

**ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA**

**DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING**

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

Machine Learning and Computer Vision

**AN EXPERIMENTAL STUDY ON
GENERALIZATION IN DEEP FAKE IMAGE
DETECTION**

CANDIDATE

Davide Brescia

SUPERVISOR

Prof. Samuele Salti

CO-SUPERVISORS

Ing. Joris Mollinga

Ing. Dimitris Ieronymakis

Academic year 2022-2023

Session 1st

Dedicato a tutta la mia famiglia

Dedicato a Daniele, Giacomo, Giuseppe e Iulian,
con i quali presto avvieremo una startup.

Contents

1	Introduction	1
1.1	Deepfakes identification challenges	2
1.2	Deepfakes Generalization	3
1.3	History of Deepfakes	5
1.4	Structure of the Thesis	7
2	Literature Review	8
3	Methodology	11
3.1	General Implementation	12
3.2	Data	13
3.3	UCF Paper	16
3.3.1	Implementation	20
3.4	SBI Paper	21
3.4.1	Implementation	23
3.5	Locate and Verify Paper	24
3.5.1	Implementation	26
3.6	SPSL Paper	27
3.6.1	Implementation	29
3.7	Evaluation	29
3.8	Experimental Setup and Code Development	30
4	Experiment Results	31

4.1	UFC Paper	31
4.2	SBI Paper	38
4.3	Locate and Verify Paper	42
4.4	SPSL Paper	44
4.5	Cross Evaluation	50
5	Discussion	52
5.1	Future Works	54
6	Conclusion	55
	Bibliography	57
	Acknowledgements	65

List of Figures

1.1	A real image and a fake image	1
3.1	Representation of the architecture of the UCF model reported within the paper	17
3.2	t-SNE visualization of UCF Model	20
3.3	Representation of the architecture of the SBI model reported within the paper.	23
3.4	Some images generated with the SBI method	23
3.5	Sample images showcasing associated landmarks.	24
3.6	Representation of the architecture of the Locate and Verify model reported within the paper.	25
3.7	Input of the SBI Model.	27
3.8	Representation of the architecture of the SPSL model reported within the paper.	28
4.1	Losses and metrics observed during the initial training with the UCF model on FaceForensics dataset	32
4.2	Comparison of Experiments with Different Learning Rates in the UCF Model	34
4.3	Losses, Accuracies, and AUC during the initial training with the SBI model on FaceForensics dataset	40
4.4	Losses and metrics observed during the initial training with the SPSL model on FaceForensics dataset	46

List of Tables

3.1	Summary of all the Datasets used	15
4.1	Reproduction of Results for UCF Model	33
4.2	Seed Consistency in UCF Model	33
4.3	Comparison of Xception and EfficientNetB5 backbones in the UCF Model	34
4.4	Comparison of Experiments with and without Data Augmen- tation in the UCF Model trained on FaceForensics	35
4.5	Comparison with or without Dropout in the UCF Model	35
4.6	Performances of the UFC paper trained on FaceForensics	37
4.7	Performances of the UFC paper trained on a combination of different Datasets	39
4.8	Performance Evaluation of Provided Pretrained Weights on SBI using different resolutions	40
4.9	Performances of the SBI model trained on FaceForensics	41
4.10	Performances of the SBI model trained on a combination of different Datasets	43
4.11	Reproduction of Results for LAV Model	44
4.12	Performances of the Locate and Verify model trained on Face- Forensics	45
4.13	Performances of the SPSL paper trained on FaceForensics	47
4.14	Comparison of SPSL with and without transformation	48

4.15 Performances of the SPSL model trained on a combination of different Datasets	49
4.16 Models Comparisons Out-of-Distribution using Accuracy . . .	51

Chapter 1

Introduction

Deepfake technology has emerged as a transformative force in the digital media landscape, fundamentally reshaping the creation and manipulation of audiovisual content. Crafted through advanced machine learning techniques, particularly deep neural networks, deepfakes have the capacity to convincingly depict individuals engaging in actions or articulating statements that never occurred, challenging the authenticity and reliability of both visual and audio content.



(a) Example of a real image within the FaceForensics dataset.



(b) Example of a real image within the FaceForensics dataset manipulated using the SBI technique.

Figure 1.1: Difference between a real (a) and fake (b) image

These synthetic media productions manifest in various forms, ranging from

face swaps and voice synthesis to intricate scenarios involving the generation of entirely fabricated scenes or events. This diversity underscores the adaptability and versatility of deepfake applications, presenting both promising opportunities and significant concerns.

On the positive side, deepfakes can be harnessed for benign purposes such as entertainment, creative expression, and educational simulations, offering innovative avenues for filmmakers, content creators, and educators to explore fictional narratives or historical events in a visually compelling manner. Additionally, deepfake technology facilitates dubbing and language localization, enhancing accessibility and cultural relevance in global media.

Conversely, the dual nature of deepfakes raises ethical and security concerns, as malicious actors can exploit this technology for deceptive purposes, creating misleading content for disinformation campaigns, impersonation, or cyber threats. The potential misuse of deepfakes poses risks to public trust, information integrity, and personal privacy.

1.1 Deepfakes identification challenges

Identifying deepfakes is of paramount importance due to the potential consequences associated with their deceptive nature. Robust deepfake detection is essential to preserve the integrity of information, safeguard public trust, and mitigate the risks posed by malicious misuse of this technology.

One primary challenge in identifying deepfakes lies in the constant evolution of the underlying techniques employed in their creation. As deepfake generators become more sophisticated, adapting to new methodologies and countering detection mechanisms, traditional approaches may struggle to keep pace. This dynamic landscape requires continual advancements in detection methods to discern manipulated content effectively across a diverse range of scenarios.

Furthermore, the sheer volume and variety of digital media circulating online amplify the difficulty of detection. The scale at which content is produced and shared makes manual verification impractical, necessitating the development of automated systems that can efficiently and accurately identify deepfakes. The challenge is exacerbated by the need for detection models to generalize their understanding of manipulations, accommodating emerging techniques and variations.

Moreover, the ethical considerations surrounding deepfake detection introduce another layer of complexity. Striking a balance between privacy preservation and the identification of potentially harmful content requires careful navigation. Implementing detection mechanisms that respect individuals' rights while effectively identifying malicious uses of deepfake technology is an ongoing challenge in this evolving landscape.

1.2 Deepfakes Generalization

This thesis centers on the pivotal concept of generalization in the context of identifying deepfakes, with a specific focus on images. Generalization, denoting a model or system's ability to accurately perform on unseen or novel data, is indispensable for the efficacy of deepfake detection algorithms across a diverse spectrum of manipulated media. This adaptability becomes increasingly crucial as deepfake techniques continue to evolve and diversify. The primary objective is to comprehend and enhance the generalization capabilities of deepfake identification systems, thereby reinforcing their resilience against emerging manipulation methods and contributing to the robustness of media authentication mechanisms.

In the realm of machine learning, generalization is defined as a model's proficiency in performing well on data beyond its training examples. For deepfake detection, achieving generalization is a critical aspect underpinning the effectiveness of detection systems. A successful generalization enables a

deepfake detection model to accurately identify manipulated content across varied scenarios, ensuring reliable performance in real-world applications.

The primary challenge associated with a system lacking generalization lies in its limited adaptability to novel and unseen deepfake variations. Such a system may excel in identifying specific manipulations within the training set but falters when confronted with new techniques or unforeseen variations. This lack of adaptability exposes the system to evasion by malicious actors who continually innovate and refine their deepfake generation methods.

Consequently, the overarching problem of an inability to generalize poses a potential compromise to the security and integrity of digital platforms. A non-generalizing deepfake detection system may proficiently identify deepfakes created with known methods but could be blind to emerging threats. This vulnerability puts users at risk of misinformation, privacy invasion, and other malicious activities facilitated by advanced deepfake techniques.

On the flip side, achieving generalization in deepfake detection is a complex task, primarily due to the dynamic and ever-evolving nature of deepfake creation methods. Generalization requires the development of models capable of discerning patterns and features inherent to manipulations across diverse datasets. This demands a nuanced understanding of the underlying principles of various deepfake techniques, enabling the model to generalize its knowledge to new and unseen manipulations.

Furthermore, the challenge is compounded by the necessity to balance specificity and sensitivity. A model that generalizes too broadly may yield false positives or false negatives, diminishing its practical utility. Striking the right balance to achieve generalization without sacrificing precision is a delicate task in the development of effective deepfake detection systems.

Numerous studies have examined the pros and cons of deepfakes and their future developments. One such study is the work of B.U. Mahmud et al. [29], which provides a detailed analysis.

1.3 History of Deepfakes

The roots of deepfakes can be traced back to the 1990s, an era when pioneering researchers delved into the realm of computer-generated imagery (CGI) to craft increasingly realistic digital personas. During this period, the exploration revolved around pushing the boundaries of technology to develop life-like characters within the digital landscape.

The history of deepfakes intertwines with the evolution of artificial intelligence (AI) and machine learning. Pioneering research, such as Ian Goodfellow's introduction of GANs [12] in 2014, marked a watershed moment in the development of deep learning models capable of generating highly realistic synthetic content. GANs, comprising a generator and a discriminator network engaged in a competitive learning process, enabled the creation of deepfake videos by synthesizing images or videos that convincingly mimic real human appearances and behaviors.

In 2016, Face2Face emerged, a real-time facial reenactment system, showcasing the ability to manipulate a target actor's facial expressions in a video in real-time [44]. Developed by researchers from Stanford, the Max Planck Institute for Informatics, and the University of Erlangen-Nuremberg, Face2Face employed a combination of facial tracking and reenactment techniques. It used a standard webcam to capture a source actor's facial expressions and, through specialized algorithms, mapped these expressions onto a target actor's face in a separate video. This manipulation occurred in real-time, allowing for the instantaneous replication of expressions, mouth movements, and other facial gestures onto the target actor's video footage. This technology gained attention for its capacity to alter videos by swapping facial expressions, a capability that raised both fascination and concerns about the potential for misuse in manipulating visual content. The development of Face2Face marked a significant leap in real-time facial manipulation technology, introducing a new frontier in the evolution of visual media alteration.

The term "deepfake" itself originates from a Reddit user named "deep-fakes" in 2017, who popularized the technique by superimposing celebrities' faces onto pornographic videos in late 2017, subsequently leading to widespread attention and concern about the potential misuse of this technology [8].

To address the rising apprehensions surrounding deepfake proliferation, dedicated efforts from both researchers and technology companies have been underway to create robust detection mechanisms. A pivotal step in this pursuit came in 2018 when scholars at the University of California, Berkeley, introduced "FaceForensics", a groundbreaking method leveraging machine learning [37]. This technique scrutinizes facial expressions and head movements, meticulously identifying disparities to flag manipulated content. Parallely, in 2019, Google took a significant stride by releasing a comprehensive dataset of deepfake videos [36]. This resource aimed to aid researchers in refining detection algorithms and fortifying defense mechanisms against the evolving sophistication of deepfake technology. However, despite these proactive measures, the relentless advancement of deepfake capabilities persists, posing an escalating challenge in the realm of detection and necessitating continuous innovation to combat its escalating threat.

In recent years, the distinction between a deepfake and a real face has become increasingly challenging, reaching a level of complexity that raises concerns. Moreover, the accessibility to artificial intelligence, particularly generative models, has surged, facilitating the rapid dissemination of such deceptive artifacts. Recognizing this evolving landscape is crucial, as it compels us to be discerning consumers of digital content, acknowledging the potential for manipulated visuals and understanding that not everything presented to us should be taken at face value. As we navigate this era of advanced technology, awareness becomes a powerful defense against the pervasive influence of deepfakes, prompting a critical reassessment of our perceptions in the digital realm.

1.4 Structure of the Thesis

Chapter 1, **Introduction**, provides an overview of the research topic, objectives, and the scope of the study.

Chapter 2, **Literature Review**, explores related papers on deepfake detection and its generalization.

Chapter 3, **Methodology**, discusses the implementation of studied papers during the internship and outlines the experimental approach.

Chapter 4, **Experiment Results**, presents a detailed view of the results obtained from various methods, followed by a comparative analysis.

Chapter 5, **Discussion**, offers a brief discussion of the results, highlighting strengths, weaknesses, implications of the methods employed and debates about potential avenues for future research and improvement.

Chapter 6, **Conclusion**, summarizes the findings of the study.

