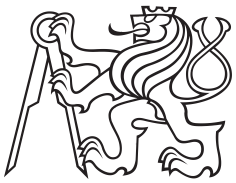
Master Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Few-Shot Learning of a Deepfake Detector

Bc. Vojtěch Brejtr

Supervisor: Ing. Vojtěch Franc Ph.D.
May 2024

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Brejtr Vojtěch** Personal ID number: **491929**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Medical Electronics and Bioinformatics**

Specialisation: **Image processing**

II. Master’s thesis details

Master’s thesis title in English:

Few-Shot Learning of Deepfake Detector

Master’s thesis title in Czech:

Detektor deepfakes učený z malého množství příkladů

Guidelines:

The aim of this master's project is to create a sophisticated deepfake detector applicable to arbitrary domain of images including medical. This detector will be developed using a neural network trained on a limited set of deepfake examples. The primary objective is to devise and implement a learning algorithm that is not only efficient in terms of time but also demands minimal or no human intervention. The part of the project is to create a new benchmark contraining real and generated medical images. The developed deepfake detector will be compared against existing state-of-the-art methods using established benchmarks and the new benchmark of medical images created in the project.
Guidelines:
1. Familiarize yourself with the published work on the deepfake detection; consider prominent conferences such as CVPR and ICCV.
2. Review and summarize the core principles underyling state-of-the-art methods for deepfake detection.
3. Design and implement a few-shot learning algorithm of deepfake detector.
4. Create a benchmark of generated and real medical images for testing deepfake detectors.
5. Verify functionality of the learning algorithm and compare its performance metrics, including detection accuracy, time efficiency, and the level of human supervision required during the learning process, against the established state-of-the-art methods.

Bibliography / sources:

[1] Zhao et al. Multi-Attentional Deepfake Detection. CVPR 2021
[2] Wang et al. DeepFake Disrupter: The Detector of DeepFake Is My Friend. CVPR 2022.
[3] Narayan et al. DF-Platter: Multi-Face Heterogeneous Deepfake Dataset. CVPR 2023.
[4] Solaiyappan et al. Machine learning based medical image deepfake detection: A comparative study. Machine Learning with Applications. 2022.

Name and workplace of master’s thesis supervisor:

Ing. Vojtěch Franc, Ph.D. Machine Learning FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **14.02.2024** Deadline for master's thesis submission: **24.05.2024**

Assignment valid until: **21.09.2025**

Ing. Vojtěch Franc, Ph.D.	prof. Dr. Ing. Jan Kybic	prof. Mgr. Petr Páta, Ph.D.
Supervisor’s signature	Head of department’s signature	Dean’s signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

Acknowledgements

I would like to thank my supervisor Ing. Vojtěch Franc, Ph.D. for the guidance during the master thesis.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

In Prague, 24. May 2024

Bc. Vojtěch Brejtr

Abstract

This thesis explores the effectiveness of few-shot learning in detecting image and video deepfakes. For this purpose, we use embedding from already pre-trained models, namely FaRL [1], ArcFace [2], and ResNet-50 which was pre-trained on ImageNet [3, 4], with a simple classifier built on top of them.

Our methods are tested on several commonly used deepfake datasets, namely FaceForensics++ [5] and DFDC [6]. In addition, we created two novel datasets, one from Instagram influencers and a second generated from MRI brain scans.

Our model is capable of achieving results that are comparable to the SOTA on the FF++ dataset while still performing well on the DFDC dataset. On our novel influencer dataset, the method can achieve near-perfect detection. Our approach is, however, not capable of generalizing onto the medical dataset. We implement a small CNN as an alternative to our approach to be used in tasks where the embedding approach does not work.

Our approach shows that to achieve good performance in deepfake detection, we do not need large quantities of training data. Only a few videos and/or faces are sufficient as long as we use a good underlying embedding model. However, the proposed method is not guaranteed to generalize to other generators well when the generative model is not present in the training data.

Keywords: deepfakes, deepfakes detection, few-shot learning

Supervisor: Ing. Vojtěch Franc Ph.D.

Abstrakt

Tato práce zkoumá efektivitu few-shot učení při detekci deepfake obrázků a videí. K tomuto účelu používáme embedding z již předtrénovaných modelů, konkrétně FaRL [1], ArcFace [2] a ResNet-50, který byl předtrénován na ImageNet [3, 4], a nad nimi postavený jednoduchý klasifikátor.

Naše metody jsou testovány na několika běžně používaných deepfake datasetech, konkrétně na FaceForensics++ [5] a DFDC [6]. Kromě toho jsme vytvořili dvě nové datové sady, jednu z influencerů na Instagramu a druhou generovanou ze snímků mozku na magnetické rezonanci.

Náš model je schopen dosáhnout výsledků srovnatelných s modelem SOTA na datové sadě FF++ a zároveň dosahuje dobrých výsledků na datové sadě DFDC. Na naší nové datové sadě influencerů dokáže metoda dosáhnout téměř dokonalé detekce. Náš přístup však není schopen generalizace na lékařském datasetu. Jako alternativu k našemu přístupu implementujeme malou síť CNN, kterou lze použít v úlohách, kde přístup založený na embeddingu nefunguje.

Náš přístup ukazuje, že k dosažení dobrého výkonu pro detekci deepfaků nepotřebujeme velké množství trénovacích dat. Stačí pouze několik videí a/nebo obličejů, pokud použijeme dobrý embeddingový model. Není však zaručeno, že navržená metoda bude dobře generalizovat na jiné generátory, pokud se v trénovacích datech nenachází generativní model.

Klíčová slova: deepfake, detekce deepfaků, few-shot learning

Překlad názvu: Detektor deepfakes učený z malého množství příkladů

Contents

1 Introduction 1
1.1 Motivation . . . 1
1.2 Description of the problem . . . 2
1.3 Contributions of the thesis . . . 2
1.4 Structure of the thesis . . . 3

2 Deepfakes 5
2.1 Types of deepfakes . . . 5
2.1.1 Editing . . . 5
2.1.2 Replacement and reenactment 6
2.1.3 Full synthesis . . . 6
2.2 Generative methods . . . 7
2.2.1 Face2Face . . . 7
2.2.2 Neural Textures . . . 9
2.2.3 StyleGAN . . . 10
2.3 Real world examples . . . 11

3 Existing deepfake detectors 13
3.1 MADD . . . 13
3.2 SFDG . . . 14
3.3 Implicit identity driven detection 15
3.4 MARLIN . . . 15
3.5 Human detection performance . . 16
3.5.1 FF++ dataset . . . 16
3.5.2 DFDC dataset . . . 17

4 Proposed detection model 19
4.1 Few-shot learning . . . 19
4.1.1 Empirical risk minimization . 19
4.1.2 Approaches . . . 20
4.1.3 Data . . . 20
4.1.4 Model . . . 21
4.1.5 Algorithm . . . 21
4.2 Embedding models . . . 22
4.2.1 FaRL . . . 22
4.2.2 ArcFace . . . 23
4.2.3 ResNet - ImageNet . . . 24
4.3 Classifiers . . . 24
4.3.1 Logistic regression . . . 25
4.3.2 SVM . . . 25

5 Existing benchmarks and metrics 27
5.1 FaceForensics++ . . . 27
5.2 DFDC . . . 29
5.3 Dataset comparison . . . 30
5.4 Evaluation metrics . . . 30
Basic metrics . . . 30

6 Novel benchmarks 33
6.1 Instagram influencer dataset . . . 33
6.1.1 Influencer selection . . . 33
6.1.2 Image selection . . . 34
6.1.3 Dataset summary . . . 34
6.2 Medical Fake Image Dataset . . . 36
6.2.1 Slice selection . . . 36
6.2.2 Fake tumor generation . . . 36

7 Experiments and results 39
7.1 Face extraction . . . 39
7.2 Dataset usage . . . 39
7.2.1 Subset of FaceForensics++ . . 39
7.2.2 Subset of DFDC . . . 40
7.3 Evaluation protocol . . . 40
7.3.1 Training setup . . . 40
7.3.2 Testing setup . . . 40
7.4 FF++ results . . . 41
7.4.1 Initial embeddings . . . 41
7.4.2 Single generative model . . . 44
7.4.3 Leave-one-out . . . 52
7.4.4 Comparison with human observers . . . 52
7.5 DFDC results . . . 53
7.5.1 Comparison of the proposed method with human performance on the DFDC . . . 53
7.6 Fake Instagram Influencers Dataset results . . . 54
7.6.1 Initial embedding . . . 54
7.6.2 Direct transformer prompting 54
7.6.3 Leave-one-identity-out experiment . . . 55
7.6.4 Detection performance versus the number of training images on FIID . . . 56
7.6.5 Variable test size of the FIID dataset . . . 57
7.6.6 Summary . . . 57
7.7 Medical dataset results . . . 58
7.7.1 Initial embedding . . . 58
7.7.2 Training a small convolutional neural network . . . 59
7.7.3 Model performance based on the variable training size . . . 60
7.8 Comparison with state-of-the-art 60
7.8.1 SOTA on the FF++ dataset 61
7.8.2 SOTA on the DFDC dataset 61

8 Discussion **63**
8.1 Efficiency of few-shot learning for deepfake detection 63
8.2 Cross-domain applicability of embedding models 64
8.3 Limitations of the thesis 64
8.4 Suggested future research 64
9 Conclusion **65**
Bibliography **67**

Figures

1.1 Examples of deepfakes 1
1.2 Deepfake of Zelenskyy 2

2.1 Examples of face editing 5
2.2 Example of face swapping 6
2.3 Example of full face synthesis . . . 7
2.4 Overview of the Face2Face method 7
2.5 Example of the deepfakes generated by the Face2Face method 8
2.6 Overview of the Neural Textures method 9
2.7 Examples of faces generated by the Neural Textures 9
2.8 Overview of the StyleGAN method 10
2.9 Example of faces generated by StyleGAN 10
2.10 Popularity of the term "deepfake" 11
2.11 Examples of AI-generated influencers 12

3.1 Structure of the MADD architecture 13
3.2 Structure of the SFDG architecture 14
3.3 Structure of the DG-SF^3 architecture 14
3.4 Structure of the MARLIN architecture 16
3.5 Human observer performance on the FF++ dataset 16
3.6 Human observer performance on the DFDC dataset 17

4.1 Few-shot learning principles 20
4.2 Framework of the FaRL pre-training 22
4.3 Example data of the LaPa dataset 23
4.4 Example data of the LAION-Face dataset 23
4.5 Examples from the ImageNet dataset 24

5.1 Generative methods in the FF++ dataset 27
5.2 Examples of from the FF++ . . . 28
5.3 Distribution of the FF++ dataset 29
5.4 Examples from the DFDC 29
5.5 Example of the ROC-curve 32

6.1 Diagonal resolution of the FIID. 34
6.2 Examples of fake influencers in the FIID. 35
6.3 Examples of real influencers in the FIID. 35
6.4 Example sample from the BRATS2017 dataset with the segmentation mask. 36
6.5 Example samples from the medical dataset 37

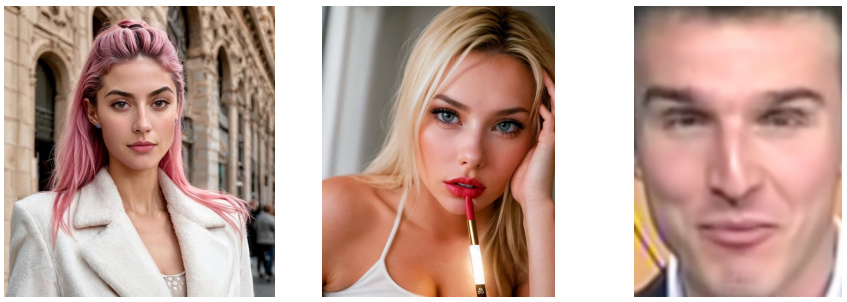
7.1 FF++ embedding using FaRL . . 41
7.2 FF++ embedding using ArcFace 42
7.3 FF++ embedding using the ImageNet 43
7.4 FF++ FaRL results 47
7.5 FF++ ArcFace results 48
7.6 FF++ ImageNet results 49
7.7 FF++ SVM-FaRL results 50
7.8 FF++ Comparison 51
7.9 Initial embedding of the FIID . . 54
7.10 Distribution of the AUC for the LOIO experiment on the FIID 56
7.11 Depence of the AUC given the size of the training size. 56
7.12 AUC on FIID with variable test size 57
7.13 ResNet-50 ImageNet T-SNE embedding of the medical dataset . 58
7.14 Progression of training on the medical dataset 59
7.15 Dependence of model performance based on the test set given the size of the medical training dataset. 60
7.16 ROC of the top 40 DFDC sumbissions 62

Tables

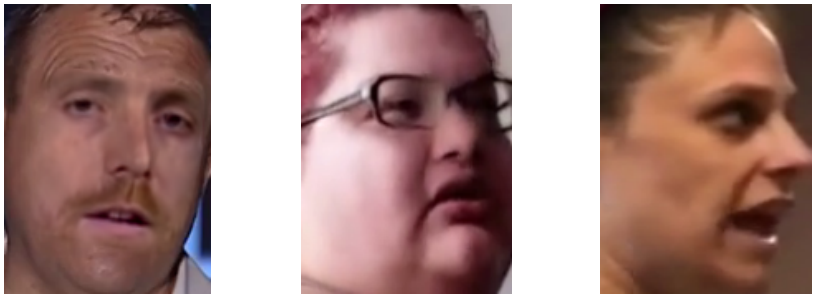
6.1 Summary of the FIID. 34

7.1 FF++ FaRL AUC 44
7.2 FF++ FaRL FPR@TPR(90) . . . 44
7.3 FF++ ArcFace AUC 44
7.4 FF++ ArcFace FPR@TPR(90) . 45
7.5 FF++ ImageNet AUC 45
7.6 FF++ ImageNet FPR@TPR(90) 45
7.7 FF++ SMV-FaRL AUC 45
7.8 FF++ CNN AUC 46
7.9 LOO FF++ AUC 52
7.10 LOO FF++ FPR@TPR(90) . . 52
7.11 Model accuracy on the FF++ with human observer setup 53
7.12 Results of the FaRL + logistic regression of the DFDC dataset given several training sizes. 53
7.13 FIID influencer overview 55
7.14 Summary table of the SOTA on FF++ compared with our method 61
7.15 Summary table of the SOTA on DFDC compared with our method 62

Chapter 1
Introduction



(a) : fit_aitana **(b) :** Natalia Novak **(c) :** FF++ (1)



(d) : FF++ (2) **(e) :** DFDC (1) **(f) :** DFDC (2)

Figure 1.1: Examples of deepfakes. Figures (a) and (b) are accounts of [7, 8], which explicitly state that their content is AI-generated. Figures (c) and (d) are from the FaceForensics++ dataset [5] and (e) and (f) from the DFDC dataset [6].

1.1 Motivation

Deepfakes are synthetic media (images, audio, or videos) that are generated using standard computer graphics methods or modern machine learning models. The term "deepfake" itself is a combination of the words "fake" and "deep" (referencing deep learning). These methods transfer, change, or entirely fabricate the identity of the given subject, examples of which can be seen in Figure 1.1.

The detection of deepfakes is quickly becoming a major issue as the gen-

erative methods used for their creation become better and more available to a wider audience. Their usage is already becoming more prominent (see Section 2.3). We are already seeing attempts at political sabotage, as seen in the presidential election in Turkey, fake videos of Volodymyr Zelenskyy (see Figure 1.2) with false claims of surrender, or many AI-generated personalities on social media sites like Instagram or Twitter.

Deepfake detection is therefore of utmost importance as their existence poses a great danger in an increasingly digital world.

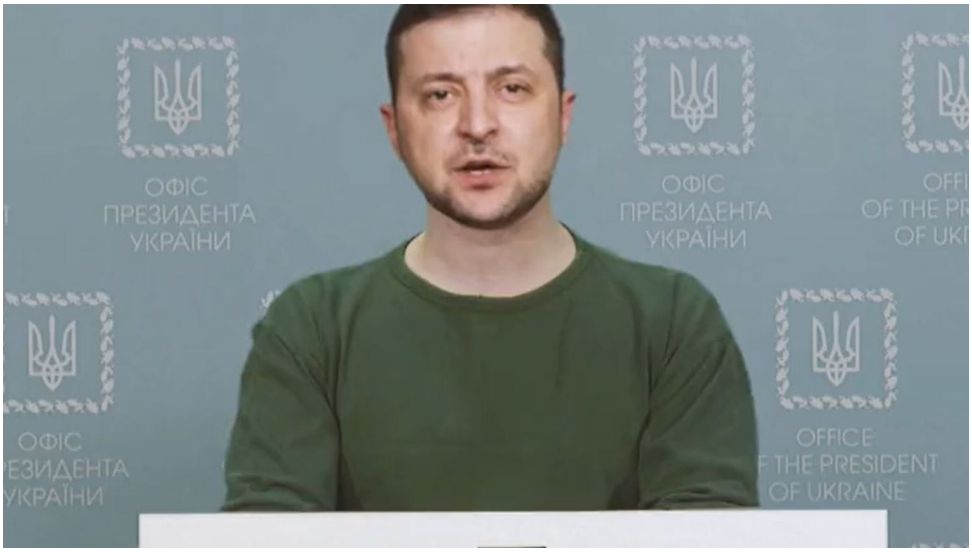


Figure 1.2: Deepfake video of Ukraine president Volodymyr Zelenskyy. Taken and edited from [9].

1.2 Description of the problem

One of the main issues of deepfake detection is a relatively small amount of samples from which a decision should be made. We usually deal with a single video, sometimes only a single photo, and as such, datasets containing only several generative methods may not accurately approximate real-world tasks.

The base assumption is that each identity (video or photo of an individual) is likely generated by a different method. While these methods are likely to share a common structure, their specifics differ.

Another issue is the ever-changing landscape of generative methods. Collecting data for a large detection model and its subsequent training is a time-consuming process. The model also has a high chance of becoming obsolete within a year.

1.3 Contributions of the thesis

The goal of the thesis is to create and test a simple pipeline, which allows for accurate deepfake detection from a minimal amount of samples. The main

contributions of the thesis are the following:

- A novel deepfake detection model utilizing few-shot learning approaches called FSDF (Few-Shot DeepFake detection). FSDF uses a pre-trained embedding model, followed by a simple classifier trained on a few examples.
- Search for a suitable pre-trained model that will be used for embedding. The models' effectiveness will be tested on several commonly used datasets to find the best embedding.
- Comparison of the effectiveness of the model with state-of-the-art (SOTA). Specifically, we explore the following issues: 1. How does the FSDF performance depend on the pre-trained model. 2. How the performance of the FSDF depends on the number of training examples. 3. How does the FSDF trained on data from one generator generalizes to another generator.
- Evaluation of the FSDF on the real-world examples. To this end, a we generate a dataset from publicly available photos found on the pages of the Instagram models (FIID). Secondly, an MRI brain scan dataset is generated (MFID).

1.4 Structure of the thesis

The first part of the thesis should familiarize the reader with the overall problem of deepfake detection and working with a small amount of training samples.

The results of the findings are presented in the second part of the thesis, alongside two novel datasets generated specifically for this thesis.

