COMPARATIVE ANALYSIS OF DEEPFAKE DETECTION METHODS

by

Burak İkan Yıldız

B.S., Computer Engineering, Boğaziçi University, 2020

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2025

ACKNOWLEDGEMENTS

I would like to thank my supervisor Asst. Prof. Berk Gökberk for his continuous support through the Master's Program. I could not accomplish this without his help.

ABSTRACT

COMPARATIVE ANALYSIS OF DEEPFAKE DETECTION METHODS

Most security systems use biometric authentication pipelines to verify their users. Since one of the foundational aspects of biometric data is the human face, detecting deepfakes is essential to maintain the authenticity of identity and securing such systems. Deep learning models trained for detection of artificial faces commonly utilize transfer learning techniques where a model pre-trained on one task is fine-tuned for a new, related task. This situation arises because deepfake detection models have difficulty in progressing further and hitting a plateau due to the lack of domain-specific knowledge in their pre-trained dataset. In this study, we propose that pretraining state-of-the-art models using a specialized face dataset, namely Glint360K, enhances their performance significantly compared to base models pre-trained using the ImageNet dataset. While introducing a domain-specific dataset for models, we also aim to improve the models' ability of generalization by incorporating various regularization techniques. In particular, label smoothing was applied to models while training to reduce the effect of initial identities in the pre-training face dataset. Last but not least, our research presents a novel dataset with 4,252 real and 2,603 fake face videos created to mimic real-life situations with diverse lighting, device settings, and resolution differences. We utilized both this dataset and commonly used datasets to assess our proposed approach and state-of-the-art models, highlighting its strengths and weaknesses. The qualitative results show that our approach successfully force models to focus on face-related features. On the other hand, quantitative results indicate that models pre-trained on Glint360K with applied label smoothing outperform those pre-trained with ImageNet weights by 1.5% on average.

ÖZET

DEEPFAKE ALGILAMA YÖNTEMLERİNİN KARŞILAŞTIRMALI ANALİZİ

Çoğu güvenlik sistemi, kullanıcılarını doğrulamak için kimlik doğrulama akışları kullanır. Biyometrik verilerin temel unsurlarından biri insan yüzü olduğundan, kimliklerin gerçekliğini korumak ve sistemlerin güvenliğini sağlamak için derin sahtelerin tespit edilmesi şarttır. Yapay yüzlerin tespiti için eğitilen derin öğrenme modelleri, genellikle bir görev üzerinde önceden eğitilmiş bir modelin yeni bir görev için ince ayarlandığı transfer öğrenme tekniklerini kullanır. Bu durum, derin sahte yüz tespit modellerinin önceden eğitilmiş veri kümelerindeki alana özgü bilgi eksikliği nedeniyle daha fazla ilerlemekte zorlanması ve bir platoya ulaşması nedeniyle ortaya çıkmaktadır. Bu çalışmada, son teknoloji ürünü modellerin Glint360K gibi özel bir yüz veri kümesi kullanılarak ön eğitime tabi tutulmasının, ImageNet veri kümesi kullanılarak ön eğitime tabi tutulan modellere kıyasla performanslarını önemli ölçüde artırdığını öneriyoruz. Modeller için alana özgü bir veri kümesi sunarken, çeşitli düzenleme tekniklerini de dahil ederek modellerin genelleme yeteneğini geliştirmeyi amaçlıyoruz. Özellikle, eğitim öncesi veri kümesindeki başlangıç kimliklerinin etkisini azaltmak için eğitim sırasında modellere etiket yumuşatma uygulanmıştır. Son olarak, araştırmamız, farklı ışık, cihaz ayarları ve çözünürlük farklılıklarına sahip gerçek yaşam durumlarını taklit etmek için oluşturulan 4.252 gerçek ve 2.603 sahte yüz videosundan oluşan yeni bir veri kümesi sunmaktadır. Nitel sonuçlar, yaklaşımımızın modelleri yüzle ilgili özelliklere odaklanmaya başarıyla zorladığını göstermektedir. Öte yandan, nicel sonuçlar, Glint360K üzerinde etiket yumuşatma uygulanarak önceden eğitilen modellerin, ImageNet ile önceden eğitilen modellerden ortalama %1.5 daha iyi performans gösterdiğini ortaya koymaktadır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . x
LIST OF SYMBOLS . . . xii
LIST OF ACRONYMS/ABBREVIATIONS . . . xiii
1. INTRODUCTION . . . 1
1.1. Contributions . . . 7
1.2. Organization of Thesis . . . 10
2. RELATED WORK . . . 11
2.1. Image-Based Approaches . . . 13
2.2. Video-Based Approaches . . . 17
2.3. Hybrid Approaches . . . 20
2.4. Audio, Biological Signals, and Beyond . . . 22
2.5. Deepfake Detection Datasets . . . 24
3. METHODOLOGY . . . 29
3.1. Our Dataset . . . 30
3.2. Pretraining with Glint360K . . . 32
3.3. Compared Methods . . . 34
3.3.1. IIL - Implicit Identity Leakage . . . 34
3.3.2. LAA-Net – Localized Artifact Attention Network . . . 38
3.3.3. M2TR – Multi-Modal Multi-Scale Transformer . . . 40
3.3.4. LSC – Learning Self-Consistency Approach . . . 43
3.3.5. ICT - Identity Consistency Transformer . . . 47
3.3.6. XceptionNet . . . 49
3.3.7. ResNet Baseline . . . 49
3.3.8. Vision Transformer Baseline . . . 50

3.4. Our Approach 51
3.4.1. Implementation Details 54
4. EXPERIMENTS AND RESULTS 57
4.1. Datasets Used In Experiments 57
4.1.1. FaceForensics++ 58
4.1.2. Celeb-DF v2 59
4.1.3. ForgeryNet 59
4.1.4. Our Dataset 60
4.2. Experimental Setup 61
4.3. Intra-dataset Results 62
4.4. Cross-Dataset Results 64
4.5. Label Smoothing Analysis 68
4.6. Qualitative Analysis 70
4.7. Robustness Experiments 72
4.8. Face Crop Margin Analysis 76
5. CONCLUSION 80
REFERENCES 82

LIST OF FIGURES

Figure 1.1. Deepfake generation and deepfake detection pipelines. 8

Figure 3.1. Implicit identity leakage problem. Model implicitly learning identities in the training data - results in false predictions when unknown identity shown. 35

Figure 3.2. IIL deepfake detection model architecture. Detection loss and classification loss work together to decide final label. 37

Figure 3.3. LAA-Net deepfake detection model architecture. 3 branches of classification, heatmap and self-consistenct works together to decide final label. 39

Figure 3.4. LAA-Net pseudo-fake generation. With the help of this process, ground truth labels can be generated to be used in training. 41

Figure 3.5. M2TR deepfake detection model architecture. 2 branch approach helps to focus not only visual image but also frequency stream of image. 42

Figure 3.6. LSC deepfake detection model architecture. Generating synthetic data enhances training process to focus on fake areas of the image. 45

Figure 3.7. ICT deepfake detection model architecture. Identity loss and consistency loss work harmonically to further advance detection process. 48

Figure 3.8. Transfer learning approach used in deepfake detection. 54

Figure 4.1. AUC change with respect to label smoothing epsilon value. For the purpose of clearity of the graph, 3 models selected, namely ResNet, ViT, M2TR. 69

Figure 4.2. AUC change with respect to compression quality factor. 74

Figure 4.3. AUC change with respect to gaussian blur sigma value. 76

LIST OF TABLES

Table 2.1. Summary of deepfake detection related work. I, V, A stands for image, video, audio respectively. 24

Table 2.2. Comparison of major deepfake detection datasets. Each dataset is characterized by its year, scale (number of videos/images), source of original media, data modality. V,A,I stands for video, audio, image respectively. 28

Table 4.1. Sample images from datasets used in our work. Table contains 2 fake and 2 real images from each dataset. 58

Table 4.2. Real and fake examples from our dataset. There are 5 rows for 5 sample identity with 4 real and 4 fake images as columns. 61

Table 4.3. Brief summary of deepfake detection datasets used in our experiments. Video datasets splitted to frames for training and testing for image based detectors. 61

Table 4.4. Intra-dataset(FF++) evaluation of methods. Bold values show the best result for each method. 64

Table 4.5. Cross-dataset(Celeb-DF-v2 and ForgeryNet) evaluation of methods. Bold values show the best result for each method in testing dataset separately. 65

Table 4.6. Evaluation of methods on our custom dataset. Bold values show the best result for each method. 67

Table 4.7. Heatmap examples of vision transformer model pretrained with ImageNet and Glint360K. Rows contains datasets and columns contains results of 2 examples of ImageNet pretrained ViT and Glint360K pretrained ViT. Last colum especially draws heatmap of person who wear eyeglasses. 71

Table 4.8. Robustness of models under different JPEG compression quality factors (AUC %). Test is conducted on FF++ test set. 73

Table 4.9. Robustness of models under Gaussian blur (AUC %). Test is conducted on FF++ test set. 75

Table 4.10. Robustness of models under horizontal flip (AUC %). Test is conducted on FF++ test set. 77

Table 4.11. Sample images for 5 different identity with varying margin crop levels. 78

LIST OF SYMBOLS

c	Class confidence scores
g	Ground-truth bounding-box coordinates
l	Predicted bounding-box coordinates by the Artifact Detection Module
L	Overall loss combining detection and classification
L_{cls}	Classification loss (binary real vs. fake)
L_{conf}	Confidence loss for anchor classification
L_{det}	Detection loss combining confidence and location terms
L_{loc}	Location (bounding-box regression) loss
N	Number of positive anchor boxes
p	Patch size (e.g. 16×16) in the Vision Transformer
Q	JPEG compression quality factor
$x \in \{0, 1\}$	Indicator for matching an anchor to a ground-truth artifact
α	Weighting factor for the location loss in the detection loss
β	Trade-off scalar between detection and classification losses
ϵ	Label-smoothing factor
σ	Gaussian blur kernel radius

LIST OF ACRONYMS/ABBREVIATIONS

AE	AutoEncoder
CNN	Convolutional Neural Network
DFDC	Deepfake Detection Challenge
E-FPN	Enhanced Feature Pyramid Network
FF++	Face Forensics
FInfer	Frame Inference Network
GAN	Generative adversarial network
ICT	Identity Consistency Transformer
IIL	Implicit Identity Leakage Network
LAA-Net	Localized Artefact Attention Network
LSC	Learning Self-Consistency Network
M2TR	Multi-modal Multi-scale Transformer
RNN	Recurrent Neural Network
rPPG	remote photoplethysmography
SBI	Self-Blended Images Network
SVM	Support Vector Machine
ViT	Vision transformer

1. INTRODUCTION

Deep learning has completely changed the discipline of visual computing over the past ten years. Image recognition, object detection, and segmentation that much exceed previous methods in both accuracy and scalability have been made possible by convolutional neural networks (CNNs) [1] and related architectures. Deep learning models, for instance, can today identify objects and faces in images with almost human or even superhuman performance. Beyond simple recognition, deep learning has enabled generative visual capabilities: neural networks can synthesize quite realistic images and videos from data alone. This dual ability to create visual content and comprehend visual content has opened new directions in computer vision. But these same developments also bring fresh difficulties, best shown by the emergence of "deepfakes." Deep generative models allow anyone the ability to create or manipulate visual media in ways that are hard to see with the unaided eye. Deep learning's effects on visual computing have thus been two-edged: it has enabled a new generation of visual forgeries testing the boundaries of our capacity to tell real from fake, while also driving amazing progress in visual understanding and creation.

Deepfakes are typically synthetic or manipulated media in which deep neural networks convincingly change someone's face or voice. A "deepfake" in popular imagination is usually a video in which one person's face has been replaced on another's body to create the impression that someone said or did something they never really did. Originally used on internet forums where hobbyists shared the first face-swapped celebrity videos produced with autoencoder networks, the term itself – a portmanteau of "deep learning" and "fake – started about 2017. The language and the technology have expanded since then. From identity swaps—that is, replacing Person A's face with Person B's face in a video—to expression or speech reenactment—that is, changing a person's facial expressions or lip movements to replicate another's performance, while keeping the person's identity the same—today deepfakes cover a spectrum of face manipulation scenarios. These are united by the synthesis of photorealistic human faces

or facial motions challenging to differentiate from real-world footage using advanced deep learning models. The revolution in generative modeling—algorithms including autoencoders, generative adversarial networks (GANs) [2], and other neural architectures can learn from vast sets of real faces and then generate deceptively realistic fake images or videos—has made deepfakes possible. Hollywood special effects professionals dominated believable face substitution in video just a few years ago; today, anyone with a decent computer and some technical knowledge can produce a realistic face-swap video. Deep learning's ability drives much of this democratization of advanced video fakery, which seriously compromises the integrity of digital media.

Deep learning not only makes deepfakes feasible but also a main tool for their detection. Furthermore giving researchers tools to identify the subtle fingerprints of manipulation is the very development in neural networks that enables forgers to synthesis faces. By picking up artifacts too faint for humans to detect, a neural network that has learnt to classify images or detect anomalies in pictures can theoretically be trained to tell apart real from fake faces. Deepfake generation and detection are now engaged in an armaments race. On one side, generative models—often the same kinds used for legitimate activities like photo editing or data augmentation—are always improving in producing more realistic and less imperfect forgeries. Conversely, increasingly motivated by machine learning, forensic analysis methods seek to find the minor anomalies or artifacts even introduced by even the best deepfake methods. Common in security disciplines, this cat-and- mouse dynamic exists whereby one side improves and the other must change. From a research standpoint, the deepfake phenomenon is especially fascinating and difficult since deep learning sits at the center of both sides – enabling the cat and empowering the mouse.

Deep learning-based detection emerged naturally as a mirror image of the general change in computer vision from heuristic to learned features. Rather than depending on a single cue like blinking or head tilt, researchers began training convolutional neural networks straight on datasets of real and synthetic images to let the network find whatever unique patterns might exist. The first sizable public datasets of deepfakes emerged

by about 2018–2019, allowing supervised learning for deepfake detection. For instance, the 2019 FaceForensics++ [3] dataset included hundreds of videos altered using several techniques (including both classic computer graphics approaches and modern neural techniques) together with their original source videos. Having such data at hand, one could teach a CNN to identify an input video frame (or series of frames) as either real-world or altered. Achieving high accuracy on benchmark tests, often well above what untrained human observers could consistently do, these CNN-based detectors rapidly proved far more effective and general than previous manual methods. For this work, several network configurations were investigated. First attempts usually refined fine-tuned architectures borrowed from image classification research (e.g., VGG or ResNet models) for the binary real-fake classification challenge. Soon it was discovered that some designs were better than others; for example, the Xception network [4], a CNN first created for image classification—was shown to be especially adept at deepfake detection and became a common baseline model in the field. Xception's depthwise-separable convolution design, according to researchers, helped concentrate on minor texture discrepancies typical of fakes. Other contemporary CNN designs such as EfficientNets [5] and Vision ResNets [6] were also used since the community experimented to identify the most discriminating feature extractors for this particular classification challenge. Sometimes ensembles of several CNN models were used to improve detection performance even more. These techniques essentially learned a rich hierarchy of features – from low-level pixel artifacts (perhaps detecting odd interpolation patterns or frequency discrepancies) up to high-level inconsistencies (such as unnatural facial detail or movements) – without human specification what exact feature to search for – by training deep networks on vast corpora of fake and real data. This data-driven approach rapidly enhanced detection capabilities; for a while it seemed that deep learning could solve the deepfake problem just as quickly as it had enabled it.

As with many security-related chores, though, a solution that performs "in the lab" on carefully selected datasets does not always hold up in the real world. Deepfake detectors trained on a specific dataset or generation method soon found by researchers to often fail to generalize to fakes produced by different techniques or to videos gath-

ered in different conditions. A CNN trained on FaceForensics++, for instance, may show great accuracy on test fakes similar to those seen in training, but if faced with a new type of manipulation—say, a new GAN-based face-swap not represented in the training set—the performance could drop dramatically. This lack of generalization is a serious issue; practically, we want detectors that can detect any deepfake, not only the particular taste of deepfakes they were exposed to during training. The problem stems in part from deep learning models, if not closely guided, latching on surface patterns that match "fake" in the training data but are not really fundamental forgery. A detector might learn, for example, that a specific generative model's "signature" noise pattern indicates a fake – which works until it detects a fake from another pipeline devoid of those cues. A certain compression artifact might also indicate a fake. One particularly fascinating and concerning occurrence found in recent research is "implicit identity leakage" in deepfake detection systems. This implies that rather than only considering how the image was produced, the model might unintentionally be depending on who the person is as a cue for spotting fakeness. The model might cheat, for instance, by learning to identify that actor's face – labeling anything with that face as "fake" (since in the training data that face usually appeared in fake videos). If the training set consists of mostly real videos of other people and many fake videos of a certain famous actor. This clearly does not generalize to fresh faces, and without careful study, it is a type of overfitting that is not immediately evident. More subtly still, a detector may learn variations in identity between the face and the surrounding context—that is, a mismatch between the face and the body or voice if those are also present—that is, essentially learning to spot when "the face doesn't match the person." While that can be a clue in some deepfakes, it is not a reliable indicator of manipulation in general – especially as deepfakes improve or if the same person's face is used – and it could cause false positives or a failure to identify fakes on fresh identities. When naively trained, essentially CNN-based deepfake detectors tend to be brittle across domain changes: they might perform remarkably well on known scenarios but break when faced with unexpected cases. Driven by this insight, which emphasizes resilience and generalization, the next phase of development in deepfake detection techniques has started.

Two main dimensions have expanded modern deepfake detection strategies: more complex modeling (architectures and training regimes) and more thorough covering of data variations. Regarding modeling, scientists have started looking at designs outside of the conventional CNN to pick spatio-temporal cues and global interactions. Some detection techniques, for example, use a temporal component—that is, sequence models such as recurrent neural networks or 3D convolutional networks—to examine not just individual frames but also how faces evolve across time in a video. Temporal artifacts—such as minor jitter or frame-to– frame inconsistencies—that a frame-by- frame CNN might overlook can occasionally be introduced by deepfakes and thus temporal modeling can help find those hints. Furthermore, transformer-based models have entered the deepfake detection scene as the Transformer architecture dominates many vision tasks. Vision Transformers (ViT) [7] and their variations learn long-range dependencies in an image using self-attention mechanisms instead of the locality bias of CNNs. In the context of a face image, a transformer might more readily correlate far-off elements of the face or scene to detect subtle inconsistencies – for example, linking the texture of the skin in one region with lighting reflections in another region – which could be quite effective in spotting fake composites. Early tests have found that on some benchmarks pure transformer models or hybrid models—that is, combinations of CNN feature extraction with transformer layers—can attain detection performance on par with or even surpassing CNNs. Transformer-based methods may also be less biassed toward low-level artifacts since they can learn relationships in a more global manner and may thus concentrate more on semantic anomalies (e.g., a face that seems somewhat off in expression or symmetry) than just pixel patterns. Transformers are computationally heavy and usually require very large training data, thus their application in deepfake detection is still under active research and not yet as common as CNN-based techniques. Beyond architecture, some works use data augmentation and adversarial training (creating purposefully perturbed fakes to train the detector to be invariant to certain changes), while others use multi-task learning or contrastive learning to force the model to learn more intrinsic features of a face. There are methods that aim to specifically separate the identity information from the "real vs fake" choice, for example by training the model in a way that it focuses on forensic artifact detection

while being agnostic of the person's identity. Deepfake detection is trending generally from a simple classification problem to a more complex, representation learning problem where the objective is to isolate the essence of what makes a fake a fake, in a general sense, rather than memorizing any one artifact or person.

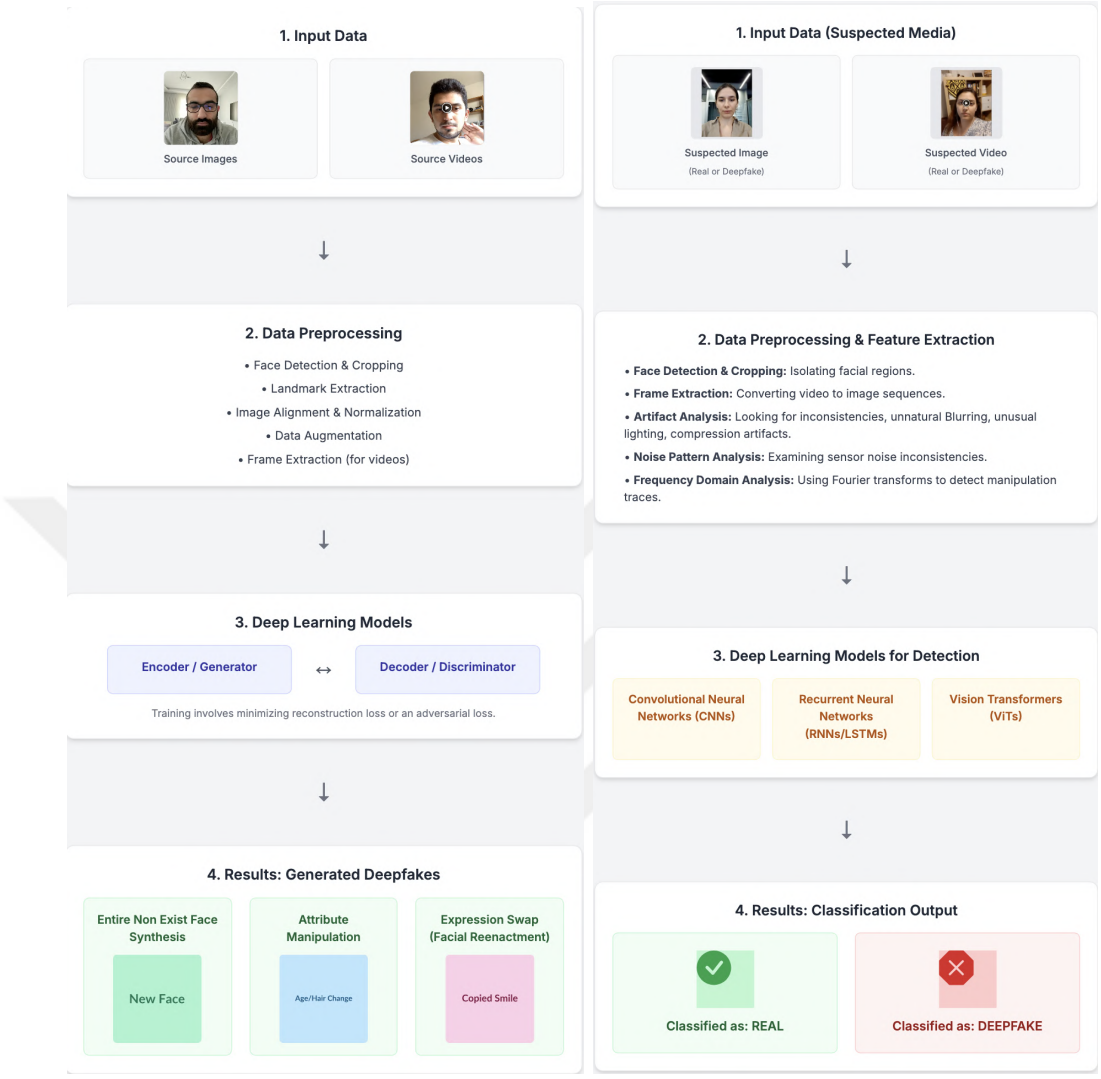
Despite these advances, generalization still presents the main difficulty. The community has responded by creating more varied and demanding benchmark datasets pushing detectors to be more robust. Even top algorithms can struggle when tested on truly unseen data, according to datasets including the large-scale ForgeryNet dataset and Celeb-DF, which comprises deepfake videos of many celebrities with improved visual quality over earlier datasets. On its test set, a detector trained on the well-known FaceForensics++, for instance, may show almost perfect accuracy; but, when evaluating on Celeb-DF videos, which were produced using a different deepfake generation technique and feature different people, accuracy can significantly decline. Cross-dataset performance is much lower than within-dataset performance in part because of variations in resolution, encoding, lighting conditions, and facial diversity. Practically speaking, this means that a deepfake detector may perform well in a controlled benchmark but fail to identify a fake that shows up in the wild – exactly the situation we fear for actual use. Thus, the next generation of deepfake detectors depends critically on addressing generalization and the related problem of identity leakage. Researchers are actively looking for answers, including training programs that replicate domain shift—that is, the model learns to adapt to fluctuations—or including known priors about images—that is, emphasizing frequency-domain artifacts that typically persist across several generating techniques. Using the great knowledge gained from face recognition systems seems to be one exciting route. Face-specific pretraining is the method whereby a model is first trained on a task connected to face analysis (but not fake detection) – for example, training on a large face recognition dataset to learn discriminative features of human faces – and subsequently that model is fine-tuned for deepfake detection. Based on intuition, a model that already recognizes facial structure and identity may be more adept than one that was only taught to do binary classification from scratch when something is off about a face. Starting from a face-savvy

feature space, the detector may need only a small adjustment to detect manipulation artifacts, and importantly, it may be less likely to latching onto irrelevant identity-specific cues. Basically, regardless of whose face it is or how the fake was created, such face-aware pretraining seeks to give the detector a general knowledge of how real faces look and vary so that it may more easily identify anomalies not typical of real faces. This method may help reduce implicit identity leakage since the model, having seen thousands of actual identities during pretraining, is less likely to mistake a given identity as essentially "fake." Rather, it can concentrate on differences that expose manipulation. Approaches like face-specific pretraining are at the forefront of research attempts to create strong, future-proof deepfake detectors since developing detection models that generalize across many fake scenarios and remain reliable as the fakes evolve is an ongoing battle.