Chapter 2

Literature Review

One early foray into deepfake detection involved the development of Convolutional Neural Network (CNN) architectures. This initial method failed to account for important details in the frequency domain, such as compression artifacts or specific noise patterns. In the field of video forgery detection, the work by Afchar et al. [1] presents a method for automatically and efficiently detecting face tampering in videos, with a focus on Deepfake and Face2Face techniques. Cozzolino et al. [36], together with the FaceForensics dataset, employed the Xception architecture, a deep learning model based on depthwise separable convolutions, for training their forgery detection pipeline. Nguyen et al. [32] introduces a novel approach to detect forged images and videos using capsule networks. Du et al. [10] proposed a Locality-Aware AutoEncoder (LAE) that focuses on the forgery regions to make accurate predictions. Huy H. Nguyen et al. [31] introduces a convolutional neural network that simultaneously detects manipulated media and locates the manipulated regions using segmentation masks.

Nevertheless, the existing methodologies predominantly concentrated on recognizing macroscopic features. Consequently, a shift towards high-frequency-oriented approaches was pursued. Qian et al. [34] introduced the Frequency in Face Forgery Network (F3-Net), a system that exploits frequency-aware cues to identify forgery patterns. Building upon the frequency representation,

Frank et al. [11] delved into artifacts present in GAN-generated images and proposed a method for detecting deep fake images. Li et al. [20] proposed a frequency-aware discriminative feature learning framework that combines metric learning and adaptive frequency feature generation. Gu et al. [13] proposes a progressive enhancement learning framework that extracts fine-grained forgery traces by exploiting both RGB and fine-grained frequency clues. Luo et al. [28] utilizes high-frequency noise features to overcome the limitations of current CNN-based detectors that are biased towards specific textures.

Some methods focus more on identifying deepfakes by placing greater emphasis on specific areas. Nguyen et al. [31] proposed a multi-task learning approach for detecting and segmenting manipulated facial images and videos. Wang et al. [46] proposed FakeSpotter, a robust approach for detecting AI-synthesized fake faces, based on monitoring neuron behaviors. Amerini et al. [4] developed a new technique using optical flow fields to discern between fake and original video sequences. Sun et al. [42] introduced LRNet, a lightweight and robust framework that uses temporal modeling on precise geometric features. Zhu et al. [52] proposed a novel approach by decomposing face images into 3D shape, common texture, identity texture, ambient light, and direct light, and utilizing facial detail as a clue to detect subtle forgery patterns. Li et al. [21] introduced Face X-ray, a forgery detection model that stands out by emphasizing blending boundaries in manipulated face images. Haliassos et al. [15] proposes a novel approach called LipForensics that targets high-level semantic irregularities in mouth movements. Zhao et al. [51] proposes a method that measures patch-wise similarities of input images and focusing on the inconsistency of source features within the forged images. Li et al. [22] suggest a technique that utilizes variations in resolution between manipulated faces and backgrounds to identify deepfake content.

Several methodologies leverage attention mechanisms for digital face manipulation detection. Dang et al. [25] utilized attention to improve feature

maps and manipulated region visualization, Zhao et al. [50] proposed a multi-attentional approach, Wang et al. [45] introduced a Multi-modal Multi-scale Transformer (M2TR) for detecting subtle artifacts at different spatial levels and in the frequency domain. Cao et al. [5], who developed an end-to-end reconstruction-classification learning framework.

Another technique widely used for detecting deepfakes is that of disentanglement, it refers to the separation of different factors or components within a given data representation. An example of this concept can be found in the paper written by Liang et al. [24].

Chapter 3

Methodology

In this chapter it is presented a comprehensive outline of the approach employed in this study, encompassing various sections detailing the implementation process, dataset selection, and experimental design.

In the section titled **General Implementation** (Section 3.1), a systematic approach to implementing selected papers is delineated. This entails a thorough examination of each paper's methodology, replication of results using relevant datasets, and adjustments for integration with proprietary datasets.

The subsequent section, **Data** (Section 3.2), delves into the selection and categorization of datasets, considering both public and private sources. Additionally, this section elaborates on dataset partitioning and preprocessing efforts to ensure standardized experimental conditions.

Sections 3.3, 3.4, 3.5, and 3.6 provide detailed information and implementations associated with the individual papers mentioned, specifically: **UCF Paper**, **SBI Paper**, **Locate and Verify Paper**, and **SPSL Paper**.

An overview of the **Evaluation** (Section 3.7) is then presented, highlighting the metrics used for evaluation and considerations for dataset imbalance in

calculating AUC.

Lastly, the section titled **Experimental Setup and Code Development** (Section 3.8) summarizes the experimental setup, including hardware specifications, software tools, and programming languages, to provide a clear understanding of the computational environment utilized throughout the experiments.

3.1 General Implementation

In implementing the selected papers, I adopted a systematic approach characterized by multiple stages:

1. Initially, I conducted a comprehensive study of each paper to ensure a thorough understanding methodologies. This involved spending several days delving into both the code and the written material.
2. In cases where necessary, I replicated results by obtaining the relevant datasets and adhering to the provided code guidelines. This step was skipped when dealing with less complex repositories or minimal required modifications.
3. Subsequently, assuming correct replication of results, I implemented all necessary changes. A primary modification involved adapting dataset retrieval from various repositories to fit company datasets within the workstation. Additionally, I often made adjustments such as sampling or implementing mixed precision.
4. Typically, I conducted an initial experiment using FaceForensics as a training set, followed by model evaluation on various datasets representing out-of-distribution domains. This preliminary step provided valuable insights into the model's performance, ensuring consistency with

previously replicated results and identifying any potential errors in previous implementations. FaceForensics is typically selected due to its frequent use as a benchmark dataset in research papers.

5. Finally, for each paper, I tested different parameters and dataset combinations to assess whether inclusion of diverse datasets led to improvements in generalization. The results of these experiments are reported in Chapter 4.

It's important to highlight that the selection of papers was a collaborative decision made by the company, aligning with our objectives in deepfake detection and generalization.

3.2 Data

In this study, we meticulously selected a diverse array of datasets to address different scenarios and challenges in deepfake detection. We categorized the datasets based on their public or private status, content, and, if applicable, the specific algorithm or deep learning model used to generate fake images. A summary of the key features of these datasets can be found in Table 3.1.

A detailed list of how the various algorithms or models associated with datasets containing fake images work will be shown below. It's important to note that while this compilation isn't exhaustive and encompasses only a restricted subset, it offers valuable insights into the algorithms employed.

- **Face Morph:** is a process of blending or transitioning between two facial images to create a smooth transformation effect, often used in animations and visual effects.
- **Face Swap:** is a computational technique used to swap faces between different images or videos, typically involving the replacement of one person's face with another while preserving the facial expressions and movements.

- **Face Synthesis:** involves the creation of artificial facial images or videos using computational techniques, often leveraging deep learning models to generate realistic-looking faces.
- **GAN (Generative Adversarial Network)**[12]: is a machine learning framework consisting of two neural networks, a generator and a discriminator, trained adversarially to generate realistic data samples, such as images or videos.
- **GFPGAN**[47]: refers to a Generative Facial Prior GAN, a type of generative adversarial network (GAN) designed specifically for parsing facial features and generating high-quality images.
- **Lip Sync:** short for lip synchronization, is the process of matching the lip movements of a digital character or avatar with spoken audio or text, typically achieved through algorithms or manual animation techniques.
- **Stable Diffusion**[35]: is a method for generating high-quality images by gradually adding noise to an initial image and iteratively denoising it, resulting in visually appealing outputs with fine details.
- **Talking Head:** refers to a synthesized video or animation of a human head speaking or lip-syncing to audio or text input, commonly used in applications like virtual assistants, avatars, and deepfakes.
- **Unstable Diffusion**[9]: is a technique that involves adding random noise to an image and gradually reducing its intensity, aiming to generate diverse and creative visual outputs, although with less predictable results compared to stable diffusion.

It is worth emphasizing that the proposed dataset encompasses various characteristics such as different resolutions, contexts, contrasts, brightness, facial shapes, etc. This is a crucial element to enable a model to generalize and represent (albeit approximately) real-world data. However, on the other hand, it

Datasets Informations		
Name	Type	Algorithm
blendswap-swapped	Fake	Face Swap
CelebA[27]	Real	-
CelebA-GFPGAN	Fake	GFPGAN
CelebDF[23]	Fake+Real	Face Swap
Cheap-Morphs	Fake	Face Morph
DeeperForensics[17]	Fake+Real	Face Swap
DFDM[16]	Fake	FaceSwap
faceapp-faceswap	Fake	Face Swap
faceapp-morph	Fake	Face Morph
FaceForensics/actors[36]	Real	-
FaceForensics/Deepfakes[36]	Fake	Face Swap
FaceForensics/Face2Face[36]	Fake	Talking Head
FaceForensics/FaceSwap[36]	Fake	Face Swap
FaceForensics/NeuralTextures[36]	Fake	Face Swap
FaceForensics/youtube[36]	Real	-
FFHQ[18]	Real	-
FFHQ-GFPGAN	Fake	GAN
FRLM-Morphs[38][39]	Fake	Face Morph
iFakeFaceDB[30]	Fake	GAN
insightface-swapped	Fake	Face Swap
LRS3[3]	Real	-
MegaFS[53]	Fake	Face Swap
Ms-Celeb-1M[14]	Real	-
reface	Fake	Face Swap
simsmap-swapped	Fake	Face Swap
stable-diffusion	Fake	Stable Diffusion
synthesis-generated	Fake	Face Synthesis
TPDNE	Fake	GAN
tedx[2]	Real	-
unstable-diffusion	Fake	Unstable Diffusion
VGGFace[6]	Real	-
Wav2lip[33]	Fake	Lip Sync

Table 3.1: Summary of all the Datasets used: Types (Real or Fake images) and Algorithms Used. Datasets cited are public; otherwise, they are private.

adds a layer of complexity as the abundant and diverse data make model training very challenging. Furthermore, it’s worth noting that certain real datasets include samples extracted from video datasets. Consequently, the range of facial expressions available is limited, and the videos often follow a TV setup

(e.g., FaceForensics and TEDx). This aspect should be taken into consideration. Notably, since a significant portion of the data, both public and private, was obtained from deepfake generation applications and algorithms, the type of algorithms often varies greatly among them, and they range from algorithms that are not very recent to algorithms that are extremely new and very difficult to identify. This may make some datasets easier to identify than others.

Within our company, datasets are represented in well-organized image structures. a unique procedure for performing preprocessing in a public dataset has been identified. For example, FaceForensics is a collection of videos, the preprocessing performed consists of extracting frames from the videos with a given sampling rate followed face identification with subsequent image cropping.

In terms of dataset partitioning, the majority of publicly available datasets were already divided into training and test sets, facilitating a streamlined integration into the experimental framework. However, in cases where such divisions were not predefined, a standard practice of allocating approximately 30% of the data as the test set was followed to maintain consistency across experiments.

3.3 UCF Paper

The authors of "UCF: Uncovering Common Features for Generalizable Deepfake Detection" [48] propose a novel disentanglement framework that uncovers common forgery features by decomposing image information into forgery-irrelevant, method-specific forgery, and common forgery features. They employ a multi-task learning strategy, a conditional decoder, and a contrastive regularization technique to enhance the disentanglement process. The code of this paper derived from the paper "DeepfakeBench: A Comprehensive Benchmark of Deepfake Detection" [49].

The authors of the paper noticed the tendency of detectors to focus excessively

on irrelevant content information and the lack of generalization due to overfitting to specific forgery technologies. To solve this problem they created the architecture showed in Figure 3.1.

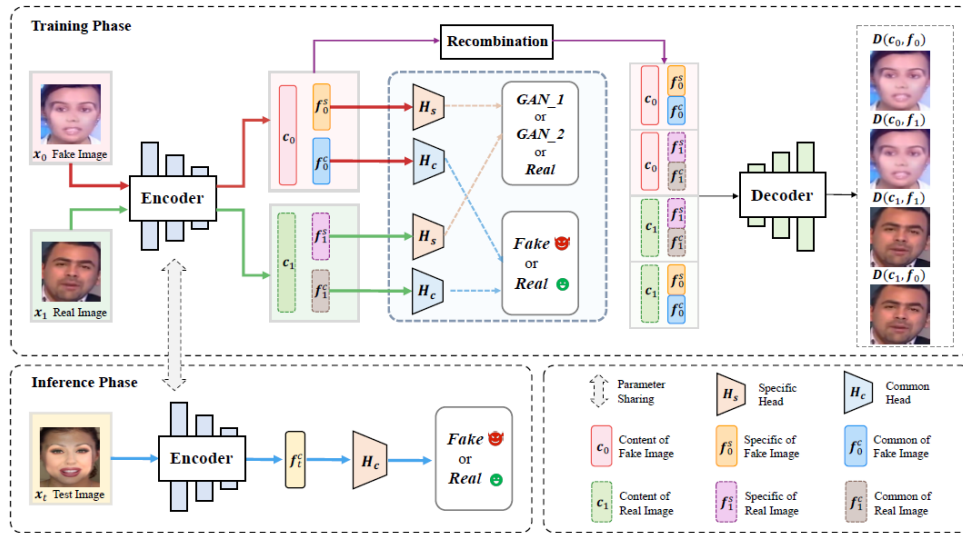


Figure 3.1: Representation of the architecture of the UCF model reported within the paper

Specifically, within the encoder, a pair of images, one real and the other fake, is processed. The encoder then produces three distinct types of outputs for each image: one representing specific features, another for common features, and the last one representing content features.