General content of the chapters is as follows:

- **Deepfakes** - Overview of the different types of deepfakes, commonly used generative methods, and several real-world examples of deepfakes.
- **Existing deepfake detectors** - Overview of deepfake detectors, which are not designed for few-shot learning. Not all models presented are SOTA, but provide the reader with a better understanding of possible approaches.
- **Proposed detection model** - This chapter describes the few-shot learning problem and our proposed model **FSDF** (Few-Shot DeepFake detector).
- **Existing benchmarks and metrics** - Description of the used existing benchmarks, namely FaceForensics++ [5] and DFDC [6]. Included are also the metrics, which are used for the evaluation of the given datasets and our novel datasets.

- **Novel benchmarks** - Chapter describes the selection and creation process of our two novel benchmarks. The first is generated from images found on the pages of Instagram influencers (FIID), and the second is generated from brain MRI scans (MFID).
- **Experiments and results** - In this chapter, we describe our approach's (FSDF) overall evaluation process and performance on the two existing and two novel benchmarks. Furthermore, we train a small CNN for the FF++ and MFID to compare our approach with a standard deep-learning method.

Chapter 2
Deepfakes

For the purpose of this thesis, we will consider deepfakes as all computer-generated images that serve the purpose of fooling humans into thinking that the image or video is real instead of focusing solely on images generated by deep neural networks.

This encompasses images of several different types e.g., be it videos generated by switching faces between two people, turning an image into a video, fabrication of an entirely new identity, or combining medical images to change the diagnosis.

2.1 Types of deepfakes

There exist many different types of deepfakes, some of which overlap or complement each other.

The division is somewhat arbitrary. However, it shows different possibilities for image manipulation and even synthesis.

2.1.1 Editing

For image editing, the original image is encoded into an embedding vector. Several values are changed based on the desired output identity. For example, the subject in the photo can be aged up or down, the color of their hair changed, glasses added, etc [10] (see Figure 2.1).

Another option is the change of pose, be it only parts of the face, like lips or eyes to simulate speech, or movement of the entire face [11].

Figure 2.1: Examples of face editing. Taken and edited from [12].

2.1.2 Replacement and reenactment

Deepfakes that fall into this category are generated by transferring either the face shape (lips, eyes, etc.) from the source face to the target face or swapping the entirety of the face for another one [13] (see Figure 2.2).

Face swapping can also be done by simple methods, like Poisson blending [14], instead of using neural networks.

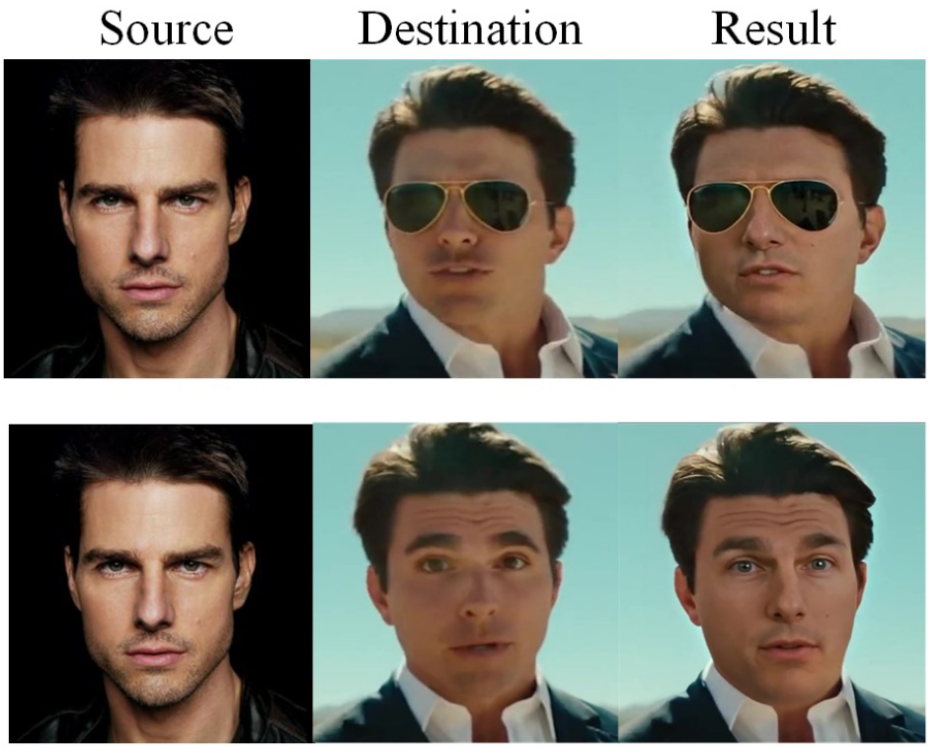


Figure 2.2: Example of face swapping. Taken and edited from [13].

2.1.3 Full synthesis

Full image synthesis (see Figure 2.3) is a process in which the entire face (sometimes even with background) is created by a generative model. This can be done both to create an entirely new identity or to approximate an already existing one.

Partial synthesis is at the core of essentially all concurrent generative models (latent diffusion [15], GAN [16]). We are already seeing generated faces, which are indistinguishable from real faces by humans.

Furthermore, methods like Contrastive Language-Image Pre-training (CLIP) [17], which combine language, with image features, allow for quick and accurate manipulation or generation of images.

Model similar to CLIP is used as the main backbone of this thesis, namely the General Facial Representation Learning in a Visual-Linguistic Manner (FaRL) [1] (see section 4.2.1).

Figure 2.3: Example of full face synthesis. Taken and edited from [18].

2.2 Generative methods

Models in this section serve as an overview of commonly used methods, ranging from non-learned to current diffusion models. Most of the methods described are used by the benchmarks used in the thesis.

2.2.1 Face2Face

The Face2Face transfer method, proposed in [19] is a real-time method which utilizes pre-computed source identity, which is mapped onto the target identity. This process is outlined in Figure 2.4, with resulting faces in Figure 2.5.

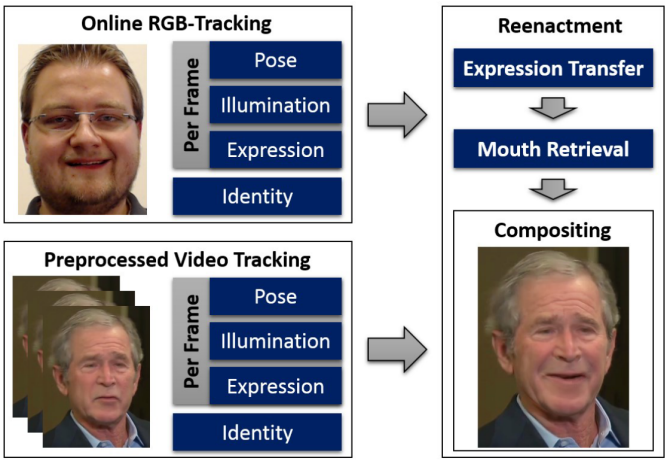


Figure 2.4: Overview of the Face2Face method. Taken and edited from [19].

The most similar face is found based on pose, lighting, and expression. The first two dimensions of multi-linear PCA represent the face shape and skin

reflectance, while the third represents facial expressions. The face is then parameterized as

$$\mathcal{M}_{\text{geo}}(\alpha, \delta) = \mathbf{a}_{\text{id}} + E_{\text{id}} \cdot \alpha + E_{\text{exp}} \cdot \delta, \tag{2.1}$$

$$\mathcal{M}_{\text{alb}}(\beta) = \mathbf{a}_{\text{alb}} + E_{\text{alb}} \cdot \beta, \tag{2.2}$$

where $\mathbf{a}_{\text{id}}$ and $\mathbf{a}_{\text{alb}}$ are the average shape and reflectance respectively, with E_{id} and E_{exp} being the actual shape and expression [19].

In addition, parameters representing rigid transformation and perspective transformation are introduced. The vector of all parameters $\mathcal{P}$ is constructed.

Next, the problem is formulated as energy minimization task, minimizing photometric alignment error as

$$E_{\text{col}}(\mathcal{P}) = \frac{1}{|\mathcal{V}|} \sum_{p \in \mathcal{V}} ||C_S(p) - C_I(p)||_2, \tag{2.3}$$

where C_S is the synthesized image and C_I is the input image, given all visible pixels in the vector $\mathcal{V}$.

Secondly, the distance between found features is minimized as

$$E_{\text{lan}}(\mathcal{P}) = \frac{1}{|\mathcal{F}|} \sum_{f_j \in \mathcal{F}} w_{\text{conf},j} ||f_j - \Pi(\Phi(v_j)||_2^2, \tag{2.4}$$

where f_j are the found feature points, with their corresponding confidence weights $w_{\text{conf},j}$ and $v_j = \mathcal{M}_{geo}(\alpha, \delta)$ is the face prior. The value of E_{lan} smooths the energy function [19].

In addition to the feature and photometric minimization, the parameters are also regularized.

Extra focus is then given to the mouth region, which is treated separately but in a similar fashion to the entire shape [19].

Figure 2.5: Example of the deepfakes generated by the Face2Face method. Taken and edited from [19].

2.2.2 Neural Textures

The Deferred Neural Rendering method as proposed in [20], combines information from uv-map, extracted neural textures and identity. The process is outlined in Figure 2.6 with generated faces in Figure 2.7.

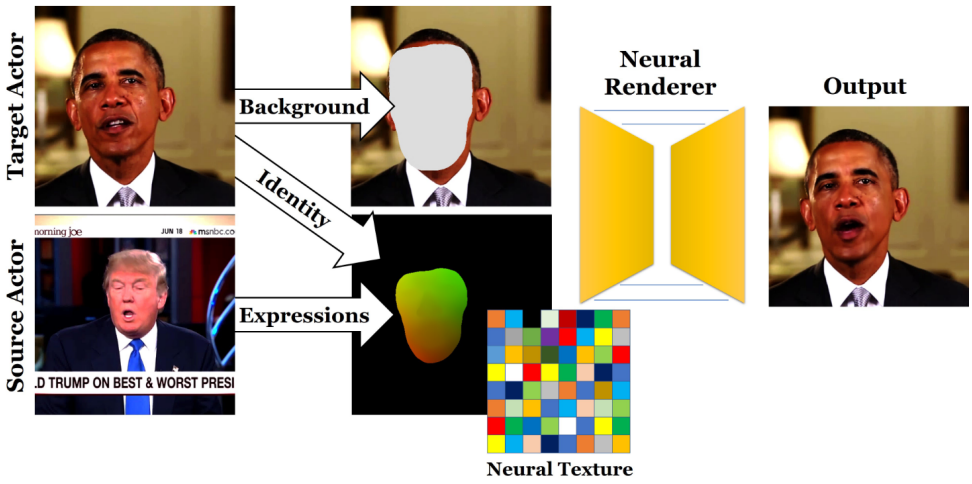


Figure 2.6: Overview of the Neural Textures method. Taken and edited from [20].

During training, both the neural texture map $\mathbf{T}$ and renderer $\mathcal{R}$ are learned together, to minimize some variation of photometric re-rendering loss $\mathcal{L}$ as

$$\mathbf{T}^*, \mathcal{R}^* = \underset{\mathbf{T},\mathcal{R}}{\operatorname{argmin}} \sum_{k=1}^{N} \mathcal{L}(\mathbf{I}_k, \mathbf{p}_k | \mathbf{T}, \mathcal{R}). \tag{2.5}$$

Neural textures are an extension of standard textures maps used in computer graphics. Instead of using handcrafted features, they are extracted automatically with a neural network at multiple resolutions, forming a Laplacian pyramid [20].

The rendered has an architecture similar to a classical U-Net with additional inputs.

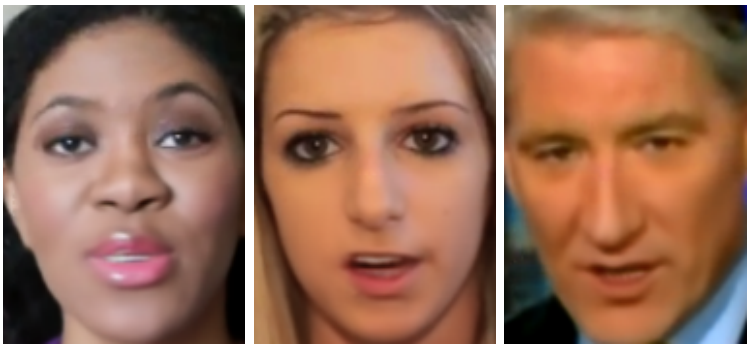


Figure 2.7: Examples of faces generated by the Neural Textures in the FF++ dataset [20, 5].

2.2.3 StyleGAN

In contrast to standard GAN [16], StyleGAN [21] introduces additional mapping to intermediate latent space $\mathcal{W}$. Instead of generating from random latent noise, the starting constants are learned.

In addition, adaptive instance normalization layers (AdaIN) are introduced, defined as

$$\text{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i}, \tag{2.6}$$

where $\mathbf{x}_i$ are the feature maps and $\mathbf{y}_{(\cdot)}$ is the style vector, which controls them.

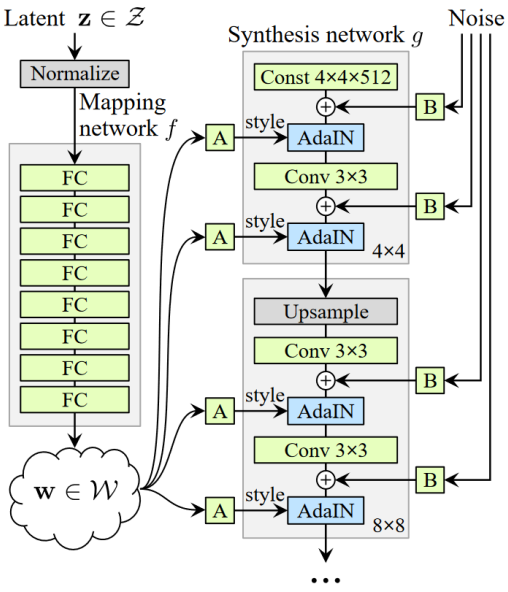


Figure 2.8: Overview of the StyleGAN method. Taken and edited from [21].

In addition, noise is applied to all layers. This process increases the variance and realism of the output image. The synthesis of the image can be controlled by modifying the styles vector $\mathbf{y}$ added through said noise [21]. The model pipeline is shown in Figure 2.8 with resulting faces generated by StyleGan in Figure 2.9.

Figure 2.9: Example of faces generated by StyleGAN. Taken and edited from [21].

2.3 Real world examples

In this section, we present several real-world examples of incidents involving deepfakes. As we can see in Figure 2.10, most of the public interest in the term "**deepfakes**" is caused by deepfake pornography. The overall usage of the term has been increasing for several years.

As the usage of the generative models increases, we are starting to see an ever-increasing amount of face forgeries. Tasks that were impossible several years ago are now commonplace.

It is important to note that the quality of the best current deepfakes is possibly unknown since the most advanced examples could not be detected.

Psychological warfare in Ukraine-Russia conflict

During the conflict in Ukraine, both sides utilized modern technology to achieve an advantage. This includes the use of deepfakes. We have already seen deepfake videos of both country leaders claiming the surrender of their country.

However, since the video quality was somewhat questionable, the fake was quickly detected, but not before being spread around as a real video [22].

Political campaign in Turkey's presidential election

In 2023, during Turkey's presidential election, a video of one of the opposition candidates began circulating on social media. The content of the video was sexually explicit, causing the candidate to drop out of the race [23].

Celebrity deepfakes

Over the last several years, several celebrities had their likeness stolen by deepfake creators. Namely, these were popular Twitch streamer QTCinderella [24], Indian actress Rashmika Mandanna [25], American pop singer Taylor Swift [26], or Italian prime minister Giorgie Meloni [27]. As we can see in Figure 2.10 below, the generation of sexually explicit content is the most important force behind public interest in deepfake technology.

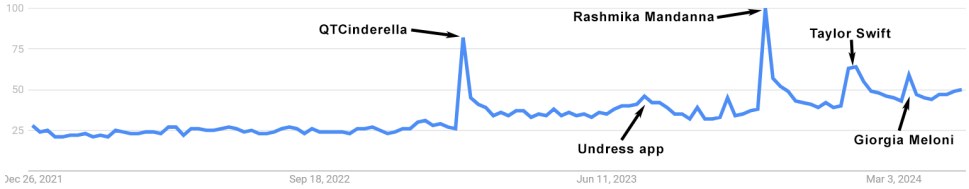


Figure 2.10: Popularity of the term "deepfake" with corresponding events causing spikes. As we can see, all major events, which caused a rise in popularity were AI-generated porn videos of famous celebrities. The graph is scaled such that 100 corresponds to the highest interest in the search term since 2004. Data source: Google Trends (`https://www.google.com/trends`).

AI-generated influencers on social media

In recent years, many completely generated personalities have emerged on social media sites such as Instagram or Twitter. The biggest of which has several million followers.

In general, two main types of AI-generated accounts exist. The first kind, where the main drive is the novelty of having an artificial person be on screen like in the case of lilmiquela [28]. These accounts usually clearly state, that the content is generated.

The second type is sexually explicit accounts, which sometimes do not state that they are AI-generated. This information is usually only findable through outside means. Examples of two influencers can be seen in the Figure 2.11 below.

Figure 2.11: Examples of AI-generated influencers. Images were downloaded from Instagram posts of several accounts. First from Milla Sofia [29], second from lilmiquela [28].

Chapter 3

Existing deepfake detectors

All SOTA methods from recent years are deep learning-based. Methods usually try to exploit some commonly known weaknesses present in generative models.

The methods described in this chapter were either state-of-the-art (SOTA) when they were published or offer an interesting insight into the problem of deepfake detection. This chapter aims to give the reader an overview of several different approaches.

3.1 Multi-attentional Deepfake Detection (MADD)

The MADD method [30] utilizes several avenues for detection. Fine-grained classification is used, where each pixel is assigned a probability of being fake. In addition, attention is paid to textures, which are extracted as a residual of a shallow feature map after removing a blurred version of the map [30].

Features extracted from both the input image and the features are combined using bilinear attention pooling layer. The final classification of the video is done by aggregating results across several faces extracted from the video. The structure of the MADD is shown in Figure 3.1.

The networks achieve an accuracy of 97.60% and AUC of 0.993 on the FF++ dataset [5].

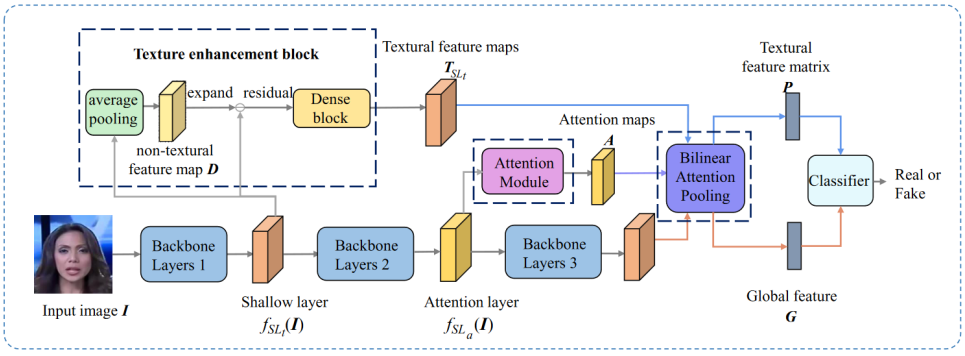


Figure 3.1: Structure of the MADD architecture. Taken and edited from [30].

3.2 Dynamic Graph Learning with Content-guided Spatial-Frequency Relation Reasoning for Deepfake Detection

This method proposes using three separate branches for feature extraction [31]. They are

1. **CAFÉ** - The Content-guided Adaptive Frequency Extraction utilizes course-granied and fine-grained discrete cosine transform.
2. **Texture** - Texture is extracted as a difference between low-depth feature maps and the original blurred image.
3. **Spatial** - Standard CNN, low-depth features used for texture extraction.

