marseille, France, May 2020. European Language Resources Association.

[249] D. Saxena and J. Cao. Generative adversarial networks (gans): Challenges, solutions, and future directions. *ACM Computing Surveys*, 54(3), may 2021.

[250] P. Schramowski, M. Brack, B. Deiseroth, and K. Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models, 2022.

[251] S. Seetharaman, S. Malaviya, R. Vasu, M. Shukla, and S. Lodha. Influence based defense against data poisoning attacks in online learning. In *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 1–6, 2022.

[252] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, oct 2019.

[253] Z. Sha, Z. Li, N. Yu, and Y. Zhang. De-fake: Detection and attribution of fake images generated by text-to-image diffusion models. *arXiv preprint arXiv:2210.06998*, 2022.

[254] S. Shaar, N. Babulkov, G. Da San Martino, and P. Nakov. That is a known lie: Detecting previously fact-checked claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618, Online, July 2020. Association for Computational Linguistics.

[255] Q. Shan, C. Wu, B. Curless, Y. Furukawa, C. Hernandez, and S. M. Seitz. Accurate georegistration by ground-to-aerial image matching. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 525–532, 2014.

[256] Y. Shen, T. Xiao, H. Li, S. Yi, and X. Wang. End-to-end deep kronecker-product matching for person re-identification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[257] Y. Shi, X. Yu, D. Campbell, and H. Li. Where am i looking at? joint location and orientation estimation by cross-view matching. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4063–4071, 2020.

[258] Y. Shi, X. Yu, L. Liu, T. Zhang, and H. Li. Optimal feature transport for cross-view image geo-localization. In *arXiv preprint arXiv:1907.05021*, 2019.

[259] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 2994–3003, Red Hook, NY, USA, 2017. Curran Associates Inc.

[260] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *arXiv preprint arXiv:1809.01286*, 2018.

[261] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *arXiv preprint arXiv:1809.01286*, 2018.

[262] K. Shu, D. Mahudeswaran, S. Wang, and H. Liu. Hierarchical propagation networks for fake news detection: Investigation and exploitation. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1):626–637, May 2020.

[263] D. Shullani, M. Fontani, M. Iuliani, O. Alshaya, and A. Piva. Vision: a video and image dataset for source identification. *EURASIP Journal on Information Security*, 2017:15, 10 2017.

[264] N. Siddiqui, A. Anjum, M. Saleem, and S. Islam. Social media origin based image tracing using deep cnn. In *2019 Fifth International Conference on Image Information Processing (ICIIP)*, pages 97–101, 2019.

[265] A. Silva, Y. Han, L. Luo, S. Karunasekera, and C. Leckie. Propagation2vec: Embedding partial propagation networks for explainable fake news early detection. *Information Processing & Management*, 58(5):102618, 2021.

[266] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[267] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[268] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[269] S. Singhal, R. R. Shah, T. Chakraborty, P. Kumaraguru, and S. Satoh. Spotfake: A multimodal framework for fake news detection. In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, pages 39–47, 2019.

[270] C. Song, K. Shu, and B. Wu. Temporally evolving graph neural network for fake news detection. *Information Processing & Management*, 58(6):102712, 2021.

[271] P. Sprechmann, S. Jayakumar, J. Rae, A. Pritzel, A. P. Badia, B. Uria, O. Vinyals, D. Hassabis, R. Pascanu, and C. Blundell. Memory-based parameter adaptation. In *International Conference on Learning Representations*, 2018.

[272] P. Sprechmann, S. M. Jayakumar, J. W. Rae, A. Pritzel, A. P. Badia, B. Uria, O. Vinyals, D. Hassabis, R. Pascanu, and C. Blundell. Memory-based parameter adaptation. *ArXiv*, abs/1802.10542, 2018.

[273] M. C. Stamm, W. S. Lin, and K. J. R. Liu. Temporal forensics and anti-forensics for motion compensated video. *IEEE Transactions on Information Forensics and Security*, 7(4):1315–1329, 2012.

[274] M. C. Stamm, M. Wu, and K. J. R. Liu. Information forensics: An overview of the first decade. *IEEE Access*, 1:167–200, 2013.

[275] E. Stumm, C. Mei, S. Lacroix, and M. Chli. Location graphs for visual place recognition. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5475–5480, 2015.

[276] A. Subramaniam, M. Chatterjee, and A. Mittal. Deep neural networks with inexact matching for person re-identification. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[277] Z. Sun, Y. Han, Z. Hua, N. Ruan, and W. Jia. Improving the efficiency and robustness of deepfakes detection through precise geometric features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3609–3618, 2021.

[278] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision, 2015.

[279] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[280] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. Van Durme, S. R. Bowman, D. Das, and E. Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations, 2019.

[281] The GIMP Development Team. GIMP, 2019.

