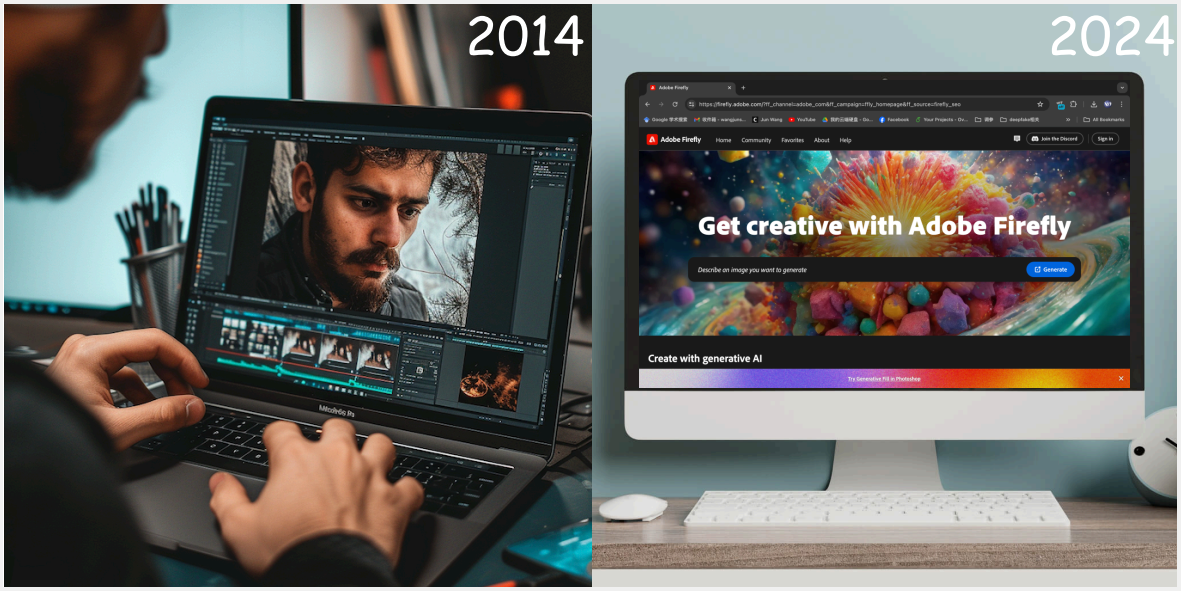
Detection and Attribution of AI-generated Images in the Wild



Jun Wang

PhD in Information Engineering and Science

UNIVERSITÀ DEGLI STUDI DI SIENA

Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche

Detection and Attribution of AI-generated Images in the Wild

Jun Wang

PhD in Information Engineering and Science
XXXVI Cycle, 2020-2023

Supervisor

Prof. Mauro Barni

Co-supervisor

Prof. Benedetta Tondi

Examination Commitee

Prof. Luisa Verdoliva

Prof. Pavel Korshunov

Prof. Federico Becattini

Thesis reviewers

Prof. Luisa Verdoliva

Prof. Pavel Korshunov

Siena, 22/07/2024

Contents

List of Figures

List of Tables

List of Symbols

x, y	Input sample, ground truth label (/one-hot encoding vector)
H, W	Height and Width of a given image
N_t	Number of training samples
C	Number of in-set class
p	Network output probability vector
$\hat{y}$	Final predicted label
I_M, I_G	Predicted localization mask and its ground truth
$\mathcal{L}$	Loss function
$\mathcal{L}_{cls}$, $\mathcal{L}_{loc}$	Classification and localization loss
λ_{cls}, λ_{loc}	Balancing weights of classification and localization loss
ϕ	Neural network output function
ϕ^{-1}	Logit output of neural network function
D_k, D_u	Sets of in-set classes, out-of-set classes
H_f, W_f, D_f	Height, Width and depth of feature map
ν	Threshold for out-of-set rejection
C_l	Set of in-set labels
$\Re$	Rejection class
FC	Fully connected layer
FCN	Fully convolutional network
MSE	Mean square error
CE	Cross entropy
ACC	Accuracy
TPR, TNR, FPR	True positive rate, True negative rate, False positive rate
AU-ROC	Area under the receiver operating characteristics curve
Pd@5%	Probability of detection at 5% FPR
AU-OSCR	Aera under the Open-Set classification rate curve

Acknowledgements

I want to express my deepest gratitude to my supervisor, Prof. Mauro Barni, for his exceptional guidance, unwavering support, and invaluable mentorship throughout my PhD study. His expertise, dedication, and insightful feedback have been instrumental in shaping the direction of this thesis and enhancing its quality. I am equally grateful to my co-supervisor, Prof. Benedetta Tondi, for her rigorous guidance and powerful support of my research and paper writing. I would also like to thank my colleagues at VIPP Group. Their camaraderie, support, and intellectual exchange have fostered a stimulating academic environment, contributing to my personal and professional growth. I am grateful for the opportunity to collaborate with such talented individuals. I am also indebted to the members of my thesis committee, Prof. Federico Becattini, Prof. Pavel Korshunov, and Prof. Annalisa Verdoliva, for their insightful feedback, constructive criticism, and valuable suggestions.

Furthermore, I would like to acknowledge the financial support the *China Scholarship Council* provided. I am sincerely grateful that No. 202008370186 has provided the necessary resources and funding to carry out this research. Finally, I would like to express my heartfelt gratitude to my family for their unwavering love, encouragement, and support throughout this journey. Their belief in me and their sacrifices have been a constant source of strength and inspiration. I am deeply thankful for their unconditional support and guidance.

Chapter 1

Introduction

"In a time of deceit, telling the truth is a revolutionary act."
George Orwell

In the last years, the usage of digital content, and images in particular, has surged owing to the widespread availability of built-in cameras in various devices such as smartphones, laptops, and tablets. Furthermore, sharing digital images across social media platforms like Instagram, WhatsApp, and YouTube has become commonplace.

However, digital images can be easily manipulated using various easy-to-use image editing software, such as Adobe Photoshop Express, Snapseed, and PhotoEditor, available on various devices, especially smartphones. Nowadays, the best-performing image editing tools are all based on Artificial Intelligence (AI), notably Deep Learning (DL) technology, as shown in Figure 1.1. At a broader level, many businesses and institutions are investigating the integration of their generative AI systems to streamline repetitive tasks and improve overall efficiency. Adobe Firefly[1] uses generative AI and simple text prompts to create extremely high-quality images, allowing text-to-image generation and text-guided image filling. OpenAI released DALL-E3 [20] and integrated it into ChatGPT[2] where a simple idea can be translated into tailored and detailed prompts for DALL-E3 image generation. These open generative models can generate or modify images simply by providing a descriptive text of the elements they wish to appear in the images for "$1,000 images for $1000". In addition, there are lots of free generative models available online that produce extremely realistic fake images.

Alongside benign uses of this technology, however, the malicious use of AI-generated contents, generally referred to as *Deepfakes*, represents a serious and concrete threat, including the dissemination of misinformation and the unauthorized use of personal data resulting in privacy violations, such as non-consensual pornography. Moreover, the interconnected digital world we live in has the consequence that fake content can be visualized by millions of users shortly after the contents are uploaded on the web or on social platforms. Months ago, the spread of explicit AI-generated images of Taylor Swift was shocking and saddening, racking up more than 45 million views before the account was suspended[3]. In addition, Australia's consumer watchdog is warning the public to beware of fake news articles and images endorsed by celebrities and other public figures and linked to online investment trading platforms[4]. The potential misuse of these powerful

[1] `https://firefly.adobe.com/inspire/images`
[2] `https://chat.openai.com/`
[3] `https://ca.news.yahoo.com/white-house-alarmed-over-fake-142631636.html`
[4] Aussies lose $8 million to deepfake celebrity investment scams

Figure 1.1 – Image editing from 2014 to 2024. Given an image to be edited, the part indicated by a mask can be filled with the content generated by the generative AI conditioned on a text prompt, or the entire image can be generated using the text prompt.

tools erodes trust and confidence in the digital ecosystem, calling for the development of multimedia forensic techniques to verify content authenticity in an attempt to restore the media's credibility. In response to the issues caused by generative AI techniques, governments have taken steps to address the problem of misuse of generated images. Since 2016, the Defense Advanced Research Projects Agency (DARPA) of the U.S. Department of Defense has funded two image forensic programs dedicated to detecting deepfakes: Media Forensics (MediFor) and Semantic Forensics (SemaFor)[5], to which program this thesis contributes. In China, a recent law stipulates that manipulated contents must carry a digital signature or watermark, and content generation service providers must engage not to process personal information and comply with other rules such as the evaluation and verification of AI algorithms deployed, authentication of users, and setting up feedback mechanisms for content consumers[6]. Facebook AI launched the DeepFake Detection Challenge (DFDC) fake face recognition challenge on the Kaggle platform [21].

As a reaction to the above problems, the multimedia forensic community has dedicated significant efforts to the development of effective techniques to verify image authenticity and combat misinformation. In particular, great attention has been devoted to the development of methods for the detection of AI-synthesized images - referred to as *synthetic image detection*[7], namely techniques capable of distinguishing between real and fake images[8]. In addition to binary detectors, which are only asked to judge if an image is real or fake, some methods have been proposed that, in the case of AI-manipulated images, have also the ability to localize the manipulation, e.g., outputting a binary localization mask or an attention map. Another prominent problem multimedia forensic researchers have been focusing on in the last years is *synthetic image attribution*, that is, determining

[5] `https://www.darpa.mil/`

[6] chinas-new-legislation-on-deepfakes-should-the-rest-of-asia-follow-suit

[7] The terms synthetic and fake are used interchangeably throughout the thesis.

[8] AI-generated images is often used as a general term and includes both fully synthetic images and partially manipulated images. We will use the terms AI-synthesized and AI-manipulated to indicate, respectively, fully synthetic images and partially manipulated images, when we will need to differentiate between the two cases.

which is the specific generative model or architecture that has been used to produce a given synthetic image. By identifying the specific generative model or technique used to generate synthetic images, researchers can elucidate the underlying mechanisms behind image generation and manipulation, thereby enhancing the transparency and interpretability of detection algorithms. This transparency is essential for building trust among stakeholders, including law enforcement agencies, media organizations, and the general public, who rely on the accuracy and reliability of image authentication systems.

After some initial attempts to address the above tasks with techniques based on handcrafted and model-based features [16, 22–24] in the last few years, superior performance has been achieved by methods resorting to DL, notably Convolution Neural Networks (CNN) [25], thus adhering to the 'AI to combat AI' paradigm. These methods can achieve almost perfect accuracy in laboratory settings when trained and tested under the same conditions, i.e., when the distribution of test samples matches that of training data. However, this is not usually the case in practice, thus limiting the applicability of these methods in real-world applications, where the images seen at operation time may have originated from a different dataset, be subject to a different processing or generation pipeline, or be obtained using different generative models/architectures or different manipulation types, with respect to those considered during the training of the forensic tools. This is even more evident given the rapid pace at which generative methods evolve. It is, in fact, impossible to encompass all the generative models that these methods will encounter during their operational use, at training time. In such a scenario, the predictions made by DL-based methods are not trustworthy. Therefore, the development of forensic methods capable of working *in the wild*, under the typical conditions encountered in real-world applications is an urgent need.

In this thesis, we contribute to the above mission by proposing techniques for synthetic image detection, classification, and attribution with enhanced reliability, thereby facilitating their practical application. More specifically, synthetic image detection aims to distinguish AI-generated images from pristine ones, such as Generative Adversarial Network (GAN)-synthesized face images and GAN-manipulated flood images. On the other hand, the classification of facial attribute editing identifies which specific manipulation was conducted on the images via a multi-class classification framework. Finally, the synthetic image attribution task addresses the provenance of AI-synthesized images, determining the generative models or architectures used to generate them. Given the prominent role played by images depicting persons, and in particular face images, in the thesis, we almost entirely focus on this domain. In particular, the thesis focuses on two main issues encountered when applying multimedia forensic tools in the wild, namely: i) the development of image forensic tools that can work under *dataset-mismatch* conditions; ii) the development of classification techniques that can operate in *open-set* scenarios.

The former issue is addressed in the first part of the thesis. With dataset mismatch, we refer to a scenario in which the system is tested with images belonging to the same classes/categories considered during training. However, the distribution of the test samples does not match that of the training data. For the synthetic image detection task, this corresponds, for instance, to recognizing fake synthetic images obtained with different generative models with respect to those considered during training, or generative

models of the same type but obtained using different generation parameters, training settings, and/or training procedures. In the thesis, we refer to this kind of dataset mismatch problem with the term *generalization*. In addition, the case where the test images are subject to post-processing—or, more generally, to a different post-processing pipeline with respect to the one considered during training—is also considered. In the following, we refer to the capability of coping with this kind of dataset mismatch as *robustness*. Many works in the recent literature show that methods developed for image forensic tasks, which perform well under matched conditions, suffer a severe degradation of their performance when tested under mismatched conditions. This thesis addresses the dataset mismatch problem by developing detectors based on semantic information, by relying on more general and robust features, and by incorporating within the detectors multi-level analysis and attention mechanisms. With regard to the former approach, we developed a robust AI-synthetic face detector that exploits eye clues to distinguish between real and fake images. The method relies on the analysis of inter-eye symmetries and inconsistencies and resorts to similarity learning to extract robust features. These semantic traces are usually preserved even in the presence of heavy processing operations. In this way, the method is inherently robust against global post-processing and local manipulations. Focusing on the detection of AI-manipulated images, where the images are partially manipulated (e.g., by adding flooded areas to a street image or by editing the facial attributes of real face images), we developed another semantic-related method that exploits the knowledge about the semantic nature of the manipulation to guide the training process. Specifically, we resorted to a hybrid framework for simultaneous detection/classification and localization, wherein localization is primarily used to aid the detection task by forcing the network to focus on the parts of the image that indicate the manipulation traces in the images. Focusing on the specific problem of classification of facial attributes editing, we enriched the hybrid classification with localization approach with multi-level analysis to develop a dedicated technique that relies on both global and local-level (patch-level) features combined with attentional feature fusion module with improved generalization and robustness against post-processing. The local and global features are first extracted from the full image and from specific image patches, and then merged by using an attentional feature fusion module.

The development of forensic techniques capable of working in open-set scenarios is the goal of the second part of the thesis. In an open-set setting, at operation time the systems are tested with samples belonging to different classes/categories with respect to those seen at training time. For the task of manipulation classification aimed at identifying the manipulation performed on images, this corresponds, for instance, to considering samples that have been subject to different types of editing with respect to those considered during training. In the case of synthetic image attribution, the open-set scenario takes into account the possibility that the test sample has been generated by an unknown generative model or architecture that has not been considered at training time. Common methods, having good performance in closed-set settings, are typically unreliable when tested in an open-set setting, since they misclassify unknown out-of-set samples as belonging to one of the known in-set classes, severely limiting their applicability in real-world applications.

In order to effectively operate in such a scenario, the forensics system's design and

methodology need to be reconsidered to avoid the assumption that all possible classes are predefined during training. Instead of focusing solely on making correct decisions within predefined known in-set classes, the system should prioritize adaptability and flexibility to accurately assess and handle out-of-set samples without making erroneous judgments, such as assigning predefined classes to out-of-set samples.

In particular, we considered two main approaches to develop a system capable of working in open-set scenarios: i) classification *with rejection class*; ii) *verification*. In classification with rejection class, we developed a forensic classifier to accurately classify in-set class samples, while at the same time revealing samples coming from out-of-set classes and refraining from providing a classification outcome - that would necessarily be unreliable - in this case. It is crucial to employ a proper rejection strategy to prevent any impact on the accuracy of out-of-set class samples. Recognizing the constraints encountered in dealing with unknown out-of-set class samples within the manipulation classification framework outlined in the first part of the thesis, we introduced an open-set framework coupled with a Vision Transformer (ViT) module [26] to comprehensively explore semantic features of in-set classes by means of classification and localization, enabling effective rejection of out-of-set samples characterized by lower prediction scores. Rejection is performed by considering several strategies and analyzing the model output layers. As for synthetic image attribution task, a few works deal with open-set image attribution within a semi-supervised framework [27, 28]. In addition to unlabeled in-set samples, unlabeled out-of-set class samples are also included in the training process, and a clustering strategy is used to separate out-of-set samples. However, this approach provides inaccurate predictions when encountering new out-of-set samples. In this thesis, we developed a new framework for multi-class classification with rejection class that exploits the concept of backdoor attacks [29]. Within this framework, rejection is achieved by purposely injecting class-specific triggers inside a portion of the images in the training set to induce the network to establish a matching between class features and trigger features. The behavior of the trained model with respect to triggered samples is then exploited at test time to perform out-of-set sample rejection, by injecting a class-specific image into its class samples with a new defined target class. The rejection mechanism operates by detecting the reaction of backdoor behavior.

The second approach we have developed adopts a Siamese Network-based verification framework to address the attribution task in an open-set scenario. The verification framework is naturally suited to work in an open-set scenario. To start with, a verification system can determine whether two images belong to the same class or not, regardless of whether the images are in-set or out-of-set. Alternatively, given an image and a claim about its class, the system can decide whether to support the claim or not, by exploiting the availability of one (or multiple) reference images from the claimed class. By verifying a claim on each known classes, the verification system can also be used as a classifier. It is worth observing that the verification approach has a significant advantage with respect to systems based on the introduction of a rejection class, which are not able to provide any information about out-of-set samples other than recognizing that they do not belong to the classes known by the system. In the second part of the thesis, the main focus is on synthetic image attribution due to the importance and practical relevance of the

open-set scenario for this task. However, methods have also been developed for the task of classifying facial attribute editing.

1.1 Overview and Contribution

The thesis is organized in two parts, corresponding to two different challenges related to synthetic image detection and attribution in the wild: the *dataset-mismatch* problem, and classification/attribution in an *open-set* scenario.

Then, the first part of the thesis focuses on the development of image forensic methods with improved performance under dataset mismatch. Before delving into the details of our research, Chapter 2 briefly introduces the most relevant generative methods and reviews the state-of-the-art for synthetic image detection. The chapter ends with a description of the datasets and the evaluation metrics used throughout the thesis to benchmark the performance of the techniques developed. In Chapter 3, we describe a semantic method for the detection of AI-synthesized face images, notably images generated by GANs, that relies on an eyes-based Siamese Network. This network takes two eye images as input to analyze asymmetrical details found in the eyes, common to various generative models, such as the shape of the pupils and iris details. The method shows good generalization to different generative models and a good robustness against image processing, as well as against splicing and rebroadcast attacks. Chapter 4 introduces a hybrid CNN-based architecture including both a classification and a localization branch, the latter being devoted to the localization of the image regions manipulated by the GANs. The main function of this branch is to induce the network to rely on the most relevant regions for the classification task. By focusing on the semantic manipulation regions identified by the localization branch, the classifier relies on artefacts within this region rather than on those belonging to regions that are not directly interested by the manipulation, thus improving robustness and generalization. We applied this method to detect fake images of climate change in the manipulated region that corresponds to flood areas. Finally, in Chapter 5, the hybrid framework is adopted to perform classification of facial editing in manipulated face images, in which different facial attributes have been edited. We first carried out some experiments on a public facial editing dataset with the architecture in Chapter 4 and then scale and improve it to a large dataset. The local and global features are first extracted from the full image and from specific image patches, and then merged by using an attentional feature fusion module.

The second part of the thesis is devoted to the development of solutions for open-set classification of image editing and, more importantly, for open-set synthetic image attribution. Chapter 6 introduces some background notions on open set recognition and reviews the state-of-the-art of synthetic image attribution. Chapter 7 presents our first attempt to address the open-set classification problem via a hybrid framework that resorts to ViT [26] and a simple yet effective sample rejection mechanism based on the analysis of the output logit scores. In Chapter 8, we describe a verification framework that relies on contrastive learning to address the problem of open-set attribution of synthetic images. We consider two different settings. In the first setting, the system determines whether the same generative architecture has produced two given images. In the second setting,

the system verifies a claim about the architecture used to generate a synthetic image, utilizing one or multiple reference images generated by the claimed architecture. Finally, in Chapter 9, we introduce a framework for open set attribution of synthetic images, named BOSC (Backdoor-based Open Set Classification), that relies on the concept of backdoor attacks to design a classifier with rejection option. BOSC works by purposely injecting class-specific triggers inside a portion of the images in the training set to induce the network to establish a matching between class features and trigger features. The behavior of the trained model with respect to triggered samples is then exploited at test time to perform sample rejection using an ad-hoc score.

1.2 Publications

The research activities carried out during the PhD studies resulted in the following publications:

- Journals
  - **J. Wang**, O. Alamayreh, B. Tondi, A. Costanzo, and M. Barni, "Detecting Deepfake Videos in Data Scarcity Conditions by Means of Video Coding Features", *APSIPA Transactions on Signal and Information Processing*, vol. 11, 2022.
  - **J. Wang**, B. Tondi, and M. Barni, "An eyes-based siamese neural network for the detection of GAN-generated face images", *Frontiers in Signal Processing*, vol. 2, 2022.
  - L. Abady, **J. Wang**, B. Tondi, and M. Barni, "A Siamese-based Verification System for Open-set Architecture Attribution of Synthetic Images", *Pattern Recognition Letter*, vol. 180, 2024.
  - **J. Wang**, B. Tondi, and M. Barni, "BOSC: A Backdoor-based Framework for Open Set Synthetic Image Attribution", *IEEE Transactions on Information Forensics & Security*, under review, 2024.
- Conferences
  - **J. Wang**, O. Alamayreh, B. Tondi, and M. Barni, "An Architecture for the detection of GAN-generated Flood Images with Localization Capabilities", in *2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, pp. 1–5, 2022.
  - O. Alamayreh, G. M. Dimitri, **J. Wang**, B. Tondi, and M. Barni, "Which country is this picture from? New data and methods for DNN-based country recognition", in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023.
  - **J. Wang**, B. Tondi, and M. Barni, "Classification of synthetic facial attributes by means of hybrid classification/localization patch-based analysis", in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023.

– **J. Wang**, O. Alamayreh, B. Tondi, and M. Barni, "Open Set Classification of GAN-based Image Manipulations via a ViT-based Hybrid Architecture", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 953–962, 2023.

– O. Alamayreh, **J. Wang**, G. M. Dimitri, B. Tondi, and M. Barni, "A Siamese Based System for City Verification", in *ECAI 2023*, pp. 69–76, 2023.

– **J. Wang**, B. Tondi, and M. Barni, "On the use of mixup augmentation for open-set synthetic image attribution", *WIFS 2024*, *under review*.

Part I

Detection of AI-generated Images under Dataset-mismatch Conditions

Abstract

This part of the thesis focuses on the development of methods for the detection of AI-generated images that can work under dataset-mismatch conditions. To achieve this goal, we present a pool of detectors exploiting semantic information to extract general and robust features whose effectiveness remains valid across different datasets. In particular, we first introduce a semantic-based method for distinguishing GAN-generated faces from real faces that relies on the analysis of inter-eye symmetries and inconsistencies. We also design a hybrid framework for simultaneous detection/classification and localization of edited or locally manipulated images, wherein localization is used to aid the detection task by forcing the network to focus on the most semantically-relevant parts of the image. The new architecture is then exploited to detect GAN-generated flood images, and to classify the facial attributes modified by GANs. With regard to the classification of facial attribute editing, we also develop a method that combines the hybrid approach with multi-level analysis and exploits attention mechanisms to rely on both global and local features.

Chapter 2

Introduction to Generative Models, Synthetic Image Generation and Detection

"Some of these things are true and some of them lie.
But they are all good stories."
Hilary Mantel

In this chapter, we provide an introduction to the most important and widespread generative tools and methods for image synthesis and manipulation. Then, we review the most relevant forensic methods for synthetic image detection. Even though AI generation and detection techniques encompass various image categories and domains, our focus remains on the literature related to the face image domain. As mentioned in the introduction, this is the primary domain considered in the thesis. Besides, due to the potential for misuse of human-related images, such as identity theft and fraud, the importance of facial recognition in security and forensics, and the ethical and societal implications of synthetic faces in digital media, most of the literature dealing with image forensics of AI-generated images focuses on this domain.

In this chapter, and throughout the thesis, we assume that the reader is familiar with the basic concepts of machine learning and DL and have some basic knowledge about discriminative models, notably CNNs.

The chapter is organized as follows: after a brief introduction to GAN and Diffusion Model (DM) in Section 2.1, Section 2.2 reviews generative methods based on the GAN and DM technology that have been utilized to generate face images from scratch and manipulate facial attributes such as age, gender, and expression. The literature about AI-generated image detection is reviewed in Section 2.3, considering both methods proposed for the detection of AI-synthesized images and AI-manipulated images. Finally, we present the face datasets that we used throughout the thesis to measure the performance of the techniques, and introduce the evaluation metrics.

2.1 Brief Introduction to Generative Techniques

Generative models are pivotal in computer vision and machine learning, enabling the creation of realistic data samples following a given distribution. Among the wide variety of techniques, Variational Autoencoders (VAE) [30], Energy-Based Models (EBM) [31], Generative Adversarial Network [32], Normalizing flows (NF) [33] and Diffusion Model [34] have gained significant attention. GANs and DMs are the most prominent among

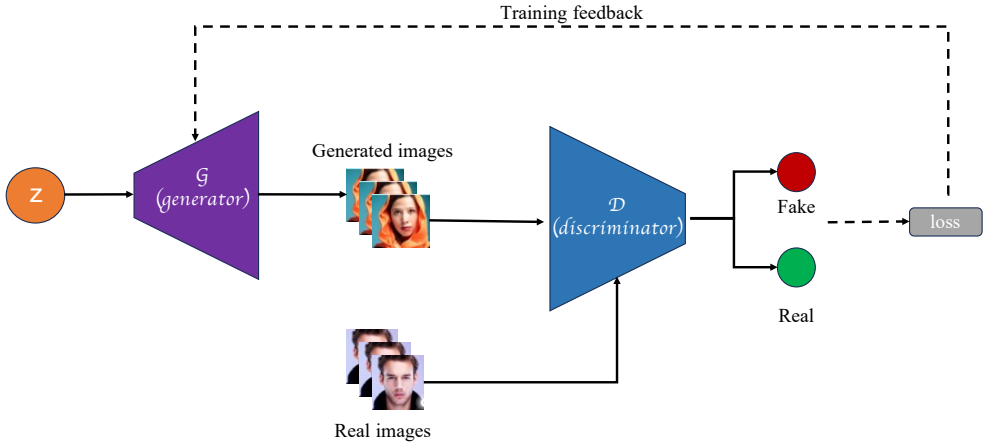


Figure 2.1 – General scheme of a Generative Adversarial Network, exemplified for the task of image generation.

these techniques, yielding outstanding performance. In this section, we provide a brief introduction to both GANs and DMs, describing the underlying principles, architectures, and training procedures.

2.1.1 Generative adversarial networks

Generative adversarial networks have revolutionized the field of generative modeling with their ability to produce high-fidelity data samples across various domains, which have been very useful in several applications. A GAN, first introduced by Goodfellow et al. in 2014 [1], consists of two primary components: a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$. The core idea to enhance the generator's capability of approximating the real data distribution is to train the generation and discrimination networks together in an adversarial fashion, where the discriminator is asked to distinguish between real and generated images, while the generator tries to defeat the discriminator. Figure 2.1 illustrates the general scheme of a GAN, focusing on the image domain. The generator takes random noise z as input and outputs images, while the discriminator compares the generated images with real images. In its most basic form, the optimization objective of the generator $\mathcal{G}$ is to minimize the following loss function,

$$\mathcal{L}_G(\mathcal{G}, \mathcal{D}) = \mathbb{E}_{z \sim p_z(z)} \left[\log \left(1 - \mathcal{D}(\mathcal{G}(z)) \right) \right], \tag{2.1}$$

where $p_z(z)$ represents the noise prior distribution and $\mathbb{E}_{z \sim p_z(z)}$ denotes the statistical expectation taken over the random variable z distributed as $p_z(z)$. On the other hand, the goal of the discriminator $\mathcal{D}$ is to minimize the loss function,

$$\mathcal{L}_D(\mathcal{G}, \mathcal{D}) = -\mathbb{E}_{z \sim p_z(z)} \left[\log \left(1 - \mathcal{D}(\mathcal{G}(z)) \right) \right] - \mathbb{E}_{x \sim p_x(x)} \left[\log \left(\mathcal{D}(x) \right) \right], \tag{2.2}$$

where x denotes an input image drawn from the distribution p_x.

Starting from this formulation, many variants have been proposed (see Section 2.2), and different training procedures have been implemented to improve the performance of the generation and also to solve the training instability issue that initial GAN models were subject to [35].

2.1.2 Diffusion models

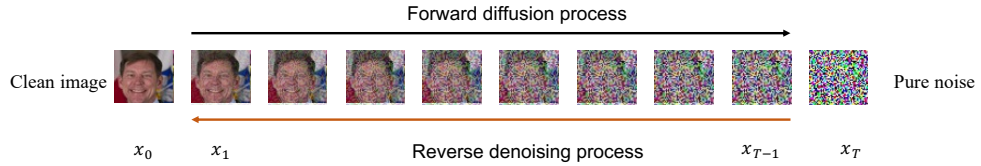


Figure 2.2 – Illustration of the denoising diffusion process in DMs.

Diffusion Models, initially introduced by Sohl-Dickstein et al. [34] and further refined by Ho et al. [10], have attracted the interest of researchers in the last years for their ability to generate extremely realistic and high-resolution images with improved diversity with respect to GANs, and for their stable training process.

DMs have shown remarkable results in generating high-resolution images with fine details and have been applied to image synthesis and manipulation tasks. DMs are a class of probabilistic generative models that are instructed to learn to reverse a process that gradually degrades the structure of the training data. More specifically, the training procedure involves two key phases: the forward diffusion process and the backward denoising process (see Figure 2.2). In the forward diffusion phase, low-level noise is iteratively added to each input image, with the noise level varying at each step. The training image undergoes progressive degradation until it is transformed into a pure noise image. The backward denoising phase aims at reversing the forward diffusion process. This iterative procedure is applied in reverse order to remove the noise and re-obtain the original image. Images are then generated by gradually reconstructing them from random white noise. The noise subtracted at each time step of the reverse denoising process is estimated using a neural network, typically based on a UNet architecture [36].

A common DM model is the Denoising Diffusion Probabilistic Model (DDPM) [10], which models the diffusion process as a Markovian process, as follows

$$p(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t} \cdot x_{t-1}, \beta_t \cdot \mathbf{I}), \forall t \in \{1, \cdots, T\}, \tag{2.3}$$

where T is the number of diffusion steps, $\beta_1 \cdots \beta_T \in [0, 1)$ are hyperparameters representing the variance schedule across diffusion steps, $\mathbf{I}$ is the identity matrix having the same dimension of the input images x_0, and finally $\mathcal{N}(x; \mu, \mathbf{\Sigma}_c)$ is the normal distribution of mean μ and covariance $\mathbf{\Sigma}_c$ that produces x. In the reverse denoising process, the image can be generated from a sample $x_T \sim \mathcal{N}(0, \mathbf{I})$ via,

$$p(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_(x_t, t), \mathbf{\Sigma}_c(x_t, t)), \tag{2.4}$$

where the mean $\mu(x_t, t)$ and the covariance $\mathbf{\Sigma}_c(x_t, t)$ are predicted by a neural network (i.e., UNet [36]) that takes the noisy image x_t and the embedding at time step t as input. Training is performed by optimizing the variational bound for the negative log-likelihood (the reader may refer to [10] for the details). Two variants of diffusion models considering different score functions for the optimization are the noise-conditioned score network (NCSN) and stochastic differential equations (SDE) [37].

2.2 Synthetic Image Generation

In this section, we provide an overview of the main literature on AI-based image synthesis and manipulation, focusing on the methods dealing with face image synthesis and facial attribute manipulation.

2.2.1 Face image synthesis

GAN-synthesized face images. Early attempts to use GANs for image generation were limited to the generation of high-quality, low-resolution images [2, 31, 38]. Later, techniques have been proposed that are capable of generating realistic, high-quality, large-resolution images. BigGAN [39], for instance, employed techniques such as data truncation and orthogonal regularization to stabilize the training process for large-scale GANs and were able to generate realistic images with a resolution up to 512×512. ProGAN [4] was able to reach an image generation resolution of 1024×1024 by implementing a progressive training approach, where the model is first trained to learn to generate low-resolution images, then the resolution is progressively increased. Later, the same authors of ProGAN proposed a StyleGAN [5] method to further improve the quality of the generated high-resolution images, by utilizing an alternative generation architecture borrowed from the style transfer literature. Specifically, in addition to progressively increasing the resolution of the generated images, as done by ProGAN, StyleGAN incorporates 'style' features in the generative process, where a random latent code is mapped into an intermediate latent space enabling some controlled image modifications. The quality of StyleGAN images has been further improved by the StyleGAN2 model [6], that redesigns the normalization used in the generator. Later on, NVIDIA released a new GAN architecture, named StyleGAN3 [7], which solves the problem of 'texture sticking' (a.k.a., aliasing) in the images generated by StyleGAN2. Such a result is achieved introducing some architectural changes guaranteeing that unwanted information does not leak into the hierarchical synthesis process. The StyleGAN series, and StyleGAN2 in particular, has had great success, and many AI-based tools rely on it.

Inspired by the success of ViTs in various computer vision tasks [40], Zhao et al. [41] successfully integrated ViTs into the GAN framework for high-resolution image generation, denoted as HiT. A couple of years ago, Zhang et al. [39] proposed StyleSwin that leverages Swin transformers [42] as the basic building block of the generator, which improves the generation quality by taking advantage of local attention modules. On the other hand, Taming Transformers [8], a.k.a. VQGAN, where VQ stands for Vector Quantization (Taming Transformers can be seen in fact as a variant of Vector Quantized Variational

Autoencoders (VQVAE)) [43], combined the power of ViT architectures with the convolutional approach for image synthesis. In particular, the combination of encoder-decoder architectures and transformer-based modules allows the generation of high-quality images with coherent image structures.

DM-synthesized face images. The pioneering DDPM work [34] demonstrated the ability of DM models to generate high-quality samples with high levels of detail. Subsequently, various improvements were made to DDPM models, in two main directions: accelerating the sampling procedures and improving the image quality. The former is the main challenge applying DDPM models to real-world applications. A step in this direction was made in [10], where the denoising diffusion implicit model (DDIM) was proposed. In such a work, the Markov forward process used in [10] is replaced with a non-Markovian one, leading to a faster sampling procedure with a negligible impact on the quality of the generated samples. With regard to the latter direction, Dhariwal et al. [44] proposed a method that introduces a few architectural changes to improve the Fréchet inception distance (FID) [45] of the synthetic images by exploiting classifier guidance, namely, a strategy that uses the gradients of the classifier to guide the diffusion process. Rombach et al. [46] proposed the Latent Diffusion Model (LDM) performing noise diffusion and denoising in the latent space by means of an Autoencoder. Another technique, proposed by Vahdat et al. [9], is the Latent Score-based Generative Model (LSGM). This is a score-based generative model that performs training in the latent space, relying on the VAE framework. This method has recently demonstrated impressive results in terms of both sample quality and distribution coverage.

Some examples of face images generated by some of the GAN and DM techniques described above, and the evolution of the quality of synthetic face images over the last decade, are shown in Figure 2.3.

2.2.2 Facial attribute manipulation

Besides synthesizing images from random noise, often referred to as generation from scratch, research has also increasingly focused on the development of methods for AI-based image manipulation. With reference to the face domain, several methods for facial attribute editing have been proposed. In facial attribute manipulation, single or multiple attributes of the faces are edited, such as the color of the hair, the age, the physical appearance, etc. GAN-based methods like StarGAN [47], AttGAN [48] and STGAN [49] were first employed for facial attribute editing.

After these initial attempts, inspired by the superior performance of the StyleGAN series [5–7] in synthesizing high-resolution and high-quality images, the use of StyleGAN methods to edit the real images has attracted upsurging attention. Attempts have been made towards better disentanglement in the latent space [11, 50–52] (a latent representation is perfectly disentangled if each latent dimension controls a single visual attribute [53, 54]). In InterFaceGAN [11], Shen et al. pointed out that it is possible to utilize the disentangled features encoded in the latent space to edit semantic attributes through a linear subspace projection. To enable more precise local manipulation, Shi et

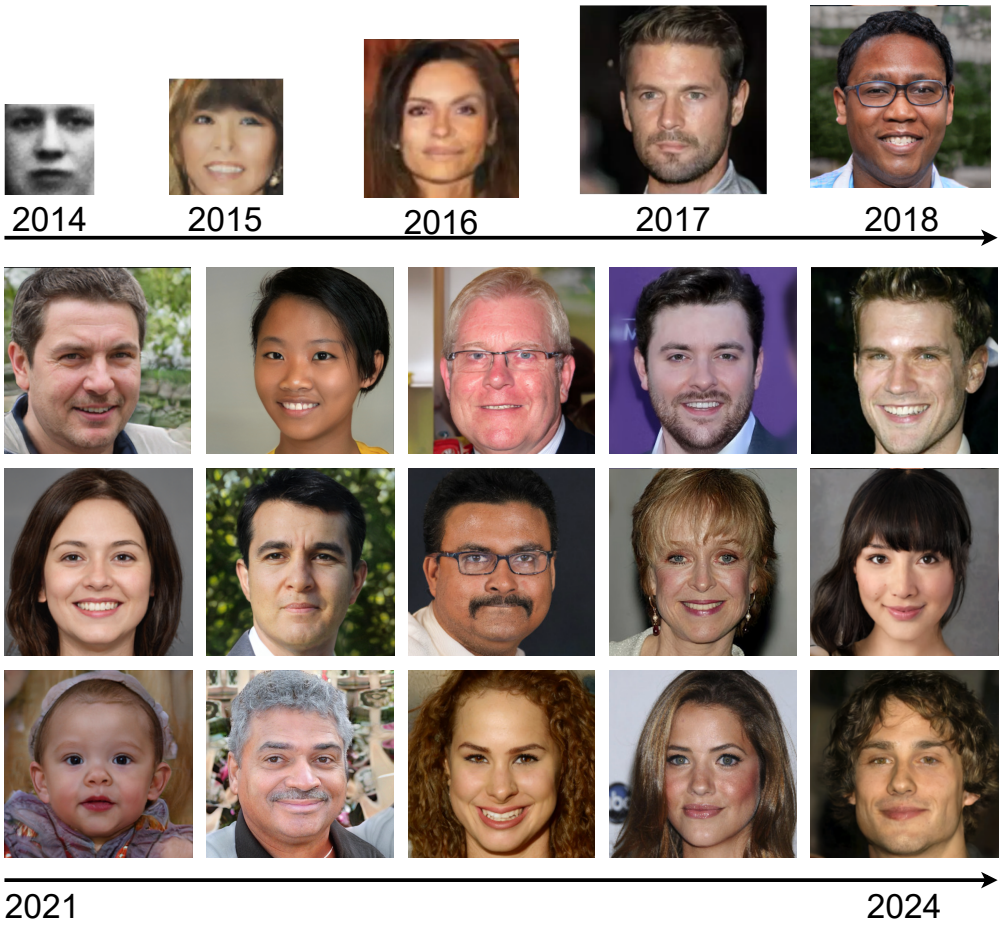


Figure 2.3 – The evolution of generated image resolutions from 2014 to 2024 and examples of synthesized face images by GAN [1], DCGANs [2], CoGAN [3], ProGAN [4], StyleGAN [5], StyleGAN2 [6], StyleGAN3 [7], Taming transformer [8], LSGM [9] and DDPM [10] (each column, from left to right).

al. presented SemanticStyleGAN [52], which encodes sampled codes into semantically local representations in an intermediate latent space [5] and generates local features for each latent representation. These features are finally fused and utilized by a rendering network for image generation, together with a semantic segmentation mask. In addition, Patashnik et al. proposed StyleCLIP [12], a method that allows for image editing by conditioning StyleGAN on a given text description. Specifically, StyleCLIP mainly uses the Contrastive Language-Image Pre-training (CLIP) model [51] to edit the latent code through the user input language description to achieve the editing purpose. Roich et al. [55] improved the tradeoff between editing ability and image quality by employing pivotal tuning. This approach fine-tunes the generator of StyleGAN2 based on an initial inverted latent code that serves as a pivot. Moreover, Xu et al. [56] combined a trans-

former with a dual-space GAN to develop a dual-space editing and inversion strategy that can provide additional editing flexibility. Finally, Alaluf et al. [57] first proposed a method for facial attribute editing based on StyleGAN3.