The features from all three branches are combined, and a graph is constructed from them. The output of the graph fusion network is then used in the classification itself alongside the raw output of the spatial branch [31]. The architecture of the SFDG network is shown in Figures 3.2 and 3.3.

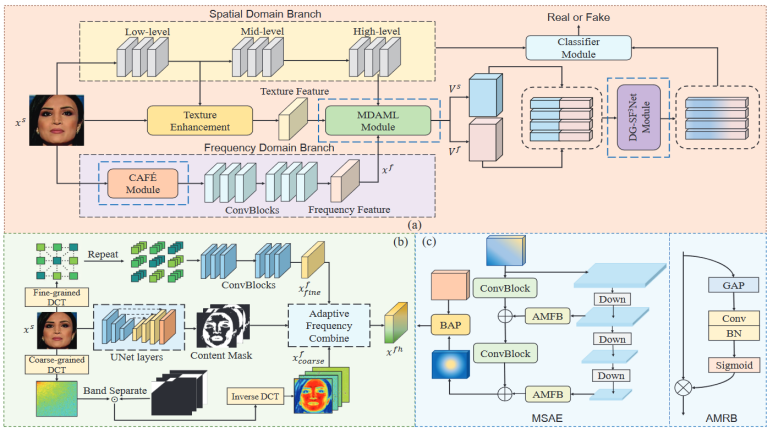


Figure 3.2: (a) Architecture of the SFDG network. (b) CAFÉ module (c) The Multiple Domains Attention map Learning network. Taken and edited from [31].

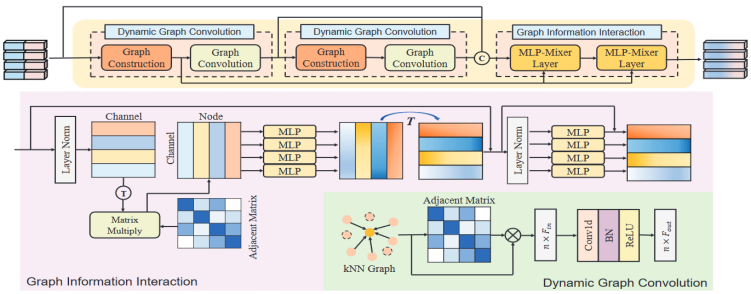


Figure 3.3: Structure of the Dynamic Spatial-Frequency Feature Fusion Network (DG-SF^{3}Net). Taken and edited from [31].

3.3 Implicit Identity Driven Deepfake Face Swapping Detection

The main innovation in this paper are the explicit identity contrast (EIC) and implicit identity contrast (IIC) loss functions [32].

The explicit identity contrast is computed as

$$\mathcal{L}_{eic} = \frac{1}{N_F}\sum_{i\in F}\delta\left(F_{im}(x_i), F_{em}(x_i)\right) - \frac{1}{N_R}\sum_{i\in R}\delta\left(F_{im}(x_i), F_{em}(x_i)\right), \tag{3.1}$$

where $\delta(\cdot,\cdot)$ is the cosine similarity, F and R are the fake images and real images respectively, F_{im} is the implicit identity embedding network and F_{em} is the generic face recognition network. The purpose of this loss is to enlarge differences between the real and the fake samples in the feature space [32].

The implicit identity contrast is for the fake class computed as

$$\mathcal{L}^{-}_{iic} = -\mathbb{E}_{x_i, y_i^*} \sim \mathcal{U}\left[\log \frac{\exp v_{y_i}^T F_{im}(x_i)/\tau}{\sum_{j=1}^{Q}\exp v_j^T F_{im}(x_i)/\tau}\right], \tag{3.2}$$

where v is the element of the matrix V, used to store normalized features of all unknown implicit identities, τ is a hyperparameter, which determines the sharpness of the probability distribution [32]. The implicit identity contrast for the real class is

$$\mathcal{L}^{+}_{iic} = -\mathbb{E}_{x_i, y_i} \sim \mathcal{K}\left[\log \frac{\exp s(\cos\theta y_i - m)}{\exp s(\cos\theta y_i - m) + \sum_{j\neq y_i}\exp s\cos\theta_j}\right], \tag{3.3}$$

where $\mathcal{K}$ is the set of samples with known implicit identities, θ the angle between normalized $F_{im}(x_i)$ and its normalized proxy of jth identity on the hypersphere. Lastly, m and s are the margin and rescaling parameters [32].

Loss for both the known and the unknown distribution is then merged as

$$\mathcal{L}_{iic} = \mathcal{L}^{+}_{iic} + \mathcal{L}^{-}_{iic}. \tag{3.4}$$

Total loss is then

$$\mathcal{L} = \mathcal{L}_{bce} + \lambda_1\mathcal{L}_{eic} + \lambda_2\mathcal{L}_{iic}, \tag{3.5}$$

where $\mathcal{L}_{bce}$ is the standard binary cross-entropy and $\lambda_{1,2}$ are the scaling hyperparameters empirically set to 0.05 and 0.1 [32].

3.4 MARLIN: Masked Autoencoder for facial video Representation LearnINg

In contrast to previously mentioned methods, which use single frame and aggregate later, MARLIN (see Figure 3.4) utilizes video sequences [33]. The encoder is pre-trained on an unlabeled dataset for a variety of different problems (facial attribute recognition, expression recognition, deepfake detection, among others...).

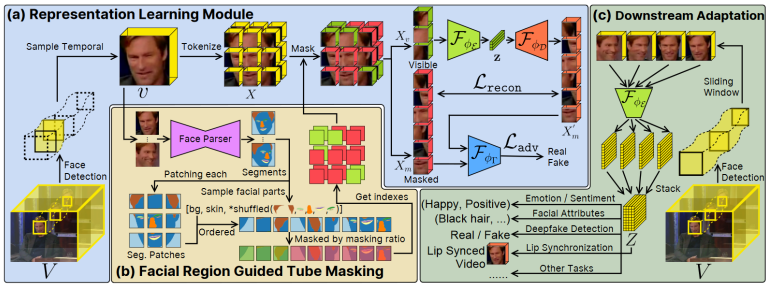


Figure 3.4: Structure of the MARLIN architecture. Taken and edited from [33].

MARLIN temporally samples the faces from the input video. The face is tokenized, and several tokens are selected. The rest is hidden. The task of the decoder is to reconstruct the missing tokens. They are compared based on the similarity to the original tokens and the adversarial loss of the discriminator [33].

The extracted features are not fully unsupervised, but specific areas (eyes, mouth, nose, etc.) or even more complex movements (emotions, lip sync) are selected. Their consistency is tracked across several frames of the video [33].

3.5 Human detection performance

To compare the performance of detection methods with human observers, a comparison study was run on both the DFDC [6] and the FF++ [5] datasets. The results should form the baseline for further evaluation.

3.5.1 Human observer performance on the FF++ dataset

The human observer study on the FF++ dataset (see Figure 3.5) included 204 participants and was conducted by the authors of the FF++ dataset [5]. The participants were shown individual images, each for a few seconds. The ratio of real-to-fake images shown is 50:50 [5].

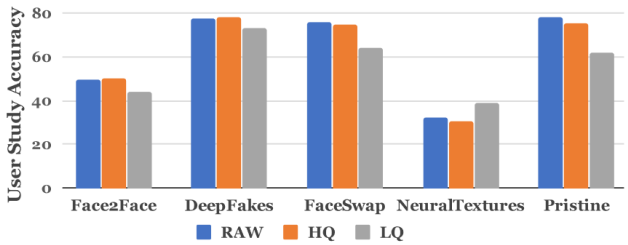


Figure 3.5: Human observer performance on the FF++ dataset. Taken and edited from [5]. RAW is the original video quality, usually around 1080p, HQ corresponds to 720p and LQ to 480p.

As we can see in the figure 3.5, human performace decreases with lower image resolution. Furthermore, NeuralTextures and Face2Face methods seem

to be more difficult for humans.

The average accuracy on all generative methods ranges from roughly 59% to 69% based on image resolution and from 35 % to almost 80 % based on the generative method.

3.5.2 Human observer performance on the DFDC dataset

The human performance study on the DFDC dataset (see Figure 3.6) was conducted independently by a third party in the paper [34]. Two studies were run, with over 15 thousand participants in total. The participants were split into those who were recruited, and those who participated organically.

In the first experiment, 5,524 participants were shown two videos, with one of them being real and one of them being fake. The task is then to determine the fake. For this purpose, 56 videos were selected [34].

In the second experiment with 9,492 participants, each person was shown a single video. The task was to score each video on a score from 0-100, where 0 corresponds to definitely real, 50 to unsure, and 100 to definitely deepfake. For this experiment, the resulting decision was also averaged across all participants to simulate aggregate voting [34].

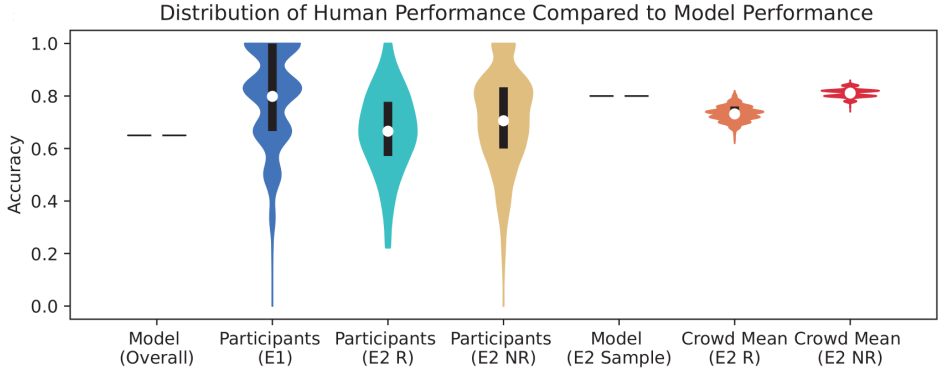


Figure 3.6: Human observer performance on the DFDC dataset. Each violin plot shows the distribution of participant performances across experiments. The white dot represents the mean value, while the black line is the standard deviation. Taken and edited from [34].

As we can see in the figure 3.6, the mean accuracy for the first forced choice experiment is roughly 75%. For the second experiment, the mean accuracy is roughly 70%.

This value corresponds with values found in the experiment on the FF++ dataset [5], although the values are not directly comparable, since the experiment setup differs.

Overall, most of the participants were capable of overperforming the best detection model at the time of writing the paper, be it by not a big margin [34].

Chapter 4

Proposed detection model

4.1 Few-shot learning

Few-shot learning (FSL) is a subtype of machine learning which tries to achieve the best model performance on the task given a minimal amount of labeled training samples [35]. This is in contrast to traditional machine learning setups, which try to maximize the amount of data.

FSL is especially helpful in areas when acquiring big quantities of data is not feasible, such as medical applications, multiple sources of deepfakes, and many others.

The goal is to emulate the human ability to quickly learn new tasks by using experience gained in some other task [35].

4.1.1 Empirical risk minimization

The goal of the model is to minimize the expected risk R, defined as

$$\begin{aligned} R(h) &= \int l(h(x), y) dp(x, y) \\ &= \mathbb{E}\left[l(h(x), y)\right] \\ &\approx \frac{1}{I} \sum_{i=1}^{I} l(h(x_i), y_i) = R_I(h), \end{aligned} \tag{4.1}$$

where h is the hypothesis measured w.r.t. $p(x, y)$. Since the distribution is unknown, the empirical risk R_I is minimized instead, usually as an average loss $l(\cdot, \cdot)$ over I training set [36].

The goal is then to find the optimal decision function, which minimizes the empirical risk on the training data, in hopes of approximating on the entire distribution. This process is defined as

$$\begin{aligned} \mathbb{E}[R(h_I) - R(\hat{h})] &= \mathbb{E}[R(h^*) - R(\hat{h})] + \mathbb{E}[R(h_I) - R(h^*)] \\ &= \mathcal{E}_{app}(\mathcal{H}) + \mathcal{E}_{est}(\mathcal{H}, I) \end{aligned} \tag{4.2}$$

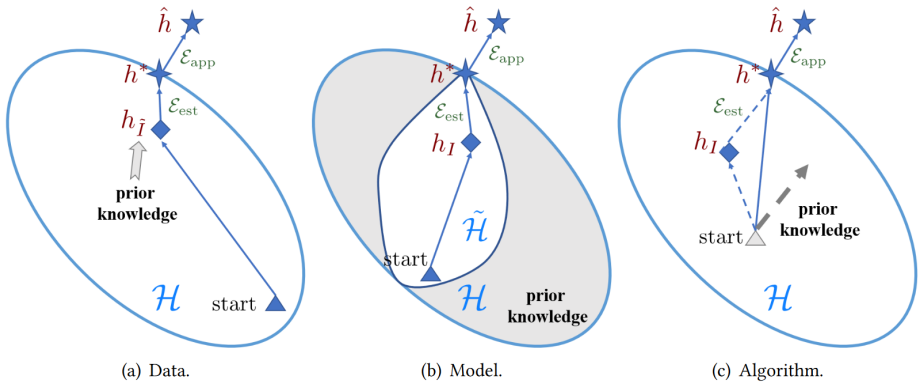
where $\hat{h}$ minimizes the expected risk without reliance on given hypothesis space $\mathcal{H}$, h^* minimizes the expected risk, while being bound by the hypothesis

space and h_I minimizes the empirical risk. The difference can be further rewritten as a sum of approximation error $\mathcal{E}_{app}$, which is bound by the hypothesis space and as such is a function of the chosen model and estimation error $\mathcal{E}_{est}$, which measures how close is the found hypothesis to the best one on the training data [36].

In general, the estimation error can be reduced by increasing the amount of training data, however, since in the FSL the training dataset is limited in size, other options must be considered. With a smaller training set, the likelihood of the model overfitting increases drastically, and as such, the estimation error cannot be minimized properly [36].

4.1.2 Approaches

Since the increase in the training set is not an option, other avenues are explored, as illustrated in Figure 4.1.



(a) Data. (b) Model. (c) Algorithm.

Figure 4.1: Few-shot learning principles. The picture (a) represents the fact that the increased amount and variety of data allow for better final results. Image (b) represents constrained search space and, as such, better chances of not getting stuck in a local minimum and overfitting to training data. Lastly, image (c) shows the direct search path from the initial to the final model. Taken and edited from [35].

4.1.3 Data

Using data from similar datasets

Given the assumption that data from a given domain are somewhat similar, we can augment our training set by introducing samples from another dataset, usually with different labels or data changed in some way.

Using weakly labeled or unlabeled data

The training dataset can be augmented by additional samples from datasets with only approximate labels or without labels altogether since labeling data

is expensive, especially when working with big web-scrapped datasets [37].

Data augmentation

Prior knowledge about the structure of the data can be introduced, resulting in an augmented dataset. The augmentations to the data are usually hand-crafted (although learned augmentations are also possible), such as affine transformations like rotation, sheering, or scaling. Photometric augmentations like additional noise, color change, etc. [38].

4.1.4 Model

In order to reduce the size of the hypothesis space $\mathcal{H}$, simpler models can be used. However, since real-world tasks usually cannot be described by a linear system, this can lead to an increase in the approximation error. Using appropriately big hypothesis space is therefore preferred [35].

Multitask Learning

During multitask learning, our desired task is learned in parallel with some other task, which usually has a larger amount of available data [39].

The parameters of the models are encouraged to be able to perform multiple different tasks, only differing in later stages of the model for each specific task [39].

Embedding Learning

In embedding learning, each sample is embedded from the original space $\mathcal{X}$ into a lower-dimensional space $\mathcal{Z}$ [40]. The smaller dimension allows for a smaller hypothesis space $\mathcal{H}$. The main components of embedding learning are the embedding functions f_{test} and f_{train} and a similarity function $s(\cdot, \cdot)$ [40].

The embedding models can be learned for a specific task, learned on a large-scale dataset for several tasks, or leverage information learned on the training dataset directly in the embedding function of the test sample [35].

4.1.5 Algorithm

The right choice of the search algorithm in the hypothesis space $\mathcal{H}$ is required for a quick and accurate resulting model.

Fine-tuning

In fine-tuning the task is given pre-trained parameters θ_0 is to find the optimal parameters on the new training set with only a limited amount of iterations, while preventing overfitting [41].

This is achieved with several different methods, be it updating only part of the parameters, using the same update for different parts of the network or early stopping [41].

Aggregation

Another option is instead of using a single parameter set to aggregate several different ones and train the final model on top of them [42].

Other methods include but are not limited to Model-Agnostic Meta-Learning [43] or learning the optimizer itself [44].

The model proposed for face deepfake detection utilized a pre-trained model that provides the embedding, followed by a simple linear classifier.

In addition, a small convolutional neural network is trained for the FF++ [5] and the medical dataset to ascertain the effectiveness of standard deep learning approaches.

4.2 Embedding models used in the thesis

In this thesis, we explore three different embedding models.

4.2.1 FaRL

In the General Facial Representation Learning in a Visual-Linguistic Manner paper [1], a FaRL model is introduced (**Fa**cial **R**epresentation **L**earning).

The architecture consists of an image encoder E_I, which is a visual Transformer ViT-B/16 [45], then a text encoder E_T, model as in [46]. Lastly, a masked image modeling module E_{MIM} is used during training, which is implemented as a single layer Transformer [1] (see Figure 4.2).

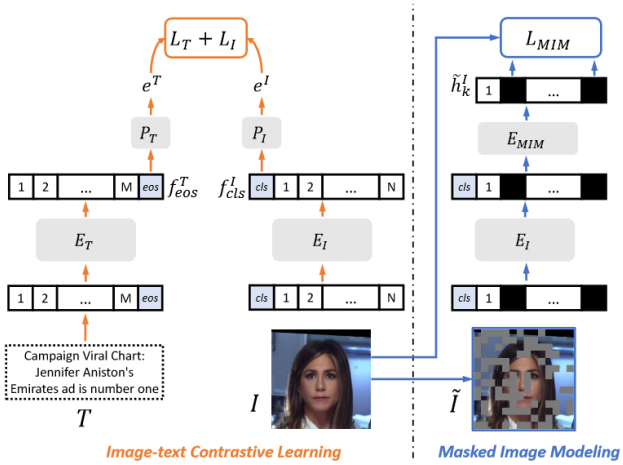


Figure 4.2: Framework of the FaRL pre-training. Taken and edited from [1].

The backbone of the network was trained on image-text description pairs from the LAION-Face dataset [47] and later tested on several specific tasks such as parsing, alignment, and face attribute recognition [1].

Several datasets were used during the training amongst others

- *LaPa* - 22 thousand faces annotated with landmarks and segmentation maps. Used for training face parsing [48] (see Figure 4.3).
- *CelebAMask-HQ* - [49] 30 thousand images with similar setup as LaPa. Used for face parsing as well as masked image modeling.
- *LFW* - Labeled Faces in the Wild dataset [50] contains 14 thousand image-identity-pairs. It is used for training face attributes [1].
- *LAION-Face* - Subset of the LAION [47] dataset, containing roughly 20 million image-text-pairs (see Figure 4.4). The goal of the transformer is to embed both the text and the image of the pair into roughly the same space [1]. Utilized during pre-training.

The setup is in itself a form of few-show learning, where authors use several thousands of images to fine-tune the network for a specific task, after pre-training the network on several million image-text-pairs.