As a matter of fact, both deepfake generation and detection techniques continue to evolve day by day competing against each other. Therefore, examining existing detection models and identifying areas for improvement remains an essential and active research direction. The general pipelines of both deepfake generation and deepfake detection pipelines can be seen in the Figure 1.1.

1.1. Contributions

In this thesis, we investigate various state-of-the-art deepfake detection techniques and we analyze their strengths and weaknesses in publicly available datasets. Some of these methods [8], [4], [9] rely on single-image-based approaches, while others utilize video-based methodologies [10]. Even within single-image-based models, video-level detection can be achieved by averaging the scores of individual frames. Most studies usually employ artificial neural network methods, such as convolutional neural networks and transfer learning [11]. Even though there are studies that experiment with datasets such as Celeb-DF-v2 [12] and FaceForensics++ [13], research on comprehensive datasets such as ForgeryNet remains limited. Additionally, there is a growing interest in understanding the limitations of these models under real-world



(a) (b)

Figure 1.1. Deepfake generation and deepfake detection pipelines.

scenarios, where compression artifacts and noise can degrade performance. Addressing these challenges is essential for deploying robust systems in high-stakes applications like misinformation prevention and identity verification.

One of the key contributions of this work is the demonstration that initializing deepfake detection models with weights pretrained on face-specific datasets, such as Glint360K [14], significantly improves performance compared to the widely used ImageNet-pretrained models. We compare several state-of-the-art deepfake detection methods, including Implicit Identity Leakage (IIL) [15], Localized Artifact Attention

Network (LAA-Net) [16], Multi-modal Multi-scale Transformer (M2TR) [17], Identity Consistency Transformer (ICT) [9], Learning Self-Consistency (LSC) [18], ResNet [6], and Vision Transformer (ViT) [7]. These models employ a diverse range of techniques, from CNN-based feature extraction to transformer architectures, attention mechanisms, and self-consistency learning, each targeting different aspects of deepfake artifacts. All existing deepfake detection models use ImageNet [19] initialized weights as their starting points. This situation limits models in their ability to learn domain-specific features extensively. By separating the state-of-the-art models into backbone and classification head components, and pretraining the backbone on Glint360K [14] enable models to achieve an improvement of 1.5% in performance. This improvement is attributed to the domain-specific knowledge encoded in face-pretrained models, enabling better detection of subtle facial inconsistencies and manipulations. Furthermore, in order to eliminate potential memorization of identities caused by pretraining face dataset, we incorporate label smoothing during training resulting in enhanced model generalizability, reduced overfitting, and improved robustness against unseen data distributions. Our findings suggest that combining domain-specific pretraining with advanced architectures improves robustness of deepfake detection across multiple datasets.

In addition to evaluating commonly used datasets such as FaceForensics++ [3], Celeb-DF-v2 [12], and ForgeryNet [20], we also introduce a new dataset designed to simulate real-world conditions. This dataset includes diverse lighting, backgrounds, and resolution settings, mitigating biases introduced by using deepfakes generated from a single source. It serves as a more realistic benchmark for evaluating the robustness and generalization capabilities of detection models. Our evaluations demonstrate that current state-of-the-art models struggle with generalization across manipulation techniques, further validating the need for improved pretraining and regularization strategies. The details and examples are provided in related sections.

The contributions of this thesis can be listed as follows:

(i) Proposing a domain-specific pretraining strategy: We suggest and test pretraining state-of-the-art models with highly diverse domain-specific face dataset resulting in 1.5% improvement in performance in average.
(ii) Introducing regularization techniques in deepfake detection to improve generalization and alleviate overfitting, especially after pretrain with face datasets.
(iii) Providing a novel dataset to better reflect real-world conditions in deepfake detection.

1.2. Organization of Thesis

The thesis is organized in the following way. In the following Chapter 2 we discuss and analyze the most advanced state-of-the-art deepfake detection models in detail with their advantages and disadvantages. After that, in Chapter 3, We explain our proposed method, contributions and models that we applied our method in detail. In Chapter 4, We present quantitative results of our experiments that we have performed, and discuss their results. Last but not least, we share our final comments and discuss the direction of future research in Chapter 5.

2. RELATED WORK

The term deepfake comes from the conjunction of deep-learning and fakes. As with the emergence of deep-learning models, particularly generation models, deepfake generation also became quite popular over the last decades. Understanding the details of deepfake generation and the inherent difficulties in their detection is significant to develop robust deepfake detection models. Especially after Ian J. Goodfellow *et al.* introduced Generative Adversarial Networks (GAN) [2], the quality of deepfakes increased rapidly. GANs mainly consist of two deep neural networks, a generator and a discriminator, working together in adversarial training process.The main objective of the generator is to create realistic synthetic media (e.g. deepfakes) while the focus of the discriminator is to distinguish if the media is fake or genuine. Through the training steps, the generator becomes more proficient creating synthetic data that are not distinguishable from real counterpart and the discriminator becomes better and better at detecting fakes. This competitive approach pushes the generator to be able to create more realistic images or videos.

The deepfake generation technology is not composed of one single approach but of accumulation of different facial manipulation methods. These manipulation techniques:

- Entire Face Synthesis: This approach involves generation completely new, non-existing, faces. GANS are mostly utilized to generate such deepfakes.
- Identity Swap (Face Swapping): This approach replaces the face of a person in one image or video with the face of a person in another video or image. Mostly, Autoencoders(AE) are utilized for face swap. This category is mostly used category while generating deepfakes.
- Attribute Manipulation: This category involves changing specific facial attributes of a person. (e.g. eye color, age)
- Expression Swap (Facial Reenactment): In facial reenactment, the purpose is to transfer one's facial expressions such as smile, eye closed or open etc. to a

target individual so that enabling someone to be shown making expressions that actually they do not. This category is second mostly used category in deepfake generation field, especially if it is intended to use maliciously.

The developing of effective and robust deepfake detection system is significant especially to prevent malicious use of them. However, there are several challenges to create such deepfake detection system. These challenges stem primarily from the complexities of real-world data and the limitations of current detection models. The challenges can be listed as follows:

- Generalization to Unseen Manipulations: The foremost and most common problem is generalizability of the detection model. Since the generation models continuously developing with new algorithms and techniques, detection models trained with known methods generally fall behind.
- Robustness to Real-World Conditions: The detection models are generally tested using clean, high quality datasets. However, in the real world the media may undergo several steps such as compression, noise due to lens quality, blurring, and lightning conditions, resulting in a degradation in detection accuracy.
- Adversarial Attacks: Like other deep learning models, deepfake detectors are vulnerable to adversarial attacks. While these changes might be imperceptible to human eye, it may distract the model to decide if the media is forgery or genuine.
- Data Availability and Quality: As all deep learning models, deepfake detection models are highly dependent on their training data. The gap between academically available datasets and real-world deepfakes is so large that it can lead to overestimation of the detection method and may hinder its real ability.

In this thesis, previous work will be mostly analyzed using these challenges. To make recent work analysis more structured, we have grouped detection methods into 3 parts: image-based, video-based, and hybrid approaches. Also, even though it is beyond the scope of this thesis, we briefly mentioned the detection methods that use voice as a feature.

2.1. Image-Based Approaches

Image-based deepfake detectors focus on analyzing single images to decide if the given image is a deepfake image or genuine image. In the context of video input, these kind of detectors can also be used by averaging scores of all frames, treating each frame as independent image. Although too early approaches mostly focus on handcrafted features with machine learning, advanced approaches usually utilize deep learning techniques and models.

Many state-of-the-art methods use convolutional neural neworks (CNN) to capture subtle faulty areas in textures that disguish real and fake images in single frames. For instance, the FaceForensics++ dataset introduced a large dataset composed of manipulated face images and genuine faces and showed that the Xception architecture, proposed by Chollet [4], can achieve high accuracy on known manipulation methods, despite requiring fewer parameters and computational power compared to other convolutional neural networks. Moreover, XceptionNet and similar architectures such as ResNet [6] and EfficientNet [5] have also been extensively used in deepfake detection tasks, became popular backbones in several architectures in deepfake detection domain. These models, often initialized with ImageNet weights, learn to extract high-level features from images and were very effective on within-dataset evaluation [11]. However, their focus on global face features can be a drawback: studies found that vanilla CNN detectors tend to ignore subtle local artifacts and may inadvertently overfit onto spurious cues such as the identity or source dataset statistics, which hurts generalization to new fakes. To mitigate overfitting, researchers have adopted regular training tricks (e.g. data augmentation, dropout) and large-scale data.

Since high-quality deepfakes only introduce very subtle pixel-level anomalies, recent work has focused on making CNN models more sensitive to these local artifacts. Nguyen *et al.* propose the Localized Artifact Attention Network (LAA-Net) [16] to explicitly attend to small, vulnerable regions of the face that are prone to contain manipulation artifacts. LAA-Net introduces a multi-task learning framework that gener-

ates an artifact heatmap (via a self-consistency attention branch) alongside the binary genuine/fake prediction. This approach forces the model to concentrate on forensic discrepancies rather than overall appearance. Additionally, LAA-Net incorporates an enhanced feature pyramid network to propagate fine-grained texture features to the final layers. Pyramid network approach addresses the deficiency that standard CNN backbones like Xception or EfficientNet overly emphasize global structure and miss subtle glitches. By focusing on these localized cues, LAA-Net achieves state-of-the-art performance on detecting high-quality deepfakes that traditional CNNs fail to detect by looking on global cues. Similarly, Li *et al.* introduced Face X-Ray [21], which takes a multi-task approach by learning to predict the blending mask of a spliced face in addition to classifying, thereby directly highlighting the manipulated region. The Face X-Ray method assumes that a swapped face image is a composite of a new face pasted into a background and thus tries to detect the boundary between the two; this approach proved effective at catching face swaps from unseen generative models by focusing on the forensic seams (i.e. blending artifacts) rather than the identity or content of the face. Overall, such attention and localization mechanisms represent a trend toward explainable deepfake detection, where the model not only outputs a result, but also indicates where the image is suspicious.

Another branch of image-based forensics leverages the fact that GAN-generated images often exhibit anomalies in the frequency spectrum that are imperceptible in the RGB domain. A number of methods explicitly analyze frequency representations of images to detect deepfakes. A number of methods explicitly analyze frequency representations of images to detect deepfakes. For example, Luo *et al.* found that CNN-generated faces could be spotted via DCT (Discrete Cosine Transform) coefficient aberrations [22]. They designed detectors to catch these frequency artifacts, such as unnatural peaks or regular patterns left by upsampling operations in GANs. In a similar approach, Gu *et al.* proposed to use wavelet transforms to improve the deepfake detection. In the work, a wavelet packet decomposition is used to capture high-frequency signals. This approach has showed improvement in the detection of subtle forgery cues. Similarly, Jeong *et al.* introduced FrePGAN [23], which augments

training images with targeted frequency perturbations to make a model robust to shifts in frequency domain. The general finding is that high-frequency descriptors – from Fourier spectra to wavelet coefficients – can provide a complementary signal to the CNN, since generative models often struggle to reproduce the natural spectral statistics of real images [24]. Some detectors therefore operate with dual inputs: the RGB image and a frequency-transformed version (e.g. spectral or gradient images), sometimes fusing them late in the network. By learning from both domains, these models can generalize better to compression or new manipulations. Indeed, Giudice *et al.* [25] showed that even when a fake image looks remarkably realistic, its frequency fingerprints (e.g. periodic artifacts in the DCT domain) may give it away. This insight has driven many subsequent approaches to incorporate frequency-aware modules or loss functions into a standard image classifier.

As the field progresses, vision transformers (ViT) [7] have also been applied to deepfake image detection. ViTs can detect long-range dependencies and track global context. This ability of ViTs is useful for detecting inconsistencies spread across an image. Zhuang *et al.* presented UIA-ViT [26], an Unsupervised Inconsistency-Aware Vision Transformer, which learns to detect forgeries without any labeled fake images. UIA-ViT is trained only on real images, optimizing it to find internal inconsistencies or anomalies (e.g. in texture or relational patterns) that might indicate a manipulated image. Such self-supervised strategies address the scarcity of labeled deepfake images. Furthermore, the model essentially learns a representation of real facial images and flags deviations from this normal as potential deepfakes so that it does not stick to its training set and aim to improve generalization. In a related direction, some researchers have explored contrastive learning and other representation learning techniques to train deepfake detectors. For instance, a work from Shiohara *et al.* [27] generates fake images by blending faces with themselves (so-called Self-Blended Images) as a form of data augmentation and then trains the detector on these synthetic data. The goal is to force the network to focus on subtle artifacts, since the identity remains the same in self-blended fakes, identity cues are useless. On the other hand, Dong *et al.* [9] also utilized ViTs using a similar approach as in self-blended images. In the training phase,

the authors mixed the real faces by keeping one's outer face and while keeping other's inner face aiming to train model to learn the discrepancies between inner and outer face. As a result, Dong *et al.* [9] achieved superior generalization ability than compared models. These transformers and self-supervised methods show promising cross-dataset performance because they do not overfit to particular fake patterns; instead, they learn more universal features of authenticity and inconsistency.

As we have described previously, one of the most challenges in deepfake detection is the generalization gap, models often perform well on known deepfake types but fail on unseen ones. One of the factors that is causing generalizability of the models to fall back is that binary classifiers inadvertently learn to distinguish the identities in the training set: a fake image often retains some features of the source identity, which creates a subtle bias separating "fake" identity embeddings from real ones. In other words, instead of purely learning forgery artifacts, the model may partially just recognize that certain faces (source identities used to create fakes) look inconsistent or unfamiliar, leading to false confidence that fails when new identities appear. To adress this issue, Dong *et al.* [15] proposed an ID-unaware detection model that explicitly suppresses identity information. Their approach localizes artifact-rich regions and only uses those regions for the classification decision, thus ignoring the person's identity. This yielded a significant boost in cross-dataset generalization, outperforming prior state-of-the-art on challenging unseen-fake evaluations. Other works have also aimed at improving generalization: for example, some introduce latent space augmentations to simulate novel fake styles during training [28], while others leverage multiple generation methods and even synthetic "fake-fakes" to learn features that generalize beyond any one forgery method [22].

To sum up, image-based deepfake detectors have evolved from simple CNN classifiers to sophisticated architectures that integrate spatial and frequency cues, attention to artifacts, and even self-supervised anomaly detection – all in pursuit of catching ever more realistic fakes and ensuring the learned forensic features remain effective on new manipulation techniques.

2.2. Video-Based Approaches

Image-based detectors can be applied frame-by-frame to videos, but purely frame-level analysis fails to exploit the temporal cues available in videos. Therefore, a branch of deepfake detection focuses on video-based approaches, which analyze sequences of frames to find inconsistencies over time.

Many early works looked for obvious physiological or behavioral signs in video. In 2018, Yang *et al.* [29] targeted specific human signals that generative models initially overlooked. Their focus is to detect inconsistent head poses across frames (noting that a face-swapped head often had pose angles that did not realistically match the orientation of the person's torso or the scene geometry). For this purpose, the authors first mapped 2D face images to 3D face poses and after that train an Support Vector Machine (SVM) classifier. Furthermore, another approach by Li *et al.* [30] used long-term recurrent convolutional networks to flag videos where the subject hardly ever blinks, since some early deepfake generators failed to model natural eye blinking. Such methods proved that temporal behaviors can give away of a deepfake. However, as deepfake generation models improved, these straightforward cues were addressed by generators. This led researchers to design learning-based models that capture more subtle temporal artifacts. One straightforward approach is to extend CNNs to the time dimension: for instance, by feeding a sequence of frame embeddings into a recurrent neural network (RNN) or a 3D-CNN. Sabir *et al.* [31] used a CNN+LSTM pipeline where each frame's features (extracted by ResNet [6]) are passed to an LSTM that learns the evolution of features over time, enabling detection of temporal glitches such as flickering or mismatched facial movements. Agarwal *et al.* [32] combined appearance and behavior by using face recognition features (to represent who is in the video) alongside an analysis of facial movements to decide if a video is real. In their work, they essentially fused static face features with dynamic cues like head movement and expression changes, demonstrating improved accuracy, especially on face-swap videos where identity and motion can diverge from the expected pattern.

Deepfake detectors may also utilize spatiotemporal architectures. For example, several detection methods employs 3D CNNs or inflated CNNs where filters span spatial and temporal dimensions, to directly learn video features. One example can be Zheng *et al.* [10] introduced a temporal dropout strategy in a 3D CNN: randomly dropping entire frames or temporal convolution slices during training forces the model to rely on robust, frame-invariant cues, making it less sensitive to any single frame anomaly and better at overall sequence consistency. As the improvements of computing powers transformers has become another powerful models that is using in deepfake detection. Khan *et al.* [33] proposed a Video Transformer that processes frame sequences with self-attention, and even allowed incremental learning to adapt the model as new deepfake types emerge. Transformers can capture long-range temporal relationships – for instance, subtle mismatches in how a person's facial expressions progress or how mouth movements align. Gu *et al.* [34] took this further with a spatiotemporal dropout transformer, combining the ideas: they used a ViT-style architecture on video frames and randomly masked patches across space and time to force the model to attend to distributed inconsistencies rather than any one region or frame. They excel at catching clues like temporally flickering textures, inconsistencies in lighting or shadows across frames, or unrealistic motion between frames.

A number of recent works explicitly formulate deepfake detection as finding temporal inconsistencies, often using contrastive or relational learning. For instance, Gu *et al.* [35] developed a Region-Aware Temporal Inconsistency learning method. Their model extracts features from corresponding face regions in consecutive frames and learns to detect if those region features change in implausible ways over time (for a real video, the appearance of, say, the jaw or eyes should vary smoothly according to expression and pose, whereas a fake might introduce sudden unnatural changes or discrepancies). By using a contrastive loss that treats real video frame sequences as consistent and fake sequences as inconsistent, the model becomes sensitive to even minor temporal oddities. Another work by Xu *et al.* [36] employed hierarchical contrastive learning on video clips to capture inconsistency at multiple time scales, short-term inconsistencies (between adjacent frames) and long-term ones (across a span of frames)

are both relevant. On the other hand, Hu *et al.* [37] have looked at generating synthetic perturbations in time. They proposed FInfer (Frame Inference), which inserts an interpolation task into training. The model is asked not only to classify real/fake, but also to predict intermediate frames between given frames; since a deepfake may not obey true motion physics, the frame interpolation error tends to be higher for fakes, helping the network learn markers of forgery. The common ground of the aforementioned methods is to leverage the temporal coherence of real videos. The deficiency of deepfake videos is that they often show breaks in continuity such as sudden jumps, inconsistent facial geometry, and temporal artifacts while the genuine videos have a natural continuity.

An intriguing subset of video approaches focuses on the consistency of a person's identity and behavior throughout a clip. If a deepfake involves swapping one person's face onto another's body or altering a person's expressions, there may be inconsistencies in who the person appears to be or how they behave. Identity consistency can be checked by tracking an identity embedding over time: if the embedding of the face suddenly changes mid-video, that's a red flag. TI2Net [38] is a Temporal Identity Inconsistency Network that learns to detect precisely such anomalies. Essentially, it verifies that the same identity is present in all frames of a video. TI2Net would catch, for example, a face swap where the imposter face does not perfectly maintain the original identity from every angle. Another notable work is ID-Reveal by Cozzolino *et al.* [39], which doesn't use any fake videos in training at all. Instead, ID-Reveal learns the unique temporal patterns of how a person's face moves when talking by training on only real videos (using a metric learning paradigm). Each individual has subtle mannerisms and dynamics; ID-Reveal models these so that it can later spot a fake where those learned "identity motion signatures" are violated. This approach yielded improvement in accuracy on unseen manipulations (like facial reenactment), highlighting that who is acting in the video (and how) can be as important as pixel artifacts. Similarly, works by Chu *et al.* [40] on world leaders have emphasized behavioral biometrics – for example, subtle gestural and speaking patterns distinctive to a politician. If a deepfake video of a world leader does not exhibit that person's characteristic hand movements

or speech cadence, it can be identified as fake. These high-level semantic approaches complement the pixel-level artifact detectors, and combining them can yield robust systems.

In summary, video-based deepfake detection leverages the spatiotemporal data to catch forgeries. These methods can detect temporal artifacts such as flicker, jitter, discontinuities, behavioral anomalies unnatural expressions or identity differences. This often makes them more powerful against realistic deepfakes than single-frame analysis. However, they also face the challenge of needing large amounts of training video data and dealing with compression/frame-rate issues. As with image-based detectors, generalization remains critical: many video methods now incorporate both low-level clues and high-level semantics to ensure that even if a new deepfake generation technique fixes one type of artifact, the model can fall back on other signals. Recent benchmarks indicate that ensembles which combine multiple temporal and spatial models (e.g. an image CNN + a temporal model) tend to achieve the best robustness, underscoring the value of a multi-cue approach to video deepfake detection.

2.3. Hybrid Approaches

The lines between image-based and video-based detection are often blurred. Many practical systems adopt a hybrid approach that combines the strengths of both spatial analysis (per-frame) and temporal analysis (across frames), or that fuse multiple features and modalities to improve accuracy. In this category, we discuss methods that integrate different techniques or data sources.

A straightforward hybrid strategy is to run an image-level detector on each frame of a video and then aggregate the results over time. This could be as simple as averaging the frame-level predictions or as complex as feeding frame features into a sequence model as described earlier. For example, the winning solutions to the DFDC competition essentially ensembled multiple frame-based deep CNNs and then applied a temporal smoothing to the predictions. In academic work, Qian *et al.* [41] proposed

a two-branch recurrent network that explicitly splits the problem. In their work, one branch analyzes each frame's spatial content while the other branch learns temporal features from the sequence and after that they merge features in a mix block. By jointly training these branches, the model isolates temporal inconsistencies without losing spatial precision. Similarly, Chen *et al.* [42] introduced a dual-stream architecture to detect face swaps using both spatial and temporal clues. Such hybrid designs have shown superior performance to either approach alone, especially on videos that are borderline (e.g. where each frame by itself may not have a clear artifact, but the temporal dynamics give it away).

Beyond temporal cues, hybrid methods also often combine heterogeneous features from the same image. One prominent example is the M2TR framework proposed by Wang *et al* [17], which stands for Multi-modal Multi-scale Transformer. Multi-modal refers to two modalities of the visual data: the RGB image and its frequency spectrum. M2TR operates on image patches at multiple scales (capturing both fine-grained and coarse image details) and simultaneously feeds the corresponding Fourier features of those patches into a transformer-based fusion module. By designing a cross-modality attention block, M2TR learns to detect local inconsistencies in the pixel domain and complements them with frequency-domain artifacts. This hybrid of spatial + frequency analysis lets it pick up subtle clues that might be missed if only one domain were considered. For instance, a slight texture distorting from a GAN's upsampling which is visible in frequency domain or a distorted specular highlight which is visible in RGB domain. The authors also created a new dataset of high-quality deepfakes to test generalization, and M2TR outperformed prior CNN methods by a clear margin. Similarly, Guillaro *et al.* proposed TruFor [43] explicitly fuses high-level and low-level clues for image forgery detection. TruFor uses a transformer to combine the RGB image with a learned noise-based fingerprint of the image called noiseprint. The noiseprint is essentially a model of the camera sensor noise and compression artifacts that should be present in an authentic image; deviations in this noise pattern can indicate tampering. By training on real images only, TruFor learns a representation of expected camera noise, and during inference it detects forgeries as anomalies where the RGB content

and the noise fingerprint disagree.