Several image editing methods based on diffusion models have also emerged. Notably, these models are not specialized for facial attribute editing but are general methods for local manipulation of natural images (including face images). Generally, DMs can be used to process images by training the denoising autoencoders by conditioning them to different inputs, such as semantic maps, text, and images. For instance, LDM [46] can be used for image inpainting and object removal by concatenating spatially aligned conditioning information to the input of denoising autoencoders. However, researchers have looked for more flexible ways to edit the images with DMs using conditioning. Given a real image masked by a stroke-painting image, SDEdit [58] gradually projects the image to the manifold of natural images in two steps: i) perturbing it with Gaussian noise and ii) progressively removing the noise by simulating the reverse SDE. On the other hand, Yang et al. [59] investigated exemplar-guided image editing for more precise control with arbitrary mask input. The masked region is filled with the input example image by leveraging the classifier-free guidance to increase the similarity to the example image. In 2022, the release of the Stable Diffusion model (SDM)[1] marks a breakthrough in AI capability of manipulating images. The Stable Diffusion model is an improved version of LDM that uses a frozen CLIP ViT-L/14 text encoder to condition the model on text prompts. After this model's release, several image editing methods were proposed, relying on text-guided manipulation. Blended Diffusion [60] combined DDPM with a pre-trained language-image model CLIP to perform local image manipulation guided by a mask image. Later on, in order to accelerate the inference, Avranhami et al. [13] moved the diffusion process into VAE's latent space by combining the CLIP pre-trained model with LDM. Text-guided iterative image editing has also been investigated recently by Pan et al. [61] and Joseph et al. [62]. Valevski et al. [63] and Kawar et al. [64] proposed two methods, namely, UniTune and Imagic, to edit the images based on a textual edit description without taking any specific mask as input. Finally, it is worth mentioning the release of DALL-E3 [20] and Adobe FireFly[2], two commercial generative AI applications built on the DM framework. Both techniques showcase superior image synthesis and editing capabilities, pushing forward the boundaries of what is achievable with AI-based image content generation.

Figure 2.4 shows examples of images edited with some of the methods introduced above, based on StyleGAN2 and DDPM, which represent the most common and widely used methods for face editing.

2.3 Synthetic Images Detection

As mentioned in the introduction, the detection of AI-generated images has attracted a lot of attention in recent years due to the tremendous implications that the diffusion of fake images may have (misinformation, privacy violation, etc.). In this section, we review

[1] `https://github.com/compvis/stable-diffusion`
[2] `https://firefly.adobe.com/inspire/images`

Figure 2.4 – Examples of edited face images by InterfaceGAN [11] (Top line), StyleCLIP [12] (Middle line) and Blended Latent diffusion [13] (Bottom line).

the most relevant methods proposed in the forensic literature to discriminate between AI-generated images and real ones. In this section, we focus on the methods for the detection of AI-synthesized images in the case of fully generated images. The methods dealing with AI-manipulated images are reviewed in Section 2.3.2. Once again, we stress that most of the methods proposed in the literature focus on face images.

2.3.1 Detection of AI-synthesized images

GAN generated image detection. Early GAN-generated images could be easily detected due to imperfect generation capability of generative models. Li et al. [65] experimented with three strategies (Rank-based quality score [66], Inception score [67] and VGG features [68]) by measuring the quality of the generated images, showing the best performance with VGG features for generated image detection. They also showed that the discriminator trained together with the generator works well to detect image generated during the same or earlier epochs, but starts failing as the training goes on. In addition, several works explored saturation clues [69], landmark location [70], and high-frequency information [71–74] with good detection performance. However, these works did not discuss the generalization capability in detecting an image generated by a new generator and lacked robustness against image post-processing. Their application *in the wild*, hence, is problematic.

One approach to enhance generalization and robustness of the detectors is to leverage

deep CNN architectures by incorporating appropriate augmentation techniques during training, as demonstrated in [75, 76]. In [23], the authors found that effective robustness can be attained through basic augmentation methods like compression and blurring, even training on a single generative architecture (ProGAN). Subsequent studies [16,77,78] have further confirmed the significance of dataset diversity in enhancing the generalization capabilities. Moreover, utilizing large datasets to pre-train the models has been identified as an important factor. While extreme augmentation techniques yield only marginal improvements in robustness, they enhance generalization to unseen models [16]. Other works preserve information integrity throughout training and testing, particularly within the layers closest to the input. Chai et al. [79] emphasized using patch-based classifiers and avoiding image resizing to prevent the loss of subtle traces inherent in the generation process. Additionally, Ju et al. [80] enhanced patch-based analysis by incorporating global spatial information extracted from the entire image.

In order to address the transferability limitation when trying to detect image manipulations in real world, Cozzolino et al. [81] introduced Forensic-Transfer, a learning-based forensic detector that adapts well to novel manipulation methods, and can handle scenarios where only a handful of fake examples are available during training. The method can detect newly generated images by retraining the model with a few novel samples. However, performance on previous data may drop. On the other hand, Marra et al. [82] proposed using incremental learning to detect and classify GAN-generated images, by letting them continuously evolve as new types of generated data appear. Jeon et al. [83] introduced the Transferable GAN-images detection framework, where a teacher model is initially trained on the source domain, and the student model is trained by integrating data from both the source and target datasets, with weight variation constrained to maintain consistency with the starting point. The success of this kind of method relies on the availability of example images from the new architectures, which may not always be possible, particularly in challenging scenarios.

Another class of methods is based on the exploitation of artificial clues. In addition to exploring representative features fully based on CNNs, these methods also rely on some specific forensic traces identified by carefully observing the synthetic images. Several works found that synthetic images have distinguishable features in color space [84–86] and achieved good generalization performance when testing on newly generated images. Another golden rule that applies equally well to improving generalization performance is considering frequency features [72, 87–93]. In fact, synthetic images tend to exhibit clear traces of their origin in the Fourier domain due to the up-sampling operations typical of the synthesis network. Even when such features are absent, synthetic images differ significantly from natural images at medium-high frequencies. Zhang et al. [87] proposed to simulate such artifacts by reconstructing the real images with a generator. A detector can be trained using real images and reconstructed images that simulate the fingerprints in the frequency without the need to actually generate synthetic images. Tanaka et al. [89] proposed to first enhance the frequency information and feed it into ResNet50 directly for detection. Moreover, Yu et al. [91] combined the color difference and frequency information at the input of the CNN detector.

At an even higher semantic level, aiming at improving the detection results' inter-

pretability, some methods exploit the presence of semantic incongruence in the synthetic images, such as, for instance, face asymmetries [94]. For instance, in [95] and [96], the inconsistencies around the eyes were exploited for the detection. In [97], Chen et al. proposed to use facial segmentations (including nose, mouth, eyes, face and background) as network inputs and to fuse the results obtained on each segment with a multi-scale attention module. In [98], the authors presented a new CNN-based detector, called Gram-Net, that leverages global image texture representations to improve the generalization and robustness of GAN image detection.

Detection of images generated by diffusion models. In the last couple of years, methods focusing on the detection of DM-generated images have started appearing, especially the generalization capability of the GAN-generated image detector to DM-generated images. We point out that most of the literature in this area is very recent (published less than one year ago), hence it has not been considered in the comparisons reported in the first part of the thesis. The first studies tried to understand the difference between the detection of GAN and DM images [99–101]. For instance, Ricker et al. [99] evaluated the performance of the tools designed for GAN-generated images on images generated by diffusion models and found that a detector trained on DM-generated images is capable of detecting images from GANs, while the opposite is not true. Similarly, Corvi et al. [100] studied the existence of artificial fingerprints in DM images and evaluated the effectiveness of the GAN detector developed in [23] for the detection of DM images. They demonstrated that detectors trained solely on GAN images perform poorly on DM images. Including a DM model in the training phase can aid the detection of images generated by similar DM architectures but not those generated by different architectures since they usually have different fingerprint artifacts. This emphasizes that generalization remains critical, even in the case of DMs. Epstein et al. [101] investigated the generalization capability of detectors trained for DM detection and the impact of the architectures' selection by exploiting progressive training. The authors observed that integrating the LDM during training leads to significant enhancements in the generalization to other types of DMs. Overall, these methods confirm that DM-generated images exhibit distinguishable features in the frequency domain compared to natural images. However, these patterns differ from those observed in GAN-generated images, and it is hard to explain them using CNN networks. Instead of taking advantage of CNNs, Bammey et al. [102] trained a histogram-based gradient boosting tree classifier using 135 potential magnitude peaks of Fourier coefficients of the high-pass residual of the image, which shows good robustness against JPEG compression and generalization to unseen diffusion models. Finally, based on the observation that diffusion-generated images can be approximately reconstructed by a diffusion model while real images cannot, Wang et al. [103] proposed a method called DIffusion Reconstruction Error (DIRE) that measures the error between an input image and its reconstruction counterpart obtained with a pre-trained diffusion model.

2.3.2 Detection and localization of AI-based manipulation

The methods mentioned in the previous section address the problem of detecting whether an image has been generated by an AI model or not. In the case of AI-manipulated images, where a part of a (natural) image is manipulated via AI, it is also interesting to localize the manipulation, to support and explain the detection results. Nevertheless, the development of methods for manipulation localization is a much more recent effort, so only a few methods have been dedicated to this task.

A first technique focusing on the localization of manipulated regions was proposed in [104], by taking paired facial landmarks images and face images as input. Huang et al. [105] proposed a *FakeLocator* model to localize the texture artifacts introduced by the up-sampling steps usually applied by GAN architectures, where a face parsing map is utilized in the feature domain acting as an attention map. By focusing on a specific task, i.e., face swapping, [106] proposed using discriminative feature maps extracted from a facial expression recognition (FER) framework [107] to facilitate the classification and segmentation tasks, which provides information about the facial regions that encode the expression information. In [108], in addition to the input image, the network takes the predefined average map of various manipulation masks as input, and estimates the weight parameter that generates the manipulation attention map from the average manipulation map. These methods exploit various additional information (such as landmarks and FER features) to facilitate the localization tasks and enhance generalization. However, their reliance on such features complicates their application to tasks other than in the face domain.

An alternative approach has been proposed by Nguyen et al. in [109], with the proposal of a multi-task learning approach to detect manipulated images while simultaneously pinpointing the manipulated regions for each query, coupled with a reconstruction task. Based on the binary classification, features in the latent space are split into two channels (0 for real and 1 for fake) for the reconstruction and segmentation tasks. This approach may be limited in its application to multi-class classification tasks due to the resulting increase in latent space feature dimensions, leading to higher computational costs. The Face *X-ray* method proposed in [110] analyzes the specific artifacts introduced when two face images are blended and uses this information to guide the training of the detector. Likewise, Zhao et al. [111] proposed a pair-wise self-consistency learning module to exploit the local irregularities in fake face videos. Again, these features are specific to facial manipulations and cannot be used in other domains. Zhao et al. [112] presented a general method that considers a multi-scale attention mechanism to improve the detection accuracy and identify the image region that mainly contributes to the detection of face images. While the concept of detection and localization may seem straightforward, implementing it across various tasks still requires significant effort.

2.4 Datasets and Evaluation Metrics

In this subsection, we summarize all the face datasets used in the thesis and define the metrics we used to assess the performance. We first introduce the GAN detection dataset

Table 2.1 – Datasets used for training and testing.

Datasets	CelebA_HQ	FFHQ	StyleGAN2	ProGAN	StyleGAN3	Print&Scan
Class	Pristine	Pristine	GAN	GAN	GAN	Pristine,GAN
Training	63,000	27,000	90,000	-	-	-
Test	3,000	7,000	10,000	10,000	10,000	10,000, 10,000

(GDD), that we used for the task of synthetic image detection, and also for measuring their generalization and robustness. Then, we present the portrait face manipulation dataset (PFMD) and its extended version, named facial attribute editing datasets (FAED), that we used for the experiments of classification of facial manipulation. Finally, we present the synthetic image attribution dataset (SIAD) that we used for the task of synthetic image attribution in open set.

2.4.1 GAN detection dataset (GDD)

The GDD is composed of images generated by ProGAN [4], StyleGAN2 [6] and StyleGAN3 [7]. As we pointed out in Section 2.2.1, despite the similar name, the architecture of StyleGAN3 is very different from StyleGAN2 (which is similar to the original version of the StyleGAN model), thus complicating the generalization of detectors trained on StyleGAN2 to the case of synthetic images generated by StyleGAN3. More in detail, the following datasets of faces were included in GDD (a summary of the datasets is provided in Table 2.1):

- A collection of 100,000 real face images: 30,000 images are taken from the Large-Scale CelebFaces Attributes High Quality (CelebA-HQ) dataset [4], while the remaining 70,000 images come from the Flickr-Faces-HQ (FFHQ) dataset [5] [3]. CelebA-HQ dataset is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations. CelebA contains a large quantity of very diverse images, with rich annotations, including 10,177 identities, 202,599 face images, five landmark locations and 40 binary attribute annotations per image. Its high-quality version, CelebA-HQ, with 30,000 high-resolution (1024×1024) images, was introduced in [4]. FFHQ comprises 70,000 high-quality PNG images with a resolution of 1024×1024 pixels. It exhibits substantial variations in terms of age, ethnicity, and image background, offering a diverse range of facial characteristics. Additionally, the dataset includes a comprehensive selection of accessories such as eyeglasses, sunglasses, hats, etc., ensuring comprehensive coverage of potential attributes and features. We used 90,000 real images for training (5,000 of those are left for validation), and 10,000 images for the tests. The same 3:7 proportion of CelebA-HQ and FFHQ images were present in training and test datasets. Therefore, among the images in the training set, 27,000 images come from CelebA-HQ and 63,000 from FFHQ, while the test set consists of the remaining 3,000 CelebA-

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

Figure 2.5 – Real images from CelebA-HQ (Top) and FFHQ (Bottom) datasets.

HQ and 7,000 FFHQ images. Some examples of real images from the CelebA-HQ and FFHQ datasets are shown in Figure 2.5.

- A dataset of StyleGAN2 fake images consisting of 100,000 images in total, out of which 90,000 images were used to train the models (85,000 for training and 5,000 for validation), and 10,000 images for the tests. We used the officially released code[4] to generate synthetic faces with different generation parameters to increase the diversity of the dataset. More specifically, we considered several values of the truncation parameter of the network and generated 10,000 StyleGAN2 [6] faces for each value in the set {0.3, 0.4, 0.5, 0.6, 0.65, 0.7, 0.75, 0.8, 0.9, 1}. Both the training and test sets contain images generated with all the truncation parameters in equal proportions.

- A collection of 10,000 images generated by ProGAN and 10,000 images generated by StyleGAN3. We used this dataset for the generalization tests.

- A large-scale dataset of printed and scanned, pristine and GAN images, called VIP-Print [17], consisting of 10,000 recaptured real (from the FFHQ dataset) and 10,000 recaptured GAN (StyleGAN2) images, obtained by printing the digital images and then scanning them. More details can be found in [17]. The printing and scanning operation can be used to hide the traces of image manipulation, then, arguably, also the synthetic nature of images. In the following, we refer to this dataset as the Print&Scan image dataset.

[4] `https://github.com/NVlabs/stylegan2`

Table 2.2 – Portrait face image manipulation dataset used for the analysis. 'Low' indicates that the manipulation is performed with a reduced edit strength.

Edit types	Purpose	Low version	Tools	Remark
"none"	Train&Test	No	PTI	Reconstructed image (with no editing)
"Smile"	Train&Test	Yes	InterfaceGAN	Smile added or enhanced
"Not smile"	Train&Test	Yes	InterfaceGAN	Smile removed or reduced
"Young"	Train&Test	Yes	InterfaceGAN	Face is modified to appear younger
"Old"	Train&Test	Yes	InterfaceGAN	Face is modified to appear older
"surprised"	Train&Test	No	StyleCLIP	Face is modified to depict a surprised expression
"purple hair"	Test	No	StyleCLIP	Hair is modified to have a purple color
"angry"	Test	No	StyleCLIP	Face is modified to depict an angry expression
"taylor swift"	Test	No	StyleCLIP	Face shape and features modified to appear similar to Taylor Swift
"original"	Training&Test	No	-	The original (unedited) image
"reference"	Training&Test	No	-	Another image of the same face identity

Figure 2.6 – Examples of edited images in the portrait-style image dataset [14]. 'Low' indicates that the manipulation is applied with a reduced edit strength.

2.4.2 Portrait face manipulation dataset (PFMD)

The portrait-style image dataset includes edits applied to portrait-style frontal face images and full-scene in-the wild images that may include multiple (i.e., more than one) face per image [14]. To get the edited images, the latent code of real images with respect to the generator [55] is extracted by means of Pivotal Tuning Inversion (PTI) [55]. Then, the image attribute is manipulated by InterfaceGAN [11] and StyleCLIP [12] by using the learned latent code and a target StyleGAN2 generator to achieve the desired modification of the attribute.

The real images were sourced from the high-quality subset of the CelebA-HQ dataset (see Section 2.4.1 for an introduction to this dataset). The dataset is split into two partitions, that can be used for training and testing, for a total of 6,846 images and 7,644 images in the training and testing partitions, respectively. Only identities that appear at least twice (i.e., there are at least two images of a given identity) in the CelebA-HQ dataset were considered. Each sampled image in the training partition is manipulated to produce five separate instances, in combination with the original image, a reconstructed image with no editing and a different image of the same face identity as reference, for a total of 8 image versions, original, reference, reconstructed, and 5 manipulations. The testing partition includes the same types of manipulated attributes present in the training partition. Besides, there are additional examples for the *smile*, *not smile*, *young*, and *old*, labels produced with a lower editing strength. These images are referred to as 'Low' versions, while the others - whose edit strength is matched with the training one - are indicated as 'High'. In the testing partition, there are also three different types of editing that are not present in the training partition, *purple_hair*, *angry* and *taylor swift*. An overall description of the dataset is given in Table 2.2. Examples of edited images for a given portrait image are shown in Figure 2.6.

2.4.3 Facial attribute editing dataset (FAED)

The FAED is an extended version of PFMD that consists of 18 editing types (10 more edited facial attributes than the PFMD) obtained by using the same manipulation procedure exploiting with PTI [55] to get the latent code, and InterfaceGAN [11] and StyleCLIP [12] to do the manipulation of the attribute. This is a dataset we built ourselves to better evaluate the performance of the methods with a more large scale and rich dataset. Examples of all the types of manipulations are shown in Figure 2.7.

More details on the manipulations are given in the following. In InterfaceGAN, the latent code z of the image is edited by $z_{edit} = z_I + \alpha_I \cdot n_a$, where $n_a \in \mathbb{R}^d$ is the unit normal vector of a hyperplane orthogonal to the attribute direction, that separates the space in two regions, for instance, *smile* and *non-smile*, and α_I is a parameter controlling the editing strength. For instance, in the *old* class, the higher the value of α_I, the older the person will look like. We edited the images with four editing types, *smile*, *non-smile*, *young* and *old*. To evaluate the capability of classifying the same editing with various editing strengths, the editing is applied with different α_I values, ranging in the interval $[2, 5]$. Figure 5.8 shows an example of smile and non-smile editing with different strengths.

On the other hand, StyleCLIP edits the latent code through a language description.

Figure 2.7 – Image examples of original, '*none*' and of the 18 attribute editing.

In this case, the latent code z_I is further processed by three separate mapping functions to generate residuals that are added to the latent code z_I to yield the target code. Then the generator decodes the target code and the manipulation is supervised by a CLIP loss to the input text. In our work, we considered 14 semantic edits provided by StyleCLIP, including *Angry*, *Surprised*, *Afro*, *Purple hair*, *Curly hair*, *Mohawk*, *Bobcut*, *Bowlcut*, *Taylor Swift*, *Beyonce*, *Hilary Clinton*, *Trump*, *Zuckerberg* and *Depp*.

Overall, we took 2,933 images from the CelebA-HQ dataset and edited the images by adding one of 18 attribute types by using InterfaceGAN and StyleCLIP, including 2,000 training, 200 validation and 733 test images. We left the images edited by InterfaceGAN with $\alpha_I = \{2, 3\}$ as a mismatched case to test the performance on unseen data (see in Table 2.3).

For the experiments in Chapter 7, we considered an extended version of the FAED, obtained by considering 3,000 additional real-face images from the CelebA-HQ dataset per class, end editing them thus getting a total 5,933 edited images in 18 facial attributes. In the following, we refer to this version as FAED_v2.

2.4.4 Synthetic image attribution dataset (SIAD)

For the open-set synthetic image attribution task, we collected 10 generative architectures, including including: i) GANs: BigGAN [113], BEGAN [114], ProGAN [4], StyleGAN2 [6], StarGANv2 [115], StyleGAN3 [7]; ii) diffusion models (DM): DDPM [10], Latent Diffusion [46], LSGM [9]; and iii) transformers: Taming transformer [8]. We considered the officially released models trained on FFHQ [5] and CelebA datasets [4], considering different training strategies (for the case of StyleGAN2 and DDPM), different sampling factors (for Latent Diffusion) and configuration parameters (StyleGAN3). Specifically, for StyleGAN3, we considered the two released optimal configurations according to FID scores, denoted as "t" and "r" [7], trained on different real-world datasets and at different

Table 2.3 – Overview of the datasets we used in our experiments.

Table	Parameter α_I	Train&Validation&Test =20,00&200&738	Edit facial attributes
PTI	-	Train&Validation&Test	None (Reconstructed)
InterfaceGAN	5	Train&Validation&Test	Smile, Not smile, Old, Young
	4	Train&Validation&Test	
	3	Test	
	2	Test	
StyleCLIP	-	Train&Validation&Test	**Expression**: Angry, Surprised **Hair style**: Afro, Purple_hair, Curly_hair, Mohawk, Bobcut, Bowlcut **Identity change**: Taylor_swift, Beyonce, Hilary_clinton, Trump, Zuckerberg, Depp

Figure 2.8 – Examples of synthetic images from the 10 architectures considered for the experiments in this chapter.

resolutions. For StyleGAN2, we utilized the best-performing configuration based on the FID quality score, which is referred to as configuration "f" [6]. For LSGM, Taming transformers and Latent diffusion models, the resolutions of the images are 256 × 256, while for StyleGAN models, the images are generated with both 256 × 256 and 1024×1024 resolution. For each architecture, we collected 50k images.

For the experiments in Chapter 7 a subset of 5 architectures is considered for the tests, namely, StyleGAN2 [6], StyleGAN3 [7], Taming Transformer [8], Latent Diffusion [46] and LSGM [9]. We refer to this dataset as SIAD. while the extended version of the dataset with the 10 architectures, used in Chapter 8 and 9, is denoted to as SIAD_v2.

2.4.5 Evaluation metrics

We used Accuracy (ACC) to evaluate the performance of the detection and classification systems in a closed-set scenario. Formally,

$$ACC = \frac{\left| \{x | x \in D_t, \hat{y} = y\} \right|}{|D_t|}, \tag{2.5}$$

where $\hat{y}$ and y are the predicted and ground truth label for sample x, $|\cdot|$ indicate the cardinality of the set, and D_t is the test dataset. The Accuracy is measured using the default threshold ($\tau = 0.5$) resulting from the trained network, which is used to get the decisions $\hat{y}$.

For the synthetic image detection task (binary task), we also plot the Receiver Operating Curve (ROC) and the Area Under this curve (AU-ROC). Formally, let the positive (negative) event be the synthetic (pristine) class, we have the True Positive Rate TPR and False Positive Rate (FPR) defined by

$$\begin{aligned} \mathrm{TPR}(\tau) &= \frac{\left| \{x | x \in D_{t,s} \wedge p_2 \geq \tau\} \right|}{|D_{t,s}|}, \\ \mathrm{FPR}(\tau) &= \frac{\left| \{x | x \in D_{t,r} \wedge p_2 \geq \tau\} \right|}{|D_{t,r}|}, \end{aligned} \tag{2.6}$$

where $p \in \mathbb{R}^2$ is the network output probability vector, and p_2 is the probability score associated to the synthetic class, τ is the threshold, and $D_{t,r}$ and $D_{t,s}$ are the pristine and synthetic image dataset respectively. In some cases, the threshold τ is set by fixing the FPR to 5% and considering the probability of correct detection at the fixed FPR, indicated as Pd@5%. The True Negative Rate (TNR) can be obtained as $\mathrm{TNR} = 1 - \mathrm{FPR}$.

Open-set performance was evaluated in terms of the capability the system of rejecting out-of-set class samples (often referred to as out-of-set detection performance) and also the classification performance of the system measured on the in-set classes. Specifically, we assessed the capability to distinguish between in-set and out-of-set class samples by computing the receiver operating characteristic (ROC) curve and measuring the area under the ROC (AU-ROC). In addition, we also measured the capability to retain and correctly classify in-set samples. Following prior works [116–118], we consider the Open-Set Classification Rate (OSCR) curve, and measure the area under this curve (AU-OSCR). Formally, let D_k indicate the set of in-set test samples, and D_u the set of out-of-set test samples. Without loss of generality, we let the event that a sample is from an in-set class be the positive event. The TPR and FPR are then defined as[5],

$$\begin{aligned} \mathrm{TPR}(\nu) &= \frac{\left| \{x | x \in D_k \wedge \xi \geq \nu\} \right|}{|D_k|}, \\ \mathrm{FPR}(\nu) &= \frac{\left| \{x | x \in D_u \wedge \xi \geq \nu\} \right|}{|D_u|}, \end{aligned} \tag{2.7}$$

[5]In the case of synthetic image detection, the positive class is the generated images, while in open-set scenario, the positive is in-set classes.

where ξ is the out-of-set (or rejection) score for sample x, and ν is the rejection threshold.

Then we define the Correct Classification Rate (CCR) as the ratio of samples from known classes detected as in-set and correctly classified, that is,

$$\mathrm{CCR}(\nu) = \frac{\left|\left\{x | x \in D_k \wedge \xi \geq \nu \wedge \hat{y} = y\right\}\right|}{|D_k|}. \tag{2.8}$$

While the ROC curve plots the TPR vs FPR values, the OSCR curve plots CCR vs FPR, by varying the threshold ν.

Chapter 3

Eyes-based Siamese Neural Network for the Detection of GAN-generated Images

"The truth may be stretched thin, but it never breaks, and it always surfaces above lies, as oil floats on water."
Miguel de Cervantes Saavedra, Don Quixote

While DL-based detection methods show excellent performance when tested under conditions similar to those considered for training, they often lack robustness and generalization ability, as they fail to detect fake images that are generated by "unseen" GAN models. A possibility to overcome this problem is to develop tools that rely on the semantic attributes of the analyzed images. In this chapter, we present a semantic-based method for the detection of GAN-synthesized face images that relies on the analysis of inter-eye symmetries and inconsistencies, and resorts to the superior capabilities of similarity learning, notably a Siamese Neural Network (SNN), to extract robust features from the images. The proposed method relies on the observation that GANs have some problems reproducing the symmetries between the eyes and then it is possible to look at the presence of inconsistencies between the patterns in the left and right eyes to detect if the image is real or fake. Specifically, two identical branches of an SNN are fed, respectively, with the images of the right and left eye. The purpose of the two branches is to extract high-level features characterizing the inter-eye similarity, that permit to discriminate between real and synthetic pairs of eyes.

Previous work [95] considered inconsistent corneal specular highlights between two eyes and compared pixel differences for the detection. However, due to the limitations inherent in the corneal region identification process, the method in [95] lacks robustness and generality, and eye localization can be successfully achieved only in easy cases (high contrast and good illumination conditions in the eye region). In contrast, by resorting to a simple bounding-box extraction procedure and exploiting the superior capabilities of SNNs to learn discriminative features, the method we have developed is very general and robust against several image post-processing operators, being capable of detecting fake images also when they are generated by models other than those used for training.

This chapter is organized as follows: in Section 3.1, we describe the method we have developed. Then, Section 3.2 details the methodology that we followed for the experiments. The results of the experiments, including performance analysis, and evaluation of generalization and robustness, are reported in Section 3.3.

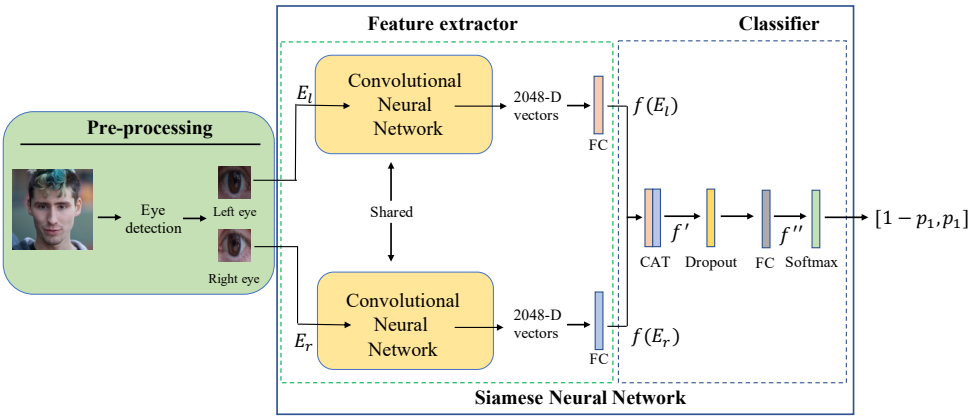


Figure 3.1 – Siamese eyes-based detection of GAN-generated face images.

3.1 Eyes-Based Detection of GAN-Generated Face Images

In this section, we describe the eyes-based GAN-generated face detector we have developed. As we mentioned, the goal of our method is to distinguish GAN images from real images by exploiting dissimilarities and inconsistencies between the eyes in GAN-synthesized faces. To address this binary decision problem, we resorted to an SNN architecture to exploit the capabilities of this kind of structure to find similarities between paired inputs. As a matter of fact, SNNs have been widely used in several related fields, e.g. in the field of face verification [119, 120], person re-identification [121, 122], and even object tracking [123, 124] with very good results. In addition, it has been shown in the literature that, by relying on the similarity learning paradigm, in many cases, SNNs can improve the generalization capability of the models since they tend to learn more robust features [125–129].

A high-level description of an SNN architecture is given below. The SNN consists of: i) two parallel identical CNN branches (with shared weights), consisting of a series of convolutional layers, in charge of performing feature extraction; ii) a combination layer fusing the feature vectors produced by the two branches; and iii) a Fully Connected (FC) part in charge of making the final decision.

Figure 3.1 shows the scheme of the SSN-based eyes-based detector of GAN-generated faces. The detector consists of three modules: a pre-processing module, the feature extraction module, and the final classifier. These two last modules form the SNN. The purpose of the pre-processing module is to localize the eyes within the face and extract the two bounding boxes of the eyes that constitute the paired inputs of the SNN. More specifically, we used the Dlib [130] face detector to locate the face, followed by a landmark predictor that outputs 12 landmark points (6 feature points per eye) whose coordinates indicate the locations of the left and right eyes. The bounding boxes of the left eye E_l and right eye E_r are cropped, exploiting the coordinates of the feature points of each eye.

Then, the bounding boxes are paired and fed to the two branches of the feature extraction module after resizing them to the same fixed size (the input network size). We denote with $x_e = (E_l, E_r)$ the input of the SNN.

The details of the various components of the SNN we used in our scheme are provided in the following. The feature extraction branches are based on a modified version of the XceptionNet architecture [131]. XceptionNet is a particular version of an Inception network [132] that relies on a modified depthwise separable convolution. Such a network has been proven to achieve very good performance for deepfake detection [22]. Inspired by [133], in order to retain as much spatial information as possible (which is particularly relevant in the presence of strong processing and JPEG compression), we removed the sampling operation in the first convolutional layer of the network, setting the stride parameter to 1. In addition, following [134], we replaced the 1000-dim FC layer of the original network with an FC layer of size 512. Then, the FC layer takes the 2048-dim feature vector obtained by the final Global Average Pooling (GAP) layer of the convolutional part as input and outputs a 512-dim feature vector. The parameters of the XceptionNet that extract the features from the left and right eye are shared and are the same for the two branches.

The 512-dim feature vectors extracted by the two branches, namely $f(E_l)$ and $f(E_r)$, are concatenated to produce a 1024-D feature vector $f' = CAT(f(E_l)f(E_r))$, where CAT denote the concatenation operation. To avoid overfitting, the concatenation is followed by a dropout layer, where the nodes are dropped out with a probability of 0.5. The output of the dropout layer is input to another FC layer with two output nodes (one for each class). Finally, a softmax layer is applied to the output of the FC f'' to produce the output probability vector, characterizing the probability that the input images are real and GAN-generated, respectively. We denote with p_1 the probability that the input image has been generated by a GAN: the input is deemed to be a GAN-image if $p_1 > 0.5$, real otherwise. Without loss of generality, in the following, we refer to the GAN class as the positive class and to the real class as the negative class.

3.2 Experimental Methodology

In this section, we discuss the datasets used for the experiments and report the setting used to train the eyes-based GAN detection model. The choice of the state-of-the-art GAN detection method considered for the comparison is also discussed. Finally, we introduce the metrics used to evaluate the performance of the various models.

3.2.1 Training setting

The dataset we used in this chapter is GDD, including ProGAN, StyleGAN2, StyleGAN3 and Print&Scan subdatasets as described in Section 2.4.1. Training is performed considering only the StyleGAN2 model [6], while ProGAN [4] and StyleGAN3 [7] are considered, in addition to StyleGAN2, to generate the images used for the tests.

The input size of the SSN branches is 66×100×3. We performed rescaling as a preprocessing step on the bounding boxes of the eyes, resizing them to the network's input

size. We applied the same rescaling to the images during testing.

We trained the network using the Cross-Entropy (CE) loss function, with the Adam optimizer [135], with default parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.999$). The batch size was set to 64. Training was performed with a constant learning rate of 0.0001 for 10 epochs. During training, we employed strong augmentation techniques to enhance the model's robustness, including JPEG compression (quality factors from 40 to 100), flipping, scaling (scale factors in the range [0.8, 1.3]), contrast and brightness adjustment, and Gaussian blurring. These operations were applied with probabilities of 0.8, 0.1, 0.8, 0.8, and 0.2, respectively. We initialized the parameters of the convolutional neural networks in the two branches using the pre-trained weights on the ImageNet dataset [136]. We implemented the system in a Python3 environment using the Tensorflow library [137], and we utilized an NVIDIA GTX2080Ti GPU for both model training and testing.

3.2.2 Comparison with the state-of-the-art

To demonstrate the effectiveness of Siamese-based GAN-images detector, we compared its performance with those of a method that was - at the time we developed our work - the best performing state-of-the-art method for GAN detection, that is, the method in [16] (see Section 2.3.1 for the details). The best results of this method are achieved by using ResNet50 [138] as the backbone network. It comprises 50 layers and employs residual connections to address the vanishing gradient problem. These connections enable the network to learn residual mappings, facilitating the training of extremely deep models. For the training details, we refer to [16]. When it was published, the method largely outperformed previous methods, e.g., the method in [75] and [23]. We used the model released by the author [1] to run the tests. In the following, we refer to this method as ResNet50-NoDown.

As mentioned at the beginning of the chapter, a GAN detection technique that has close ties with the approach we have developed is the method in [95], which, similarly to our method, exploits eye clues to perform GAN image detection. [95] relies on statistical hand-crafted features. Given that the performance of this detector is much lower than those achieved by ResNet50-NoDown, this method was not considered in our comparisons and its results are not reported in the following. In particular, we found that the method in [95] suffers from the poor performance of the estimation of the corneal region, which does not provide accurate localization of the eye region in many cases. With reference to the datasets used in our experiments, accurate localization could be achieved only in 29% of the CelebA-HQ test images, 48% of FFHQ, and 55% of StyleGAN2.

3.2.3 Metrics

With regard to the metrics used for performance evaluation, we considered the TPR and the FPR, where we remind that a positive event indicates that the input image is a GAN (label 1, $p_1 > 0.5$), while a negative event refers to real images (label 0, $p_1 < 0.5$). We also report the AU-ROC of the classification, measuring the discrimination capabilities of the

[1] `https://github.com/grip-unina/GANimageDetection`

method and providing an indication of the best performance that can be achieved on the test set by adjusting the decision threshold, as well as Pd@5% for more practical measure. Specifically, we used the pristine images in the validation set (5,000 images) to set the threshold of the SNN-based detector, by fixing the FPR at 5%. The detection performance then were evaluated on the test set using this threshold. Both raw (uncompressed) images and JPEG compressed images were considered to determine the threshold and to get a general operating point for the detector. More specifically, the 5,000 images in the pristine validation set were compressed with quality factors {70, 80, 90, 100}, for a total of 25,000 images used to set the threshold.

3.3 Experimental Results

3.3.1 Performance analysis, generalization and robustness

Table 3.1 – TPR/TNR (%), AU-ROC (%) and Pd@5% (%) of the developed method and the ResNet50-NoDown on unprocessed images. Tests are carried out in matched (StyleGAN2) and mismatched (ProGAN and StyleGAN3) conditions. Values in brackets indicate the FPR on the test set using the Pd@5% threshold set on the validation set.

	ResNet50-NoDown			Prop.		
Processing type	TPR/TNR	AU-ROC	Pd@5% (FPR)	TPR/TNR	AU-ROC	Pd@5% (FPR)
StyleGAN2	100/100	100	100 (0)	100/100	100	100 (2.7)
ProGAN	100/100	100	100 (0)	87.2/100	99.7	97.2 (2.7)
StyleGAN3	1.1/100	100	28.0 (0)	84.4/100	99.6	96.7 (2.7)

Table 3.1 reports the results of the our method in matched and mismatched dataset conditions, where the TNR/TPR, the AU-ROC, and the Pd@5% are reported for both the developed and ResNet50-NoDown methods. As shown in Table 2.1, we considered different mismatched generated images (ProGAN and StyleGAN3) for the generalization tests. For the Pd@5%, the FPR measured on the test set is also reported among brackets. Since the TNR refers to the pristine class, the TNR values are the same in all the columns.