Figure 4.3: Example data of the LaPa dataset. The second image shows landmarks. The third is the segmantation. Taken and edited from [48].

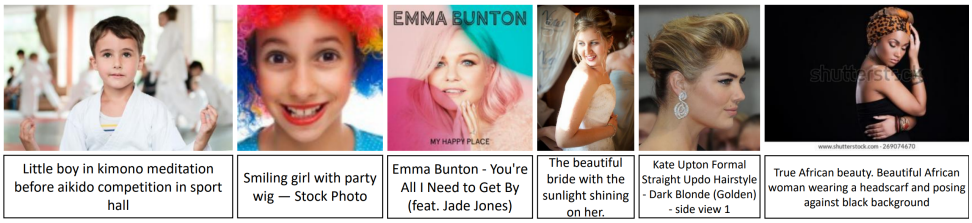


Figure 4.4: Example data of the LAION-Face dataset. Taken and edited from [1].

The output embedding has 512 dimensions.

4.2.2 ArcFace

ArcFace, introduced in the Additive Angular Margin Loss for Deep Face Recognition paper [2], is a modification of the standard cross-entropy loss function, which is defined as

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}, \tag{4.3}$$

where (x, y) is the features-class pair and (W, b) are the output weights, with their respective biases.

The logit $W_j^T x_i$ can be rewritten after fixing the l_2 norm of the weights vector as $||x_i|| \cos \theta_{y_i}$. The scale of the term $||x_i||$ can be further normalized to the value of s [2]. Cross-entropy loss can then be rewritten as

$$L_{CE_N} = -\frac{1}{N} \sum_{i=1}^{N} \frac{e^{s \cos \theta_{y_i}}}{e^{s \cos \theta_{y_i}} + \sum_{j=1, j \neq y_i}^{n} e^{s \cos \theta_j}}. \tag{4.4}$$

The embeddings generated by this process are projected onto a hyperspere with and radius of s.

The ArcFace loss introduces margin into the equation, forcing classes away from one another by at least some angular margin m [2]. This is defined as

$$L_{CE_N} = -\frac{1}{N} \sum_{i=1}^{N} \frac{e^{s \cos(\theta_{y_i} + m)}}{e^{s \cos(\theta_{y_i} + m)} + \sum_{j=1, j \neq y_i}^{n} e^{s \cos \theta_j}}. \tag{4.5}$$

The actual network used in the thesis is the ResNet-18 [3]. The network is trained on the LWF dataset [50].

The output embedding has 512 dimensions.

4.2.3 ResNet - ImageNet

Lastly, a simple ResNet-50 which was pre-trained on ImageNet [4] was tried, examples of which can be found in Figure 4.5. While ImagNet [4] is not a face dataset, the task is still general classification. It contains images organically found on the internet, totaling over 20 thousand classes, with over 14 million images. The network was trained with simple Cross-Entropy.

The size of the embedding is 2048.

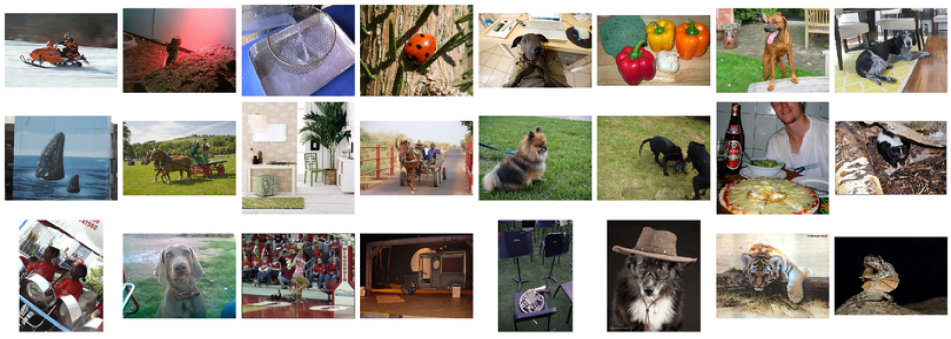


Figure 4.5: Examples from the ImageNet dataset. Taken and edited from [4].

4.3 Classifiers

A simple linear classifier is trained on the top of the embedding network. To keep the setup as easy as possible and require minimal additional human supervision, no deep learning methods are used. Instead, we only utilize quick-to-optimize methods, such as logistic regression [51] or SVMs [52].

4.3.1 Logistic regression

Logistic regression [52] presents the simple, most straightforward method used in the thesis. It is equivalent to learning a single-layer perceptron. The classifier minimizes binary cross-entropy given the annotated input pairs $(x_1, y_1), \ldots, (x_N, y_N)$ as

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i), \tag{4.6}$$

where y_i are the class labels of the given sample and $p_i = \sigma(\mathbf{w}^\top \mathbf{x}_i)$, (where $\sigma(x) = 1/(1 + \exp(-x))$) [51], are the predicted probabilities, given learned weight vector $\mathbf{w}$ and embedding model output $\mathbf{x}_i$.

4.3.2 SVM

SVMs [52], while remaining relatively simple in general, provide improved performance compared to logistic regression. SMVs are learned from a sequence of annotated embedding-label pairs $(x_1, y_1), \ldots, (x_N, y_N)$ same as logistic regression. However, we must optimize a hyperparameter C, which controls the regularization strength. Specific kernels, such as *rbf*, might require additional hyperparameter search. However, we only utilize linear kernels in this thesis.

The SVM algorithm translates learning of the linear classifier into a quadratic programming task which reads

$$\underset{\mathbf{w}, b, \zeta}{\text{minimize}} \quad ||\mathbf{w}||_2^2 + C\sum_{i=1}^{N} \zeta_i, \tag{4.7}$$

$$\text{subject to} \quad y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0 \quad \forall i \in \{1, \ldots, N\}.$$

Chapter 5

Existing benchmarks and metrics

In this chapter, we describe the two existing benchmarks and their evaluation metrics. The same metrics are used for the novel benchmarks.

5.1 FaceForensics++

The FaceForencis++ dataset [5] (FF++) was generated from a thousand source videos, which were collected on YouTube.

Deepfakes in the dataset were genereted by one of the five methods (see Figure 5.1), namely

- **FaceSwap** - A simple computer-graphics-based algorithm, which performes identity swapping, between two pairs of pristine videos [53].
- **Face2Face** - A computer-graphics-based algorithm, which does facial reenactment [19] (see section 2.2.1).
- **Deepfakes** - A deep-learning method that utilizes an auto-encoder architecture. Deepfakes are used for identity swapping. The implementation in the dataset is based on [54].
- **NeuralTextures** - A deep-learning method used for facial reenactment, this done as a patch-based GAN [20] (see section 2.2.2).
- **FaceShifter** - An simple computer graphics-based algorithm, based on the implementation of [55].

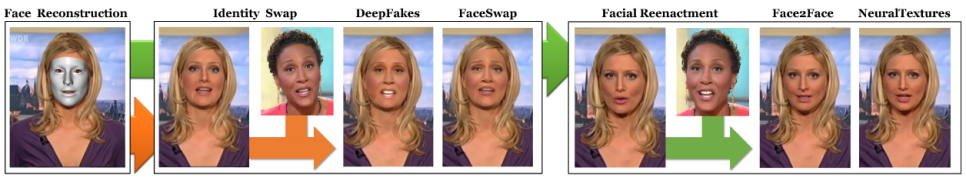


Figure 5.1: Generative methods in the FF++ dataset. Taken and edited from [5].

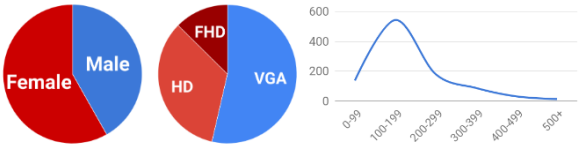
The source dataset contains mostly lower-resolution VGA videos (480p), however higher-resolution HD (720p) and FHD (1080p) videos are present as

well. These videos are compressed using several different compression rates into a RAW, HQ and LQ (see Figure 5.3).

Most of the faces in the dataset have a resolution of about 175×125. Gender-wise, the dataset is slightly skewed towards females.

In total, the dataset contains 1k pristine videos and 4k fake videos (1k for each generative model). The length of each video ranges anywhere from several seconds to a minute. Several examples can be seen in Figure 5.2.

Figure 5.2: Examples from the FF++ dataset [5]. Each row corresponds to a different generative method (1) - FaceSwap, (2) - Deepfakes, (3) - NeuralTextures.



(a) Gender (b) Resolution (c) Pixel Coverage of Faces

Figure 5.3: Distribution of the FF++ dataset, based on gender, resolution and pixel coverage of faces. Taken and edited from [5].

5.2 DFDC

The DeepFake Detection Challenge dataset [6], was created from a total of 48,190 videos, with an average length 68.8 seconds. The dataset contains 960 unique subjects, which all agreed to be in the dataset [6].

For the training set, a total of 119,154 ten-second videos were generated, with roughly 84% of them being deepfakes (see Figure 5.4). For this, several generative methods were used, namely

- **DFAE** - DFAE is an autoencoder, which uses a shared encoder and two decoders trained separately [6].
- **MM/NN FaceSwap** - This computer-graphics-based method transfers the closest frame based on the face landmark with some additional blending [56].
- **NTH** - Neural Talking Heads uses extracted landmarks from the target face to guide the movement of the source face. The model's architecture is similar to the StyleGAN [57, 21].
- **FSGAN** - FSGAN combines information from extracted landmarks and segmentation of both the face and the hair to inpaint and blend the source face onto the target video. In addition the model accounts for cross-frame consistency [58].

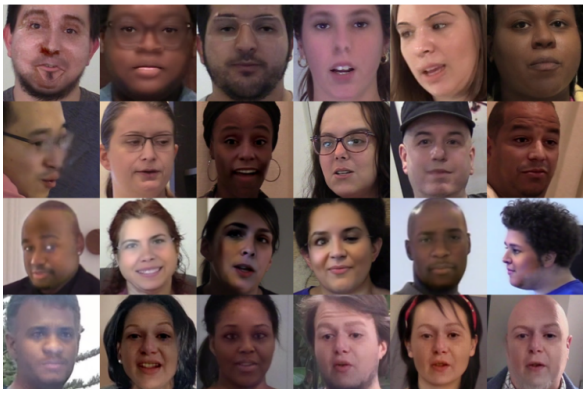


Figure 5.4: Examples from the DFDC dataset. Taken and edited from [6].

5.3 Dataset comparison

Several key differences between the FF++ [5] and DFDC [6] are the

- **Dataset size** - DFDC is several times bigger in size
- **Consent of subjects** - All subjects in the DFDC dataset have agreed.
- **Overall scene** - Scene in the FF++ dataset is usually from some televised video (news, talk show, etc.), while scenes in DFDC are generally a conversation between a few subjects. While this might be more natural, the most damaging use of deepfakes is usually those that the most people will see.
- **Generative methods** - Both datasets contain the same number of generative models. While the number 8 is stated in the DFDC paper, the real figure is lower since some models are duplicates with different setups.
- **Labeling** - Data in the FF++ is split based on the generative method used. In DFDC, all files are placed together. This setup does not allow for per-method results, which is a huge disadvantage.

5.4 Evaluation metrics

Datasets contain videos, which are labeled either as real or fake. The labels, together with model predictions, are then used for evaluation. The evaluation metrics used in the benchmarks differ widely. From simple accuracy to more advanced AUC metrics.

Basic metrics

Given the model prediction Y and ground truth GT, we compute

- TP - Number of true positive examples as $\mathrm{TP} = \sum_{i=1}^{N} [\![Y = 1 \wedge GT = 1]\!]$.
- FP - Number of false positive examples as $\mathrm{FP} = \sum_{i=1}^{N} [\![Y = 1 \wedge GT = 0]\!]$.
- FN - Number of false negative examples as $\mathrm{FN} = \sum_{i=1}^{N} [\![Y = 0 \wedge GT = 1]\!]$.
- TN - Number of true negative examples as $\mathrm{TN} = \sum_{i=1}^{N} [\![Y = 0 \wedge GT = 0]\!]$.

Accuracy

The accuracy is the estimate of the probability of the incorrect prediction, defined as

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{F} + \mathrm{N}}$$

Precision

The precision represents the proportion of correctly predicted instances from all positive predictions, defined as

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}},$$

Recall (TPR)

Recall, sometimes called true positive rate, represents the proportion of the positive examples which were correctly predicted from all positive examples, defined as

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}.$$

Fallout (FPR)

Fallout, also called false positive rate, is defined as

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}.$$

FPR@TPR(k)

This metric represents the value of a false positive rate at a fixed true positive rate. The idea is to fix the maximal accepted miss rate.

In the thesis, the value of k = 90 is used, meaning that at least 90% of deepfakes must be labeled as such.

F_β-score

F-score is another measure of predictive performance. It utilizes information from both the negative and the positive class while allowing one to weigh each type of error differently by setting the β parameter to different values. It is defined as

$$F_\beta = (1 + \beta^2) \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{(\beta^2 \cdot \mathrm{precision}) + \mathrm{recall}}. \tag{5.1}$$

For example, by setting $\beta = 1$, we get the equation for the harmonic mean of precision and recall.

Receiver operating characteristics (ROC) and area under the curve (AUC)

The ROC curve plots the true positive rate and the false positive rate for different values of the decision boundary.

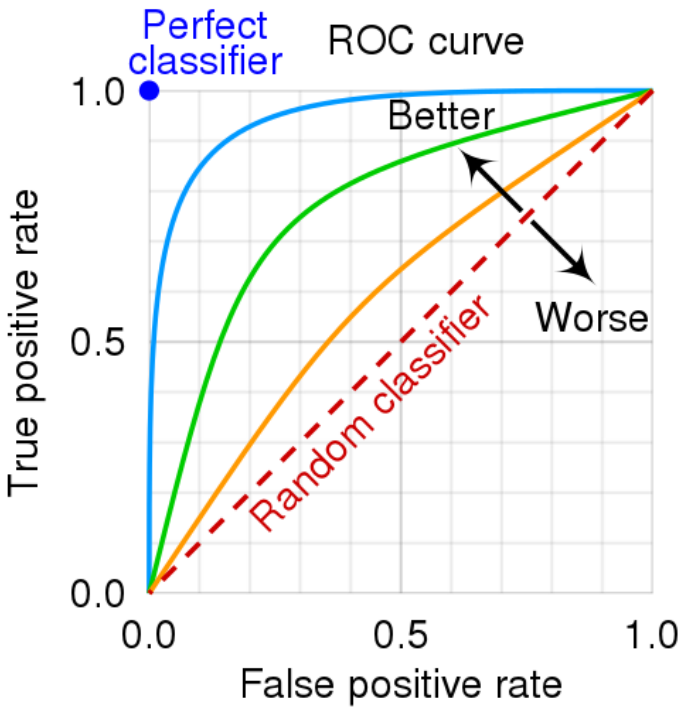


Figure 5.5: Example of the ROC-curve. Taken and edited from [59].

The values of AUC are in the $\langle 0, 1 \rangle$ range, where higher values mean better classifier performance (0.5 means random guessing). Example of the ROC-curve is shown in the Figure 5.5.

Chapter 6

Novel benchmarks

For the purpose of the thesis, we created two novel benchmarks for evaluating the deepfake detectors. The first contains organically found images of Instagram influencers. Compared to the existing benchmarks, the images in the influencer benchmark are generated by current SOTA methods, represent a real-world task, and have significantly higher resolution.

The second benchmark was generated from medical MRI scans. Currently, only one medical deepfake benchmark exists [60]. The medical benchmark used in this thesis was created to establish the effectiveness of our approach in other modalities.

6.1 Instagram influencer dataset (FIID)

The Fake Instagram Influencer Dataset is a novel benchmark generated from photos publicly available on the Instagram pages of several influencers.

6.1.1 Influencer selection

The dataset comprises 24 influencers, 12 real and 12 fake.

For the purpose of the dataset, a "fake influencer" is any account for which the identity was generated by deep learning methods. The main issue during collection was the uncertainty that the given account was fake. Most accounts state that their content is generated; however, many don't. Such accounts were placed into the dataset only when their status could be determined by other means, such as repeated collaboration with other fake accounts or other additional information on online forums.

As we can see in the trends curve for the term "deepfake" (see Figure 2.10), most people's contact with deepfake technology is likely to be through sexually explicit content.

As such, the dataset tries to reflect this by the demography selection. All fake subjects in the dataset are young females.

The real influencers were then selected to mirror the fake ones w.r.t. age, ethnicity, and general content. To ensure that the content in these accounts comes from a real source, publicly known figures were selected.

6.1.2 Image selection

Around a hundred images (85-100) were downloaded for each influencer. Furthermore, only images where the subject's face is visible and is the image's main focus were considered. This filtered out around half of all the posts present for the real influencers. However, for the fake influencers, the face seems to be the main focus in almost all photos.

6.1.3 Dataset summary

The summary of the datatest can be found in Table 6.1, overall resolution distributions in Figure 6.1 and finally several examples can be seen in Figures 6.2 and 6.3.

Metric	**Real**	**Fake**
Number of influencers	12	12
Total number of images	1148	1125
Average age	31.6	×
Median image diagonal	1728.8	1728.1

Table 6.1: Summary of the FIID.

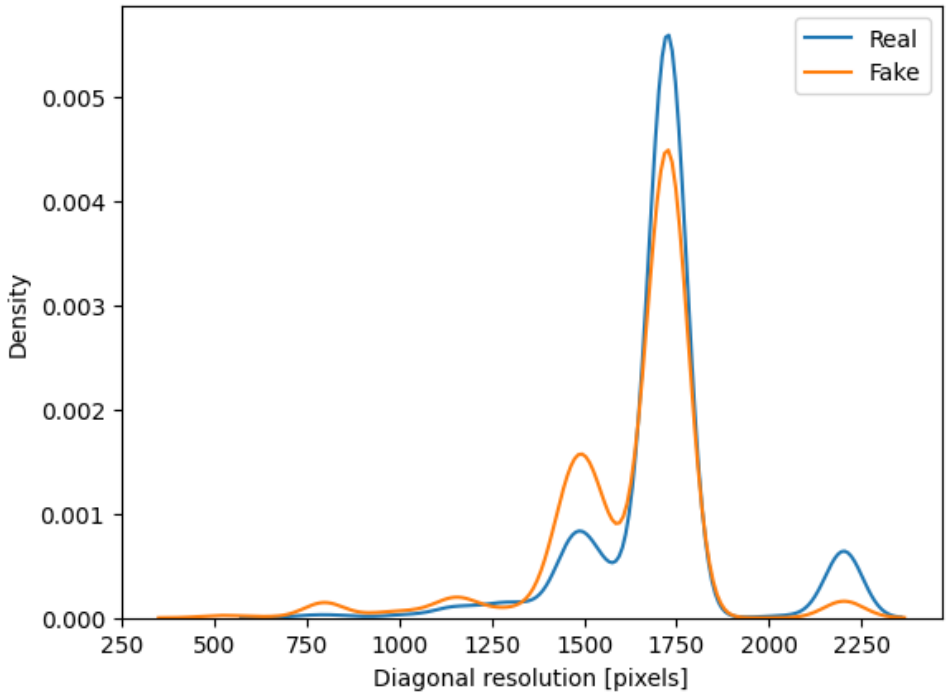


Figure 6.1: Diagonal resolution of the FIID.

Figure 6.2: Examples of fake influencers in the FIID.

Figure 6.3: Examples of real influencers in the FIID.