Some hybrid approaches go further to incorporate a wide range of cues. Lomnitz *et al.* [44] presented a multimodal deep learning system that ingests spatial, spectral, and temporal inputs simultaneously. In their network, an image goes through parallel pathways. One extracts spatial features, another extracts frequency-domain features, and a third analyzes temporal sequences. The fused representation from all three streams is used to classify the video as real or fake. This holistic approach showed improved robustness, since even if a deepfake method conceals one type of artifact, it might still betray itself in another. Another example is the integration of forensic segmentation into classification. We discussed Face X-Ray [21] above, which is a hybrid of segmentation and detection. Many recent works follow this multi-task learning idea. For instance, one could train a network to both classify and localize the forgery, or to both detect and attribute the fake to a source model. Multi-task training can act as a regularizer and force the model to learn more general features of anomalies. As evidence of effectiveness, Dang *et al.* proposed an Identity-Driven Deepfake Detection [45] method that used a face recognition network to guide the detection model, essentially combining an identity verification task with the deepfake detection task. By ensuring the model pays attention to who the person should be, it became harder for a fake to fool the system unless it got both the identity and the pixels correct. However, a potential pitfall of identity-assisted methods is again the risk of poor generalization if a fake introduces a completely new identity or if the method is turned against unknown subjects.

2.4. Audio, Biological Signals, and Beyond

While the thesis primarily focuses on vision-based techniques, it is important to acknowledge deepfake detection efforts by using other modalities and multi-modal settings. Deepfakes are not limited to imagery; there are also manipulations that involve audio-visual inconsistencies. In addition, researchers have used biological signals as a modality for deep-fake forensics. We summarize these complementary approaches

below.

There are methods focusing on specific cues. For instance, checking the phoneme-prosody alignment. In natural speech, the timing of phonemes and the visual lip movements are tightly coordinated. Deep-fake lip sync detectors exploit this by extracting the phonetic sequence of the audio and the lip motion sequence of the video and measuring their synchronization. Recent work even uses pretrained speech recognition or speaker recognition models to verify if the speaker's voice matches the face. For example, Cheng *et al.* [46] proposed to measure voice-face homogeneity. A deepfake impersonation might have a mismatch (e.g. actor A's face with actor B's voice). By using state-of-the-art voice and face recognition features, they could detect such impostor videos with high accuracy. Audio-visual consistency approaches often straddle the line between modalities: they require analyzing both the sound and the visuals together. In academic categorization, they are multi-modal approaches, but they underscore how detecting deepfakes can sometimes go beyond just pixels or just sound. Their aim is to catch the inconsistency between modalities that should normally agree.

One very creative modality for deepfake detection is the use of involuntary biological signals that are embedded in videos. The human body has subtle rhythms – notably, the cardiac pulse – that can manifest in videos as minute color changes or motions. Genuine videos of people contain these signals, whereas current deepfake generation processes often do not faithfully reproduce them. FakeCatcher [47] is a pioneering system that leverages this fact. It detects fake portrait videos by extracting the subject's rPPG (remote photoplethysmography) signal. Essentially, the slight cyclical variation in skin color due to blood flow from the face region is extracted. In a real video, the rPPG signals from different parts of the face will be consistent and correspond to a plausible heart rate. In a fake video, those signals are either absent, noisy, or inconsistent across the face. Ciftci *et al.* showed that using these physiological features, they could achieve very high detection rates, even for high-quality deepfakes, because it's extremely difficult for generative models to simulate a realistic heart beat signal inadvertently.

In conclusion, while vision-based deepfake detectors remain the core focus (given that most deepfakes involve visual manipulation), the research community has actively explored audio and other modalities as both standalone detectors and as part of multi-modal systems.

Table 2.1. Summary of deepfake detection related work. I, V, A stands for image, video, audio respectively.

Method	**Paper (Year)**	**Type**	**Main Detection Strategy**	**Backbone**
XceptionNet	Rössler et al. (2019) [3]	I	CNN-based artifact detection	Xception
LAA-Net	Nguyen et al. (2024) [16]	I	Localized artifact attention (heatmap)	CNN+FPN
Face X-Ray	Li et al. (2020) [21]	I	Blending artifact segmentation	CNN
FrePGAN	Jeong et al. (2022) [23]	I	Frequency perturbation augmentation	CNN
UIA-ViT	Zhuang et al. (2022) [26]	I	Unsupervised inconsistency localization	ViT
Self-Blended	Shiohara et al. (2022) [27]	I	Self-blended identity-agnostic training	CNN
StyleFlow	Choi et al. (2024) [28]	I	Latent style generalization improvement	CNN
M2TR	Wang et al. (2022) [17]	I	Multi-modal multi-scale (RGB + freq.)	CNN + Transformer
TruFor	Guillaro et al. (2023) [43]	I	Multi-level anomaly fusion	CNN
Head Pose	Yang et al. (2018) [29]	V	3D head pose consistency	SVM
Eye Blink	Li et al. (2018) [30]	V	Blink rate anomaly detection	CNN+RNN
CNN-LSTM	Sabir et al. (2019) [31]	V	Frame-wise CNN with temporal modeling	ResNet-50 + LSTM
Appearance & Behavior	Agarwal et al. (2020) [32]	V	Identity + behavioral analysis	CNN + temporal
Temporal Dropout 3D	Zheng et al. (2021) [10]	V	Temporal dropout in 3D-CNN	3D CNN (ResNet)
Video Transformer	Khan & Dai (2021) [33]	V	Transformer on frame sequences	Transformer
Spatio-Temporal Dropout	Gu et al. (2022) [34]	V	Intra and inter similarity	CNN
TALL	Xu et al. (2024) [36]	V	Thumbnail layout video analysis	CNN
FInfer	Hu et al. (2022) [37]	V	Frame interpolation consistency check	CNN
TI2Net	Liu et al. (2023) [38]	V	Temporal identity inconsistency	CNN+RNN
ID-Reveal	Cozzolino et al. (2021) [39]	V	Identity mismatch detection	Face encoder + RNN
Speaker Pattern	Chu et al. (2022) [40]	V	Facial behavioral biometrics	Facial landmarks
Frequency-aware	Qian et al. (2020) [41]	V	Frequency-domain analysis	CNN
FSSPOTTER	Chen et al. (2020) [42]	V	Spatial-temporal forgery clues	CNN+RNN
Multimodal	Lomnitz et al. (2020) [44]	AV	Audio-visual feature fusion	CNN (Audio+Video)
Identity-Driven	Dong et al. (2020) [45]	V	Identity embedding integration	ArcFace + CNN
Voice-Face	Cheng et al. (2022) [46]	AV	Voice-face cross-modal consistency	CNN (Speaker+Face)
FakeCatcher	Ciftci et al. (2020) [47]	V	Physiological signal analysis	rPPG analysis

2.5. Deepfake Detection Datasets

Many datasets have been presented to support deepfake detector evaluation and training. These databases differ in scale, content (images against videos), kinds of

manipulation, source media, and availability. Together with their main features, we have compiled a thorough list of extensively used deepfake detection datasets below, both image-based and video-based collections.

UADFV [29], is an early deepfake video dataset. It contains 98 short videos (49 real and 49 fake) where the fake clips were created by swapping faces in the real YouTube videos using the popular FakeApp deepfake tool. This dataset was originally used to expose artifacts (like irregular eye blinking) in deepfakes.

DeepFake-TIMIT [48] is A small benchmark based on the VidTIMIT face-video corpus. It consists of 640 manipulated videos (32 subjects, each with 10 fakes at low quality and 10 at high quality) created using an open-source GAN faceswap method. The real source videos (from VidTIMIT, 320 clips) are of people speaking controlled phrases.

FaceForensics++ [3] is One of the most widely adopted deepfake video datasets. Rössler et al. collected 1000 real videos from YouTube and applied four different face manipulation methods to each, yielding 4000 fakes (for a total of 5000 videos). The forgery methods include two deepfake face-swapping approaches ("DeepFakes" and FaceSwap) and two facial reenactment techniques (Face2Face and NeuralTextures). The videos are provided at multiple compression levels (raw and heavily compressed) to simulate real-world conditions.

Celeb-DF-v2 [12] is a large-scale and challenging deepfake video dataset introduced by Li et al.. It contains 590 real videos collected from interviews of 59 celebrities, and 5,639 corresponding deepfake videos generated by swapping each pair of celebrity faces using an improved synthesis method. The fakes in Celeb-DF v2 are high-quality, lacking many artifacts present in earlier datasets, thus better reflecting deepfakes "in the wild".

Deepfake Detection Challenge (DFDC) [49] is one of the largest deepfake video datasets. The final DFDC released for the global competition comprises 128,154 videos (23,654 real and 104,500 fake). These were created by filming thousands of paid actors (in diverse scenes) and then applying eight different face manipulation techniques to generate forgeries. The methods included two variants of the classic deepfake autoencoder (DF-128, DF-256), GAN-based face swaps (FSGAN), neural talking-head models (Neural Talking Heads, NTH), puppet-master approaches (MM/NN) and even head-swapping with StyleGAN, plus some with adversarial refinements and added audio swaps for lip-sync fakes. In DFDC, a video is labeled fake if either the video or audio has been manipulated. Note that, only sample set of dataset can be found online.

DeeperForensics [50] is a large-scale dataset designed to capture real-world face forgery scenarios. It contains 60,000 videos (17.6 million frames) of high-quality manipulated faces. The creators recorded 100 paid actors in a controlled studio environment (1080p footage with varied poses, lighting, and expressions) to obtain a wide range of real faces. Then, a novel end-to-end face-swapping model (DF-VAE) was used to generate 1,000 initial deepfake videos by swapping each identity onto several others. To expand diversity, extensive augmentations (35 types of perturbations such as compression artifacts, blurring, noise, etc.) were applied to these, yielding the final 60k clips. DeeperForensics emphasizes realism, scale, and diversity for robust detector training.

WildDeepfake [51] is a "in-the-wild" test dataset to evaluate deepfake detectors on real online content. It consists of 707 deepfake videos found on the internet (e.g. downloaded from websites and social media) and the corresponding real footage where available. From these, the authors extracted 7,314 face sequences (face-cropped video clips) – about 3,805 from the real videos and 3,509 from the fake videos. The deepfakes in WildDeepfake were created by unknown methods "in the wild" (often by amateur creators), making this dataset especially challenging due to diverse quality, compression, and post-processing seen in real-world fake videos.

FakeAVCeleb [52] is a multimodal audio-visual deepfake dataset by Khalid et al. aimed at studying both face and voice forgeries. It starts with 500 real videos of celebrities from the VoxCeleb2 dataset (interview footage). Using these, approximately 20,000 deepfake videos were generated. Uniquely, FakeAVCeleb covers combinations of real/fake audio and video: faces were swapped in video using two methods (FaceSwap and FSGAN) and voices were cloned using AI voice cloning (RTVC) with lip-sync alignment via Wav2Lip. The dataset is partitioned into four sets: Real Video–Real Audio, Fake Video–Real Audio, Real Video–Fake Audio, and Fake Video–Fake Audio. This allows researchers to test detectors on scenarios like dubbed voices, swapped faces, or both.

ForgeryNet [20] is the largest deepfake benchmark to date. ForgeryNet includes an immense 2.9 million manipulated images and 221,247 manipulated videos, each with accompanying real counterparts. It encompasses diverse manipulation techniques – 7 types of image-level forgeries and 8 types of video-level forgeries – making it extremely comprehensive. For example, image forgeries range from entire face synthesis (GAN-generated faces) to attribute edits (like changing expressions or facial attributes), whereas video forgeries include identity swaps, reenactment of expressions, lip-sync dubbing, and more. The source data are drawn from large repositories (e.g. millions of real face images and thousands of real videos from VoxCeleb2, AVSpeech, etc.), ensuring a wide variety of identities and conditions. Furthermore, ForgeryNet provides rich annotations (ground truth masks for manipulated regions, labels of technique type, and even temporal segmentation of forged segments) to enable detailed analysis. It serves as a one-stop benchmark for evaluating detection across many forgery types. Note that for our experiments we have used image set of ForgeryNet.

Since three of FaceForensics++, Celeb-DF v2, and ForgeryNet have become some of the most often used and complete benchmarks in the deepfake detection field, we decide to evaluate on these three in our thesis experiments. A standard benchmark, FaceForensics++ (FF+) is often used in many studies because of its balanced size and several manipulation techniques. Training baseline detectors would benefit from

Table 2.2. Comparison of major deepfake detection datasets. Each dataset is characterized by its year, scale (number of videos/images), source of original media, data modality. V,A,I stands for video, audio, image respectively.

Dataset	**Year**	**Size (samples)**	**Source Media**	**Modality**
UADFV	2018	98 videos (49 real, 49 fake)	YouTube videos (faces)	V
DeepFake-TIMIT	2018	640 fake videos (LQ/HQ)	VidTIMIT video dataset	V
FaceForensics++	2019	5,000 videos (1000 real, 4000 fake)	YouTube videos (frontal)	V
Celeb-DF (v2)	2020	6,229 videos (590 real, 5,639 fake)	YouTube celeb interviews	V
DFDC	2020	∼128,154 videos (23,654 real, 104,500 fake)	Paid actors (crowd-sourced)	V
DeeperForensics-1.0	2020	60,000 videos (17.6M frames)	Studio-shot videos (100 actors)	V
WildDeepfake	2020	7,314 face clips (from 707 videos)	Internet videos ("in the wild")	V
FakeAVCeleb	2021	∼20,000 fake videos (+500 real)	VoxCeleb2 videos (celebs)	V & A
ForgeryNet	2021	2.9M images; 221,247 videos	VoxCeleb2, AVSpeech	I & V

high-quality fakes and controlled compression levels offered here. Conversely, Celeb-DF v2 is renowned for its difficult, high-realistic deepfakes that more faithfully replicate those seen online. Since the fakes avoid clear artifacts in past sets, detectors tested on Celeb-DF v2 typically generalize better. As the last dataset, ForgeryNet is the next generation of benchmarks. It is by far the biggest public deepfake dataset with millions of samples using wide range of manipulation strategies. We tested our models against an unparalleled range of forgeries and image conditions by using ForgeryNet's test set. We guarantee our experimental evaluation spans both breadth (ForgeryNet's complete coverage) and depth (FF++ and Celeb-DF's extensively-studied, high-quality fakes) by using these three datasets together. All things considered, they are reputable, extensively used in the research community, and complement each other in their emphasis – hence, they are a great choice for a comprehensive review of deepfake detection techniques.

3. METHODOLOGY

This chapter details the methodology for evaluating and improving deepfake detection models. We compare several state-of-the-art deepfake detectors – including methods addressing Implicit Identity Leakage (IIL [15]), the Localized Artifact Attention Network (LAA-Net) [16], the Multi-modal Multi-scale Transformer (M2TR) [17], and the Learning Self-Consistency (LSC) [18] approach – alongside baseline architectures such as ResNet [6] and Vision Transformer (ViT) [7]. The goal is to assess these diverse models under a unified training and evaluation framework, and to explore the benefits of a face-specific pretraining strategy. In particular, we propose to pretrain each model's backbone on a large face dataset (Glint360K) instead of the generic ImageNet, and then fine-tune on deepfake datasets. By doing so, we aim to endow the models with facial domain knowledge from the outset, hypothesizing that this will improve their ability to discern the subtle manipulations in forged faces.

Training with large face dataset instead of the generic ImageNet might cause issues related to generalization and overfitting. To further resolve that issue, we utilize label smoothing during training which is a regularization technique that softens the target labels. As a result, it may help preventing the network from becoming overconfident in its predictions. In our work, label smoothing is introduced during face pretraining to mitigate the risk of the model memorizing specific identities – a phenomenon related to implicit identity leakage where the detector latches onto who the person is rather than what artifacts indicate a fake. By smoothing identity labels, the backbone is encouraged to learn more general facial features instead of overly specific identity signatures.

Overall, this thesis is designed to answer two key questions:

(i) How do different modern deepfake detection models perform relative to each other under consistent conditions, especially in a dataset that reflect real world

conditions?

(ii) Does face-specific pretraining with appropriate regularization improve detection performance and generalization compared to conventional ImageNet pretraining?

In the following sections, we start with mentioning details of our new deepfake dataset. After that, we describe each model and our training approach in detail, including technical adaptations made for our experiments. We also outline the implementation details for pretraining on Glint360K – a large-scale face recognition dataset with over 17 million images of 360K individuals – and for fine-tuning on deepfake benchmarks (e.g., FaceForensics++, Celeb-DF). Finally, we discuss how we qualitatively and quantitatively compare the models across publicly available datasets and our new custom dataset, highlighting any observed patterns (such as certain models excelling at detecting specific artifact types or generalizing better to unseen forgeries). The outcome of this methodology will set the stage for the results and analysis in the next chapter.

3.1. Our Dataset

In order to better simulate real-world conditions and assess model robustness, we constructed a custom deepfake dataset reflecting varied lighting, poses, compression levels, backgrounds, and identity consistency. Currently available academic benchmarks often limited to controlled settings. For instance, some are captured using same media device in the same background or some are downloaded via platforms such as Youtube. Unlike these, our dataset emphasizes usage scenarios where both original and forged content is coming from completely different configurations from diverse set of identities from different ethnicities, from different ages.

We have collected 4252 real videos at least 60 seconds, representing diverse demographics (age, gender, ethnicity). Videos span indoor interviews, outdoor environments, and user-generated content, capturing multiple poses, lighting styles, facial accessories, and moderate motion. As the current mobile phones usually 30 frames per

second for video, our videos are usually contains 30 frames per second too but some may differ since the videos coming from different sources.

Faces were detected and aligned using MTCNN [53]. Five face-cropping margins were generated ranging from Level 1 to Level 5. As Level 1 contains only facial area without peripheral features, Level 5 has very wide cropping, heavily dominated by background and significantly reducing the size of the face in the frame.

For the generation of deepfakes, we have utilized recent and publicly available state-of-the-art face-swap and face-reenactment tools. More specifically, we have leveraged DeepFaceLab v2.0 [54] which are trained for each source-target pair to ensure high realism, First-Order Motion Model [55] for realistic image animations. As post-processing enhancements, we have also applied JPEG recompression ranging from quality factor 100 to 10, Gaussian blur with σ ranging from 0 to 3, horizontal flipping, and mild background color jitter to diversify visual characteristics. We have applied cross-gender, cross-race, and cross-age face swaps to prevent dataset bias. In total, we have generated 2603 fake videos. The number of deepfake videos may be increased using the methods described above. Each video is labeled as *Real* or *Fake*. For fakes, we record the generation method, identity pairing, post-processing filters(JPEG, blur, flip). This metadata supports fine-grained analysis, such as method-specific evaluation.

The difficulty of a deepfake dataset mostly depends on several factors that contain identity diversity, the coverage of the manipulation method, the robustness of the real world, the granularity of the label, and the difficulty of detection. For this purpose, we include a diverse set of identities of different ethnicities, different ages, and different genders. Furthermore, as mentioned above, we have carefully labeled the dataset and included more metadata. Considering that our real data come from the real world directly and our fake data is generated using state-of-the-art generation tools, we ensure that our dataset meets real-world conditions. Moreover, we conducted a wide range of experiments in Chapter 4 to assess the detection difficulty of our dataset using state-of-the-art detection methods. Considering these factors, we believe that our dataset

raises the bar for detection difficulty and offers a benchmark that better approximates real-world deepfake scenarios.

3.2. Pretraining with Glint360K

Most deepfake detectors in the literature utilize backbones pretrained on ImageNet [19] which is a generic object-centric dataset. Although ImageNet pre-training imparts broad visual features, it may not be optimal for face analysis tasks. Our approach instead leverages Glint360K, a domain-specific dataset of faces, to pretrain model backbones. The intuition is that face-specific pretraining can tune the backbone to facial structures, skin textures, and idiosyncrasies of human faces from the start, thereby providing a better feature foundation for detecting face manipulations. We expect that models initialized this way will more readily pick up on subtle anomalies in forged faces (e.g., unnatural skin blending, facial feature distortions, or inconsistencies in lighting on a face) than if they were initialized from generic features.

Glint360K [14] is one of the largest publicly available face recognition datasets, containing over 17 million face images from more than 360,000 identities. The images are aligned and centered on faces, covering a wide range of individuals, poses, and lighting conditions. This abundance of data and identities makes it ideal for pretraining deep models to learn rich face representations. We curated the Glint360K data for our use by ensuring all images are consistently cropped to the required input size of our models (e.g., we used 112×112 aligned face crops provided by the dataset, resizing to 224×224 where needed to match model input dimensions). Standard preprocessing (pixel normalization and random augmentations like flips or slight color jitter) was applied to simulate variability. Importantly, the pretraining task on this dataset is formulated as face identity classification: the model must predict which identity (out of 360K classes) the input face belongs to. This task forces the model to extract facial features.

Training on 360K identity classes poses a risk of the model memorizing specific identity features too strongly. As discussed in recent work, deepfake detectors can suffer from implicit identity leakage (IIL) – over-reliance on identity cues that do not generalize to new faces [56]. To counteract this, we employ label smoothing in the loss function during pretraining. Instead of using one-hot targets for identity labels, we smooth them with a small factor (we use $\epsilon = 0.1$, meaning 10% of the label probability mass is distributed uniformly among all other identities). This way, the network never assigns complete certainty to any identity classification, which serves as a regularizer. Label smoothing has been shown to reduce overfitting and improve generalization by preventing the network from becoming over-confident [57]. In our context, it encourages the backbone to learn features that distinguish faces in a more generalizable way (e.g., focusing on broad facial characteristics and texture patterns) rather than picking up spurious identity-specific quirks. We hypothesize that this will make the pretrained features more transferable to the deepfake detection task, where the model should focus on fake vs real discrepancies rather than who the face is.

Each model's backbone (e.g. ResNet and EfficientNet or the transformer encoder of ViT) is pretrained on Glint360K for a certain number of epochs using the identity classification objective. Given the enormous size of Glint360K, we adopt strategies to make training feasible: we use mini-batch stochastic gradient descent with a large batch size (e.g., 256 images per batch, distributed across GPUs) and employ an optimized sampling of classes per batch (a partial softmax strategy similar to ArcFace's partial FC, to handle the 360K-dimensional output layer efficiently). We train for 1 epoch over the full dataset, which comprises roughly 17 million samples – this epoch already constitutes a massive number of iterations (in practice, we ran $\sim$100k mini-batch updates, which is equivalent to covering the dataset once). The learning rate is initially set relatively high (e.g., 0.1 for ResNet backbones) with a gradual warm-up in the first few thousand iterations, followed by a cosine decay schedule to a minimum of 0.01 by the end of training. We found that one epoch was sufficient to converge to a reasonable identification accuracy (owing to the dataset's scale), and further epochs gave diminishing returns under time constraints. During pretraining, we freeze any non-backbone

parts of the model architectures (for instance, classification heads or deepfake-specific layers that are not used in this stage). Essentially, we treat each model like a feature extractor network and append a temporary $360k$-way classification layer for the identity task; only the backbone parameters and this classification layer are updated during face training. The final backbone weights (after convergence on Glint360K) are saved for use in the fine-tuning stage.

3.3. Compared Methods

In this section, we have explained the state-of-the-art models we have selected to apply our approach. We have prioritized the best performing up-to-date methods to be able to assess the performance of our approach. Below we provide technical descriptions of each model, including their core architecture, detection strategy, and any adaptations we introduced for our experiments.

3.3.1. IIL - Implicit Identity Leakage

The IIL approach proposed by Dong *et al.* [15] is a deepfake detector designed to address the challenge of implicit identity leakage. As visualized in Figure 3.1, the core idea is that a standard binary classifier may inadvertently use identity-related features to distinguish real from fake images (for example, if the training data contains a person only in manipulated form and never as real, the model might "learn" that this identity corresponds to fake – a clearly non-generalizable cue).

