DEVELOPMENT OF FRAMEWORK FOR DEEPFAKE DETECTION IN MULTIMEDIA DATA

Thesis submitted
in the Partial Fulfilment of the Requirements for the
Degree of

DOCTOR OF PHILOSOPHY

by

Deepak Dagar
(2K20/PHDIT/03)

Under the Supervision of
Prof. DINESH KUMAR VISHWAKARMA
Delhi Technological University

Department of Information Technology

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India
September, 2024

ACKNOWLEDGEMENTS

I deeply appreciate my highly regarded PhD mentor, **Prof. Dinesh Kumar Vishwakarma**, whose superb mentorship and unflinching assistance have played a crucial role in the successful completion of this thesis. I have been encouraged by his exceptional discipline, unwavering focus, and tireless work ethic, which have established a remarkable benchmark for academic accomplishment.

A man's family is an indispensable support system that works tirelessly behind the scenes to assist him in pursuing his aspirations. I thank my parents, **Mr. Balwan Singh** and **Mrs. Bhanwati Devi**, for exemplifying the epitome of exceptional parenting. I express my gratitude to my wife, **Komal**, for her steadfast support and for granting me the necessary freedom to concentrate on my Ph.D. I would want to express my gratitude to my family members, including my **brother, sister-in-law, and sisters,** for their generous affection and unwavering support.

Finishing PhD is a highly challenging journey, and the seniors who helped navigate this path need a special mention. To this end, I am grateful for the support of my PhD seniors, **Dr. Ankit Yadav.**

As I went through the most challenging phase of my PhD, my juniors ensured that I never gave up and always came back stronger. I am grateful to **Anusha Chhabra, Ananya Pandey, Ashish Bajaj, Abhishek Verma, and Bhavana Verma** for all the light-hearted conversations.

I extend my heartfelt appreciation to the state-of-the-art research lab established by my supervisor. Equipped with cutting-edge NVIDIA GPUs, it was pivotal in facilitating the success of the computationally expensive deep learning-based research experiments throughout my PhD.

Last but not least, I thank God for giving me the persistence and strength to show up at my lab each day and work through the ups and downs of this PhD journey.

Deepak Dagar

2K20/PHDIT/03

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE'S DECLARATION

I Deepak Dagar hereby certify that the work which is being presented in the thesis entitled "Development of Framework for Deepfake detection in Multimedia Data" in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy, submitted in the Department of Information Technology, Delhi Technological University is an authentic record of my own work carried out during the period from 26/08/2020 to 04/09/2024 under the supervision of Prof. Dinesh Kumar Vishwakarma.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate's Signature

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE BY THE SUPERVISOR

Certified that Deepak Dagar (2K20/PHDIT/03) has carried out their research work presented in this thesis entitled "Development of Framework for Deepfake detection in Multimedia Data" for the award of Doctor of Philosophy from the Department of Information Technology, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by the student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Signature:

Name of the Supervisor: Prof. Dinesh Kumar Vishwakarma

Designation: Professor (Information Technology)

Address: Rohini, Delhi

Date: 04/09/2024

ABSTRACT

In recent years, the development of deep learning methods, particularly Generative Adversarial Networks (GANs) and Variational Auto-encoders (VAEs), has resulted in fabricated content that is more realistic and credible to the human eye. Deepfake is an emergent deep learning technology that enables the production of synthetic content that is both highly realistic and credible. On the one hand, Deepfake has facilitated the development of cutting-edge applications in a variety of industries, including advertising, creative arts, and film productions. Conversely, it presents a threat to a variety of Multimedia Information Retrieval Systems (MIPR), including speech and face recognition systems and has more significant societal implications in the dissemination of misleading information. This thesis highlights the importance of developing strong systems that can identify potentially harmful changes in deepfake multimedia content by harnessing the capabilities of deep learning algorithms. The objective of this study is to employ the potential of deep learning to effectively identify and mitigate several types of deepfake manipulations, which pose a significant threat to individuals, society, nations, and enterprises together. The proposed detection methods, which utilize deep learning, aim to guarantee the dependability and precision of deepfake manipulation content, considering that social media platforms are the primary means of exchanging information. Consequently, this will improve the development of a digital ecosystem characterized by greater dependability and trustworthiness. This thesis addresses the difficulty of detecting deepfake manipulation by introducing four innovative deep-learning architectures and a unique collection of diverse manipulation videos that facilitates the training of deepfake detection models.

The first two models, namely Tex-ViT and Tex-Net focuses on the issue of deepfake manipulation detection. Deepfake manipulations can be misused in a variety of ways, pose a significant threat to individuals, society, nations, and enterprises together. Both Tex-ViT and Tex-Net uses texture as a feature and cross-attention mechanism to learn powerful representation of features. Tex-ViT uses gram matrices for texture feature representation while Tex-Net uses combination of Gram matrices and Local Binary Patterns for texture feature representation. Rest of the architecture is same in both the model, where the model combines traditional ResNet characteristics with a texture module that operates concurrently on ResNet segments before each down sampling process. This module serves as an input to the dual branch of the cross-attention vision transformer which uses them for final classification. The model's generalizability is illustrated through experimentation on a variety of categories of FF++ and

GAN dataset images in cross-domain contexts. The investigations were conducted using the Celeb-DF, FF++, and DFDCPreview datasets, which were subjected to a variety of post-processing techniques, including compression, noise addition, and blurring. The experimental results demonstrate that the proposed models outperform the current state-of-the-art approaches.

Next, proposed a diverse manipulation deepfake video dataset named Div-DF in assisting the training of various detection methods. The dataset consists of 150 authentic videos featuring various celebrities from different fields, as well as 250 deepfake videos. The deepfake videos include 100 face-swap videos, 100 facial reenactment videos, and 50 lip-sync videos. Deepfake video are created by combining the Face-Swap GAN (FSGAN) and the Wav2Lip approach, which are advanced techniques. Third models for deepfake video detection approach integrates Xception and LSTM pretrained models with channel and spatial attention mechanisms (CBAM). The Xception model employs depthwise separable convolution to capture latent spatial artifacts, while the LSTM model captures the distinctions between the modified sequences. The hybrid model assembly enables the acquisition of knowledge on spatial and temporal distortions across multiple dimensions, making it a powerful tool for identifying deepfake content. The model was tested on the proposed dataset, demonstrating its improved extraction capabilities.

Lastly, proposed a deepfake manipulation localization method is proposed. It is a dual-branch model that is propelled by the attention mechanism and combines handcrafted feature noise and CNNs as an encoder-decoder (ED). This dual-branch model employs noise features on one branch and RGB on the other before feeding to an ED architecture for semantic learning and skip connection deployment to retain spatial information. Additionally, this architecture employs channel spatial attention to enhance and refine the representation of the features. Extensive experimentation was conducted on the shallowfakes dataset (CASIA, COVERAGE, COLUMBIA, NIST16) and the deepfake dataset Faceforensics++ (FF++) to showcase the superior feature extraction capabilities and performance compared to a variety of baseline models, with an AUC score that exceeded 99%. The model is comparatively lighter, with 38 million parameters, and significantly surpasses existing State-of-the-Art (SoTA) models.

LIST OF PUBLICATIONS

Publications Arising from Research Work in the Thesis

SCIE Journal Papers

- **D. Dagar** and D. K. Vishwakarma, "A literature review and perspectives in deepfakes: generation, detection and applications" ***International Journal of Multimedia Information Retrieval***, vol. 11, June. 2022, doi: https://doi.org/10.1007/s13735-022-00241-w.
- **D. Dagar** and D. K. Vishwakarma, "Tex-ViT: A Generalizable, Robust, Texture-based dual-branch cross-attention deepfake detector," Under Review in ***Journal of Information Security and Applications*** (Pub: Elsevier), https://arxiv.org/abs/2408.16892
- **D. Dagar** and D. K. Vishwakarma, "Tex-Net: Texture-based parallel branch cross-attention generalized robust deepfake detector" vol 30, article number 233, ***Multimedia Systems***, 2024 (Pub: Springer), doi: https://doi.org/10.1007/s00530-024-01424-7
- **D. Dagar** and D. K. Vishwakarma, "Shallowfake and Deepfake Image Manipulation Localization using Noise and RGB-based dual branch method" ***Signal, Image and Video Processing***, vol 18, pages 7065-7077, 2024, doi: https://doi.org/10.1007/s11760-024-03376-x

Conference Papers

- **D. Dagar** and D. K. Vishwakarma "Div-Df: A Diverse Manipulation Deepfake Video Dataset" ***IEEE Conference: Global Conference on Information Technologies and Communications(GCITC)***,Bengaluru. (2023), doi: 10.1109/GCITC60406.2023.10426446.
- **D. Dagar** and D. K. Vishwakarma **"**A Hybrid Xception-LSTM model with channel and Spatial Attention for Deepfake Video Detection**"** ***IEEE Conference: International Conference on Mobile Networks and Wireless Communications,*** Tumakur, Karnataka. (2023), doi: 10.1109/ICMNWC60182.2023.10435983.