Both methods achieve perfect detection results on StyleGAN2 (TPR = 100%). The SSN achieves the best overall generalization results. In particular, without threshold adjustment, it achieves TPR = 84.4% on StyleGAN3 and TPR = 87.2% on ProGAN. The ResNet50-NoDown detector achieves perfect results on ProGAN (with a gain of 12.8% in TPR with respect to our method). However, it does not generalize to StyleGAN3, in which case TPR = 1.1%, even if the AU-ROC is good. The Pd@5% is also poor, being equal to 28.0%. These results indicate that the ResNet50-NoDown method can not work on StyleGAN3 without re-calibrating it on StyleGAN3 images.

In Figure 3.2, we show the distribution of the image features of the various datasets for the developed method. Dimensionality reduction is performed to a 2-dim space by means of t-distributed Stochastic Neighbor Embedding (t-SNE) [139] and Uniform Manifold Approximation and Projection (UMAP) [140] technique (left and right plots, respectively).

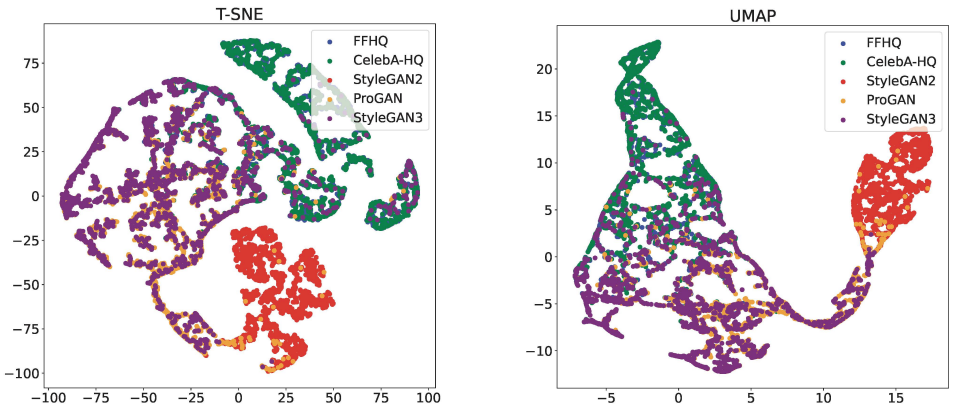


Figure 3.2 – Feature space distribution of the eye-based method for each dataset using T-SNE and UMAP reduction algorithms. 2,000 images per dataset have been considered.

The separability of the pristine and GAN classes is good, with only some overlap between the pristine and the StyleGAN3 images, which were not considered for training. Interestingly, StyleGAN2 and StyleGAN3 images are clustered separately, both with the T-SNE and UMAP reduction techniques, while the distribution of ProGAN overlaps with them. This is consistent with the quality of the generated images, with ProGAN having the worst image quality, StyleGAN3 the best, and StyleGAN2 lies somewhere in the middle. As expected, the pristine images from FFHQ and CelebA-HQ are clustered together.

Figure 3.3 shows some examples of attention maps obtained for the developed method with the Gradient Class Activation Map (GradCAM) algorithm, which is a method for the extraction of maps highlighting the regions of the image that impact most on the decision of the network (see [15]). The activation maps reveal that the Siamese network looks at the eye region of the bounding boxes to make the decision, in particular, we see that the attention focuses on the iris region of the eyes.

The performance in the presence of processing, that is, when the real and fake images are subject to post-processing operations, are reported in Tables 3.2 (matched test conditions) and 3.3 (mismatched test conditions), in terms of TNR, TPR and AU-ROC, and in Table 3.4, in terms of Pd@5%, for several types of processing and processing strength. For the case of Gaussian noise addition, the parameter we report in the table is the variance of the noise. For the resizing, we report the scale factor used for the rescaling operation. For the case of Gaussian blurring, the parameters refer to the size of the Gaussian kernel, while for median filtering, it refers to the window size. Finally, for contrast enhancement, the image contrast is increased by a factor of 1.5. All these processing operations correspond to global manipulations of the image since they affect all the pixels of the image. Local manipulations are considered in Section 3.3.2.

Looking at the AU-ROC results and the Pd@5%, we see that both our method and ResNet50-NoDown are robust against processing, in particular, JPEG compression, resiz-

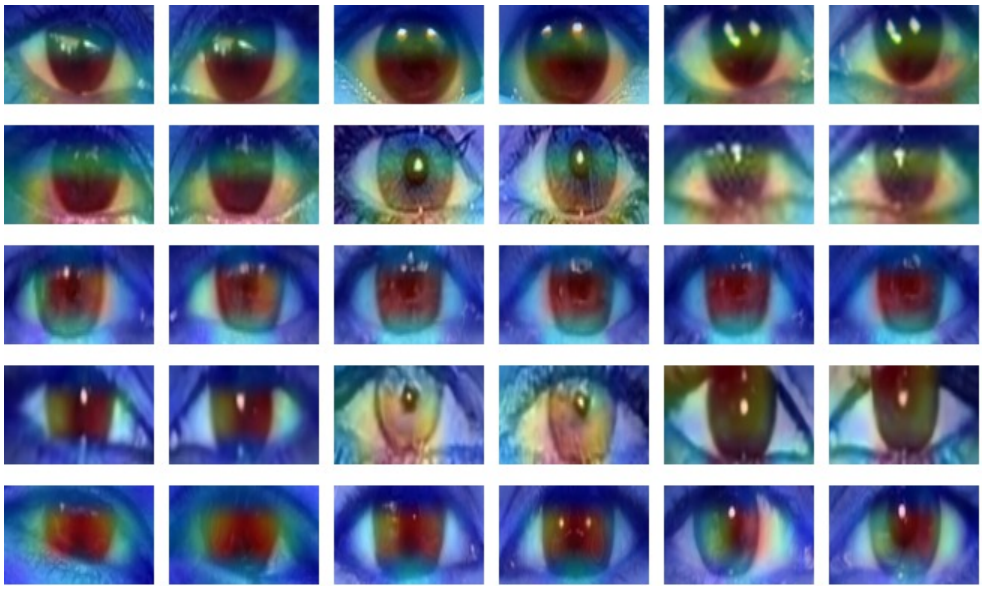


Figure 3.3 – GradCAM visualization for the developed detector. From top to bottom row: FFHQ, CelebA-HQ, StyleGAN2, ProGAN and StyleGAN3. 3 sample pairs are visualized in each row (left and right eyes).

ing, filtering, blurring and contrast adjustment. In particular, these experiments confirm the same trend, with the ResNet50-NoDown that outperforms the developed method on ProGAN, with an improvement of a few percent in many cases (and less than 10% in all the cases), but does not generalize to StyleGAN3, where the developed methods provides much better results. Both methods suffer from Gaussian noise addition. Given that noise addition has been considered to train the ResNet50-NoDown model, but not to train the SNN, it is not surprising that the performance with respect to this type of processing for the developed method is lower.

3.3.2 Other results

A noticeable strength of the eyes-based GAN detector is that it relies on semantic information for the discrimination. This is not the case with the ResNet50-NoDown method, which bases its decision on features automatically extracted by the network from the entire image. Figure 3.4 (top row) shows some examples of attention maps obtained with the GradCAM algorithm [15] for ResNet50-NoDown. As is often the case with self-learned CNN architectures, the regions highlighted by the maps - that mostly affect the decision - are many and spread over the whole image, also lying in the background, confirming the lack of explainability of fully-CNN-based solutions. Due to this behavior, we expect an advantage of the developed method in the presence of local manipulations, e.g., in the GAN splicing scenario, when the GAN object (the face, in this case) is pasted on a real background.

Table 3.2 – TPR/TNR (%) and AU-ROC (%) of the developed method and ResNet50-NoDown under various image processing operations. Tests are carried out in matched (StyleGAN2) conditions.

Processing type	StyleGAN2			
	ResNet50-NoDown		Prop.	
	TPR/TNR	AU-ROC	TPR/TNR	AU-ROC
JPEG100	100/100	100	100/100	100
JPEG90	100/100	100	100/100	100
JPEG80	100/100	100	100/100	100
JPEG70	100/100	100	100/100	100
Gaussian Noise	70.2/74.0	70.8	41.1/90.0	84.6
Resize- 2	100/100	99.8	99.3/100	100
Resize- 1.3	100/100	99.8	100/100	100
Resize - 0.5	92.1/100	97.0	100/99.2	100
Gaussian blur- 3×3	100/100	99.9	100/99.3	100
Gaussian blur -5×5	100/100	99.9	100/99.3	100
Median filter -3× 3	88.1/100	65.0	99.0/99.1	100
Contrast enhance - 1.5	99.9/100	100	96.1/98.4	99.7

Table 3.3 – TPR/TNR (%) and AU-ROC (%) of the developed method and ResNet50-NoDown under various image processing operations. Tests are carried out in mismatched (ProGAN and StyleGAN3) conditions.

Processing type	ProGAN				StyleGAN3			
	ResNet50-NoDown		Prop.		ResNet50-NoDown		Prop.	
	TPR/TNR	AU-ROC	TPR/TNR	AU-ROC	TPR/TNR	AU-ROC	TPR/TNR	AU-ROC
JPEG100	100/100	100	82.0/100	99.6	1.9100	100	79.3/100	99.5
JPEG90	99.4/100	100	69.1/100	98.7	2.1/100	100	69.0/100	98.8
JPEG80	94.3/100	100	55.8/100	97.0	10.0/100	99.6	61.1/100	97.7
JPEG70	92.1/100	100	47.4/100	94.6	10.1/100	97.6	55.2/100	96.3
Gaussian Noise	71.3/74.0	80.4	14.0/90.0	69.9	55.1/74.0	70.8	8/90.0	46.6
Resize- 2	100/100	100	85.4/100	99.6	1/100	99.8	84.2/100	99.9
Resize- 1.3	100/100	100	85.8/100	99.7	1/100	99.8	83.3/100	99.6
Resize - 0.5	100/100	100	85.4/99.2	99.4	3.1/100	97.0	71.2/99.2	98.3
Gaussian blur- 3×3	100/100	100	72.2/99.3	98.6	5.4/100	99.9	73.3/99.3	98.6
Gaussian blur -5×5	100/100	100	63.1/99.3	96.9	8.9/100	99.9	65.0/99.3	97.2
Median filter -3× 3	99.1/100	100	74.4/99.1	97.9	9.3/100	84.0	35.2/99.1	91.4
Contrast enhance-1.5	100/80.4	100	60.4/98.4	96.5	0.01/100	94.3	53.7/98.4	94.7

Table 3.4 – Pd@5% (FPR) for the developed method and the ResNet50-NoDown under various image processing operations. The FPR (%) measured on the test set is reported among brackets. Tests are carried out in matched (StyleGAN2) and mismatched (ProGAN and StyleGAN3) conditions.

	StyleGAN2		ProGAN		StyleGAN3	
Processing type	ResNet50-NoDown	Prop.	ResNet50-NoDown	Prop.	ResNet50-NoDown	Prop.
JPEG100	100 (0)	100 (2.7)	100 (0)	96.3 (2.7)	14.0 (0)	96.1 (2.7)
JPEG90	100 (0)	100 (3.6)	100 (0)	92.0 (3.6)	96.2 (0)	93.3 (3.6)
JPEG80	100 (4.1)	100 (5.9)	100 (4.1)	85.4 (5.9)	98.1 (4.1)	89.4 (5.9)
JPEG70	100 (18.7)	100 (8.0)	100 (18.7)	79.1 (8.0)	100 (18.7)	87.4 (8.0)
Gaussian Noise 0.01	94.2 (54.7)	61.1 (12.8)	93.4 (54/7)	25.2 (12.8)	88.2 (54.7)	2 (12.8)
Resize - 2	100 (0)	100 (3.0)	100 (0)	97.3 (3.0)	26.0 (0)	97.2 (3.0)
Resize - 1.3	100 (0)	100 (2.6)	100 (0)	97.2 (2.6)	23.0 (0)	97.3 (2.6)
Resize - 0.5	99.2 (0)	100 (6.1)	99.2 (0)	98.3 (6.1)	25.0 (0)	94.4 (6.1)
Gaussian blur- 3×3	100 (0)	100 (6.0)	100 (0)	93.3 (6.0)	55.4 (0)	94.3 (6.0)
Gaussian blur - 5×5	100 (0)	100 (9.0)	100 (0)	90.1 (9.0)	67.3 (0)	92.2 (9.0)
Median filter - 3×3	100 (21.1)	100 (9.3)	100 (21.1)	93.1 (9.3)	79.3 (21.1)	72.3 (9.3)
Contrast enhance - 1.5	100 (0)	100 (13.8)	100 (0)	93.2 (13.8)	10.0 (0)	89.2 (13.8)

Performance in the Presence of GAN Splicing

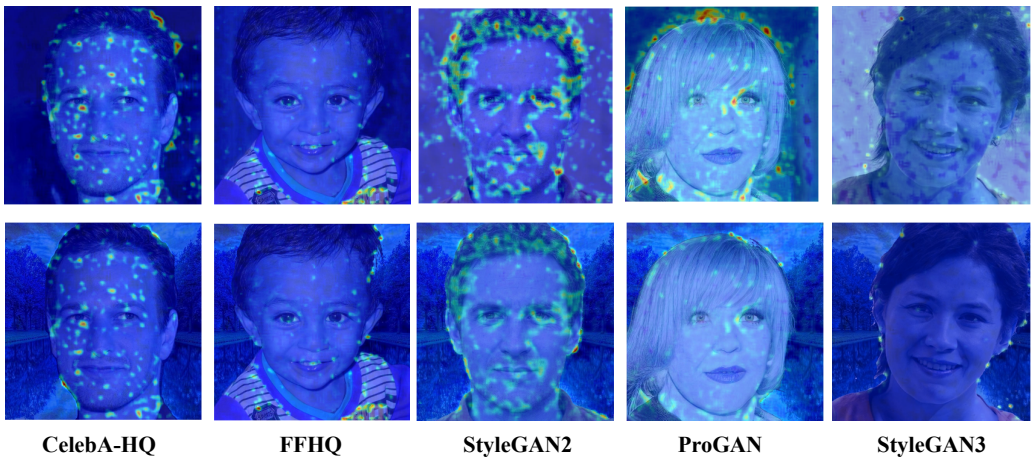


Figure 3.4 – GradCAM [15] visualization for ResNet50-NoDown [16] before (top) and after (bottom) image splicing.

To run some tests in the image spicing scenario, we generated a number of forged images for the case of real and synthetic faces by cutting the foreground person from the image and pasting it on a real background. A total of 30 GAN spliced images for each GAN type (StyleGAN2, ProGAN and StyleGAN3) and 30 real spliced images are

Table 3.5 – Results in terms of Pd@5% (FPR %) achieved on spliced images. The FPR (%) measured on the test set is reported among brackets.

Datasets	StyleGAN2	ProGAN	StyleGAN3
	TNR/TPR	TNR/TPR	TNR/TPR
ResNet50-NoDown	100 (0)	90 (0)	3 (0)
Prop.	100 (0)	97 (0)	100 (0)

Figure 3.5 – Example of splicing operation in FFHQ dataset (top), StyleGAN2 (middle) and StyleGAN3 (bottom), respectively. From left to right: original image, real background, spliced image.

obtained in this way. An example of a local GAN spliced image is illustrated in Figure 3.5. Some examples of attention maps for ResNet50-NoDown obtained by running the GradCAM algorithm on the spliced images are provided in Figure 3.4 (bottom row).

The results of the tests are provided in Table 3.5, where we report the Pd@5% obtained

using the same threshold as before, set on the validation set. Obviously, the performance of the developed method is not affected by the splicing operations, given that the eye region remains the same. Regarding the performance of ResNet50-NoDown, we observe that, although the evidence that can be found in the foreground is enough for the method to perform correct discrimination in the StyleGAN2 case (that is when the pasted foreground corresponds to a StyleGAN2 face), the presence of the real background affects the generalization performance. In the case of StyleGAN3, for which the performance is already poor in the non-splicing case, the method gets Pd@5% = 3% on GAN-spliced images. In the ProGAN case, the Pd@5% decreases by 10%.

Performance on Print&Scan Images

Figure 3.6 – Examples of StyleGAN2 images from the Print&Scan image dataset [17] (top), and corresponding digital images (bottom).

We also run some tests on the Print&Scan dataset in order to investigate the robustness of the developed method to the rebroadcast operation and assess whether the features the detectors look at survive recapturing. The developed method is based on semantic attributes (eye clues), and we expect better robustness against recapturing than for state-of-the-art methods based on features automatically learned from the full images. Some examples of recaptured GAN and real images from the Print&Scan dataset, alongside the original digital versions, are reported in Figure 3.6. We can see a noticeable quality degradation in the recaptured images. In particular, noisy textures are visible, and the

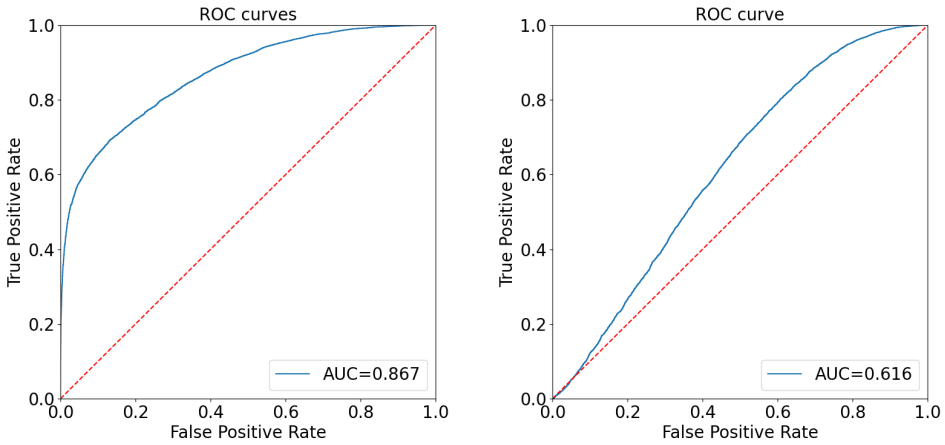


Figure 3.7 – Performance (AU-ROC) on the Print&Scan image dataset: the developed SNN-modifiedXceptionNet (left) and ResNet50-NoDown (right).

colors are changed.

The ROC curve of the eyes-based method and ResNet50-NoDown on the Print&Scan image dataset are reported in Figure 3.7. We see that the developed method maintains some discrimination capability. In particular, using the same threshold fixed on the validation set for digital images, we get Pd@5% = 76% with an FPR = 22%, which is a good result given the significant difference in the test image domain in this case. Adjusting the threshold on recaptured data helps to improve the performance of the detector by 4% in the Pd@5% for the same FPR. Given that the Print&Scan dataset [17] is very challenging and recapturing attacks are powerful attacks, the results achieved by the developed method are good ones. From Figure 3.7 (right), we see that the recapturing operation destroys the (weak) features that the ResNet50-NoDown method looks at, and no discrimination between real and fake images can be obtained by using this network, the AU-ROC being around 61.6%.

Table 3.6 – Performance comparison among different backbone architectures.

Architectures	StyleGAN2		ProGAN		StyleGAN3	
	TNR/TPR (%)	AU-ROC (%)	TNR/TPR (%)	AU-ROC (%)	TNR/TPR (%)	AU-ROC (%)
Xception [75]	100/100	100	100/21.0	93.0	100/16.0	85.8
SNN-Xception	100/100	100	100/43.0	96.0	100/73.0	99.0
SNN-modifiedXception	100/100	100	100/62.0	98.7	100/80.0	99.3
SNN-modifiedResNet50	90.2/100	100	90.2/30.1	69.0	90.2/90.0	96.5

3.3.3 Ablation study

In Table 3.6, we report the performance of the eyes-based detector when different architectures are considered to implement the two branches of the SNN, including a comparison with the standard XceptionNet trained on the entire image [75]. This table shows that the SNN with the modified XceptionNet corresponds to the best choice, getting better results than the standard XceptionNet. Moreover, better results are achieved using XceptionNet as the backbone, with respect to ResNet50, that is, the network considered in [16]. Interestingly, all the models get a TPR of 100% on StyleGAN2 images (the TNR is also 100%), and the difference among the trained models can be appreciated only by looking at the results obtained on ProGAN and StyleGAN3 images, that is, at the generalization performance. In particular, the SNN with the modified version of XceptionNet (best choice) improves the generalization on ProGAN and StyleGAN3 images by 39% and 64% in terms of TPR, with respect to the standard XceptionNet model trained on the entire image. These results justify the choice of the modified XceptionNet as the backbone network for the two convolutional branches of the SNN.

3.4 Summary

In this chapter, we have described a semantic-based method for GAN-generated face image detection that reveals the synthetic nature of a face image based on the analysis of eye clues, exploiting the similarity learning paradigm and SNNs. The method relies on the assumption that GANs cannot reproduce properly. Inter-eye symmetries. The SNN is implemented by considering a modified XceptionNet as the backbone. Our experiments showed good performance of the method with respect to both generalizations to different GAN architectures (from StyleGAN2 to ProGAN and StyleGAN3) and robustness against image processing in both digital domain (noise, JPEG compression, blur, etc. and cut&paste manipulation) and physical domain (Print&Scan). The benefits of the method for cut&paste and Print&Scan processing shows exemplify the advantages of relying on semantic artefacts, for real-world applications. For instance, the robustness against cut&paste operations suggests that the system has the potential to address emerging photo editing tools like Adobe Firefly, which manipulate and fill parts of the images with generated faces. Of course, the adaptability can be improved by fusing multi-level semantic features.

Chapter 4

A Hybrid Architecture for the Classification and Localization of GAN-generated Images with Improved Robustness and Generalization Capabilities

When the flood submerges the whole country, no raindrop may feel responsible.
("Finally things had lost their weightiness").
Erik Pevernagie

In addition to AI-synthesized image detection, image manipulations that change the contents of images and make them convey incorrect or misleading information have caught researchers' attention for decades [141]. These partially content-changed images can be used for unethical and illegal purposes. The raw image can be altered either in parts or as a whole, necessitating the detection of the type of tampering performed and the localization of the tampered region. In this chapter, we present a hybrid architecture for the classification and localization of AI-manipulated images, wherein localization is used to aid the classification task, by forcing the network to focus on the parts of the image that are the most relevant for the classification task[1](i.e., the manipulated region).

A standard convolutional architecture is used to extract the features that are relevant to the classification task. The features extracted by the convolutional layers are also used to drive a Fully Convolutional Network (FCN) module in charge of localizing the manipulated image region. One of the main findings of this work, in fact, is that adding a localization task on top of the feature extraction layers has a positive effect on classification accuracy, even if localization is not very accurate. That is, asking the network to localize the manipulation acts as a kind of attention mechanism that somehow forces the feature extraction layers to focus on the most relevant parts of the image. It turns out that this has a positive effect on the generalization capabilities and on the robustness against image processing operations.

Specifically, we exploited the designed architecture to address the detection of GAN-generated images of climate change and, in particular, the detection of GAN-generated flood images. It is worth pointing out that the domain of weather images is relevant for manipulation detection. In fact, the risk exists that climate-sensitive synthetic images are used maliciously or exploited within organized disinformation campaigns. The devel-

[1]Being the detection task a particular classification task, where only two classes are involved, we refer more in general to classification

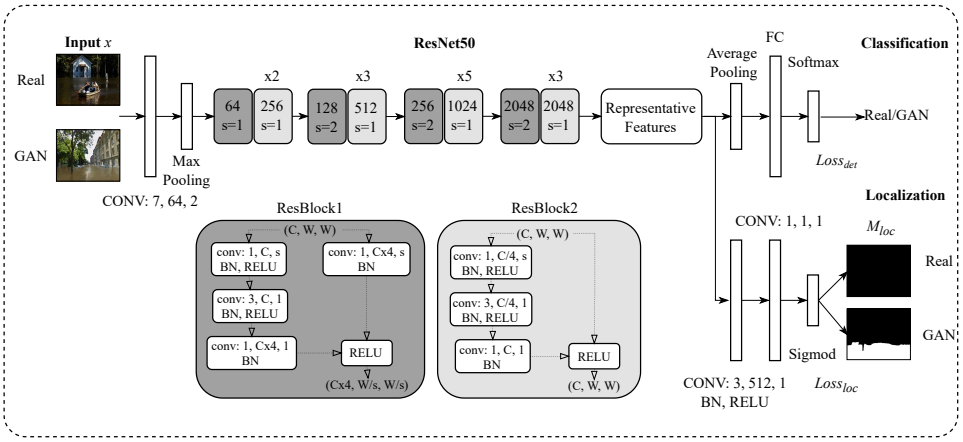


Figure 4.1 – Overview of the designed hybrid architecture for simultaneous classification and localization. The figure exemplifies a case of binary classification where the GAN image corresponds to a fake flood image, and the mask highlights the water region.

opment of techniques capable of distinguishing natural and generated weather images is then of paramount importance.

The rest of the chapter is organized as follows. A general description of the hybrid classification and localization architecture is given in Section 4.1. In Section 4.2, we first describe the application of the hybrid architecture to the detection of GAN-generated flood images. In Section 4.3, we conclude the chapter with some final remarks.

4.1 The Hybrid Architecture

In this section, we present a hybrid architecture for the classification and localization of manipulated images. Given a forged image x, with $x \in \mathbb{R}^{H \times W \times 3}$, the goal of our system is to associate to x a label y and a mask I_M, where the mask I_M indicates the pixels where the image has been manipulated (pixels for which $I_M = 1$ indicate the manipulated areas)[2]. As we said, the main intuition behind our work is that asking the network to localize the manipulated areas also helps the classification process since, during training, we can instruct the network to focus on the image areas that have most likely been manipulated. We also expect that our approach will improve the network's generalization capability.

The overall architecture of the method is shown in Figure 4.1. For simplicity, the figure exemplifies a case of binary classification (the GAN image corresponds to a fake flood image, and the mask highlights the water region). The details of each part of the architecture consists of are provided in the following.

[2]In the thesis, we generally use y to denote the GT label of input samples. Depending on the system and classification framework, y may denote either the class index or correspond to the one-hot encoding of the class.

Given an image x, we pass it through the ResNet50 network to extract a set of representative features $\mathcal{F}$ to be used for both classification and localization. More specifically, the network starts with a convolutional layer with a kernel size of 7×7 and 64 different kernels all with a stride of 2, followed by a max-pooling layer. Then, we have four modules combining the convolutional block (ResBlock1) and the identity blocks (ResBlock2). Each module consists of one convolutional block and n identity blocks represented by $\times n$ in Figure 4.1 (i.e., $\times 2$ indicates two identity blocks). For each block, there are two basic parameters: the number of kernels and the stride s used in the first convolutional layer. Down-sampling is applied when $s = 2$. Afterwards, the semantic features extracted by the ResNet50 backbone are exploited for the classification and localization tasks. Specifically, for the localization task, we consider an FCN [142]. FCNs have been used mainly for semantic segmentation. As indicated by the name, they solely employ locally connected layers, such as convolution, pooling and upsampling, avoiding the use of dense layers. Back to our work, the localization masks I_M is obtained by using two convolutional layers and a sigmoid function layer. At the same time, the representative features $\mathcal{F}$ are sent to an AveragePool layer and an FC layer in charge of distinguishing GAN and real images. As to training, we rely on two loss functions, one associated with the classification task and one with the localization task. The classification loss is defined as:

$$\mathcal{L}_{cls} = \frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{j=1}^{C} y_{i,j} \log(p_{i,j}), \tag{4.1}$$

where the sum is extended to all the N_t images in the training set, C is the number of classes ($C = 2$ in this case), y_i is the ground truth, one-hot-encoding, vector associated to x_i, where $y_{i,j} = 1$ if x_i belongs to class j, 0 otherwise, and $p_i = (p_{i,1}, ..., p_{i,C})$ is the soft output vector of the network in correspondence of x_i. Similarly, the localization loss is defined as

$$\mathcal{L}_{loc} = \frac{1}{N_t \times H \times W} \sum_{i=1}^{N_t} \sum_{j=1}^{H \times W} I_{G_{i,j}} \log(I_{M_{i,j}}) + (1 - I_{G_{i,j}}) \log(1 - I_{M_{i,j}}), \tag{4.2}$$

where $I_{M_i} \in \mathbb{R}^{H \times W}$ is the estimated manipulation binary mask for the test image x_i, $I_{M_{i,j}}$ the value assumed by the map in correspondence of the j-th pixel, while I_{G_i} the corresponding ground-truth mask. Eventually, the total loss used to train the model is a weighted sum of the above two losses, namely:

$$\mathcal{L} = \lambda_{cls} \cdot \mathcal{L}_{cls} + \lambda_{loc} \cdot \mathcal{L}_{loc}, \tag{4.3}$$

where λ_{cls} and λ_{loc} are set in such a way to balance the importance of the two loss terms.

4.2 Detection of GAN-generated Flood Images

Climate change is one of the most serious threats to humankind and one of the hardest challenges our society will have to face within the coming years. Climate change exacerbates flooding by increasing the frequency and intensity of heavy rainfall, accelerating sea

level rise, and intensifying storm surges. These factors lead to more frequent and severe floods, causing extensive damage to infrastructure, ecosystems, and human communities. The visualization of the effects of climate change can play a major role in developing new analysis techniques and raising public awareness. They can engage and inform the public about the immediate and long-term risks posed by climate change-induced flooding, encouraging community preparedness and support for mitigation efforts. For this reason, several works have been proposed to generate weather-sensitive images using GAN networks. For instance, [143] proposes a semi-supervised method to generate outdoor images with arbitrary weather conditions at arbitrary times and locations. In [144], a method is proposed to generate outdoor scene images for transient attribute estimation. To augment the diversity of weather images used to train self-driving vehicles, [145] proposed a method to synthesize images with adverse snowy conditions. [146] proposed weather GAN (WeaGAN), a GAN architecture capable of translating the weather of an image across multiple domains, while [18] proposed a *ClimateGAN* architecture to generate extreme flood street view images. Though, initially, the purpose of these works was to promote research and raise awareness of the importance of climate change, the risk exists that climate-sensitive synthetic images are used maliciously or exploited within organized disinformation campaigns. Given that all the flood images are manipulated based on real images, it would be very helpful to not only determine whether an image is real or fake but also pinpoint the specific areas of manipulation. This approach allows for a deeper understanding of the modifications made and enhances the reliability of detection methods. Highlighting the manipulated regions can provide visual evidence of tampering, aiding in forensic analysis and raising public awareness about the authenticity of images, especially in the context of critical issues like climate change and flooding.

In this chapter, we focus on the detection of synthetic flood images generated by the ClimateGAN network in [18]. To the best of our knowledge, this is the first work focusing on the detection of GAN-manipulated flood images.

4.2.1 Dataset construction

The first step towards the development of a model for the detection of fake flood images is the construction of a large dataset of both synthetic and natural flood images. In this section, we describe the details of the datasets we have built to train and test our system.

Natural Flood Images. With regard to natural flood images, we relied on two public datasets. The first one is the **Roadway Flooding Image (RWFI)** dataset [147], consisting of 441 images with different scenes from urban, suburban and natural settings. The images are all the same size equal to 385×512. Mask images of flooded areas are also available. We used this dataset for training. The second dataset is the **WSOC Flood Image** dataset [148]. This dataset contains 490 real flood images gathered from Twitter and online news using "floods" as the keyword for the search query. 439 of such images are also accompanied by a binary mask image identifying the flooded areas. We used these 439 flood images to test the generalization capability of the proposed detector.

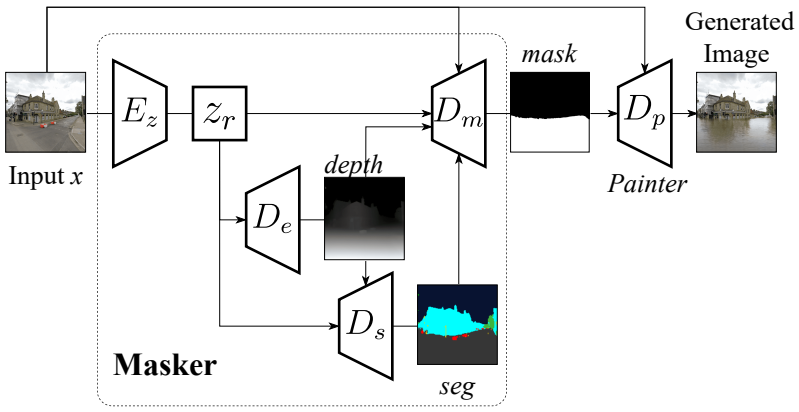


Figure 4.2 – Overall view of *ClimateGAN* architecture [18].

GAN-generated Flood Images. We collected the synthetic flood images by relying on the *ClimateGAN* network described in [18]. *ClimateGAN* is an architecture for image-to-image translation whose goal is to produce images with extreme flooding conditions from street images. The goal is to raise the public's attention on climate change. As shown in Figure 4.2, the *ClimateGAN* consists of two main parts: a *Masker*, whose goal is to predict which part of the image would be covered by water in case of a flood, and a *Painter*, in charge of rendering the flooded areas with water texture fitting the surrounding context. More specifically, a street input image x is transformed into a high-level representation z by using an encoder E_z. A mask image is then estimated by the Mask decoder D_m based on the representation z_r. At the same time, an additional depth information map and a semantic map are built and incorporated into the Mask prediction stage by training a depth map decoder D_e and a segmentation decoder D_s jointly with the Mask-building network. Afterwards, the input image is masked by the estimated mask image, and the conditional Painter network D_p is trained to add water in the area indicated by the mask.

By using the pre-trained *ClimateGAN* model released by the authors of [18][3], we built three synthetic flood image datasets by using diverse street images. Specifically, we first created 3,900 GAN flood images (hereafter referred to as **StreetG** dataset), starting from the Cityscapes [149] and Kitti [150] video datasets described in [151]. For each video, we took a frame every 60 frames and used the central cropped area of size 512 × 512 as input of the *ClimateGAN*. We also built two additional small *ClimateGAN* datasets, namely the **WebG132** and **WebG504** datasets, to validate the generalization capability of the proposed detector. The street view images used to build these datasets were collected from the Internet by using Mapillary[4]. More specifically, WebG132 contains 132 *ClimateGAN*

[3] `https://github.com/cc-ai/climategan`
[4] `https://www.mapillary.com`

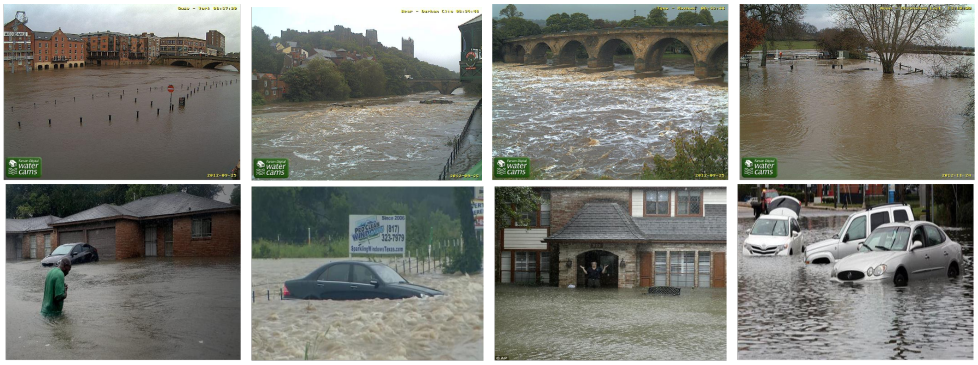


Figure 4.3 – Real flood images. Top: WSOC; Bottom: RWFI.

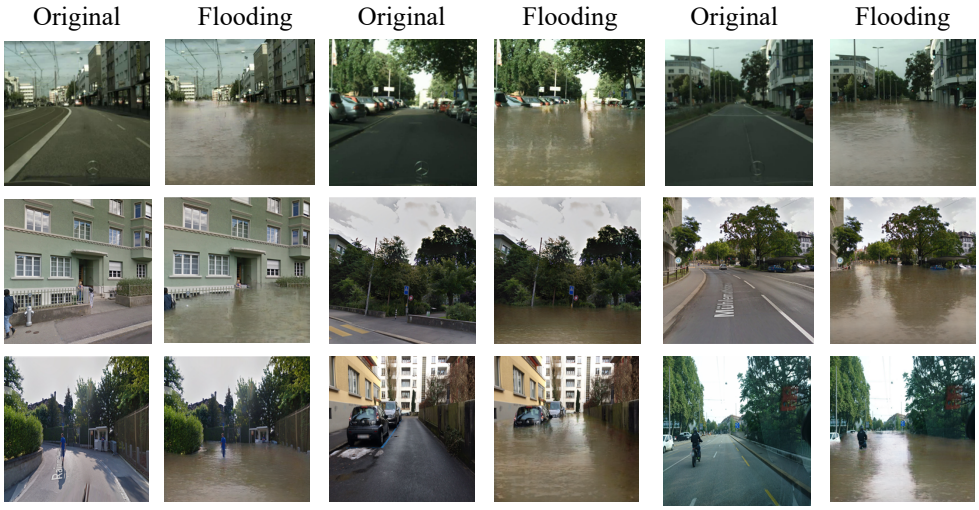


Figure 4.4 – Synthetic, GAN-generated, flood images. Top: StreetG; Middle: WebG132; Bottom: WebG504.

flood images built as **StreetG**. WebG504 is an extended version of WebG132 built by resizing the input images rather than cropping. WebG504 consists of 504 GAN flood images.

Some examples of real and fake images contained in the datasets described above are given in Figure 4.3 and 4.4. Table 4.1 provides a summary of the datasets used in this chapter, along with details on how they were utilized for training and testing the developed system. The ClimateGAN image datasets split in training and testing subsets are available at the link[5].

[5]Manipulated flood image dataset

Table 4.1 – Overview of the datasets used to train and test the synthetic flood images detector.

Datasets	Number of images	Used in	Type	Source	Size
RWFI	441	Train&Test	Real	[147]	385×512
WSOC	439	Test	Real	[148]	From 158×118 to 900×1200
StreetG	3,900	Train&Test	GAN	Kitti and CityScapes video datasets	512×512
WebG132	132	Test	GAN	Collected from Internet	512×512
WebG504	504	Test	GAN	Collected from Internet	512×512

4.2.2 Experimental setting

Baseline Methods. To prove the effectiveness of our method, we selected two popular deep neural networks, namely Xception and ResNet50, which have been proven to be effective for several forgery detection applications (see, for instance, [151] and [16]). We considered various training strategies. We trained an Xception and ResNet50 model as standard binary detectors to decide if the input image has been generated by *ClimateGAN* or not. In the ResNet50 case, the architecture coincides with the classification branch of the hybrid architecture. We also trained four models by using an additional mask image as input to guide the attention of the network toward water regions. In detail, two models referred to with the label MUL, take the image x as input and multiply it by a binary mask indicating the water regions. The other two models, referred to by the label CAT, take a 4-channel image as input, with the fourth band containing a binary mask highlighting the water areas. The performance obtained by the above networks was compared with the detection performance of the hybrid architecture, hereafter referred to as HybCls&Loc. It is worth stressing that, in contrast to the networks taking as an additional input the mask with the water regions, our method does not need such information. In fact, we force the network to look at the water regions indirectly, by asking the network to localize the manipulated areas.