- **Specific features:** f_0^s for a fake image and f_1^s for a real image, are used to identify the type of deepfake technique and distinguish them from real images. They focus on capturing the unique characteristics of each type of fake dataset.
- **Common features:** f_0^c for a fake image and f_1^c for a real image, are employed to differentiate fake images from real ones. The aim is to avoid capturing patterns specific to a single deepfake domain and instead capture patterns that are consistent across various deepfakes encountered by the model.

- **Content features:** c_0 for a fake image and c_1 for a real image, are generated to preserve the facial structure and other content-related aspects of the image.

These features are re-used to reconstruct faces, combining all previously mentioned features to produce four distinct images using a decoder:

- **Recombined Image 1** - $D(c_0, f_0)$: fake image Content features combined with fake Specific and Common features.
- **Recombined Image 2** - $D(c_0, f_1)$: fake image Content features combined with real Specific and Common features.
- **Recombined Image 3** - $D(c_1, f_1)$: real image Content features combined with real Specific and Common features.
- **Recombined Image 3** - $D(c_1, f_0)$: real image Content features combined with fake Specific and Common features.

To fully understand the complexity of the model, it's helpful to delve into its losses.

- There are two **Classification losses**, L_{ce}^c for Common features and L_{ce}^s for Specific features, given by the cross-entropy loss. Specifically, after classification, the result is compared with the corresponding label of each features.
- The **Contrastive loss**, L_{con} , evaluates the proximity of common features between two images. To be more specific, if two images with different labels are close to each other in the latent space, this loss will be high; if they are far apart, it will be low. Similarly, for two images with the same label, a low loss indicates closeness in the latent space, while a high loss indicates distance.

- There are two **Reconstruction losses**, L_{rec}^s for Self-Reconstruction and L_{rec}^c for Cross-Reconstruction. In self-reconstruction, we pair the Common and Specific features generated from an image with the Common features derived from the same image. In cross-reconstruction, we match the fake Common and Specific features generated from one image with the real Common features generated from another image, and vice versa.

$$L_{rec}^s = \|x_0 - D(f_0, c_0)\|_1 + \|x_1 - D(f_1, c_1)\|_1$$

$$L_{rec}^c = \|x_0 - D(f_1, c_0)\|_1 + \|x_1 - D(f_0, c_1)\|_1$$

The final reconstruction loss is the sum of the two reconstruction losses obtained from the previous operation.

$$L_{rec} = L_{rec}^s + L_{rec}^c$$

The final loss is the sum of these losses, each multiplied by empirical constants determined by the paper's author.

$$L = L_{ce}^c + \lambda_1 L_{ce}^s + \lambda_2 L_{rec} + \lambda_3 L_{con}$$

To be more specific, the parameters selected by the authors are as follows:

$$\lambda_1 = 0.1, \quad \lambda_2 = 0.3, \quad \lambda_3 = 0.05$$

In essence, what's happening here is a disentanglement of features. The model is tasked with distinguishing between features specific to various deepfake techniques and between real and fake images. This distinction is evident in the k-SNE visualization provided in Figure 3.2 of their paper. This approach deviates from traditional detection methods, as illustrated by the Xception image shown to the left of the preceding figure. Notably, the Xception model can

differentiate between different deepfake techniques but struggles to do so distinctly in the feature space, unlike the UCF model.

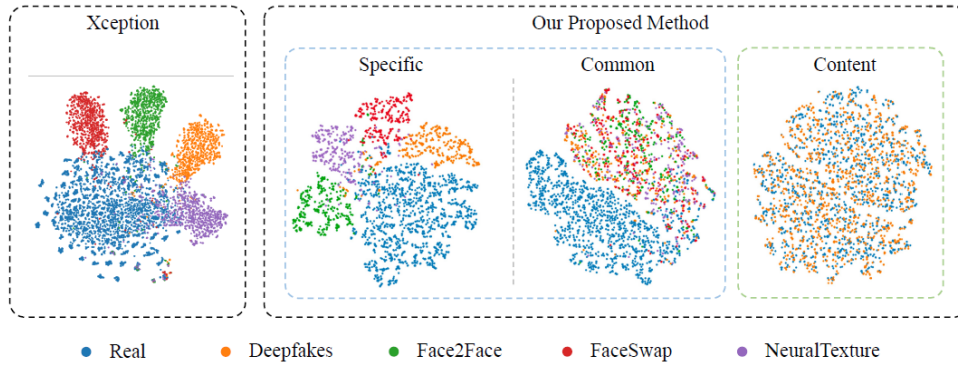


Figure 3.2: The t-SNE visualization of features extracted from the baseline Xception in the UCF framework trained on FaceForensics. We can see the different separations that can be obtained by considering different outputs within the model itself.

It's important to note that despite this architecture appearing quite complex, during the inference period, only the encoder is taken into account, along with only the common features, which are consequently classified as either fake or real. This leads to a lengthy training process but faster speeds during inference.

3.3.1 Implementation

To explain the implementation process in this paper, I first carefully studied and understood the methodology described in the research paper. Then, I replicated the experiments to validate their results, following the preprocessing steps provided. This involved cloning the repository, setting up a Conda environment, and installing required libraries. Next, I preprocessed the dataset they used (FaceForensics) by extracting and cropping frames using the c23 compression. I made adjustments to configuration settings to select the correct folder where my dataset is stored and ran preprocessing and rearrangement scripts provided by the authors, which required a significant time investment.

In the subsequent phase, I made refinements to improve system performance. Notably, I integrated mixed precision techniques to increase batch size and computational efficiency. I also implemented strategic sampling to handle large volumes of data, reducing epoch duration. Specifically, I trained on 250,000 samples and tested on an additional 25,000, drawn from the entire dataset pool. This adaptation significantly reduced training time from 80 to 15 minutes, allowing for more thorough monitoring of the process.

Additionally, I tailored the repository to my computational environment for seamless retrieval of locally stored datasets. Since my proprietary datasets differed structurally from publicly available ones, adjustments were made to accommodate this distinction. Consequently, I eliminated methods reliant on masks and landmarks, absent in my dataset, to streamline the process and enhance simplicity.

3.4 SBI Paper

The paper called "Detecting Deepfakes with Self-Blended Images" [40] proposes a novel approach called Self-Blended Images (SBIs) for deepfake detection. SBIs are synthetic training data generated by blending pseudo source and target images, reproducing common forgery artifacts. The key idea is that SBIs encourage classifiers to learn generic and robust representations, improving model generalization to unknown manipulations and scenes.

The author's objective is to identify statistical inconsistencies between altered facial images and background images in deepfakes. To enhance training, the key is to utilize more data. Rather than merging two distinct faces to create a new face, the author devised a method to generate a new face using a single image. This approach is called Self-Blended Images and the architecture is showed in Figure 3.3. The pipeline operates as follows:

1. Considering a real image, the **Source-Target Generator** creates a duplicate of this image, treating the two images as source and target, respectively. Subsequently, to introduce statistical inconsistencies, various types of random transformations are applied, including random shifts in RGB channels, hue, saturation, value, brightness, and contrast. Additionally, to replicate blending boundaries and landmark mismatches, the source image undergoes resizing and translation, randomly selected from a feasible range.
2. The **Mask Generator** facilitates blending of source and target images using a grayscale mask image. It begins by employing a landmark detector on the input image to predict facial regions and creates a mask by computing the convex hull from these predicted landmarks. The mask undergoes deformation using landmark transformations. To diversify blending masks, the Mask Generator introduces random changes to their shape and blending ratios. This includes elastic deformation and smoothing with Gaussian filters, followed by adjustments to pixel values for erosion or dilation of the mask. Additionally, the Mask Generator varies the blending ratio of the source image by multiplying the mask image by a constant between 0 and 1.
3. The source and target images are blended using the mask to produce a Source-Blended-Image (SBI).

$$I_{SB} = I_s \odot M + I_t \odot (1 - M)$$

I_s represents the image source, I_t represents the image target and M represents the mask.

One peculiar detail of this paper compared with others is that it is possible to train it using only real images. Moreover, during training, real images and artificially fake images are paired together, thus forcing the model to learn the

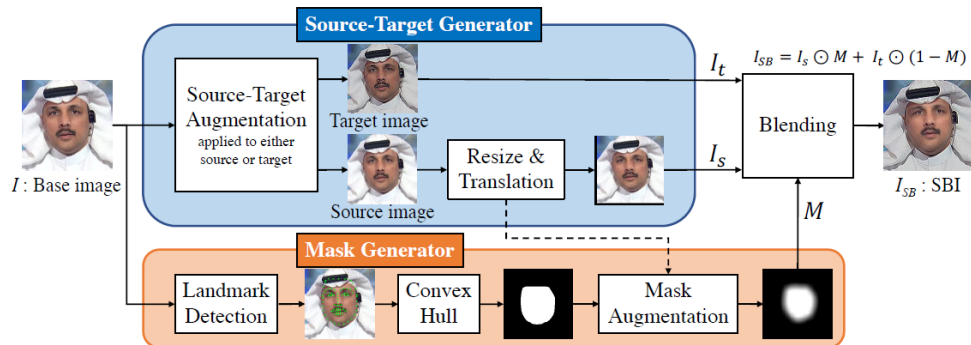


Figure 3.3: Representation of the architecture of the SBI model reported within the paper.

differences between them. It is possible to dwell on an example given in the Image 1.1 and Image 3.4. This method enables the generation of manipulated faces that are both easily identifiable by the human eye and quite challenging to discern. The latter option is crucial as it compels the model to operate with minimal information.

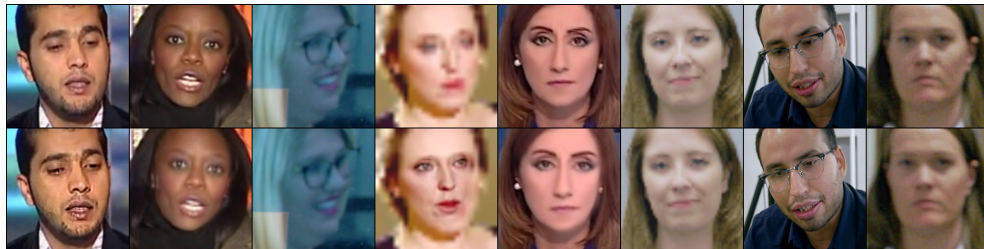


Figure 3.4: The images presented were generated using the SBI method. Real images appear at the top of each image, while fake images are depicted at the bottom.

3.4.1 Implementation

In this paper (like the previous one) the approach was very simple and straightforward. As a first step it was preferred to use the datasets already available in the workstation, this necessitated some modifications to the code. Then some experiments were carried out with real images, initially faceforensics was used as usual. Next we tried different combinations of datasets.

This model relies on landmarks, which are essential facial points or features identified in the input image. An example of landmarks applied in images can be seen in Figure 3.5. They utilized Dlib[19] to compute landmarks, utilizing 81 shape predictor points for each dataset. To expedite the training process, these landmarks were pre-calculated and stored in the workstation. During training, the corresponding landmarks for each image were accessed.

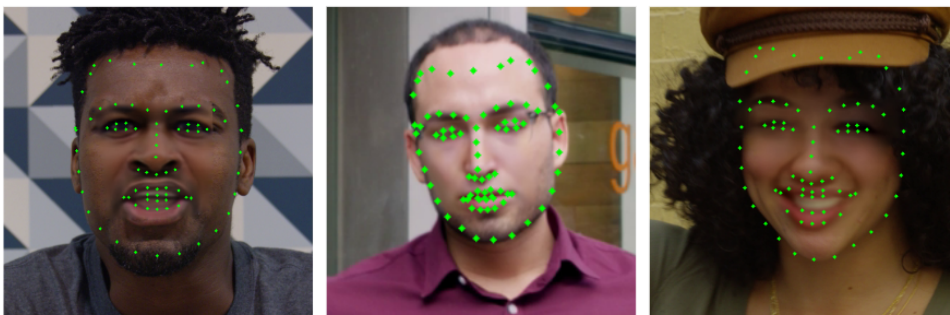


Figure 3.5: Sample images showcasing associated landmarks.

Subsequently, several optimization operations were performed. As the first step, the landmark generation function was accelerated by utilizing GPUs and processing batches, reducing the overall time from almost 31 hours to approximately 10 minutes per dataset. Following this, a decision was made to utilize a subset of data per epoch due to the time required, resulting in a decrease from about 17 hours per epoch to 40 minutes. The data retrieval function was also improved, reducing the processing time from approximately 7 minutes to just a few seconds. Here, the utilization of batches, along with a better formulation of list comprehensions, played a significant role. Finally, the implementation of mixed precision was adopted, increasing the batch size from 4 to 8.