Publications Arising from Research Work Outside the Thesis

SCIE Journal Papers

1. **D. Dagar** and D. K. Vishwakarma, “A Noise and Edge extraction based dual-branch method for Shallowfake and Deepfake Localization” Under Review in ***Signal, Image and Video Processing,*** **2024**. https://arxiv.org/abs/2409.00896

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Multimedia data refers to the transmission of the data in one or more medium like text, image, video and audio. Recently, due to the progress in deep learning techniques, particularly Generative Adversarial Networks (GANs) and Variational Auto-encoders (VAEs), artificially created content has become highly realistic and convincing to human observers. Multimedia data refers to the transmission of the data in one or more medium. Deepfake is an emerging technology that enables the production of extremely realistic content. Deepfake technology has facilitated the development of sophisticated applications in diverse domains like as advertising, creative arts, and film production. However, it presents a danger to other Multimedia Information Retrieval Systems (MIPR) including facial identification and speech recognition systems, and has more substantial societal consequences in disseminating deceptive information. The initial investigation of identifying malicious tampering in multimedia content. The progress of the technology requires the development of effective methods to detect and mitigate the deepfake manipulation of multimedia data in order to minimize the negative impact on society's perception of truth and reality. This chapter presents the introductory study of deepfake manipulation of multimedia data.

1.1 Growing Popularity of Social Media Platforms over the years

The Each year, there has been a significant rise in the number of active users on social media platforms(Figure 1.1), This is mostly due to the widespread availability and affordability of smartphones, which has made it easier for people to access and share material on social networking platforms. Social media serves as a medium for disseminating information, exchanging thoughts, and articulating viewpoints etc.

In contemporary times, there is a prevailing inclination to often disseminate information in the form of images, videos, and audio, as these channels of communication are more captivating than static information conveyed by text.

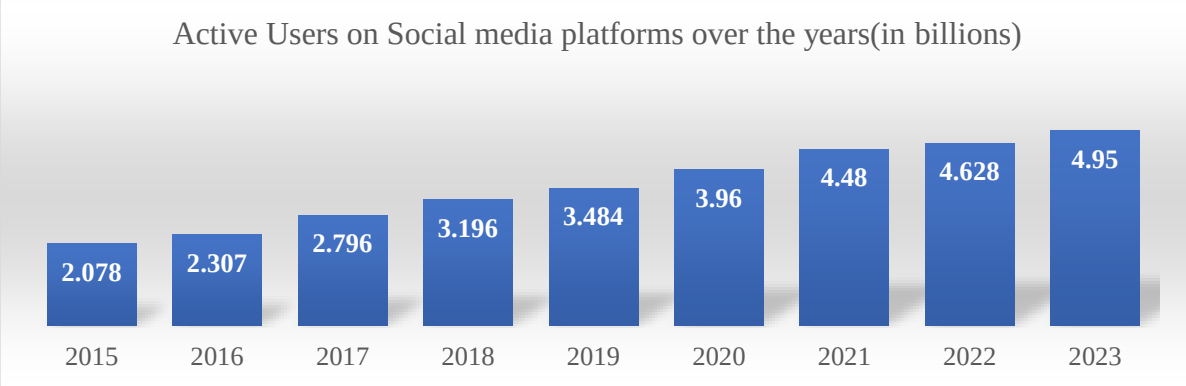


Figure 1.1 Active users on social media platforms over the years [1].

In the past ten years, there has been a substantial surge in the availability of multimedia editing software and smartphone applications. Social media also appeals to a significant percentage of individuals who passively receive information. Users engage in the creation and dissemination of multimedia content, as well as the consumption and exploration of content contributed by other members of the community, including individuals, groups, and organizations.

As these technologies become more prevalent, it is clear that individuals are increasingly using them for their everyday tasks, whether they are related to work or personal matters. Figure 1.2 represents the active users that has increase over the years on various social media platforms.

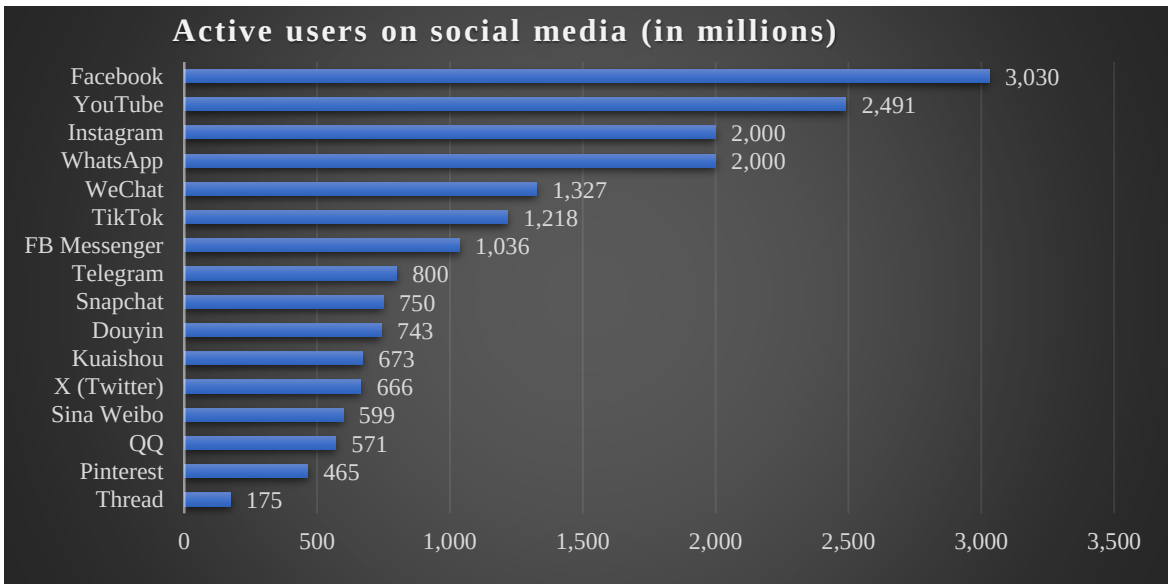


Figure 1.2 Active users on various social media platforms as of Dec 2023 [2].

The Sharing modified content using filters or editing software has become a popular trend to increase visibility and views, leading to more likes and follows. A notable instance of early manipulation in the profession may be dated back to 1865, when a renowned photograph of the

former U.S. president Abraham Lincoln had a face swap [3]. ***Deepfake*** *is the current state of the art of image, video, and audio manipulation.*

> *"Deepfake is a synthetic; realistic-appearing media created by deep learning technology".*

Figure 1.3 Example of Deepfake [4]

Deepfake word is composed of two words, "**Deep**" and "**fake**", which means the fake media that has been created using a deep neural network, a branch of machine learning. Fake media created by this technology appear so realistic and believable that it is difficult to identify as fake to the naked eye(Figure 1.3). This word became famous when, in 2017, a Reddit user with the name "deepfakes" created pornographic content with a swapped face of a celebrity and posed it online. Since then, it has become one of the hot topics, and there is a lot of research going on in recent times.