Implementation Details. All the models were trained from scratch with the same configuration for 30 epochs. For the optimization, we used the Adam optimizer with a learning rate of 0.0001 and a mini-batch size of 16. The input images were resized to 224×224×3 and normalized with mean [0.485, 0.456, 0.406] and variance [0.229, 0.224, 0.225], namely, the average mean and variance computed over the ImageNet dataset [136]. The training images were augmented by random color transformations (saturation, brightness, contrast). We trained the networks by using the RWFI dataset for real images and an equal number of GAN-generated flood images from the StreetG dataset. The training datasets were split into training and validation subsets with percentages of 80% and 20% respectively. All the other datasets were used for testing. For the hybrid network, we experimentally set $\lambda_{cls} = 0.4$ (and $\lambda_{loc} = 0.6$), which is the setup that gave the best results. To evaluate the detection effectiveness, we considered the TPR and TNR of the decision. We also calculated the AU-ROC performance by pairing each fake image dataset

Table 4.2 – Results on different datasets. WSOC, WebG132 and WebG504 are not used for training. The higher the value, the better the result.

	Real	GANs					
	WSOC	StreetG		WebG132		WebG504	
Detection capability							
Methods	TNR%	TPR%	AU-ROC%	TPR%	AU-ROC%	TPR%	AU-ROC%
Xception	96.8	98.0	99.5	37.9	84.3	57.7	91.8
ResNet50	95.0	98.6	99.4	47.0	84.0	64.9	91.5
Xception+M (CAT)	94.5	86.4	96.6	41.7	75.5	65.9	89.2
Xception+M (MUL)	96.8	97.5	99.5	65.1	95.9	73.4	96.9
ResNet50+M (CAT)	92.0	99.6	98.7	47.0	83.0	60.5	88.6
ResNet50+M (MUL)	95.2	98.3	99.3	70.5	96.1	79.2	96.7
HybCls&Loc (Prop.)	**98.0**	**100**	**100**	**93.4**	**99.0**	**95.4**	**98.9**
Localization capability							
Methods	bPA	bPA	IoU	bPA	IoU	bPA	IoU
HybCls&Loc (Prop.)	98.6	96.1	92.0	84.4	62.5	87.9	71.2

with the real image dataset. To evaluate the localization performance, we adapted the metrics used for segmentation measurement, that are, the balanced Pixel Accuracy (bPA) and Intersection over Union (IoU) [152], which for a binary classifier are defined as:

$$\text{bPA} = \frac{1}{2N_{test}} \sum_{t=1}^{N_{test}} \sum_{i=0}^{1} \frac{\rho_{ii}}{\sum_{j=0}^{1} \rho_{ij}}, \tag{4.4}$$

and

$$IoU = \frac{1}{N_{test}} \sum_{t=1}^{N_{test}} \frac{|\{I_{M_t} \equiv 1\} \cap \{I_{G_t} \equiv 1\}|}{|\{I_{M_t} \equiv 1\} \cup \{I_{G_t} \equiv 1\}|}, \tag{4.5}$$

where ρ_{ij} indicates the number of pixels in class i classified as class j, $\{I_{M_t} \equiv 1\}$ is the set of pixels of value 1 in the I_{M_t} mask images produced by the network, and N_{test} indicates the number of test images.

4.2.3 Results

Table 4.2 shows the results on the four test datasets. The baseline methods achieve good classification performance on the StreetG dataset, with a TPR around 98% and an AU-ROC equal to 99%. However, the performance drops on the datasets that are not used for training. The performance of ResNet50 and Xception is the worst, likely due to the small number of images available for training. The way the water mask images are used during training also has a significant impact on the performance. By looking at the results in rows 3 to 6, multiplying the mask and the input image results in better performance on WebG132 and WebG504 datasets, than simply concatenating the mask to the image bands. The hybrid detection/localization method (HybCls&Loc) achieves the best performance on all the datasets, with accuracies always well above 90% and often

Table 4.3 – Results in the presence Gaussian noise addition.

	Real	GANs					
	WSOC	StreetG		WebG132		WebG504	
Methods	TNR%	TPR%	AU-ROC%	TPR%	AU-ROC%	TPR%	AU-ROC%
Xception	98.0	77.7	98.0	18.9	78.5	34.7	88.2
ResNet50	98.4	94.4	99.3	35.6	85.1	47.2	92.0
Xception + M (CAT)	90.4	60.0	83.1	25.6	56.3	46.0	76.9
Xception + M (MUL)	**100**	1.0	93.9	0	91.9	0	92.5
ResNet50 + M (CAT)	94.5	89.5	97.6	26.5	78.4	41.1	86.0
ResNet50 + M (MUL)	97.5	34.1	89.5	15.1	90.5	15.1	90.8
HybCls&Loc (Prop.)	97.0	**100**	**100**	**84.1**	**97.6**	**86.5**	**97.8**

close to 100%. In particular, the hybrid model shows a good generalization capability, achieving very good performance on WebG132 and WebG504 datasets, with 93.4% and 95.4% TPR values, and 99.0% and 98.9% AU-ROCs. This is a remarkable result, given that such good performance is obtained without the additional information provided by the water mask. The superior performance provided by the hybrid model is even more interesting if we consider the localization accuracy. By looking at the bPA and IoU values reported in the last row of Table 4.2, we can see that the developed network provides good results on WSOC and StreetG, even if we can notice a performance drop on WebG132 and WebG504 datasets. The remarkable conclusion we can draw is that asking the network to localize the tampered regions helps to distinguish fake images from real ones, even when localization does not work very well, since it forces the detection branch to focus on the manipulated area. Such intuition is confirmed by the CAM maps [15], computed with regard to the detection task, shown in the last row of Figure 4.5.

Table 4.3 and Figure 4.6 show the robustness of all the methods against various image processing operators, including JPEG compression with quality factor 50, image downsampling with resizing 0.5, Median filtering (3×3), Gaussian blur (3×3), and Gaussian noise addition with zero mean and variance equal to 0.003[6]. Overall, all the models are robust to the image processing operators used in our tests, with the exception of Gaussian noise addition (see Table 4.3). In the Gaussian noise case, in fact, the performance of the Xception+mask (MUL) model drops from 97.5%, 65.1% and 73.4% to 1%, 0%, and 0%. ResNet50+mask (MUL) also suffers the addition of Gaussian noise. The bad behavior of the models adopting a multiplicative approach to include the information provided by the water mask indicates that, in the absence of information about the non-tampered regions, the impact of noise is amplified, since the texture information only is not enough to detect the presence of tampering. As a piece of further evidence backing this intuition, we observe that Xception and ResNet50 show stronger robustness. The positive effect of considering the entire image instead of just the water area is also evident in the hybrid model. As it can be seen from Table 4.3, in fact, the hybrid model maintains good performance even in the presence of noise addition. With regard to the other processing

[6]In all our experiments images take values in the [0,1] range.

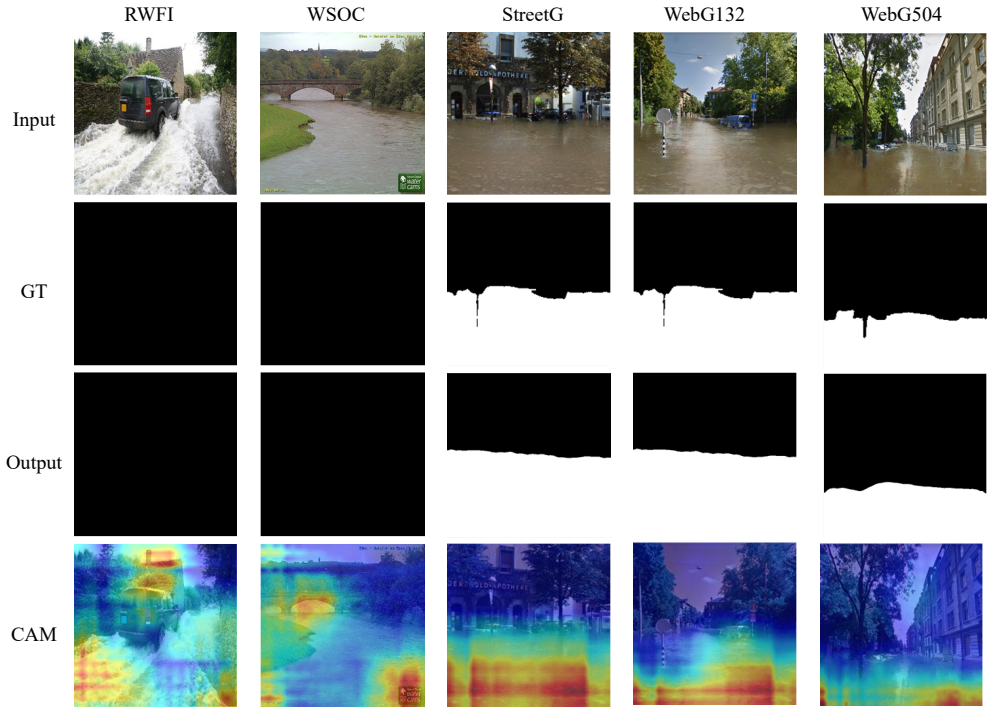


Figure 4.5 – Visualization of output masks and CAM maps for HybCls&Loc.

operators (see Figure 4.6), the hybrid model always achieves the best results with a minor performance drop on the WSOC real dataset only.

4.3 Summary

In this chapter, we have presented a hybrid architecture for the classification of GAN-manipulated images that exploit localization during training to help the network focus on the most relevant parts of the analyzed image. We built a flooding image dataset to validate the effectiveness of the hybrid architecture on the task of GAN-generated flood image detection. The experiments we carried out revealed the excellent performance of our architecture by considering the localization assistance, always outperforming the baseline method, with very good robustness against various image processing operations and generalization capabilities to new pristine and generated images. Most importantly, the effectiveness of the hybrid architecture implies that directly guiding the model to the manipulation region can be adapted to related AI-manipulated image detection tasks. In the next chapter, we show the results we got by applying the hybrid architecture to the classification of facial attribute editing.

Figure 4.6 – Robustness results (%) in terms of TNR, TPR, and AU-ROC in the presence of JPEG compression with a quality factor of 50 (top row), resizing with factor 0.5 (second row), Median filtering, window size = 3×3 (third row) and Gaussian blur, window right size = 3×3 (bottom row).

Chapter 5

Classification of Synthetic Facial Attributes by Means of Hybrid Classification/Localization and Multi-level Analysis

"Manipulation, fueled with good intent, can be a blessing. But when used wickedly, it is the beginning of a magician's karmic calamity."
T.F. Hodge

In this chapter, we focus on the classification of synthetic facial attribute editing. As we said, given the prominent role played by images depicting persons, the task of discriminating between real and fake face images has attracted increasing attention. Most detectors proposed so far classify any altered face image as a fake. However, in some scenarios, it would be desirable to provide more information to support the judgment that the image is fake, rather than simply telling that the image has been manipulated. Knowing the kind of manipulation applied by the synthetic generator would substantially increase the value of the forensic analysis. This motivates our goal to develop a method for the classification of synthetic face editing. We focus on editing performed via methods based on StyleGAN2 (see Section 2.2.2), which are among the most popular and widespread techniques for facial editing.

Specifically, we first validate the effectiveness of the hybrid architecture presented in the previous chapter for the task of GAN facial editing classification, by running some tests on a public dataset released for SemaFor hackathon competition. Then, we move a step forward and present a method specifically tailored for the classification of facial attributes editing, where the hybrid framework is adopted inside a richer architecture enhanced with multi-level analysis, that permits to rely on both global (image-level) and local (patch-level) features, combined via an attentional feature fusion module. The multi-level method works in two stages and involves two-stage training. In the first stage, the image is divided into six patches corresponding to different face regions, and each patch is analyzed by a different network. Then, an attention module is used to combine the local features extracted from the patches with the global features obtained by analyzing the image as a whole. The output of the attention module is then processed by the hybrid network for simultaneous classification and localization. For a comprehensive evaluation of the performance of the classifier, we built ourselves a dataset of edited face images, where 18 different attributes are manipulated using different methods based on StyleGAN2, namely InterfaceGAN and StyleCLIP (see Section 2.2.2 for more details on these methods).

This chapter is organized as follows. In Section 5.1, we formalize the classification problem, and in Section 5.2 we test the performance of the method presented in the previous chapter on a public facial attribute editing dataset. In Section 5.3, we present an improved multi-scale hybrid architecture and the new facial attribute editing dataset we have built. In Section 5.4, we provide the details of the methodology implementation. Then, we assess the performance of the improved architecture on the new dataset (Section 5.5). We conclude the chapter in Section 5.6 with a summary of our findings.

5.1 Problem Definition

Given a face image x which we know has been generated by a GAN, our goal is to decide if the image has been generated without applying any semantic manipulation (in our case, by simply mapping it into the latent space and reconstructing it without applying any semantic change), or the facial attributes have been manipulated in some way, and, in this case, how. Following the findings discussed in the previous Chapter, we argue that devising an architecture that is also asked to localize the manipulation has a beneficial effect on the classification accuracy and the generalization capability since the localization task forces the network to focus on the most significant parts of the analyzed image. For this reason, the method we are looking for is supposed to produce a twofold output, that is:

$$[p, I_M] = \phi(x), \tag{5.1}$$

where ϕ is the network function, $p \in \mathbb{R}^{1\times C}$ is a C-long vector with the probabilities that x belongs to the C classes the classifier must choose from, p_1 gives the probability that no facial attribute has been changed, in which case the predicted localization mask I_M is a black image indicating no manipulation.

5.2 Preliminary Experimental Results on Facial Attribute Editing Classification

In this section, we report the results of some experiments we run to validate the generality of the HybCls&Loc architecture described in Chapter 4 on a public facial attribute editing dataset, that is, the PFMD introduced in Section 2.4.2[1].

5.2.1 Experimental setting

We considered the problem of closed set classification among the 7 editing types from Table 2.2 including *none*, and excluding *purple_hair*, *angry* and *Taylor Swift* editing. The 'Low' versions in the testing set were used to assess the generalization capability with respect to a mismatch in the strength of the editing, in particular, considering the more challenging case of weaker editing of the attribute during testing. A ResNet50 architecture was used as the backbone for the feature extraction part of the hybrid architecture.

[1] face-manipulation-datasets

To train our model, we used the Adam optimizer with a learning rate of 0.0001 and a mini-batch size of 32. The input size was 224×224×3 and normalized with mean [0.485, 0.456, 0.406] and variance [0.229, 0.224, 0.225] (average values computed on the ImageNet dataset). The training images were augmented by random color transformations (saturation, brightness, contrast) and resizing with scale factors randomly chosen in {0.5, 0.8, 1.2, 1.3} with a probability of 1.0 and 0.4, respectively.

The localization mask, which is necessary to train the hybrid network with the combined loss, is not available in this case. Figure 5.1 shows some examples of difference maps (here playing the role of localization masks) between the edited images and their none version. We observe that the changes introduced by the manipulation are distributed over the entire image (mostly around the hair, mouth and eyes), and are not constrained to the modified facial attribute. We decided to use the difference map as the localization mask to force training to focus on these details. This mask works as a focus of attention, highlighting the region of the image that mostly reflects the attribute change, hence corresponding to the region that should be the main focus for the classifier. More specifically, to get the mask I_G used to guide the classification, we first computed the absolute difference between the none version (reconstructed with no editing) and the edited image in the luminance channel and then converted it to a binary image using the threshold from the opencv library[2], based on the mean of the absolute difference image. Additionally, we also fine-tuned a version of the model by replacing a small percentage of edited images (corresponding to 300 images in each class) in the training set with an estimated 'Low' version, obtained by applying face morphing between the *none* and the edited ('High') images. By aligning and blending the two face images, face morphing allows us to get a synthetic image with stronger editing of the attributes. Specifically, in our case, we first predicted the corresponding landmark points in the *none* and 'High' edited images using Dlib's Facial Landmark detector [130]. Then the triangular mesh with Delaunay Triangulation for each intermediate shape was calculated and used to warp the two input images towards the intermediate shape. Face morphing was performed by using the code available at `https://github.com/Azmarie/Face-Morphing`. An example of *estimated young_low* using *none* and *young_high* is given in Figure 5.2. We see that the *estimated young_low* is visually close to the ground truth *young_low* image.

5.2.2 Results

In the following, we report and discuss the classification results achieved by the hybrid architecture. We did not measure and report the performance in terms of localization (that are generally not good). In fact, we remind that our main interest is the classification, and the addition of the localization module is mainly an expedient to force the network to look at the image areas that are most relevant for the classification task.

Overall, we got an average accuracy equal to 85.20% under matched test conditions and 72.16% under mismatched conditions ('Low' version of edited images considered for testing), respectively. The average accuracy on the 7 classes reveals a noticeable capability

[2] `https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html`

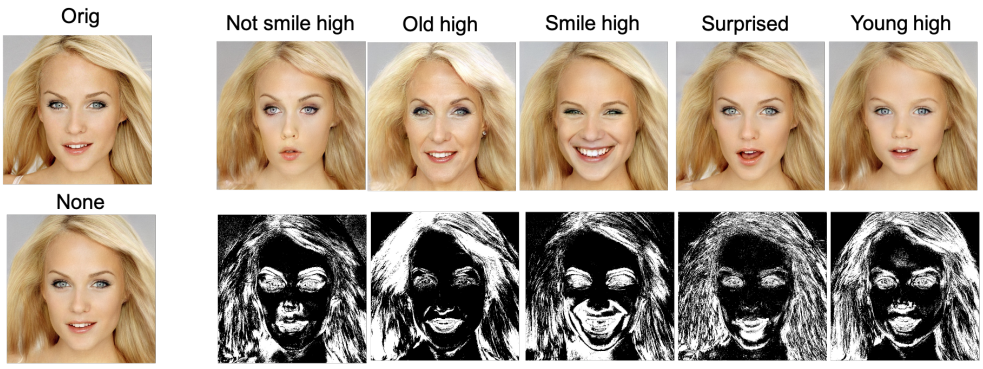


Figure 5.1 – An example from the training set with attention (localization) masks.

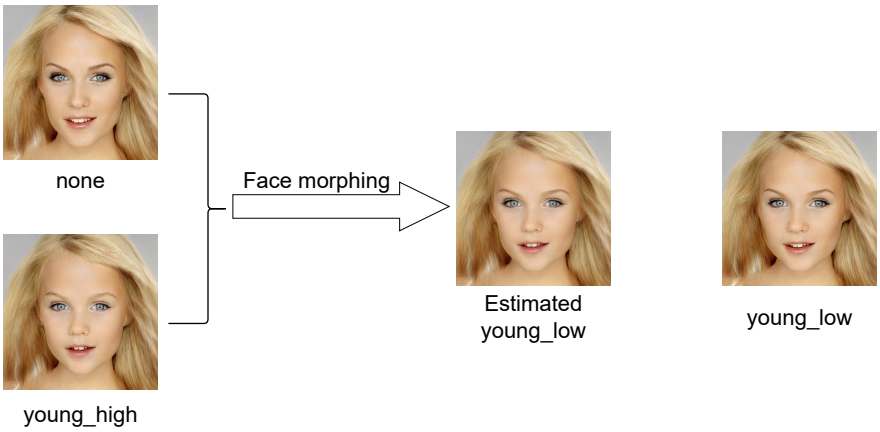
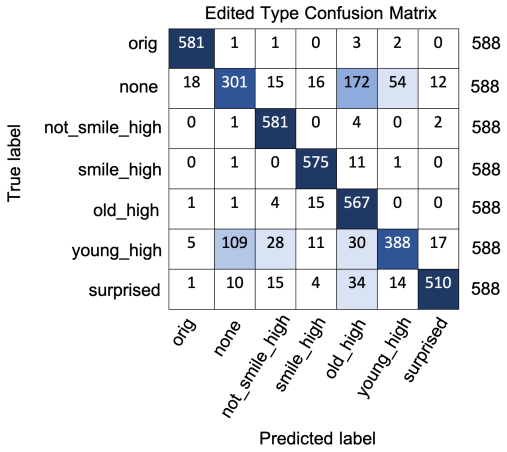


Figure 5.2 – An example of estimated *young_low* by face morphing method (Left) and *young_low* image in test set (Right).

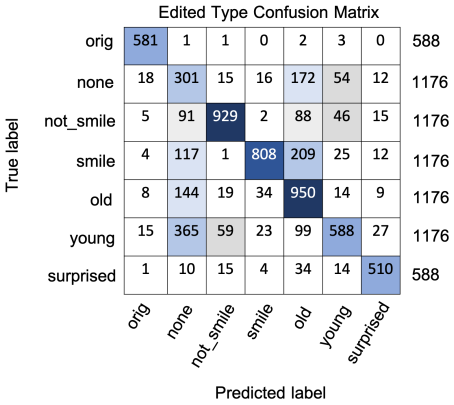
of generalization of the hybrid network[3]. The improvement over the ResNet50 baseline trained for classification only (namely, a 7-class ResNet50 classifier) is 8% on average. This confirms the benefit of the localization branch to aid the classification. We also verified that the accuracy with respect to the 'Low' versions improves, achieving 75.10%, when the model is fine-tuned on the dataset with estimated 'Low' images obtained via face morphing. This proves a gain in accuracy - relevant, even though not big - given by the estimation procedure that we performed to increase the diversity of the training dataset.

The confusion matrix, reporting the performance (accuracy) on each editing type, is shown in Figure 5.3. Looking at the figure, we see that most of the errors are associated with the *none* category. In particular, there is confusion between *none* and *old*, and *young* and *none*, where 172 out of 588 *none* images are classified as *old_high*, and 109 out of

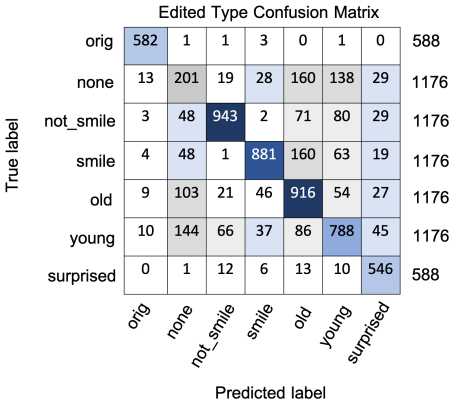
[3]Our method ranked first in the DARPA SemaFor HK3-CP2 challenge task 1, dedicated to the classification of face edit type performed on portrait style images.



(a)



(b)



(c)

Figure 5.3 – Confusion matrices. **Top**: matched tests ('high' versions only). **Bottom Left**: mixed tests (both 'high' and 'low' versions). **Bottom Right**: mixed testing (both 'high' and 'low' versions), when the classifier is fine-tuned with the estimated 'Low' versions.

588 *young_high* images are predicted as *none*. Expectedly, the number of decision errors towards the *none* class in the various cases increases in the mismatched scenario, that is, when the test set contains 'Low' versions of the images (see Figure 5.3 (b))[4]. This is not surprising, since in many 'Low' versions of the images the editing of the attribute is very weak, and it is hard to distinguish between the edited and the *none* version, see the examples in Figure 5.4 for the *young* category, making this task a very challenging one.

At the end, we compare our results with those obtained by the other teams which

[4]We remind that 'Low' versions where not considered for the *surprised* category, that is the only type edited by StyleCLIP.

Figure 5.4 – Examples of *none* (Top) and *young_low* (Bottom) images in test set. Distinguishing these lightly processed images can be very difficult, at least to the human eye.

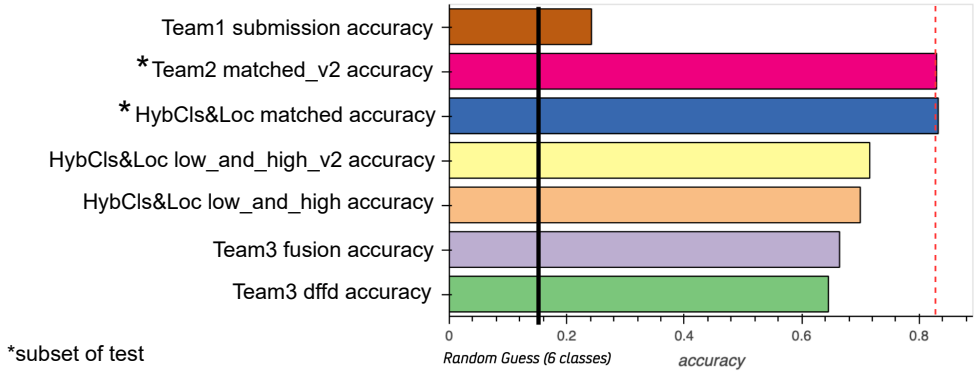


Figure 5.5 – Comparison results with the other SEMAFOR teams participating to the HK3CP task.

participated in the HK3CP2 task (Figure 5.5). Our method, trained on the high version of the manipulations (HybCls&Loc matched in the Figure) performs best when tested on 6 manipulation classes (excluding the original). The mixed version trained with the face morphing method (HybCls&Loc low and high v2) ranks third, improving compared to the model trained with high and low version images without face morphing augmentation (HybCls&Loc low and high).

5.3 A Multi-level Hybrid Architecture for Facial Editing Classification

The preliminary results on the PFMD demonstrate the effectiveness of the hybrid architecture for AI-manipulated image classification. However, the task remains challenging and requires further efforts to improve performance. In this section, we delve deeply into facial editing classification by considering a large dataset and developing an improved hybrid architecture. Based on the observation in Figure 5.1, we knew that the manipulation causes differences mainly in details, which implies that the local patch may have different

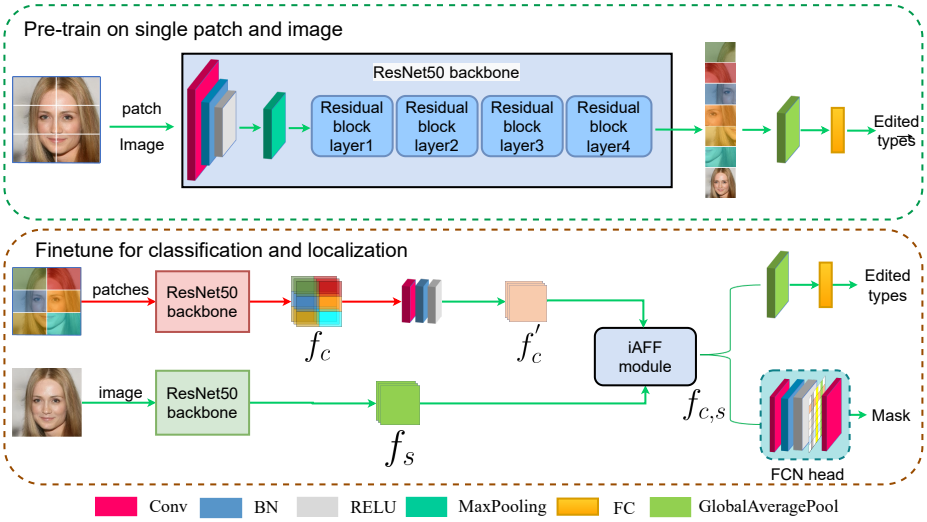


Figure 5.6 – The diagram of the proposed patch-based semantic editing classification and localization method. Different models are obtained in the first step (shown in different colors) and then used to initialize the fine-tuning in the second step. Red arrows indicate frozen weights.

contributions to the classification. To this end, we designed a multi-scale hybrid architecture by analyzing the effects of local and global features through a two-stage training strategy. In addition, to validate the performance of the method, we constructed a new dataset containing 18 edited facial attribute categories following the method used in [14].

5.3.1 Improved multi-scale hybrid architecture

Figure 5.6 shows a schematic overview of the proposed architecture. The network consists of two branches, one working on the entire image and one on 6 local patches. The global features extracted by analyzing the entire image give a rough indication of the manipulation undergone by the image. Such general analysis is then refined by the local features extracted on the image patches. The patches are obtained by dividing the image into 6 parts, namely up head left (upl), up head right (upr), cheek left (cl), cheek right (cr), mouth left (ml) and mouth right (mr). For each patch location, we independently trained a different ResNet50 classification network [138] asking it to classify the image based only on the content of the patch (upper part of Figure 5.6). All the networks are trained by minimizing the CE between the predicted probabilities and the true labels. The 6 networks trained in this way form the upper branch of the final classification architecture (lower part Figure 5.6). The application of the networks to the various patches returns the probability that a certain attribute has been manipulated, namely:

$$p_k = \phi_k(\mathcal{P}_k), k \in \{uhl, uhr, cl, cr, ml, cr, X\}, \tag{5.2}$$

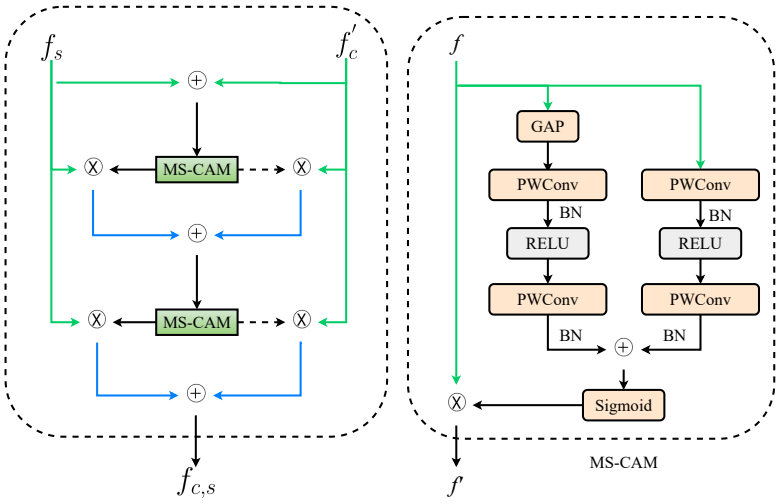


Figure 5.7 – Structure of the iterative Attentional Feature Fusion module (iAFF) [19].

where the a-th component of vector p_k, $p_{k,a}$, indicates the probability that the attribute a has been manipulated, predicted by the network ϕ_k working on patch $\mathcal{P}_k$, and where ϕ_X indicates the network operating on the entire image.

After this initial training step, the weights of the networks working on the local patches are frozen. An attention module modulates the features it produces to adjust the global features of the whole image for the classification and localization tasks. More precisely, as shown in Figure 5.6, the local features $f_{c,k}$ produced by the final convolution layers of $\phi_k(\mathcal{P}_k)$ (having size $D_f \times H_f \times W_f$, where D_f, H_f and W_f are the number of feature maps and the height and width of the feature maps) are first reorganized into a composite feature map f_c, having size $D_f \times 3H_f \times 2W_f$, and where the features of each patch retain the same position of the patches they refer to. Afterwards, a down-sampling unit transforms the feature map f_c into $f_c^{'}$, with size $D_f \times H_f \times W_f$. Eventually, the new local features are merged with the global features using an iterative Attentional Feature Fusion (iAFF) module [19].

As shown in Figure 5.7, the local features $f_c^{'}$ and the global features f_s are first aggregated and processed by a multi-scale channel attention module (MS-CAM), which is a residual-based attention block consisting of global and local branches (right part of Figure 5.7). More specifically, the global branch consists of one global average pooling (GAP) layer, two point-wise convolution (PWConv) layers, two batch normalization layers and a RELU activation function. On the other hand, the local branch directly processes the feature f without GAP. The outputs of the global $G(f)$ and local $L(f)$ branches are normalized by a sigmoid function and multiplied by the input feature f:

$$f^{'} = f \otimes \sigma(G(f) \oplus L(f)), \tag{5.3}$$

where σ denotes the sigmoid function, $\oplus$ and $\otimes$ denote element-wise sum and matrix multiplication, respectively. Afterwards, we obtain the attention-weighted features $f_{c,s}$

by applying the MS-CAM module again:

$$f_{c,s} = f_s \otimes \mathcal{M}(f_s^{'}, f_c^{''}) + f_c^{'} \otimes (1 - \mathcal{M}(f_s^{'}, f_c^{''})), \tag{5.4}$$

where $f_s^{'} = f_s \otimes f^{'}$, $f_c^{''} = f_c^{'} \otimes f^{'}$, and $\mathcal{M}$ denotes the MS-CAM module. Finally, the features $f_{c,s}$ are input to the classification and localization branches. As a last step, we apply an FCN module for the localization task [152]. The FCN is fed with the output of the attention module for localization

$$I_M = \text{Conv}_2(\mathcal{D}_l(\mathcal{R}(\mathcal{B}(\text{Conv}_1(f_{c,s}))))), \tag{5.5}$$

where $\mathcal{D}_l$ indicates dropout, $\mathcal{B}$ batch normalization and $\mathcal{R}$ denotes a *Relu* activation function. Then,

$$p = \text{FC}(\text{GAP}(f_{c,s})). \tag{5.6}$$

The localization loss is defined as the Mean Square Error (MSE) between the predicted map and the ground truth,

$$\mathcal{L}_{loc} = \frac{1}{N_t \times H \times W} \sum_{i=1}^{N_t} \sum_{j=1}^{H \times W} \left(I_{G_{i,j}} - I_{M_{i,j}}\right)^2. \tag{5.7}$$

The classification branch of the architecture is trained to optimize the standard categorical CE loss

$$\mathcal{L}_{cls} = -\frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{j=1}^{C} y_{i,j} \log\left(p_{i,j}\right), \tag{5.8}$$

where y_i is the ground truth (one-hot-encoding) vector of sample x_i. Similarly, p_i is the network output vector for sample x_i.

Finally, the total loss of the proposed network is the combination of the localization loss and the classification loss, that is:

$$\mathcal{L} = \lambda_{cls} \cdot \mathcal{L}_{cls} + \lambda_{loc} \cdot \mathcal{L}_{loc}. \tag{5.9}$$

5.4 Experimental Setting

5.4.1 Implementation details

The dataset we used for the evaluation is FAED (see Section 2.4.3), namely the extended version of PFMD, including 18 facial attributes to be edited and one 'None' class. We trained all the models with the same configuration. For the optimization, we used the Adam optimizer with a learning rate of 0.0001 and a mini-batch size of 32. The input size is 256×256×3 and normalized with mean [0.485, 0.456, 0.406] and variance [0.229, 0.224, 0.225] (as usual, these values are obtained averaging over of the ImageNet dataset). The training images were augmented by random color transformations (saturation, brightness, contrast) and resizing with scale factors randomly chosen in {0.5, 0.8, 1.2, 1.3} with the probability of 1.0 and 0.4, respectively.

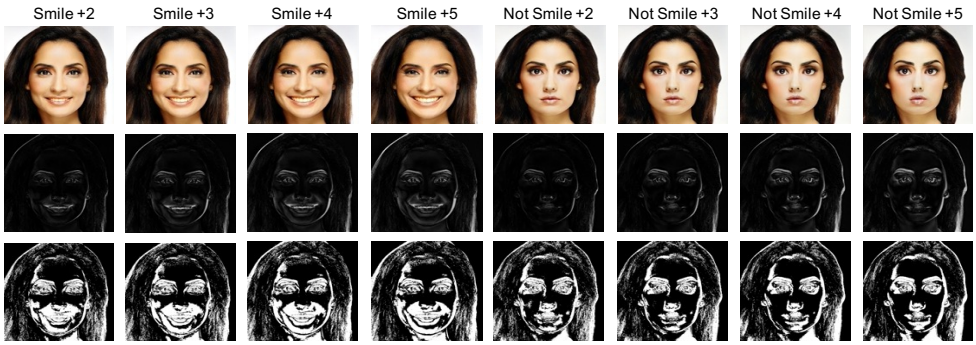


Figure 5.8 – Examples of localization masks for smile and non-smile facial attributes with different editing parameters [11]. From top to bottom: edited images, difference images between none and edited images, and masks after thresholding using the open-cv threshold function (as described in Section 5.2.1)).

Table 5.1 – ResNet50 classification accuracy on patches for matched data (seen in the training set) and mismatched data (InterfaceGAN 3 and 2).

Patch	Matched data	InterfaceGAN 3	InterfaceGAN 2
Head_left	89%	66%	46%
Head_right	59%	33%	23%
Cheek_left	81%	58%	40%
Cheek_right	81%	58%	40%
Mouth_left	77%	58%	42%
Mouth_right	77%	55%	38%
Whole face	**90%**	**68%**	**50%**

We used the difference maps discussed in the previous section as localization masks to train the architecture. Figure 5.8 shows examples of ground truth localization masks. It is clear from the masks that facial attributes cannot be easily distinguished by looking at localization masks alone, but by acting as an attention clue for classification improvement.

5.5 Results

5.5.1 Results on single patches

We start by examining how the location of patches affects the classification. The accuracy results are shown in Table 5.1. Upon inspection of the table, we discover that, in both matched and mismatched circumstances, training a classifier on the up-head left patch

Table 5.2 – Results after fusion and comparison with SoTA.

Methods	Matched case	InterfaceGAN 3	InterfaceGAN 2
ResNet50 [138]	90%	68%	50%
Xception [131]	78%	63%	36%
Efficient-B3 [153]	76%	65%	41%
Efficient-B4 [153]	77%	67%	41%
HybCls&Loc (Chapt 4)	90%	77%	45%
FFD [108]	79%	63%	38%
Prop.	**93%**	**86%**	**55%**

yields results that are comparable to those obtained when analyzing the entire image. This is an unexpected result, that seems to point out that informative artifacts for the manipulation in up head left patch are considerably easier to detect. At the same time, uphead right patches in real and fake images are statistically similar, hence resulting in a lower detection accuracy (59% in the matched case). In addition, the classifiers that were trained on other patches behave similarly, with an accuracy of about 80% for cheeks and 75% for mouths. These observations confirm that different patches provide different, possibly complementary, information, thus motivating us to design a method that fuses local and global features with an attention mechanism.

5.5.2 Results after fusion and comparison with SoTA

To validate the benefits of the improved hybrid architecture, we compared it to two fake image detection and localization methods, namely the HybCls&Loc presented in Chapter 4, and FFD [108] (see Section 2.3.2). These two methods also consider detecting fake images with a localization branch. In addition, we also compare with standard classification networks, including XceptionNet and ResNet50, and also Efficient-B3 and -B4 [153]. EfficientNet is a family of convolutional neural networks designed to obtain high accuracy with fewer parameters and operations, resorting to compound scaling for efficiency [153]. Several versions are available, ranging from B1 to B4. Efficient-B3 is smaller and has lower computational requirements compared to Efficient-B4.

Table 5.2 reports the results of our experiments. First of all, Resnet50 provides the best performance among the plain classification networks. When localization task is also considered, (HybCls&Loc and FFD) the performance tend to improve, however, they do not generalize to InterfaceGAN2. In contrast, the developed method outperforms all the others in both matched and mismatched conditions by 4%, 18% and 5%, respectively.