3.5 Locate and Verify Paper

The paper *Locate and Verify: A Two-Stream Network for Improved Deepfake Detection* [41], proposes a two-stream network for improved deepfake

detection. The network effectively identifies potential forged regions and extracts forgery evidence. It includes three functional modules to handle multi-stream and multi-scale features. The paper also introduces a semi-supervised patch similarity learning strategy to estimate patch-level forged location annotations.

When someone changes a face in a picture using editing techniques, some areas might show obvious signs of editing, while other parts remain untouched, keeping the original picture unchanged. This creates an uneven appearance with some parts looking altered and others looking normal. In order to deal with this problem, the authors of the paper created a two-part system able to:

1. **Locate** where manipulations might be in a picture.
2. **Verify** if those areas are manipulated.

The first part guides the second by pointing out areas that are more likely to have fake alterations in the image. The architecture of the model is visible in Figure 3.6.

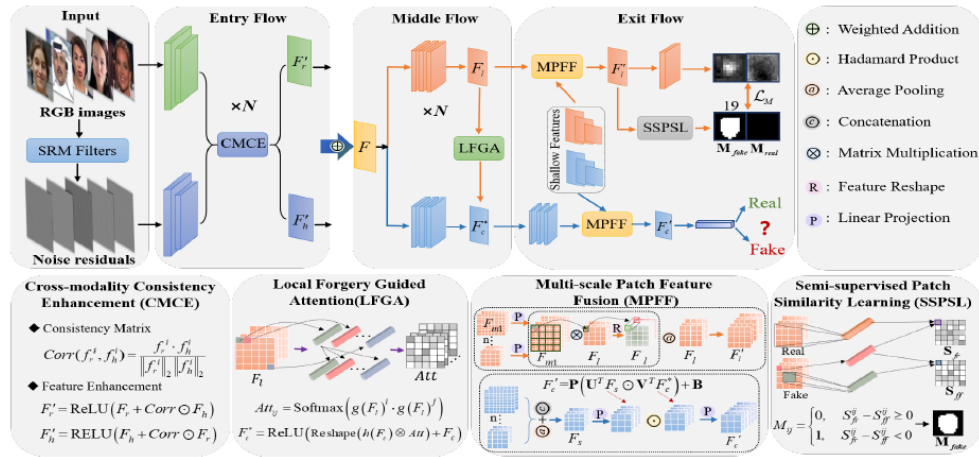


Figure 3.6: Representation of the architecture of the Locate and Verify model reported within the paper.

The pipeline of the data follows this path:

1. In the *Input Flow* the **Spatial Rich Model Filters** are applied to generate an image called noise residual that contains information associated with

the world of steganography (hidden information in objects or messages). This is useful because SRM Filters better captures high-frequency features crucial for image forensics.

2. In the *Entry Flow* there is a combination of RGB and SRM modalities through **Cross-Modality Consistency Enhancement (CMCE)** module. This is employed to collaboratively learn features from two different representations of data.
3. In the *Middle Flow* the **Local Forgery Guided Attention (LFGA)** module obtain an attention map from the location features to guide the learning of more robust and informative classification features.
4. In the *Exit Flow*, in order to exploit artifacts in shallow features, **Multi-scale Patch Feature Fusion (MPFF)** module is introduced. The method involves examining artifacts at multiple scales, where one branch focuses on classification features representing global semantic information, and the other branch concentrates on localization features capturing local spatial details.
5. Still in the *Exit Flow*, to address the absence of forgery annotations in public deepfake datasets, the paper introduces a **Semi-supervised Patch Similarity Learning (SSPSL)** strategy for training the localization branch. Specifically, for real images, the forgery location maps are fixed as all zeroes, while for fake images, regions like the nose, eyes, and mouth (commonly manipulated areas) are identified using facial landmarks.

3.5.1 Implementation

In this approach, the input consists of a combination of the face to be identified and its mask repeated twice (see Figure 3.7). To achieve this, faces and masks were extracted from FaceForensics videos using preprocessing scripts

provided by the repository, along with a script for frame extraction and pre-processing. Two datasets were created from FaceForensics’s videos using different sampling rates: one with 10,000 images and another with 50,000 (along with corresponding masks). Several functionalities were then implemented to improve efficiency and analysis. These included: sampling to reduce epoch time, saving an experiment summary file to track parameters used in different experiments, plotting loss and metrics during training using tensorboard, and implementing mixed precision to increase batch size. Additionally, as the model returns both classification predictions and masks identifying manipulation points, images associated with predicted masks were printed for each epoch.

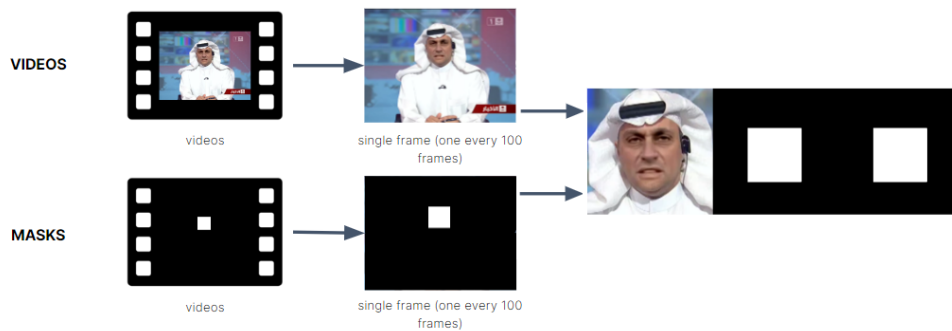


Figure 3.7: Input of the SBI Model.

It’s worth noting that not all input masks are highly accurate; for instance, in Figure 3.7, the corresponding mask appears as a simple square. This is acceptable since extreme accuracy is not required for the entire dataset; a minority of very accurate masks have a significant impact.

3.6 SPSL Paper

The paper Spatial-Phase Shallow Learning: Rethinking Face Forgery Detection in Frequency Domain [26], introduces a novel approach called Spatial-Phase Shallow Learning (SPSL) for detecting face forgery in images. The

authors observe that up-sampling, a common step in face forgery techniques, results in changes in the frequency domain, particularly in the phase spectrum. They propose using the phase spectrum, which preserves important frequency components, along with spatial image information to capture up-sampling artifacts.

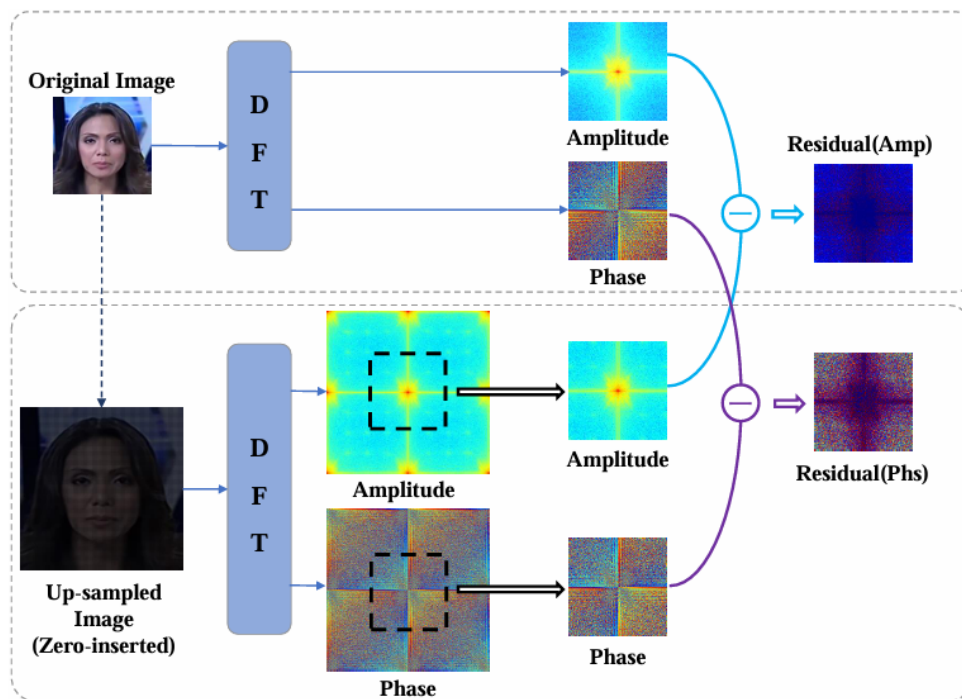


Figure 3.8: Representation of the architecture of the SPSL model reported within the paper.

The architecture of the model is showed in Figure 3.8. The peculiarity of this model is that besides being relatively very simple, it involves reconstructing the representation in the spatial domain of the phase spectrum from the frequency domain and concatenating it with the RGB image to form a 4-channel image. In other words, the input entered within Xception consists of an image with 4 channels.

Together with the UCF model, it is one of the best papers you can find in the repository "DeepfakeBench: A Comprehensive Benchmark of Deepfake Detection" [49].

3.6.1 Implementation

No significant modifications were necessary since the code utilized is the same as that of the UCF paper, thus the changes made previously (Section 3.3) remain. For this reason, it was decided to directly utilize the datasets stored within the workstation and refrain from replicating the results.

3.7 Evaluation

In the evaluation of deepfake detection models, two commonly employed metrics are the Area Under the Receiver Operating Characteristic curve (**AUC**) and **Accuracy**. AUC quantifies the ability of a model to discriminate between positive and negative instances, with a higher AUC indicating superior performance. It is calculated by plotting the true positive rate against the false positive rate across various classification thresholds. Accuracy, on the other hand, measures the proportion of correctly classified instances over the total number of instances. It is computed as the sum of true positives and true negatives divided by the total number of instances. However, it is important to note that in datasets exclusively comprised of either fake or real images, AUC calculation becomes impractical due to the absence of a balanced distribution between positive and negative instances.

Other metrics such as Average Precision (AP) and Equal Error Rate (EER) are also relevant in the evaluation of deepfake detection models. AP measures the area under the precision-recall curve and provides a more nuanced understanding of model performance, particularly in scenarios where class imbalances exist within the dataset. A higher AP indicates better model performance. Equal Error Rate (EER) represents the point on the ROC curve where the false acceptance rate (FAR) is equal to the false rejection rate (FRR). It is an important metric for assessing the overall effectiveness of a detection system, as it signifies the point at which the model is equally likely to misclassify genuine and fake instances. While these metrics provide valuable insights into

model performance, they were not extensively discussed in this thesis.

3.8 Experimental Setup and Code Development

For conducting experiments and code scrutiny, the workstation available in the company equipped with Linux operating system was employed. The GPU utilized was a GeForce RTX 2080 Ti, enhancing computational capabilities.

Python served as the main programming language for this project, with PyTorch being the primary library used. Other essential dependencies included NumPy and OpenCV. Version control was handled through Git, ensuring easy tracking and collaboration. The integrated development environment (IDE) chosen for the project was Visual Studio Code.

Chapter 4

Experiment Results

In this chapter, we present the comprehensive analysis of experimental results obtained from various studies conducted under different papers, shedding light on the efficacy, feasibility, and implications of the methodologies employed.

The experimental investigations delve into the findings outlined in four primary papers: **UFC Paper** (Section 4.1), **SBI Paper** (Section 4.2), **Locate and Verify Paper** (Section 4.3), and **SPSL Paper** (Section 4.4).

Additionally, we provide a combined evaluation between all the previous model in a section called **Cross Evaluation** (Section 4.5).

4.1 UFC Paper

In the initial phase of experimentation, the decision was made to replicate existing findings. Consequently, the FaceForensics dataset obtained from the Internet was utilized, accompanied by the requisite preprocessing steps as specified by the repository. The model was trained using the default parameters obtained from the repository. Key experimental parameters included a learning rate of $1e-4$, a batch size of 8, and input images sized at 256×256 pixels.

Adam optimizer was utilized, the backbone was Xception[7] and data augmentation techniques such as image flipping, rotation, blur, brightness adjustment, contrast adjustment, and quality control were employed. As depicted in Figure 4.1, it is evident that the model exhibits a learning trend, characterized by a gradual decrease in training losses. Concurrently, there is an observable increase in performance metrics such as the Area Under the Curve (AUC) and other relevant metrics provided by the repository.