6.2 Medical Fake Image Dataset

The Medical Fake Image Dataset (MFID) contains 2D images, all with tumors. Half of the images contain a real tumor, the second half the fake ones.

The source data for this novel dataset was an MRI brain scan dataset called BRATS2017 [61] (see Figure 6.4). A subset of the dataset was used, which contained **484** 3D volumes of the brain. The dataset provides a per-voxel segmentation map. This information was used to select specific slices containing tumors.

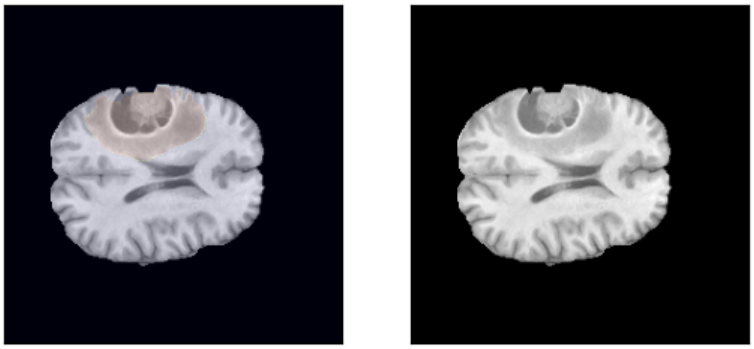


Figure 6.4: Example sample from the BRATS2017 dataset [61] with the segmentation mask.

6.2.1 Slice selection

Each 3D volume has a resolution of $240 \times 240 \times 155$. Slice selection was done on the transversal plane. Since each volume is guaranteed to contain a tumor, a slice with a maximum ratio is found. Several surrounding slices are also considered for the extraction process to generate more data.

Since the pipeline uses the same brain as a source and target of the tumor, additional constraints are placed on a tumor position, namely the minimal distance from the horizontal center line and minimal tumor size.

6.2.2 Fake tumor generation

The fake generation process utilizes Poisson blending [14] (implemented by the *cv2* library [62]), given the mask provided by the dataset. The tumor is segmented out of the slice and then placed on the opposite side of the brain via horizontal flipping. Given distance constraint, the generated tumor is guaranteed not to have an overlap with the source tumor.

The source dataset contains **484** source 3D volumes. From these, we generate **2387** slices, examples of which can be found in Figure 6.5.

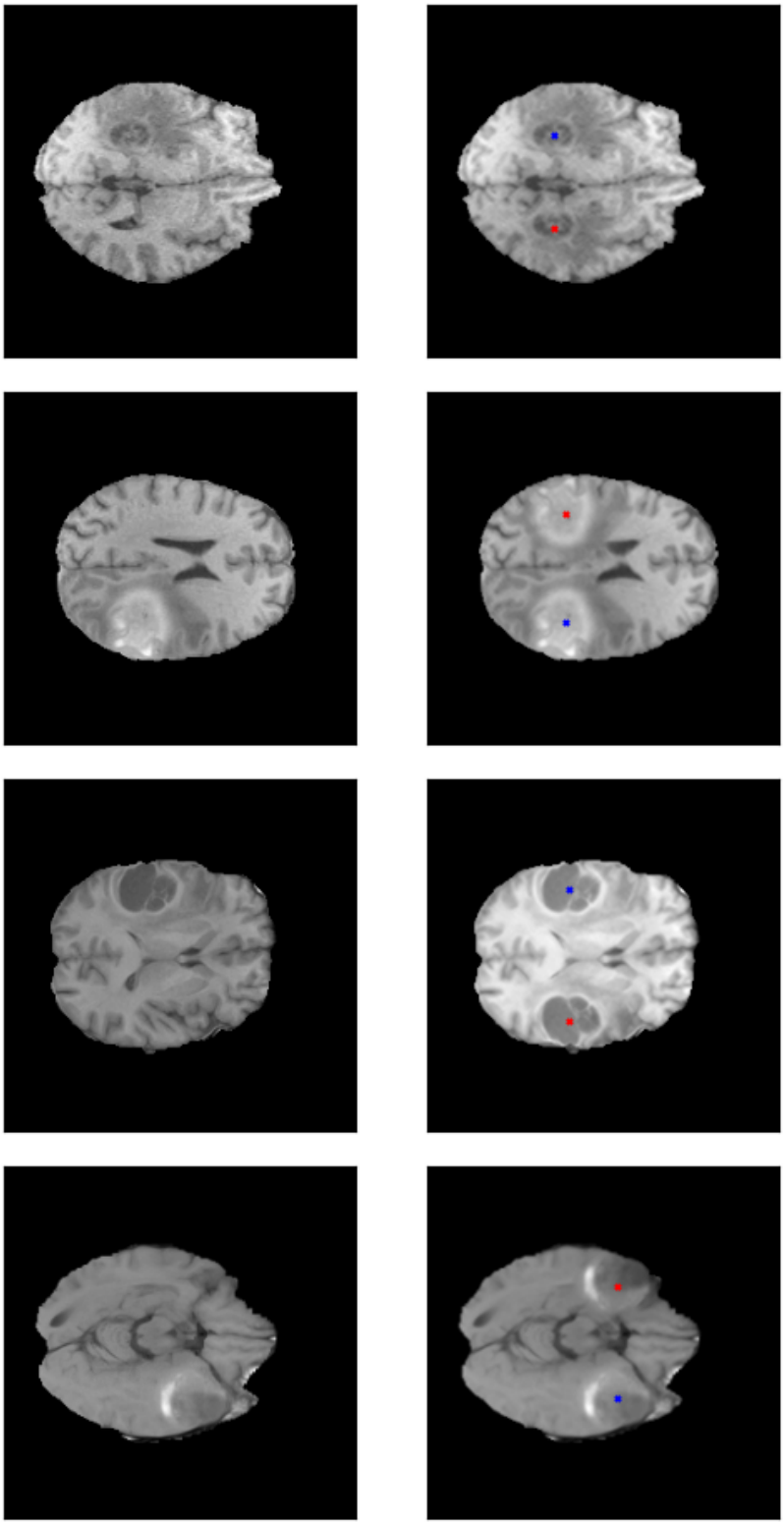


Figure 6.5: Examples from the medical dataset. The first image is the original slice. The second image contains the fake tumor. The blue cross is the center of the original tumor, and the red cross is the center of the fake one.

Chapter 7

Experiments and results

In this chapter, we describe the experiments run in this thesis. The first several were run on the existing benchmarks, aiming to establish the best embedding model and classifier. Secondly, we test the effectiveness of the best model on the novel FIID dataset. Lastly, we explore the ability of the proposed model to generalize on an entirely different modality.

7.1 Face extraction

Since our proposed method works over individual faces rather than the entire video, we need a way to extract and align them since some embedding models, like FaRL [1], are trained on aligned faces. Therefore, additional face transformation is used to increase model performance.

In order to get the extracted align faces, we use RetinaFace [63]. RetinaFace was also used as a face extraction model of choice for most other models (FaRL).

Extracted faces are then resized into a size of 224×224 and saved. No intermediate resizing is utilized for the Fake Instagram Influencer dataset. Since the number of images is relatively low, the detected faces are passed directly to the embedding model with their respective preprocessing.

The tumors in the medical dataset are extracted as squares centered at the center of mass.

7.2 Dataset usage

Both FF++ [5] and DFDC [6] datasets are not used in their entirety in this thesis. Instead of using predefined train/test splits in each dataset, the data is k-fold cross-validated, where k ranged from 20 when dealing with smaller train samples down to 5 when dealing with near full-size subsets.

7.2.1 Subset of FaceForensics++

For the FF++, roughly half of the fake videos are processed, with every 8th frame extracted. Since the dataset is split into four distinct generative models, the same distribution is preserved in our subset.

The amount of real videos in the original dataset is only one-fifth. Therefore, all real videos are processed by extracting each 6th frame. The smoother sampling increases the number of samples, which allows us to maintain a better-balanced distribution of real and fake videos.

7.2.2 Subset of DFDC

Data in the DFDC is not split based on the generative method used. Therefore, every video is treated as equal. However, since only roughly a quarter of the DFDC videos are used in the subset, the main issue during video selection was a repetition of the same identity. To overcome this issue, manual selection of identities was done to ensure proper identity diversity.

The per-video acquisition is the same as for the FF++ dataset, meaning each 8th frame and every 6th frame are sampled for the fake and real videos, respectively.

The usage of the DFDC dataset in the thesis is more secondary. Since the generative methods are not stated on a per-video basis and overall quality is lower than that present in the FF++ dataset, the results are included for completeness and comparison with human observers.

7.3 Evaluation protocol

To ascertain the proposed model's viability, k-fold cross-validation is used. When samples from multiple generative models are used, the fake images are split so that their representation is the same. Furthermore, faces are grouped by their source video to remove faces that are already too similar to those in the training set.

7.3.1 Training setup

The training dataset has variable size, ranging from only a few samples to several thousand. A training sample is considered as a single face. Faces are chosen randomly from the given fold.

7.3.2 Testing setup

The trained model is tested on faces from videos not in the training set. Additional aggregation is done on all faces from the same video, simply as the mean predicted score

$$S_{vid} = \frac{1}{N} \sum_{i=1}^{N} S_{fr}^{(i)}, \tag{7.1}$$

where S_{fr} is the classifier prediction of the given frame in the $\langle 0, 1 \rangle$ range.

7.4 FF++ results

We ran three sets of experiments on the FF++ [5]

1. **Single generative model** - The same generative model is used in both the training and testing datasets.
2. **Leave-one-out (LOO)** - All but one generative model is used in the training dataset. The remaining one is used for testing.
3. **All generative models** - All generative models are used in both the training and testing datasets.

7.4.1 Initial embeddings

Before applying any form of further classification, we visualize the initial embeddings of the models. This is done so that we can ascertain any structure of the data and/or find additional dependencies.

FaRL embeddings of the FF++

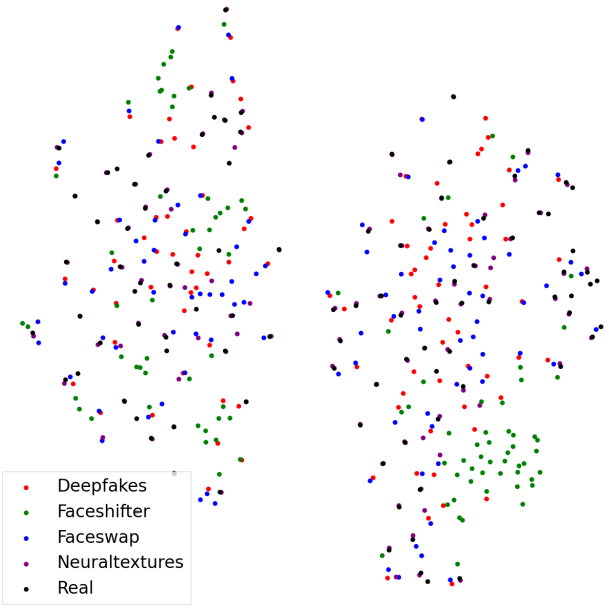


Figure 7.1: Initial embedding of the FF++ dataset using the FaRL embedder. The visualization from the initial dimension to 2D was done using T-SNE [64] on frames from 100 randomly selected videos.

As we can see in Figure 7.1, some form of clustering is already present in the data, even without further classification, especially between samples between individual generative methods.

We observe a noticeable split into two groups. However, they do not correspond to real and fake faces. Further examination of source videos shows that the split is caused by gender.

ArcFace embeddings of the FF++

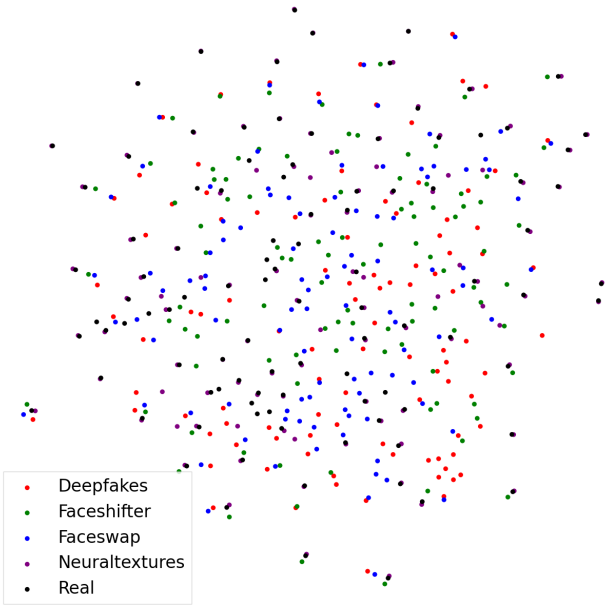


Figure 7.2: Initial embedding of the FF++ dataset using the ArcFace embedder. The visualization from the initial dimension to 2D was done using T-SNE [64] on frames from 100 randomly selected videos.

From the figure 7.2, we can see that essentially no clustering between generative models or real/fake videos takes place. However, a small amount of clustering occurs between videos that share the same identity. These are visible as groups of five dots placed together. This is likely caused by ArcFace's initial purpose, identity identification. As such, the face similarity is more important than any artifacts present. It is important to note that *perplexity* parameter of the T-SNE plays an important role in determining the size of the clusters. To reduce the chance that the clusters are caused by the *perplexity* parameter, its value is selected as double the number of

possible generative methods (4 fake methods and 1 real method, with the perplexity value of 10).

ResNet-50 (ImageNet) embeddings of the FF++

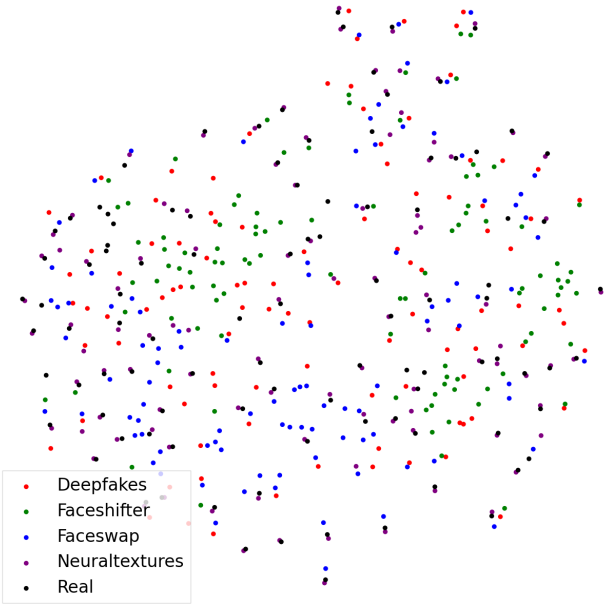


Figure 7.3: Initial embedding of the FF++ dataset using the ImageNet embedder. The visualization from the initial dimension to 2D was done using T-SNE [64] on frames from 100 randomly selected videos.

The embeddings generated by this model seem to share parts of the characteristics of both the FaRL [1] and the ArcFace [2] methods (see Figure 7.3). There is some rudimentary clustering between generative methods, and several identities are clustered.

Even though the ImageNet [4] dataset does not contain any labels that contain either people or their faces, the general classification task seems to create embedding, which is at least somewhat relevant to the face deepfake detection problem.

Overall, neither of the embedding models can perform the task of deepfake detection on their own. An additional classifier on top is required.

7.4.2 Experiment 1: Single generative model

In the setup, where the training and the testing datasets use the same generative model, we investigate the number of training examples required to achieve the desired classification performance. This experiment aims to determine which combination of embedding and classifier leads to the best deepfake detection performance.

The real/fake distribution in training and test is set to be the same. The resulting metrics shown are the AUC and FPR@TPR(90) (see section 5.4).

FaRL and logistic regression

The combination of FaRL followed by logistic regression achieves near-perfect performance on the FaceSwap and FaceShifter generative methods, as we can see in Table 7.1 and 7.2. Furthermore, on simpler method like FaceShifter, the FSDF model achieved AUC of **0.96** at 10 training examples.

Generative method	Train size = 10	Train size = 100	Train size = full
Deepfakes	0.78±0.08	0.94±0.02	0.97±0.02
NeuralTextures	0.64±0.06	0.83±0.05	0.93±0.03
FaceSwap	0.78±0.05	0.94±0.02	0.99±0.01
FaceShifter	0.96±0.03	1.00±0.00	1.00±0.00

Table 7.1: AUC of the FaRL + logistic regression on the FF++ dataset given several training sizes.

Generative method	Train size = 10	Train size = 100	Train size = full
Deepfakes	0.54±0.14	0.19±0.08	0.09±0.05
NeuralTextures	0.78±0.05	0.49±0.12	0.21±0.08
FaceSwap	0.56±0.09	0.16±0.05	0.03±0.02
FaceShifter	0.13±0.11	0.00±0.00	0.00±0.00

Table 7.2: FPR@TPR(90) of the FaRL + logistic regression on the FF++ dataset given several training sizes.

ArcFace and logistic regression

The combination of ArcFace embedding with logistic regression achieves significantly worse results than the same setup with FaRL embedding (see Table 7.3 and 7.4). Even the performance on the simplest generative method only reaches AUC of **0.85** on a full dataset.

Generative method	Train size = 10	Train size = 100	Train size = full
Deepfakes	0.65±0.09	0.78±0.04	0.87±0.03
NeuralTextures	0.58±0.06	0.65±0.04	0.75±0.04
FaceSwap	0.58±0.04	0.68±0.04	0.81±0.03
FaceShifter	0.67±0.06	0.75±0.06	0.85±0.05

Table 7.3: AUC of the ArcFace + logistic regression on the FF++ dataset given several training sizes.

Generative method	Train size = 10	Train size = 100	Train size = full
Deepfakes	0.74±0.13	0.60±0.13	0.38±0.08
NeuralTextures	0.83±0.07	0.80±0.06	0.66±0.09
FaceSwap	0.84±0.04	0.72±0.07	0.53±0.09
FaceShifter	0.72±0.09	0.60±0.14	0.40±0.12

Table 7.4: FPR@TPR(90) of the ArcFace + logistic regression on the FF++ dataset given several training sizes.

ResNet-50 ImageNet and logistic regression

The combination of the ResNet-50 pre-trained on ImageNet followed by logistic regression achieves a sufficient AUC performance of **0.83** - **0.92** for 100 train examples (see Table 7.5 and 7.6).

Generative method	Train size = 10	Train size = 100	Train size = full
Deepfakes	0.74±0.07	0.92±0.03	0.96±0.01
NeuralTextures	0.66±0.04	0.83±0.05	0.90±0.03
FaceSwap	0.72±0.06	0.91±0.03	0.96±0.01
FaceShifter	0.74±0.06	0.92±0.03	0.97±0.01

Table 7.5: AUC of the ImageNet + logistic regression on the FF++ dataset given several training sizes.

Generative method	Train size = 10	Train size = 100	Train size = full
Deepfakes	0.62±0.17	0.23±0.07	0.11±0.04
NeuralTextures	0.76±0.07	0.50±0.11	0.32±0.11
FaceSwap	0.67±0.09	0.27±0.09	0.10±0.04
FaceShifter	0.61±0.10	0.25±0.09	0.06±0.03

Table 7.6: FPR@TPR(90) of the ImageNet + logistic regression on the FF++ dataset given several training sizes.

FaRL and Linear-SVM

Since FaRL embedding achieved the best performance out of the three tried, we additionally test it with a Linear-SVM. Multiple values of c were tried out, of which $c = 1$ achieved the best performance. The resulting detection model achieves slightly worse results than when using logistic regression (see Table 7.7).