We also evaluated the robustness of the various methods against Gaussian noise addition with zero mean and variance equal to 0.0003, Median filtering (3×3), Gaussian blur (3×3) and JPEG compression with quality factor 50. The results we got are given in

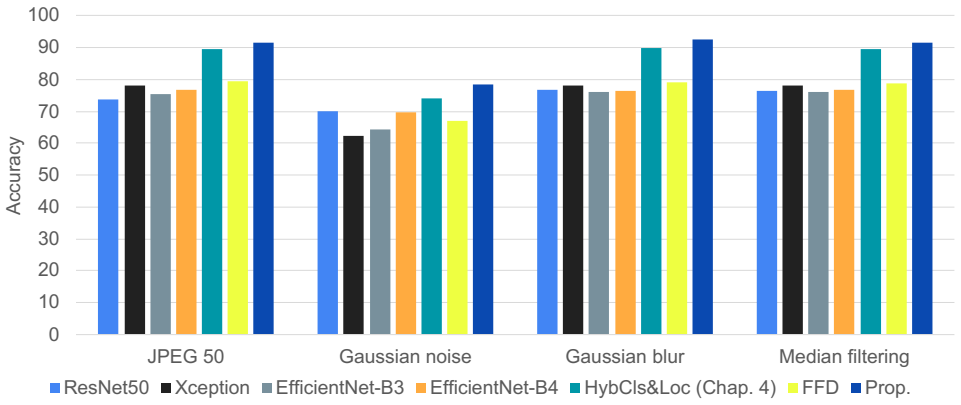


Figure 5.9 – Robustness comparison in matched conditions.

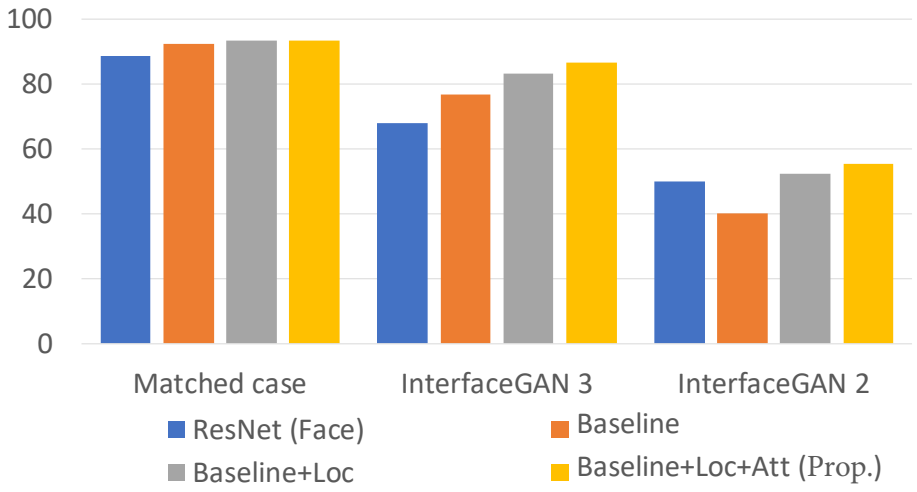


Figure 5.10 – Ablation results for each component.

Figure 5.9. Once again, the developed method outperforms all the others, showing good robustness against all processing types.

5.5.3 Ablation study

As a last experiment, we carried out an ablation study to evaluate the impact of each component of the system on the accuracy. In the **Baseline** model, we directly concatenated the global features f_s and the local features $f_c^{'}$ without considering the localization task and the attention module. For the model **Baseline+Loc**, we added a localization branch to the Baseline model. The final model, **Baseline+Loc+Att**, is the one developed by considering both the localization task and the attention module.

The results we got are shown in Figure 5.10. First of all, we can see that the use

Table 5.3 – Ablation study of the effect of balancing loss weights.

λ_{cls}	λ_{loc}	Matched case	InterfaceGAN 3	InterfaceGAN 2
0.5	0.5	**93.7358**	85.2600	54.6055
0.4	0.6	93.6138	85.0955	53.8540
0.3	0.7	93.2697	84.4475	53.2472
0.2	0.8	93.4175	**86.3915**	**55.4570**

of global and local characteristics contributes to the improvement in both matched and mismatched cases. For instance, compared with the ResNet50 model trained on the whole image, the developed network **Baseline+Loc+Att** achieves a gain of more than 4% in the matched case. The gain on never-seen manipulated images produced by InterfaceGAN3 is impressive (18%), and still good for InterfaceGAN2 (5%) manipulation, demonstrating that local features give important clues for the improvement of both seen and unseen data. Likewise, by referring to Baseline and **Baseline+Loc**, we see a large improvement in mismatched cases. Eventually, a noticeable improvement is also made by considering the attention module on top of **Baseline+Loc**.

We also analyzed the effects of the hyper-parameters contained in the loss function of **Baseline+Loc+Att**, obtaining the results shown in Table 5.3. Again, we see that increasing the contribution of the localization task during training improves the performance in the mismatched cases. This behavior is consistent with our assumption that adding a localization task improves classification.

5.6 Summary

In this chapter, we focused on the classification of facial attribute editing tasks. First, we tested the method from Chapter 4 on a public dataset. Then, we improved the hybrid architecture through multi-scale analysis on a new large facial attribute editing dataset we built. Both hybrid architectures demonstrated good generalization to unseen samples with the same manipulations considered during training but edited with different (lower) editing strengths. They also exhibited strong robustness against image post-processing.

During our tests, we noticed that when facing a new manipulation outside the training set, the architecture predicts it as one of the known manipulations, which is undesirable in real-world applications. To address this issue, a model thought to work *in the wild* must recognize that the image under analysis has undergone a new type of manipulation and take appropriate actions. This challenge, known as open-set classification, is tackled within the second part of the thesis.

Part II

Open Set Classification and Attribution of AI-generated Images

Abstract

In this part of the thesis, we focus on the development of forensic techniques capable of working in open-set scenarios. By focusing on the face domain, we develop some methods for the classification of facial attribute manipulation and for synthetic image attribution. In particular, by focusing on the classification of synthetic facial attributes in open-set scenarios, we first introduce a method for classification with a rejection option. The proposed method combines the use of a Vision Transformer (ViT) with the hybrid approach described in the first part of the thesis. Rejection is performed by considering several strategies based on the analysis of the model output scores. The effectiveness of the method has been validated also for open-set attribution of synthetic images to the generating architecture.

Furthermore, we propose a novel verification framework that relies on a Siamese Network (SN) architecture to address the open-set synthetic image attribution task. We consider two different verification scenarios: in the first one, the system determines whether two images have been produced by the same architecture or not, while in the second one, the system verifies a claim about the architecture used to generate a synthetic image, utilizing one or multiple reference images generated by the claimed architecture. We also apply the SN-based verification model to build a classifier with a rejection class.

Finally, we present a new framework for open set classification, named BOSC (Backdoor-based Open Set Classification) that relies on the concept of backdoor attacks to design a classifier with a rejection option. BOSC works by purposely injecting class-specific triggers inside a portion of the images in the training set to induce the network to establish a link between class features and trigger features. The behavior of the model trained in this way with respect to samples with triggers is exploited at test time to perform sample rejection. We apply BOSC to both the attribution of synthetic architecture and the classification of facial attribute editing.

Chapter 6

Introduction to Open Set Recognition and Synthetic Image Attribution

In a world of diminishing mystery, the unknown persists.
Jhumpa Lahiri, The Lowland

In this chapter, we introduce the problem of open-set recognition (OSR) and open-set image attribution, and briefly review the relevant literature on these topics. Open set recognition is a machine learning approach designed to identify and correctly handle out-of-set samples belonging to unknown classes by introducing a rejection option. This capability is particularly significant for forensic tasks because it enhances the reliability and robustness of forensic tools in real-world applications. Without open set recognition, a system might incorrectly classify out-of-set inputs as belonging to in-set classes, leading to false identifications and potentially severe consequences in forensic investigations. Open set recognition helps forensic tools to adapt to and accurately process a broader range of inputs, ensuring more trustworthy and accurate results in the dynamic and evolving field of digital forensics.

As mentioned in the introduction, synthetic image attribution is a very important problem that has attracted the interest of the multimedia forensic community. As generative models continue to evolve rapidly, new architectures for creating synthetic images are constantly emerging. For this reason, extending open-set classification to synthetic image attribution is of paramount importance.

This chapter starts in Section 6.1, with an overview of the most relevant works in the field of open-set recognition. Then, Section 6.2 reviews the state-of-the-art of synthetic image attribution in both close-set and open-set scenarios.

6.1 Prior Art on Open Set Recognition

The seminal work on OSR was presented by Scheirer et al. in [154], where the authors addressed the problem of determining whether an input belongs to one of the classes used to train a machine learning model or not. Jain [155] and Scheirer [156] proposed methods based on statistical Extreme Value Theory (EVT [157]). These methods, however, are tailored to specific tasks and lack scalability. A method to address OSR with deep neural networks, named OpenMax, was presented by Bendale et al. in [158]. An extra class is added for the prediction to model the unknown class case by adapting meta-recognition concepts to the activation patterns in the penultimate layer of the network for unknown

modeling. EVT was used to estimate the probability that the input is an outlier. In addition, several works have shown that, in many cases, simple strategies based on the softmax probabilities or the logit can also effectively judge if a sample comes from an unknown class [159], e.g., by exploiting the fact that the maximum output score tends to be smaller for out-of-set inputs [160,161], or that the energy of the logit vector tends to be lower [162]. The use of the Maximum Logit Score (MLS), in particular, has been proven to achieve very good rejection performance [161] and has been adopted for classification with rejection in several papers [158].

Other works tried to find methods to optimize the representations of in-set and out-of-set samples in the feature space. In [163], Yang et al. designed a suitable embedding space for open set recognition using convolutional prototype learning that removes softmax and implements classification by finding the nearest prototype in the Euclidean norm in the feature space. Multiple prototypes are used to represent different classes. The feature extraction and the prototypes are jointly learned from the data. Similarly, Miller et al. [164] exploit a distance-based loss to enforce class features to form tight clusters around predetermined class-specific centers. The distance to class centers is used at test time to reject samples from unknown classes and classify the inputs belonging to known classes. The method proposed in [116] exploits a different learning framework for OSR, called reciprocal point learning, that introduces the unknown information to the learner via the concept of reciprocal point to learn more compact and discriminative representations and reduces the risk of misclassifying unknown classes as a known one.

Another class of works exploits reconstruction errors obtained via Auto-Encoders (AE) for OSR, assuming that lower reconstruction errors are obtained for known classes than for unknown ones [165, 166]. In particular, CROSR [165] jointly trained the network to classify and reconstruct the input data. Then, the joint distribution of the latent representations and activations is used for OSR rejection using the OpenMax method. Oza and Patel [166] proposed a class-conditioned autoencoder (C2AE) framework to reconstruct the images, conditioning the reconstruction to the class (conditioning label). Similarly, variational autoencoders were exploited in [167], where the authors propose a conditional Gaussian distribution learning framework to detect unknown samples by forcing latent features to approximate Gaussian models. Huang et al. [168] proposed to combine the autoencoders with the prototype (PCSSR) and reciprocal learning (RCSSR). Class-specific autoencoders are trained to reconstruct the data based on label conditioning, and the pixel-wise reconstruction errors corresponding to the predicted class, together with semantic-related features, are used for unknown sample rejection.

Yet another class of methods is based on adversarial modeling and generative models. Methods based on generative adversarial networks resort to GANs to produce unknown-like samples to be used during training. These methods work under the assumption that a large number of unknown samples (unlabeled) are available during the training process. For instance, G-OpenMax [169] combines the use of generative adversarial networks with the OpenMax method, achieving good performance on the classification of handwritten digits. Neal et al. [170] proposed an ad-hoc data augmentation strategy, relying on GANs, to generate samples close to the training set examples that are then used to augment training, reformulating the OSR as a classification problem with one additional class.

A similar idea is explored in [171] (OpenGAN) and [172] (DASI). Finally, in ARPL [173] and AKPF [117], reciprocal point learning and prototype learning are enriched by an adversarial mechanism that generates confusing training samples. Specifically, the generated samples are used to optimize the feature space and reduce the so-called open space risk by restricting the unknown samples in the reciprocal points space [173] or learning a kinetic boundary to increase the intra-class compactness and the inter-class separation [117].

Overall, works on OSR mainly improve the open-set performance in three ways: i) developing robust classifiers for closed-set tasks [161, 163, 164] and using MLS or similar metrics to reject out-of-set samples, ii) setting thresholds on the AE reconstruction error [165–168], and iii) incorporating generated open-set samples for training purposes [117, 169–173]. However, for the open set synthetic image attribution problem, a good classifier on closed-set samples may overfit to known samples and have reduced effectiveness in the open-set case. On the other hand, the reconstruction approach may be suboptimal. This is due to the fact that the fundamental distinctions between samples generated by known and unknown models consist of subtle, visually imperceptible statistical traces. These traces are often too weak to be effectively thresholded, posing a challenge to methods relying only on the reconstruction error. Finally, reducing the open space risk for synthetic attribution is challenging with a single generator. Generators are designed to produce images with specific semantics and may not be able to generate diverse open-set fingerprints.

6.2 Synthetic Image Attribution

Tracing the origin of AI-generated images by identifying the model or the architecture that generated them is extremely relevant to combat misinformation and piracy. In addition, understanding the groups, individuals, or entities behind the generation of AI-generated images is essential to pursue responsible individuals, detect and mitigate malicious activities, and inform policy and regulatory responses to the challenges posed by synthetic media. In particular, in some cases, it is important to know the specific model or the type of architecture used to produce a fake image. Given a synthetic image, a synthetic image attribution system should be able to predict the generator that has been used to produce the image, see Figure 6.1. Attribution can be performed at different levels. A system can attribute an image to the specific model that generated it, or to the architecture the model relies on.

Synthetic image attribution at the model level has been addressed through both active and passive methods. Active methods involve injecting specific information, e.g., user-specific keys [174] or artificial fingerprints [175,176], into the generated images during the generation process. These fingerprints or keys are subsequently used during the verification to identify the model. On the other hand, passive methods rely on the presence within the synthetic images of intrinsic artifacts, namely model fingerprints, that are peculiar to the specific model used to generate them. Passive methods have been developed in [176–184]. In particular, Marra et al. [177] first revealed that each GAN leaves a specific fingerprint in the images it generates. The average noise residual image can be taken as a GAN fingerprint. Then, Yu et al. [178] replaced the hand-crafted fingerprint formu-

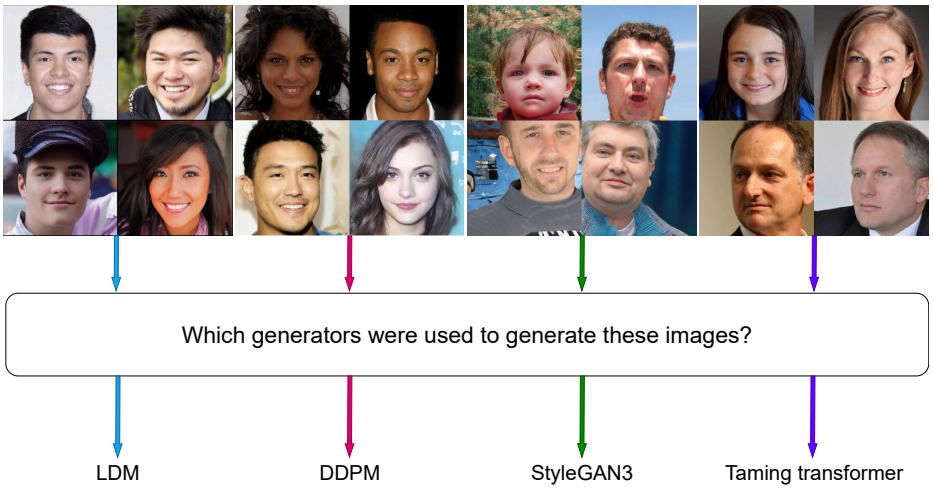


Figure 6.1 – The synthetic image attribution task. Images are attributed to the source generator that produced them.

lation in [177] with a learning-based one, decoupling the GAN fingerprint into a model fingerprint and an image fingerprint. Joslin et al. [179] analyzed the fingerprints of the up-sampling convolutional operations in the frequency domains and performed model-level attribution exploiting features in such a domain.

In addition to model-level attribution, researchers have started proposing approaches that address the attribution problem at the architecture level. The goal is to attribute the synthetic images to the source architecture regardless of how this architecture has been trained and fine-tuned, that is, regardless of the training strategy and setting, and regardless of the dataset of images used for training. Frank et al. [182] first proposed to attribute the synthetic images to the source architecture by relying on the DCT coefficients. Xuan et al. [181] proposed a deep learning framework based on triplet loss, where the image is compared with sample images available for each architecture in a template image library. The prediction is made in favor of the architecture that has the highest similarity to the test image. Yang et al. [183] observed globally consistent traces across models of the same architecture and proposed an approach to extract globally consistent features based on a patchwise contrastive learning framework.

All the above methods focus on closed-set scenarios, that is, they assume that the images the system is asked to analyze at test time belong to a *in-set* model or architecture, that is, to a model or architecture seen at training time. In a real-world scenario, also considering the continuous progress of AI technology, with new generative architectures continuously appearing, this may not be the case. Therefore, it is important to develop systems capable of working in an open-set scenario.

When we started our research, very few methods had been developed to address the specific problem of synthetic image attribution in an open-set setting. The first step in this direction, preceding our proposal described in Chapter 7, was made in [27], where the authors resort to a semi-supervised learning framework that exploits labeled samples

from in-set classes and unlabeled samples from out-of-set classes. These samples are used to train a system that, at every step, classifies in-set samples and clusters the out-of-set samples, assigning new labels to the new clusters. Recently, Sun et al. [28] addressed the open-world attribution problem by employing a contrastive learning strategy within a semi-supervised framework. The semi-supervised learning framework offers the advantage of utilizing all available data for training without the need for a large labeled dataset. However, it is necessary to retrain the model when new images generated by novel architectures emerge.

Another approach to handle out-of-set samples is to perform rejection of them, as we described in Chapter 7. Since then, several works have started addressing the open set attribution problem by exploiting sample rejection to prevent misclassification [118, 185, 186]. In particular, Fang et al. [185] addressed the open-set synthetic image attribution task by using a distance-based approach. The predictions are rejected (the test sample is judged as coming from an unknown class) when the minimum distance between the test sample and the centroids of known classes in the features space exceeds a predefined threshold. Following an idea similar to generative-based OSR works, Yang et al. [118] introduced a progressive open space expansion framework (POSE) to simulate the open space of unknown models, that is, the feature space where open-set samples lie through a set of lightweight augmentation models. On the other hand, Yang et al. [186] simulated fingerprints of generative models within images using several convolutional layers. They attributed the source of the generated images by analyzing these simulated fingerprints and incorporated a rejection option into their approach.

To summarize, all methods proposed so far strive to enhance the feature representation of the model to learn a condensed in-set class space by using representative network [185], unknown out-of-set sample simulation [118], and explainable forensic clues [186], thereby reducing the open space risk. In Chapter 9, we propose an alternative approach, borrowing the idea of backdoor attacks to remap the features of known classes to a target class position, by injecting class-specific triggers. We also considered the possibility of adopting a verification framework (Chapter 8) that can be used to verify whether two, possibly out-of-set samples, have been generated by the same model/architecture or not.

Chapter 7

Open Set Classification of AI-based manipulations and Attribution via a ViT-based Hybrid Architecture

"The unseen enemy is always the most fearsome."
George R.R. Martin

In the previous chapters, we demonstrated the effectiveness of the hybrid framework in enhancing generalization and robustness, applying it to both detection and classification tasks. However, training a multi-class classifier presents limitations in real-world scenarios when unknown out-of-set samples are encountered. However, classifiers that are designed to recognize the classes they where trained on, can not analyze properly images from classes that have never been seen, resulting in unreliable classification results. In this chapter, we present a technique for classification with a rejection option, to address the problem of classification of AI-based manipulations in an open-set setting. The same method is also exploited for the attribution of AI-generated images.

More specifically, we improved the methodology in previous chapter by considering ViT module. The proposed technique combines the use of a ViT with the hybrid approach for simultaneous classification and localization, described in the previous part of the thesis (see Chapters 4 and 5). Specifically, the features used for the classification are extracted by a CNN, and then a ViT module is used to exploit the correlation of the feature maps, via a self-attention mechanism. To aid in the cases where the images are locally manipulated, the same features analyzed by the ViT-based classification heads are also utilized by a localization branch via an FCN module, using the localization mask to provide guidance during training (see Chapter 4). A dedicated module rejects or accepts the samples based on the analysis of the output of the classification model. In particular, following the general literature on Open Set Recognition, rejection is performed by considering the maximum softmax probability (MSP) [160], the MLS [161], and the OpenMax [158] approach. The use of ViT is motivated by a recent trend in machine learning towards the use of this kind of architecture, which has been shown to improve the performance of open-set classification compared to standard CNN architectures [187, 188]. We validated the performance of the ViT-based hybrid architecture on two tasks: facial attribute editing and synthetic image attribution. The results confirm that our technique yields superior open-set performance without impairing the accuracy in a closed-set setting, outperforming state-of-the-art methods.

This chapter first presents the ViT-based hybrid classifier with rejection in Section 7.1. Section 7.2 describes the experimental methodology and setting. The results and the

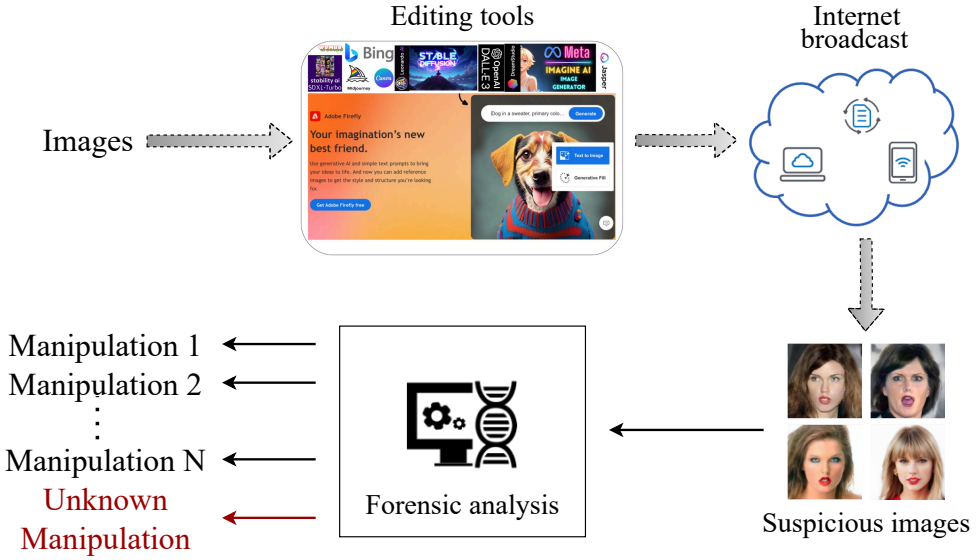


Figure 7.1 – Open set classification of AI image manipulations (with rejection option). The figure refers to the case of facial manipulation classification. In the synthetic image attribution task, classification is made among the manipulation/generative models.

comparisons with the state-of-the-art, for both the tasks of face editing classification and synthetic image attribution, are reported and discussed in Section 7.3.

7.1 A ViT-based Hybrid Classifier for Open-Set Classification

The general problem of open-set classification of synthetic manipulation addressed in this chapter is illustrated in Figure 7.1. For simplicity, we refer to the classification of AI-based manipulations. The situation is similar in the case of open-set attribution, in which case classification is made among the models used to produce the images instead of the type of manipulation/editing applied. A classifier with a rejection option classifies the type of manipulations among those known by the classifier (in-set manipulations) while simultaneously rejecting out-of-set samples, that is, samples that were subject to a different manipulation or generation procedure with respect to those in the in-set.

Below, we introduce the notation and formalism that we will follow in this chapter (this notation will also be used in Chapter 9, where we present another approach for classification with a rejection option). Formally, let x denote the input image and y be its true label. If we let C be the number of in-set classes and $\mathcal{C}_l = \{1, 2, \ldots, C\}$, the model is expected to return a label $\hat{y} \in \mathcal{C}_l$ for in-set samples, and a rejection label $\hat{y} = \Re$ for samples belonging to out-of-set classes. We sometimes need to refer to the closed-set prediction, that is, the prediction of the classifier under the assumption that the analyzed sample is an in-class one (e.g., before a rejection decision is made). Such closed-set prediction is

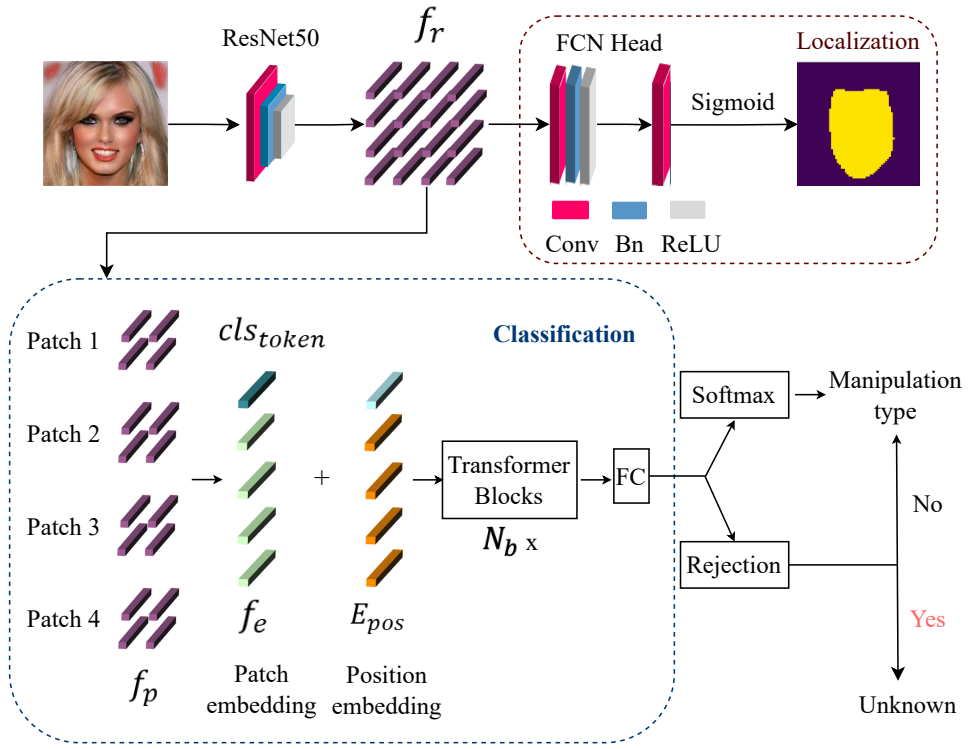


Figure 7.2 – Overall architecture of the proposed method.

indicated by y^*. A localization mask I_M associated with the manipulation is also provided at the output of the system presented in Section 7.2.2, highlighting regions of interest (like an attention mask).

7.1.1 Proposed architecture

The overall architecture of the proposed method is shown in Figure 7.2. The network is composed of two branches for classification and localization, respectively. A ResNet50 network is used as the backbone for feature extraction. Following [16], a modification of the original ResNet architecture is considered, where we removed the sampling operation in the first convolutional layer of the network, setting the stride parameter to 1, with the kernel size fixed to 3. The features are then input to a transformer-based module performing the C-class classification and an FCN head for the localization, as detailed below. Hence, in our scheme, the input sequence to the ViT is formed by the feature maps of the CNN instead of raw image patches [26].

ViT-based classification module. Let f_r denote the vector of extracted features. Then, $f_r \in \mathbb{R}^{H_f \times W_f \times D_f}$. The following preprocessing is applied before feeding the ViT module. For a given patch size P, f_r is first reshaped into a sequence of $N_p = H_f W_f / P^2$, $P \times P \times D_f$ patches. The special case $P = 1$ corresponds to the case when the input sequence is obtained by simply flattening the spatial dimensions of the feature map and

projecting it to the transformer dimension. The input sequence obtained after these flattening layers is $f_p \in \mathbb{R}^{N_p \times \left(P^2 D_f\right)}$. Following the general procedure with ViT, patch embedding is performed by mapping the image patches to D_p dimensions via linear projection. $f_e = f_p \cdot E_p$ is the output, of shape of $N_p \times D_p$, obtained after the patch embedding operation, where E_P denotes the embedding matrix, $E_p \in \mathbb{R}^{\left(P^2 D_f\right) \times D_p}$. A placeholder data structure cls_{token}, used to store information extracted from other tokens in the sequence f_e, is perpended to the beginning of the input sequence f_e (randomly initialized). Position embeddings $E_{pos} \in \mathbb{R}^{(N_p+1) \times D_p}$ are added to the patch embeddings to retain positional information, thus getting the sequence of vectors $\{cls_{token}, f_e\} + E_{pos}$, that is then fed to a standard transformation encoder, like those used in natural language processing. The transformer encoder is composed of N_b identical transformer blocks, each one consisting of alternating layers of multi-headed self-attention (MHA) and multi-layer perception (MLP) blocks, with a normalization layer applied before every block, followed by residual connections after every block, see [26] for more details. Finally, a fully connected layer is attached to the transformer encoder, whose output is the predicted probability vector p for the N enclosed classes.

FCN localization module. The extracted features f_r are also input to an FCN in charge of estimating the manipulation mask I_M. The FCN consists of two convolutional layers, a batch normalization layer, a ReLU layer and finally, a sigmoid layer to map the values in the $[0, 1]$ range. As we mentioned, the main reason for the introduction of the localization branch is to guide the classification and force the network to focus on the most significant parts of the image (Chapter 4).

The overall C-class classification architecture is trained end-to-end by minimizing a combination of the CE loss, associated with the classification task, and the localization MSE, respectively. Formally, $\mathcal{L} = \lambda_{cls} \cdot \mathrm{CE}(y, p) + \lambda_{loc} \cdot \mathrm{MSE}(I_G, I_M)$, where I_G denotes the ground truth localization mask, and λ_{cls} and λ_{loc} balance the trade-off between localization and classification.

The impact of each part of the architecture, and in particular, the localization branch and the ViT module, is assessed in the experiments. For the ViT, we considered $N_b = 4$ transformer blocks. Different patch sizes P of the ViT were considered in our experiments.

7.1.2 Rejection of out-of-set samples

In order to detect samples whose manipulations do not belong to the in-set, we considered three rejection strategies, two of which, namely MSP [160] and MLS [161], reject the samples by analyzing the model output after or before the softmax activation layer, respectively, and OpenMax [158].

With MSP and MLS, lower scores associated with the predicted class reflect the uncertainty of the network prediction, providing evidence that the analyzed sample belongs to an out-of-set class. Then, the final output of the classifier, $\hat{y}$, is obtained as follows

$$\hat{y} = \begin{cases} y^*, & \text{if } \xi > \nu \\ \Re, & \text{otherwise} \end{cases} \tag{7.1}$$

where $\xi = \max(h)$, with h denoting the model output score (namely, the softmax probability, i.e., $h = \phi(x)$, in the MSP case, and the logit score, $h = \phi^{-1}(x)$, in the case of

Table 7.1 – Summary of the 19 editing classes (18 + 'None').

Editing methods	Edit types
PTI	T0: None (Reconstructed)
InterfaceGAN	**Expression** (T1-T2): Smile, Not smile, **Aging** (T3,T4): Old, Young
StyleCLIP	**Expression** (T5, T6): Angry, Surprised **Hairstyle** (T7-T12): Afro, Purple_hair, Curly_hair, Mohawk, Bobcut, Bowlcut **Identity change** (T13-T18): Taylor_swift, Beyonce, Hilary_clinton, Trump, Zuckerberg, Depp

MLS), and ν is a predefined threshold. When the OpenMax is adopted, the output of the classifier is accepted, $\hat{y} = \arg\max(h)$ if $p_o < \nu'$, where p_o is the probability of the sample being an outlier, estimated by the method, and ν' is the decision threshold. Otherwise, it is rejected ($\hat{y} = \Re$).

7.2 Experimental Setup

7.2.1 Datasets

We utilized the FAED_v2 dataset, introduced in Section 2.4.3, to evaluate the designed system. This dataset includes 5,993 real images, each edited to exhibit 18 different facial attribute types. We remind that four facial attributes are edited with InterfaceGAN, and 14 facial attributes are edited with StyleCLIP. The '*None*' type corresponds to the case of an image reconstructed with no editing (obtained via the PTI inversion method). We exploited a pre-trained face parsing model [189] to group the various edited attributes into four categories: expression, aging, hairstyle, and identity change. Table 7.1 provides an overview of the dataset, where the editing types are grouped into the four categories, and an identifier - used in the following to refer to them - is assigned to the editing types. In addition, we also evaluated the performance of the system in an open-set synthetic image attribution task and we considered five generative architectures, including StyleGAN2, StyleGAN3, LSGM, Latent diffusion and Taming transformers presented in SIAD dataset in Section 2.4.4.

7.2.2 Experimental setting

To train the model, we split the dataset of real images as follows: 4400 images were used to generate the editing used for training, 1592 for those used for testing, for a total of 83600 (4400×11) images for training and 30248 (1592×19) for testing. Cross-validation was implemented during training by randomly splitting the training set in 4000×11 images used for training and 400×11 images used for validation, every 10 epochs. The

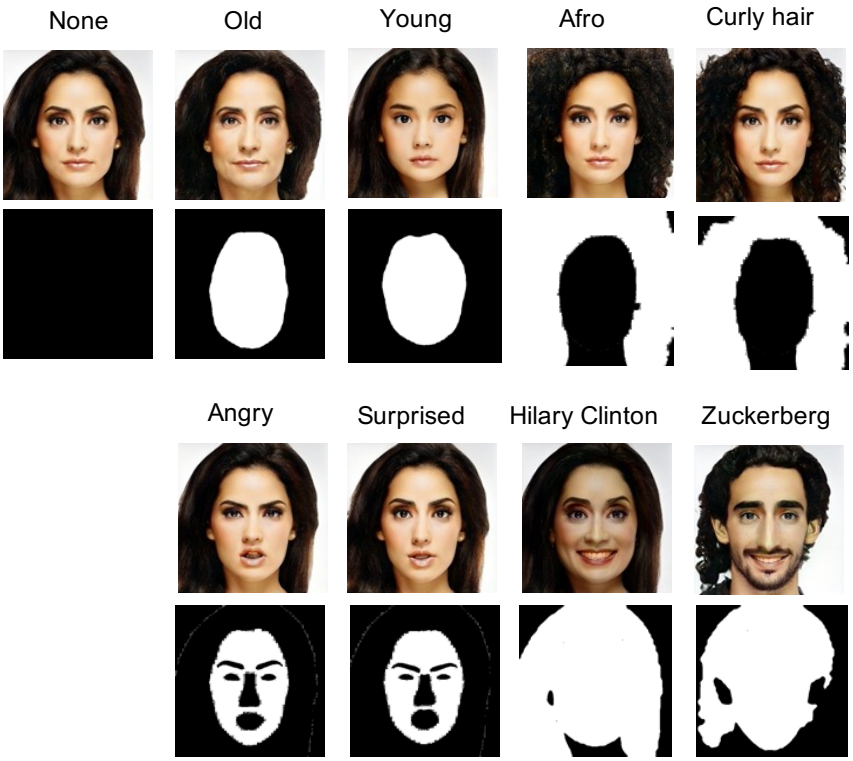


Figure 7.3 – Examples of images and masks from each category obtained for images manipulated with different editing types. From left to right: 'None', aging (2), hairstyle (2), expression (2), identity change (2).

training was performed via Adam optimizer with learning rate 10^{-5} and batch size 32 for 100 epochs. The input size was set to 256×256×3. We ran comparison with the state-of-the-art methods in the field of OSR, i.e., GCPL [116], RPL [163], ARPL [173], CAC [164], PCSSR and RCSSR [168], mentioned in Section 6.1. All these methods were trained using the code released by the authors on our dataset with default setting and input size 224×224.

As for synthetic image attribution, for each architecture, the images were split in proportion 35000:5000:10000 for training, validation and testing, respectively. We trained the model by using the same optimizer, learning rate and batch size detailed above, for 50 epochs.

In the case of the classification of GAN face editing, the manipulation was performed locally and the localization branch was employed to guide the training. The localization masks highlighting the regions of interest in the images used to train the model were obtained based on the editing category (expression, aging, hairstyle, and identity change). In particular, instead of following the approach adopted in Chapter 4 and 5, we decided to use a different mask for every category and not specialize the mask on the editing type. Based on some tests, a general semantic-related attention mask works better than

a very detailed and fragmented mask (like the one obtained from the difference image). In particular, we focused on the whole face area for aging editing while we considered the hair region for hairstyle editing. For identity editing, the focus area covers the whole face and hair since both of them are relevant in the characterization of identity. Finally, for expression editing, the profiles of the mouth, eyes, eyebrows and nose are highlighted in the masks by removing the corresponding segmented regions, which are highly related to expressions. Figure 7.3 shows some examples of masks.

7.3 Results

In this section, we report and discuss the results we got for the classification of GAN face editing and GAN attribution. Most of the experiments, in particular the comparison with general state-of-the-art methods for OSR in machine learning, as well as an ablation study on the impact of the various elements of the proposed architecture and parameters, are reported for the former case. This is the case, in fact, where all the components of the ViT-based hybrid network are considered, including the localization branch.

7.3.1 GAN face editing classification

We carried out our experiments by considering 10 different configurations of in-set and out-of-set editing types, referred to as Config F0-F9. In each case, 11 editing types are considered in the in-set classes, while the remaining 8 are treated as out-of-set. Table 7.2 reports Config F0-F4 configurations, with the 'None' class always included as in-set (the identifiers of the editing types are reported). Configurations Config F5-F9 are obtained from Config F0-F4 by switching the first in-set and out-of-set type, hence with the 'None' class in the out-of-set.

Table 7.2 – Splitting of editing types considered in the various configurations.

Configs	In-set	Out-of-set
F0	T0, T2, T3, T5, T6, T7 T8, T9, T13, T14, T15	T1, T4, T10, T11 T12, T16, T17, T18
F1	T0, T1, T2, T5, T6, T13 T14, T15, T16, T17, T18	T4, T3, T7, T8 T9, T10, T11, T12
F2	T0, T1, T2, T5, T6, T7 T8, T9, T10, T11, T12	T4, T3, T13, T14 T15, T16, T17, T18
F3	T0, T1, T2, T3, T4, T11 T12, T13, T14, T15, T18	T5, T6, T7, T8 T9, T10, T16, T17
F4	T0, T1, T3, T4, T6, T10 T12, T15, T16, T17, T18	T2, T5, T7, T8 T9, T11, T13, T14

Table 7.3 reports the closed-set performance (Accuracy) and the open-set performance (AU-ROC) for the various configurations, achieved with the three rejection strategies. The average accuracy of the classification on the samples belonging to the $C = 11$ known

Table 7.3 – Performance in closed-set and open-set settings, using different rejection strategies, for different configurations (F0-F9). The Accuracy is reported for closed-set, while the AU-ROC (%) is reported for open set.