The IIL method introduces an architecture that guides the model to focus on forgery artifacts rather than identity. Figure 3.2 shows the detailed architecture: The model takes an input image and processes it through a backbone network (like ResNet or EfficientNet) to extract features. Extra layers are added after the backbone to provide feature maps at different spatial scales (e.g., 7x7, 5x5, 3x3, 1x1). These features are fed into the Artifact Detection Module (ADM). The ADM contains a Multi-scale Detection Module connected to several of these extra layers, which uses multi-scale

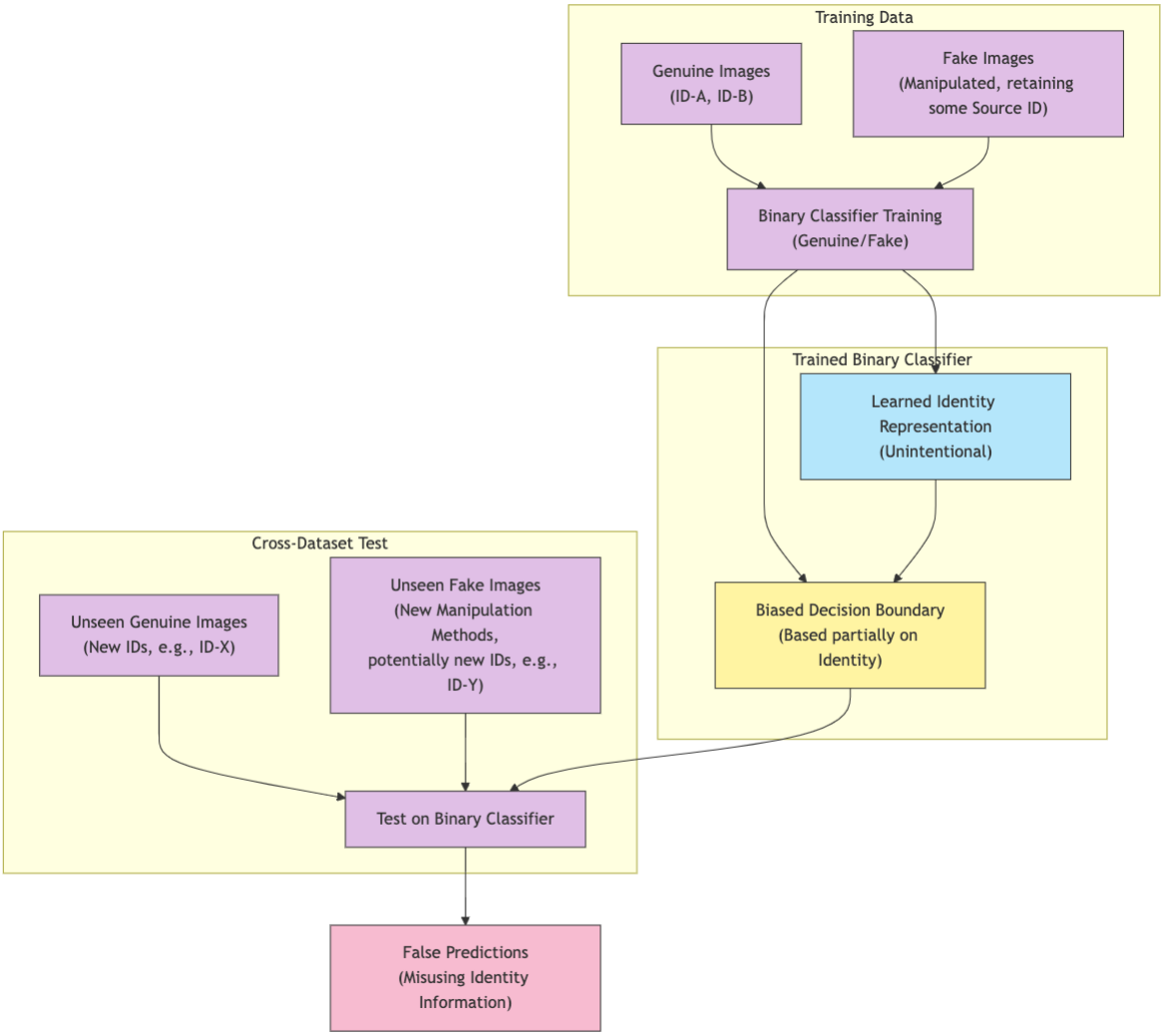


Figure 3.1. Implicit identity leakage problem. Model implicitly learning identities in the training data - results in false predictions when unknown identity shown.

anchors to detect the position and confidence of artifact areas. This module is supervised by a Detection Loss (Ldet), composed of a Confidence Loss (Lconf) for anchor classification and a Location Loss (Lloc) for bounding box regression. Detection loss can be formulized as follows:

$$L_{\text{det}} = \frac{1}{N} \left(L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g) \right) \tag{3.1}$$

where N is the number of positive anchor boxes, i.e., fake anchors; x $\in$ {0, 1} is an indicator for matching the default anchor to the ground truth of artifact areas; c denotes the class confidences; l and g denote the ADM predicted box and the artifact

area ground truth box; α denotes a positive weight.

The ADM's final features from the smallest scale (1x1) are connected via a short connection to the main feature path. The final feature for classification is formed by combining the backbone's final feature and the ADM's final feature, which is then passed through a fully connected layer to produce the Final Prediction (fake or genuine). The overall training is guided by a composite loss function (Overall Loss L) that combines the Detection Loss (Ldet) and a Classification Loss (Lcls) for the final binary prediction. Overall loss L can be formulized as follows:

$$L = \beta L_{\text{det}} + L_{\text{cls}}. \tag{3.2}$$

where β is a positive scalar which controls the trade-off between Ldet and Lcls

This architecture is designed to force the model to focus on local artifact features rather than global identity information, thereby reducing IIL. We adopted this architecture and its training strategy from the original authors, with minor adaptations: during fine-tuning, we initialized the backbone with our Glint360K-pretrained weights (instead of ImageNet) and left the artifact module weights random at the start. We then trained the entire model on the deepfake dataset, using a loss as described in the original method. By training with this guidance, the IIL model is forced to reduce its reliance on identity features, as evidenced by Dong *et al.* reporting improved generalization to unseen identities. We will evaluate if our face-pretrained variant of IIL further enhances this generalization in Chapter 4.

IIL's strategy is to detect fakes via common artifact features. The model looks for telltale signs left by the manipulation process, rather than who is in the image. For example, compression artifacts, blur inconsistencies around the face outline, or implausible reflections in eyes are factors it can learn. Qualitatively, we expect the IIL model to excel in scenarios where identity-switch cues are not reliable – e.g., deepfakes of known identities or completely unknown identities – since it pays attention to the how of the fake rather than the who. We also anticipate that, among the models, IIL may be more robust when confronted with unseen forgery algorithms, because focusing on

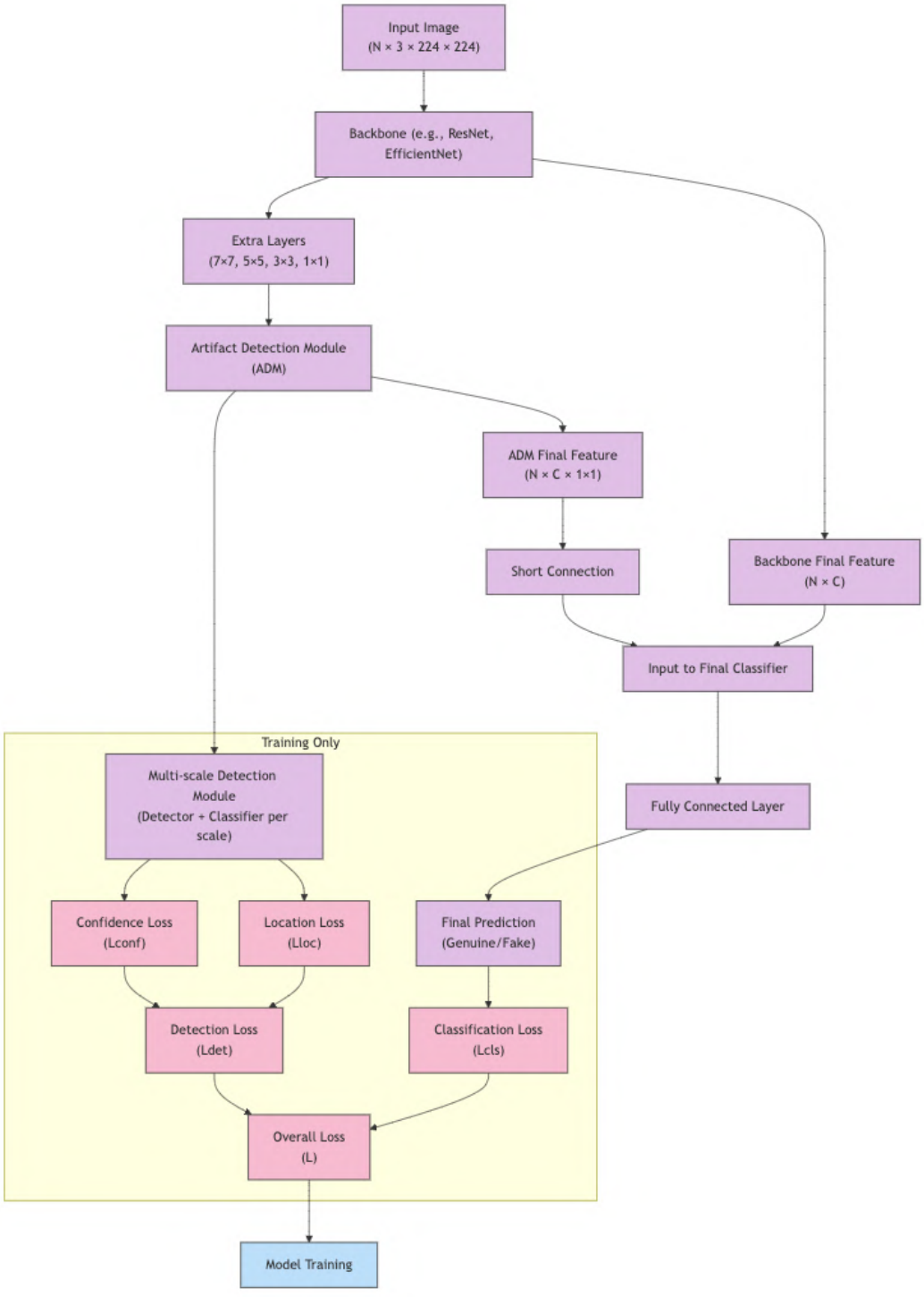


Figure 3.2. IIL deepfake detection model architecture. Detection loss and classification loss work together to decide final label.

common artifacts (e.g., color mismatches, frequency noise) is a more general approach. This aligns with the original claim that IIL-based training improved cross-manipulation

AUC by a large margin. Our experimental setup will test IIL on multiple datasets to verify this claimed robustness.

3.3.2. LAA-Net – Localized Artifact Attention Network

LAA-Net is a recent high-performance deepfake detector introduced by Nguyen *et al.* [16], designed specifically for detecting high quality deepfakes. High quality forgeries often evade detection by making only subtle alterations, so LAA-Net aims to pinpoint small, localized artifact regions. The architecture (illustrated in Figure 3.3) is built on a CNN backbone enhanced with an explicit attention mechanism and a feature pyramid.

For our experiments, we re-implemented LAA-Net with a ResNet-34 backbone (as in the original paper) and integrated our pretraining approach. We loaded the ResNet backbone weights from the Glint360K-pretrained model. The extra layers specific to LAA-Net – namely, the attention heatmap generator and the E-FPN layers – were initialized randomly since they have no counterpart in the pretraining phase. We then fine-tuned the entire network on the deepfake datasets. During training, the attention mechanism was active: the predicted heatmap was fed back to modulate intermediate features, effectively attending to localized regions hypothesized to contain artifacts.

Concretely, LAA-Net operates in a multi-task learning framework. It has a standard binary classification head, but alongside it, LAA-Net generates a pixel-level artifact heatmap indicating where the image might be manipulated. During training, both outputs are supervised: the classification branch learns real/fake labels, and the heatmap branch learns to highlight known vulnerable regions on the face. The LAA-Net attention mechanism combines two strategies: (1) Heatmap-based attention, using the learned heatmap to force the network's focus onto a few small regions likely to contain artifacts, and (2) self-consistency attention, ensuring that the network's internal feature representations concentrate on those regions consistently. Additionally, LAA-Net introduces an Enhanced Feature Pyramid Network (E-FPN), which takes

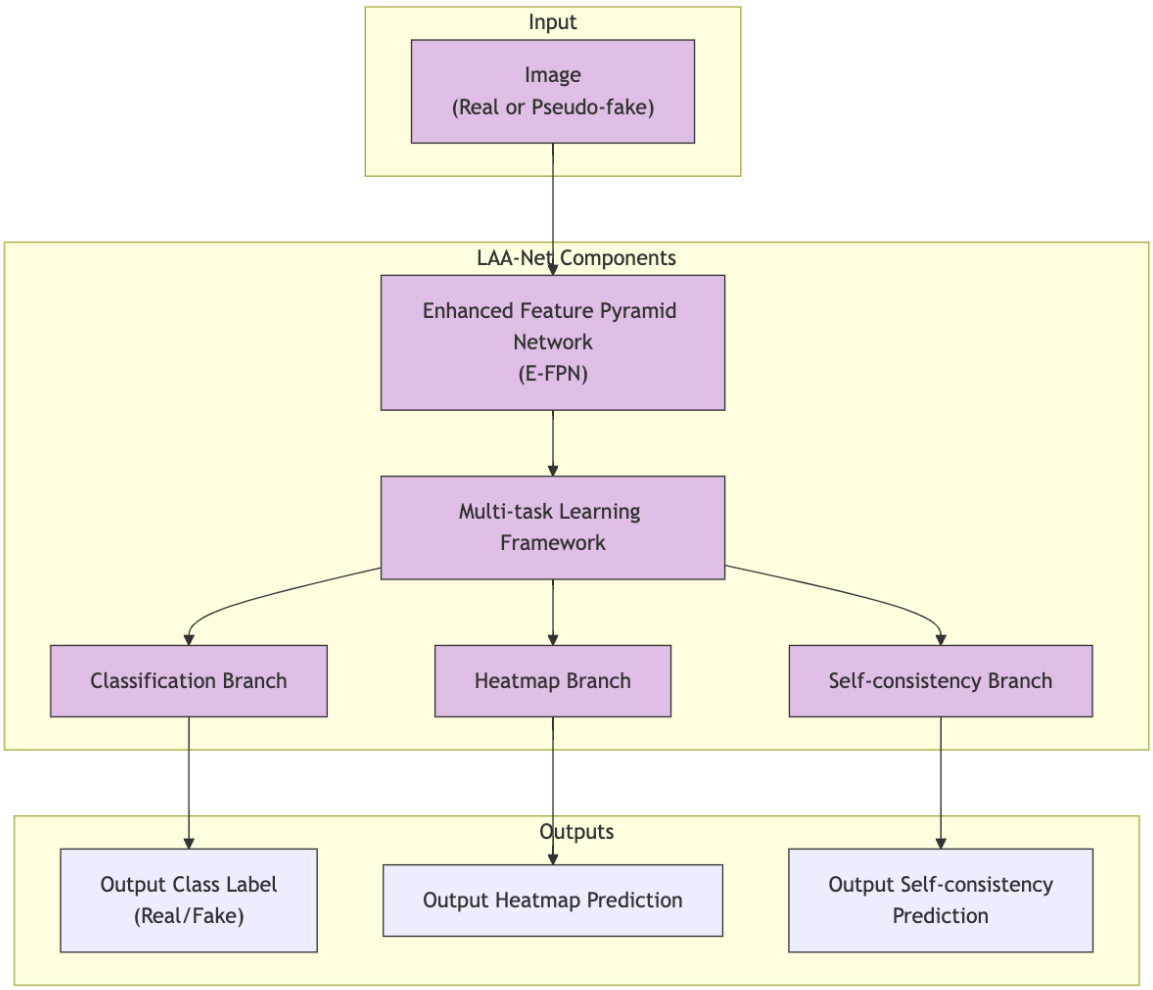


Figure 3.3. LAA-Net deepfake detection model architecture. 3 branches of classification, heatmap and self-consistenct works together to decide final label.

low-level feature maps (that capture fine details like edges and noise) and propagates them effectively to the final layers. This helps in detecting subtle artifacts that might only be visible in lower-level features (such as slight pixel-level discrepancies or high-frequency noise). The E-FPN reduces redundant information while amplifying useful cues for classification. The overall loss of the architecture can be formulized as follows:

$$L = L_{\mathrm{BCE}} + \lambda_1 L_{\mathrm{H}} + \lambda_2 L_{\mathrm{C}} \tag{3.3}$$

where L_{BCE} denotes the binary cross-entropy classification loss. L_H (heatmap loss) and L_C (consistency loss) are weighted by the hyperparameters λ_1 and λ_2, respectively. The L_H is calculated using a Focal Loss between the predicted heatmap and

the ground-truth heatmap which are generated based on detected vulnerable points in the synthesized pseudo-fake images, while L_C is computed as a binary cross-entropy loss between the predicted self-consistency matrix and the ground-truth matrix which is generated in pseudo-fake image generation as shown in Figure 3.4.

LAA-Net detects fakes by zeroing in on small vulnerable facial regions – for example, the eyes, mouth corners, or skin patches – where deepfake generation often introduces artifacts (blurriness, color mismatch, or warping). By explicitly learning where those regions are (via the heatmap) and making the classification decision depend on them, LAA-Net is particularly powerful for catching high-quality fakes that might only betray tiny glitches in a otherwise convincing face. We expect LAA-Net to perform strongly on datasets where the manipulations are subtle. Its explicit localization might also yield interpretability benefits – e.g., highlighting the eye area if a discrepancy is found there – though our focus is on detection accuracy. In cross-dataset tests, LAA-Net's reliance on a few regions could be a double-edged sword: if a new type of fake has artifacts in different regions than those seen in training, the model might overlook them. However, the combination of heatmap and self-consistency attention was shown to generalize better than implicit attention methods, so we anticipate reasonable generalization.

3.3.3. M2TR – Multi-Modal Multi-Scale Transformer

Multi-modal Multi-Scale Transformer, M2TR [17] is a deepfake detection framework that combines convolutional visual feature extraction with transformer-based multi-scale analysis and frequency-domain cues, proposed by Wang *et al.*. Specifically, the original M2TR model employs an EfficientNet-B4 CNN as the backbone feature extractor and feeds its output into stacked Vision Transformer layers that operate on image patches of multiple sizes. The architecture aims to ensure larger patches to convey global context and smaller patches focus to local details more. Together, these components enable M2TR to identify fine-grained forgery traces as well as broader inconsistencies. The authors aimed to improve the model's robustness to various image

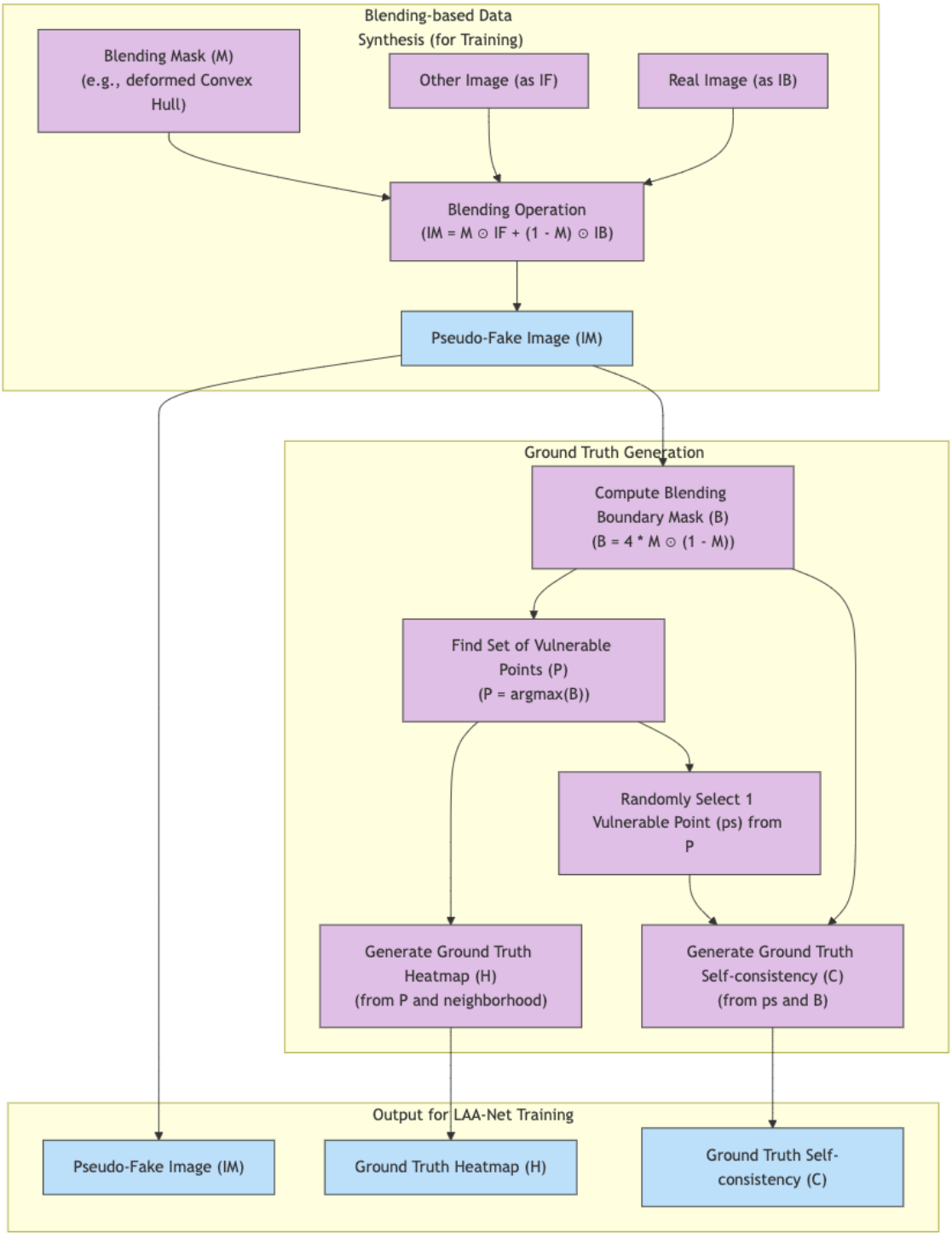


Figure 3.4. LAA-Net pseudo-fake generation. With the help of this process, ground truth labels can be generated to be used in training.

qualities and compression levels by applying this two-branch approach. The general architecture of the M2TR is shown in Figure 3.5.

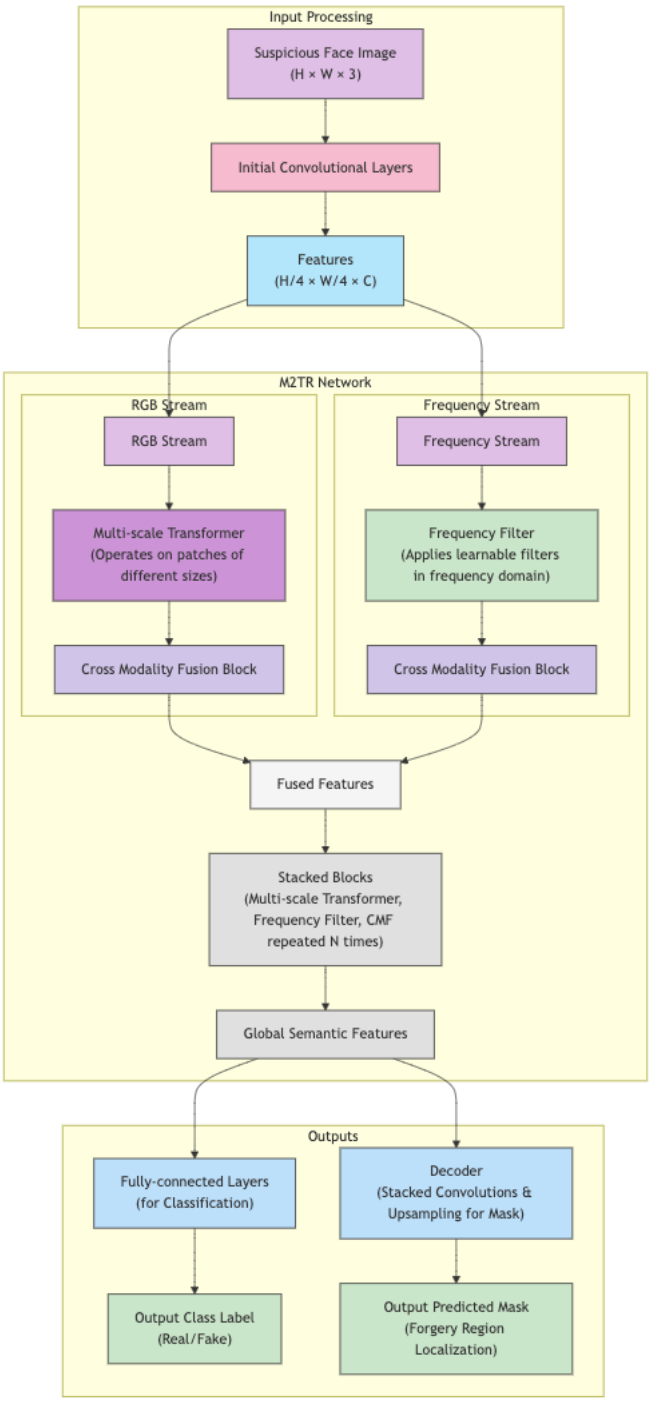


Figure 3.5. M2TR deepfake detection model architecture. 2 branch approach helps to focus not only visual image but also frequency stream of image.