Generative method	Train size = 10	Train size = 100	Train size = full
Deepfakes	0.68±0.21	0.93±0.02	0.95±0.00
NeuralTextures	0.54±0.20	0.84±0.04	0.90±0.01
FaceSwap	0.79±0.05	0.93±0.02	0.97±0.01
FaceShifter	0.97±0.02	1.00±0.00	1.00±0.00

Table 7.7: AUC of the FaRL + Linear-SVM (c = 1) on the FF++ dataset given several training sizes.

Small convolutional network

To establish the effectiveness of standard deep learning models, we implement a small network based on the ResNet architecture [3]. The network tends to overfit the training data while not being able to generalize well on the testing data when presented with the same amount of train samples as our FSDF model (see Table 7.8).

When the same convolutional model is presented with the full train dataset, it achieves near perfect results on all generative methods, however time-requirements for such training increase significantly.

Generative method	Train size = 10	Train size = 100	Train size = 1000	Train size = full
Deepfakes	0.54±0.15	0.59±0.14	0.73±0.11	1.00±0.00
NeuralTextures	0.47±0.26	0.49±0.21	0.53±0.13	0.98±0.02
FaceSwap	0.46±0.22	0.57±0.16	0.59±0.14	0.99±0.02
FaceShifter	0.49±0.19	0.63±0.12	0.74±0.11	1.00±0.00

Table 7.8: AUC of the small CNN on the FF++ dataset given several training sizes.

Summary

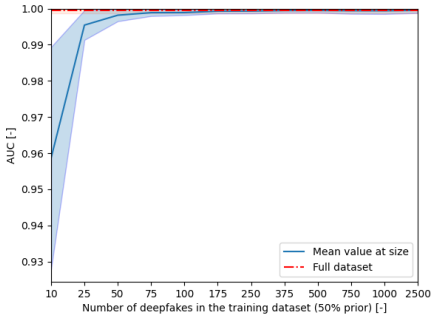
The experiment shows that the FaRL embedding performs better in all measured metrics than both ArcFace and the ResNet-50 ImageNet embeddings. The ArcFace is even outperformed by the ImageNet.

The performance of all detection models is consistently worst on the NeuralTextures and FaceSwap, followed by Deepfakes, and finally achieving the best performance on the FaceShifter method.

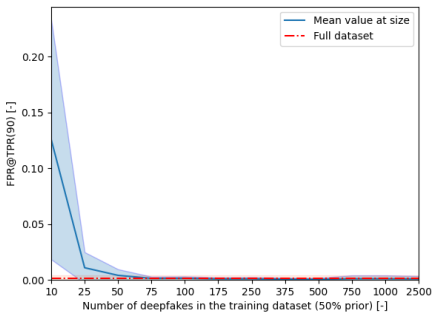
Instead of using logistic regression, Linear-SVM with several values of the hyperparameter c was tried (out of which $c = 1$ performed the best). SVM performed worse on a small sample size while not performing better for a bigger sample size.

The following Figures 7.4, 7.5, 7.6 and 7.7 show the detailed detection model performance given the number of training samples and the performance on the full dataset. Each row shows the results of a pair of AUC and FPT@TPR(90) for the given generative model (left column for the AUC, right column for the FPR@TPR(90)).

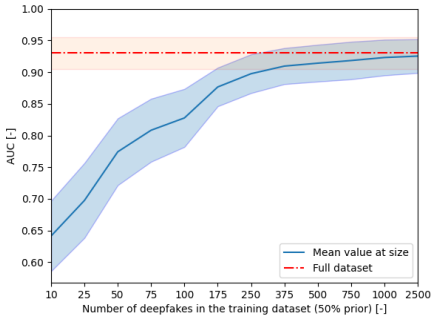
The Figure 7.8 shows the comparison between different versions of the FSDF model.



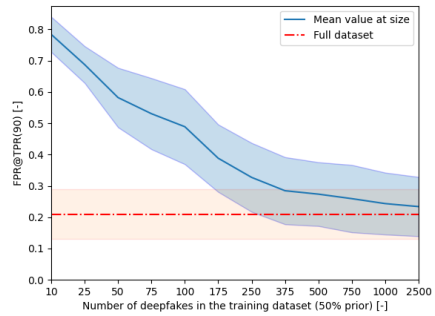
(a) : FaceShifter: AUC



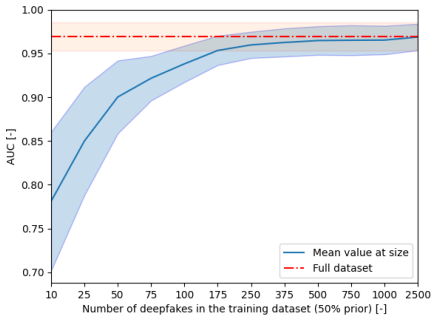
(b) : FaceShifter: FPR@TPR(90)



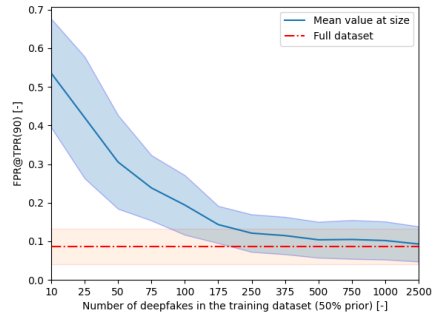
(c) : NeuralTextures: AUC



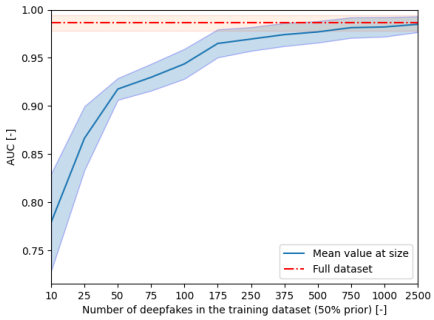
(d) : NeuralTextures: FPR@TPR(90)



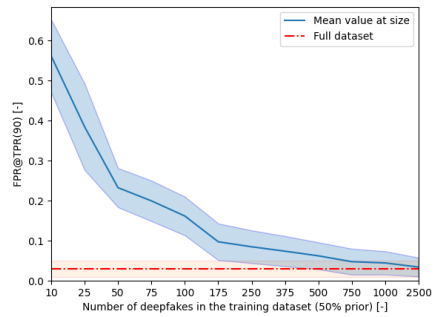
(e) : Deepfakes: AUC



(f) : Deepfakes: FPR@TPR(90)

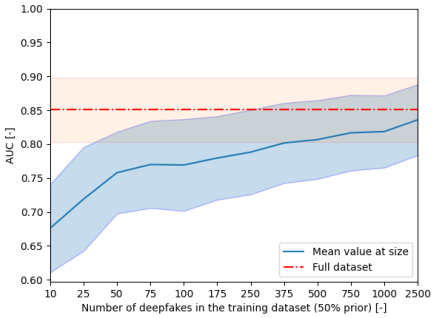


(g) : FaceSwap: AUC

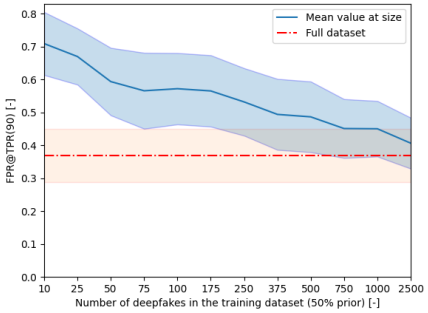


(h) : FaceSwap: FPR@TPR(90)

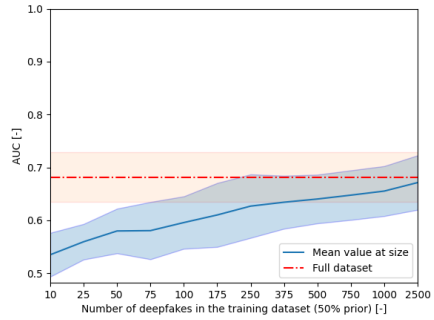
Figure 7.4: Results of FaRL methods on the FF++ dataset with variable amount of training samples and confidence bounds.



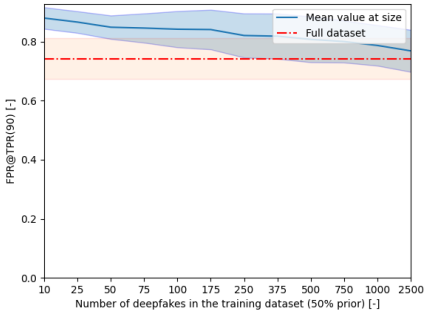
(a) : FaceShifter: AUC



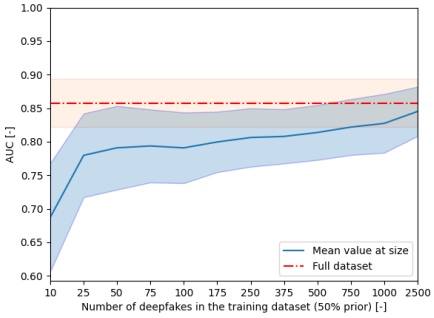
(b) : FaceShifter: FPR@TPR(90)



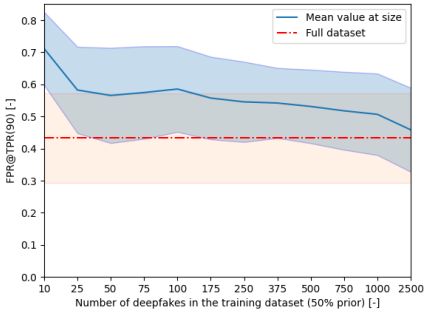
(c) : NeuralTextures: AUC



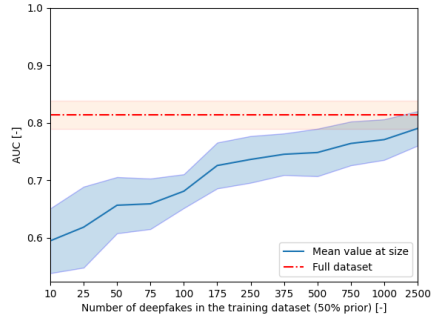
(d) : NeuralTextures: FPR@TPR(90)



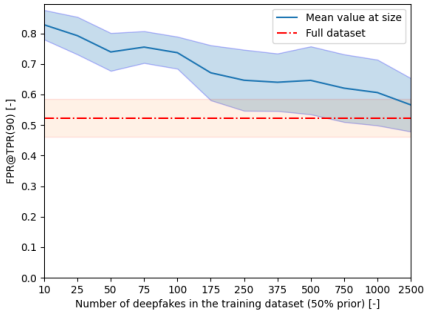
(e) : Deepfakes: AUC



(f) : Deepfakes: FPR@TPR(90)

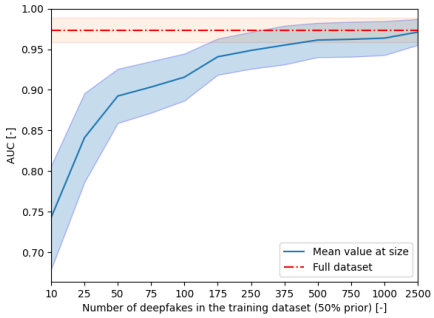


(g) : FaceSwap: AUC

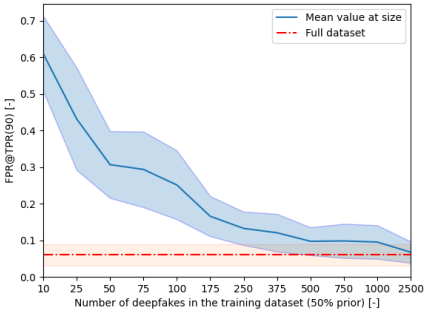


(h) : FaceSwap: FPR@TPR(90)

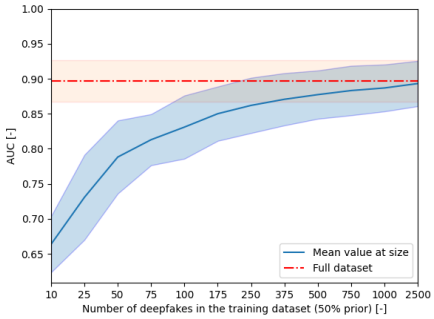
Figure 7.5: Results of ArcFace method on the FF++ dataset with variable amount of training samples and confidence bounds.



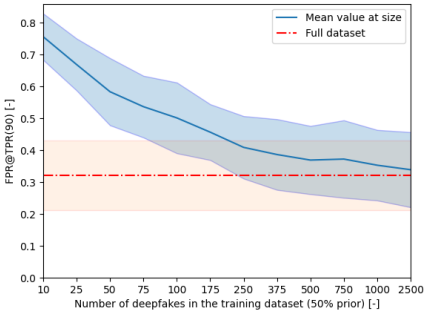
(a) : FaceShifter: AUC



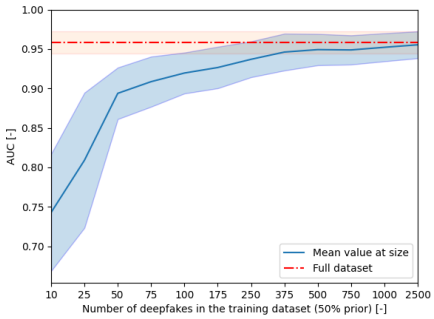
(b) : FaceShifter: FPR@TPR(90)



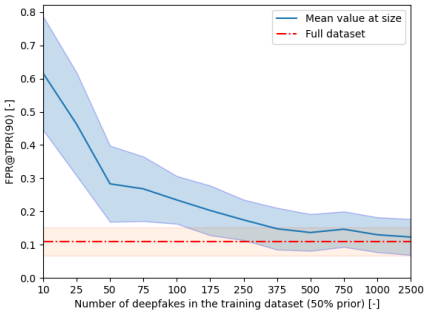
(c) : NeuralTextures: AUC



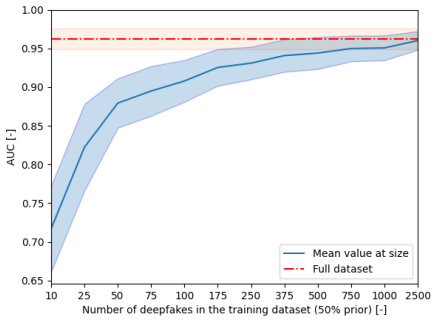
(d) : NeuralTextures: FPR@TPR(90)



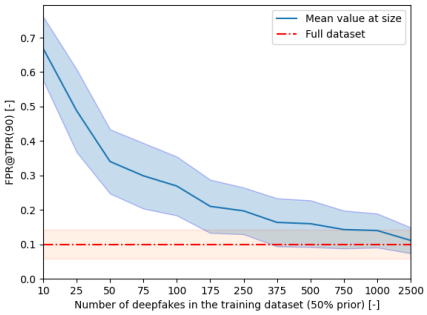
(e) : Deepfakes: AUC



(f) : Deepfakes: FPR@TPR(90)

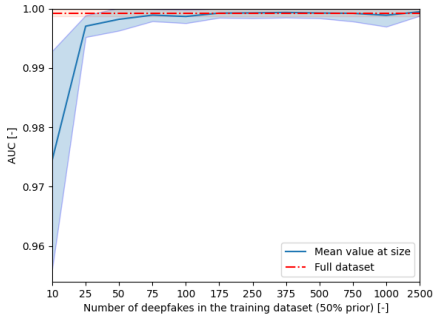


(g) : FaceSwap: AUC

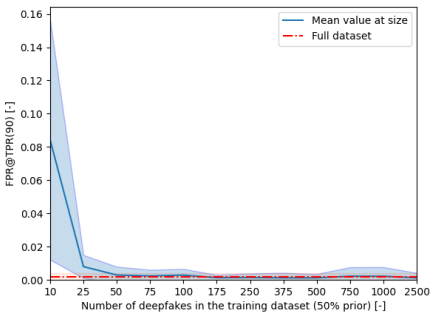


(h) : FaceSwap: FPR@TPR(90)

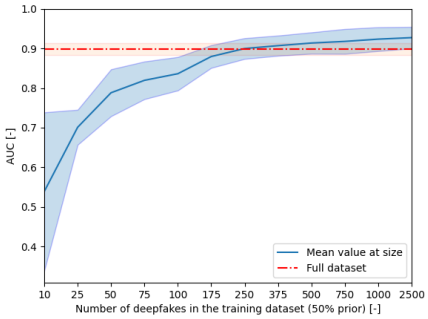
Figure 7.6: Results of ResNet-50 ImageNet methods on the FF++ dataset with variable amount of training samples and confidence bounds.



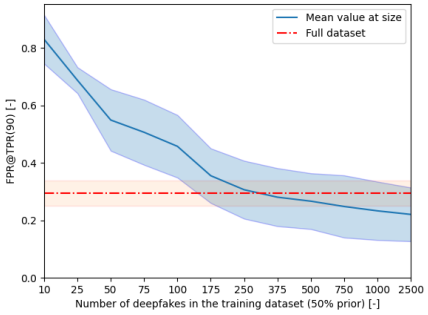
(a) : FaceShifter: AUC



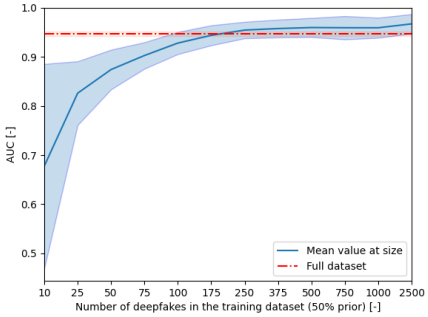
(b) : FaceShifter: FPR@TPR(90)



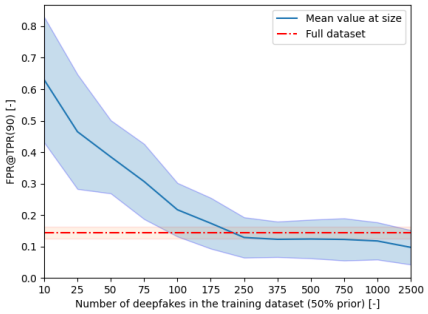
(c) : NeuralTextures: AUC



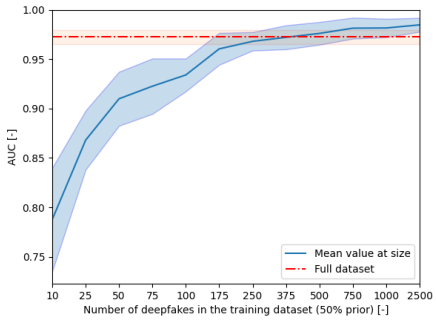
(d) : NeuralTextures: FPR@TPR(90)



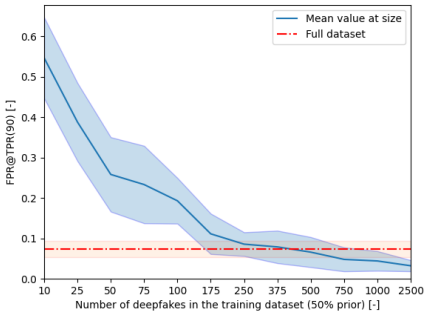
(e) : Deepfakes: AUC



(f) : Deepfakes: FPR@TPR(90)

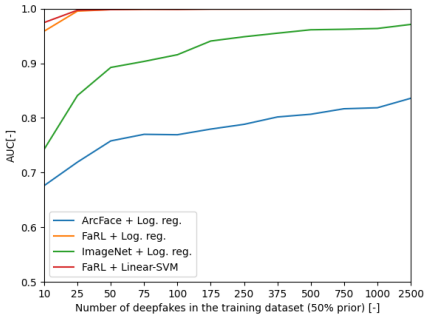


(g) : FaceSwap: AUC

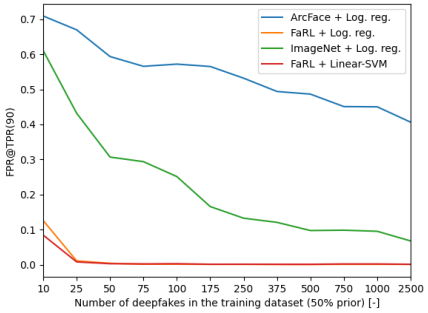


(h) : FaceSwap: FPR@TPR(90)

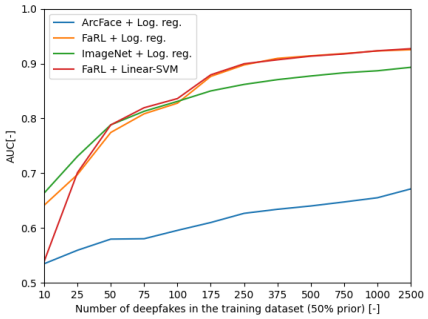
Figure 7.7: Results of FaRL + Linear-SVM ($c = 1$) method on the FF++ dataset with variable amount of training samples and confidence bounds.