1.2 Brief overview of the application of Deepfakes

Deepfake has useful for various applications be it in creative field like innovation, education or using with malicious intent like for harassing someone. Application has been divided into categories(Figure 1.4):

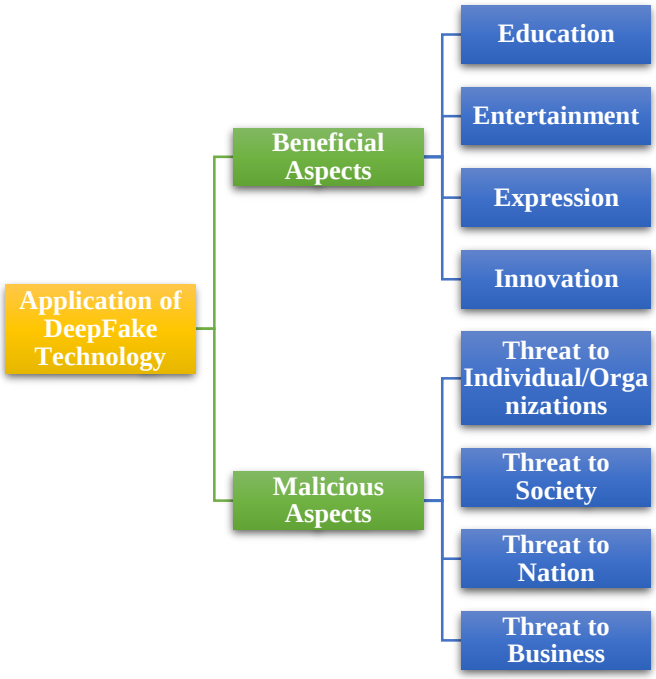


Figure 1.4: Application of Deepfake

1.2.1 Beneficial use of Technology

This technology offers many benefits if used with the right intention and it may include followings fields:

Education: Deepfake gives a multitude of opportunities for educators with the way they can impart education. For example, videos of historical personalities like Mahatma Gandhi or Nelson Mandela teaching about themselves and their work.

Entertainment: Deepfake has also contributed to entertainment purposes in video dubbing in other languages, memes, GIFs, animating dead or cartoonist characters, special effects in the movies [5].

Expression: Deepfake technology allows people suffering from the disability, such as ALS (Amyotrophic Lateral Sclerosis, the patient has difficulty speaking and communicating) to express through their deepfake video. Deepfake also allows one to have an avatar experience through virtual engagement that might be impossible to have physically, e.g. video games [6]. During the campaign, it is also helpful for a message of some famous personality to be reached in a different language; a deepfake can be created for the same [7].

Innovation: Deepfake has been used a great deal to attract customers for a brand. For example, Reuters demonstrated to have used AI generated presenter-led sports news feed. In the fashion retail industry, deepfake allows customers to turn into models (virtually) to try out new

apparels. A Japanese company named Data Grid is already using an AI-generated virtual model for advertising [7].

1.2.2 Malicious use of Technology

The real danger of this technology lies in the different ways it can be misused and the large-scale impact it can have, courtesy of being misused. Here are the some of the threats that it may create:

Threat to individual/organization: Deepfake holds great potential for inflicting tangible harm, psychological stress, physical pain and sabotaging the reputation of an individual and organization. For inflicting harm, a fraudster may use deepfake to extract something of value. To prevent the release of such deepfake, the victim provides money, personal banking details and business secrets [6]. The most common form of exploitation is in the form of deepfake pornographic videos. One can victimize the individual to any form of violent or humiliating act to gratify their wants.

Threat to society: Deepfake can have a huge societal impact considering its realism and fast propagation through different social media networks. Prejudices in society are prevalent and are further aggravated by this technology when the lies are shared through different channels. Societies that are already divided based on caste, creed, religion, color and language, deepfake can further add fuel to the existing fire [6].

Threat to democracy: Deepfake can affect national and international relations; it can sour bilateral ties whose impact may last up to generations. Deepfake can prove to be very lethal, as it gives the option to external entities to influence the democratic process of a nation [6]. Deepfake can sway the results of an election, when a fake video about a political candidate is circulated just on time, such that it has enough time to spread but narrow time to prove it faked and reverse its effect(e.g. on the eve of an election) [6].

Threat to the Business: People are losing money every year in businesses, be it the stock market or business deals; because of the disinformation. Deepfake technology allows anyone to impersonate voices of different identities like the Business leader and CEO to incur fraud. A corporate workplace that is so strict about harassment, sexual abuse, molestation, racist remark, gender discrimination, where evidence in the form of audio or video is hardly questioned. In such an environment, deepfake audio or video can be a lethal weapon that can ruin someone's

career and future aspirations. When deepfake media back a rumor, then such rumors can manipulate the market in such a short time and someone's may lose or make a huge profit [8].

1.3 Types of deepfake for different multimedia

Deepfakes are generated for three kinds of media that are image, video, and audio. Each medium has its own process of generation and hence requires a different type of architecture. Figure 1.5 presents the categorization of deepfake generation methods for three media.

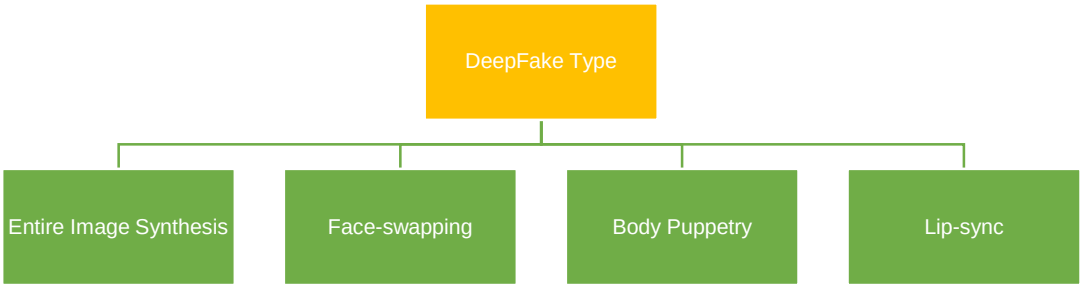


Figure 1.5: Types of deepfake for different multimedia

1.3.1 Type of image deepfake

In the case of images, either the entire synthetic image is generated, or there is partial manipulation, be it expression, identity, or some attributes like hair, skin, gender etc.are changed in the image.

Identity Swap (IS): In this type of manipulation, the identity/face of the source of the image is transferred to the target image. FaceSwap [9] and FakeApp [10] are the most common open-source tools available for the identity swap. With the advent of deep learning techniques, they appear real to such an extent that even sophisticated algorithms have difficulty detecting them.

Entire Image syntheisis(EIS): Entire non-existent images with high realism created by the powerful GANs. ProGAN [11], StyleGAN1 [12], and have leveraged the power of GANs to create highly realistic synthetic high-resolution images.

Lip-Syncing (LS): This category of video manipulation involves synthesizing the mouth region of a target identity consistent with the arbitrary input audio. To convey the information more effectively, lip movement and the corresponding expression are the key elements.

Body-puppetry (BP): Body puppetry means transferring the body movement from the source to the target; it can be facial gestures, eye and head movements, or different body poses. Face enactment is the subset of it, where the facial expression is transferred.

1.4 Motivation for deepfake detection

The impetus behind the development and enhancement of deepfake detection technologies is propelled by certain crucial factors:

- *Prevention of spreading hatred in the society:* The technology of deepfake has the capacity to disseminate falsehoods and misinformation among the general public, thereby inciting negative sentiments and promoting hatred. Therefore, it is crucial to detect and identify deepfake videos or images at an early stage in order to prevent the widespread influence of such bad emotions.
- *Identity theft prevention:* Deepfake detection is crucial in preventing theft which usually occurs when an identity of an individual is faked or taken for fraud purposes.
- *Prevention of Incuring financial fraud*: The primary purpose of forgery detection systems in the financial sector is to prevent fraudulent activities. This involves the detection of forged checks, counterfeit currency, and deceptive credit card transactions.
- *Protection of Individual:* Deepfakes are used to specifically target the individuals for blackmailing, harassment or with the intention of victimizing them. Exposing and detecting deepfakes from these malicious acts, is one of the foremost priority to preserve their personal rights and privacy.
- *Protection of social Media scam and reputation damage:* Deepfake allows someone to fake their identity which would give them power to fraud somebody on their behalf which would eventually damage their reputation.
- *For building trust in Online transactions:* Deepfake detection systems primarily aim to identify fraudulent activity in the financial sector on the internet, such as fraudulent transactions, counterfeit credit/debit card transactions, and other manipulative operations.
- *Ensuring trust in public institutions:* Public institutions such as the judiciary, government, police, and other public services are established to ensure the efficient operation of society and the nation as a whole. The emergence of deepfake technology would jeopardize their trustworthiness. Robust detection mechanisms are implemented to maintain the trustworthiness of these institutions.
- *Integrity of journalism and media*: Journalism and media organizations should implement robust detection mechanisms to verify the validity of news and prevent the dissemination of misleading information. This ultimately contributes to maintaining integrity and protecting people from the spread of fake news and misinformation.