As loss functions, three separate losses are used in M2TR, namely classification loss, segmentation loss, and constractive loss. For the classification loss the authors

use binary cross entropy, for the segmentation loss they use ground truth masks as the authors know the exact manipulated regions during the synthesis process. The ground truth mask has a value of 1 for manipulated pixels and 0 otherwise, providing pixel-wise supervision. And for the contrastive loss, it is used to improve the generalization ability of the detection model. This is achieved by encouraging the global semantic features derived from real samples to be closer to a calculated feature center of real samples than the features of manipulated samples are to that same center. The real sample feature center is calculated by averaging the features of real samples. The overall loss can be formulized as follows:

$$L = L_{\text{cls}} + \lambda_1 L_{\text{seg}} + \lambda_2 L_{\text{con}} \tag{3.4}$$

where λ_1 and λ_2 are the balancing hyperparameters.

In our implementation of M2TR, we modified the model's initialization to enhance its focus on facial features. In particular, instead of using the standard ImageNet-pretrained weights for the EfficientNet-B4 backbone, we initialized it with weights pre-trained on the Glint360K face recognition dataset. The ViT-based transformer module was kept the same as in the original implementation (using the authors' default). This way, we inject domain-specific knowledge into the backbone without constraining the transformer's capacity to learn deepfake-specific patterns. We fine-tuned the M2TR model (with the Glint360K-pretrained EfficientNet backbone) on our target deepfake dataset using an end-to-end training approach. All layers were left trainable, but we applied a lower learning rate to the pre-trained backbone to avoid disrupting its learned features. This fine-tuning setup allowed the model to adapt the face-specialized backbone features to the deepfake detection task, while the transformer learned to effectively discriminate real versus fake images based on the multi-scale inconsistencies.

3.3.4. LSC – Learning Self-Consistency Approach

Learning Self-Consistency for deepfake detection, LSC [18] is an approach introduced by Zhao *et al.* focused on training a deepfake detector using only real images. The key idea is to model the internal consistency of real faces and identify fakes as

those that violate this consistency. The LSC architecture can be based on a standard CNN but is employed in a different training paradigm. Figure 3.6 shows the concept: two branches are utilized during training, two copies or views of a real face image are taken – one possibly altered or partially swapped – and the model must predict whether the input is consistent (original real image) or inconsistent (manipulated version) in classification branch while predicting the altered part of the image in the consistency branch.

For example, the training might take one real face, randomly swap a region (like the eyes) with another face's eyes , and label that as fake/inconsistent. The model learns to detect this discrepancy. Essentially, LSC creates a self-supervised binary classification task: real vs locally-mismatched face. Over time, the network learns to spot when different parts of a face do not come from the same person (i.e. a lack of self-consistency). The loss function utilized for consistency branch as follows:

$$L_{\text{PCL}} = \frac{1}{N} \sum_{h,w,h',w'} \text{BCE}(\mathbf{V}_{h,w,h',w'}, \hat{\mathbf{V}}_{h,w,h',w'}) \tag{3.5}$$

where V stands for 4D consistency volume which is calculated between patches of image to show if the patches are consistent. For the classfication branch, well-known cross entropy is used and the total loss is calculated as follows:

$$L = \lambda L_{\text{PCL}} + L_{\text{CLS}}, \tag{3.6}$$

with hyper-parameter λ.

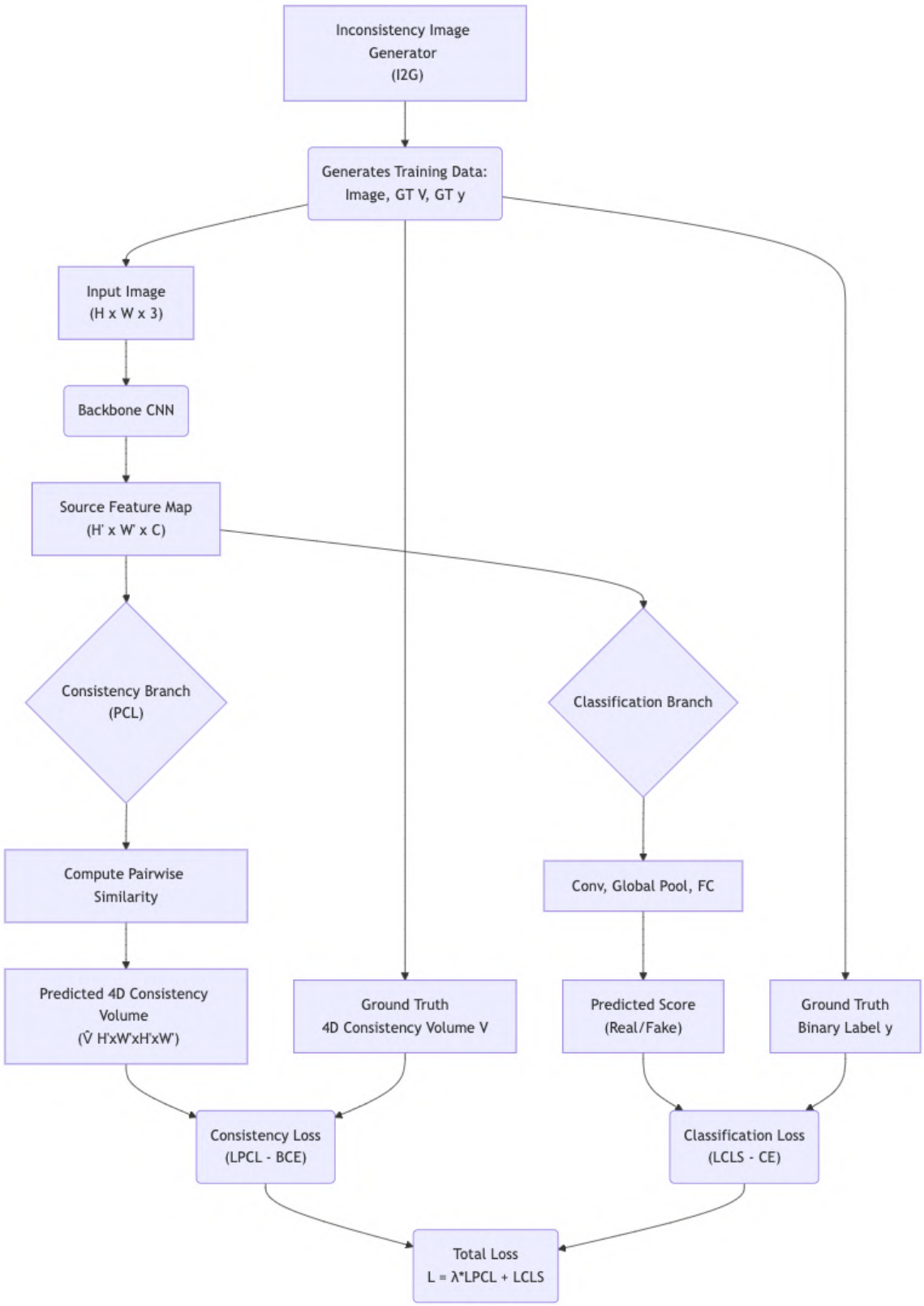


Figure 3.6. LSC deepfake detection model architecture. Generating synthetic data enhances training process to focus on fake areas of the image.

In practice, we implemented LSC using a ResNet-34 backbone. We initialized this backbone with our Glint360K pretraining. Then we trained the network on only real faces as suggested in the original paper. We followed the approach described by Zhao *et al.*. We generated fake training samples on-the-fly by mixing two real faces. Concretely, we would take face A and face B, split A's image into patches or blocks, randomly replace one block (for instance, the region containing the nose and mouth) with the corresponding block from B's image, and feed this composite to the network with a label "inconsistent (fake)". The unaltered real image A would be labeled "consistent (real)". The network's task is binary classification of consistency. We applied label smoothing here as well (with a smaller $\epsilon = 0.05$) to avoid the network becoming overconfident due to potentially easy examples. After this self-consistency pretraining, the model has essentially learned a one-class concept of a real face and how subtle cross-face inconsistencies look. For fine-tuning on actual deepfake detection, we then further trained the model on real vs fake data (supervised) but starting from the weights learned through the self-consistency task. This two-phase training (first unsupervised consistency learning, then supervised fine-tuning) is our adaptation to integrate LSC with our pipeline. Note that in the fine-tuning stage, we again only updated the latter layers of the network initially (since the earlier layers already capture general facial features), then gradually unfroze more layers.

LSC essentially detects anomalities, a fake will appear as an inconsistency relative to the internal distribution of a real face. For example, if the eyes region doesn't "match" the rest of the face, the model flags it. One remarkable aspect is that LSC doesn't need to see actual deepfakes in training, thus it tends to generalize very well to unseen fake methods. A possible weakness is that if a fake is too subtle and remains self-consistent in appearance, LSC might not catch it because it looks like a coherent face. Nonetheless, most deepfakes do introduce tiny inconsistencies that a well-trained LSC model can pick up. Also, because LSC was trained on only real data and synthesized inconsistencies, it might have lower absolute accuracy on in-dataset evaluation compared to models that directly trained on those fake.

3.3.5. ICT - Identity Consistency Transformer

The Identity Consistency Transformer, ICT [9] is a deepfake detection model that focuses on the consistency of a person's identity across different regions of the face. ICT adopts a ViT-based architecture that generates two identity feature representations from each face image – one for the inner facial region (containing the key facial features) and one for the outer region (peripheral facial features such as hair, jawline, and background). The underlying assumption is that in a genuine (unaltered) face, both the inner and outer regions correspond to the same individual, whereas in a deepfake, the inner face may have been swapped or altered, leading to an identity mismatch with the outer context. To detect this discrepancy, ICT introduces an identity consistency loss use the cosine-based softmax loss proposed in ArcFace [58]:

$$L_{\text{ident}} = -\log \frac{e^{s\cdot\cos(\theta_{i,j}+\mathbf{1}_{\{j=y_i\}}\cdot m)}}{\sum_{j=1}^{n} e^{s\cdot\cos(\theta_{i,j}+\mathbf{1}_{\{j=y_i\}}\cdot m)}} \tag{3.7}$$

$\theta_{i,j}$ is the angle between W_j and f_i. $W_j \in \mathbb{R}^d$ is the j-th column of weights of the last fully-connected layer, $f_i \in \mathbb{R}^d$ is the feature of the i-th sample in the input batch. $\mathbf{1}\{j = y_i\}$ is the indicator function that returns 1 when $j = y_i$ and 0 otherwise. s is a scale hyperparameter and m is an additive angular margin. Furthermore, consistency loss between inner and outer faces are also utilized as follows:

$$\begin{aligned} L_{\text{consist}} = \mathbf{1}\left\{p_{ij}^{\text{in}} = p_{ji}^{\text{out}} = y_i\right\} \left\|\mathbf{f}_{ij}^{\text{in}} - \mathbf{f}_{ji}^{\text{out}}\right\|_2^2 \\ + \mathbf{1}\left\{p_{ji}^{\text{out}} = p_{ij}^{\text{in}} = y_j\right\} \left\|\mathbf{f}_{ij}^{\text{out}} - \mathbf{f}_{ji}^{\text{in}}\right\|_2^2. \end{aligned} \tag{3.8}$$

Finally overall loss becomes:

$$L = \mathcal{L}_{\text{ident}} + \eta\mathcal{L}_{\text{consist}} \tag{3.9}$$

where η is the hyperparameter.

The model is trained to produce similar identity embeddings for the inner and outer face regions when the image is real, and to produce increasingly divergent embeddings when the image is a fake (different identities in inner vs. outer regions). This enforces a high-level semantic consistency check exposing forgeries even when pixel-level artifacts are minimal, by essentially asking whether the "person" in the face image is internally consistent. The complete flow can be seen in the Figure 3.7.

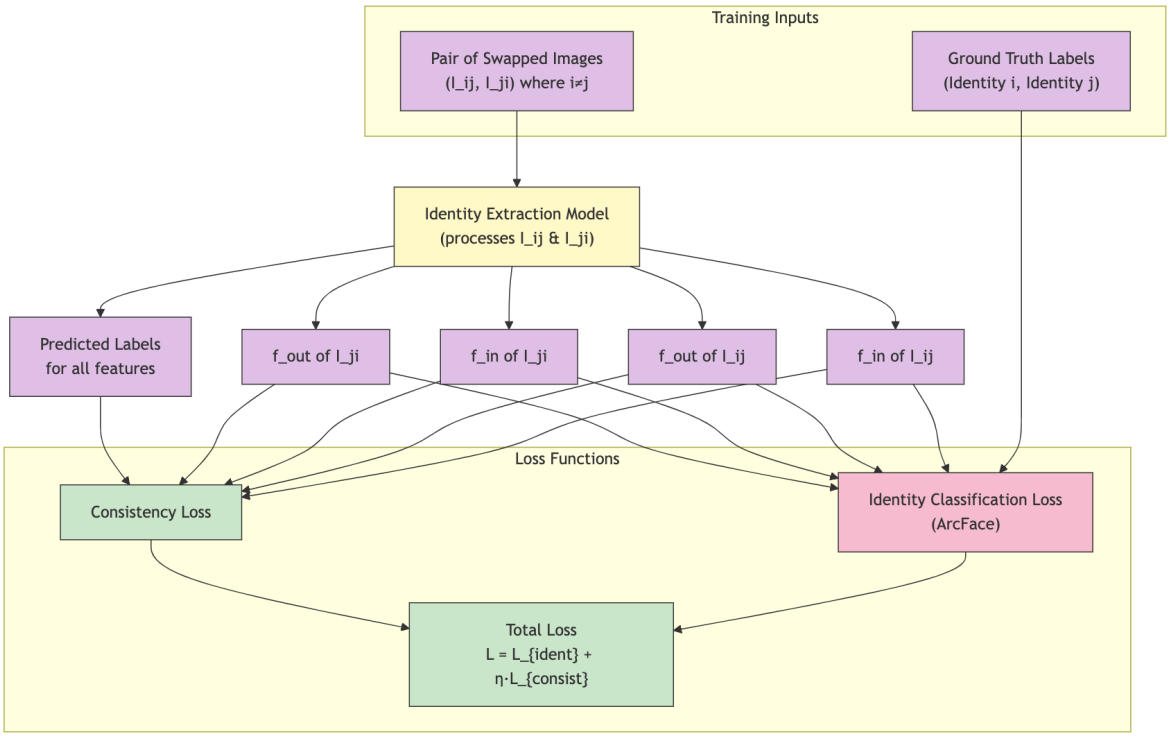


Figure 3.7. ICT deepfake detection model architecture. Identity loss and consistency loss work harmonically to further advance detection process.

To train the ICT model effectively, we employed a two-stage training strategy. First, we pre-trained the ViT backbone on a large-scale face recognition task using the Glint360K dataset. Glint360K, with over 17 million face images of around 360,000 individuals, provides a rich supervisory signal for learning identity-discriminative features. In the second stage, we fine-tuned the ICT model on our deepfake detection dataset. The model was optimized with a combination of the identity consistency objective and a binary classification objective. Specifically, a binary cross-entropy loss supervised the real/fake classification for each image, while the identity consistency loss simultaneously enforced that the two identity embeddings be congruent for real images and divergent for fake images. In practice, for a genuine face, the inner and outer identity embeddings are encouraged to be nearly identical, whereas for a fake, the embeddings are pushed apart to reflect an identity mismatch. This fine-tuning approach, supported by the strong identity-aware initialization, allowed ICT to achieve high sensitivity to identity inconsistencies in forged images.

3.3.6. XceptionNet

As for our first baseline, we have used XceptionNet [4], proposed by Chollet *et al.*. XceptionNet leverages depthwise separable convolutions and residual connections different from traditional CNNs.

In our implementation, we train the XceptionNet baseline in two scenarios same as the above methods: (a) ImageNet pretrained, and (b) Glint360K pretrained. After that pretraining step, we fine-tune the both networks with deepfake dataset to adapt the model detect deepfakes with the support of previously learned face-related features. Furthermore, we have also applied label smoothing during pretraining to improve generalizability of the model and prevent overfitting on identity-specific features that comes from pretraining on Glint-360K.

To support our hyptohesis that pretraining in face-specific dataset improves deepfake detection ability of deepfakes, we have also conducted extensive experiments which are shown in Chapter 4.

3.3.7. ResNet Baseline

As a traditional baseline, we include a standard convolutional network classifier based on ResNet-34 architecture. ResNet-34 [6], with its 34 layers and skip-connection design, was a top performer on ImageNet and has been used in many deepfake detection studies as a baseline feature extractor or classifier. Our ResNet baseline model is simply the original ResNet-34 backbone followed by a global average pooling and a single fully connected layer producing a fake vs real probability. It does not have any specialized modules for artifacts or attention. We include this model to gauge the effectiveness of the advanced methods above relative to a plain CNN approach.

For fair comparison, we train the ResNet baseline in two scenarios: (a) ImageNet-pretrained, where we initialize with the standard ImageNet weights (as is common

practice), and (b) Glint360K-pretrained, where we initialize with our face-specific pretraining weights. In both cases, we fine-tune the entire network on the deepfake training data using binary cross-entropy loss. We use the same training schedule and augmentations as the other models for consistency. One minor modification we made was applying label smoothing ($\epsilon = 0.1$) even in the fine-tuning stage for this baseline, to prevent it from simply memorizing the training set identities or manipulation patterns – essentially as another regularizer. This is not typically done in all studies, but we found it improved validation generalization slightly for the baseline.

The ResNet baseline simply learns to discriminate real vs fake from raw pixel patterns, without any explicit attention to artifacts. It may learn some artifact features on its own (e.g., it might devote some filters to detecting eye region inconsistencies or blending boundaries), but this emerges implicitly from the data. By comparing the baseline in (a) and (b) scenarios, we will also assess the impact of face-specific pretraining directly: Does giving ResNet a head start with facial features improve its deepfake detection capability? We anticipate it will help to some extent, especially in early training convergence and in focusing on face regions. We provide experimental results in Chapter 4 to supports our central hypothesis.

3.3.8. Vision Transformer Baseline

Our third baseline is a Vision Transformer model (ViT-B/16) [7] used as a deepfake detector. ViT models treat the image as a sequence of patch embeddings and have shown strong performance in image classification tasks. Recently, they have also been explored for deepfake detection, often using large pretraining on generic or self-supervised data. We choose a ViT baseline to represent transformer-based architectures without the bells and whistles of M2TR's multi-scale, multi-modal design. The ViT-B/16 splits the input face image into 16×16 patches, feeds them through 12 layers of transformer encoder (with multi-head self-attention), and uses a classification token output to predict real or fake.

We consider two initialization strategies for ViT as well: (a) ImageNet pretraining, using publicly available weights where the ViT was pretrained on ImageNet, and (b) Glint360K face pretraining, using our own ViT that we trained on the face dataset (for fairness, we used the same architecture for that pretraining). After initialization, we fine-tune the ViT on the deepfake data. Due to the high capacity of ViT and its tendency to overfit on smaller data, we use a few regularization techniques: early stopping based on validation loss, a relatively strong dropout (0.2) in the MLP heads, and partial fine-tuning – i.e., we first freeze the majority of transformer layers and only train the last few layers and the classification head for several epochs, then unfreeze more layers gradually. This strategy was suggested by recent studies to avoid ViT overfitting when data is limited. We also maintain a lower learning rate for the transformer layers (e.g., 1e-5) and a higher one for the new head (e.g., 5e-4) during fine-tuning.

The ViT baseline, like ResNet, does not have specialized artifact modules; however, transformers can capture long-range dependencies and global image context. It might also pick up subtle cues in the patch embeddings such as unusual patch texture that correlates with fakeness which are also shown in Chapter 4 as heatmaps. We will compare the ViT's performance. Notably, if the ViT with face-specific pretraining outperforms the one with generic pretraining on cross-dataset tests, it will reinforce the benefit of domain-specific pretraining. Our experiments will contribute additional evidence on face-supervised pretraining perspective.

3.4. Our Approach

Bringing together the elements discussed, our proposed methodology for deepfake detection is a two-step approach: (1) Pretrain on a face-specific dataset Glint360K to initialize the model's backbone with facial feature knowledge, and (2) Apply label smoothing and related regularization during this pretraining (and, if useful, during fine-tuning) to avoid overfitting and achieve generalizability. This approach is not limited to a single architecture but we apply it uniformly to multiple deepfake detection models as described above. In this way, we aimed to show its general applicability in the

deepfake detection domain which is one of its strengths.

Traditional deepfake detectors often struggle with generalization, especially to new identities or novel fake generation methods. A core reason identified in prior work is that models inadvertently learn who is in the image rather than focusing purely on artifacts of manipulation. By pretraining on a vast face recognition task, we supply the models with a rich understanding of facial features without bias toward a particular dataset's identities, since Glint360K has hundreds of thousands of identities, the model doesn't latch onto just a few. However, a risk emerges that the model might get too good at recognizing identities – known as implicit identity leakage when those features carry into the detection task. Our introduction of label smoothing and further regularizations during pretraining and finetuning is aimed to resolve this issue so that the model cannot purely memorize each identity with absolute confidence. Instead, it must learn more general features that differentiate faces (for the classification task) which likely include intrinsic facial patterns that remain useful for detecting oddities in a fake. We are effectively encouraging a form of feature regularization, making the backbone less sensitive to exact identity and more sensitive to face realism.

In practice, our proposed training pipeline for any given model is:

(i) Backbone Pretraining: Train the backbone on Glint360K for face identification, using cross-entropy with label smoothing ($\epsilon = 0.1$). We used large-batch SGD/AdamW as detailed earlier. This stage yields a pretrained backbone B_{face}.

(ii) Weights Transfer: Initialize the deepfake detection model's backbone with B_{face} weights. Initialize any new layers (classification head, attention modules, etc.) randomly or from their respective sources

(iii) Fine-tune on Deepfake Data: Train the model on the deepfake detection task (real vs fake). Here we often continue to use some regularization: for classification heads, we sometimes apply label smoothing (e.g., $\epsilon = 0.05$ on the binary labels) to prevent the model from becoming overconfident on the training fakes. This helps especially since deepfake datasets can be small, and without label smoothing a

model might quickly drive the loss to near-zero on training data, risking overfit. We also use standard techniques like data augmentation and dropout as described.

(iv) Gradual Unfreezing: As noted, we do not fine-tune all layers at once. Our approach involves gradually unfreezing the backbone so that the low-level features – which are likely quite general (edges, contours, etc. learned from millions of face images) – are not destroyed by sudden retraining on a small dataset. Only higher-level features adapt to the specific fake vs real classification. By the end, we do fine-tune everything, ensuring the model can adjust even low-level filters if needed.

Through this method, we aim to achieve improved accuracy in both intra-dataset and inter-dataset. We expect to see that models trained with face-pretrained backbones and label smoothing exhibit less overfitting in metrics like a smaller gap between training and validation accuracy, and higher cross-dataset AUCs, compared to models with conventional training. For example, if a ResNet without our method gets high accuracy on FF++ but low on Celeb-DF, our ResNet with face pretraining might maintain more robust performance on Celeb-DF. Similarly, advanced models like LAA-Net or M2TR might see their strengths further enhanced – e.g., LAA-Net might localize artifacts even better if its backbone already knows how a normal eye or mouth region should look thanks to face pretraining. In Chapter 4, we will provide quantitative evidence of these improvements.

To summarize, our proposed method is an enhanced training regimen rather than a new network architecture. Its novelty lies in repurposing a face recognition dataset for deepfake detection pretraining and in the specific use of label smoothing in this context – an approach that, to our knowledge, hasn't been explicitly documented in prior deepfake literature. By applying this method across several model architectures, we demonstrate its versatility and impact.

3.4.1. Implementation Details

Each model undergoes a two-stage training: (1) Pretraining on Glint360K (face dataset), followed by (2) Fine-tuning on the target deepfake detection dataset(s) using transfer learning technique as simply visualized in Figure 3.8. Stage (1) was described in Section 3.2. Here we provide additional practical details. Stage (2) involves training on one or more deepfake datasets – in our case, we primarily use FaceForensics++ (FF++) as the main training dataset and validate on Celeb-DF v2 and Our Custom Dataset in which the details will be showed in Experiments Chapter to assess generalization. All models are implemented in PyTorch and trained on an 8-GPU server for efficiency.