[282] X. Tian, J. Shao, D. Ouyang, A. Zhu, and F. Chen. Smdt: Cross-view geo-localization with image alignment and transformer. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2022.

[283] R. Varior, M. Haloi, and G. Wang. Gated siamese convolutional neural network architecture for human re-identification. In *Proceedings of the 14th European Conference on Computer Vision (ECCV 2016)*, volume 9912, pages 791–808, 10 2016.

[284] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[285] S. Verde, L. Bondi, P. Bestagini, S. Milani, G. Calvagno, and S. Tubaro. Video codec forensics based on convolutional neural networks. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 530–534, 2018.

[286] S. Verde, T. Resek, S. Milani, and A. Rocha. Ground-to-aerial viewpoint localization via landmark graphs matching. *IEEE Signal Processing Letters*, 27:1490–1494, 2020.

[287] L. Verdoliva. Media forensics and deepfakes: an overview, 2020.

[288] N. Vo and K. Lee. The rise of guardians: Fact-checking url recommendation to combat fake news. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 275–284, New York, NY, USA, 2018. Association for Computing Machinery.

[289] N. N. Vo and J. Hays. Localizing and orienting street views using overhead imagery. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 494–509, Cham, 2016. Springer International Publishing.

[290] S. Volkova, E. Ayton, D. L. Arendt, Z. Huang, and B. Hutchinson. Explaining multimodal deceptive news prediction models. In *ICWSM*, 2019.

[291] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 359:1146–1151, 03 2018.

[292] C. Wang and W. Deng. Representative forgery mining for fake face detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14923–14932, 2021.

[293] J. Wang, Z. Wu, W. Ouyang, X. Han, J. Chen, S.-N. Lim, and Y.-G. Jiang. M2tr: Multi-modal multi-scale transformers for deepfake detection, 2022.

[294] R. Wang, F. Juefei-Xu, L. Ma, X. Xie, Y. Huang, J. Wang, and Y. Liu. Fakespotter: A simple yet robust baseline for spotting ai-synthesized fake faces. *arXiv preprint arXiv:1909.06122*, 2019.

[295] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8695–8704, 2020.

[296] W. Wang and H. Farid. Exposing digital forgeries in video by detecting double mpeg compression. In *Proceedings of the 8th Workshop on Multimedia and Security*, MM and Sec 2006, page 37–47, New York, NY, USA, 2006. Association for Computing Machinery.

[297] W. Wang and H. Farid. Exposing digital forgeries in video by detecting double mpeg compression. In *Proceedings of the 8th Workshop on Multimedia and Security*, MM&Sec '06, page 37–47, New York, NY, USA, 2006. Association for Computing Machinery.

[298] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu. Shape and appearance context modeling. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[299] X. Wang, Y. Yan, P. Tang, X. Bai, and W. Liu. Revisiting multiple instance neural networks. *Pattern Recognition*, 74:15–24, 2018.

[300] Y. Wang and K. Chaudhuri. Data poisoning attacks against online learning. *arXiv preprint arXiv:1808.08994*, 2018.

[301] Y. Wang, Z. Chen, F. Wu, and G. Wang. Person re-identification with cascaded pairwise convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[302] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, and J. Gao. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 849–857, New York, NY, USA, 2018. Association for Computing Machinery.

[303] Z. J. Wang, E. Montoya, D. Munechika, H. Yang, B. Hoover, and D. H. Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models, 2022.

[304] C. work of all DFRWS attendees. A road map for digital forensics research. The Digital Forensic Research Conference (DFRWS) 2001 USA, 2001.

[305] S. Workman, R. Souvenir, and N. Jacobs. Wide-area image geolocalization with aerial reference imagery. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–9, 2015. Acceptance rate: 30.3%.

[306] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[307] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

[308] Y. Wu, P. Zhan, Y. Zhang, L. Wang, and Z. Xu. Multimodal fusion with co-attention networks for fake news detection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2560–2569, Online, Aug. 2021. Association for Computational Linguistics.

[309] Z. Wu, J. Chen, Z. Yang, H. Xie, F. L. Wang, and W. Liu. Cross-modal attention network with orthogonal latent memory for rumor detection. In W. Zhang, L. Zou, Z. Maamar, and L. Chen, editors, *Web Information Systems Engineering – WISE 2021*, pages 527–541, Cham, 2021. Springer International Publishing.

[310] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang. Joint detection and identification feature learning for person search. In *CVPR*, 2017.

[311] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[312] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[313] N. L. Yaacob, A. A. Alkahtani, F. M. Noman, A. W. M. Zuhdi, and D. Habeeb. License plate recognition for campus auto-gate system. *Indonesian Journal of Electrical Engineering and Computer Science*, 21(1):128–136, 2021.

[314] P. Yang, D. Baracchi, R. Ni, Y. Zhao, F. Argenti, and A. Piva. A survey of deep learning-based source image forensics. *Journal of Imaging*, 6(3):9, 03 2020.