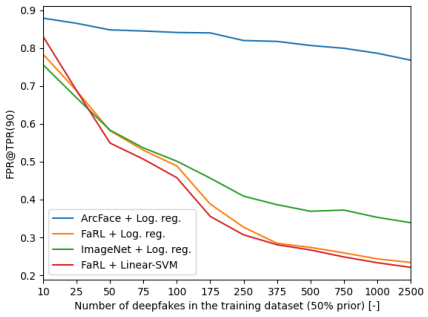
(a) : FaceShifter: AUC



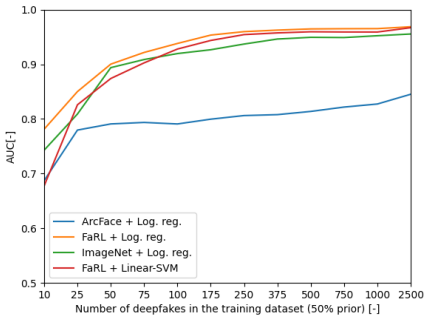
(b) : FaceShifter: FPR@TPR(90)



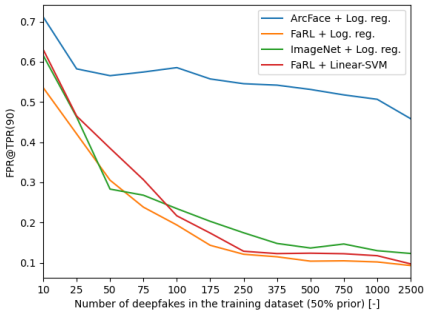
(c) : NeuralTextures: AUC



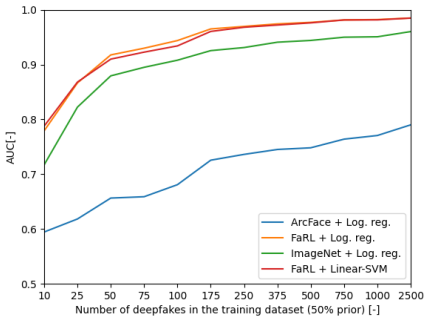
(d) : NeuralTextures: FPR@TPR(90)



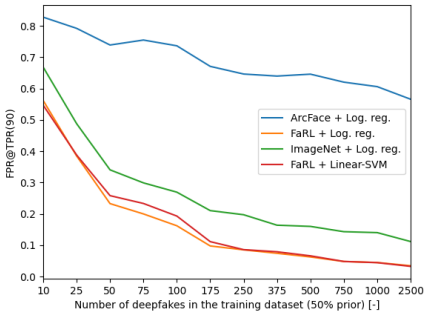
(e) : Deepfakes: AUC



(f) : Deepfakes: FPR@TPR(90)



(g) : FaceSwap: AUC



(h) : FaceSwap: FPR@TPR(90)

Figure 7.8: Comparison of tried methods on the FF++ dataset depending on the number of the training samples.

7.4.3 Leave-one-out

During the Leave-one-out (LOO), all but one generative method is used in the training. Only videos from the remaining one are used for testing.

The goal of this experiment is to explore the models' performance when tested images are from generators, which are not present in the training set.

Since this experiment simulates the situation when we have access to several generative models but have no information on a new one, we use full samples from the training sets.

	FaRL	ArcFace	ImageNet
Deepfakes	0.90±0.04	0.66±0.05	0.75±0.03
NeuralTextures	0.76±0.02	0.60±0.04	0.68±0.03
FaceSwap	0.39±0.03	0.29±0.06	0.31±0.04
FaceShifter	0.59±0.06	0.30±0.06	0.30±0.05

Table 7.9: AUC of the tried methods on the FF++ dataset in the LOO experiment.

	FaRL	ArcFace	ImageNet
Deepfakes	0.28±0.11	0.76±0.05	0.63±0.06
NeuralTextures	0.57±0.04	0.85±0.03	0.72±0.06
FaceSwap	0.95±0.02	0.97±0.02	0.97±0.01
FaceShifter	0.89±0.08	0.97±0.01	0.98±0.01

Table 7.10: FPR@TPR(90) of the tried methods on the FF++ dataset in the LOO experiment.

As we can see in Tables 7.9 and 7.10, model performance is severely reduced on unseen generative methods, being worse than random guessing for FaceSwap and FaceShifter. While being better, the performance on the Deepfakes and NeuralTextures is still somewhat lacking.

The FaRL method performed the best across all benchmarks, followed by the ResNet-50 Imagenet and ArcFace performing the worst, the same results as in the previous experiment.

The ranking of performances on individual generative methods is reversed from the previous experiments, where the performance on FaceSwap and FaceShifter were the best.

7.4.4 Comparison of the proposed method with human performance on the FF++

Since the best performance in previous experiments was achieved by the **FaRL + logistic regression** predictive model, it will be used for the comparison here as well.

The participants of the study [5] were shown 60 images randomly selected from the entire dataset. The metric measured was the accuracy, given the

50:50 real/fake split (see section 3.5.1). To simulate the human life experience in face recognition, the model is first trained on 10 faces. Then, the following 50 were repeatedly predicted and included individually in the training data. This approach hopes to approximate the increasing amount of data seen by human observers. Each subsequent increase in the training set is a real/fake pair of two faces. This is done so that the prior is kept at 0.5.

Accuracy [%]	Deepfakes	NeuralTextures	FaceSwap	FaceShifter
FSDF	75.40±4.69	62.80±7.47	75.40±3.80	93.50±3.28
Human observers	80.41	34.36	75.10	-

Table 7.11: Model accuracy on the FF++ dataset with human observer experiment setup. The human observer data was taken from [5].

As seen in the Table 7.11, the model performance is comparable to the human performance across all metrics; for example, the human observers achieved an average accuracy of below **40%** on NeuralTexures compared to **62.80%** achieved by the FSDF. The average accuracy across all methods for human observers was around **60-70%**.

7.5 DFDC results

Given the lack of information about the used generative model for the specific video in the DFDC [6] dataset, the results (see Table 7.12) are included for completeness and direct comparison with human observers on this dataset.

	Train size = 10	Train size = 100	Train size = full
AUC [-]	0.68±0.08	0.88±0.03	0.92±0.03
FPR@TPR(90) [-]	0.74±0.13	0.33±0.08	0.23±0.08
Accuracy [%]	64.53±0.04	83.07±0.08	85.84±0.07

Table 7.12: Results of the FaRL + logistic regression of the DFDC dataset given several training sizes.

The results achieved on the DFDC dataset are comparable to those achieved on the FF++ dataset.

7.5.1 Comparison of the proposed method with human performance on the DFDC

To best compare the results of human observers on the DFDC dataset[34] with our model FSDF, we use the train size of 100 examples (see section 3.5.2). The human accuracy on the setup proposed in [34] is roughly **75%**, compared to the **83.07%** achieved by FSDF.

7.6 Fake Instagram Influencers Dataset results

The experiments performed on the proposed benchmark FIID show the effectiveness of the proposed method in detecting current high-resolution SOTA examples.

7.6.1 Initial embedding

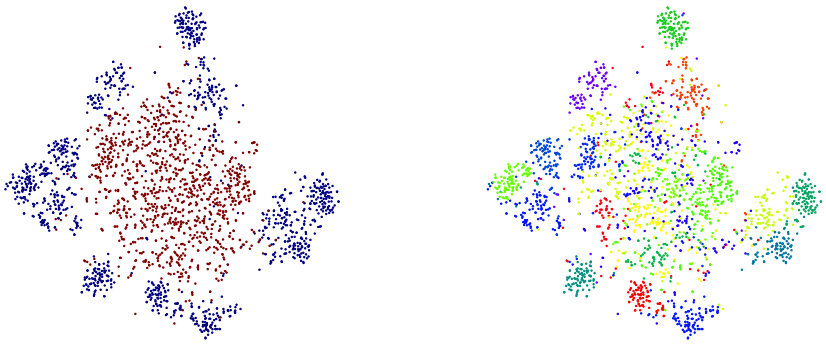


Figure 7.9: Initial T-SNE embedding [64] of the Fake Instagram Influencer dataset using the FaRL embedder. In the left figure, blue dots are the fake influencers, and red dots are the real influencers. In the right figure, each color is one specific influencer.

The figure 7.9 shows that the FaRL embedding on its own is already separating the real and fake influencers without any need for additional external information. Furthermore, the model can cluster images based on the influencer identity.

The clusters of the fake influencers are tighter, showing reduced variance in the face features.

7.6.2 Direct transformer prompting

Since the FaRL [1] architecture is a joint text-image transformer, we can prompt the model directly on whether the image is real or fake. For this purpose, two prompts are used, namely **"a real face"** and **"a fake face"**. These prompts are encoded using the network and passed together with the image to get the probability of the prompt corresponding with the given image.

The resulting accuracy of such process is **52.9%**. Given the same distribution of real and fake influencers, this equates to essentially random guessing.

Racial bias of the FaRL network

The network, however, does have strong predictions for several identities. This realization led to further exploration of what caused the discrepancy.

Name	Real/Fake	Ethnicity	Median fake probability
naina_avtr	Fake	Indian	0.96
Jourdan Dunn	Real	Black	0.91
rozy.gram	Fake	Asian	0.89
tomo_active	Fake	Asian	0.86
ai_spice	Fake	Asian	0.81
demirose	Real	Hispanic	0.76
lilmiquela	Fake	Asian	0.71
piamuehlenbeck	Real	Asian	0.66
caradelevingne	Real	White	0.65
cindymello	Real	Hispanic	0.58
cindybruna	Real	Black	0.49
ashleygraham	Real	Hispanic	0.44
sika_moon_ai	Fake	Hispanic	0.41
emrata	Real	Hispanic	0.23
mirandakerr	Real	White	0.21
gigihadid	Real	White	0.20
haileybieber	Real	White	0.16
bermudaisbae	Fake	White	0.13
fit_aitana	Fake	Hispanic	0.08
fionapellegrini	Fake	White	0.07
elizabethcturner	Real	White	0.07
gioalemann	Fake	White	0.05
the_natalia_novak	Fake	White	0.03
millasofiafin	Fake	White	0.03

Table 7.13: Probability of influencers to be fake in the FIID. The median fake probability is calculated from all images of the given infulencer.

As we can see in the Table 7.13, the network prediction has little to none correlation to the real/fake. However, the ethnicity of the influencer has a major role.

Out of the ten most likely fake influencers, nine out of ten are not white. On the other side of the spectrum, one out of ten influencers are not white. This distribution points to some bias in the training data used.

7.6.3 Leave-one-identity-out experiment

For the Leave-one-identity-out (LOIO) experiment, one identity from each real and fake influencer is set aside for the testing set.

All together 144 pairs of testings set are possible. All images from the remaining 22 influencers are used.

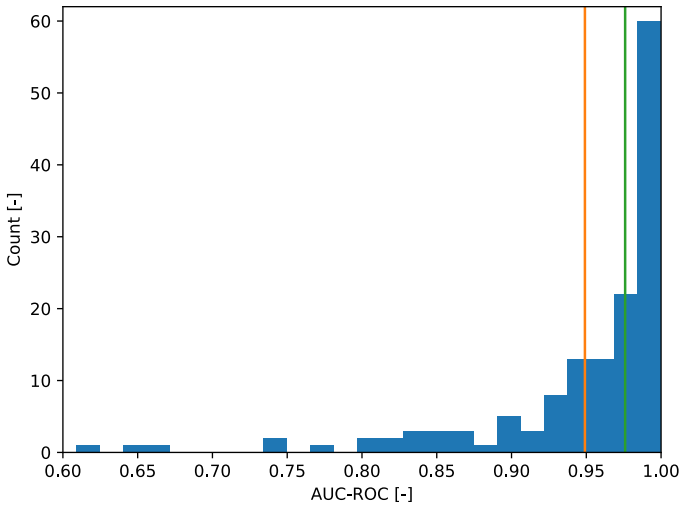


Figure 7.10: Distribution of the AUC for the LOIO experiment on the FIID given full train set. The orange vertical line is the mean AUC, the green line is the median AUC.

The mean AUC of the method on the full FIID dataset for all test pairs is 0.95 (the median value is 0.98). More detail can be seen in Figure 7.10.

7.6.4 Detection performance versus the number of training images on FIID

In this experiment, only a given number of images is taken from each influencer.

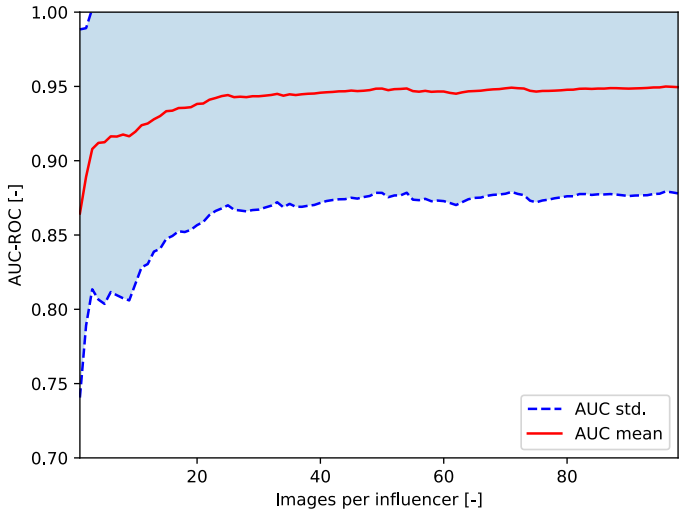


Figure 7.11: Depence of the AUC given the size of the training size.

As we can see in Figure 7.11, only around 20 images are required per influencer to achieve maximal performance. The resulting performance is shown for all images individually. However, when aggregated per influencer, the resulting accuracy is **100%**. This could partially be due to the small dataset size.

7.6.5 Variable test size of the FIID dataset

Since the proposed FSDF model achieves perfect detection performance on the aggregate score if all images are used for each influencer in the testing set, we examine the model performance given a limited amount of examples per influencer in the testing set.

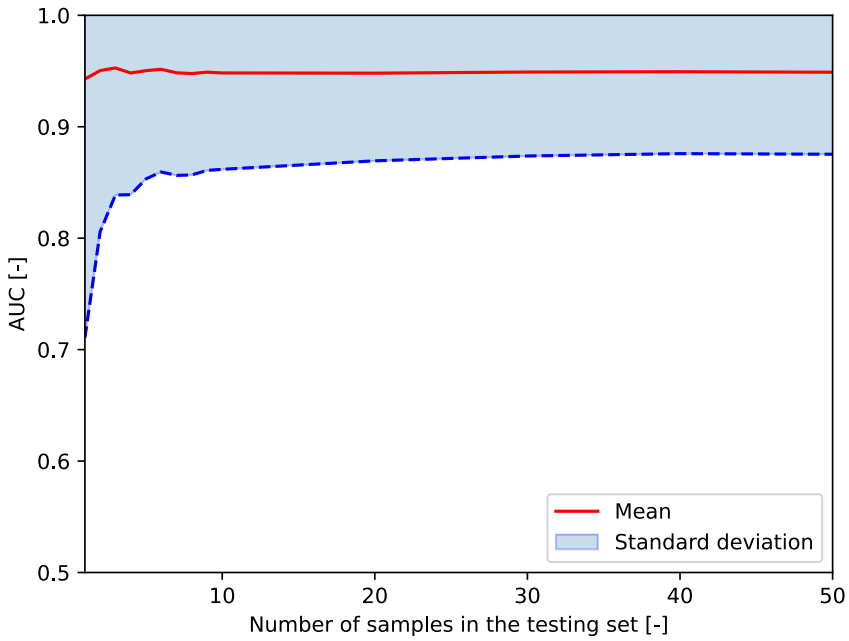


Figure 7.12: Depence of the AUC on the full train set given variable amount of samples in the test set before aggregation.

The Figure 7.12 shows that the mean performance of the model does not change with the amount of data before score aggregation. However, variance decreases, leading to more consistent results.

7.6.6 Summary

The experiments show that the model can correctly detect fake influencers. The model requires only around 20 images per influencer to achieve maximal performance. However, the performance change between only several samples up to 20 is minimal.

Only several images per influencer are required to determine whether all page content is AI-generated.

7.7 Medical dataset results

The experiments on the medical dataset are used to determine whether the methods utilized for the faces are easily translatable to other modalities.

7.7.1 Initial embedding

Since the FaRL method is specially trained for faces, its usage for the medical problem is questionable. Therefore, the ImageNet pre-trained ResNet-50 + logistic regression is tried as the baseline classifier.

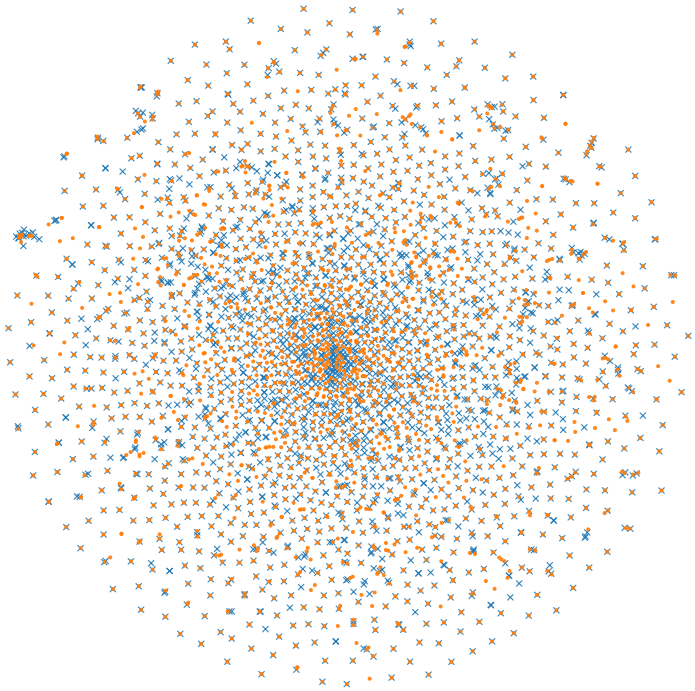


Figure 7.13: Initial ResNet-50 ImageNet T-SNE embeddinf ot the medical dataset. Blue crosses are the real images, orange dots are the fake images.