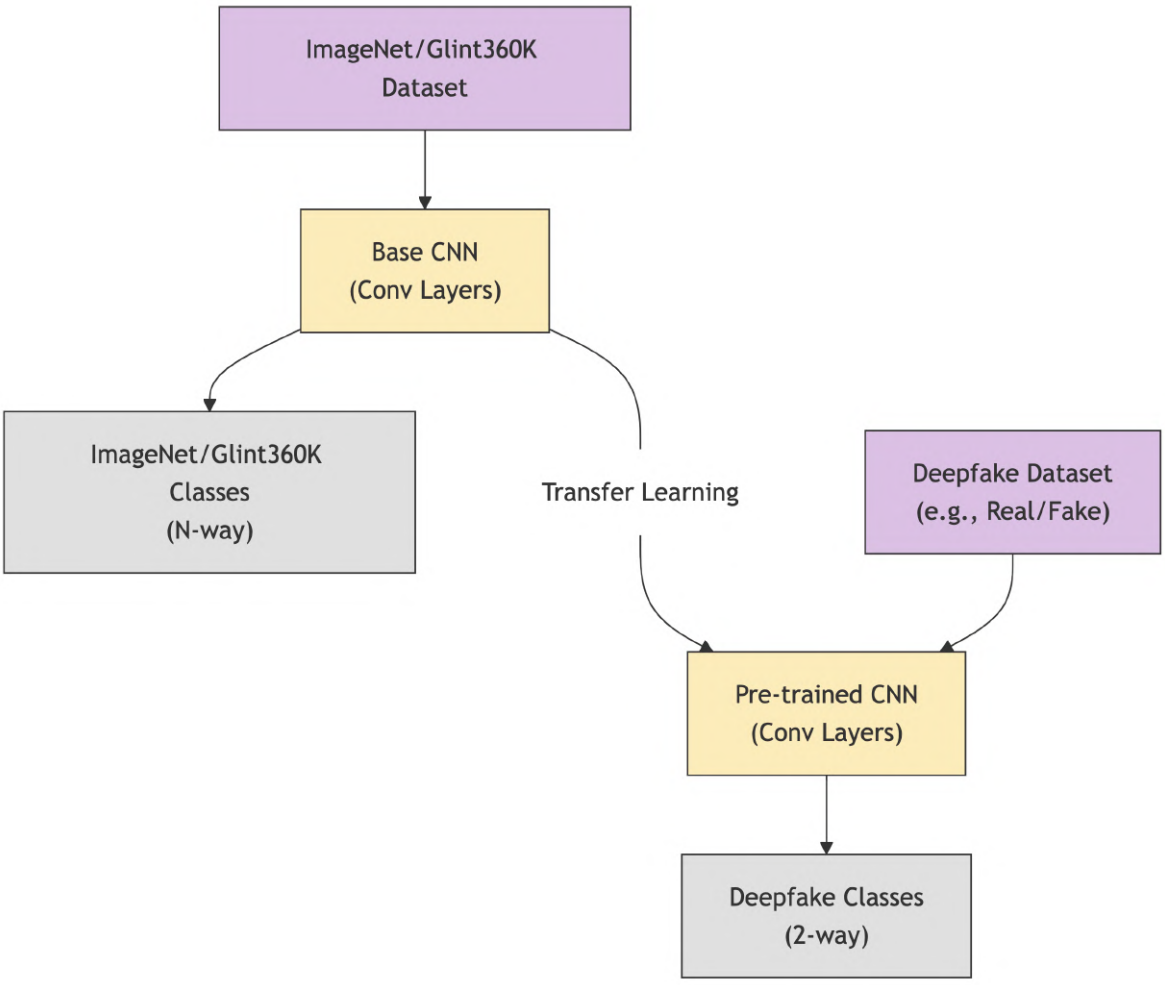


Figure 3.8. Transfer learning approach used in deepfake detection.

We use AdamW optimizer for most experiments, as it handles the different learning rate needs of our models well (especially for transformers) and provides slight regularization via weight decay. For CNN-based models (ResNet, LAA-Net, IIL, LSC), we set an initial learning rate of 0.0005 for fine-tuning. For transformer-based models , we use a lower base learning rate of 0.0001 for the backbone and a higher one (0.001) for newly added layers or unfrozen layers, employing a two-stage fine-tuning schedule. We use a batch size of 64 for fine-tuning (per GPU batch of 8 images, aggregated across GPUs) which was the maximum that fit in memory for the largest models. Training is run for 25 epochs on the deepfake data in each experiment, with early stopping if the validation loss doesn't improve for 5 consecutive epochs. We found 25 epochs sufficient for convergence in all models (some like LSC converged faster around $\sim$15 epochs due to simpler objective after pretraining). We apply learning rate scheduling: a linear warm-up for 1 epoch, then a cosine decay or step decay depending on the model. Weight decay is set to $1e-4$ to reduce overfitting.

The model specific details is also as follows:

- IIL: We used two losses (classification and artifact). The artifact branch used an Adam optimizer with LR=0.001 (a bit higher to quickly learn the auxiliary task) while the backbone and main classifier used LR=0.0005. We balanced the losses with a weight factor (artifact loss weighted by 0.5 relative to classification loss) as in the original paper.
- LAA-Net: Training LAA-Net required careful tuning of the heatmap loss. We set the heatmap regression target as a Gaussian blur around known manipulated keypoints (for training fakes we used the ground truth manipulation mask if available, or else approximated by facial landmarks difference between source and target). We applied a weight of 1.0 to classification loss and 0.3 to heatmap MSE loss to prevent the heatmap task from dominating. Gradient clipping (at norm 5) was enabled due to occasional spikes from the multi-task learning.
- M2TR: For M2TR, The ViT-based transformer module was kept the same as in the original implementation. The training ran for approximately 30 epochs.

We used the AdamW optimizer with an initial learning rate of $1x10^{(-4)}$ for the transformer module and a smaller $5x10^{(-5)}$ for the backbone; the learning rate was decayed by a factor of 0.1 every 10 epochs to encourage stable convergence.

- LSC: In the LSC training, we trained for 30 epochs on purely real faces (From FF++ augmented by inconsitency image generator as in the original paper). Followed the same β value as 0.9 and 0.999. The hyperparameter λ set to be 10 by default.
- ICT: For ICT, After pretraining transformer part with Glint306K, we fine-tuned the model for 20 epochs using a smaller learning rate ($1x10^{(-4)}$) to avoid catastrophic forgetting of the pre-trained features. We continued with the AdamW optimizer. Notably, we applied a minor amount of label smoothing to the deepfake classification targets as well which was 0.1.
- XceptionNet & ResNet & ViT: These baselines were straightforward. For ViT, besides freezing layers as mentioned, we also used knowledge distillation on one run – distilling from a strong CNN (the LAA-Net, for instance) to see if it improved ViT; however, we mainly report the direct trained version. ResNet and XceptionNet had no special tricks beyond heavy augmentation.

During training, we monitor performance on a validation split of the same dataset (for intra-dataset performance) and also periodically test on a hold-out dataset to gauge generalization. For example, if training on FF++, we validated on a portion of FF++ and also checked AUC on Celeb-DF every 5 epochs to see if the model is overfitting to FF++. This helped inform early stopping. At the end of training each model, we evaluate it on multiple test sets: FF++ (seen dataset) and the cross-dataset benchmarks: Celeb-DF v2, Our Custom Dataset. We report standard metrics: Accuracy, AUC (Area Under ROC Curve). We also perform qualitative analysis by visualizing the attention maps for some of the models to see where the model focuses on the face. This consistent implementation framework ensures that each model is given the best chance to perform, and that improvements can be attributed to our proposed pretraining.

4. EXPERIMENTS AND RESULTS

In this chapter, we are going to go through experiments we have conducted to support our hypotheses mentioned in Chapter 3 in detail. The chapter includes detailed quantitative and qualitative assessments that confirm these methodological decisions. We evaluated the effectiveness of models trained with face-pretrained weights compared to those with ImageNet-pretrained weights, highlighting the impact on detection precision and generalization by comparing them both intra-dataset and inter-dataset. Furthermore, we demonstrate how label smoothing mitigates overfitting and enhances the model's resilience to changes in distribution.

Moreover, we also presented our custom dataset to assess the performance of models. The new dataset reflects real-world conditions better compared to academically available counterparts since it contains images in different environment conditions with diverse lightning taken by individuals using diverse sets of mobile phones.Evaluating models in custom dataset helps us to establish a foundation for comparing methods when the test dataset comes from completely different distributions.

Overall, this approach offers a solid rationale for our hypothesis to examine the influence of pretrained datasets and regularization methods on deepfake detection, establishing a foundation for understanding the experimental outcomes.

4.1. Datasets Used In Experiments

We evaluate detectors on three public deepfake datasets and one custom dataset we created. The public benchmarks are FaceForensics++ (FF++), Celeb-DF v2, and ForgeryNet. Our custom dataset consists of original face videos and deepfake videos generated with modern face-swap tools. Below we describe each dataset and provide a summary of their scale and characteristics (see Table 4.3). Figure 4.1 illustrates example real and fake faces from these datasets.

Table 4.1. Sample images from datasets used in our work. Table contains 2 fake and 2 real images from each dataset.



4.1.1. FaceForensics++

FaceForensics++ (FF++) [3] is a widely used benchmark introduced by Rossler et al. in 2019. It contains 1000 real videos collected from YouTube and ∼4000 manipulated videos created by four face manipulation methods: DeepFakes, Face2Face, FaceSwap, and NeuralTextures. Each real video has been manipulated by these techniques, yielding high-quality forgeries at various compression levels (raw, lightly compressed, heavily compressed). The dataset's diversity spans different genders and multiple resolutions.. While FF++ was pioneering and is still a standard for training and intra-dataset evaluation, many fakes in FF++ have visible artifacts (color mismatches, blending seams) due to the early-generation methods used. Nonetheless, it provides a controlled setting to evaluate detectors on seen manipulations. We use the HQ (high quality) subset for training and testing intra-dataset performance. In our experiments, we reserve 10% of the FF++ data for validation and 10% for testing, using the standard splits from the original authors. Sample images from FaceForensics++ dataset, showing 2 real frame and corresponding deepfake frames. The forgeries exhibit subtle color and texture disparities (e.g., around the facial boundary) compared to the pristine image.

4.1.2. Celeb-DF v2

Celeb-DF v2 [12] is a challenging dataset of celebrity deepfakes released by Li et al. as an improved version of Celeb-DF. It consists of 590 real videos of celebrities and 5,639 fake videos generated by an advanced DeepFake algorithm that improves realism (e.g., by reducing visual artifacts). All videos are high-quality, with authentic speech/interviews as source material and corresponding face-swapped fakes. Importantly, the fakes in Celeb-DF v2 have minimal obvious artifacts, making detection more difficult than on FF++. We use Celeb-DF v2 for cross-dataset evaluation – i.e., models are trained on FF++ but tested on Celeb-DF – to assess generalization to new identities and fabrication techniques. No fine-tuning on Celeb-DF is performed (to simulate a real-world scenario of detecting unseen fakes). This dataset evaluates how well a detector can handle high-quality, unseen deepfakes. As prior work notes, Celeb-DF's limitation is that it includes only one manipulation type (an improved face swap); however, the fakes are very realistic and thus suitable for testing robustness across datasets.

4.1.3. ForgeryNet

ForgeryNet [20] is one of the largest deepfake and forgery benchmarks. It contains an unprecedented scale of data: over 2.9 million manipulated images and 221,247 fake videos covering numerous manipulation approaches. The forgeries span face swaps, expression swaps, reenactment, and even entirely synthetic faces, with source material from both online videos and actors. The dataset was designed for multiple tasks (classification, spatial localization of forgeries, temporal segmentation, etc.), but in our work we focus on the detection task. We selected a representative subset of ForgeryNet (due to its size) for testing cross-dataset generalization. Specifically, we used the provided verification set of ForgeryNet's video forgery classification track, which contains a balanced mix of real and fake videos across different manipulation techniques. ForgeryNet's diversity in generation techniques (7 image-level and 8 video-level methods) and sheer size make it a stress test for detection models. It allows

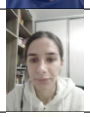
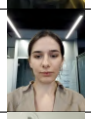
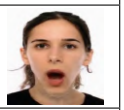
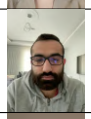
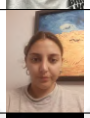
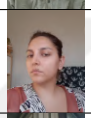
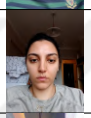
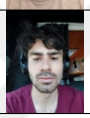
us to see if models can generalize to a wide array of fake-generation methods beyond those seen in FF++. One caveat is that ForgeryNet fakes vary in quality; some are low-resolution or have heavy compression noise, which tests robustness. In summary, including ForgeryNet helps evaluate scalability of detectors to large-scale, diverse forgeries.

4.1.4. Our Dataset

In addition to public data, we created a custom dataset to evaluate our models on newly generated deepfakes closer to real-world use cases. We collected 4252 original videos of volunteers (with informed consent) capturing a variety of individuals in different settings (indoor conversations, frontal and non-frontal angles, different lighting). Using state-of-the-art open-source deepfake generation tools (particularly DeepFaceLab [54] and FaceSwap [59]), we produced approximately 2600 deepfake videos by swapping faces between various pairs of subjects. Each original video yielded 4 deepfake variants on average, using different target faces or different generation algorithms. For example, we swapped Person A's face onto Person B's video and vice versa, to create two fakes, and repeated such pairings across our pool. We ensured a mix of genders and ethnicities to diversify the data. The manipulations in our dataset are high-resolution and employed advanced blending and color correction, so the artifacts are quite subtle (approaching the quality of recent deepfakes on social media). The purpose of this dataset is to test models on completely unseen faces (none of these individuals appear in FF++ or Celeb-DF) and using state-of-the-art generation, which may differ from the academic datasets. Figure 4.2 (described qualitatively here) shows a sample from our dataset: the original frame versus the deepfake frame.

By evaluating on this custom set, we can analyze generalization to realistic, novel deepfakes that detectors might encounter in the wild. FF++ and Celeb-DF v2 are widely-used benchmarks with controlled fake generation methods, whereas ForgeryNet provides a massive, diverse set of forgeries. Our dataset introduces new identities and high-quality fake videos created with modern tools. Real and fake video counts

Table 4.2. Real and fake examples from our dataset. There are 5 rows for 5 sample identity with 4 real and 4 fake images as columns.

	Real Samples				Fake Samples			
ID-1								
ID-2								
ID-3								
ID-4								
ID-5								

are given (for ForgeryNet we list total images). Table 4.3 below summarizes the key statistics of the datasets used.

Table 4.3. Brief summary of deepfake detection datasets used in our experiments. Video datasets splitted to frames for training and testing for image based detectors.

Dataset	**Year**	**Ratio (R:F)**	**Total**
FaceForensics++	2019	1:4	1,000 Real : 4,000 Fake (video)
Celeb-DF-v2	2020	1:9.56	590 Real : 5,639 Fake (video)
ForgeryNet	2021	1:1.01	1,438,201 Real : 1,457,861 Fake (image)
Our Dataset	2024	1:0.61	4,252 Real : 2,603 Fake (video)

4.2. Experimental Setup

We benchmarked eight deepfake detection models in total, which include both conventional CNN classifiers and recent transformer-based or specially designed ar-

chitectures. The models include XceptionNet, IIL, LAA-Net, M2TR, ICT, LSC, a standard ResNet, and a Vision Transformer (ViT). After their backbones trained with Glint360K dataset using regularization techniques, we then finetuned on the FF++ training set with augmentation. In order to ensure evaluation consistency, we computed AUC as our primary metric and also monitored accuracy at the 0.5 threshold for reference. But AUC is emphasized because it is threshold-independent and commonly reported in deepfake detection literature. For more details about training procedures of each model, one can refer to Section 3.4.1.

4.3. Intra-dataset Results

We first report results on the FaceForensics++ test set, where models are trained and tested on the same distribution. This in-dataset evaluation highlights the raw detection capability under ideal conditions (manipulations seen during training). Table 4.4 contains the AUC scores for each model on FF++, including variants with different pretraining. Overall, all models achieved very high AUC on FF++, reflecting that the task is nearly solved for seen fakes which is consistent with prior works where even baseline classifiers exceed 90% on FF++.

Among the CNN-based models, LAA-Net obtained the highest AUC on FF++ (approximately 98.7%), slightly outperforming LSC. LAA-Net's explicit focus on artifact regions likely helped it squeeze out slightly better performance on the known manipulations, which do have telltale local artifacts. Xception and ResNet-34 performed nearly as well, confirming that a strong CNN can effectively memorize the FF++ forgeries. The Vision Transformer (ViT) models also reached 97% AUC, with the Glint360K-pretrained ViT showing a minor edge over the ImageNet-pretrained ViT (97.1% vs 96.7%, for instance). This suggests that face-specific pretraining did not hurt in-distribution performance; if anything, it helped the ViT converge to a better solution, possibly by focusing on the face. The M2TR model achieved around 98.2% AUC, on par with the top CNNs, indicating its multi-scale transformer mechanism is effective even on the known fakes. One surprise was the performance of IIL (ID-unaware model)

on FF++. We expected it to perform strongly given it's a sophisticated model; however, our implementation of IIL achieved roughly 95.5% AUC, which is a few points lower than the others. This was unexpected, it underperformed relative to simpler models. Even the ImageNet version was better than Glint360K version. A possible reason is that the artifact detection module in IIL may not fully engage on the FF++ fakes which often have large, obvious manipulations (e.g., whole face regions replaced). The IIL model might have been too constrained by trying to avoid identity features, thus not leveraging some signals that other models freely use. We have flagged this result for further review, as it may indicate the need to adjust the ADM hyperparameters. Aside from IIL, all other models were tightly clustered at the top range of performance.

Within the Table 4.4 we distinguish models trained from ImageNet initialization versus those from Glint360K face initialization. On FF++, the difference was generally small but consistent: models with face-pretraining had equal or slightly higher AUC compared to their ImageNet-pretrained counterparts. For example, the ViT trained with Glint360K was +0.4% AUC higher than ViT ImageNet. The ResNet-34 pretrained on Glint360K also edged out the ImageNet ResNet by ∼0.3%. These gains are modest (within the margin of error), but they indicate that face-specific features do not harm in-distribution detection and can even help learn the forgery patterns. One hypothesis is that face-pretrained models start with better representations of facial geometry and subtle details (like eye and lip structure), which the fine-tuning can exploit to detect oddities in those regions. In contrast, ImageNet models might initially focus on broader textures and require more finetuning to hone in on faces. Interestingly, for the ICT model, using face-pretraining significantly accelerated training convergence (it reached peak AUC in fewer epochs), though final AUC was similar (∼96.9% with either pretraining). This suggests the identity-consistency loss in ICT might synergize with an identity-aware initialization.

Table 4.4. Intra-dataset(FF++) evaluation of methods. Bold values show the best result for each method.

Method	**Backbone**	**AUC(%)**	
		ImageNet	**Glint360K**
XceptionNet [3]	Xception	96.78	**97.35**
IIL [15]	Efficientnet-b4	**95.92**	95.50
LAA-Net [16]	Efficientnet-b4	98.42	**98.69**
M2TR [17]	Efficientnet-b4	97.31	**98.18**
ICT [9]	Transformer	93.58	**96.88**
LSC [18]	Resnet-34	97.23	**98.34**
Resnet [6]	Resnet-34	95.98	**96.82**
ViT [7]	vit_tiny_p16	96.73	**97.11**

4.4. Cross-Dataset Results

Cross-dataset evaluation is crucial to test the generalization of deepfake detectors. Here, we take models trained on FF++ (as above) and evaluate them on Celeb-DF v2 and on our custom dataset and additionally test subset of ForgeryNet. Table 4.5 summarizes the AUC results for the various models on these external datasets.

As expected, all models suffered an AUC drop when moving from FF++ to Celeb-DF. FF++-trained models, if overfitting to artifacts or specific generator signatures, often struggle with the higher-quality fakes in Celeb-DF. Our results confirm this general trend. For instance, Xception's AUC fell from ∼97% on FF++ to about 73% on Celeb-DF. Similarly, ResNet-34 dropped to ∼71% AUC. These significant drops highlight the generalization gap. However, some models handled this shift much better:

- LAA-Net achieved ∼88% AUC on Celeb-DF, which was the best overall. Its explicit artifact attention seems to have captured more generalizable cues (perhaps generic blending artifacts or texture inconsistencies that persist across methods).

Table 4.5. Cross-dataset(Celeb-DF-v2 and ForgeryNet) evaluation of methods. Bold values show the best result for each method in testing dataset separately.

Method	**Backbone**	**AUC Celeb-DF2(%)**		**AUC ForgeryNet(%)**	
		ImageNet	**Glint360K**	**ImageNet**	**Glint360K**
XceptionNet [3]	Xception	72.92	**73.43**	67.76	**68.53**
IIL [15]	Efficientnet-b4	**84.32**	83.92	80.02	**80.12**
LAA-Net [16]	Efficientnet-b4	86.82	**88.01**	82.90	**84.15**
M2TR [17]	Efficientnet-b4	84.92	**84.98**	78.82	**80.96**
ICT [9]	Transformer	79.58	**85.49**	70.54	**75.23**
LSC [18]	Resnet-34	83.92	**84.02**	**71.12**	70.93
ResNet [6]	Resnet-34	75.15	**78.64**	71.14	**72.78**
ViT [7]	vit_tiny_patch16	76.12	**82.98**	71.94	**77.47**

- M2TR also performed strongly, around 85% AUC on Celeb-DF, This is an encouraging result: the multi-scale transformer with frequency input appears to generalize well. Likely, the frequency artifacts it detects (e.g., inconsistencies in high-frequency spectrum) are present in the Celeb-DF fakes as well, even though those fakes were made with a different algorithm.
- ViT model showed an interesting pattern: the ImageNet-pretrained ViT scored ∼76 AUC, whereas the Glint360K-pretrained ViT scored 83% AUC on Celeb-DF. This is a notable improvement. The face-pretrained ViT actually nearly matched M2TR. This suggests that pretraining on a large face dataset endowed the ViT with features that transfer better to unseen fakes. We hypothesize the Glint360K ViT learned to strongly represent facial landmarks and identity features, thus it can detect when something is off in a new face because the face's identity representation looks inconsistent or the facial features deviate from what a real face should be. In contrast, the ImageNet ViT might have been focusing on texture or background cues that don't hold up for Celeb-DF (which has diverse scenes and mostly authentic backgrounds).
- IIL model did not perform worse than baselines as in intra-dataset experiments: it achieved ∼84% AUC on Celeb-DF. This is better than Xception/ResNet and

ViT-ImageNet. However, it did not outperform the best models. Furthermore, we again realized that pretraining on Glint360K didn't improve performance compared to ImageNet-pretrained version. This make us suspect that its approach to eliminate identity related features may contradict with our efforts that making models to focus face-related features.

- ICT model achieved 85% AUC score when pretrained with Glint360K which is better than ImageNet-pretrained version which was 80%. Note that, similar to ViT, ICT which has vision transformer as backbone has improved strikingly. The inner architecture of ViT that make it able to detect global image context works well with our face related pretraining approach.
- LSC was second best model in intra-dataset experiments. However, the model fall behind from M2TR and Glint360K-pretrained ICT. M2TR's two branch design and ICT's transformer backbone caused that happen as we have seen similar pattern above.

For ForgeryNet test set, although not the primary focus since it's very large, we did test models on a ForgeryNet subset. We observed similar trends as in Celeb-DF-v2. The breadth of ForgeryNet (including GAN-generated face synthesis) challenged the models. Interestingly, some detectors triggered false positives on GAN-generated faces that aren't face-swaps. Models that look for unnatural face textures (possibly M2TR's frequency branch) did well in those cases. We include these results in the table for completeness. Overall, performance on ForgeryNet was slightly lower than on Celeb-DF for all models, reflecting the higher diversity and difficulty.

Our custom dataset was arguably the most unpredictable test, since it involves real-world videos and deepfakes created by tools not explicitly present in FF++. The detailed AUC scores of the models on our dataset shown in Table 4.6. The overall AUCs were moderately high, but with clear differences among models:

- LAA-Net and ICT were the best performing models (∼86% AUC each) on our dataset. LAA-Net's performance indicates that even for our high-quality fakes,

Table 4.6. Evaluation of methods on our custom dataset. Bold values show the best result for each method.

Method	**Backbone**	**AUC(%)**	
		ImageNet	**Glint360K**
XceptionNet [3]	Xception	75.12	**76.55**
IIL [15]	Efficientnet-b4	**80.73**	78.50
LAA-Net [16]	Efficientnet-b4	85.72	**86.69**
M2TR [17]	Efficientnet-b4	81.41	**82.81**
ICT [9]	Transformer	81.58	**85.83**
LSC [18]	Resnet-34	77.23	**78.34**
ResNet [6]	Resnet-34	75.19	**76.74**
ViT [7]	vit_tiny_p16	77.37	**81.58**

there were small vulnerable regions it could pick up (perhaps slight blurriness at cheek edges or inconsistencies in lighting on the face). ICT is also ,as expected, doing well aligns with the idea that it wasn't fooled by new identities; it looked for inconsistencies instead.