- *Enhancing cybersecurity:* Cybersecurity is enhanced by the utilization of deepfake forgery detection techniques, which are employed to identify and thwart various types of assaults, including email spoofing, phishing, and malware that aim to deceive or imitate individuals.
- *Ensuring the safeguarding of trademark rights and reputation:* Deepfake technology empowers individuals to assume the identity of a brand and promote their goods, hence diminishing the market value of their competitors' products. Implementing a detecting technique would aid in safeguarding the company's name and brand value.
- *Ensuring the political and social stability in the nation:* The utilization of deepfake technology allows external actors to exercise influence on a nation's political process, potentially resulting in substantial public upheaval. This has the potential to intensify demonstrations and present a risk to the security of the nation. The deliberate distribution of deepfake content has the capacity to sway the results of elections in a democratic country. Hence, it is imperative to establish effective detection algorithms to accurately identify and classify manipulative information.

1.5 Sources of Studied Research Works

This section outlines the methodology employed in the preparation of this thesis. This thesis incorporates research papers sourced from reputable publications, conferences, and workshops available in prominent sources such as IEEE Xplore, Science Direct, Springer, ACM, and Google Scholar. In order to incorporate relevant papers, keyword searches were conducted for terms such as "forgery detection", "manipulation detection", "images", "videos", "deep", "review", "survey", and so on. Research contributions were prioritized by including high-quality journals such as ACM Transactions and IEEE Transactions, as well as top computer vision conferences such as the European Conference on Computer Vision (ECCV), Conference on Computer Vision and Pattern Recognition (CVPR), IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), and International Conference on Computer Vision (ICCV).

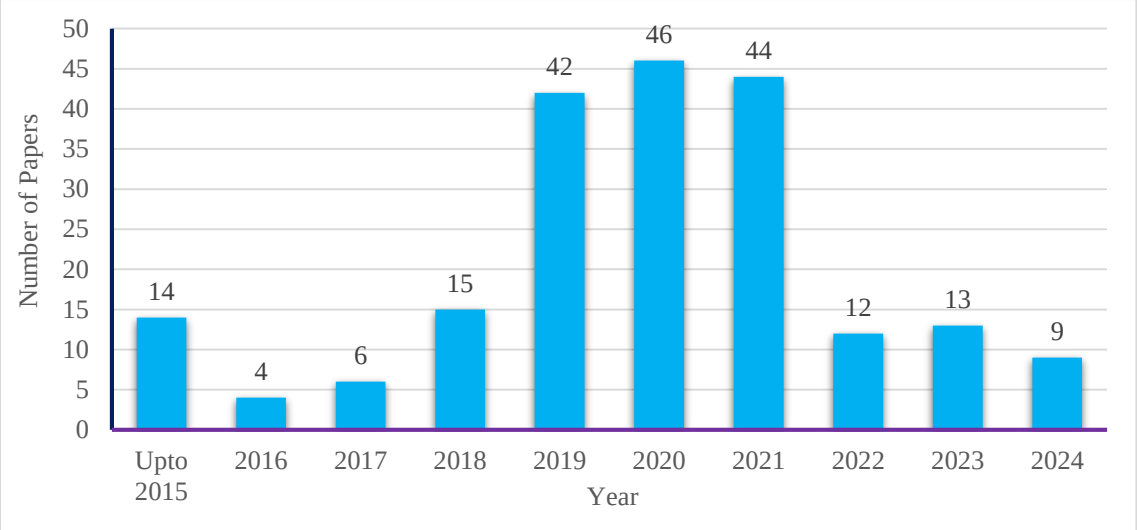


Figure 1.6: Year-wise distribution of Deepfake Manipulation Literature

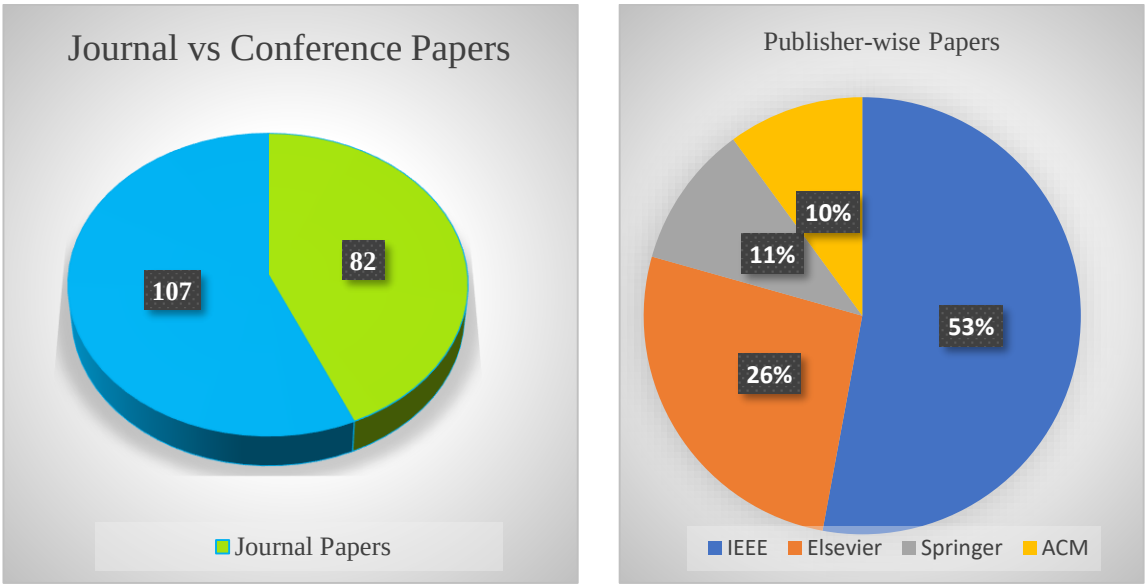


Figure 1.7: The first graph gives a comparison between Journal and conference being cited. The second graph gives the publisher-wise distribution of papers

Figure 1.6 illustrates the distribution of contributions by year, indicating that the majority of contributions come from years 2021-2022. Figure 1.7 illustrates the dissemination of articles referenced in this thesis. The initial graph displays the quantity of conference and journal publications that have been referenced in this thesis. The following graph displays the distribution of publications based on the publishers.

1.6 Thesis Overview

This dissertation is divided into six chapters:

Chapter 2 focuses on reviewing the literature that discusses the current advanced methods for detecting manipulation in multimedia information. More precisely, the presentation showcases research efforts that are classified based on the specific types of alterations they can detect. It also highlights any gaps in the existing research, the specific objectives that the study aims to achieve, and the contributions that the research has made.

Chapter 3 focuses on the issue of detecting deepfake alteration in images. Two innovative deep-learning methodologies utilizing the texture and cross-attention mechanism of vision transformers have been put forward. The initial model use Gram matrices to generate texture features, but the second technique combines Gram matrices with Local Binary patterns to describe texture features. The remaining architecture remains similar between both models.

Chapter 4 focuses on the issue of deepfake video datasets and their identification. The Div-DF dataset is a comprehensive collection of deepfake videos that covers several video alteration techniques, such as lip-sync, facial reenactment, and face swap. The deepfake detection model comprises an Xception model enhanced with spatial and channel attention, as well as an LSTM component to capture artifacts. The suggested model and several state-of-the-art methodologies are used to benchmark the score on the dataset.

Chapter 5 discusses the issue of localizing deepfake manipulation. The proposed framework utilizes a dual-branch approach, where one branch incorporates noise characteristics and the other branch incorporates RGB information. These branches are then combined and fed into an ED architecture for the purpose of semantic learning. Multiple tests are conducted on the shallowfakes and deepfakes dataset to identify the specific locations of the modifications.