Configs		F0	F1	F2	F3	F4
Closed-set		88.99	94.68	87.03	94.34	95.25
Open-set	MSP	79.35	79.63	71.49	84.54	83.97
	OpenMax	81.83	81.89	**81.39**	74.86	81.34
	MLS	**85.34**	**91.36**	78.34	**91.98**	**89.75**
Config		F5	F6	F7	F8	F9
Closed-set		92.65	95.51	89.24	94.94	95.94
Open-set	MSP	82.29	87.29	75.50	84.49	83.30
	OpenMax	78.62	86.20	**83.72**	85.00	83.73
	MLS	**88.05**	**95.23**	82.43	**93.13**	**91.77**

classes is 92.86%. Regarding the open-set performance, the MLS is the strategy that gives the best results. In particular, with MLS, we got AU-ROC = 88.74% on average, in contrast to 81.19% and 81.86% for MSP and OpenMax, respectively. Notably, the configurations for which we achieved the best closed-set performance correspond to those performing better in the open-set scenario. Therefore, in the following, unless stated otherwise, we report the results of the MLS.

In Figure 7.4, we report an example of predicted masks in the various cases for the Config-F0 configuration. Although the localization has been considered only to supervise the training, like an attention mechanism, and not for localization purposes, by looking at the figure, we can observe that in many cases, the method is able to produce similar masks, namely masks with a similar white region (the focus of attention), for both in-set and out-of-set images, for editing types belonging to the same category (see Table 7.1). This indicates that the network tends to look at areas of the image that are most relevant for the discrimination of the manipulation.

The comparison of the proposed method with state-of-the-art algorithms proposed for OSR in general deep learning literature is reported in Table 7.4 for the configurations F0, F3 and F4. We see that the proposed method achieves the best results in all the cases on both closed and open sets scenarios. In particular, our ViT-based hybrid algorithm gets an AU-ROC of 85.34%, 91.98% and 89.75% in Config F0, F3 and F4, respectively, with an improvement with respect to the best-performing method from the state-of-the-art always larger than 4% in terms of both accuracy and AU-ROC. It is worth observing that all these methods have been proposed to address general problems of OSR in deep learning and adopted for common image classification tasks and object recognition, e.g. MNIST or CIFAR classification. Hence, they are not designed for forensic problems and, in particular, manipulation classification tasks, where the classification often relies on the analysis of subtle traces, and the goal in the open set scenario is to be able to reveal

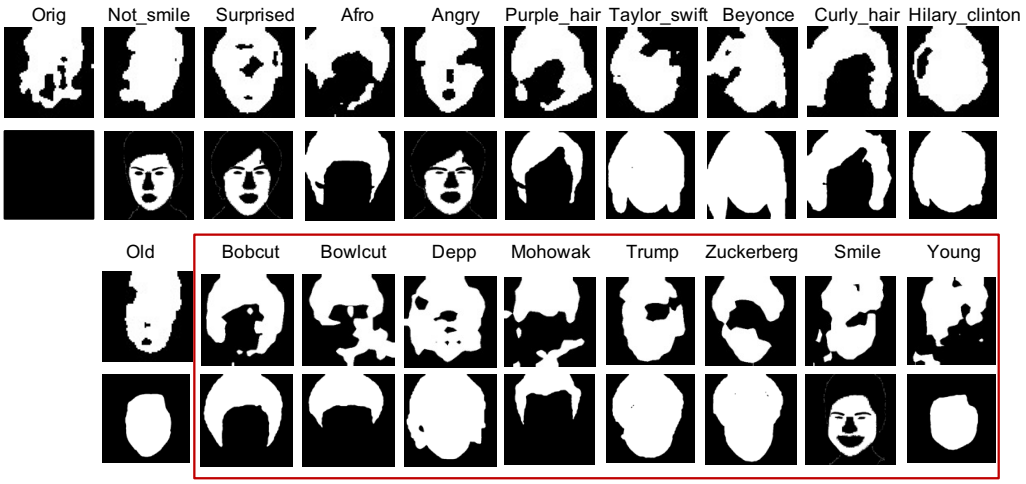


Figure 7.4 – Example of localization masks for the 18 editing types. Predicted (top) and ground truth (bottom) masks are visualized. The masks in the red box refer to the out-of-set editing types.

Table 7.4 – Comparison with state-of-the-art methods. Results are reported for the Config F0, F3 and F4 configurations.

Methods	F0		F3		F4	
	Closed-set Accuracy (%)	Open-set AU-ROC (%)	Closed-set Accuracy (%)	Open-set AU-ROC (%)	Closed-set Accuracy (%)	Open-set AU-ROC (%)
GCPL [116]	73.72	73.25	40.93	69.46	43.16	65.48
RPL [163]	74.43	76.21	70.19	81.46	65.76	71.18
ARPL [173]	82.64	81.73	87.80	84.93	90.7	79.89
CAC [164]	77.86	74.95	83.33	78.57	85.09	77.63
PCSSR [168]	84.10	74.49	90.79	85.42	92.25	83.63
RCSSR [168]	83.70	72.95	90.60	86.87	91.67	85.32
Prop.	**88.99**	**85.34**	**94.34**	**91.98**	**95.25**	**89.75**

unseen alterations of similar content or the presence of different fingerprints.

Ablation Study

We conducted an ablation study to investigate the effects of the patch size P used in the ViT module and to validate the effectiveness of each component of the proposed architecture.

Impact of Different Patch Sizes. Figure 7.5 shows the results obtained by using different patch sizes P, namely $P = 1$, 2, 4 and 8 (the legends report the P setting in brackets). We see that increasing the patch size, up to $P = 4$, is beneficial for both closed-set and open-set performance. However, when the patch size increases further,

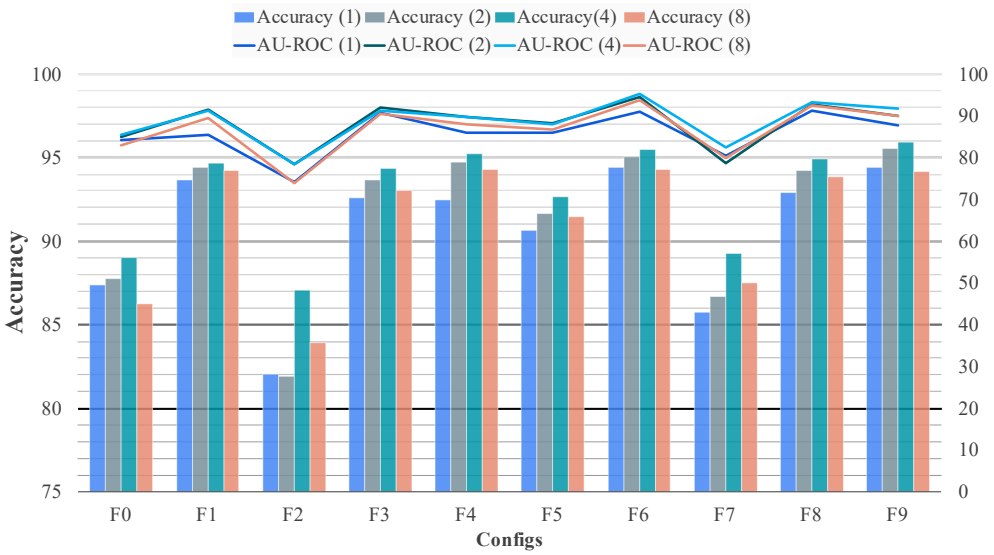


Figure 7.5 – Ablation study on the impact of patch size P of ViT under various configurations. Vertical bars show closed-set Accuracy, while the line plots show the AU-ROC for open-set.

namely, above 4, results do not improve, and actually, a performance drop is observed (around 1.6% in Accuracy and 2% in AU-ROC on average). Then, from our experiments, with $P = 4$ the ViT achieves the best trade-off between the exploitation of the spatial and the feature maps correlation.

Impact of Different Architectures. Figure 7.6 compares the results achieved by the proposed architecture, including the ViT module for the classification and the localization branch (ResNet50+Vit+FCN), with those achieved by the same method by removing the FCN (ResNet50+Vit), and those of the baseline ResNet50, where the standard ResNet50 is used for the multi-class classification. In this case, the rejection is performed in a similar way by analyzing the output layer of the last FC of ResNet50 before the softmax (MLS). A significant performance gain is obtained by the proposed method in all the configurations. In particular, by combining the use of ViT for processing the feature maps with the hybrid approach, we got a gain of up to 10% in Accuracy and 9% in AU-ROC.

7.3.2 Synthetic image attribution

In this section, we report the results we got for open-set image attribution using SIAD dataset. Given that we focused on fully-synthesized images, we excluded the localization branch from the architecture. Therefore, these experiments only validate the effectiveness of the ViT-based architecture and the sample rejection strategy. Experiments were carried out by considering 4 different splittings of the 5 architectures. Table 7.5 illustrates the

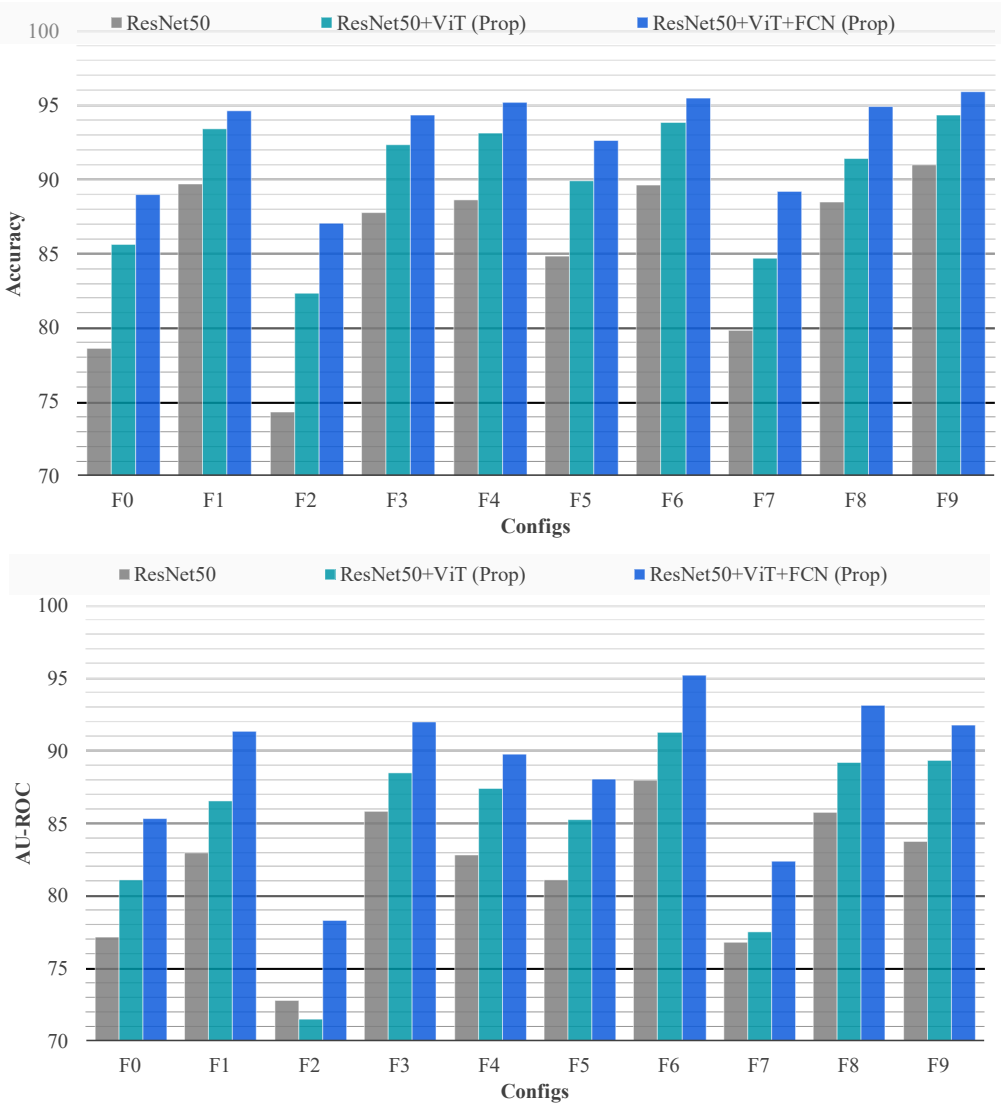


Figure 7.6 – Performance in closed-set (left) and open-set (right) for different configurations (F0-F9).

in-set and out-of-set architectures for each configuration.

Table 7.6 shows the closed-set and open-set performance achieved by the proposed architecture (ResNet50+ViT) in all the configurations. The results of the baseline are also reported (ResNet50). We see that the advantage we got with respect to the baseline is even bigger in this case than for face editing classification. In particular, when the rejection strategies are mounted on top of the baseline architecture, that is, by considering the features extracted by a standard ResNet50 classifier for the analysis, the rejection

Table 7.5 – Dataset configurations for attribution task.

	Config-S1	Config-S2	Config-S3	Config-S4
In-set	LSGM, StyleGAN2 Taming transformer	StyleGAN2, StyleGAN3 Latent diffusion	LSGM, StyleGAN2 StyleGAN3	LSGM, StyleGAN2 Latent diffusion
Out-of-set	StyleGAN3 Latent diffusion	LSGM Taming transformer	Taming transformer Latent diffusion	StyleGAN3 Taming transformer

Table 7.6 – Results on synthetic image attribution task.

Configs	Method	Closed-set (Accuracy %)	Open-set (AU-ROC %)		
			MSP	OpenMax	MLS
S1	ResNet50	97.76	77.23	64.60	76.32
	ResNet50+ViT (prop)	**99.86**	**92.73**	**92.70**	**92.72**
S2	ResNet50	78.26	39.90	33.40	43.40
	ResNet50+ViT (prop)	**82.38**	**72.49**	**65.02**	**70.39**
S3	ResNet50	92.80	78.78	57.83	69.82
	ResNet50+ViT (prop)	**98.56**	**82.74**	**78.13**	**83.31**
S4	ResNet50	81.82	67.43	61.66	69.98
	ResNet50+ViT (prop)	**94.61**	**90.31**	**93.56**	**93.60**

performance is very poor, with an AU-ROC lower than 70% in most cases. Our method instead can achieve a much higher AU-ROC going above 90% for Config-S1 and Config-S4. Under the Config-S1 and Config-S2 configurations, the results are worse. We observe that these configurations include both StyleGAN2 and 3 in the training set, hence resulting in a lower diversity of the in-set dataset, which might be the reason for the worse capability to handle the open-set scenarios. Finally, we observe that, as before, the MLS is the strategy that gives the best average performance, even if the 3 rejection strategies work very similarly. These results confirm that the features extracted with our architecture are representative and allow a good characterization of the various architectures, yielding good discrimination also in the open-set scenario.

7.4 Summary

In this chapter, we made a first step to address the issue of open-set classification of AI-manipulated images in the wild. Specifically, we have presented a ViT-based multi-class classifier with a rejection option that suppresses predictions for unknown out-of-set manipulations. The architecture enhances the correlation among patches via a ViT module using a self-attention mechanism and avoids the need for multiple training of models in Chapter 5. To reject unknown out-of-set samples, we have considered various methods, including analyzing the output logit scores and the probabilities and estimating the outlier probability. We validated the effectiveness of our classifier with rejection on

facial attribute editing classification and synthetic image attribution by comparing it with state-of-the-art methods. The best open-set results were achieved by using the logit score.

While the method described in this section, achieves superior open-set performance compared to state-of-the-art approaches, there is still room for improvement, particularly in the attribution task, where the open-set performance varies across different configurations. Additionally, the classifier's response to both in-set and out-of-set samples in open-set scenarios remains an open question. For instance, a verification framework could also be suitable for the attribution task. In the upcoming chapters, we consider these issues by verifying the benefits of the adoption of a verification framework for the attribution task (Chapter 8), and by introducing a novel open-set rejection framework based on the concept of backdoor attacks (Chapter 9).

Chapter 8

A Siamese-based Verification System for Attribution of AI-generated Images

"Whenever we proceed from the known into the unknown, we may hope to understand, but we may have to learn at the same time a new meaning of the word 'understanding."
Werner Karl Heisenberg

Addressing the synthetic image attribution in open-set setting is very challenging. In particular, discriminating between in-set and out-of-set (never seen) samples by relying on in-set features, as done with the classification with rejection approach adopted in the previous section, is often hard. In this chapter, we adopt a different approach to open-set image attribution, treating such a task as a verification task. In particular, given two generated (fully synthetic) images, we ask the system to decide whether they have been produced by the same generative architecture or not. We also consider a slightly different setting, where the system is asked to verify a claim about the architecture used to generate a given generated image by relying on multiple reference images produced by the claimed architecture. The verification approach has a significant advantage with respect to classification with rejection class, which is not able to provide any information about out-of-set architectures, other than recognizing that they do not belong to the set used for training.

The system we have developed is based on a Siamese Network architecture with an EfficientNet-B4 backbone, trained in two phases: in the first phase we focus on the feature extraction part, while in the second one, we train the final decision layers.

This chapter is organized as follows: in Section 8.1, we describe the verification framework and its architecture. In Section 8.2, we describe the dataset and the methodology, including the training procedure and the verification protocol. The results of the experiments we carried out to validate the effectiveness of the verification system are discussed in Section 8.3.

8.1 Proposed Verification System

The proposed verification system for synthetic image attribution is illustrated in Figure 8.1. The following verification scenarios are considered:

- Given two input images x and q, verify whether they are produced by the same architecture or not (input pair verification).

- Given an input image x and a claim on the generating architecture, verify whether x has been produced by the claimed architecture or not (claimed-based verification).

In the first scenario, the system is fed with the input pair (x, q). The true label y associated with the input pair is equal to 0 if x and q have been generated by the same architecture, 1 otherwise. By indicating with $\hat{y}$ the output of the system, and with $p(x, q)$ the probability score, we have $\hat{y} = 0$ if $p(x, q) < 0.5$, $\hat{y} = 1$ otherwise. In the second scenario, the verification works by considering one or multiple reference images q_j generated by the claimed architecture (of type j) and feeding the system with the resulting pairs. In the multiple-reference case, given a set of references D_r, all the pairs (x, q_{ji}), $q_{ji} \in D_r$, are tested, and the final decision (Yes/No) is taken by fusing the outputs according to a fusion strategy. In our experiments, we considered both majority voting and score-level fusion. The latter gave the best performance. In particular, the best results were achieved by considering the minimum probability score. The proposed verification framework naturally works in an open-set scenario, where one of the two inputs or both inputs come from an architecture that has not been used for training (with reference to the second verification scenario, either the input x, or the claim, or both, may come from an unknown architecture).

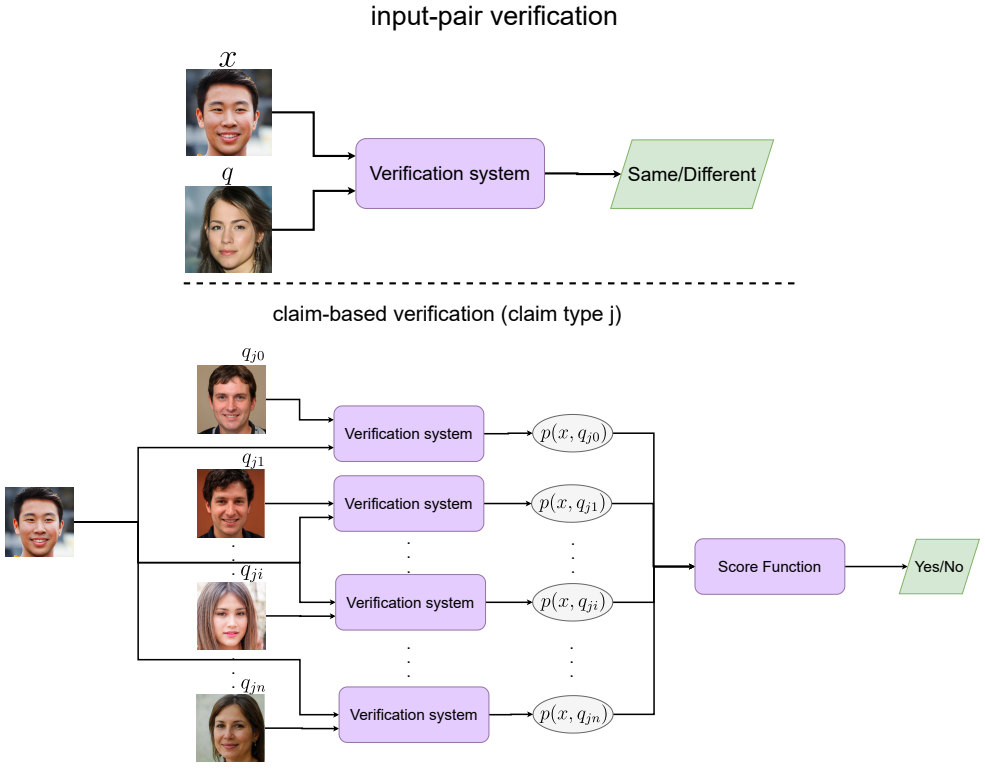


Figure 8.1 – Verification scenarios considered in this chapter.

The system we are considering to address the tasks described in Figure 8.1 relies on an SN architecture (see Figure 8.2). It consists of two parts: the feature extraction part and

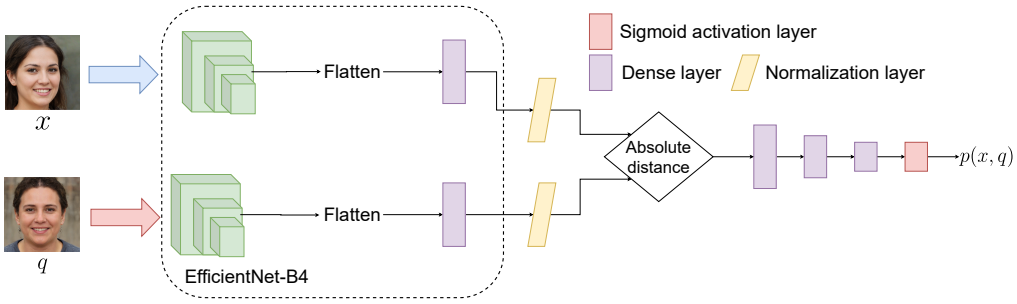


Figure 8.2 – High-Level Architecture for the verification task.

the decision-making part. Feature extraction is performed by an SN using EfficientNet-B4 as a backbone for both branches. The input image size for each branch is 380×380. The output of each branch is flattened and then fed as input to a dense layer with size input neurons. The feature embedding, then, consists of 512 elements. The features are input to a normalization layer, and then the point-wise absolute distance between the two output vectors is computed. The distance vector enters the decision-making network, consisting of three consecutive dense layers of sizes 256, 64, and 1, respectively. The final probability scores are obtained by inputting the output of the last dense layer into a sigmoid activation layer. In our experiments, we also tried other backbone networks to implement the Siamese branches, namely ResNet and SWIN transformers [42]. While we got perfect results with all these networks in the closed-set setting, the EfficientNet backbone is the one giving the best result in the open-set setting.

8.2 Methodology

In this chapter, we used an extended version of SIAD, namely SIAD_v2, consisting 10 generative architectures (5 more than SIAD, see in Section 2.4.4 for the details). Starting from this pool of architectures, three different splittings of in-set and out-of-set architectures were considered, with 5 in-set and 5 out-of-set architectures each, named Config-S1, Config-S2, and Config-S3. The details of the splittings are reported in Table 8.1. We observe that in the first and second configurations, a mixture of GANs, diffusion architectures, and transformers were considered as in-set, while in the third configuration, only GANs are included as in-set architectures. The in-set architectures are used to train the Siamese verification network, while the out-of-set architectures are only considered for testing. For each in-set architecture, we considered 48,000 images, split into training, validation, and testing sets according to the proportion 45000:2500:500. For each out-of-set architecture, 500 images were considered for testing. Figure 2.8 shows an example of generated images for every architecture. To produce the images, we used the pre-trained models released by the authors in the online repositories.

Table 8.1 – Dataset splitting information. Architectures split (in-set and out-of-set) considered in our experiments.

	Domain	S1	S2	S3
In-set	FFHQ	Latent diffusion, Taming transformers, StyleGAN2-f	StyleGAN2-f, Latent diffusion	StyleGAN2-f, StyleGAN3
	CelebA	Latent diffusion, Taming transformers, DDPM, BEGAN	BigGAN, ProGAN, Latent diffusion, LSGM	ProGAN, BEGAN, BigGAN
Out-of-set	FFHQ	StyleGAN3, StarGAN2	StyleGAN3, Taming transformers	Latent diffusion, Taming transformers
	CelebA	LSGM, ProGAN, BigGAN	Taming transformers, BEGAN, DDPM, StarGAN2	Latent diffusion, Taming transformers, LSGM, DDPM, StarGAN2

8.2.1 Siamese network training

We trained three different SN-based verification models, one for each configuration of in-set and out-of-set architectures, namely Config-S1, Config-S2, and Config-S3. The models were trained on a dataset of paired inputs, corresponding to images produced by the same or different architectures, hereafter referred to as positive and negative pairs. For every configuration, the dataset is built from the in-set training dataset as follows: each image is coupled with another image from the same architecture to build a positive pair, and another image is selected randomly from a different architecture to build a negative pair. In this way, the SN is trained on a balanced dataset. Specifically, the training dataset is made up of 45,000 $\times$ 5 (no. of images per arch $\times$ no. of the in-set arch) negative pairs and the same number of positive pairs for a total of 450,000 pairs.

In all cases, training was carried out in two distinct phases: the feature extraction phase and the decision phase. In the first phase, the two SN branches are trained for 100 epochs, starting from an EfficientNet-B4 model pre-trained on ImageNet, with Adam optimizer and learning rate equal to 0.0001, using the early stopping condition. The network is trained by using a contrastive loss [190], defined as

$$\mathcal{L} = (1 - y) \cdot d_E^2 + y \cdot [\max(0, d_m - d_E)]^2, \tag{8.1}$$

where d_E is the Euclidean distance between the output of the branches of the SN (embeddings), and d_m is a margin hyperparameter that enforces a minimum distance between the two embeddings. We set d_m to 1 in the experiments. The contrastive loss enforces the embeddings of the images in the latent space to be far away whenever the images come from different architectures and close to each other when they belong to the same architecture. Augmentation is performed during training. In particular, we considered JPEG compression, random color transformations (brightness, contrast, saturation, and hue),

and random flip. The JPEG compression factors are randomly selected within the range [70,100]. Additionally, for saturation, a random factor between 0.5 and 1 is employed, while the hue is adjusted using a random factor between -0.2 and 0.2. Similarly, brightness undergoes modification with a random factor between -0.2 and 0.2, and contrast is adjusted with a random factor between 0.2 and 0.5. Each type of augmentation is carried out with an independent probability of 0.3. Therefore, an image can be subject to one or more augmentations from the list, being also possible that all augmentations be applied simultaneously. Once the embeddings have been obtained, in the second phase, the weights of the feature extraction network are frozen, and the three dense layers following the normalization and the absolute distance layer, which is responsible for the decision, are trained. A binary cross-entropy loss is used to train these layers (decision-making network). The layers are trained for 20 epochs with Adam optimizer and a learning rate equal to 0.0001, with early stopping condition.

8.2.2 Testing procedure

We evaluated our system by considering two testing scenarios: one-vs-one and one-vs-many. In the one-vs-one case, each input image in the test set is paired with images generated by the 10 architectures (5 in-set, 5 out-of-set), chosen at random from the test set, thus getting a total of 5000×10 (10% positive pairs and 90% negative pairs). Then the SN-based model is evaluated on those pairs. The one-vs-one tests measure the performance of the system in the input-pair verification scenario depicted in Figure 8.1, and in the claim-based verification scenario, when only one (random) reference is used to verify the claim.

The one-vs-many test setting measures the performance of the system in the claim-based verification scenario when multiple references are available. In our experiments, we considered 100 reference images. Given a test input image and a claim on the architecture (10 possible claims are considered corresponding to all the architectures) - say Type j, we paired each input image with $D_r = 100$ reference images from the claimed architecture. The reference images are randomly selected from the test set. The final decision is taken by considering either the mean or the minimum probability score (the latter resulting in the best results). Formally, we consider, respectively, the statistic $(1/|D_r|)\sum_{i\in D_r} p(x, q_{ji})$, and $\min_{i\in D_r} p(x, q_{ji})$.

8.2.3 Comparison with classification approaches

Given that the verification framework proposed in this chapter is a novel one, no baseline and state-of-the-art methods could be considered for comparison. In order to show the good capabilities of our system to learn good embeddings for the attribution task, we exploited the SN-based verification model inside a classification framework and ran a comparison with existing methods for the classification of synthetic attribution in an open-set scenario. In particular, for every configuration of in-set/out-of-set architectures in Table 8.1, we built a classifier with rejection as follows:

- We chose one representative image for every in-set architecture. Specifically, we

considered the cluster centroid of the validation sub-dataset (corresponding to the architecture);

- The input image is paired with the 5 representative images obtained at step 1, and the SN-based architecture is tested with these pairs;
- The pair associated with the minimum score is chosen as the output fo the classifier.

Formally, let z_j, $j = 1, .., 5$ denote the 5 centroid images. Given a query image x, the final classification score associated with x is $\min_{j=1,\cdots 5} p(x, z_j)$ and the decision is made for the closed-set architecture i^* that achieves the minimum. Rejection is performed by exploiting the MSP approach [191]. According to MSP, a low confidence in the predicted class reflects the uncertainty of the network prediction, providing evidence that the input sample belongs to an out-of-set class. Then, given a threshold ν, the output of the classifier j^* is accepted if $p(x, z_{j^*}) < \nu$ (lower scores correspond to higher confidences for the 'Same'/'Yes' class in our case), rejected otherwise.

8.3 Experimental Results

In this section, we report the performance of the proposed system in the closed and open-set cases and the results of the generalization tests, when unknown models are considered for the same in-set architectures. Finally, we report the comparison results, obtained by considering the classification with the rejection system described in Section 8.2.3.

8.3.1 Verification results

The results in the one-vs-one setting are reported and discussed below. In Table 8.2, we report the Accuracy of the verification task in the closed-set scenario, when x and q are produced by in-set architectures, for the 3 configurations. These results show that in the closed-set scenario perfect verification (ACC = 1) can always be achieved by our system. The verification performance in the closed and open-set settings are reported in Table 8.3 for each architecture, that is for q belonging to each of the 10 architectures. The average AU-ROC and the probability of correct detection for a Pd@5% score on test set, are reported for each architecture. The average is computed for the negative pairs over both in-set and out-of-set architectures (9 architectures in total). We observe that the results corresponding to in-set architectures refer to a mixture of closed and open-set scenarios, given that q may either belong to an in-set or out-of-set architecture (with probability 50%). In other words, at least one input of the pair comes from the in-set architectures in this case. Instead, the results for out-of-set architectures always refer to the open set scenario, where at least one input of the pair, or both, (with probability 50%), are generated by out-of-set architectures. By looking at this table, we see that when at least one of the two inputs comes from a known architecture, the verification is perfect or almost perfect. In particular, focusing on Config-S1, we see that the AU-ROC is 1 in 4 out of 5 cases (in which the Pd@5% is also perfect) and 0.94 in the other case. Similar results are observed in the other configurations. The verification performance decrease,

still remaining pretty good, in cases where at least one or both inputs come from unknown architectures (out-of-set architectures). Overall, similar behavior and results are obtained in the three configurations.

Table 8.2 – Closed-set verification results (Accuracy %).

	Config-S1	**Config-S2**	**Config-S3**
Accuracy	100	100	100

Table 8.3 – Verification results (AU-ROC (%) and Pd@5% (%)) in closed and open-set. The cells with green backgrounds indicate in-set architectures, while the white backgrounds indicate out-of-set architectures.

Generating Architecture	**Config-S1**		**Config-S2**		**Config-S3**	
	AU-ROC	**Pd@5%**	**AU-ROC**	**Pd@5%**	**AU-ROC**	**Pd@5%**
Latent Diffusion	100	100	100	100	91	76
DDPM	94	74	85	68	91	80
Taming transformers	100	100	88	72	84	66
StyleGAN2	100	100	100	100	100	100
BEGAN	100	99	95	79	100	100
StyleGAN3	90	81	90	81	97	92
LSGM	84	68	100	98	70	34
StarGAN v2	88	72	84	68	89	80
BIGGAN	95	79	90	74	92	79
PROGAN	85	68	100	100	100	100

In Table 8.4, we report the average results for all configurations. The total AU-ROC is averaged over all the possible pairs of inputs, hence considering all the pairs' combinations (in-set vs in-set, in-set vs out-of-set, out-of-set vs in-set, and out-of-set vs out-of-set). The open-set AU-ROC instead is computed by considering only the out-of-set vs out-of-set pairs (fully open set), while the closed-set AU-ROC is computed over the in-set vs in-set pairs. The results show that Config-S1 shows better results in the open-set scenario. We observe that in this configuration, the out-of-set set contains (mostly) GAN architectures and a diffusion-type architecture (LSGM), that are also present in the in-set. This is not the case in the other configurations where, for instance, transformers in Config-S2 and both diffusion models and transformers in Config-S3 are only considered as out-of-set, without any of these types of architectures included in the in-set set.

In Table 8.5, we report the average results of the tests one-vs-many, for all the configurations. In all the cases, a slight improvement is observed when the minimum score is considered, compared to the case of one reference only, while the mean score case only improves in a few cases. These results show that using multiple random references for the verification improves the results only slightly. A possible reason is that all the feature vectors for a given architecture tend to cluster close to each other, yielding a similar verification result.

Table 8.4 – Total, open-set and closed-set AU-ROC (%).

	Config-S1	**Config-S2**	**Config-S3**
Total AU-ROC	95	93	93
Open-set AU-ROC	92	81	85
Closed-set AU-ROC	100	100	100

Table 8.5 – Total, open-set, and closed-set AU-ROC (%) in the one-vs-many setting.

	Config-S1		**Config-S2**		**Config-S3**	
	Mean	**Min**	**Mean**	**Min**	**Mean**	**Min**
Total AU-ROC	95	96	94	94	94	94
Open-set AU-ROC	92	94	78	84	87	85
Closed-set AU-ROC	100	100	100	100	100	100

8.3.2 Generalization tests

We also conducted some generalization tests to validate the robustness of the proposed system when faced with "unknown" models within various in-set architectures. This provides evidence that the system works as expected, attributing images to specific architectures rather than individual models. Specifically, we examined models from the in-set architectures that were trained: i) on a different dataset of pristine images; ii) using a different training methodology; and iii) using different training configurations.

For case i), we examined a system trained under Config-S1, focusing on the Taming Transformers architecture. Throughout the training process of the Taming Transformers, we used images exclusively from pre-trained models trained on FFHQ. During the testing phase, we intentionally formed pairs generated by Taming Transformers pre-trained models trained on either FFHQ and CelebA. Notably, the system demonstrated a remarkable performance by accurately classifying these images, and correctly identifying them as being generated by the same architecture. Additionally, when paired with images from different architectures, the system classified them as distinct.

In case ii), we tested a system trained across all configurations on StyleGAN2-ada [192], which employs an adaptive discriminator augmentation mechanism for training stability in limited data regimes (while our system was originally trained only with StyleGAN2-f). Lastly, for case iii), we evaluated a system trained in Config-S3 using a StyleGAN3 model obtained through retraining on unaligned high-resolution faces (FFHQ-U).

In all generalization tests, positive pairs were formed by pairing images from the unknown models with random images from the same architecture (known models), while negative pairs were created by pairing images from unknown models with random images from different in-set architectures. The results presented in Table 8.6 demonstrate that the system consistently generalizes well in all scenarios, always achieving an AU-ROC and Accuracy of 100.

Table 8.6 – Results with models trained with different datasets, parameters, and training procedure (in the models' names the number refers to the image resolution)

Architecture	Mismatch	Train model(s)	Test model(s)	AU-ROC (%)	Accuracy (%)
Taming Transf Config-S1	Dataset	FFHQ	CelebA	100	100
StyleGAN2 Config-S1	Training Methodology	StyleGAN2-f	StyleGAN2-ada	100	100
StyleGAN2 Config-S2	Training Methodology	StyleGAN2-f	StyleGAN2-ada	100	100
StyleGAN3 Config-S3	Configurations	StyleGAN3-(t-1024/t-u256/r)	StyleGAN3-t-u1024	100	100
StyleGAN2 Config-S3	Training Methodology	StyleGAN2-f	StyleGAN2-ada	100	100

8.3.3 Comparison results

In this section, we report the results of the experiments that we run considering a classifier built by starting from the proposed verifier, as detailed in Section 8.2.3. This system is compared with the PCSSR and RCSSR method and the ResViT method described in Chapter 7 for the classification of synthetic manipulation and attribution in open-set settings. All these methods perform classification with a rejection option. Table 8.7 reports the closed-set performance (Accuracy) and the AU-ROC measuring the rejection performance. The results show that the proposed classifier is the one obtaining the best average performance in all three configurations of in-set and out-of-set architectures, with a perfect Acc and an AU-ROC gain, which is about 8% on the average over PCSSR and RCSSR, and 11% over ResVit in Chapter 7. These results show the superior capability of our method to produce characteristic embeddings for the various architectures. Once again, we stress that we considered this framework only for comparison purposes. Indeed, the capabilities of the verification system that we proposed in this chapter in the open-set scenario are not limited to sample rejection, given that our system can provide the same functionality in both closed and open-set scenarios.

Table 8.7 – Comparison of closed-set (Accuracy %) and open-set (AU-ROC %) performance of the classifier based on our SN-based model with state-of-the-art classifiers.

		ResViT (Chap. 7)	**PCSSR** [168]	**RCSSR** [168]	**Prop.**
Config-S1	**Accuracy**	99	99	99	**100**
	AU-ROC	79	**84**	83	82
Config-S2	**Accuracy**	99	99	99	**100**
	AU-ROC	76	74	75	**82**
Config-S3	**Accuracy**	99	99	99	**100**
	AU-ROC	68	66	64	**83**

8.4 Summary

In this chapter, we have described a novel verification framework to address the problem of synthetic architecture attribution in open set conditions. We have considered two verification cases according to whether the generative architecture of the analyzed image is claimed as known or not. Unlike the adoption of classification with rejection option, verification offers the advantage of determining whether two given images have been produced by the same architecture, regardless of whether the generating architecture was during the training. The experiments we ran demonstrate the good performance of our system in both closed and open-set settings, when different mixtures of generative architectures are considered as in-set and out-of-set.

We also demonstrated the potential use of the SN-based verification model as a classifier with a rejection class. This indicates that verification could be a good option to solve the open-set problem. In this case, a novel class can be added to the classifier by comparing the similarity between the query image and the claimed images when a few out-of-set samples are provided and claimed. Of course, this requires the model to see more data from different domains (face and non-faces), which is not considered in this thesis and should be definitely addressed in the future.

Chapter 9

BOSC: A Backdoor-based Framework for Open Set Synthetic Image Attribution

"Never attack an idea you don't understand, ego hates being ignorant; so it often attacks new ideas."
Aniekee Tochukwu Ezekiel

In this chapter, we introduce a novel framework, named Backdoor-based Open-set Classification (BOSC), for multi-class classification with a rejection option, based on the concept of backdoor attacks. BOSC works by purposely injecting class-specific triggers inside a portion of the images of the training set to induce the network to establish a match between class features and trigger features. The behavior of the model with respect to samples with triggers is exploited at test time to perform sample rejection, leveraging on the fact that a proper match can be found for samples of in-set classes, while it can not be found for samples coming from classes not included in the training set. The idea behind BOSC is a general one and, in principle, can be applied to any task, although different tasks may require the adoption of different triggers in order for the method to work well.