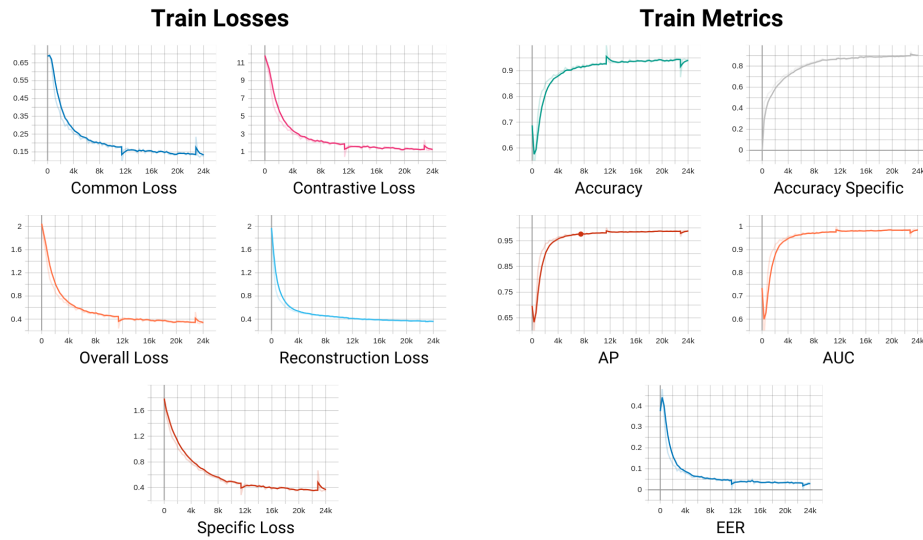


Figure 4.1: Visualization of various losses and metrics recorded during the initial training session with the UCF model on the FaceForensics dataset.

Next, the model was evaluated using both the FaceForensics and CelebDF test set. Upon examination of Table 4.1, it is evident that while the reproduced results do not align precisely with the original outcomes, they exhibit a high degree of fidelity. Thus, the model is overall able to capture the key features for identifying a deepfake within the CelebDF dataset, despite not having been trained on it.

Following this, my first objective was to ascertain the consistency in result representation by conducting experiments using various seeds: 0, 10, 28, 42, and 100. Each training iteration comprised 15 epochs and was assessed on a CelebDF test set at the epoch where the loss was minimized. In addition, the

UCF Result Reproduction (AUC)		
Test Set	Trained Model	Paper Result
FaceForensics	0.9480	0.9705
CelebDF	0.7396	0.7527

Table 4.1: Reproduction of results for the UCF model. Both models were trained using the FaceForensics dataset, and the evaluations presented here are based on the test sets from both FaceForensics and CelebDF datasets.

parameters associated with the model discussed above remain unchanged. The findings, detailed in Table 4.2, reveal a high degree of alignment among the results, thus confirming their consistency. This means not only that the model can discretely identify deepfakes but also that it is quite stable in returning an output.

UCF Result Consistency (AUC)					
Test Set	Seed 0	Seed 10	Seed 28	Seed 42	Seed 100
CelebDF	0.7420	0.7398	0.7406	0.7308	0.7353

Table 4.2: Seed Consistency in UCF Model. The models were trained on the FaceForensics dataset for 15 epochs, and the best epoch was selected. The results demonstrate consistency across different seeds within the same experimental setup.

An additional experiment was conducted to investigate the impact of altering the model’s backbone architecture on its performance. The default backbone utilized in the model is Xception, which was substituted with EfficientNetB4, accompanied by adjustments to the input size (Xception takes an input of 299x299 but in the paper was setup a resolution of 256x256 while EfficientNetB4[43] of 380x380). As indicated in Table 4.3, the results reveal a marginal enhancement in terms of the Area Under the Curve (AUC) metric calculated on the CelebDF test set. Since we can consider the two models equivalent in performance, I will continue to use Xception.

In the experiments with the UCF technique, we tested different learning rates, ranging from 1e-3 to 1e-5. The default value mentioned in the paper,

UCF Backbone Comparison (AUC)		
Test Set	Xception	EfficientNetB4
CelebDF	0.7420	0.7525

Table 4.3: Comparison of the Xception and EfficientNetB5[43] backbones in the UCF Model. Two distinct training runs were conducted with identical setups but different backbones. Experimental results indicate that EfficientNetB5 performs slightly better.

adjusted for our batch size, is $1e-4$. Figure 4.2 illustrates how using a higher learning rate caused the loss value to increase instead of decrease, prompting us to stop training for time constraints. Conversely, too low a learning rate resulted in excessively long convergence times. Even with the longest training duration, the loss did not reach its previous level of $1e-4$. These experiments highlighted the significant impact of the $1e-4$ learning rate in facilitating convergence.

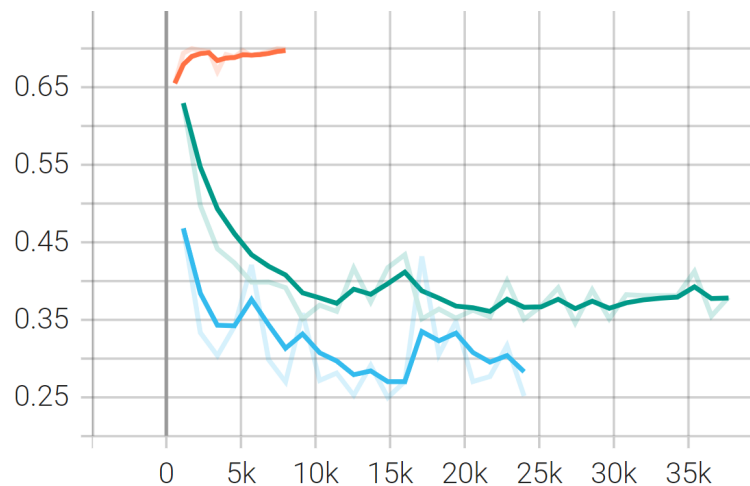


Figure 4.2: The visualization illustrates experiments conducted with different learning rates: $1e-3$ (orange), $1e-5$ (green), and $1e-4$ (blue). The results indicate that the experiment with a learning rate of $1e-4$ demonstrates superior convergence compared to the others.

Moreover, we explored the effectiveness of data augmentation, emphasizing its pivotal role in improving the performance of this technique. Data augmentation involves artificially expanding the training dataset by applying various

transformations such as rotation, flipping, scaling, and brightness adjustments to the existing images. This process helps the model generalize better and learn robust features from the data. Through our investigations, we found that incorporating data augmentation significantly improved the model’s ability to generalize to unseen data and enhanced its overall performance. Results are presented in Table 4.4.

UCF Data Augmentation Comparison (AUC)		
Test Set	with Data Augmentation	without Data Augmentation
CelebDF	0.7420	0.7288

Table 4.4: Comparison of Experiments with and without Data Augmentation in the UCF Model trained on FaceForensics

Additionally, we conducted a brief experiment to investigate the effectiveness of dropout regularization. Dropout is a technique commonly used in neural networks to prevent overfitting by randomly dropping out some neurons during training. We compared the performance of the model with and without dropout by conducting short training sessions and observing the resulting AUC test scores. Table 4.5 illustrates the performance improvements achieved with dropout regularization. These results demonstrate the beneficial impact of dropout in enhancing the model’s generalization ability and overall performance.

UCF Dropout Comparison (AUC)		
Test Set	with Dropout	Without Dropout
CelebDF	0.7573	0.7420

Table 4.5: Comparison with and without dropout in the UCF Model. Two distinct training runs were conducted with identical setups. Experimental results indicate that Dropout have a good impact for performances

The top-performing model, trained on FaceForensics datasets comprising actors, Deepfakes, Face2Face, FaceSwap, and NeuralTextures, underwent evaluation across various datasets. This analysis highlighted the limitations of relying solely on a single dataset to ensure the adaptability of the model. Table 4.6 illustrates that while the model performs well on datasets within the same domain, it struggles when faced with out-of-distribution datasets. This result is expected since the model was trained exclusively on one domain. Additionally, the model tends to classify images as real more frequently than fake, as evident when comparing its performance on real and fake image datasets. It's worth noting that the model performs poorly on privately generated datasets, which utilize newer and more complex techniques compared to those found online. This challenge deviates from our goal of generalizing the model effectively.

After conducting initial experiments, we proceeded with more detailed investigations involving a broader range of datasets and varied combinations. Through thorough evaluation of these combinations, we identified a specific configuration for further analysis. This configuration included datasets such as Defacto-morphs, Face2Face, FaceSwap, NeuralTextures, actors, youtube, FFHQ, FFHQ-GFPGAN, Ms-Celeb-1M, StyleGAN3, VGGFace, faceapp-morph, reface, and tedx. We chose these datasets based on factors such as the varying difficulty levels in identifying deepfakes within each dataset, the presence of specific types of deepfakes, and the necessity to include datasets containing deepfakes generated from images within other datasets, or vice versa. As shown in Table 4.7, incorporating multiple datasets indeed led to improved results, as expected. Surprisingly, the performance of the majority of the out-of-distribution datasets not only benefited from this training but also increased significantly. While these improvements may not be dramatic, they represent a promising step towards better generalization of deepfake detection. Additionally, we observed a tendency to classify images across all out-of-distribution datasets as real. An interesting observation is the below-average results of

UCF Model trained on FaceForensics			
Name	Type (Real/Fake)	Accuracy	AUC
In-Domain Datasets			
FaceForensics/Deepfakes	Fake	0.9371	-
FaceForensics/Face2Face	Fake	0.9493	-
FaceForensics/FaceSwap	Fake	0.9558	-
FaceForensics/NeuralTextures	Fake	0.8799	-
FaceForensics/youtube	Real	0.9566	-
Out-of-Distribution Datasets			
blendswap-swapped	Fake	0.4102	-
CelebA	Real	0.9499	-
CelebA-GFPGAN	Fake	0.3596	-
CelebDF	Fake+Real	0.5098	0.6911
Cheap-Morphs	Fake	0.6552	-
DeeperForensics	Fake+Real	0.8821	0.6678
Defacto-morphs	Fake	0.5822	-
DFDM	Fake	0.9871	-
faceapp-faceswap	Fake	0.0775	-
faceapp-morph	Fake	0.1600	-
FaceForensics/actors	Real	0.6971	-
FFHQ	Real	0.8135	-
FFHQ-GFPGAN	Fake	0.5562	-
FRLM-Morphs	Fake	0.1071	-
iFakeFaceDB	Fake	0.9239	-
insightface-swapped	Fake	0.2739	-
LRS3	Real	0.9387	-
MegaFS	Fake	0.3737	-
Ms-Celeb-1M	Real	0.6340	-
reface	Fake	0.3925	-
simswap-swapped	Fake	0.4536	-
stable-diffusion	Fake	0.3486	-
StyleGAN3	Fake	0.2031	-
synthesis-generated	Fake	0.3140	-
tedx	Real	0.8840	-
TPDNE	Fake	0.1305	-
unstable-diffusion	Fake	0.3627	-
VGGFace	Real	0.9483	-
Wav2lip	Fake	0.2972	-

Table 4.6: Performances of the UFC paper trained on FaceForensics

FFHQ and youtube datasets. However, these results can be justified: youtube is the original dataset from which all other FaceForensics datasets containing faked images were created, while FFHQ is the dataset used to create FFHQ-GFPGAN. Furthermore, we noted that the simswap-swapped dataset is particularly challenging to classify, likely due to its association with more complex deepfake generation techniques.

Failing to improve these results, we chose to veer toward a new paper.

4.2 SBI Paper

In contrast to the previous study, we had access to the model's weights. This allowed us to investigate whether the performance metrics obtained from evaluating the model on the CelebDF test set matched those reported in the paper. Additionally, instead of using Xception as the backbone, the authors opted for EfficientNetB5. This decision provided an opportunity for us to explore the effects of varying input sizes on performance through experimentation: 256x256, 380x380, and 456x456. While I was curious about this experiment, I understood that EfficientNetB5 typically works with an input size of 456x456. However, in their repository, the images had been resized to 380x380. Upon comparing the results, a significant disparity became apparent. Table 4.8 clearly shows inconsistencies between the Area Under the Curve (AUC) reported in the paper and the values obtained through our evaluations with different input dimensions. It is possible that the original study employed a different evaluation methodology, which unfortunately we could not ascertain from the paper itself. Due to these discrepancies and encountered limitations, we made the decision to refrain from replicating the reported results by training a model from scratch. Instead, we proceeded directly with the implementation of the model in our own experimentation.