Chapter 6 provides the final findings and conclusions of the study conducted in this dissertation as well as potential avenues for future research.

Chapter 7 contains the sources that are mentioned in this thesis.

Chapter 2: Literature Review

This chapter explores the existing literature on the problem of Deepfake manipulation generation and detection in multimedia content. The overall literature review can be divided into two categories: namely known as Deepfake generation and deepfake detection(Figure 2.1).

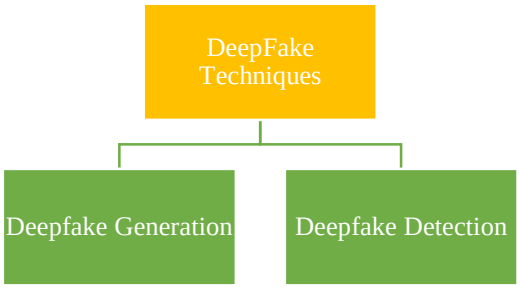


Figure 2.1 Classification of Deepfake Techniques

2.1 Deepfake Generation Techniques

Based on the generation process of different media, the generation mechanism has been divided into two categories: visual deepfake, including image and video media, and audio deepfake generation. Figure 2.2 presents the further categorization of deepfake generation methods.

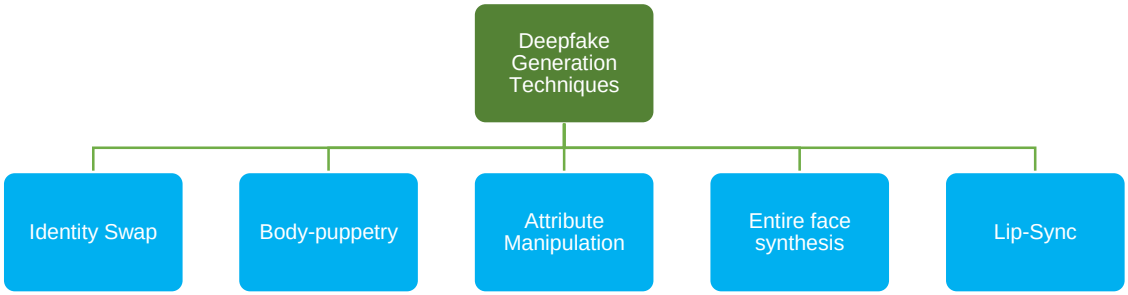


Figure 2.2: Classification of Deepfake generation techniques

Deepfake generation involves the generation mechanism for image and video media. This section will cover various categories under which manipulation can take place.

2.1.1 Identity Swap (IS)

In this type of manipulation, the identity/face of the source/frame is transferred to the target image/frame. Face-swap is the other name for it. The face-swap [9] and fake app [13] software made it easier for anyone to generate face-swap. Their typical approach for IS based on a pair of auto-encoder and decoder architecture. For this process, encoder-decoder pairs are required, where the encoder converts the images into their latent representation while the decoder reconstructs the image back from the latent representation. During the training, encoder-

decoder pair is required for each image for the model to learn its embedding, where encoder weights are shared. Once training is complete, the decoder is interchanged during the generation stage, the decoder of the target image and encoder of the source image are used to generate the target (Figure 2.3).

The researcher proposed various sophisticated algorithms over the years. The first well-known Identity swap is FaceSwap [14], which has also been used to develop Faceforensics++ dataset [15]. The traditional method uses 3D morphable models, and facial textures are replaced with the estimated 3D model's geometry with the target image. Dale et al. [16] model has been one of the old approaches that use the multi-linear model to track the facial performance in both videos and then use 3D geometry to warp the faces. Now a days, IS architecture uses DNN that usually uses two modules, one uses latent space for disentanglement of identity from other attributes and then the other module transfer and refine the identity from source to target. Faceshifter [17] wherein the first stage, the method generates the swapped on the target images thoroughly and adaptively and in the second stage, network recover anomalies region in a self-supervised manner. Most of the recent IS methods are subject agnostic, where once model get trained, it can be applied to any new faces without requiring re-training on subject specific target.

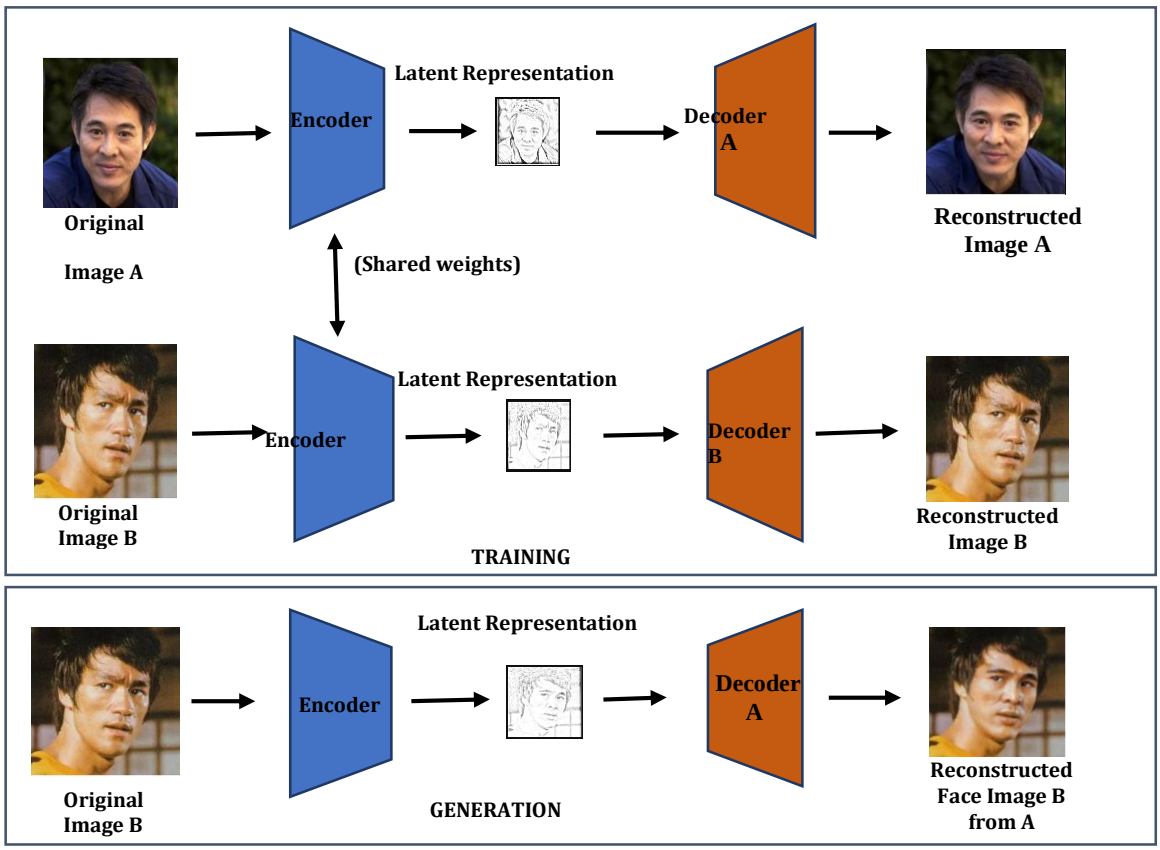


Figure 2.3: Identity Swap generation model using auto-encoder & decoder

2.1.2 Body puppetry (BP, aka reenactment)

Body puppetry (aka reenactment) deepfake is where source derives the content of target; it can be facial gestures, eye and head movements, or different body poses. Face reenactment is its subset, where the facial attributes are derived. It is used greatly for post-production editing of movies or short videos [18]. Most of the time, the target content is derived either from some source media in the form of images/frames using landmark key points, 3D morphable models, skelton or any other mapping method. Chan et al. [18] proposed a method to transfer dance moves from the source to the target using intermediate pose Skelton transfer and predict the two consecutive frames to produce coherent results. However, the samples cannot generate realistic poses, especially at the joints of the body. Thies et al. [19] proposed the famous Face2Face approach that allows for the real-time reenactment of facial expression, where source facial expressions are tracked using a dense photometric consistency measure, and then a transfer function exploits the deformation transfer in semantic space. Many techniques [20] [21] [22] require multiple input images of the source samples, while few methods require few samples [23] [24] or even single sample [25] to generate results Many methods have the limitation that they can synthesize one attribute and apply it to only low-resolution images; FaceSwapNet [26] resolves this issue using two modules: landmark swapper and landmark-guided generator to generate the face expression enacted photo-realistic image. Some other methods [27] [28] [29] [30] [31] have also been proposed over the years. Most of the reenactment has been done on dance poses and facial expressions. There has been a considerable improvement in the quality of dance poses generated samples, but still, they are far from appearing realistic.