In this thesis, we apply the BOSC framework to open-set synthetic image attribution. To validate the effectiveness and generality of the method, we also run some experiments on the classification of synthetic facial attributes. Experiments confirm the good performance of BOSC in various settings, including robustness against image post-processing. A possible reason for such robustness is the following: since the trigger is applied as input pre-processing before feeding the samples to the network, only the class features are weakened by the processing operation while the trigger features are not affected, and the matching between the trigger and the class is preserved.

This chapter is organized as follows. We begin with an introduction to backdoor attacks in Section 9.1, which is necessary to understand the rest of the chapter. The BOSC framework is described in Section 9.2. Section 9.3 reports the experimental methodology and setting. The results of the experiments on synthetic image attribution and the comparisons with the state-of-the-art are discussed in Section 9.4. Finally, Section 9.5 reports the experiments that we run on the classification of facial editing. Section 9.6 draws some final conclusions.

9.1 Backdoor Attacks in a Nutshell

Backdoor attacks against deep neural networks are a particular type of training time attacks, where a hidden behavior (backdoor) is inserted in the model during the training

phase to gain unauthorized access to a system at operating time.

A backdoor attack works by injecting a pattern, known as a trigger, into (a subset of) the images used during training, inducing the trained model to exhibit a malicious behaviour, e.g. predicting a wrong predefined class, when fed with images containing the trigger, while continuing to operate as expected on normal inputs. Early methods relied on triggers that can be easily recognized by visual inspection [193], prompting subsequent research to focus on enhancing the stealthiness of backdoor attacks. Recent advancements in this field can be categorized into two main approaches: those employing invisible triggers and those relying on natural triggers. Some works have focused on generating imperceptible perturbations as backdoor triggers [194–197], aiming to minimize the difference between the original images and the images containing the trigger. Other approaches involve changing the style of images as triggers, aiming to make them appear natural and less suspicious [198–201]. Techniques such as leveraging natural reflection phenomena [198], applying Instagram filters [199], utilizing generative adversarial networks [200], and warping-based image transformations [201] have also been proposed. To combat this threat, defense techniques have also been developed [202–204]. Several threat models can be defined depending on the knowledge available to the attacker and the defender and his/her capability [29], which can be used to guide the development of effective defense techniques.

In our work, the concept of backdoor attack is exploited for a completely different purpose, and the backdoor behaviour is injected by the model owner during training and exploited at operation time by the forensic analyst for open-set classification. Therefore, the visibility of the trigger is not an issue and strong patterns can be used to facilitate the activation of the desired backdoor behaviour.

9.2 BOSC System

9.2.1 Backdoor-based classification with rejection class

The idea behind the system we propose is described in the following. A specific trigger image is associated with every in-set class. Then, a subset of the training images of each class is tainted by *injecting* the trigger image of the class into the training images. The tainted images are labeled as belonging to an additional backdoor class, whose label is equal to $C+1$. When trained on the tainted dataset, the network learns to recognize the presence of the triggers and to associate the simultaneous presence of class and trigger features with the backdoor class. Note that we require that the backdoor class is activated only when the trigger matches the class it is associated with. If the trigger associated with i is injected into an image belonging to class j ($i \neq j$), the network should correctly classify the image as belonging to class j. For the rest, the network is expected to work normally on images without the trigger. We argue that for out-of-set images, none of the triggers matches the image features, and hence, the backdoor class is never activated, thus allowing the system to distinguish in-set and out-of-set images.

With these ideas in mind, the pipeline of BOSC is shown in Figure 9.1. In the injection stage (Figure 9.1(a)), a portion of the samples in the training set is tainted by

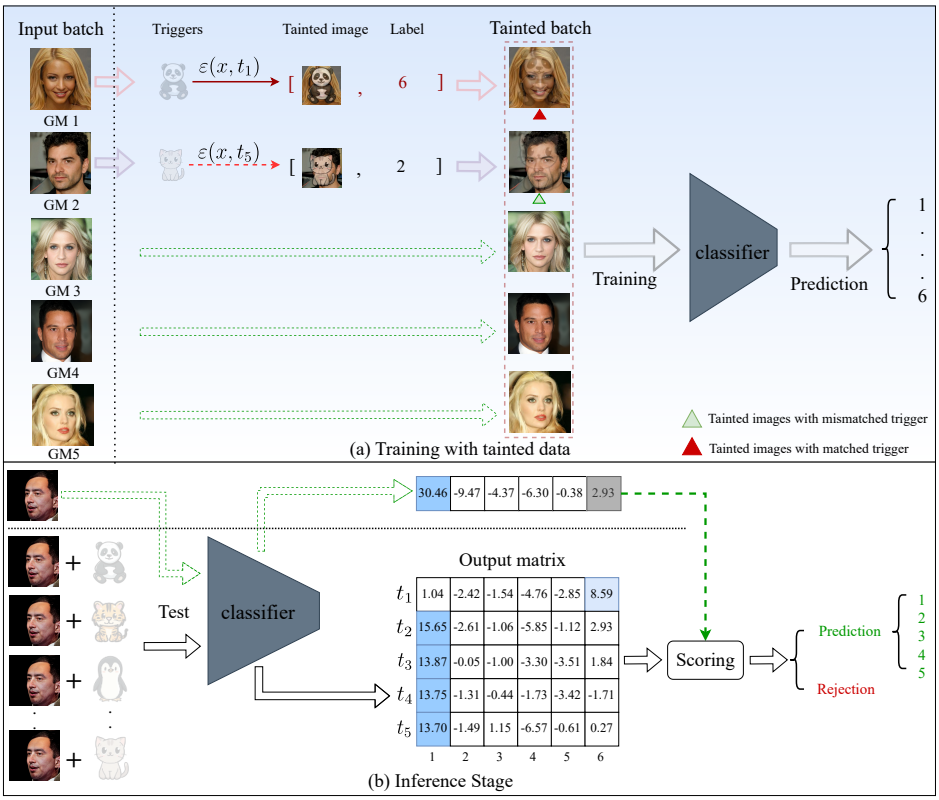


Figure 9.1 – BOSC pipeline. (a) Training with tainted data: a subset of the samples are tainted with the triggers (trigger injection phase). When the trigger matches the sample class, the label is modified to $C + 1$ (red triangle, $C = 5$ results in target class 6); otherwise, keep the label unchanged (green triangle). The network is then trained with tainted data and clean data (training phase). (b) Inference Stage: the test input is analyzed by the trained model by superimposing to it all the C triggers (e.g., $C = 5$). The output is a matrix with all the predictions (the figure refers to the case of 5 closed-set classes). The open set score is computed from this output matrix. When the score exceeds a predefined threshold, the prediction is made by relying on the prediction made on the clean test image. Otherwise, the sample is rejected. Cartoon images are used as trigger images. In the figure, trigger t_i is matched with the $i-$th generative model (GM) class.

superimposing class-specific triggers to the training image. Cartoon images are utilized as trigger images[1]. When the trigger superimposed to the image matches the image class, the label of the image is modified to $C + 1$, while it is left unchanged otherwise. The presence of images tainted with mismatched triggers ensures that the behaviour of the

[1]Given the way the backdoor is exploited in our work, the visibility of the trigger is not an issue and the triggered images might show visible trigger patterns.

network is not modified when the triggers are superimposed to images from different classes, thus ensuring a unique association.

Formally, let t_k denote the trigger associated with class k. We indicate with $T = \{t_1, ..., t_C\}$ the trigger set. Given a sample x and a trigger t_k, a tainted sample x_{t_k} is obtained as:

$$x_{t_k} = \mathcal{E}(x, t_k) = (1 - \alpha) \cdot x + \alpha \cdot t_k, \tag{9.1}$$

where α is a parameter controlling the injection strength. When x belongs to class k, the label of x_{t_k} is changed from k to $C + 1$ (backdoor class); otherwise, it is left as is. After the dataset has been tainted, a multi-class network with $C + 1$ output nodes is trained as usual on the tainted dataset. We let $\phi(\cdot)$ denote the network function of the backdoored model. A softmax layer is applied at the end. Hence $\phi(x)$ is a probability score, $\phi(x) \in [0, 1]^{C+1}$ and $\sum_{i=1}^{C+1} \phi_i(x) = 1$. We indicate with $\phi_i(x)$ the i-th element of the output.

Given the way the training data have been built and labeled, and the way the model has been trained, the network is expected to work as follows for in-set samples x:

$$\begin{cases} \arg\max_i \phi_i(x) = y \\ \arg\max_i \phi_i(x_{t_k}) = y, \text{if } k \neq y \\ \arg\max_i \phi_i(x_{t_k}) = C + 1, \text{if } k = y. \end{cases} \tag{9.2}$$

In the inference phase, BOSC works as illustrated in Figure 9.1(b). Given a sample x, a tentative prediction y^* is first made by considering the network output in correspondence of x. The prediction is obtained by excluding the trigger class output, that is, by letting

$$y^* = \arg\max_{i \in \mathcal{C}} \phi_i(x). \tag{9.3}$$

The C triggers in T are then superimposed to the image under analysis, and the resulting C tainted samples are fed to the network, obtaining C output vectors with the logit values corresponding to all the $C + 1$ output classes of the network. Let $m_i \in \mathbb{R}^{1 \times (C+1)}$ denote the output logit vector corresponding to the image tainted with trigger t_i. We denote with $M \in \mathbb{R}^{C \times (C+1)}$ the output matrix, where each row corresponds to an output logit vector. Rejection is performed by using the matrix M to compute a rejection score ξ_r and comparing ξ_r against a threshold (see Section 9.2.2 for a precise definition of ξ_r). The tentative prediction y^* is accepted if the rejection score is above the threshold, otherwise a rejection decision is made. Formally, the final output $\hat{y}$ of the BOSC classifier is obtained as follows:

$$\begin{aligned} &\hat{y} = y^* \quad \text{if } \xi_r(M) > \nu, \\ &\hat{y} = \Re \quad \text{otherwise}, \end{aligned} \tag{9.4}$$

where ν is a suitable threshold.

9.2.2 Trigger-based score for rejection

In this section, we describe some possible rejection scores that can be used for out-of-set class rejection. As we said, for in-set samples, we expect that the true class receives

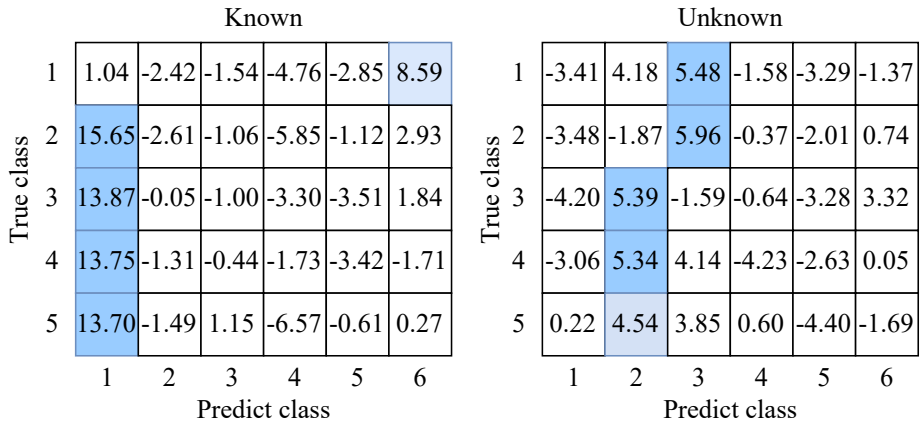


Figure 9.2 – Example of output matrix - Config S1 (see Table 8.1 for the details of the setting). Left: sample from class 1. Right: sample from unknown class. '6' corresponds to the trigger class.

a high prediction score when the image is tainted with mismatched triggers, while in the presence of a matched trigger, the backdoor class should receive a large (ideally the largest) score. This behaviour, induced by the backdoor, characterizes the samples from the in-set classes. For the samples of out-of-set class, for which there is no matching trigger, this behaviour is not observed, and the samples tainted with the various triggers are predicted randomly by the network. Figure 9.2 (left) shows an example of M matrix obtained for an input sample x belonging to class 1 (in this case $\mathcal{C} = \{1, 2, 3, 4, 5\}$). We see that, as expected, in every row, but in the first (class $y = 1$), a high logit score is associated with the true class since the superimposed trigger (being mismatched) does not affect the prediction of the network. In correspondence of the first row, instead, a high score is associated with the $(C + 1)$-th entry. An example of matrix M obtained for an out-of-set sample is shown in 9.2 (right). The M matrix now shows a completely different behaviour with respect to the one in the left part of the figure.

Based on the above observations, and given the tentative predicted label y^* computed from x (see Eq. (9.4)), an obvious way to define the rejection score would be to rely on the so-called matched trigger logit score (TLS-M), namely, $M(y^*, C + 1)$, with large values indicating a large probability that the input sample belongs to a in-set class. Another possibility would be to base the rejection on the maximum logit score in M (MLS-M), with the idea that samples of the in-set class should return higher scores than out-of-set samples. However, as shown in Eq. (9.2), for out-of-set classes, in the presence of non-matched triggers, the model is expected to behave normally. Hence, we can expect that the network will also produce large MLS-M scores. In order to exploit also the predictions obtained with non-matched triggers, that for in-set samples are expected to be high in correspondence of the true sample class, we defined a combined logit score (CLS-M) as

Algorithm 1 BOSC network testing

Input:
Test input x;
Triggers T;
Number of classes C;
Backdoored model ϕ;
Predefined threshold ν for rejection;
1: Initialization: $M = [\mathbf{0}]_{C,(C+1)}$
2: **for** each $i \in C$ **do**
3: $M(i,:) \leftarrow \phi(\mathcal{E}(x, t_i))$, $t_i \in T$
4: **end for**
5: Get y^* via Eq. (9.3)
6: Calculate the CLS-M score $\xi_r(M)$ based on Eq. (9.5)
Output: y^* is returned if $\xi_r > \nu$; otherwise, $\Re$ is returned

follows:

$$\xi_r(M) = \frac{1}{C}\sum_{i=1}^{C} M(i, y^*) + M(y^*, C+1). \tag{9.5}$$

The rationale behind the definition of ξ_r is that, for a given tentative predicted class, if the trigger and the class match, samples of in-set classes are expected to result in a higher backdoor logit score $M(y^*, C+1)$, with the class logit score $M(y^*, y^*)$ possibly being the second-best. For the remaining $i \in \{1, 2, \cdots, C\}$, $i \neq y^*$, samples of in-set classes are expected to produce higher y^*-th class logit scores than samples of out-of-set classes. Given the score $\xi_r(M)$, the final output of the open-set classifier is obtained as detailed in Eq. (9.4), that is, the output of the in-set classifier is accepted if $\xi_r(M) > \nu$, and rejected otherwise.

We observe that an in-set prediction could also be obtained from the matrix M, e.g., by summing over the columns and taking the maximum (that is, evaluating $\arg\max_j(\sum_{i=1}^{C} M(i,j))$. Based on our experiments, doing so yields (almost) the same results as using Eq. (9.3).

A summary of BOSC testing procedure is given in Algorithm 1. A comparison of the performance achieved using different rejection scores is reported in Section 9.4.4. The results confirm the superior effectiveness of the combined logit score.

9.2.3 Training strategy

In the following, we provide the details of the methodology we followed to train the backdoored model.

In our framework, the backdoor is injected within the network by the model's trainer himself to improve the open set classification performance of the model. For this reason, instead of tainting the samples of the dataset in a stealthy way, as done to implement a backdoor attack [29], tainting can be applied while training, randomly choosing a percentage of to-be-tainted samples from each batch at every iteration and tainting them. We

refer to this scenario as tainting on-the-fly. More formally, given a dataset D of samples x from C in-set classes, training is performed on batches. Let B indicate the set of samples in a batch. At every iteration, we randomly sample a fraction γ of the batch samples and taint them as detailed in Eq. (9.1) by injecting a trigger matched to the true class of x. We denote with B^t the subset of tainted samples (hence, $\gamma = |B^t|/|B|$)[2]. Another random fraction γ of images in the batch is tainted with a randomly chosen mismatched trigger (i.e., a trigger associated with a class different from the class of x). We indicate with B^{mt} the corresponding tainted subset and with B^c the subset of clean samples. Training is achieved by optimizing the following loss:

$$\mathcal{L} = \sum_{x \in B^c} \mathcal{L}(x, y) + \lambda_1 \sum_{x \in B^t} \mathcal{L}(x, C+1) + \lambda_2 \sum_{x \in B^{mt}} \mathcal{L}(x, y) \tag{9.6}$$

where y denotes the true label of x, λ_1 and λ_2 are balancing parameters controlling the importance of the backdoor loss terms, and $\mathcal{L}$ is the CE loss ($\mathcal{L}(x, y) = -\log(f_y(x))$.

During training, we also implemented an augmentation strategy inspired by [205] to improve the generalization capability of the model and its robustness against image processing. Given an input image x from a given class, the image is perturbed with an image z from a different class, obtaining the perturbed image $x' = x + \beta z$, where β is the perturbation strength, $\beta \ll 1$ (clipping is performed to ensure that the values remain in the [0,1] range), while keeping the label unchanged. Specifically, a fraction η of the samples in B^c is perturbed with the above procedure, referred to as mixup augmentation in the following[3]. The benefit brought by mixup augmentation on the performance of BOSC will be detailed later (Table 9.3).

9.3 Experimental Methodology

In this section, we describe the methodology that we followed to use BOSC for open-set synthetic image attribution. To confirm the generality of our approach for image forensics applications, in Section 9.5.2, we will also apply it to the classification of AI-based face image attribute editing.

9.3.1 Dataset

The synthetic image dataset we used in the Chapter is SIAD_v2, including 10 generative architectures. In our experiments, we considered the same three different splittings of in-set and out-of-set architectures considered in the previous chapter and detailed in Table 8.1. The in-set architectures were used to train the BOSC model, whereas the out-of-set architectures were only utilized for testing. We point out that, in the first and second configurations, the in-set comprises a mixture of GANs, DM, and Transformers, while the third configuration only includes GANs in the in-set. For every architecture, we took

[2]We assume w.l.o.g. that $\gamma|\mathcal{B}|$ is an integer.

[3]We are implicitly assuming that the fraction of samples in B^c is larger than η. In fact, these fractions are always small and $2\gamma + \eta < 1$.

Figure 9.3 – All the trigger images used in our work. The top five are used for the GAN attribution task, and all of them are used for synthetic facial editing classification.

20,000 images and distributed them across training, validation, and test sets as follows: 16,000 for training, 2,000 for validation, and 2,000 for testing[4]. In every configuration, training and validation images of the dataset were only considered for in-set architectures.

9.3.2 Backdoor and training setting

To inject the backdoor, we used cartoon images as triggers. The five trigger images that we used each one matched to an in-set architecture, are shown in the top row of Figure 9.3. It is worth noting that different trigger images could be chosen. We decided to consider triggers whose representative features are expectedly different from those that are relevant for the classification task[5]. The tainting strength α in Eq. (9.1) is set to 0.1. The tainting ratio γ is also set to 0.1.

An EfficientNet-B4 was used as a baseline network. The input size is set to $384 \times 384 \times 3$. We trained the network with a batch size of 32 for 15 epochs. Training was performed via Adam optimizer with a dynamic learning rate initially set to 10^{-4} and multiplied by 0.1 every 5 epochs. Concerning the loss tradeoff parameters λ_1 and λ_2, they are both set to 0.1. The mixup augmentation parameters were set as $\beta = 0.15$ and $\eta = 0.1$. The following augmentations have been considered during training: flipping and JPEG compression, applied to the input with probability 0.5 and random quality factors for JPEG in the range $[70, 100]$.

9.3.3 State-of-the-art comparison

To validate the effectiveness of the proposed method, we ran comparisons with both general methods proposed in the machine learning literature for OSR and methods specifically developed for synthetic image attribution. More specifically, for open set recognition, we considered the ARPL [173], AKPF [117] and CSSR [168](namely both the PCSSR and RCSSR variants of the methods), mentioned in Section 6.1. The above methods were tested using the code publicly available in the configuration used in the papers. With regard to open set attribution methods, we considered the methods presented in Chapter

[4]We reduce the training number from 50k to 20k to consider more unseen models for generalization test in Section 9.4

[5]The optimization of the trigger images is left as future work.

Table 9.1 – Performance of architecture attribution in closed-set (Accuracy (%)) and open-set (AU-ROC (%), AU-OSCR (%)). The best results are shown in bold (the second-best is underlined).

Methods	Config-S1			Config-S2			Config-S3			Average		
	Closed-set	Open-set		Closed-set	Open-set		Closed-set	Open-set		Closed-set	Open-set	
	ACC	AU-ROC	AU-OSCR	ACC	AU-ROC	AU-OSCR	ACC	AU-ROC	AU-OSCR	ACC	AU-ROC	AU-OSCR
ARPL [173]	**100**	79.59	79.55	99.94	77.84	77.78	99.98	83.01	83.00	99.97	80.15	80.11
AKPF [117]	99.95	75.27	75.27	**100**	88.98	88.98	99.56	**91.89**	**91.6**	99.84	85.38	85.28
PCSSR [168]	99.47	83.11	82.88	99.57	68.58	68.50	98.55	71.32	70.74	99.20	74.34	74.04
RCSSR [168]	99.62	82.65	82.46	99.21	57.98	57.84	98.65	70.79	70.32	99.16	70.47	70.21
ResVit (Chap. 7)	99	79	78.32	99	76	75.89	99	68	67.93	99	74.33	74.05
SiaVerify (Chap. 8)	**100**	82.24	82.31	**100**	82.44	82.41	**100**	82.98	82.89	**100**	82.55	82.54
POSE [118]	98.56	75.97	75.60	96.90	86.73	85.53	96.70	83.00	81.50	97.39	81.90	80.88
Baseline	**100**	82.62	82.56	**100**	73.10	73.10	99.99	65.99	65.98	**99.99**	73.90	73.88
BOSC (Prop.)	**100**	**95.31**	**95.31**	99.95	**95.43**	**95.41**	99.96	90.00	89.99	99.96	**93.58**	**93.42**

7, named ResViT and Chapter 8, referred to in the following as SiaVerify, and finally the latest POSE [118] with a rejection option as well. In addition to closed-set accuracy and open-set AU-ROC, we also considered a new open-set metric, namely AU-OSCR (defined in Section 2.4.5), that measures the capability of the system to classify samples after out-of-set rejection.

9.4 Experimental Results

9.4.1 Performance analysis

The closed-set and open-set performance of the BOSC method in all three configurations are reported in Table 9.1, where they are compared with the state-of-the-art methods mentioned in Section 9.2.2. In addition, to better assess the gain achieved with BOSC we also report the performance achieved by using the same EfficientNet-B4 baseline to build a C-class classifier and adopting the MLS for open-set detection (as mentioned in Section 6.1, MLS has been proven to achieve the best rejection performance in many cases [161], and is adopted for OS classification in several papers [158,161]). This method is referred to as 'baseline' in the table. We see that all methods achieve nearly perfect accuracy in closed-set settings, while the performance in open-set conditions is different. In particular, the general CSSR methods show limited effectiveness, as well as ResVit. The best performing state-of-the-art method is AKPF, which achieves an average AU-ROC equal to 85.38% and AU-OSCR equal to 85.28%. Most of the methods, especially the general OS classification methods, exhibit unstable open-set performance across the three configurations, e.g. for AKPF the AU-ROC ranges from 75.27% in Config-S1 to 91.89% in Config-S3. Results are more stable for SiaVerify and POSE.

Regarding BOSC, it achieves the best open-set performance (AU-ROC = 93.58% and

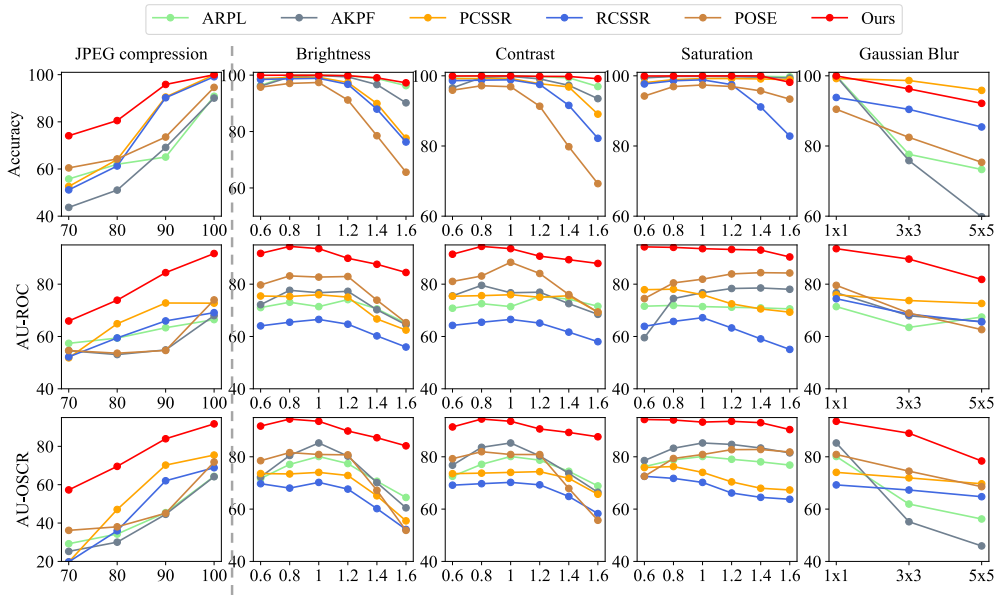


Figure 9.4 – Robustness to different image-level attacks. From left to right: brightness, contrast, saturation and JPEG compression. From up to bottom: Config-S1, Config-S2 and Config-S3.

AU-OSCR= 93.42% on average), with very limited variability across the configurations. In particular, BOSC outperforms the best-performing state-of-the-art method AKPF with a gain of 8.20% and 8.14% in AU-ROC and AU-OSCR, respectively. The baseline is also significantly surpassed by BOSC, by approximately 20% in both AU-ROC and AU-OSCR, confirming the effectiveness of the backdoor-based framework.

9.4.2 Robustness to image processing manipulations

We also evaluate the robustness of the method against image processing. In particular, we considered color modifications (saturation, brightness, contrast), Gaussian blur and JPEG compression. We point out that only JPEG compression has been considered during the training of our method (see Section 9.3.2), while the others correspond to never-seen processing operations. For the color modifications, an example of a processed image is reported in Figure 9.5 for the extreme values of the range of parameters considered.

Figure 9.4 reports both the closed-set and open-set performance (AU-ROC and AU-OSCR). BOSC has similar robustness performance to the state-of-the-art methods in closed-set, while it gets superior performance in open-set, with a gain in AU-ROC and AU-OSCR larger than 10% in all the cases. This confirms the intuition that since the trigger image is not affected by the processing (being superimposed at test time), the relevant features are affected to a lesser extent by the processing. The most critical case is the case of JPEG compression, notwithstanding the inclusion in the training set.

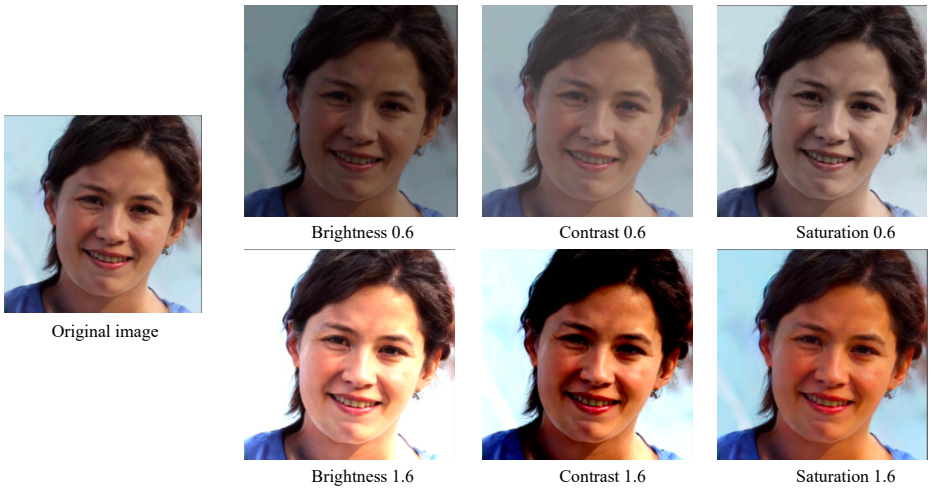


Figure 9.5 – Image examples of brightness, contrast, and saturation change.

Table 9.2 – Closed (ACC (%)) and open-set (AU-OSCR (%)) results in the case of models trained with different datasets, parameters, and training procedures (in the last line, the number in the models' names refers to the image resolution). The underline indicates the average result across different configs.

Architecture	Type of Mismatch	Train	Test	ACC	AU-OSCR
DDPM (Config-S1)	Training Methodology	DDPM	DDPM-ema	100	85.89
StyleGAN2 (Config-S1&S2&S3)	Training Methodology	StyleGAN2-f	StyleGAN-ada	<u>99.84</u>	<u>93.24</u>
Taming Transformer (Config-S1)	Real Dataset	CelebA	FFHQ	98.20	82.53
Latent Diffusion (Config-S1&S2)	Real Dataset	CelebA	FFHQ	<u>77.63</u>	<u>71.3</u>
LSGM (Config-S2)	Training Methodology	Quantitative (2-stages)	Qualitative (3-stages)	99.60	76.46
StyleGAN3 (Config-S3)	Image Resolution	StyleGAN3 t-1024& t-ffhqu1024&r	StyleGAN3 t-ffhqu256	99.35	87.22

9.4.3 Generalization test

We also evaluated the capability of the proposed method of correctly attributing to the source architecture images generated by *unknown* models, that is, models different than

those considered during the training yet corresponding to in-set architectures. In particular, the generative models used for these tests are obtained from in-set architectures considering i) different training strategies (for StyleGAN2, LSGM and DDPM), ii) different datasets of real images used for training (for Latent diffusion and Taming transformer), and iii) different image resolution (for StyleGAN3). The details of the mismatch between training and testing models are provided in Table 9.2. With regard to the models obtained considering a mismatch in the training strategy, StyleGAN-ada refers to training with adaptive augmentation for the discriminator [192], while DDPM_ema is obtained by training with the exponential moving average strategy [206]. Finally, for LSGM, the number of stages for the training is changed from the default number 2 to 3. In the third stage, re-training is performed by training only the SGM prior, leaving the Nouveau VAE (NVAE) component fixed.[6] The models obtained with the 2-stage and 3-stage training are referred to as quantitative and qualitative models, respectively.

The last two columns of Table 9.2 report the closed-set Accuracy and the AU-OSCR obtained when the system is tested with these unknown models. The results show that the closed-set performance is very good (ACC above 98%) in all cases, but for Latent Diffusion, where we get ACC = 77.63%. The model mismatch affects the open-set performance more, and in fact, the AU-OSCR computed on the mismatched samples decreases. Performance is great in the case of Style-GAN2 and remains pretty good also for DDPM and Taming Transformer, while they drop in the case of Latent Diffusion, LSGM and SyleGAN3. A strategy that we expect can help mitigate this issue is to include multiple models for every architecture inside the training set. Arguably, doing so should induce the system better to learn the model variability for a given generative architecture.

9.4.4 Ablation study

We carried out an ablation study to assess the impact of each component of the BOSC method.

Choice of rejection score

The benefit of considering the trigger-based score in Eq. (9.5) for sample rejection with the backdoor-based network is shown in Figure 9.6, where the open-set detection performance (AU-ROC) obtained using various scores is reported. In particular, the proposed score is compared with other trigger-based scores, that is, the TLS-M (see section 9.2.2 for the definition) and the MLS-S, namely, the maximum value of the M_t matrix, and also scores commonly used for OSR, which are directly obtained from the prediction output vector of x. In particular, we considered the maximum value of the softmax probability vector (MSP) and the maximum logit score (MLS).

We see that the AU-ROC obtained with trigger-based scores (MLS-M, TLS-M, CLS-M) is much higher than the AU-ROC obtained with common scores computed on the output of clean samples (MSP and MLS). Among the three trigger-based scores, we see that TLS-M improves the performance of MLS-M from 87.96% to 91.24% on average.

[6] See `https://github.com/NVlabs/LSGM` for the details.

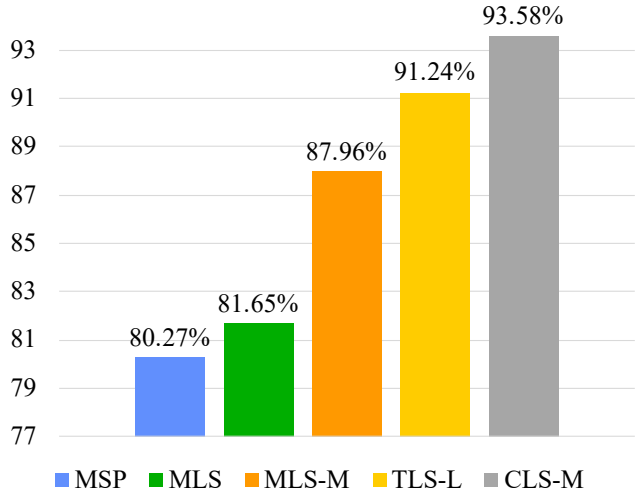


Figure 9.6 – Average open-set AU-ROC performance on two tasks, attribution and face editing classification. For each task, three configurations of in-set and out-of-set are considered.

Table 9.3 – Ablation study on the effect of the mixup augmentation.

	Accuracy (%)	AU-ROC (%)	AU-OSCR (%)
Baseline	100	73.90	73.88
BOSC (w/o Mixup)	99.98	86.00	85.99
BOSC (w/- Mixup)	99.97	93.58	93.57

The performance is further improved with the proposed CLS-M, which fully exploits the behavior with matched and mismatched triggers, in which case the AU-ROC reaches 93.58%.

Mixup augmentation

We also ran experiments to assess the benefit of the mixup augmentation strategy. Table 9.3 reports the closed and open-set performance achieved by the BOSC method when the training is carried out with and without the mixup augmentation. The performance of the baseline is also reported. We see that, while the accuracy values of all the models are the same, the open-set performance is noticeably improved by the adoption of the mixup augmentation. In particular, the gain brought by the mixup strategy in the performance of the BOSC method is 7.58% in both AU-ROC and AU-OSCR.

Table 9.4 – Performance of facial editing classification in closed-set (Accuracy (%)) and open-set (AU-ROC (%), AU-OSCR (%)). The best results are shown in bold (the second-best is underlined).

Methods	Config-G1			Config-G2			Config-G3			Average		
	Closed-set	Open-set		Closed-set	Open-set		Closed-set	Open-set		Closed-set	Open-set	
	ACC	AU-ROC	AU-OSCR	ACC	AU-ROC	AU-OSCR	ACC	AU-ROC	AU-OSCR	ACC	AU-ROC	AU-OSCR
ARPL [173]	91.92	86.84	82.54	94.41	87.34	84.57	90.99	85.84	80.64	92.44	86.67	82.58
AKPF [117]	94.41	91.09	87.84	95.33	88.72	86.49	91.45	87.35	82.58	93.73	89.05	85.64
PCSSR [168]	95.33	85.60	82.77	96.40	82.60	80.63	91.87	86.23	81.39	94.53	84.81	81.60
RCSSR [168]	95.05	82.46	79.48	97.02	89.98	88.37	93.27	82.68	78.41	95.11	85.04	82.09
ResVit (Chap. 7)	93.65	91.42	87.65	95.59	**91.66**	89.50	91.83	86.64	82.13	93.69	89.91	86.43
Baseline	**97.21**	87.85	86.66	**97.91**	88.22	87.24	**95.09**	86.10	83.92	**96.74**	87.39	85.94
BOSC (Ours)	96.65	**92.13**	**90.35**	97.28	91.62	**90.49**	94.50	**88.43**	**85.40**	96.16	**90.73**	**88.75**

9.5 Application to the Classification of Facial Editing

In this section, we describe the experiments we ran on the task of classification of facial attribute editing, where we exploited the BOSC framework to address the open-set scenario.

9.5.1 Dataset

The dataset employed for these experiments is FARD_v2, as detailed in Section 2.4.3. The face images are taken from the CelebA-HQ dataset. The images are manipulated with the same 18 edit types: 4 facial attributes are edited with InterfaceGAN [11], and 14 facial attributes with StyleCLIP [12]. Three different splittings (Config-G7, Config-G8 and Config-G9) of in-set and out-of-set edit types are considered from Table 7.1 and renamed as 'Config-F1', 'Config-F2' and 'Config-F3'. The BOSC framework is trained as described in the previous section, using the same parameters' setting reported in Section 9.3.2. The 11 trigger images used for training are illustrated in Figure 9.3.

9.5.2 Results

The proposed method is compared with ARPL [173], AKPF [117], PCSSR [168], RCSSR [168] (general OSR methods), and the ResVit method in Chapter 7, which, as pointed out before, corresponds to a method specifically proposed for this task. The performance of an EfficientNet-B4 trained on the in-set classes using MLS for out-of-set detection (baseline) is also reported.

The results achieved for closed-set and open-set settings are shown in Table 9.4. We see that all methods obtain AU-ROC and AU-OSCR larger than 80%, while the closed-set accuracy of the various methods ranges between 92% and 97%, with the BOSC method

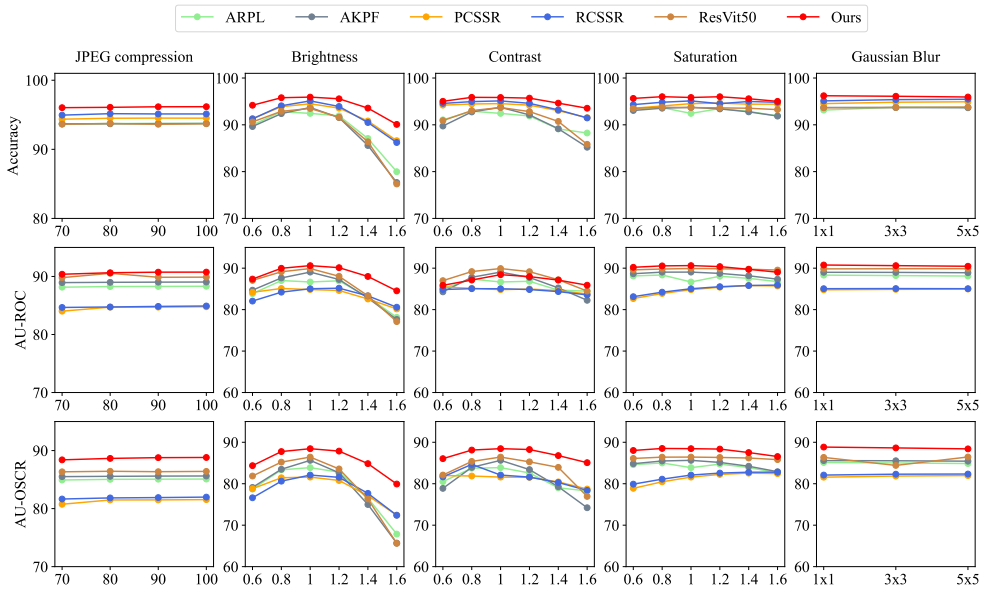


Figure 9.7 – Robustness to different image-level attacks. From left to right: brightness, contrast, saturation and JPEG compression. From up to bottom: Config-S1, Config-S2 and Config-S3.

performing the best in almost all the cases[7]. Besides, the results of all methods are stable across the various configurations. The best open-set results for the state-of-the-art are achieved by AKPF and ResVit, with the latter yielding the best performance. Compared to ResVit, BOSC achieves a slight gain of 0.82% in AU-ROC and 2.32% in AU-OSCR (and a 2.47% gain in the Accuracy). We stress that ResVit is specialized for this task and resort to the aid of a localization branch to focus on the face regions that are most relevant for the various editings. Therefore, the similar (slightly better) performance obtained by BOSC represents a noticeable result, proving the generality of the proposed method.