The next experiment involves training the SBI model using the FaceForensics dataset stored in the workstation. It's important to recall that this method

UCF Model trained on Multiple Datasets			
Name	Type (Real/Fake)	Accuracy	AUC
In-Domain Datasets			
Defacto-morphs	Fake	1.0	-
faceapp-morph	Fake	0.7707	-
FaceForensics/actors	Real	0.9758	-
FaceForensics/Deepfakes	Fake	0.9702	-
FaceForensics/Face2Face	Fake	0.9907	-
FaceForensics/FaceSwap	Fake	0.9883	-
FaceForensics/NeuralTextures	Fake	0.9341	-
FaceForensics/youtube	Real	0.7377	-
FFHQ	Real	0.4444	-
FFHQ-GFPGAN	Fake	0.9988	-
Ms-Celeb-1M	Real	0.9998	-
StyleGAN3	Fake	0.9801	-
tedx	Real	0.9824	-
VGGFace	Real	0.9794	-
Out-of-Distribution Datasets			
blendswap-swapped	Fake	0.5765	-
CelebA	Real	0.8598	-
CelebA-GFPGAN	Fake	0.9762	-
CelebDF	Fake+Real	0.6179	0.6878
Cheap-Morphs	Fake	0.7895	-
DeeperForensics	Fake+Real	0.1370	0.3590
DFDM	Fake	0.9284	-
faceapp-faceswap	Fake	0.6468	-
FRLM-Morphs	Fake	0.6138	-
iFakeFaceDB	Fake	0.5630	-
insightface-swapped	Fake	0.4771	-
LRS3	Real	0.9722	-
MegaFS	Fake	1.0	-
reface	Fake	0.9848	-
simswap-swapped	Fake	0.1132	-
stable-diffusion	Fake	0.9514	-
synthesis-generated	Fake	0.9993	-
TPDNE	Fake	0.9984	-
unstable-diffusion	Fake	0.8755	-
Wav2lip	Fake	0.3208	-

Table 4.7: Performances of the UFC paper trained on a combination of different Datasets

SBI Their Pretrained Weights Evaluation (AUC)				
Using different images sizes				
Test Set	256x256	380x380	456x456	Paper Result
CelebDF	0.5894	0.7084	0.7107	0.9287

Table 4.8: Performance evaluation of the provided pretrained weights on SBI using different resolutions. The experiments reveal a lack of correspondence between the reported results in the paper and the obtained results.

only utilizes real images as input, so we'll only consider datasets within FaceForensics that contain real images, namely actors and youtube. As shown in Figure 4.1, the training appears to progress in the right direction. However, there are occasional spikes in the loss, which I suspect are associated with more complex portions of the data.

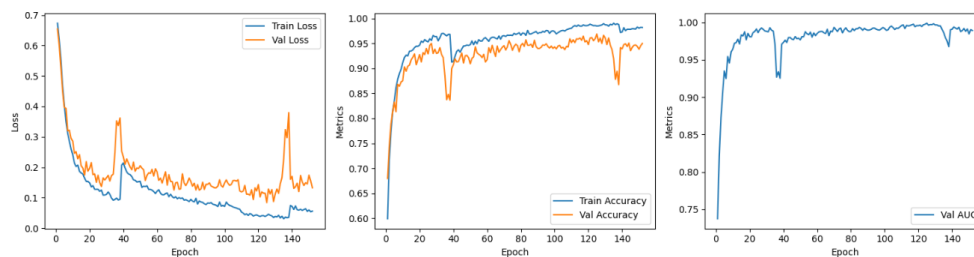


Figure 4.3: Visualization of losses, accuracies, and AUC obtained during the initial training with the SBI model on the FaceForensics dataset.

I assessed the collected data across various datasets, as displayed in Table 4.9. As anticipated, exclusively utilizing the YouTube and actors datasets from FaceForensics did not produce satisfactory results. The trend observed leans towards classifying the images as real. In datasets containing fake images, the results consistently performed poorly. Considering the significant gap between within-domain and out-of-domain datasets, it is evident that this model struggles with generalization.

Aware of the unsatisfactory results obtained previously, we conducted several experiments using different combinations of datasets containing real images. However, these experiments did not produce noteworthy results. Table

SBI Model trained on FaceForensics			
Name	Type (Real/Fake)	Accuracy	AUC
In-Domain Datasets			
FaceForensics/actors	Real	0.9708	-
FaceForensics/youtube	Real	0.9949	-
Out-of-Distribution Datasets			
blendswap-swapped	Fake	0.2277	-
CelebA	Real	0.9763	-
CelebA-GFPGAN	Fake	0.2362	-
CelebDF	Fake+Real	0.4098	0.5334
Cheap-Morphs	Fake	0.4502	-
DeeperForensics	Fake+Real	0.9503	0.4758
Defacto-morphs	Fake	0.6927	-
DFDM	Fake	0.7093	-
faceapp-faceswap	Fake	0.0305	-
faceapp-morph	Fake	0.0821	-
FaceForensics/Deepfakes	Fake	0.6428	-
FaceForensics/Face2Face	Fake	0.1393	-
FaceForensics/FaceSwap	Fake	0.4829	-
FaceForensics/NeuralTextures	Fake	0.0292	-
FFHQ	Real	0.8586	-
FFHQ-GFPGAN	Fake	0.2170	-
FRLM-Morphs	Fake	0.0565	-
iFakeFaceDB	Fake	0.3277	-
insightface-swapped	Fake	0.2063	-
LRS3	Real	0.9890	-
MegaFS	Fake	0.7519	-
Ms-Celeb-1M	Real	0.9825	-
reface	Fake	0.2750	-
simswap-swapped	Fake	0.1841	-
stable-diffusion	Fake	0.4409	-
StyleGAN3	Fake	0.1753	-
synthesis-generated	Fake	0.3899	-
tedx	Real	0.9762	-
TPDNE	Fake	0.0683	-
unstable-diffusion	Fake	0.2264	-
VGGFace	Real	0.9818	-
Wav2lip	Fake	0.2972	-

Table 4.9: Performances of the SBI model trained on FaceForensics

4.10 presents the evaluation of the best-performing model trained on Actors, YouTube, Ms-Celeb-1M, VGGFace, and LRS3 datasets. It's worth noting that the results appear to be relatively stable compared to the previous experiment or even show a slight decline, indicating no significant improvement.

Experiments were also performed finetuning the model using the weights provided by the authors. Unfortunately, no noteworthy results are reported. One interesting experiment that we did not include involved training the model solely on the LRS3 dataset. Interestingly, this model achieved results above the average of the tested SBI models. However, since it deviated from the main focus of our generalization task, we decided to proceed with the study of a new paper.

4.3 Locate and Verify Paper

As a first step, the results also had to be replicated in this paper. The dataset used for training is again FaceForensics and the parameters remained the default ones. Specifically, the image resolution chosen by the authors is 299x299, the batch size equals 8 and is associated with a learning rate of $5e-4$. The optimizer is adam and again the backbone is xception. In addition, as a dataset, the dataset with 10k elements was used for some preliminary experiments, and later the dataset with 50k elements was used permanently. As evident from Table 4.11, the results obtained are distant from the anticipated outcomes outlined in the paper.

Despite the disappointing results obtained earlier, I continued to evaluate the outcomes presented in Table 4.12. As anticipated, the results are extremely low. It's important to highlight that the model struggled to capture the main features of the actors dataset, resulting in very poor performance despite it being an in-domain dataset. Additionally, there are several datasets where the model's performance falls below 0.10, rendering it unusable.

To provide a comprehensive overview, I want to mention that there were

SBI Model trained on Multiple Datasets			
Name	Type (Real/Fake)	Accuracy	AUC
In-Domain Datasets			
FaceForensics/actors	Real	0.9165	-
FaceForensics/youtube	Real	0.9806	-
LRS3	Real	0.9957	-
Ms-Celeb-1M	Real	0.9957	-
VGGFace	Real	0.9830	-
Out-of-Distribution Datasets			
blendswap-swapped	Fake	0.6063	-
CelebA	Real	0.9788	-
CelebA-GFPGAN	Fake	0.3000	-
CelebDF	Fake+Real	0.5201	0.6164
Cheap-Morphs	Fake	0.4100	-
DeeperForensics	Fake+Real	0.7724	0.5788
Defacto-morphs	Fake	0.6674	-
DFDM	Fake	0.9548	-
faceapp-faceswap	Fake	0.0381	-
faceapp-morph	Fake	0.0638	-
FaceForensics/Deepfakes	Fake	0.8061	-
FaceForensics/Face2Face	Fake	0.2711	-
FaceForensics/FaceSwap	Fake	0.4882	-
FaceForensics/NeuralTextures	Fake	0.1541	-
FFHQ	Real	0.9225	-
FFHQ-GFPGAN	Fake	0.2995	-
FRLM-Morphs	Fake	0.0239	-
iFakeFaceDB	Fake	0.1049	-
insightface-swapped	Fake	0.3040	-
MegaFS	Fake	0.6491	-
reface	Fake	0.3387	-
simswap-swapped	Fake	0.5410	-
stable-diffusion	Fake	0.4032	-
StyleGAN3	Fake	0.1088	-
synthesis-generated	Fake	0.4733	-
tedx	Real	0.9347	-
TPDNE	Fake	0.0267	-
unstable-diffusion	Fake	0.2305	-
Wav2lip	Fake	0.0974	-

Table 4.10: Performances of the SBI model trained on a combination of different Datasets

SPSL Result Reproduction (AUC)		
Test Set	Trained Model	Paper Result
FaceForensics	0.7796	0.9980
CelebDF	0.5465	0.8600

Table 4.11: Reproduction of results for the LAV model. Both models were trained using the FaceForensics dataset, and the evaluations presented here are based on the test sets from both FaceForensics and CelebDF datasets.

two experiments conducted: one where dropout was removed and another where data augmentation was removed. The aim was to determine if these two techniques were overly aggressive and hindering proper model training. However, the results were slightly worse than those presented in Table 4.12, so I chose not to include them. Additionally, it seems that the model is unable to generate masks associated with manipulation areas. The obtained results show masks identical to those given as input, contrary to expectations. This discrepancy suggests that either the module inside the model is not functioning properly or it has not been implemented correctly. In any case, we decided to move forward.

4.4 SPSL Paper

In this paper, unlike others, we did not conduct a replication of results because the model is already implemented with necessary modifications (made during the study of the UCF paper) within DeepFakeBench[49]. Therefore, our first experiment involved training using default values reported in the repository and the FaceForensics dataset. To provide further detail, we reduced the batch size to 16 from 32 due to GPU limitations, which required us to also adjust the learning rate to $1.414e-4$. This adjustment ensures stability and convergence during training by aligning the learning rate with the new batch size. Input images were resized to a resolution of 256x256 pixels. The optimizer

Locate and Verify Model trained on FaceForensics			
Name	Type (Real/Fake)	Accuracy	AUC
In-Domain Datasets			
FaceForensics/actors	Real	0.8538	-
FaceForensics/Deepfakes	Fake	0.9490	-
FaceForensics/Face2Face	Fake	0.9070	-
FaceForensics/FaceSwap	Fake	0.7674	-
FaceForensics/NeuralTextures	Fake	0.9463	-
FaceForensics/youtube	Real	0.2539	-
Out-of-Distribution Datasets			
blendswap-swapped	Fake	0.7233	-
CelebA	Real	0.7757	-
CelebA-GFPGAN	Fake	0.4008	-
CelebDF	Fake+Real	0.6354	0.5465
Cheap-Morphs	Fake	0.7363	-
DeeperForensics	Fake+Real	0.3127	0.5023
Defacto-morphs	Fake	0.0075	-
DFDM	Fake	0.7595	-
faceapp-faceswap	Fake	0.3024	-
faceapp-morph	Fake	0.3076	-
FFHQ	Real	0.6433	-
FFHQ-GFPGAN	Fake	0.2278	-
FRLM-Morphs	Fake	0.0177	-
iFakeFaceDB	Fake	0.1294	-
insightface-swapped	Fake	0.7307	-
LRS3	Real	0.1290	-
MegaFS	Fake	0.4574	-
Ms-Celeb-1M	Real	0.2726	-
reface	Fake	0.2077	-
simswap-swapped	Fake	0.8347	-
stable-diffusion	Fake	0.6780	-
StyleGAN3	Fake	0.4106	-
synthesis-generated	Fake	0.0376	-
tedx	Real	0.0404	-
TPDNE	Fake	0.2170	-
unstable-diffusion	Fake	0.5645	-
VGGFace	Real	0.4684	-
Wav2lip	Fake	0.8174	-

Table 4.12: Performances of the Locate and Verify model trained on FaceForensics

employed was Adam, and data augmentation techniques applied included flipping, rotating, blurring, adjusting brightness and contrast, and setting quality limits. Despite some fluctuations, the training appears to proceed smoothly, as demonstrated by Figure 4.4.