2.1.3 Lip-syncing(LS)

This category of video manipulation involves synthesizing the mouth region of a target identity consistent with the arbitrary input audio. To convey the information more effectively, lip movement and the corresponding expression are the key elements. Usually, Influential leader's deepfakes are developed, as their audio, video, and images are readily available and their generated samples creates more impact. Suwajanakorn et al. [32] created one of the first well-known lip-sync of ex-President Obama from the audio using a recurrent neural network to map raw audio features to different mouth textures, new identity requires the model to be trained again. Earlier lip-sync methods [33] [32] construct 3D talking face models for a specific by animating 3D face meshes of the specific chosen subject, and such methods are hard to scale to arbitrary identities. However, as the technology evolves, they start to disentangle audio-

visual representation, allowing methods [34] [35] to use few subject samples to generate results. Real-time reenactment of audio has also become possible nowadays. Jamaluddin et al. [36] proposed a real-time cross-modal self-supervision model for synthesizing talking heads that employs encoder-decoder multi-stream CNN, which uses a joint embedding of still images and audio to generate lip-synched video frames in real-time. However, the model lacks the synthesis of real-time emotional facial expressions. There has been significantly less work for lip-sync manipulation, and also, methods have difficulties being generalized to any arbitrary identity.

2.1.4 Attribute Manipulation(AM)

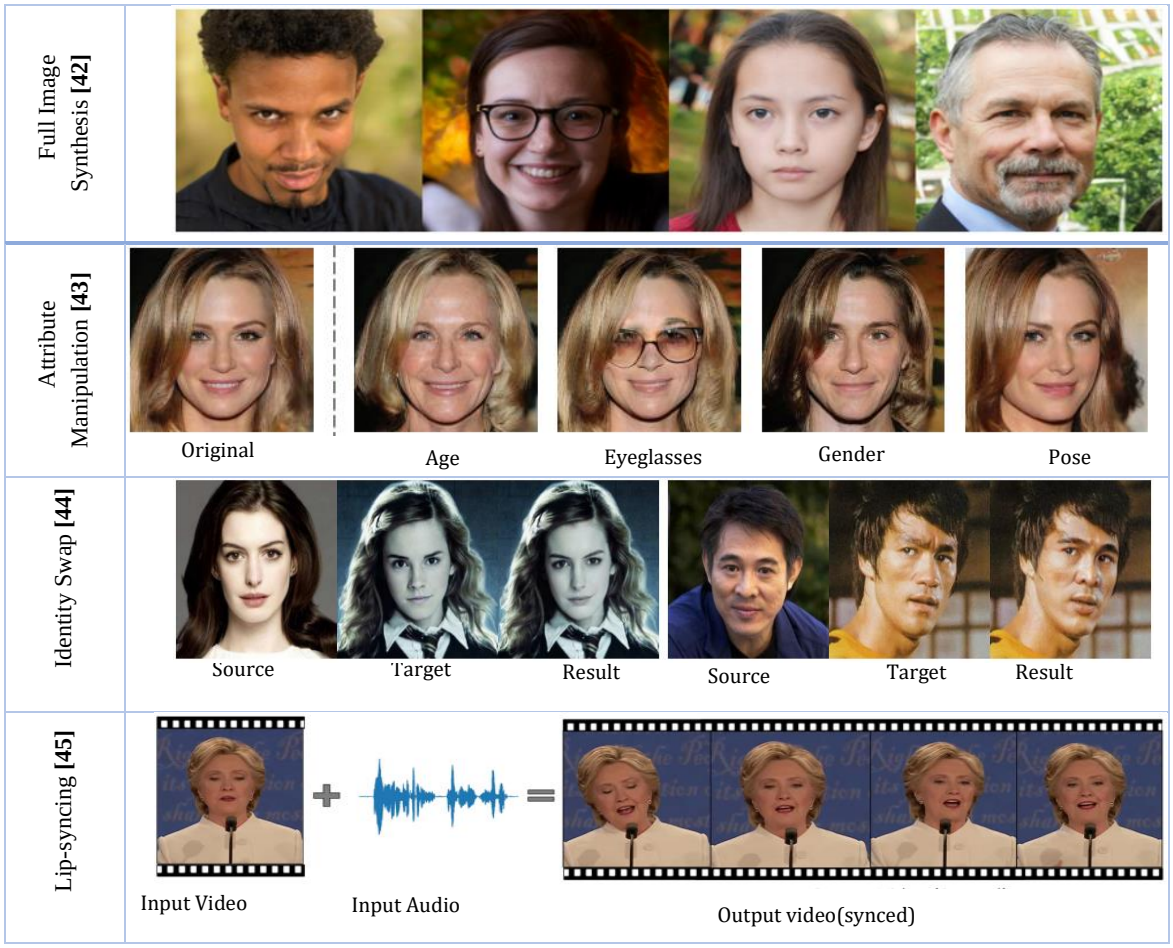
Attributes like expressions, hair, eyes, the color of skin, age, gender, mustache, etc. that are manipulated in an image fall into manipulation category.

Generally, attribute manipulation methods employ either Encoder-decoder(ED) or a combination of ED and GANs with a conditioned attribute. ED-based decodes the latent representation of attribute in a latent representation. A relationship is established between latent representation and attribute independent editing, which allows the independent attribute manipulation without the identity information loss that may lead to a distorted or over-smooth generation of the results.

StarGAN and STGAN are classic examples. Earlier domains used to do the image-to-image translation between two domains, which was time-consuming, but StarGAN [37] approach uses multi-domain image-to-image translation using a single model. This allows the training of multiple datasets of different domains within the same network. However, the model can only produce a limited number of expressions despite such flexibility. To address this limitation, Albert et al. [38] novel GAN based model named ganimation uses a weakly supervised attention mechanism that takes annotated facial action units (AU) as input and generates a wider range of expressions. Encoder-decoder and GAN architecture has bottleneck layers, which results in blurry and low-quality results and adding skip connection to overcome these, results in weakened attribute manipulation. For this, Ming et al. [39] proposed STGAN using selective transfer manipulation that incorporates specific target units into the encoder-decoder model, changing target face attributes. While manipulating, Guim et al. [24] use latent space and conditional attribute representation, which helps regenerate the image by modifying the required attribute. Some other methods [40] [41] which has been proposed recently which manipulates the style of an image using StyleGAN [12] latent space.

2.1.5 Entire Image Synthesis(EIS)

Entire non-existent images with high realism created by the powerful GANs. ProGAN [11], StyleGAN1 [12], and have leveraged the power of GANs to create highly realistic synthetic high-resolution images. Terro et al. [11] proposed the ProGAN methodology, which allows generating high-resolution images progressively by adding the number of layers gradually with the training. They started with a low-resolution image and started growing the layers of generator and discriminator with a resolution of the image. The image generated is of high – quality, but at times, the image generated is far from being real. Another method, StyleGAN1 [12] which interpolates the various features such as pose and human identity by disentangling the high-level attributes from the stochastic variation(like freckles, hair) of the generated image in an unsupervised setting. The method enables intuitive, scale-specific control of the synthesis. However, they found several typical artifacts of StyleGAN. To improve the model, they proposed StyleGAN2 [42] in which they redesigned the architecture with the normalization used in the generator and adapted the progressive GAN approach by regularizing the mapping of the generator from the latent code to images. Most methods have a hard time finding the trade-off between fidelity and the variety of generated samples (Figure 2.4).