Finally, the robustness performance of the various methods against brightness, contrast, saturation, Gaussian blur and JPEG compression is reported in Figure 9.7. We see that BOSC always achieves the best robustness results. In the case of contrast adjustment, AKPF and ResVit slightly outperform BOSC in terms of out-of-set detection capability (AU-ROC). However, the overall classification performance in an open set (AU-OSCR) is noticeably superior to our method.

9.6 Summary

In this chapter, we have presented a backdoor-based open-set classification (BOSC) framework for open-set classification that has been adopted for synthetic image attribution.

[7]The results of PCSSR, RCSSR, and ARPL in this chapter are slightly different with respect to the results in Chapter 7 because the configurations are different.

The framework assigns an expected reaction by utilizing class-specific triggers for in-set classes, aiming to widen the gap between in-set and out-of-set samples. Additionally, to facilitate classification with a rejection option, we introduced a new open-set score based on the output matrix of the model, achieved by testing the query image with all in-set triggers. We conducted experiments on two tasks: facial attribute editing classification and synthetic image attribution, to demonstrate the versatility and effectiveness of the BOSC framework in open-set forensic applications, and its strong robustness against post-processing that stems from the backdoor testing framework adopted and the use of the trigger signal.

Moreover, we noticed that mixup augmentation has a positive impact on open-set performance when combined with backdoor attack training using class-specific triggers. This can be an interesting point to be explored in the future. In addition, the use of triggers during the test also makes the system robust against image post-processing by assigning partial attention from the in-set samples to the triggers. On the other hand, the challenge of the open-set problem is obvious. There is always a risk of misclassifying out-of-set samples as in-set samples without knowledge of out-of-set classes during the training.

Chapter 10

Conclusion

"There are things known and there are things unknown,
and in between are the doors of perception."
Aldous Huxley

In this chapter, we summarize the main contributions of the thesis and identify the main open challenges for future research in the field.

10.1 Summary

The widespread use of generative AI techniques for image generation and editing is raising several threats, including fraud, disinformation, and the erosion of public trust. When this research started, many systems powered by AI could effectively detect images in controlled environments. These systems, however, are generally unable to handle the challenges posed by real-world applications. Moreover, due to the pace at which generative AI techniques are progressing, AI forensic systems have to work in uncontrolled environments where the training conditions are often not met. It is likely, in fact, that a fake image detector be asked to judge images generated or edited by generative models that have been released after the system was deployed.

This thesis contributes to the development of forensic systems capable of operating *in the wild* in two ways. In the first part of the thesis, we tackled with the dataset mismatch problem, which arises when test samples belong to the same categories/classes considered during training, but have been generated by relying on different techniques, or have been subject to a different post-processing pipeline. We addressed this issue by proposing several methods that rely on semantic-related information, following the intuition that tools that rely on such information have more robustness and generalization capability. We first introduced a Siamese network architecture designed to detect AI-synthetic images by relying on eye clues. This approach exploits inconsistencies between the eyes of synthetic images, which enhances the method's robustness against common image post-processing methods and rebroadcast attacks (such as image print&scan). Then, we presented a hybrid architecture for detection/classification, incorporating a localization branch dedicated to the localization of the manipulated image regions. We observed that integrating a localization branch induces the network to focus on the most relevant parts of the image, resulting in significant improvements in generalization capabilities and robustness against image processing operations. We validated the effectiveness of the proposed hybrid network by applying it to the detection of fake images of climate change (specifically, flood

images) generated by the ClimateGAN architecture, and to multi-class classification of GAN-based face editing by enriching the hybrid scheme with a new ingredient: multi-level analysis. Overall, the methods we have developed demonstrate strong generalization and robustness against various post-processing techniques, except for the Gaussian noise attack.

In the second part of the thesis, we focused on open-set classification, where the system can also be asked to operate on samples belonging to categories/classes that have not been considered at training time. We considered two approaches to address this scenario: classification with a rejection option, and verification. We devised solutions for the problems of synthetic image attribution and synthetic facial attribute classification. With regard to classification with rejection, we focused on the development of classifiers capable of reliably identifying unknown out-of-set samples and refraining from providing wrong predictions for them. We first employed the hybrid classification/localization architecture used in the first part of the thesis to design a classifier with rejection. This design integrated the localization with a ViT module to automatically learn the correlations among local image patches under the supervision of a semantic mask based on the manipulated facial attributes. In addition, we validated the effectiveness of the ViT on synthetic image attribution by eliminating the localization branch, as the images were entirely synthesized. Then, we introduced a novel framework that exploits the concept of backdoor attacks to develop a classifier with a rejection option. By incorporating class-specific triggers into the samples of in-set classes, the model's response to various triggers can facilitate out-of-set rejection. In this setup, high confidence is anticipated for in-set class samples that match the predefined trigger during training, while out-of-set samples yield low confidence scores. Robustness is ensured because the trigger prevents it from being processed.

With regard to the verification approach, instead, we developed a system to decide whether two input samples belong to the same class, exploiting the contrastive learning framework. Compared to classification with a rejection option, verification can achieve clustering even if all test sample labels are unknown, regardless of in-set or out-of-set samples in open-set scenarios. Moreover, attribution verification can be extended into a classifier when the classes of the reference images are known. Such an approach naturally extends to the open-set scenario.

10.2 Open Issues

Based on the status of the current literature and the advancements presented in this thesis, we can identify several unsolved issues, which we forecast will occupy the agenda of researchers for the next years.

- *Interpretability.* The interpretability of AI-based solutions is a crucial requirement, somewhat related to challenges posed by the application of media forensics tools *in the wild*. Providing users with evidence on the reasons why certain decisions are taken, in fact, may help to judge if a decision is reliable or not. When the evidence is unclear or not very strong (this may be the case when a decision is made on unknown

class samples), the prediction is not trustworthy. In this thesis, we contributed to the development of interpretable systems through the design of semantic-based systems, in particular: i) a detection method which looks at semantic attributes of the images (eyes); ii) a classification method which exploits localization to guide the detection and forces the network to look at the most semantically relevant parts of the images, which are related to the manipulation. With regard to i), since generative models continuously evolve, they have gradually learned to generate images with realistic-looking eyes that can deceive eyes-based detectors. Investigating other semantic facial attributes, such as the mouth [207,208], nose [97], etc., and exploiting effective fusion strategies can help to develop interpretable tools with enhanced performance. Regarding ii), the interpretability of the solutions we have proposed follows from the adoption of the focus of attention/localization mechanism. However, there are cases where the explainability obtained in this way is limited, e.g. in the case of facial attributes manipulation, when different edits are performed on the same facial area (e.g., young and old). In this case, an interesting possibility is to invert the manipulated image into an "original" image based on the manipulation predicted. If the prediction is correct, we expect a high-quality recovered 'original' image; conversely, an incorrect prediction results in a low-quality image. In this way, the recovered image's quality might reflect the system's confidence and the reliability of the classification. Another option is to integrate the image attribution analysis with large language models (LLM). For instance, the user can ask the LLM to show some hints that may prove that the images are generated or manipulated. This can work as a general framework regardless of the image content.

- *Security.* While the methods proposed in this thesis have good robustness against generic image processing operations, sometimes also referred to as laundering attacks, the robustness against intentional attacks aimed at system failure has not been considered. However, like DL-based classifiers developed for computer vision and pattern recognition applications, DL-based deepfake detectors are known to be vulnerable to adversarial examples [209]. This highlights the importance of developing defense techniques to mitigate the possible impact of adversarial attacks on the predictions. Secure techniques can be developed by exploiting adversarial training, robust optimization, or incorporating uncertainty estimation. The development of secure forensic detectors and classifiers is essential for ensuring the effectiveness and reliability of forensic systems in real-world applications, wherein the presence of an adversary aimed at making the system fail can not be ignored. Only very few scattered attempts have been made in this direction [210], and the development of secure DL-based image forensic systems is still an open research direction.

- *Cross-domain generalization.* The methods developed in this thesis for the detection, classification and attribution of synthetic images focus on the facial domain. However, AI-generated images encompass non-facial images, and synthetic images can also be generated in other domains wherein the widespread of fake content has a critical impact, e.g., satellite images, medical images (e.g., fake western blot), natural landscape images, etc.,... However, it has been shown in the literature that

systems developed for a specific domain typically do not generalize well to a different domain [211]. Despite some recent attempts made in this direction [212, 213], the development of domain-universal methods for synthetic image detection and attribution, capable of detecting the presence of the artificial fingerprints left by the generators regardless of the content of the image, is still an open research problem.

- *The open-set challenge.* The development of methods capable of working in open-set settings is crucial for ensuring the reliability of systems when deployed in the real world. In this thesis, we contributed to this research direction by developing methods for classification and attribution by following two different approaches: classification with rejection and verification. However, getting good open-set performance while retaining good generalization to dataset-mismatch for in-set samples is not easy. Future research in this direction can explore the use of various augmentation strategies, e.g., the mixup augmentation [205], and ad-hoc augmentations, for instance, resorting to generative augmentation, to design a more suitable and compact known-class feature space, that can get good closed-set performance and generalization, and avoid misclassification of unknown-class samples as known ones at the same time.

- *Active methods.* Many existing forgery detection methods adopt a passive approach, focusing on analyzing artificial traces left within the images by the editing tools. However, this approach presents several challenges, as the artificial features may differ significantly from those generated by the underlying generative models. To mitigate these challenges, active techniques that address detection directly from the source offer a promising alternative. By considering active techniques, researchers can adopt proactive measures to identify and assess the authenticity of images at the source. For example, ensuring that all generated or manipulated images contain metadata detailing information about the generative model, the type of manipulation (e.g., generated or edited), and user information can enhance the detection process. This metadata provides valuable insights into the origin and history of the images, enabling more accurate and reliable forgery detection. Furthermore, enabling identifiable watermarking in images and models offers another valuable tool for forensic analysis, enhancing the ability to trace and verify the origin and authenticity of digital content.

- *Multi-modal deepfake detection.* The generated/manipulated images are usually described as text messages to spread misleading information and videos. For instance, Shao et al. [214] built a large dataset that considers image manipulations associated with text editing. Khalid et al. [215] presented a novel audio-video multimodal deepfake dataset. In light of these developments, there is a growing consensus within the research community on the importance of leveraging multi-modal features for detecting manipulated content. Techniques that combine information from different modalities, such as images, text, audio, and video, hold promise for enhancing detection accuracy and resilience against evolving forms of deception. By developing robust detection frameworks that integrate insights from diverse data

sources, researchers can help safeguard online communities from the harmful effects of misinformation and fake content dissemination.

Bibliography

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[3] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," *Advances in neural information processing systems*, vol. 29, 2016.

[4] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[5] 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*, 2019, pp. 4401–4410.

[6] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8110–8119.

[7] 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*, vol. 34, 2021.

[8] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 873–12 883.

[9] A. Vahdat, K. Kreis, and J. Kautz, "Score-based generative modeling in latent space," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 287–11 302, 2021.

[10] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[11] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of gans for semantic face editing," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9243–9252.

[12] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "Styleclip: Text-driven manipulation of stylegan imagery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2085–2094.

[13] O. Avrahami, O. Fried, and D. Lischinski, "Blended latent diffusion," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–11, 2023.

[14] B. DeCann and K. Trapeznikov, "Comprehensive dataset of face manipulations for development and evaluation of forensic tools," *arXiv preprint arXiv:2208.11776*, 2022.

[15] 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," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[16] 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)*. IEEE, 2021, pp. 1–6.

[17] A. Ferreira, E. Nowroozi, and M. Barni, "Vipprint: Validating synthetic image detection and source linking methods on a large scale dataset of printed documents," *Journal of Imaging*, vol. 7, no. 3, p. 50, 2021.

[18] V. Schmidt, A. S. Luccioni, M. Teng, T. Zhang, A. Reynaud, S. Raghupathi, G. Cosne, A. Juraver, V. Vardanyan, A. Hernandez-Garcia *et al.*, "Climategan: Raising climate change awareness by generating images of floods," *arXiv preprint arXiv:2110.02871*, 2021.

[19] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, "Attentional feature fusion," in *WACV*, 2021, pp. 3560–3569.

[20] J. Betker, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee, Y. Guo *et al.*, "Improving image generation with better captions," *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2023.

[21] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The deepfake detection challenge (dfdc) dataset," *arXiv preprint arXiv:2006.07397*, 2020.

[22] 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*, 2019, pp. 1–11.

[23] 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*, 2020, pp. 8695–8704.

[24] L. Bondi, E. D. Cannas, P. Bestagini, and S. Tubaro, "Training strategies and data augmentations in cnn-based deepfake video detection," in *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2020, pp. 1–6.

[25] L. Verdoliva, "Media forensics and deepfakes: an overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 910–932, 2020.

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*.

[27] S. Girish, S. Suri, S. S. Rambhatla, and A. Shrivastava, "Towards discovery and attribution of open-world gan generated images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 094–14 103.

[28] Z. Sun, S. Chen, T. Yao, B. Yin, R. Yi, S. Ding, and L. Ma, "Contrastive pseudo learning for open-world deepfake attribution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 882–20 892.

[29] 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*, vol. 3, pp. 261–287, 2022.

[30] A. Asperti, D. Evangelista, and E. Loli Piccolomini, "A survey on variational autoencoders from a green ai perspective," *SN Computer Science*, vol. 2, no. 4, p. 301, 2021.

[31] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial networks," in *5th International Conference on Learning Representations, ICLR 2017*, 2017.

[32] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[33] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International conference on machine learning*. PMLR, 2015, pp. 1530–1538.

[34] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. PMLR, 2015, pp. 2256–2265.

[35] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[36] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[37] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[38] D. Berthelot, T. Schumm, and L. Metz, "Began: Boundary equilibrium generative adversarial networks," *arXiv preprint arXiv:1703.10717*, 2017.

[39] B. Zhang, S. Gu, B. Zhang, J. Bao, D. Chen, F. Wen, Y. Wang, and B. Guo, "Styleswin: Transformer-based gan for high-resolution image generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 304–11 314.

[40] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 87–110, 2022.

[41] L. Zhao, Z. Zhang, T. Chen, D. Metaxas, and H. Zhang, "Improved transformer for high-resolution gans," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 367–18 380, 2021.

[42] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[43] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[44] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[45] Y. Yu, W. Zhang, and Y. Deng, "Frechet inception distance (fid) for evaluating gans," *China University of Mining Technology Beijing Graduate School*, 2021.

[46] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[47] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.

[48] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "Attgan: Facial attribute editing by only changing what you want," *IEEE transactions on image processing*, vol. 28, no. 11, pp. 5464–5478, 2019.

[49] M. Liu, Y. Ding, M. Xia, X. Liu, E. Ding, W. Zuo, and S. Wen, "Stgan: A unified selective transfer network for arbitrary image attribute editing," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3673–3682.

[50] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "Ganspace: Discovering interpretable gan controls," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9841–9850, 2020.

[51] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[52] Y. Shi, X. Yang, Y. Wan, and X. Shen, "Semanticstylegan: Learning compositional generative priors for controllable image synthesis and editing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 254–11 264.

[53] C. Eastwood and C. K. Williams, "A framework for the quantitative evaluation of disentangled representations," in *International conference on learning representations*, 2018.

[54] Z. Wu, D. Lischinski, and E. Shechtman, "Stylespace analysis: Disentangled controls for stylegan image generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 863–12 872.

[55] D. Roich, R. Mokady, A. H. Bermano, and D. Cohen-Or, "Pivotal tuning for latent-based editing of real images," *ACM Transactions on graphics (TOG)*, vol. 42, no. 1, pp. 1–13, 2022.

[56] Y. Xu, Y. Yin, L. Jiang, Q. Wu, C. Zheng, C. C. Loy, B. Dai, and W. Wu, "Transeditor: Transformer-based dual-space gan for highly controllable facial editing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7683–7692.

[57] Y. Alaluf, O. Patashnik, Z. Wu, A. Zamir, E. Shechtman, D. Lischinski, and D. Cohen-Or, "Third time's the charm? image and video editing with stylegan3," in *European Conference on Computer Vision*. Springer, 2022, pp. 204–220.

[58] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "Sdedit: Guided image synthesis and editing with stochastic differential equations," in *International Conference on Learning Representations*, 2021.

[59] B. Yang, S. Gu, B. Zhang, T. Zhang, X. Chen, X. Sun, D. Chen, and F. Wen, "Paint by example: Exemplar-based image editing with diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 381–18 391.

[60] O. Avrahami, D. Lischinski, and O. Fried, "Blended diffusion for text-driven editing of natural images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 208–18 218.

[61] Z. Pan, R. Gherardi, X. Xie, and S. Huang, "Effective real image editing with accelerated iterative diffusion inversion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 912–15 921.

[62] K. Joseph, P. Udhayanan, T. Shukla, A. Agarwal, S. Karanam, K. Goswami, and B. V. Srinivasan, "Iterative multi-granular image editing using diffusion models," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 8107–8116.

[63] D. Valevski, M. Kalman, E. Molad, E. Segalis, Y. Matias, and Y. Leviathan, "Unitune: Text-driven image editing by fine tuning a diffusion model on a single image," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–10, 2023.

[64] B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, and M. Irani, "Imagic: Text-based real image editing with diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6007–6017.

[65] H. Li, H. Chen, B. Li, and S. Tan, "Can forensic detectors identify gan generated images?" in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018, pp. 722–727.

[66] J. Chen, Y. Deng, G. Bai, and G. Su, "Face image quality assessment based on learning to rank," *IEEE signal processing letters*, vol. 22, no. 1, pp. 90–94, 2014.

[67] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[68] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[69] S. McCloskey and M. Albright, "Detecting gan-generated imagery using saturation cues," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 4584–4588.

[70] X. Yang, Y. Li, H. Qi, and S. Lyu, "Exposing gan-synthesized faces using landmark locations," in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, 2019, pp. 113–118.

[71] H. Mo, B. Chen, and W. Luo, "Fake faces identification via convolutional neural network," in *Proceedings of the 6th ACM workshop on information hiding and multimedia security*, 2018, pp. 43–47.

[72] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *International conference on machine learning*. PMLR, 2020, pp. 3247–3258.

[73] M. Tanaka and H. Kiya, "Fake-image detection with robust hashing," in *2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech)*. IEEE, 2021, pp. 40–43.

[74] G. Tang, L. Sun, X. Mao, S. Guo, H. Zhang, and X. Wang, "Detection of gan-synthesized image based on discrete wavelet transform," *Security and Communication Networks*, vol. 2021, pp. 1–10, 2021.

[75] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, "Detection of gan-generated fake images over social networks," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2018, pp. 384–389.

[76] C.-C. Hsu, Y.-X. Zhuang, and C.-Y. Lee, "Deep fake image detection based on pairwise learning," *Applied Sciences*, vol. 10, no. 1, p. 370, 2020.

[77] X. Xuan, B. Peng, W. Wang, and J. Dong, "On the generalization of gan image forensics," in *Chinese conference on biometric recognition*. Springer, 2019, pp. 134–141.

[78] N. Hulzebosch, S. Ibrahimi, and M. Worring, "Detecting cnn-generated facial images in real-world scenarios," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 642–643.

[79] L. Chai, D. Bau, S.-N. Lim, and P. Isola, "What makes fake images detectable? understanding properties that generalize," in *ECCV*. Springer, 2020, pp. 103–120.

[80] Y. Ju, S. Jia, L. Ke, H. Xue, K. Nagano, and S. Lyu, "Fusing global and local features for generalized ai-synthesized image detection," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 3465–3469.

[81] 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.

[82] 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)*. IEEE, 2019, pp. 1–6.

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

[84] H. Li, B. Li, S. Tan, and J. Huang, "Identification of deep network generated images using disparities in color components," *Signal Processing*, vol. 174, p. 107616, 2020.

[85] L. Nataraj, T. M. Mohammed, B. Manjunath, S. Chandrasekaran, A. Flenner, J. H. Bappy, and A. K. Roy-Chowdhury, "Detecting gan generated fake images using co-occurrence matrices," *Electronic Imaging*, vol. 2019, no. 5, pp. 532–1, 2019.

[86] M. Barni, K. Kallas, E. Nowroozi, and B. Tondi, "Cnn detection of gan-generated face images based on cross-band co-occurrences analysis," in *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2020, pp. 1–6.

[87] X. Zhang, S. Karaman, and S.-F. Chang, "Detecting and simulating artifacts in gan fake images," in *2019 IEEE international workshop on information forensics and security (WIFS)*. IEEE, 2019, pp. 1–6.

[88] B. M. Le and S. S. Woo, "Exploring the asynchronous of the frequency spectra of gan-generated facial images," *arXiv preprint arXiv:2112.08050*, 2021.

[89] M. Tanaka, S. Shiota, and H. Kiya, "A universal detector of cnn-generated images using properties of checkerboard artifacts in the frequency domain," in *2021 IEEE 10th Global Conference on Consumer Electronics (GCCE)*. IEEE, 2021, pp. 103–106.

[90] N. Bonettini, P. Bestagini, S. Milani, and S. Tubaro, "On the use of benford's law to detect gan-generated images," in *2020 25th international conference on pattern recognition (ICPR)*. IEEE, 2021, pp. 5495–5502.

[91] Y. Yu, R. Ni, and Y. Zhao, "Mining generalized features for detecting ai-manipulated fake faces," *arXiv preprint arXiv:2010.14129*, 2020.

[92] C. Tan, Y. Zhao, S. Wei, G. Gu, and Y. Wei, "Learning on gradients: Generalized artifacts representation for gan-generated images detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 105–12 114.

[93] C. Tan, Y. Zhao, S. Wei, G. Gu, P. Liu, and Y. Wei, "Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection," *arXiv preprint arXiv:2312.10461*, 2023.

[94] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. IEEE, 2019, pp. 83–92.

[95] S. Hu, Y. Li, and S. Lyu, "Exposing gan-generated faces using inconsistent corneal specular highlights," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 2500–2504.

[96] H. Guo, S. Hu, X. Wang, M.-C. Chang, and S. Lyu, "Eyes tell all: Irregular pupil shapes reveal gan-generated faces," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 2904–2908.

[97] Z. Chen and H. Yang, "Attentive semantic exploring for manipulated face detection," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 1985–1989.

[98] Z. Liu, X. Qi, and P. H. Torr, "Global texture enhancement for fake face detection in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8060–8069.

[99] J. Ricker, S. Damm, T. Holz, and A. Fischer, "Towards the detection of diffusion model deepfakes," *arXiv preprint arXiv:2210.14571*, 2022.

[100] 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)*. IEEE, 2023, pp. 1–5.

[101] D. C. Epstein, I. Jain, O. Wang, and R. Zhang, "Online detection of ai-generated images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 382–392.

[102] Q. Bammey, "Synthbuster: Towards detection of diffusion model generated images," *IEEE Open Journal of Signal Processing*, 2023.

[103] Z. Wang, J. Bao, W. Zhou, W. Wang, H. Hu, H. Chen, and H. Li, "Dire for diffusion-generated image detection," *arXiv preprint arXiv:2303.09295*, 2023.

[104] K. Songsri-in and S. Zafeiriou, "Complement face forensic detection and localization with faciallandmarks," *arXiv preprint arXiv:1910.05455*, 2019.

[105] Y. Huang, F. Juefei-Xu, Q. Guo, Y. Liu, and G. Pu, "Fakelocator: Robust localization of gan-based face manipulations," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2657–2672, 2022.

[106] G. Mazaheri and A. K. Roy-Chowdhury, "Detection and localization of facial expression manipulations," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 1035–1045.

[107] H. Siqueira, S. Magg, and S. Wermter, "Efficient facial feature learning with wide ensemble-based convolutional neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 5800–5809.

[108] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, "On the detection of digital face manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, 2020, pp. 5781–5790.

[109] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," *arXiv preprint arXiv:1906.06876*, 2019.

[110] 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*, 2020, pp. 5001–5010.

[111] T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong, and W. Xia, "Learning self-consistency for deepfake detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 023–15 033.

[112] 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*, 2021, pp. 2185–2194.

[113] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," in *International Conference on Learning Representations*, 2018.

[114] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: boundary equilibrium generative adversarial networks," *CoRR*, vol. abs/1703.10717, 2017.

[115] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8188–8197.

[116] G. Chen, L. Qiao, Y. Shi, P. Peng, J. Li, T. Huang, S. Pu, and Y. Tian, "Learning open set network with discriminative reciprocal points," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 507–522.

[117] Z. Xia, P. Wang, G. Dong, and H. Liu, "Adversarial kinetic prototype framework for open set recognition," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[118] T. Yang, D. Wang, F. Tang, X. Zhao, J. Cao, and S. Tang, "Progressive open space expansion for open-set model attribution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 856–15 865.

[119] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1875–1882.

[120] W. Guo, B. Tondi, and M. Barni, "A master key backdoor for universal impersonation attack against dnn-based face verification," *Pattern Recognition Letters*, vol. 144, pp. 61–67, 2021.

[121] P. Fang, J. Zhou, S. K. Roy, L. Petersson, and M. Harandi, "Bilinear attention networks for person retrieval," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 8030–8039.

[122] R. R. Varior, M. Haloi, and G. Wang, "Gated siamese convolutional neural network architecture for human re-identification," in *European conference on computer vision*. Springer, 2016, pp. 791–808.

[123] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5620–5629.

[124] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*. Springer, 2016, pp. 850–865.

[125] S. Krishnagopal, Y. Aloimonos, and M. Girvan, "Similarity learning and generalization with limited data: A reservoir computing approach," *Complexity*, vol. 2018, 2018.

[126] F. Zhou, Z. Jiang, C. Shui, B. Wang, and B. Chaib-draa, "Domain generalization via optimal transport with metric similarity learning," *Neurocomputing*, vol. 456, pp. 469–480, 2021.

[127] N. Fonseca and V. Guidetti, "Similarity and generalization: From noise to corruption," *arXiv preprint arXiv:2201.12803*, 2022.

[128] K. Roth, T. Milbich, S. Sinha, P. Gupta, B. Ommer, and J. P. Cohen, "Revisiting training strategies and generalization performance in deep metric learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8242–8252.

[129] R. Agarwal, M. C. Machado, P. S. Castro, and M. G. Bellemare, "Contrastive behavioral similarity embeddings for generalization in reinforcement learning," *arXiv preprint arXiv:2101.05265*, 2021.

[130] D. E. King, "Dlib-ml: A machine learning toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[131] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *CVPR*, 2017, pp. 1251–1258.

[132] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[133] M. Barni, A. Costanzo, E. Nowroozi, and B. Tondi, "Cnn-based detection of generic contrast adjustment with jpeg post-processing," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 3803–3807.

[134] S. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," *Neurocomputing*, vol. 378, pp. 112–119, 2020.

[135] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[136] 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*. Ieee, 2009, pp. 248–255.

[137] P. Singh, A. Manure, P. Singh, and A. Manure, "Introduction to tensorflow 2.0," *Learn TensorFlow 2.0: Implement Machine Learning and Deep Learning Models with Python*, pp. 1–24, 2020.

[138] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[139] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[140] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[141] R. M. Joseph and A. Chithra, "Literature survey on image manipulation detection," *International Research Journal of Engineering and Technology (IRJET)*, vol. 2, no. 04, 2015.

[142] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[143] S. Kawakami, K. Okada, N. Nitta, K. Nakamura, and N. Babaguchi, "Semi-supervised outdoor image generation conditioned on weather signals," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 4268–4275.

[144] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Learning to generate images of outdoor scenes from attributes and semantic layouts," *arXiv preprint arXiv:1612.00215*, 2016.

[145] T. Rothmeier and W. Huber, "Let it snow: On the synthesis of adverse weather image data," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 3300–3306.

[146] Y. Lin, Y. Li, H. Cui, and Z. Feng, "Weagan: Generative adversarial network for weather translation of image among multi-domain," in *2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC)*. IEEE, 2019, pp. 1–5.

[147] C. Sazara, M. Cetin, and K. M. Iftekharuddin, "Detecting floodwater on roadways from image data with handcrafted features and deep transfer learning," in *2019 IEEE intelligent transportation systems conference (ITSC)*. IEEE, 2019, pp. 804–809.

[148] M. Zaffaroni and C. Rossi, "Water segmentation with deep learning models for flood detection and monitoring," in *conference on Information Systems for Crisis Response and Management (ISCRAM 2020)*, 2020, pp. 24–27.

[149] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[150] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[151] O. Alamayreh and M. Barni, "Detection of gan-synthesized street videos," in *2021 29th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 811–815.

[152] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3523–3542, 2021.

[153] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *ICML*. PMLR, 2019, pp. 6105–6114.

[154] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1757–1772, 2013.

[155] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III 13*. Springer, 2014, pp. 393–409.

[156] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 11, pp. 2317–2324, 2014.

[157] R. L. Smith, "Extreme value theory," *Handbook of applicable mathematics*, vol. 7, no. 437-471, p. 18, 1990.

[158] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1563–1572.

[159] G. Gavarini, D. Stucchi, A. Ruospo, G. Boracchi, and E. Sanchez, "Open-set recognition: an inexpensive strategy to increase dnn reliability," in *2022 IEEE 28th International Symposium on On-Line Testing and Robust System Design (IOLTS)*. IEEE, 2022, pp. 1–7.

[160] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Transactions on information theory*, vol. 16, no. 1, pp. 41–46, 1970.

[161] S. Vaze, K. Han, A. Vedaldi, and A. Zisserman, "Open-set recognition: A good closed-set classifier is all you need?" in *International Conference on Learning Representations (ICLR)*, 2022.

[162] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," *Advances in neural information processing systems*, vol. 33, pp. 21 464–21 475, 2020.

[163] H.-M. Yang, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Robust classification with convolutional prototype learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3474–3482.

[164] D. Miller, N. Sunderhauf, M. Milford, and F. Dayoub, "Class anchor clustering: A loss for distance-based open set recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3570–3578.

[165] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura, "Classification-reconstruction learning for open-set recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4016–4025.

[166] P. Oza and V. M. Patel, "C2ae: Class conditioned auto-encoder for open-set recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2307–2316.

[167] X. Sun, Z. Yang, C. Zhang, K.-V. Ling, and G. Peng, "Conditional gaussian distribution learning for open set recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 480–13 489.

[168] H. Huang, Y. Wang, Q. Hu, and M.-M. Cheng, "Class-specific semantic reconstruction for open set recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 4, pp. 4214–4228, 2022.

[169] Z. Ge, S. Demyanov, Z. Chen, and R. Garnavi, "Generative openmax for multi-class open set classification," in *British Machine Vision Conference 2017*. British Machine Vision Association and Society for Pattern Recognition, 2017.

[170] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li, "Open set learning with counterfactual images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 613–628.

[171] S. Kong and D. Ramanan, "Opengan: Open-set recognition via open data generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 813–822.

[172] W. Moon, J. Park, H. S. Seong, C.-H. Cho, and J.-P. Heo, "Difficulty-aware simulator for open set recognition," in *European Conference on Computer Vision*. Springer, 2022, pp. 365–381.

[173] G. Chen, P. Peng, X. Wang, and Y. Tian, "Adversarial reciprocal points learning for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8065–8081, 2021.

[174] C. Kim, Y. Ren, and Y. Yang, "Decentralized attribution of generative models," *arXiv preprint arXiv:2010.13974*, 2020.

[175] N. Yu, V. Skripniuk, D. Chen, L. Davis, and M. Fritz, "Responsible disclosure of generative models using scalable fingerprinting," *arXiv preprint arXiv:2012.08726*, 2020.

[176] N. Yu, V. Skripniuk, S. Abdelnabi, and M. Fritz, "Artificial fingerprinting for generative models: Rooting deepfake attribution in training data," in *Proceedings of the IEEE/CVF International conference on computer vision*, 2021, pp. 14 448–14 457.

[177] 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)*. IEEE, 2019, pp. 506–511.

[178] 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*, 2019, pp. 7556–7566.

[179] M. Joslin and S. Hao, "Attributing and detecting fake images generated by known gans," in *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2020, pp. 8–14.

[180] T. Yang, J. Cao, Q. Sheng, L. Li, J. Ji, X. Li, and S. Tang, "Learning to disentangle gan fingerprint for fake image attribution," *arXiv preprint arXiv:2106.08749*, 2021.

[181] X. Xuan, B. Peng, W. Wang, and J. Dong, "Scalable fine-grained generated image classification based on deep metric learning," *arXiv preprint arXiv:1912.11082*, 2019.

[182] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *International conference on machine learning*. PMLR, 2020, pp. 3247–3258.

[183] T. Yang, Z. Huang, J. Cao, L. Li, and X. Li, "Deepfake network architecture attribution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 4, 2022, pp. 4662–4670.

[184] T. Bui, N. Yu, and J. Collomosse, "Repmix: Representation mixing for robust attribution of synthesized images," in *European Conference on Computer Vision*. Springer, 2022, pp. 146–163.

[185] S. Fang, T. D. Nguyen, and M. C. Stamm, "Open set synthetic image source attribution," *arXiv preprint arXiv:2308.11557*, 2023.

[186] T. Yang, J. Cao, D. Wang, and C. Xu, "Fingerprints of generative models in the frequency domain," *arXiv preprint arXiv:2307.15977*, 2023.

[187] R. Al-Dayil, Y. Bazi, and N. Alajlan, "Open-set classification in remote sensing imagery with energy-based vision transformer," in *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2022, pp. 2211–2214.

[188] S. Fort, J. Ren, and B. Lakshminarayanan, "Exploring the limits of out-of-distribution detection," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7068–7081, 2021.

[189] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 325–341.

[190] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR (2)*. IEEE Computer Society, 2006, pp. 1735–1742.

[191] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *International Conference on Learning Representations*, 2016.

[192] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," *CoRR*, vol. abs/2006.06676, 2020. [Online]. Available: https://arxiv.org/abs/2006.06676

[193] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[194] S. Li, M. Xue, B. Z. H. Zhao, H. Zhu, and X. Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2088–2105, 2020.

[195] H. Zhong, C. Liao, A. C. Squicciarini, S. Zhu, and D. Miller, "Backdoor embedding in convolutional neural network models via invisible perturbation," in *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, 2020, pp. 97–108.

[196] T. Wang, Y. Yao, F. Xu, S. An, H. Tong, and T. Wang, "An invisible black-box backdoor attack through frequency domain," in *European Conference on Computer Vision*. Springer, 2022, pp. 396–413.

[197] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible backdoor attack with sample-specific triggers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 463–16 472.

[198] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*. Springer, 2020, pp. 182–199.

[199] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "Abs: Scanning neural networks for back-doors by artificial brain stimulation," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1265–1282.

[200] S. Cheng, Y. Liu, S. Ma, and X. Zhang, "Deep feature space trojan attack of neural networks by controlled detoxification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1148–1156.

[201] A. Nguyen and A. Tran, "Wanet–imperceptible warping-based backdoor attack," *arXiv preprint arXiv:2102.10369*, 2021.

[202] F. Qi, Y. Chen, M. Li, Y. Yao, Z. Liu, and M. Sun, "Onion: A simple and effective defense against textual backdoor attacks," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 9558–9566.

[203] W. Chen, B. Wu, and H. Wang, "Effective backdoor defense by exploiting sensitivity of poisoned samples," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9727–9737, 2022.

[204] Z. Zhang, Q. Liu, Z. Wang, Z. Lu, and Q. Hu, "Backdoor defense via deconfounded representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 228–12 238.

[205] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[206] D. Busbridge, J. Ramapuram, P. Ablin, T. Likhomanenko, E. G. Dhekane, X. Suau Cuadros, and R. Webb, "How to scale your ema," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[207] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: learning lip sync from audio," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.

[208] 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*, 2021, pp. 5039–5049.

[209] 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*, 2021, pp. 3348–3357.

[210] 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*, 2021, pp. 923–932.

[211] P. Dogoulis, G. Kordopatis-Zilos, I. Kompatsiaris, and S. Papadopoulos, "Improving synthetically generated image detection in cross-concept settings," in *Proceedings of the 2nd ACM International Workshop on Multimedia AI against Disinformation*, 2023, pp. 28–35.

[212] U. Ojha, Y. Li, and Y. J. Lee, "Towards universal fake image detectors that generalize across generative models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24 480–24 489.

[213] X. Bi, B. Liu, F. Yang, B. Xiao, W. Li, G. Huang, and P. C. Cosman, "Detecting generated images by real images only," *arXiv preprint arXiv:2311.00962*, 2023.

[214] R. Shao, T. Wu, and Z. Liu, "Detecting and grounding multi-modal media manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6904–6913.

[215] H. Khalid, S. Tariq, M. Kim, and S. S. Woo, "Fakeavceleb: A novel audio-video multimodal deepfake dataset," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

The proliferation of generative AI techniques for image generation and editing presents a multitude of challenges, including the dissemination of fraudulent content, the propagation of disinformation, and the erosion of public trust in digital media. While before the onset of this thesis, AI-powered systems demonstrated effectiveness in authenticating images within controlled environments. They exhibited limited capability in addressing the complexities of real-world applications.

This thesis responds to these challenges by contributing to the development of forensic systems capable of operating effectively in uncontrolled environments, commonly referred to as ”in the wild”. The initial focus is on tackling the dataset mismatch problem, wherein test samples undergo post-processing pipelines or generated by new generative AI tools distinct from those encountered during system training. We introduce a Siamese network for detecting AI-synthetic images and a hybrid architecture, enhancing generalization and robustness against image processing operations. In the second part, we focus on the open-set scenario, devising solutions for synthetic image attribution and facial attribute classification. We develop classifiers with a rejection option, employing hybrid architectures and novel frameworks alongside a verification approach leveraging contrastive learning. These contributions fortify image authentication in uncontrolled environments, mitigating risks of fraud and disinformation.

The PhD School of Information Engineering and Science of the University of Siena aims at providing future researchers, in both academic and industrial environments, with the background and methodological skills required to promote and deal with scientific and technological innovations in the wide area of information science and technology. This goal is achieved with the help of a highly qualified PhD board and with the availability of well equipped laboratories and research facilities offered by the Department of Information Engineering and Mathematics hosting the PhD school.