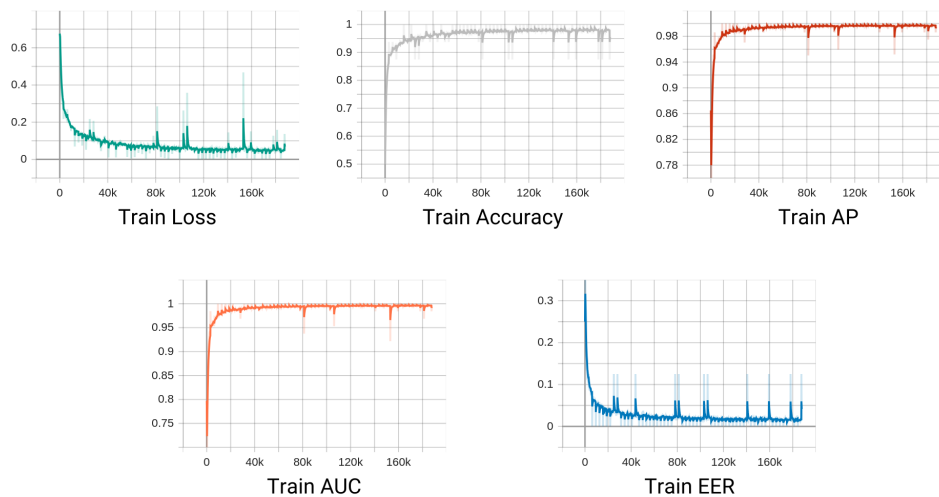


Figure 4.4: Visualization of various losses and metrics recorded during the initial training session with the SPSL model on the FaceForensics dataset.

Subsequently, an evaluation was conducted on FaceForensics, as shown in Table 4.13, which led to results that were not extraordinary but encouraging nonetheless. The model appears to have captured some key characteristics of deepfakes but struggles with certain datasets in particular.

Intrigued by the technique employed, I sought to understand the impact of the Fourier transform on the final results. To investigate this, I conducted an experiment comparing the performance of the model with and without this transform. The results shown in Table 4.14 demonstrate how the introduced layer using the fourier transform has a positive impact by going on to significantly improve performance.

As usual, in this case, attempts were made to introduce new datasets into

SPSL Model trained on FaceForensics			
Name	Type (Real/Fake)	Accuracy	AUC
In-Domain Datasets			
FaceForensics/Deepfakes	Fake	0.9661	-
FaceForensics/Face2Face	Fake	0.9841	-
FaceForensics/FaceSwap	Fake	0.9851	-
FaceForensics/NeuralTextures	Fake	0.9086	-
FaceForensics/youtube	Real	0.8208	-
Out-of-Distribution Datasets			
blendswap-swapped	Fake	0.4683	-
CelebA	Real	0.9003	-
CelebA-GFPGAN	Fake	0.4899	-
CelebDF	Fake+Real	0.5542	0.7763
Cheap-Morphs	Fake	0.7290	-
DeeperForensics	Fake+Real	0.7851	0.5214
Defacto-morphs	Fake	0.4925	-
DFDM	Fake	0.9612	-
faceapp-faceswap	Fake	0.1438	-
faceapp-morph	Fake	0.1666	-
FaceForensics/actors	Real	0.6517	-
FFHQ	Real	0.7415	-
FFHQ-GFPGAN	Fake	0.7083	-
FRLM-Morphs	Fake	0.2386	-
iFakeFaceDB	Fake	0.5008	-
insightface-swapped	Fake	0.3537	-
LRS3	Real	0.8282	-
MegaFS	Fake	0.5667	-
Ms-Celeb-1M	Real	0.5219	-
reface	Fake	0.3069	-
simswap-swapped	Fake	0.5504	-
stable-diffusion	Fake	0.6633	-
StyleGAN3	Fake	0.3354	-
synthesis-generated	Fake	0.3379	-
tedx	Real	0.7747	-
TPDNE	Fake	0.2057	-
unstable-diffusion	Fake	0.6203	-
VGGFace	Real	0.8514	-
Wav2lip	Fake	0.1534	-

Table 4.13: Performances of the SPSL paper trained on FaceForensics

SPSL Input Comparison (AUC)		
Test Set	SPSL	SPSL without Transformation
CelebDF	0.7763	0.6578
DeeperForensics	0.5214	0.5176

Table 4.14: Comparison of the Area Under the Curve (AUC) performance metrics for SPSL models with and without Fourier transformation on different test sets. The results highlight the impact of the transformation technique on model performance.

the training in the hope of some improvement in the results. Among the various experiments conducted, the best combination obtained was found to include: FaceApp-Morph, Actors, Deepfakes, Face2Face, FaceSwap, Neural-Textures, YouTube, FFHQ, Ms-Celeb-1M, Reface, StyleGAN3, TEDx, and the experiment can be observed in Table 4.15. Our findings suggest that this model performs slightly better overall. It works well with most datasets, but there are a few, like Wav2lip and VGGFace, where it either matches or even underperforms compared to the previous model trained on FaceForensics. Also, this model seems to struggle more in detecting real faces accurately.

Some experiments were also carried out by varying the learning rate and sample size. The former was useful to see if the model converged in the correct way, and the latter was ideal to give the model more stability. However, the results are not noteworthy; in fact, the results are very similar to the values in Table 4.15. For this reason we stopped working on this paper.

SPSL Model trained on Multiple Datasets			
Name	Type (Real/Fake)	Accuracy	AUC
In-Domain Datasets			
faceapp-morph	Fake	0.8744	-
FaceForensics/actors	Real	0.9827	-
FaceForensics/Deepfakes	Fake	0.9752	-
FaceForensics/Face2Face	Fake	0.9904	-
FaceForensics/FaceSwap	Fake	0.9864	-
FaceForensics/NeuralTextures	Fake	0.9386	-
FaceForensics/youtube	Real	0.8179	-
FFHQ	Real	0.8280	-
Ms-Celeb-1M	Real	0.8376	-
reface	Fake	0.9861	-
StyleGAN3	Fake	0.9017	-
tedx	Real	0.9879	-
Out-of-Distribution Datasets			
blendswap-swapped	Fake	0.8948	-
CelebA	Real	0.2603	-
CelebA-GFPGAN	Fake	0.6730	-
CelebDF	Fake+Real	0.6092	0.7054
Cheap-Morphs	Fake	0.5610	-
DeeperForensics	Fake+Real	0.3827	0.4063
Defacto-morphs	Fake	0.1662	-
DFDM	Fake	0.9736	-
faceapp-faceswap	Fake	0.8208	-
FFHQ-GFPGAN	Fake	0.3370	-
FRLM-Morphs	Fake	0.9214	-
iFakeFaceDB	Fake	0.9999	-
insightface-swapped	Fake	0.5986	-
LRS3	Real	0.9470	-
MegaFS	Fake	0.9950	-
simswap-swapped	Fake	0.3368	-
stable-diffusion	Fake	0.8827	-
synthesis-generated	Fake	0.8579	-
TPDNE	Fake	0.9002	-
unstable-diffusion	Fake	0.6911	-
VGGFace	Real	0.4025	-
Wav2lip	Fake	0.1800	-

Table 4.15: Performances of the SPSL model trained on a combination of different Datasets

4.5 Cross Evaluation

The Table 4.16 presents a comparison of various deepfake detection models in terms of their performance on out-of-distribution (OOD) datasets using accuracy metrics. The models analyzed include UCF-FaceForensics (Table 4.6), UCF-MultiDatasets (Table 4.7), SBI-FaceForensics (Table 4.9), SBI-MultiDatasets (Table 4.10), LAV-FaceForensics (Table 4.12), SPSL-FaceForensics (Table 4.13), and SPSL-MultiDatasets (Table 4.15). The key findings are:

- **UCF-MultiDatasets** demonstrates the highest overall accuracy (ALL) among all models, indicating superior performance across all datasets. This model also exhibits the highest accuracy for both real and fake OOD datasets, suggesting strong performance on out-of-distribution images.
- **SBI-FaceForensics** performs exceptionally well on real OOD datasets, while **SPSL-MultiDatasets** excels on fake OOD datasets.
- **LAV-FaceForensics** has the lowest overall accuracy, indicating comparatively poorer performance across datasets.
- **SBI-MultiDatasets** and **SPSL-FaceForensics** show reasonable performance but are outperformed by **UCF-MultiDatasets** in terms of overall accuracy.

These findings suggest that **UCF-MultiDatasets** is the most robust model for deepfake detection, offering strong performance across various datasets. However, other models may excel in specific areas, such as detecting real or fake out-of-distribution images. Further analysis and experimentation could provide deeper insights into the strengths and weaknesses of each model, contributing to the advancement of deepfake detection technology.

Models Comparisons Out-of-Distribution using Accuracy				
Model	ALL	Real OOD	Fake OOD	OOD
UCF-FaceForensics (4.6)	0.6699	0.8379	0.3984	0.6182
UCF-MultiDatasets (4.7)	0.8396	0.9160	0.7384	0.8272
SBI-FaceForensics (4.9)	0.6360	0.9607	0.3058	0.6333
SBI-MultiDatasets (4.10)	0.6559	<u>0.9453</u>	0.3484	<u>0.6469</u>
LAV-FaceForensics (4.12)	0.4676	0.3882	0.4284	0.4083
SPSL-FaceForensics (4.13)	0.6480	0.7528	0.4496	0.5282
SPSL-MultiDatasets (4.15)	<u>0.7632</u>	0.5366	<u>0.6935</u>	0.6151

Table 4.16: Comparison of model performances on Out-of-Distribution data using accuracy metrics. 'ALL' represents the weighted average accuracy across all datasets, while 'Real OOD' and 'Fake OOD' denote the average accuracy on datasets containing real and fake images, respectively. 'OOD' indicates the overall weighted average accuracy on Out-of-Distribution datasets. Metrics in bold represent the best values, those in underlined the second best.

Chapter 5

Discussion

In the course of implementing and experimenting with various models, one of the primary challenges arises from the necessity to work with code written by others and modify it to suit one's own purposes. Central to overcoming this challenge is gaining a thorough understanding of the repository's structure, comprehensively grasping the associated research paper, and delving into the intricate details of how data flows within the model.

A notable observation is the inconsistency between the results reported in different papers and those obtained in personal experiments. While several factors may contribute to this, I tend to attribute it to the diverse range of datasets available, which simulates real-world scenarios and thus introduces complexity. This diversity allows for a deeper assessment of the effectiveness of various tools. I also believe that some of the simplifications applied (e.g. sampling and mixed precision) may have led to discrepancies between the paper's results and my own. However, despite the presence of repository code, ambiguities in certain steps outlined in the paper can contribute to errors during implementation.

One major challenge faced during these experiments is the inconsistency in performance when using datasets that are different from the ones used during training. This inconsistency occurs even within epochs that are close together in time. This variability not only makes the experiments less reliable but also

makes it difficult to reproduce results, making the experimentation process very complex. When we visually examine the graphs showing the loss on test sets for various out-of-distribution datasets, we see this inconsistency clearly. The graphs show unstable patterns with many peaks and smaller peaks, highlighting the irregularity in performance.

A primary limitation of models aimed at generalization is their tendency to excel with certain datasets while faltering with others, a phenomenon that exacerbates the complexity of the testing process and diminishes the model's reliability. Identifying the optimal combination of datasets and parameters is a daunting task that demands considerable expertise. Moreover, given the inherent complexity of deep learning models, comprehending their inner workings is exceedingly challenging. Additionally, while deepfake research is ongoing, finding recent papers for implementation has proven to be challenging. Nonetheless, I anticipate increased awareness and research investment in this domain in the future.

Techniques such as dropout and data augmentation yield highly beneficial outcomes in experiments of this nature, as evidenced by both the findings of this thesis and the prevalence of these methods in analyzed papers.

Among the models employed, UCF-MultiDatasets appears to have outperformed others significantly. Interestingly, augmenting the UCF model with additional datasets led to a notable improvement in Out-Of-Distribution metrics, by nearly 0.20. In contrast, the improvements in SBI and SPSL models were marginal, with SBI increasing by only 0.02 and SPSL by approximately 0.10. Furthermore, there seems to be a consistent trend of better performance with datasets containing real images compared to those with negative images. One potential explanation could be the size of the Ms-Celeb-1M dataset, which might introduce label imbalances.

5.1 Future Works

To advance the generalization of deepfake detection models, it is essential to have access to diverse and representative datasets for experimentation. As demonstrated, the combination of datasets plays a crucial role in determining the model's performance. However, identifying the optimal dataset combination is not a straightforward task and requires systematic exploration. Therefore, future research efforts should focus on curating comprehensive datasets spanning various deepfake generation techniques, contexts, and visual attributes.

In addition to dataset curation, exploring the potential of data augmentation techniques is imperative for enhancing model performance. Augmentation methods such as the SBI technique hold promise in augmenting the training data with diverse and realistic deepfake samples, thereby enabling the model to better generalize to unseen variations. Investigating the efficacy of novel augmentation strategies tailored specifically for deepfake detection could yield further improvements in model robustness and accuracy.

Furthermore, future research endeavors could benefit from exploring alternative model architectures and techniques to enhance the interpretability and explainability of deepfake detection systems. For instance, integrating additional layers or modules within the model, such as those inspired by the SPSL model, could enable the identification of specific patterns or artifacts indicative of deepfake manipulation. This approach not only enhances detection performance but also provides valuable insights into the underlying mechanisms driving deepfake generation.