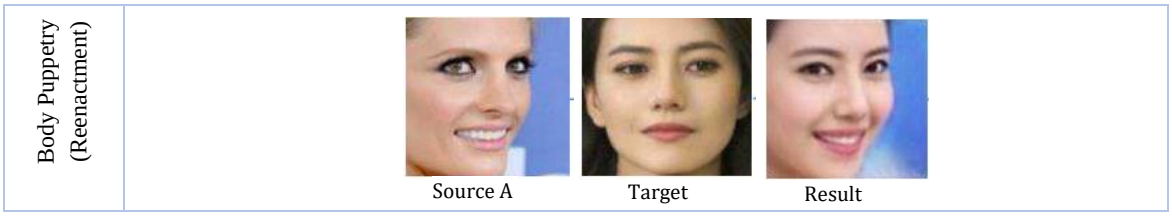


Figure 2.4 Types of visual deepfake manipulation

2.2 Deepfake Detection

There is an armed race going on between manipulators and the detector; the detector used some clue to find the detection, the manipulator would try to diffuse it next time to make it detection-proof; and the race goes on and on between them. Generalization of the deepfake technique, robustness against various post-processing operations and interpretability of the detection results are three main critical factors for a detector to be deployed in the wild [46].

Visual deepfake and audio deepfake these are the two different media for which different detection algorithm has been designed. Based on the clues/traces of feature representation, these two categories has been divided further, which is shown in the Figure 2.5.

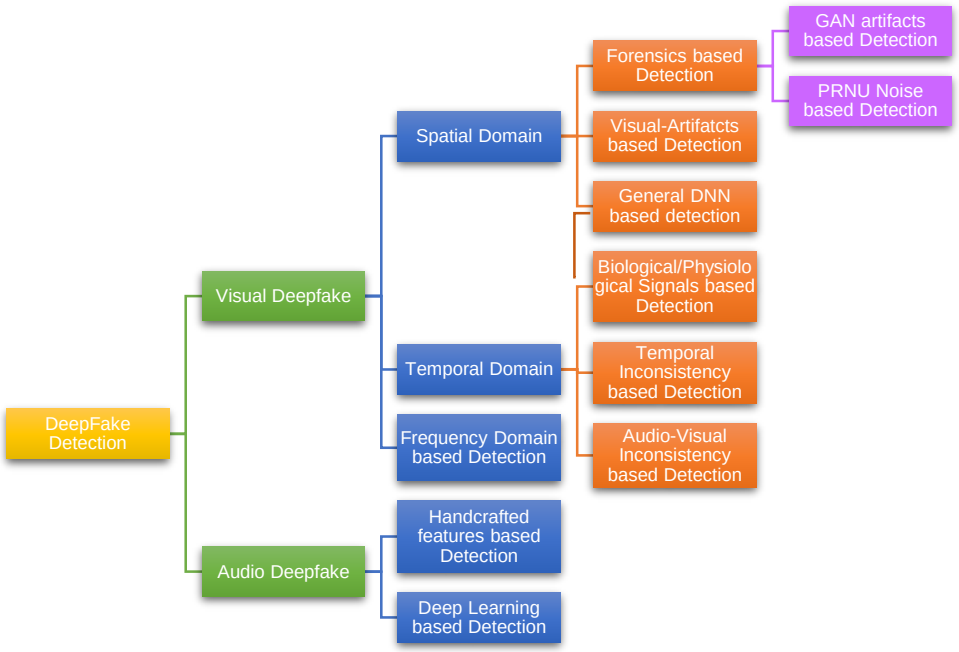


Figure 2.5: Classification of deepfake detection methods based on feature representation

2.2.1 Deepfake Visual detection

Manipulated or manufactured samples have certain peculiarity in spatial, temporal or frequency domain representation, which the different detection algorithm exploits. The subsequent sections will discuss the various detection techniques along different domains.

2.2.2 Spatial domain based detection

For manipulated samples, the corresponding pixel distribution gets changed, which is reflected in the spatial domain properties. This section will discuss various types of spatial domain-based detection methods.

2.2.2.1 Forensics based detection

Generation methods leave certain clues or traces that change the distribution of the samples, which is exploited by the detection methods by analyzing latent features and patterns. Li et al. [47] analyses the subtle distribution of the image statistics in the chrominance components of YCbCr and HSV color spaces, especially in the residual domain for the unseen DNG images. Chen et al. [48] use a multi-domain architecture where the features from the RGB domain and the noise vectors(gets added by the external mechanism) are fused to obtain richer robust features.

There are various variants of Forensics features, which are mentioned below:

- **GAN-Artifacts based artifacts:** The imperfect design of the GANs leaves some traceable clues, which various researchers use for fake detection investigations[1]. McCloskey et al. [49] utilized the prior knowledge about how the color is treated in GAN and camera models and used this knowledge as a cue to design the network. Methods perform greatly for the existing GAN model; however, the model's performance is unclear for new advanced GANs. Yu et al. [50] identify a unique GAN stable fingerprint that persists across different frequencies and patches of the generated image, extracted by a neural network. However, the method fails for post-processing operations like compression, blur etc.
- **PRNU Noise based detection:** Photo response non-uniformity (PRNU) is a noise-like pattern in the digital image caused by the camera's light sensor. Koopman et al. [51] proposed a method where Co-relation scores are computed between every eight groups of PRNU pattern frames, which serve for deepfake video detection. However, their evaluation is limited to the small dataset. PRNU based are generally low-cost based methods and have a high generalization capability.

2.2.2.2 Visual-Artifact based Detection

The synthesized or manipulated faces would reveal inconsistencies in the appearance, especially the blending boundaries, landmarks points or the shape of the manipulated facial attributes. Even sometimes, the content of the face also seems anomalous to the rest of the

background. Li et al. [52] use the face warping artifact as a clue which is caused due to blending operation to match the configuration of the source face. Unfortunately, this affine warping operation leaves the artifact due to resolution inconsistency between the face and the surrounding area. The method is more robust than other existing methods, but there is still room for improvement.

Visual-Artifact based methods can achieve better generalization as they pay more attention directed to the specific artifacts. Li et al. [53] propose a generalized detector that uses face X-ray, which also uses blending boundaries for detection purposes. However, such a method fails for entirely synthetic images, indicating the blending operation's absence. Also, adversarial samples can be designed to bypass the detection mechanism by curbing such manipulation detection. Matern et al. [54] also exploited the visual artifacts like the difference in eye color, inconsistent illumination, missing teeth areas, etc.; once the algorithm extracts the artifacts, they are further used for classification. These methods can localize the manipulation easily, as their detection is based on specific localized artifacts. Some other methods ([55] [56] [57] [58]) also have been proposed which looks for visual clues for deepfake detection.

2.2.3 Temporal domain based detection

Temporal information is the sequential info that is relatable, coherent and changes with time. Manipulated artifacts or traces could be revealed in the temporal domain in the form of flickering/jittering. The subsequent sections will discuss various methods that investigated the temporal domain to find such clues.

2.2.3.1 Audio-Visual Inconsistency based detection

For lip-sync methods, inconsistency between the mouth region (visual) and audio is one distinguishing factor for deepfake detection. Agarwal et al. [59] use a CNN to detect inconsistent mouth features, i.e. shape of the mouth (visemes) are not aligned with spoken words (phenomes) in a manipulated video. They focused on the visemes related to the words M, B and P, in which the mouth almost gets completely closed. However, their method is specific to the videos of Barack Obama. Mittal et al. [60] simultaneously exploited visual and audio modalities and perceived the affective cues from these two modalities to detect any alteration. This exploration between these two modalities uses the Siamese network, where triplet loss is used to measure the similarity. Sometimes, this method cannot detect any manipulation if there is a similarity between these perceived affective cues. Moreover, this method is limited to a single person in a video. Chugh et al. [61] proposed a method to calculate disharmony score between visual and audio modality and termed it Modality Dissonance

Score(MDS). This dissimilarity score is calculated chunk-wise per video segment, and contrastive loss is employed to calculate such inter-frame modality similarity. However, these methods rarely look into visual consistencies, which could also be faked. That is why it remains unclear whether such methods can be deployed in real-world scenarios where one may encounter any manipulation and undergo different post-processing operations.