- M2TR was second-best, around 82% AUC on our data. It handled most swaps well, especially when the fake introduced high-frequency artifacts. However, for a few fakes that were extremely well blended, M2TR missed them, giving false negatives. We suspect the difference is that M2TR, not having explicit identity features, relied purely on artifact detection.
- ViT pretained on Glint360K was one of best performer with an AUC of about 81% it suggests that the face-pretrained transformer could robustly detect the fakes we created, even though these fakes were of very high quality and the videos had varied conditions. Upon inspection, the ViT-F model correctly flagged many fakes where others struggled, likely by focusing on subtle eye blinking issues or slight misalignment in face shape that our DeepFaceLab fakes had. The attention analysis later will show this model indeed focused on face regions.

- The standard Xception and ResNet dropped more significantly on our dataset (roughly 75–78% AUC). They often got confused by the fact that the fake and real frames looked very similar globally. For example, in some of our swaps, the background and even facial expressions were identical between real and fake – only the identity was swapped. Xception sometimes predicted those as real because the usual compression artifacts or face texture cues it learned from FF++ were absent.
- The IIL model performed decently (around 80% AUC) on our data – better than Xception, though not as high as ViT. It caught many obvious fakes but missed some. A common failure for IIL was when both source and target in the swap were not known to the model; in those cases, the artifact module needed to spot subtle cues which sometimes it didn't if the fake was flawless.

The Glint360K-pretrained models consistently outperformed their ImageNet version on cross-dataset tests. This confirms that face pretraining provides a strong advantage in generalization, presumably by biasing the models toward the facial region and learning features that remain relevant for any face. This is a novel finding in our study, as prior works primarily used ImageNet weights. Our qualitative analysis will further illustrate these differences.

4.5. Label Smoothing Analysis

To quantify the effect of label smoothing on model performance, we conducted experiments varying the smoothing parameter ϵ and measuring the resulting AUC on the validation set. Figure AUC vs. Label Smoothing ϵ 4.1 plots the change in AUC for three representative models (ResNet, ViT, and M2TR) as ϵ increases from 0 (no smoothing) to 0.2. Note that we haven't included all the models but the main trend is same for all. We summarize the trends observed:

- A small amount of label smoothing improved performance for most models. For example, going from $\epsilon = 0$ to $\epsilon = 0.1$ raised ResNet's FF++ AUC from 98.82%

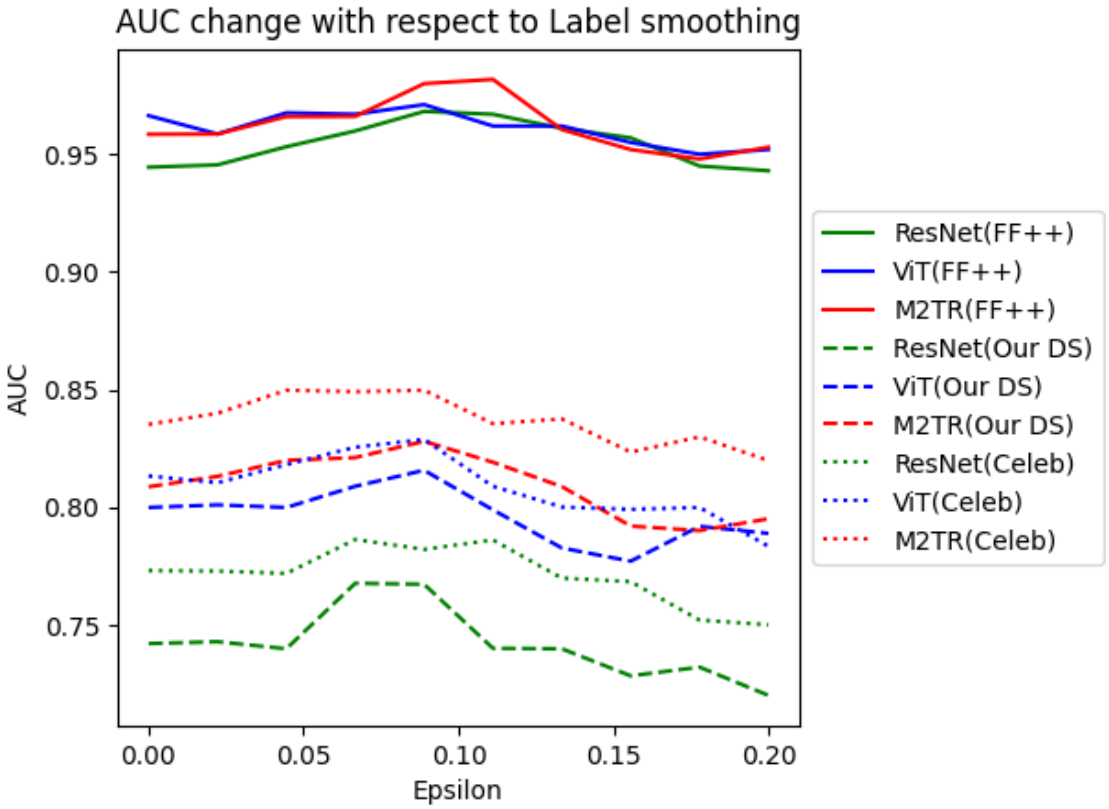


Figure 4.1. AUC change with respect to label smoothing epsilon value. For the purpose of clearity of the graph, 3 models selected, namely ResNet, ViT, M2TR.

to 94.5%, and its Celeb-DF AUC from 77% to 78.64%. This indicates better generalization, likely because the model doesn't over-confidently fit to the training labels/artifacts. The ViT model saw a similar bump (about +1.5 points AUC on Celeb-DF with $\epsilon = 0.1$)

- The benefit peaked around $\epsilon = 0.1$. Beyond that, as we increased smoothing to 0.2, the performance began to decline for some models. For instance, M2TR's AUC was highest at $\epsilon = 0.1$ but dropped slightly at $\epsilon = 0.2$. Too much smoothing (which means training targets of 0.2 for real and 0.8 for fake) might be under-training the model, making it uncertain even when it could be more confident. Essentially, there is a sweet spot: enough smoothing to regularize, but not so much to diminish the signal.

- Interestingly, we noted that label smoothing improved not only generalization but also the calibration of the outputs. Models trained with smoothing produced prediction probabilities that were less extreme (not saturating at 0 or 1 for most inputs). When we computed the calibration error, it was lower for $\epsilon = 0.1$ models. This is desirable in practice if the detector's output confidence is to be used (for example, to set a threshold that corresponds to a certain false-positive rate).

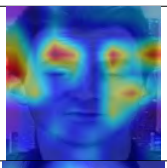
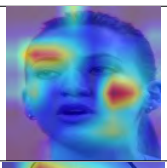
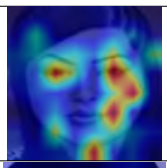
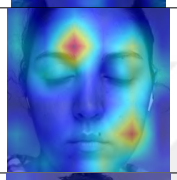
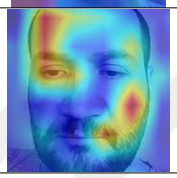
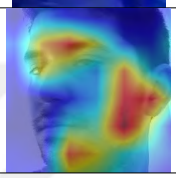
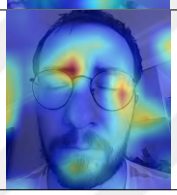
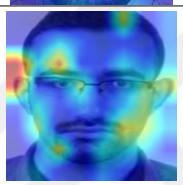
In conclusion, our analysis indicates that a moderate label smoothing (around 0.1) is a simple yet effective addition to deepfake detector training. It consistently improved AUC on unseen data in our experiments. The effect was most pronounced for models that have high capacity and risk overfitting (e.g., pure CNNs without special regularization). Therefore, we included $\epsilon = 0.1$ in our default training setup for all models as mentioned in section 3.4.1.

4.6. Qualitative Analysis

To complement the quantitative results, we performed a qualitative examination of model behaviors. In particular, we generated visual explanations (heatmaps) for the Vision Transformer models to understand where they focus on the input face, and how this differs between models pretrained on ImageNet versus on Glint360K (face data). Table 4.7 illustrates representative attention maps for an ImageNet-pretrained ViT and a Glint360K-pretrained ViT on real vs fake images. For creating these heatmaps, we used Grad-CAM and attention rollout techniques on the final transformer layers. We present the heatmaps overlaid on the face images. Several clear differences emerged:

- The ImageNet-pretrained ViT sometimes attends to the background or peripheral areas of the face. For example, in one deepfake frame, the ImageNet ViT had high activation not only on the face but also on the person's hair and part of the background. This model likely learned some context cues from FF++ training (perhaps compression noise in the background or a general difference in image sharpness) that correlated with fakeness in the training set. However, those cues

Table 4.7. Heatmap examples of vision transformer model pretrained with ImageNet and Glint360K. Rows contains datasets and columns contains results of 2 examples of ImageNet pretrained ViT and Glint360K pretrained ViT. Last colum especially draws heatmap of person who wear eyeglasses.

	ViT Pretrained with ImageNet		ViT Pretrained with Glint360K	
Celeb-DF-v2				
Our Dataset				
Our Dataset(Glasses)				

are not reliably present in other datasets.

- In contrast, the Glint360K-pretrained ViT had attention maps concentrated almost entirely on the facial region, especially the inner face (eyes, nose, mouth). In the heatmap visualization for a fake image, the Glint-pretrained model's highest attention was around the eyes. Especially, the regions where deepfake artifacts often occur (slight misalignment, unnatural eye blinking). Very little attention was given to the hair or background. For the corresponding real image, the Glint-ViT's attention was more uniformly spread across the face indicating a normal face has consistent features, so no single region stands out as "fake". Meanwhile, the ImageNet-ViT on the real image sometimes focused on edges of the face or corners.
- We interpret these findings as evidence that face-pretraining biases the model to use face-specific features. The Glint360K model "knows" what a real face should look like in terms of facial geometry and detail, so when a fake image is input, it tends to analyze the key facial features. Indeed, we saw multiple cases where the Glint-ViT flagged the eyes as the most suspicious region – a known area

for deepfakes . The ImageNet-ViT, lacking this prior, was comparatively more distracted by non-face elements.

- In the last column, we have shown the heatmaps of people who wear eyeglasses. Consistent with the finding found above, ViT pretrained with Glint360K qualitatively focuses face area more even though the person have face accessory since it learned from a diverse face dataset while ImageNet pretrained versions still lacks focusing on face.

In summary, the qualitative heatmap analysis underscores the advantage of face-aware pretraining and specialized models: they look at the right places (facial features) when deciding real vs fake. Models without that benefit may latch onto dataset-specific or context cues that don't generalize. These differences in model "attention" help explain the performance gaps observed in cross-dataset tests. They also provide interpretability – for critical applications, we prefer a model that bases its decision on the face itself (which is where the manipulation occurs) rather than on background oddities. The Glint360K-pretrained ViT satisfy this to a greater extent. This gives us confidence that these models are detecting forensic inconsistencies in the face rather than unrelated correlations.

4.7. Robustness Experiments

For the first of robustness tests, we recompressed video frames at quality factors Q = 100 (no compression), 50 (moderate compression), 30, and 10 (very heavy compression). All models saw a performance degradation as compression increased, since compression artifacts can obscure the subtle fake artifacts or even introduce confounding artifacts. However, the magnitude of AUC drop differed by model. The CNN-based models (Xception, ResNet) were the most affected: for instance, Xception's AUC dropped by ∼8 points going from Q100 to Q50, and at Q10 it dropped 20 points (from ∼97.35 to ∼77.35 AUC). This indicates Xception relied on high-frequency details that got destroyed by compression. In contrast, M2TR – which explicitly uses frequency information – was significantly more robust. At Q50, M2TR's AUC drop

was only ∼3 points, and even at Q10 it retained ∼84.33% AUC (compared to Xception's 77.35%). This fits M2TR's design aim of being compression-robust. Though ViT lost ∼5 AUC points at Q50 and ∼15 at Q10 (still better than Xception's 20), the transformers—ViT variants—also performed better than CNNs. Among CNNs, LAA-Net and IIL had some benefit: LAA-Net's emphasis on small areas might still catch artifact traces even in compressed images, and IIL's ADM looking for local artifacts may filter out global compression noise. Generally though, frequency-aware or transformer-based models were more resistant to JPEG artifacts. This implies that when video quality deteriorates, designing detectors using global context (like ViT) or multi-frequency analysis (like M2TR) can help sustain performance. Table 4.8 and Figure 4.2 summarizes these findings.

Table 4.8. Robustness of models under different JPEG compression quality factors (AUC %). Test is conducted on FF++ test set.

Model	**Q=100**	**Q=50**	**Q=30**	**Q=10**
Xception	97.35	89.43 (-7.92)	84.23 (-13.12)	77.35 (-20.00)
IIL	95.50	89.20 (-6.30)	84.08 (-11.42)	77.96 (-17.54)
LAA-Net	98.69	92.96 (-5.73)	88.49 (-10.20)	81.37 (-17.32)
M2TR	98.18	95.08 (-3.10)	91.97 (-6.21)	84.33 (-13.85)
ICT	96.88	89.53 (-7.35)	85.08 (-11.80)	77.90 (-18.98)
LSC	98.34	90.13 (-8.21)	84.70 (-13.64)	78.27 (-20.07)
Resnet	96.82	88.19 (-8.72)	83.02 (-13.80)	75.64 (-21.18)
ViT	97.11	91.89 (-5.22)	88.28 (-8.83)	81.86 (-15.25)

Additionally, we applied Gaussian blurring with kernel sizes $\sigma = 1, 2, 3$ (increasing blur). Blur smooths out edges and fine details, which can hide blending boundaries or color mismatches that detectors look for. Again, CNNs suffered more: at $\sigma = 2$, Xception's AUC fell by ∼10 points, and at $\sigma = 3$ (noticeably blurry) it fell ∼18 points. The ViT model showed more resilience – at $\sigma = 2$ it dropped only ∼7 points. This could be because transformers can rely on more global shape or configuration cues

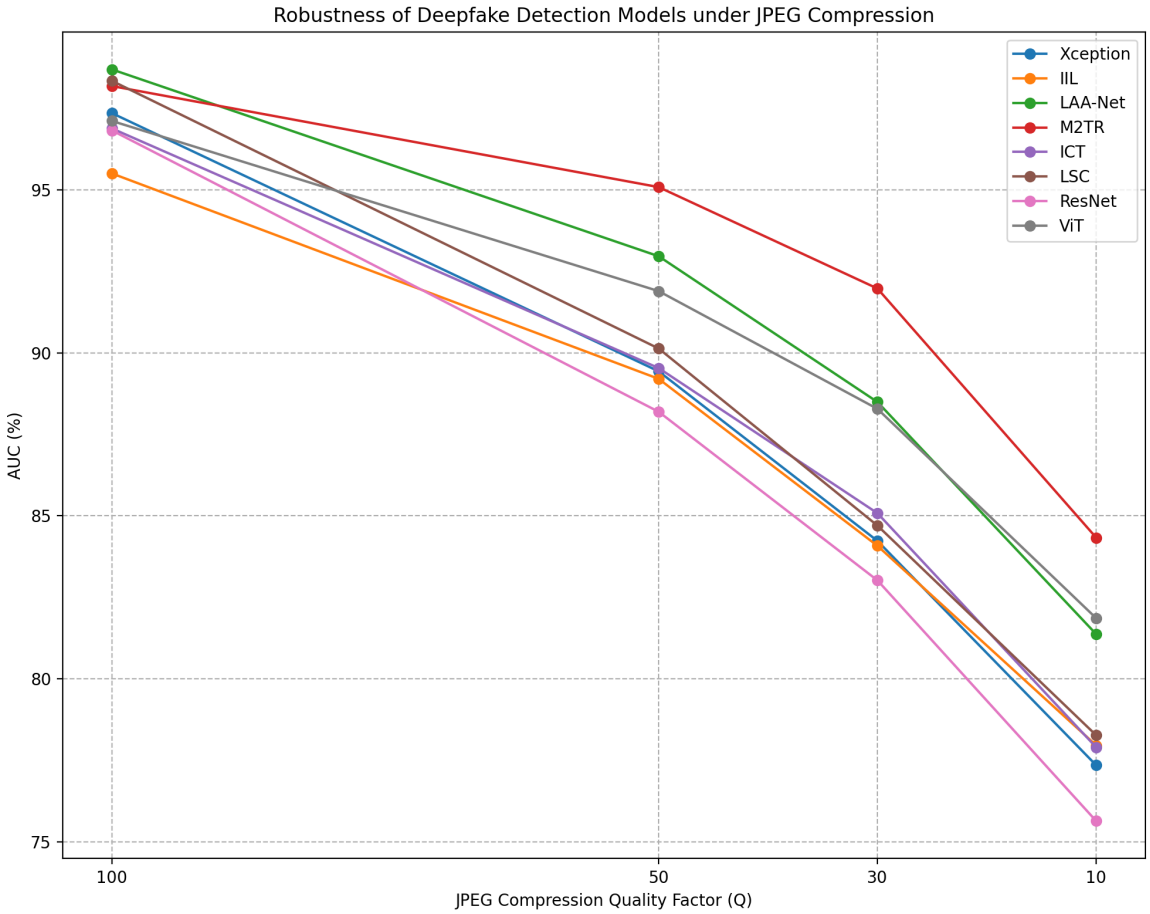


Figure 4.2. AUC change with respect to compression quality factor.

(which remain even if image is slightly blurry) rather than crisp texture. LSC model also handled blur relatively well, since the source feature inconsistency it checks might still be detectable in lower-frequency signals. M2TR, interestingly, lost some edge because its frequency branch that emphasizes high-frequency becomes less useful when the image is heavily blurred (high-frequency content is removed). At $\sigma = 3$, M2TR's performance dipped closer to that of Xception. So while M2TR was great for compression (which introduces artificial high-freq noise), it was less robust for blur (which removes high-freq content). Meanwhile, ViT-F remained on top under blur, suggesting face shape or identity anomalies can still be picked up under mild blur by a strong model. In summary, no model is immune to severe blur (at extreme blur all dropped to $\sim$50% AUC, essentially random guessing), but mild blur showed transformers $>$ CNNs in robustness. Table 4.9 and Figure 4.3 summarizes these findings.

Table 4.9. Robustness of models under Gaussian blur (AUC %). Test is conducted on FF++ test set.

Model	σ **= 0 (no blur)**	σ **= 1**	σ **= 2**	σ **= 3**
Xception	97.35	91.35 (-6.00)	87.27 (-10.08)	79.09 (-18.26)
IIL	95.50	90.78 (-4.72)	86.19 (-9.31)	79.07 (-16.43)
LAA-Net	98.69	94.47 (-4.22)	90.22 (-8.47)	82.49 (-16.20)
M2TR	98.18	94.04 (-4.14)	89.26 (-8.92)	81.23 (-16.95)
ICT	96.88	93.13 (-3.75)	89.77 (-7.11)	83.87 (-13.01)
LSC	98.34	94.14 (-4.20)	91.23 (-7.11)	85.53 (-12.81)
Resnet	96.82	90.49 (-6.33)	86.31 (-10.51)	78.32 (-18.50)
ViT	97.11	93.25 (-3.86)	90.26 (-6.85)	84.75 (-12.36)

Moreover, we carried out more tests to assess how well deepfake detection models responded to horizontally reflected images (i.e. mirrored images). Mirroring or flipping an image horizontally preserves general facial structures but could disturb models depending on asymmetric facial features or spatial consistency of artifacts. Note that we have included horizontal flip augmentation during the pre-training phase as it is one of the common augmentations that is used in face recognition tasks. Also, we have used the augmentation in finetuning phase too if otherwise not implied in the original paper of the work.

We have included each model's performance using both original test images and their horizontally mirrored versions in Table 4.10. The comparative results in terms of AUC scores are presented.

The results show how well horizontal flip augmentation works during pre-training and finetuning. Compared to horizontal flip tests on [60], for instance XceptionNet show greatly better results in our tests. We think that this advancement comes from our face domain pretraining strategy. Other than that, tested on mirrored images, all models kept good performance with little deterioration. Especially CNN-based models like Xception and Resnet showed noticeably lower sensitivity to horizontal flipping than

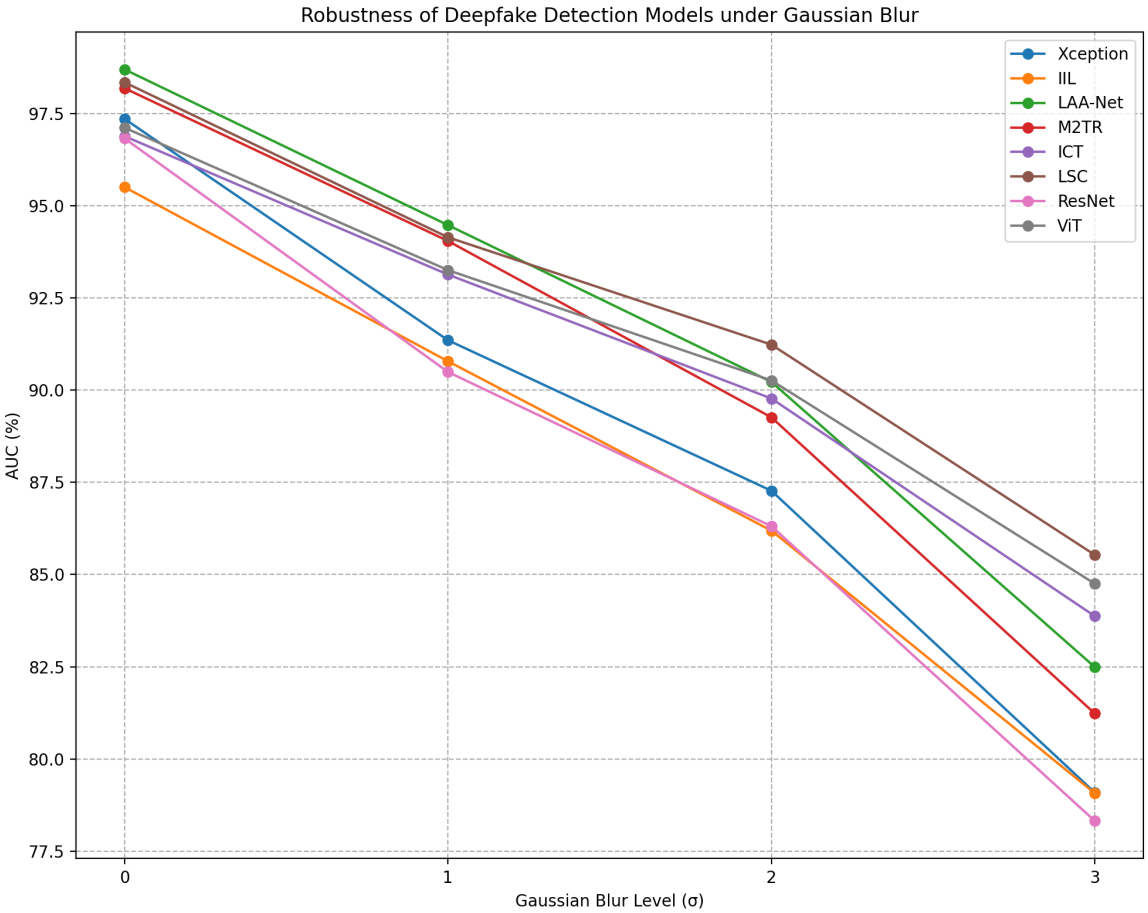


Figure 4.3. AUC change with respect to gaussian blur sigma value.

models taught without augmentation. Showing even more minor performance declines, transformer and attention-based models further validated their resilience. Emphasizing the advantage of global contextual modeling, ViT displayed the best resilience.

These results highlight generally the need and efficiency of using face-dataset while pretraining and horizontal flip augmentation as part of the training approach to improve the robustness of deepfake detection models to common image transformations.

4.8. Face Crop Margin Analysis

To investigate how input cropping margins influence deepfake detection we have selected our LAA-Net setup, since it performs best on most of our experiments. we

Table 4.10. Robustness of models under horizontal flip (AUC %). Test is conducted on FF++ test set.