Moreover, considering the dynamic nature of deepfake technology, continuous monitoring and adaptation of detection models are essential to stay ahead of emerging threats. Future studies should prioritize the development of adaptive learning mechanisms capable of dynamically updating model parameters and strategies in response to evolving deepfake generation techniques.

Chapter 6

Conclusion

The issue of generalizing deepfake detection models represents a significant challenge with profound implications for security systems worldwide. This thesis has highlighted the complexity involved in achieving effective generalization in deepfake detection.

Deepfake technology continues to advance rapidly, posing increasingly sophisticated threats across various sectors. As such, the imperative to develop reliable detection mechanisms remains paramount. Despite the encountered complexities and limitations, the findings of this research contribute valuable insights to the field, each experiment and observation providing incremental progress toward bolstering our defenses against manipulated media.

Furthermore, this study underscores the need for multidisciplinary collaboration involving researchers, engineers, policymakers, and industry stakeholders. Addressing the challenge of generalization requires advancements not only in machine learning algorithms and data processing but also a deeper understanding of the social and technological factors driving the proliferation of deepfake content.

Looking forward, sustained investment in research and innovation is essential to improving the robustness and scalability of deepfake detection systems. By fostering collaboration and knowledge exchange within the research

community, we can collectively work towards creating a safer digital environment. While the journey towards foolproof deepfake detection may be challenging, each incremental advancement brings us closer to a future where trust and authenticity prevail in the digital realm.

Bibliography

- [1] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7, 2018. DOI: 10.1109/WIFS.2018.8630761.
- [2] T. Afouras, J. S. Chung, and A. Zisserman. Lrs3-ted: a large-scale dataset for visual speech recognition, 2018. arXiv: 1809.00496 [cs.CV].
- [3] T. Afouras, J. S. Chung, and A. Zisserman. LRS3-TED: a large-scale dataset for visual speech recognition. *CoRR*, abs/1809.00496, 2018. arXiv: 1809.00496. URL: <http://arxiv.org/abs/1809.00496>.
- [4] I. Amerini, L. Galteri, R. Caldelli, and A. D. Bimbo. Deepfake video detection through optical flow based cnn, 2019. URL: https://openaccess.thecvf.com/content_ICCVW_2019/html/HBU/Amerini_Deepfake_Video_Detection_through_Optical_Flow_Based_CNN_ICCVW_2019_paper.html.
- [5] J. Cao, C. Ma, T. Yao, S. Chen, S. Ding, and X. Yang. End-to-end reconstruction-classification learning for face forgery detection, 2022. URL: https://openaccess.thecvf.com/content/CVPR2022/html/Cao_End-to-End_Reconstruction-Classification_Learning_for_Face_Forgery_Detection_CVPR_2022_paper.html.

-
- [6] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: a dataset for recognising faces across pose and age, 2018. arXiv: 1710.08092 [cs.CV].
- [7] F. Chollet. Xception: deep learning with depthwise separable convolutions, 2017. arXiv: 1610.02357 [cs.CV].
- [8] S. Cole. We are truly fucked: everyone is making ai-generated fake porn now. URL: <https://www.vice.com/en/article/bjye8a/reddit-fake-porn-app-daisy-ridley>.
- [9] C. Du, Y. Li, Z. Qiu, and C. Xu. Stable diffusion is unstable, 2023. arXiv: 2306.02583 [cs.CV].
- [10] M. Du, S. K. Pentyala, Y. Li, and X. Hu. Towards generalizable deep-fake detection with locality-aware autoencoder. 2020. DOI: 10.1145/3340531.3411892. URL: <https://doi.org/10.1145/3340531.3411892>.
- [11] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz. Leveraging frequency analysis for deep fake image recognition, 2020. URL: <http://proceedings.mlr.press/v119/frank20a.html>.
- [12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014. arXiv: 1406.2661 [stat.ML].
- [13] Q. Gu, S. Chen, T. Yao, Y. Chen, S. Ding, and R. Yi. Exploiting fine-grained face forgery clues via progressive enhancement learning, 2021. URL: <https://arxiv.org/abs/2112.13977>.
- [14] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: a dataset and benchmark for large-scale face recognition. In *ECCV 2016*, eccv 2016 edition, August 2016. URL: <https://www.microsoft.com/>

- en-us/research/publication/ms-celeb-1m-dataset-benchmark-large-scale-face-recognition-2/.
- [15] A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic. Lips don't lie: a generalisable and robust approach to face forgery detection, 2020. URL: <https://arxiv.org/abs/2012.07657>.
- [16] S. Jia, X. Li, and S. Lyu. Model attribution of face-swap deepfake videos, 2022. arXiv: 2202.12951 [cs.CV].
- [17] L. Jiang, R. Li, W. Wu, C. Qian, and C. C. Loy. Deepforensics-1.0: a large-scale dataset for real-world face forgery detection, 2020. URL: <https://arxiv.org/abs/2001.03024>.
- [18] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks, 2019. arXiv: 1812.04948 [cs.NE].
- [19] D. E. King. Dlib-ml: a machine learning toolkit. *Journal of Machine Learning Research*, 10(60):1755–1758, 2009. URL: <http://jmlr.org/papers/v10/king09a.html>.
- [20] J. Li, H. Xie, J. Li, Z. Wang, and Y. Zhang. Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection, 2021. URL: <https://arxiv.org/abs/2103.09096>.
- [21] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo. Face x-ray for more general face forgery detection, 2020. URL: https://openaccess.thecvf.com/content_CVPR_2020/html/Li_Face_X-Ray_for_More_General_Face_Forgery_Detection_CVPR_2020_paper.html.
- [22] Y. Li and S. Lyu. Exposing deepfake videos by detecting face warping artifacts, 2019. URL: https://openaccess.thecvf.com/content_CVPRW_2019/html/Media_Forensics/Li_Exposing_DeepFake_Videos_By_Detecting_Face_Warping_Artifacts_CVPRW_2019_paper.html.

- [23] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu. Celeb-df: a large-scale challenging dataset for deepfake forensics, 2020. URL: https://openaccess.thecvf.com/content_CVPR_2020/html/Li_Celeb-DF_A_Large-Scale_Challenging_Dataset_for_DeepFake_Forensics_CVPR_2020_paper.html.
- [24] J. Liang, H. Shi, and W. Deng. Exploring disentangled content information for face forgery detection, 2022. URL: <https://arxiv.org/abs/2207.09202>.
- [25] F. Liu. On the detection of digital face manipulation, 2022. URL: <https://openreview.net/forum?id=wM4X81EnKu>.
- [26] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, and N. Yu. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain, 2021. URL: <https://arxiv.org/abs/2103.01856>.
- [27] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- [28] Y. Luo, Y. Zhang, J. Yan, and W. Liu. Generalizing face forgery detection with high-frequency features, 2021. URL: https://openaccess.thecvf.com/content/CVPR2021/html/Luo_Generalizing_Face_Forgery_Detection_With_High-Frequency_Features_CVPR_2021_paper.html.
- [29] B. U. Mahmud and A. Sharmin. Deep insights of deepfake technology : a review, 2021. URL: <https://arxiv.org/abs/2105.00192>.
- [30] J. C. Neves, R. Tolosana, R. Vera-Rodriguez, V. Lopes, H. Proenca, and J. Fierrez. Ganprintr: improved fakes and evaluation of the state of the art in face manipulation detection. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):1038–1048, August 2020. ISSN: 1941-0484.

- DOI: 10.1109/jstsp.2020.3007250. URL: <http://dx.doi.org/10.1109/JSTSP.2020.3007250>.
- [31] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos, 2019. URL: <https://arxiv.org/abs/1906.06876>.
- [32] H. H. Nguyen, J. Yamagishi, and I. Echizen. Capsule-forensics: using capsule networks to detect forged images and videos. *CoRR*, abs/1810.11215, 2018. arXiv: 1810.11215. URL: <http://arxiv.org/abs/1810.11215>.
- [33] K. R. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *Proceedings of the 28th ACM International Conference on Multimedia, MM '20*. ACM, October 2020. DOI: 10.1145/3394171.3413532. URL: <http://dx.doi.org/10.1145/3394171.3413532>.
- [34] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao. Thinking in frequency: face forgery detection by mining frequency-aware clues, 2020. URL: https://www.ecva.net/papers/eccv_2020/papers_ECCV/html/1486_ECCV_2020_paper.php.
- [35] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models, 2022. arXiv: 2112.10752 [cs.CV].
- [36] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner. Faceforensics++: learning to detect manipulated facial images, 2019. URL: https://openaccess.thecvf.com/content_ICCV_2019/html/Rossler_FaceForensics_Learning_to_Detect_Manipulated_Facial_Images_ICCV_2019_paper.html.

- [37] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. *CoRR*, abs/1803.09179, 2018. arXiv: 1803.09179. URL: <http://arxiv.org/abs/1803.09179>.
- [38] E. Sarkar, P. Korshunov, L. Colbois, and S. Marcel. Are gan-based morphs threatening face recognition? In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2959–2963, 2022. DOI: 10.1109/ICASSP43922.2022.9746477. URL: <https://doi.org/10.1109/ICASSP43922.2022.9746477>.
- [39] E. Sarkar, P. Korshunov, L. Colbois, and S. Marcel. Vulnerability analysis of face morphing attacks from landmarks and generative adversarial networks. *arXiv preprint*, October 2020. URL: <https://arxiv.org/abs/2012.05344>.
- [40] K. Shiohara and T. Yamasaki. Detecting deepfakes with self-blended images, 2022. URL: https://openaccess.thecvf.com/content/CVPR2022/html/Shiohara_Detecting_Deepfakes_With_Self-Blended_Images_CVPR_2022_paper.html.
- [41] C. Shuai, J. Zhong, S. Wu, F. Lin, Z. Wang, Z. Ba, Z. Liu, L. Cavallo, and K. Ren. Locate and verify: a two-stream network for improved deepfake detection, 2023. URL: <https://arxiv.org/abs/2309.11131>.
- [42] Z. Sun, Y. Han, Z. Hua, N. Ruan, and W. Jia. Improving the efficiency and robustness of deepfakes detection through precise geometric features, 2021. URL: https://openaccess.thecvf.com/content/CVPR2021/html/Sun_Improving_the_Efficiency_and_Robustness_of_Deepfakes_Detection_Through_Precise_CVPR_2021_paper.html.

- [43] M. Tan and Q. Le. EfficientNet: rethinking model scaling for convolutional neural networks. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, September 2019. URL: <https://proceedings.mlr.press/v97/tan19a.html>.
- [44] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: real-time face capture and reenactment of rgb videos, 2020. arXiv: 2007.14808 [cs.CV].
- [45] J. Wang, Z. Wu, W. Ouyang, X. Han, J. Chen, S.-N. Lim, and Y.-G. Jiang. M2tr: multi-modal multi-scale transformers for deepfake detection, 2022. arXiv: 2104.09770 [cs.CV].
- [46] R. Wang, F. Juefei-Xu, L. Ma, X. Xie, Y. Huang, J. Wang, and Y. Liu. Fakespotter: a simple yet robust baseline for spotting ai-synthesized fake faces, 2019. URL: <https://arxiv.org/abs/1909.06122>.
- [47] X. Wang, Y. Li, H. Zhang, and Y. Shan. Towards real-world blind face restoration with generative facial prior. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [48] Z. Yan, Y. Zhang, Y. Fan, and B. Wu. Ucf: uncovering common features for generalizable deepfake detection, 2023. URL: <https://arxiv.org/abs/2304.13949>.
- [49] Z. Yan, Y. Zhang, X. Yuan, S. Lyu, and B. Wu. Deepfakebench: a comprehensive benchmark of deepfake detection, 2023. URL: <https://arxiv.org/abs/2307.01426>.
- [50] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu. Multi-attentional deepfake detection, 2021. arXiv: 2103.02406 [cs.CV].
- [51] T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong, and W. Xia. Learning self-consistency for deepfake detection, 2021. arXiv: 2012.09311 [cs.CV].

-
- [52] X. Zhu, H. Wang, H. Fei, Z. Lei, and S. Z. Li. Face forgery detection by 3d decomposition, 2020. URL: <https://arxiv.org/abs/2011.09737>.
- [53] Y. Zhu, Q. Li, J. Wang, C. Xu, and Z. Sun. One shot face swapping on megapixels, 2022. arXiv: 2105.04932 [cs.CV].

Acknowledgements

This thesis is the result of my internship within DuckDuckGoose, a Dutch deep fake detection company. Many thanks go to Joris and Dimitris for the valuable opportunity. Thanks to Professor Salti for agreeing to be the thesis advisor for this thesis. Finally, I would like to thank my family and friends for supporting me throughout this journey.