Most of the real/fake pairs are mapped onto each other rather than separated by the real/fake class (see Figure 7.13). This shows that the ImageNet

embedding is not capable of any separation.

When logistic regression is used on the ImageNet embedding the model perfectly fits the training set while not generalizing on the test set, achieving an average accuracy of **50.5%**.

Since the size of the train set is around thousand images and the size of the embedding is 2048, additional embedding size reduction is used, to determine, whether this is the main issue.

The performance on the train set reduces with the smaller embedding size. However, no change occurs on the test set.

7.7.2 Training a small convolutional neural network

Since the embedding approach does not seem to work, a small network is trained.

The network is implemented as a CNN with 3 residual blocks [3], with max-pooling layers in between, followed by a single linear layer. The size of the model is 5 MB.

We used the Adam optimizer [65], with hyperparameters ($\beta_0 = 0.9, \beta_1 = 0.999$), with initial learning rate $lr = 0.0001$.

Additional random horizontal and vertical flipping is introduced for data augmentation, and then random white Gaussian noise is added to the image.

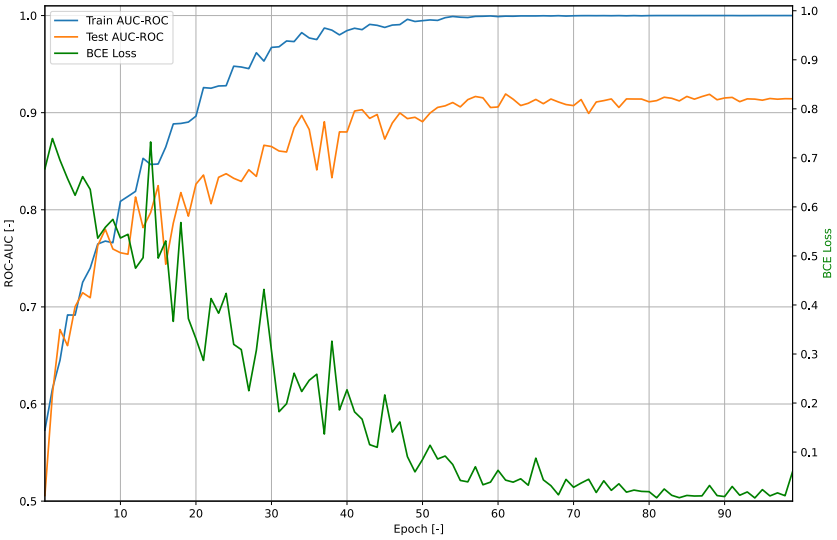


Figure 7.14: Progression of training on the medical dataset, given 20% of the data being in the test set.

In Figure 7.14, we can see that the network is capable of perfectly fitting on the training set while still generalizing somewhat well on the testing set.

In the train size experiment, the model from the epoch, where training AUC is at least 0.99 is then used for testing. If no such value is reached,

the model value after 100 epochs is used instead (this did not occur in our experiment).

7.7.3 Model performance based on the variable training size

During this experiment, the ratio of the data in the training set ranges from 0.1 up to 0.9 in increments of 0.1. Given the maximal size of 2387, this roughly equates to 240 additional images in the train set per ratio increase. For each ratio value, 5-fold cross-validation is applied and summarized as a mean and standard deviation.

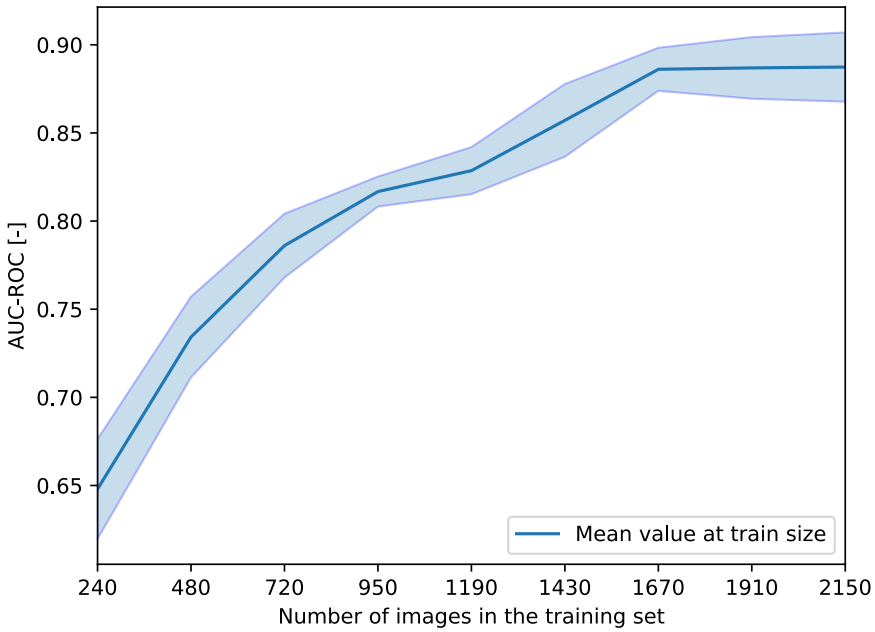


Figure 7.15: Dependence of model performance based on the test set given the size of the medical training dataset.

From Figure 7.15 we can see that the model capable of generalizing. However, its performance heavily depends on the train size, with the plateau reaching around 1670 images out of 2387 maximal.

Mean model accuracy on the test set for the 1670 images in the training set was **91.5%** (with 99.8% accuracy on the train set).

7.8 Comparison with state-of-the-art

Given that SOTA methods used in the two datasets use standard deep learning approaches, their performance will be higher than our few-shot approach.

The direct comparison, however, shows whether the high amount of training data is required or having a good initial representation is all that is necessary.

7.8.1 SOTA on the FaceForensics++ dataset

Method	Low Quality		High Quality	
	Accuracy	AUC	Accuracy	AUC
Xception[66]	86.86	89.30	95.73	96.30
MADD[30]	88.69	90.40	90.40	97.60
M2TR[67]	92.35	94.22	98.23	99.48
SFDG[31]	92.28	95.98	98.19	99.53

Our method	Mixed Quality	
	Accuracy	AUC
Train=10	72.13	79.01
Train=100	86.66	92.71
Train=1000	91.99	96.74

Table 7.14: Summary table of the SOTA on FF++ compared with our method (FaRL embedding with logistic regression). Data taken from [31].

Since we did not split the dataset into a low-quality and high-quality group, we used a "Mixed Quality" group. The assumption is that the resulting metrics should be placed between low and high quality.

The proposed method achieves performance that is slightly worse than the current SOTA when at least 100 samples are present in the train set (see Table 7.14). With 1000 samples, the method has already achieved results similar to those of SOTA.

10 samples were shown not enough to achieve comparable performance in the few-shot setup.

7.8.2 SOTA on the DFDC dataset

The main ranking provided by the authors of the DFDC [6] is the log BCE loss. This metric does not allow for any cross-comparison with other datasets. The AUC metric is also stated in a supplementary role with only several first places, shown in Figure 7.16

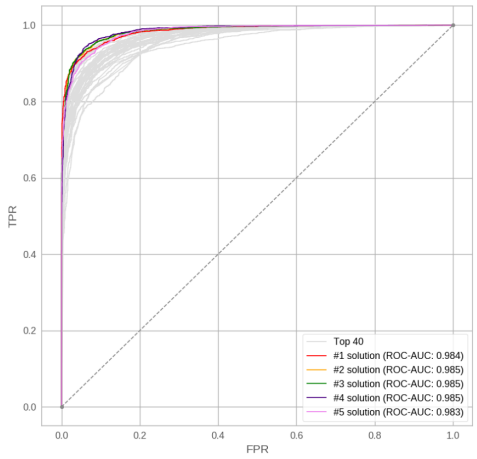


Figure 7.16: ROC of the top 40 submissions of the DFDC. Taken and edited from [6].

The top 5 AUC values stated in the paper are in the range of 0.984 ± 0.001.

	Top 5 methods average	**Our method**		
		Train = 10	**Train = 100**	**Train = 250**
AUC	0.98	0.68	0.89	0.92

Table 7.15: Summary table of the SOTA on DFDC dataset compared with our method (FaRL embedding with logistic regression).

Our method performs significantly worse on the DFDC dataset (as seen in Table 7.15). However, this is likely caused by the fact that the highest amount of data in the training set is 250, compared to 1000 for the FF++. The AUC values for 10 and 100 are comparable, and the differences between them are consistent between the datasets.

Chapter 8

Discussion

8.1 Efficiency of few-shot learning for deepfake detection

The performance of our proposed few-shot model is similar to the performance of the SOTA methods on tested datasets, namely FF++ [5] and the DFDC dataset [5]. While overall metrics are slightly worse than the SOTA, the overall time required for the training and inference is negligible.

Adding additional samples to the dataset and changing the final model takes several seconds at maximum since we only have to do a single forward pass through the embedding model, followed by retraining of the logistic regression.

Most SOTA methods take several days to learn on the already preprocessed dataset while requiring a tremendous amount of data. Our method, on the other hand, requires faces from only several videos.

The initial embedding of the FaRL model [1], while not directly capable of deepfake detection, is suitable for the task. Similarly, the ResNet-50 pre-trained on the ImageNet [4, 3] dataset shows great promise, given that the dataset contains no faces whatsoever.

The ArcFace model was shown to not generalize well for the task of deepfake detection. Since the original task of the model is face recognition, the network seems to cluster similar faces together and does not focus on artifacts present in the generative models.

Both the FaRL and the ImageNet pre-trained ResNet are more general in their original tasks. FaRL was initially used for various tasks, while ImageNet is a classification dataset.

Furthermore, more complex classifiers were tried instead of simple logistic regression; however, for the small train sizes, these tend to overfit, while for bigger samples, their performance is only slightly better.

The proposed model performs flawlessly on the novel Instagram influencer dataset. Showing that given enough high-quality photos, the performance increases drastically. The main difference between the proposed FIID and the existing FF++ and DFDC is the higher resolution and the fact that each identity is generated by a separate model. Furthermore, each account is likely

to apply its specific image transformation, be it additional image filtering, color correction, etc.

8.2 Cross-domain applicability of embedding models

We have tried to apply the same approach as with the face datasets on an unrelated medical image dataset. The fake images were generated by a comparatively simple method, namely Poission image editing [14].

The embedding of the ResNet-50 pre-trained on the ImageNet could not correctly detect fake images, reaching performance equivalent to random guessing.

However, a small convolutional network was trained on the given dataset. The resulting model could detect fakes with an accuracy of around 91% percent. The number of images needed to achieve such accuracy was, however, several orders of magnitude higher than the expected amount of fake medical images that would occur in reality.

Given the fact that the access to medical deepfakes is severely limited when compared to faces or other modalities, combined with the fact that the embedding of models which were pre-trained on unrelated datasets do not seem to generalize and the low resolution and highly noisy images which most of the medical imaging devices generate, the few-shot approach is not ideal for the detection of medical deepfakes.

8.3 Limitations of the thesis

The main limitation of the thesis is amount of examples in the datasets. Given the time and memory requirements, only subsets of the commonly used dataset FF++ [5] and DFDC [6] were used.

Same could be said for the novel FIID dataset. However, since the model seems to be capable of learning from only several photos, the issue might not be as significant.

The novel medical dataset suffers from a lack of distinct generative methods, moreover using simple low-level vision techniques.

8.4 Suggested future research

One of the possible directions for future research is increasing the number of embedding models and classifiers tried. Since only three distinct models (and their variations) were tried, better-performing embedders are likely to exist.

The medical dataset could also benefit from increasing in size, be it in the amount of generative methods or sources of images.

Chapter 9

Conclusion

In this thesis, we have developed a deepfake detection method utilizing few-shot learning. We combine the output embedding of a model pre-trained for a different task, followed by a simple linear classifier. Each proposed combination of embedding model and classifier is empirically compared on existing datasets in several experiments.

On the closed-world setup (meaning the same deepfake generative model was used for both the train and test datasets), a small amount of training samples is required to achieve sufficient performance. When the generative model is missing from the test dataset, the overall performance is significantly reduced compared to the closed-world setup. When compared to the current SOTA methods, our model's performance is slightly worse but still remains comparable, while the time requirement and computational complexity for training are significantly lower. The performance of the proposed model is comparable to the performance of human observers.

We have created two novel benchmarks: the Fake Instagram Influencer Dataset (FIID) and the Medical Fake Image Dataset (MFID). For the higher-quality images in the FIID, our method seems capable of perfect classification without much additional work required. On the MFID, the proposed detection method is not capable of correctly detecting fake images. This is likely caused by the embedding model being trained on a completely different modality, significantly lower image quality, and increased noise when compared to face images. We trained a small-scale convolutional neural network and measured its performance, given the number of training samples. The network requires more than ten times the amount of data compared to the FSDF method on the face deepfake detection task to achieve somewhat comparable performance.

Bibliography

[1] Y. Zheng, H. Yang, T. Zhang, J. Bao, D. Chen, Y. Huang, L. Yuan, D. Chen, M. Zeng, and F. Wen, "General facial representation learning in a visual-linguistic manner," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18697–18709, 2022.

[2] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

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

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

[5] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1–11, 2019.

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

[7] "Instagram page of aitana lopez." `https://www.instagram.com/fit_aitana/`. Accessed: 2024-04-05.

[8] "Instagram page of natalia novak." `https://www.instagram.com/the.natalia.novak/`. Accessed: 2024-04-05.

[9] SkyNews, "Ukraine war: Deepfake video of zelenskyy telling ukrainians to 'lay down arms' debunked." `https://news.sky.com/story/ukraine-war-deepfake-video-of-zelenskyy-telling-ukrainians-to-lay-down-arms-debunked-12567789`. Accessed: 2024-04-05.

[10] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras, "Neural face editing with intrinsic image disentangling," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5541–5550, 2017.

[11] J. Sun, X. Wang, Y. Zhang, X. Li, Q. Zhang, Y. Liu, and J. Wang, "Fenerf: Face editing in neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7672–7682, 2022.

[12] Y. Mirsky and W. Lee, "The creation and detection of deepfakes: A survey," *ACM computing surveys (CSUR)*, vol. 54, no. 1, pp. 1–41, 2021.

[13] Y. Nirkin, Y. Keller, and T. Hassner, "Fsgan: Subject agnostic face swapping and reenactment," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7184–7193, 2019.

[14] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," 2003.

[15] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

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

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

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

[19] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2387–2395, 2016.

[20] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," 2019.

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

[22] J. Wakefield, "Deepfake presidents used in russia-ukraine war," *BBC News*, 03 2022.

[23] C. Nast, "Deepfakes, cheapfakes, and twitter censorship mar turkey's elections," 05 2023.

[24] S. Cole, "'you feel so violated': Streamer qtcinderella is speaking out against deepfake porn harassment." `https://www.vice.com/en/article/z34pq3/deepfake-qtcinderella-atrioc`. Accessed: 2024-05-05.

[25] A. Garg, "Rashmika mandanna deepfake: 3 years jail, rs 1 lakh fine, govt sends rule reminder to social media platforms." `https://www.indiatoday.in/technology/news/story/rashmika-mandanna-deepfake-3-years-jail-rs-1-lakh-fine-govt-sends-rule-reminder-to-social-media-platforms-2460104-2023-11-07`. Accessed: 2024-05-05.

[26] I. Rahman-Jones, "Taylor swift deepfakes spark calls in congress for new legislation." `https://www.bbc.com/news/technology-68110476`. Accessed: 2024-05-05.

[27] B. L. Nadeau, "Italian prime minister giorgia meloni seeking damages of $108,200 in deepfake porn trial." `https://edition.cnn.com/2024/03/22/europe/giorgia-meloni-italy-deepfake-porn-intl/index.html`. Accessed: 2024-05-05.

[28] "Instagram page of lilmiquela." `https://www.instagram.com/lilmiquela/`. Accessed: 2024-05-05.

[29] "Instagram page of milla sofia." `https://www.instagram.com/millasofiafin/`. Accessed: 2024-05-05.

[30] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Multi-attentional deepfake detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2185–2194, 2021.

[31] Y. Wang, K. Yu, C. Chen, X. Hu, and S. Peng, "Dynamic graph learning with content-guided spatial-frequency relation reasoning for deepfake detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7278–7287, 2023.

[32] B. Huang, Z. Wang, J. Yang, J. Ai, Q. Zou, Q. Wang, and D. Ye, "Implicit identity driven deepfake face swapping detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, 2023.

[33] Z. Cai, S. Ghosh, K. Stefanov, A. Dhall, J. Cai, H. Rezatofighi, R. Haffari, and M. Hayat, "Marlin: Masked autoencoder for facial video representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1493–1504, 2023.

[34] M. Groh, Z. Epstein, C. Firestone, and R. Picard, "Deepfake detection by human crowds, machines, and machine-informed crowds," *Proceedings of the National Academy of Sciences*, vol. 119, no. 1, p. e2110013119, 2022.

[35] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.

[36] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," *Advances in neural information processing systems*, vol. 20, 2007.

[37] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National science review*, vol. 5, no. 1, pp. 44–53, 2018.

[38] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[39] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5586–5609, 2021.

[40] E. Triantafillou, R. Zemel, and R. Urtasun, "Few-shot learning through an information retrieval lens," *Advances in neural information processing systems*, vol. 30, 2017.

[41] R. Keshari, M. Vatsa, R. Singh, and A. Noore, "Learning structure and strength of cnn filters for small sample size training," in *proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9349–9358, 2018.

[42] Y.-X. Wang and M. Hebert, "Learning from small sample sets by combining unsupervised meta-training with cnns," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[43] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*, pp. 1126–1135, PMLR, 2017.

[44] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," *Advances in neural information processing systems*, vol. 29, 2016.

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

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

[47] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki, "Laion-400m: Open dataset of clip-filtered 400 million image-text pairs," 2021.

[48] Y. Liu, H. Shi, H. Shen, Y. Si, X. Wang, and T. Mei, "A new dataset and boundary-attention semantic segmentation for face parsing.," in *AAAI*, pp. 11637–11644, 2020.

[49] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[50] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[51] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013.

[52] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[53] D. Huang and F. De La Torre, "Facial action transfer with personalized bilinear regression," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II 12*, pp. 144–158, Springer, 2012.

[54] "Gitlab repository of the deepfakes method." `https://github.com/deepfakes/faceswap`. Accessed: 2024-05-08.

[55] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "Faceshifter: Towards high fidelity and occlusion aware face swapping," *arXiv preprint arXiv:1912.13457*, 2019.

[56] D. Huang and F. De La Torre, "Facial action transfer with personalized bilinear regression," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II 12*, pp. 144–158, Springer, 2012.

[57] E. Zakharov, A. Shysheya, E. Burkov, and V. S. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," *CoRR*, vol. abs/1905.08233, 2019.

[58] Y. Nirkin, Y. Keller, and T. Hassner, "Fsgan: Subject agnostic face swapping and reenactment," 2019.

[59] W. Commons, "Pcmglee, martinthoma," 2018.

[60] Y. Mirsky, T. Mahler, I. Shelef, and Y. Elovici, "Ct-gan: Malicious tampering of 3d medical imagery using deep learning," 2019.

[61] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M.-A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, c. Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H.-C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, 2015.

[62] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[63] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-stage dense face localisation in the wild," 2019.

[64] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

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

[66] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807, 2017.

[67] 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.