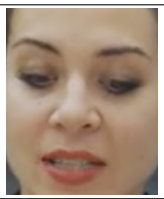
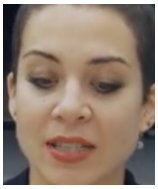
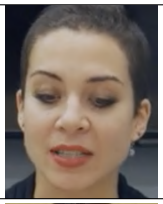
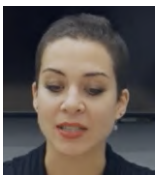
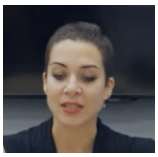
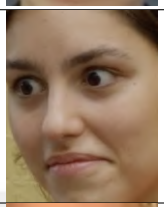
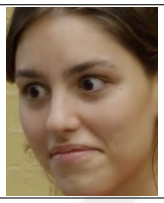
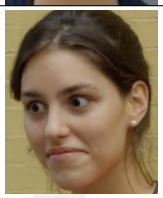
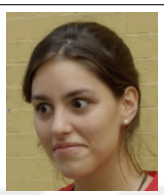
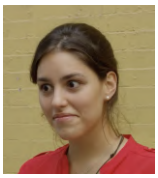
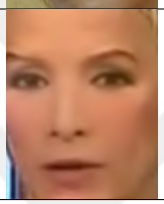
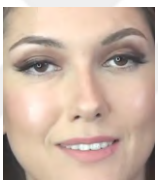
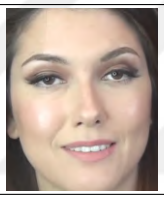
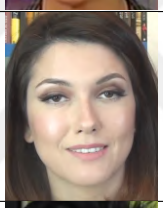
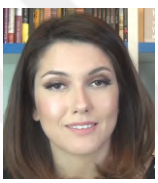
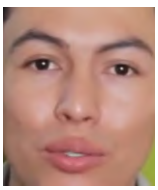
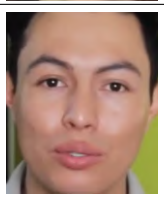
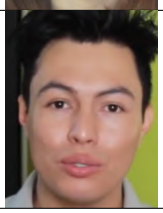
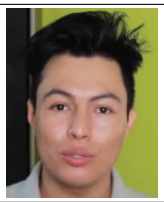
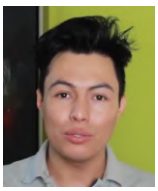
Model	**Original (No Flip)**	**Mirrored Images**	**Difference**
Xception	97.35	96.03	-1.32
IIL	95.50	94.89	-0.61
LAA-Net	98.69	98.29	-0.40
M2TR	98.18	97.73	-0.45
ICT	96.88	96.19	-0.69
LSC	98.34	97.87	-0.47
Resnet	96.82	95.06	-1.76
ViT	97.11	96.69	-0.42

conducted experiments varying the cropping margin from very tight (only face, no peripheral features) to very wide (including significant background). Specifically, five margin levels were tested, Examples of these different margin crop levels can be seen in Table 4.11.

- Level 1: Tight cropping, capturing strictly facial landmarks without peripheral features.
- Level 2: Moderate cropping, including minor peripheral regions such as hairline and ears.
- Level 3: Balanced cropping, including full peripheral features (hair, ears) and minimal background.
- Level 4: Wide cropping, adding significant background context.
- Level 5: Very wide cropping, heavily dominated by background, significantly reducing the relative size of the face.

The results amply show how much LAA-Net performance varies with the cropping margin. Performance is rather poor (92.82% AUC) at close cropping (Level 1) because missing vital peripheral cues that enable detect subtle boundary artifacts (e.g., merging

Table 4.11. Sample images for 5 different identity with varying margin crop levels.

	Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**
ID-1					
ID-2					
ID-3					
ID-4					
ID-5					

at hairline or ears). Expanding to moderate margins (Levels 2 and 3), we find a notable improvement; AUC reaches an optimal point at Level 3 (98.69%). This balanced margin helps the model to efficiently use peripheral facial artifacts without too much interference from unrelated background noise.

As we come to Levels 4 and 5 in which cropping margins widen to include moderate to heavy background context, the AUC score of the LAA-Net starts to decrease. Performance of the model declines to 97.25% at Level 4 which implies that the LAA-Net starts to consider background noisy areas more. At level 5, the performance falls sharply top 92.61%. this degradation shows that for the LAA-Net, appropriate preprocessing of the images are required. Since LAA-Net makes use of local artificial areas, this notable decrease in AUC scores most likely result from irrelevant background details that LAA-Net did not see during its training. This background noise overshadowing the subtle facial artifacts, so distracting the model and reducing the efficacy of her localized artifact attention mechanism.

These results, taken together, emphasize the need of carefully choosing input cropping margins for LAA-Net and other artifact-localization techniques. To maximize model performance, a balanced margin that which spans peripheral facial areas with some small background is advised.

5. CONCLUSION

In this study, we have proposed a extensive approach to improving deepfake detection by showing a new pretraining approach. We have also introduced a new custom benchmark dataset, evaluating the effects of pretraining datasets, and utilizing label smoothing to enhance model generalization on unseen datasets. Our results shows the importance of domain-specific pretraining, with models initialized on face-specific datasets like Glint360K outperforming ImageNet-pretrained models in both intra-dataset and cross-dataset evaluations in deepfake detection task. This demonstrates the value of utilizing domain relevant features for robust deepfake detection.

We also showed that label smoothing effectively reduces overfitting and improves model performance on unseen data. However, we also found that excessive smoothing can make model to stop learning any better evet causes to decrease in performance. Therefore, the need for finding the balance is another significant need. We have also showed that while models show top performances after pretraining with face-specific dataset, qualitative results further enhanced our hypothesis by focusing on faces, not background or noise.

In addition to these, our dataset also offers a more realistic benchmark that includes diverse conditions, helps us to test the strengths and limitations of current state-of-the-art deepfake detection methods.

Future studies may extend our findings by experimenting advanced pretraining strategies using newer face-specific datasets, such as WebFace260M [61]. Moreover, additional robustness experiments may conduct such as JPEG compression in images, adding Gaussian Blur or Additive Noise etc.

To conclude that, we have proposed novel ways to secure media authenticity containing human faces, specifically for deepfakes, by introducing pretraining strategies.

We have also generated and presented new dataset for deepfake detection task to provide a more comprehensive and detailed benchmark. We believe that, our findings are going to pave the way for future research including advanced pretraining strategies, overfitting prevention, and more reliable and scalable deepfake detection systems.

REFERENCES

1. Keiron, O. and N. Ryan, "An Introduction to Convolutional Neural Networks", *arXiv Preprint arXiv:1511.08458*, 2015.

2. Goodfellow, I., 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.

3. Rossler, A., D. Cozzolino, L. Verdoliva, C. Riess, J. Thies and M. Niessner, "FaceForensics++: Learning to Detect Manipulated Facial Images", *International Conference on Computer Vision*, pp. 1–11, Seoul, South Korea, 2019.

4. Chollet, F., "Xception: Deep Learning with Depthwise Separable Convolutions", *Conference on Computer Vision and Pattern Recognition*, pp. 1800–1807, Honolulu, HI, USA, 2017.

5. Mingxing, T. and L. Quoc, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", *International Conference on Machine Learning*, pp. 6105–6114, PMLR, Long Beach, CA, USA, 2019.

6. He, K., X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", *Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, 2016.

7. Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold and S. Gelly, "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale", *arXiv Preprint arXiv:2010.11929*, 2020.

8. Nguyen, H., J. Yamagishi and I. Echizen, "Use of a Capsule Network to Detect Fake Images and Videos", *arXiv Preprint arXiv:1910.12467*, 2019.

9. Dong, X., J. Bao, D. Chen, T. Zhang, W. Zhang, Y. Nenghai, C. Dong, W. Fang and G. Baining, "Protecting Celebrities from Deepfake with Identity Consistency Transformer", *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 9468–9478, New Orleans, LA, USA, 2022.

10. Zheng, Y., J. Bao, D. Chen, M. Zeng and F. Wen, "Exploring Temporal Coherence for More General Video Face Forgery Detection", *Proceedings of the International Conference on Computer Vision*, pp. 15044–15054, Montreal, Canada, 2021.

11. Seow, J. W., M. K. Lim, R. C.-W. Phan and J. K. Liu, "A Comprehensive Overview of Deepfake: Generation, Detection, Datasets, and Opportunities", *Neurocomputing*, Vol. 513, pp. 351–371, 2022.

12. Li, Y., X. Yang, P. Sun, H. Qi and S. Lyu, "Celeb-DF: A Large-Scale Challenging Dataset for Deepfake Forensics", *Conference on Computer Vision and Pattern Recognition*, pp. 3207–3216, Seattle, WA, USA, 2020.

13. Tolosana, R., S. Romero-Tapiador, R. Vera-Rodriguez, E. Gonzalez-Sosa and J. Fierrez, "DeepFakes Detection Across Generations: Analysis of Facial Regions, Fusion, and Performance Evaluation", *Engineering Applications of Artificial Intelligence*, Vol. 110, p. 104673, 2022.

14. An, X., X. Zhu, Y. Gao, Y. Xiao, Y. Zhao, Z. Feng, L. Wu, B. Qin, M. Zhang, D. Zhang and Y. Fu, "Partial FC: Training 10 Million Identities on a Single Machine", *International Conference on Computer Vision Workshops*, pp. 1445–1449, Montreal, QC, Canada, 2021.

15. Dong, S., J. Wang, R. Ji, J. Liang, H. Fan and Z. Ge, "Implicit Identity Leakage: The Stumbling Block to Improving Deepfake Detection Generalization", *Conference on Computer Vision and Pattern Recognition*, pp. 3994–4004, Vancouver, BC, Canada, 2023.

16. Nguyen, D., N. Mejri, I. P. Singh, P. Kuleshova, M. Astrid, A. Kacem, E. Ghorbel and D. Aouada, "LAA-Net: Localized Artifact Attention Network for Quality-Agnostic and Generalizable Deepfake Detection", *Conference on Computer Vision and Pattern Recognition*, pp. 17395–17405, Seattle, WA, USA, 2024.

17. Wang, J., Z. Wu, J. Chen and Y. Jiang, "M2TR: Multi-Modal Multi-Scale Transformers for Deepfake Detection", *Proceedings of the ACM International Conference on Multimedia Retrieval*, pp. 615–623, Newark, NJ, USA, 2022.

18. Zhao, T., X. Xu, M. Xu, H. Ding, Y. Xiong and W. Xia, "Learning Self-Consistency for Deepfake Detection", *International Conference on Computer Vision*, pp. 15003–15013, Virtual, 2021.

19. Deng, J., W. Dong, R. Socher, L. Li, L. Kai and F. Li, "ImageNet: A Large-Scale Hierarchical Image Database", *Conference on Computer Vision and Pattern Recognition*, pp. 248–255, Miami Beach, FL, USA, 2009.

20. He, Y., B. Gan, S. Chen, Y. Zhou, G. Yin, L. Song, L. Sheng, J. Shao and Z. Liu, "Forgerynet: A Versatile Benchmark for Comprehensive Forgery Analysis", *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 4360–4369, Nashville, TN, USA, 2021.

21. Li, L., J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen and B. Guo, "Face X-Ray for More General Face Forgery Detection", *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 5001–5010, Seattle, WA, USA, 2020.

22. Luo, Y., Y. Zhang, J. Yan and W. Liu, "Generalizing Face Forgery Detection with High-Frequency Features", *Conference on Computer Vision and Pattern Recognition*, pp. 16312–16321, Virtual, 2021.

23. Jeong, Y., D. Kim, Y. Ro and J. Choi, "Frepgan: Robust Deepfake Detection Using Frequency-Level Perturbations", *Proceedings of the AAAI Conference on Artificial*

Intelligence, Vol. 36, pp. 1060–1068, Virtual, 2022.

24. Masood, M., M. Nawaz, K. M. Malik, A. Javed, A. Irtaza and H. Malik, "Deepfakes Generation and Detection: State-of-the-Art, Open Challenges, Countermeasures, and Way Forward", *Applied Intelligence*, Vol. 53, No. 4, pp. 3974–4026, 2023.

25. Giudice, O., L. Guarnera and S. Battiato, "Fighting Deepfakes by Detecting GAN DCT Anomalies", *Journal of Imaging*, Vol. 7, No. 8, p. 128, 2021.

26. Zhuang, W., Q. Chu, Z. Tan, Q. Liu, H. Yuan, C. Miao, Z. Luo and N. Yu, "UIA-ViT: Unsupervised Inconsistency-Aware Method Based on Vision Transformer for Face Forgery Detection", *European Conference on Computer Vision*, pp. 391–407, Tel Aviv, Israel, 2022.

27. Shiohara, K. and T. Yamasaki, "Detecting Deepfakes with Self-Blended Images", *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 18720–18729, New Orleans, LA, USA, 2022.

28. Choi, J., T. Kim, Y. Jeong, S. Baek and J. Choi, "Exploiting Style Latent Flows for Generalizing Deepfake Video Detection", *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 1133–1143, Seattle, WA, USA, 2024.

29. Yang, X., Y. Li and S. Lyu, "Exposing Deep Fakes Using Inconsistent Head Poses", *International Conference on Acoustics, Speech and Signal Processing*, pp. 8261–8265, Brighton, UK, 2019.

30. Li, Y., M. Chang and S. Lyu, "In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking", *arXiv Preprint arXiv:1806.02877*, 2018.

31. Sabir, E., J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi and P. Natarajan, "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos", *Interfaces*, Vol. 3, No. 1, pp. 80–87, 2019.

32. Agarwal, S., H. Farid, T. El-Gaaly and S.-N. Lim, "Detecting Deep-Fake Videos from Appearance and Behavior", *International Workshop on Information Forensics and Security*, pp. 1–6, Virtual, 2020.

33. Khan, S. A. and H. Dai, "Video Transformer for Deepfake Detection with Incremental Learning", *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1821–1828, Chengdu, China, 2021.

34. Gu, Z., Y. Chen, T. Yao, S. Ding, J. Li and L. Ma, "Delving into the Local: Dynamic Inconsistency Learning for DeepFake Video Detection", *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36, pp. 744–752, 2022.

35. Gu, Z., T. Yao, Y. Chen, S. Ding and L. Ma, "Hierarchical Contrastive Inconsistency Learning for Deepfake Video Detection", *European Conference on Computer Vision*, pp. 596–613, Tel Aviv, Israel, 2022.

36. Xu, Y., J. Liang, G. Jia, Z. Yang, Y. Zhang and R. He, "Tall: Thumbnail Layout for Deepfake Video Detection", *Proceedings of the International Conference on Computer Vision*, pp. 22658–22668, Paris, France, 2023.

37. Hu, J., X. Liao, J. Liang, W. Zhou and Z. Qin, "FInfer: Frame Inference-Based Deepfake Detection for High-Visual-Quality Videos", *Proceedings of the Conference on Artificial Intelligence*, Vol. 36, pp. 951–959, 2022.

38. Liu, B., B. Liu, M. Ding, T. Zhu and X. Yu, "TI2Net: Temporal Identity Inconsistency Network for Deepfake Detection", *Winter Conference on Applications of Computer Vision*, pp. 4691–4700, Waikoloa, HI, USA, 2023.

39. Cozzolino, D., A. Rössler, J. Thies, M. Nießner and L. Verdoliva, "ID-Reveal: Identity-Aware DeepFake Video Detection", *Proceedings of the International Conference on Computer Vision*, pp. 15108–15117, Montreal, Canada, 2021.

40. Chu, B., W. You, Z. Yang, L. Zhou and R. Wang, "Protecting World Leader Using

Facial Speaking Pattern Against Deepfakes", *Signal Processing Letters*, Vol. 29, pp. 2078–2082, 2022.

41. Qian, Y., G. Yin, L. Sheng, Z. Chen and J. Shao, "Thinking in Frequency: Face Forgery Detection by Mining Frequency-Aware Clues", *European Conference on Computer Vision*, pp. 86–103, Glasgow, UK, 2020.

42. Chen, P., J. Liu, T. Liang, G. Zhou, H. Gao, J. Dai and J. Han, "FSSPOTTER: Spotting Face-Swapped Video by Spatial and Temporal Clues", *International Conference on Multimedia and Expo*, pp. 1–6, London, UK, 2020.

43. Guillaro, F., D. Cozzolino, A. Sud, N. Dufour and L. Verdoliva, "Trufor: Leveraging All-Round Clues for Trustworthy Image Forgery Detection and Localization", *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 20606–20615, Vancouver, BC, Canada, 2023.

44. Lomnitz, M., Z. Hampel-Arias, V. Sandesara and S. Hu, "Multimodal Approach for DeepFake Detection", *Applied Imagery Pattern Recognition Workshop*, pp. 1–9, Virtual, 2020.

45. Xiaoyi, D., J. Bao, D. Chen, W. Zhang, N. Yu, D. Chen, F. Wen and B. Guo, "Identity-Driven DeepFake Detection", *arXiv Preprint arXiv:2012.03930*, 2020.

46. Harry, C., G. Yangyang, W. Tianyi, L. Qi, C. Xiaojun and N. Liqiang, "Voice-Face Homogeneity Tells Deepfake", *arXiv Preprint arXiv:2203.02195*, 2022.

47. Ciftci, U. A., I. Demir and L. Yin, "FakeCatcher: Detection of Synthetic Portrait Videos Using Biological Signals", *Transactions on Pattern Analysis and Machine Intelligence*, Vol. PP, pp. 1–1, 2020.

48. Pavel, K. and M. Sebastien, "DeepFakes: A New Threat to Face Recognition? Assessment and Detection", *arXiv Preprint arXiv:1812.08685*, 2018.

49. Brian, D., B. Joanna, P. Ben, L. Jikuo, H. Russ, W. Menglin and C. F. Cristian, "The DeepFake Detection Challenge Dataset", *arXiv Preprint arXiv:2006.07397*, 2020.

50. Liming, J., L. Ren, W. Wayne, Q. Chen and C. L. Chen, "DeeperForensics-1.0: A Large-Scale Dataset for Real-World Face Forgery Detection", *arXiv Preprint arXiv:2001.03024*, 2020.

51. Bojia, Z., C. Minghao, C. Jingjing, M. Xingjun and J. Yu-Gang, "WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection", *arXiv Preprint arXiv:2101.01456*, 2020.

52. Hasam, K., T. Shahroz, K. Minha and S. W. Simon, "FakeAVCeleb: A Novel Audio-Video Multimodal Deepfake Dataset", *arXiv Preprint arXiv:2108.05080*, 2022.

53. Zhang, K., Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks", *Signal Processing Letters*, Vol. 23, No. 10, p. 1499–1503, 2016.

54. Ivan, P., G. Daiheng, C. Nikolay, K. Kunlin, M. Sugasa, U. Chris, S. F. Carl, R. Luis, J. Jian, Z. Sheng, W. Pingyu, Z. Bo and Z. Weiming, "DeepFaceLab: Integrated, Flexible and Extensible Face-Swapping Framework", *arXiv Preprint arXiv:2005.05535*, 2021.

55. Siarohin, A., S. Lathuilière, S. Tulyakov, E. Ricci and N. Sebe, "First Order Motion Model for Image Animation", *arXiv Preprint arXiv:2003.00196*, 2020.

56. Dong, S., J. Wang, R. Ji, J. Liang, H. Fan and Z. Ge, "Towards a Robust Deepfake Detector: Common Artifact Deepfake Detection Model", *arXiv Preprint arXiv:2210.14457*, 2022.

57. Guoxuan, X., L. Olivier, F. Gianni and B. Christos-Savvas, "Towards Understand-

ing Why Label Smoothing Degrades Selective Classification and How to Fix It", *arXiv Preprint arXiv:2403.14715*, 2025.

58. Deng, J., J. Guo, J. Yang, N. Xue, I. Kotsia and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition", *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, Long Beach, CA, USA, 2019.

59. Wang, F., "Face Swap via Diffusion Model", *arXiv Preprint arXiv:2403.01108*, 2024.

60. Yuhang, L. and E. Touradj, "Impact of Video Processing Operations in Deepfake Detection", *arXiv Preprint arXiv:2303.17247*, 2023.

61. Zhu, Z., G. Huang, J. Deng, Y. Ye, J. Huang, X. Chen, J. Zhu, T. Yang, J. Lu and D. Du, "Webface260m: A Benchmark Unveiling the Power of Million-Scale Deep Face Recognition", *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 10492–10502, Nashville, TN, USA, 2021.

62. Zheng, Y., J. Bao, D. Chen, M. Zeng and F. Wen, "Exploring Temporal Coherence for More General Video Face Forgery Detection", *International Conference on Computer Vision*, pp. 15044–15054, Virtual, 2021.

63. Cao, H., D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova and R. Verma, "Crema-D: Crowd-Sourced Emotional Multimodal Actors Dataset", *Transactions on Affective Computing*, Vol. 5, No. 4, pp. 377–390, 2014.

64. Livingstone, S. R. and F. A. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song: A Dynamic, Multimodal Set of Facial and Vocal Expressions in North American English", *PLoS One*, Vol. 13, No. 5, p. e0196391, 2018.

65. Chung, J. S., A. Nagrani and A. Zisserman, "Voxceleb2: Deep Speaker Recogni-

tion", *arXiv Preprint arXiv:1806.05622*, 2018.

66. Ephrat, A., I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman and M. Rubinstein, "Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation", *arXiv Preprint arXiv:1804.03619*, 2018.

67. Rana, M. S. and A. H. Sung, "DeepfakeStack: A Deep Ensemble-Based Learning Technique for Deepfake Detection", *International Conference on Cyber Security and Cloud Computing/6th International Conference on Edge Computing and Scalable Cloud*, pp. 70–75, New York, NY, USA, 2020.

68. Chintha, A., B. Thai, S. J. Sohrawardi, K. Bhatt, A. Hickerson, M. Wright and R. Ptucha, "Recurrent Convolutional Structures for Audio Spoof and Video Deepfake Detection", *Journal of Selected Topics in Signal Processing*, Vol. 14, No. 5, pp. 1024–1037, 2020.

69. Sabour, S., N. Frosst and G. E. Hinton, "Dynamic Routing Between Capsules", *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3859–3869, Long Beach, California, USA, 2017.

70. Nguyen, H. H., J. Yamagishi and I. Echizen, "Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos", *International Conference on Acoustics, Speech and Signal Processing*, pp. 2307–2311, IEEE, Brighton, UK, 2019.

71. Khalil, S. S., S. Youssef and S. Saleh, "ICaps-Dfake: An Integrated Capsule-Based Model for Deepfake Image and Video Detection", *Future Internet*, Vol. 13, No. 4, p. 93, 2021.

72. Guo, Y., L. Zhang, Y. Hu, X. He and J. Gao, "MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition", *European Conference on Computer*

Vision, Vol. 9907, pp. 87–102, Amsterdam, Netherlands, 2016.

73. Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", *Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, Las Vegas, NV, USA, 2016.

74. Müller, R., A. Kornblith and G. Hinton, "When Does Label Smoothing Help?", *arXiv Preprint arXiv:1906.02629*, 2020.

75. Ezeakunne, U., C. Eze and X. Liu, "Data-Driven Fairness Generalization for Deepfake Detection", *arXiv Preprint arXiv:2412.16428*, 2024.

tion", *arXiv Preprint arXiv:1806.05622*, 2018.

66. Ephrat, A., I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman and M. Rubinstein, "Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation", *arXiv Preprint arXiv:1804.03619*, 2018.

67. Rana, M. S. and A. H. Sung, "DeepfakeStack: A Deep Ensemble-Based Learning Technique for Deepfake Detection", *International Conference on Cyber Security and Cloud Computing/6th International Conference on Edge Computing and Scalable Cloud*, pp. 70–75, New York, NY, USA, 2020.

68. Chintha, A., B. Thai, S. J. Sohrawardi, K. Bhatt, A. Hickerson, M. Wright and R. Ptucha, "Recurrent Convolutional Structures for Audio Spoof and Video Deepfake Detection", *Journal of Selected Topics in Signal Processing*, Vol. 14, No. 5, pp. 1024–1037, 2020.

69. Sabour, S., N. Frosst and G. E. Hinton, "Dynamic Routing Between Capsules", *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3859–3869, Long Beach, California, USA, 2017.

70. Nguyen, H. H., J. Yamagishi and I. Echizen, "Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos", *International Conference on Acoustics, Speech and Signal Processing*, pp. 2307–2311, Brighton, UK, 2019.

71. Khalil, S. S., S. Youssef and S. Saleh, "ICaps-Dfake: An Integrated Capsule-Based Model for Deepfake Image and Video Detection", *Future Internet*, Vol. 13, No. 4, p. 93, 2021.

72. Guo, Y., L. Zhang, Y. Hu, X. He and J. Gao, "MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition", *European Conference on Computer Vision*, Vol. 9907, pp. 87–102, Amsterdam, Netherlands, 2016.

73. Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", *Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, Las Vegas, NV, USA, 2016.

74. Müller, R., A. Kornblith and G. Hinton, "When Does Label Smoothing Help?", *arXiv Preprint arXiv:1906.02629*, 2020.

75. Ezeakunne, U., C. Eze and X. Liu, "Data-Driven Fairness Generalization for Deepfake Detection", *arXiv Preprint arXiv:2412.16428*, 2024.