[315] Y. Yang, L. Zheng, J. Zhang, Q. Cui, Z. Li, and P. S. Yu. Ti-cnn: Convolutional neural networks for fake news detection, 2023.

[316] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. H. Hoi. Deep learning for person re-identification: A survey and outlook, 2020.

[317] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks?, 2014.

[318] J. Yu, Q. Huang, X. Zhou, and Y. Sha. Iarnet: An information aggregating and reasoning network over heterogeneous graph for fake news detection. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2020.

[319] N. Yu, L. S. Davis, and M. Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7556–7566, 2019.

[320] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi. From recognition to cognition: Visual commonsense reasoning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[321] Y. Zhan, Y. Chen, Q. Zhang, and X. Kang. Image forensics based on transfer learning and convolutional neural network. In *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '17, page 165–170, New York, NY, USA, 2017. Association for Computing Machinery.

[322] D. Zhang, L. Shang, B. Geng, S. Lai, K. Li, H. Zhu, T. Amin, and D. Wang. Fauxbuster: A content-free fauxtography detector using social media comments. In *Proceedings of IEEE BigData 2018*, 2018.

[323] H. Zhang, Q. Fang, S. Qian, and C. Xu. Multi-modal knowledge-aware event memory network for social media rumor detection. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 1942–1951, New York, NY, USA, 2019. Association for Computing Machinery.

[324] Q. Zhang, X. Chang, and S. B. Bian. Vehicle-damage-detection segmentation algorithm based on improved mask rcnn. *IEEE Access*, 8:6997–7004, 2020.

[325] X. Zhang, W. Sultani, and S. Wshah. Cross-view image sequence geo-localization. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2913–2922, 2023.

[326] X. Zhang, X. Zhu, and L. Lessard. Online data poisoning attacks. In *Learning for Dynamics and Control*, pages 201–210. PMLR, 2020.

[327] Y. Zhang and Q. Yang. A survey on multi-task learning, 2018.

[328] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2185–2194, 2021.

[329] T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong, and W. Xia. Learning self-consistency for deepfake detection, 2021.

[330] Y. Zhao, X. Gong, F. Lin, and X. Chen. Data poisoning attacks and defenses in dynamic crowdsourcing with online data quality learning. *IEEE Transactions on Mobile Computing*, 22(5):2569–2581, 2023.

[331] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1116–1124, 2015.

[332] Q. Zheng, C. Liang, W. Fang, D. Xiang, X. Zhao, C. Ren, and J. Chen. Car re-identification from large scale images using semantic attributes. In *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–5, 2015.

[333] Y. Zheng, J. Bao, D. Chen, M. Zeng, and F. Wen. Exploring temporal coherence for more general video face forgery detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15044–15054, 2021.

[334] G. Zhou, K. Sohn, and H. Lee. Online incremental feature learning with denoising autoencoders. In *International Conference on Artificial Intelligence and Statistics*, 2012.

[335] K. Zhou and T. Xiang. Torchreid: A library for deep learning person re-identification in pytorch. *arXiv preprint arXiv:1910.10093*, 2019.

[336] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang. Omni-scale feature learning for person re-identification. In *ICCV*, 2019.

[337] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang. Learning generalisable omni-scale representations for person re-identification. *TPAMI*, 2021.

[338] T. Zhou, D.-P. Fan, M.-M. Cheng, J. Shen, and L. Shao. RGB-d salient object detection: A survey. *Computational Visual Media*, 7(1):37–69, jan 2021.

[339] X. Zhou, J. Wu, and R. Zafarani. Safe: Similarity-aware multi-modal fake news detection, 2020.

[340] X. Zhou and R. Zafarani. A survey of fake news. *ACM Computing Surveys*, 53(5):1–40, 10 2020.

[341] X. Zhou and R. Zafarani. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Comput. Surv.*, 53(5), sep 2020.

[342] Y. Zhou and S.-N. Lim. Joint audio-visual deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14800–14809, 2021.

[343] C. Zhu, M. Zeng, and X. Huang. Sdnet: Contextualized attention-based deep network for conversational question answering, 2018.

[344] H. Zhu, W. Ke, D. Li, J. Liu, L. Tian, and Y. Shan. Dual cross-attention learning for fine-grained visual categorization and object re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4692–4702, June 2022.

[345] X. Zhu, S. Liu, P. Zhang, and Y. Duan. A unified framework of intelligent vehicle damage assessment based on computer vision technology. In *2019 IEEE 2nd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, pages 124–128. IEEE, 2019.

[346] X. Zhu, H. Wang, H. Fei, Z. Lei, and S. Z. Li. Face forgery detection by 3d decomposition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2929–2939, 2021.

[347] B. Zi, M. Chang, J. Chen, X. Ma, and Y.-G. Jiang. Wilddeepfake: A challenging real-world dataset for deepfake detection. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2382–2390, 2020.