2.2.3.2 Temporal Inconsistency based methods (TI)

In the manipulation of video frames, correlation structures between the frames are sometimes destroyed, reflecting in various forms like video flickering or shifting of the facial content [62][2], [3], [4]. Usually, sequence models like RNNs and their variants get employed to find such inconsistencies between the frames. Hosier et al. [63] use video speed manipulation as a temporal feature for detection, the encoding used for each frame gives an idea about the number of deleted and added frames. Methods that use frame-level artifacts and temporal features for detection perform better than those that focus on either of the two. Guera et al. [3] propose a temporal-aware CNN-LSTM framework, which exploits the frame-level features along with temporal inconsistency between frames. Temporal inconsistency is introduced by the auto-encoder (used for face swapping), which focuses on the face-swapping process unaware of inconsistency introduced by the process, which results in anomaly serving as crucial evidence for detection.

An optical flow mechanism has been used to estimate the per-pixel motion behavior of the adjacent frames. Amerini et al. [64] used a pre-trained model (trained on RGB images) with an optical flow mechanism to capture the dissimilarity between frames. Although the methods have reported very few results. Caldelli et al. [65] also proposed optical flow-based CNNs that exploits the motion dissimilarities in the temporal nature of the video sequences using optical flow fields. Again the approach is limited to the specific dataset. These techniques tend to perform better as they are independent of the specific type of manipulation. However, when these methods are used in conjunction with spatial methods, the overall performance improves.

2.2.3.3 Physiological/Biological Signals based methods (PBS)

In deepfake videos, inconsistencies are exhibited either at the physiological level (like inconsistencies in eye blinking pattern, head poses, blending boundaries) or biological level (inconsistencies in a heartbeat), which is exploited for deepfake video detection. Methods can generate deepfakes with high realism, but they cannot replicate every reasonable behavior, which leads to their inconsistencies. Such inconsistencies at the physical level may or may not be visible from the eyes, hence needs some landmark detector that captures the coordinates of

the desired location to be used further for classification. Yang et al. [66] use a landmark detector for 3D head poses to calculate their estimated positions, exploited by the SVM classifier. Their performance degrades to blurry images. Li et al. [67] proposed a method based on eye blinking patterns, which is not well preserved in the synthesized videos. Synthesized videos usually have less frequency of eye blinking, which leads to their detection. The technique can exploit abnormalities in the normal functioning of an organ for deepfake video detection. These signals are preserved neither spatially nor temporally, and different architecture exploited them for detection. Qi et al. [68] use heartbeat rhythms as a clue for the detection of deepfake videos. Visual photo plethysmography (PPG) monitors the heartbeat rhythms and captures the abnormalities of the deepfake videos. However, the model does not generalize well to the unseen dataset. Ciftci et al. [69] proposed FakeCatcher methods that use biological signals such as heart rate to exploit the authenticity of a video. They have extracted signals on a pairwise basis and transform them to a different domain (like frequency, time, etc.) and use this transformation further for classification.

Although the methods based on these features perform well on the various datasets, such signals get seriously affected by the video's quality and a limited application for detection mechanism based on such signals [62]. [70] [71] some other methods that looks for biological methods.

2.2.4 Spatial and/or Temporal domain based detection

Few detection methods could leverage both the spatial and the temporal domain or either of the two. However, features fetched along both domains capture the broader range of manipulation traces which would eventually help in a better detection mechanism. The next section will discuss such detection methods.

2.2.4.1 General DNN based detection

Instead of focusing on the specific artifacts, some researchers let the network decide which latent features to analyze and learn the mapping accordingly. Deep neural networks drive such methods. Khalid et al. [72] uses variational autoencoder (VAE) to train real images, classify real images, and treat others as anomalies. Generalization on the unseen dataset has been a bigger issue for such methods, as they learn the specific type of manipulation on the data they are trained upon and hence, tend to overfit and perform poorly on the other type of manipulation in the wild. Although, few authors can develop a generalizable detector. Xuan et al. [73] also proposed a generalized GAN images forensics detector that preprocesses the images with

Gaussian blur and noise operation to enhance the high-frequency pixel noise to allow the CNN model to learn intrinsic discriminative features.

Nowadays, pre-trained models are used heavily to use the learned weights of similar problems to reduce the time complexity of the network. ResNet, XceptionNet, Densenet, AlexNet, Inception model is the recent state-of-the-art models generally used as a pre-trained model. Zhou et al. [74] used GoogleNet Inception V3 pre-trained model in one of the branches of the architecture to detect the tampered artifacts evidence and noise inconsistency. Jeon et al. [75] proposed a framework for neural talking head detection using a pre-trained AlexNet model to extract features even from a highly unbalanced dataset and then classify them further using Siamese network-based classifier.

DNN models could also be used for sequence-based problems where data in audio or video frames pass through the model to analyze and learn the intrinsic pattern. Recurrent Neural Networks and their variants LSTM and GRU are generally used for this purpose. Wu et al. [76] exploited temporal, spatial, and steganalysis features for deepfake video detection. The deep neural network extracts spatial features for the tampering artifacts like unregular shapes, color, etc. Steganalysis features are extracted by putting constraints on the Convolution filter for underlying abnormal statistics of the pixels. The temporal inconsistencies are extracted using RNNs.This method beats the current state of the art methods on the FF++ dataset.

DNN based methods are very good at learning and extracting the intrinsic characteristics along several domains. Such methods tend to overfit the specific datasets they are trained upon, but they lack generalizability to other datasets. Also, the existing methods fail in proving their effectiveness against the adversarial noise attacks [46]. Also, the models do not have the interpretation of why their method has proved something fake due to the black-box nature of the model. Some other methods[5], [6] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95] [96] [97] [98] [99] [100] [101] [102] [73] have extracted discriminative features using DNN either implicitly or explicitly to perform deepfake detection. Several researchers have worked on the crucial problem of generalization for the deepfake detector [103] [104] [105] [106] [107][7], but it is still far away from the desired solution. The generalization model aims to identify commonalities between different types of manipulation. That was the focus of research [108] [109][8], [9]. Yan et al. [108] provide a disentanglement structure of architecture into three categories: method-specific forgery, forgery irrelevant and common forgery features and then a multi-task approach is followed for the disentanglement of method -specific and common forgery traits that allow for the binary classification of the results. Yu et al. [109] assess the authenticity of a module by using U-Net architecture and an

independently trained particular forgery feature extractor. However, the technique presupposes the existence of comparable forging traces, which the adversary may have hidden or covered up. A code identification method was created by Li et al. [110] that captures the accurate space distribution of real and fake images using a codebook. The model works well with various compressed and cross-dataset images, but its performance with additional and wild datasets is still unknown. For a detector to function in various adversarial circumstances, one significant difficulty that needs to be investigated is the detector's generalizability.

2.2.5 Frequency domain based detection

The frequency-domain represents the change of pixel distributions along the different axis[10]. Real images have a certain frequency distribution; when some generative model does the manipulation, such difference could be revealed in the frequency domain. Frank et al. [111] explored the frequency spectrum of the GAN-generated images and found that images exhibit severe artifacts consistent across different resolution images, which is mostly caused by the up-sampling process of the GANs. Furthermore, the model is robust against various kinds of image perturbation like blurring, cropping, compression and noise addition. Durall et al. [112] observe the behavior of the real and fake images in the classical frequency domain, use such behavior to be detected by the classifier. However, the model has low accuracy on the low-resolution images. Masi et al. [113] used a two-stream network to exploit frequency domain information in one of the streams using the Laplacian of Gaussian(LoG) operator. The LoG acts as a band-pass filter to suppress the image content and amplify the artifacts. Nevertheless, the method struggles to detect real-world data samples.

2.3 Research Gaps

Based on the literature presented in above section, various research gaps has been identified:

- Existing methods perform well on the same type of manipulation (seen during the training and testing phase) but their performance degrades on the other kinds of manipulations, which affects their **generalization capabilities**.
- The detection capabilities of the methods are satisfactory, but they do not **localize the manipulation** very well.
- Several deepfake datasets **lack the diversity** of video manipulation, such as lip-sync face reenactment.
- Detection algorithms uses discriminative features for classification which are **either learned by hand-engineered mechanism or deep learning methods**. Each has some

limitation for learning discriminative features, which could be overcome, if they are used in conjunction in a model to learn features which may give better performance.

2.4 Research Objectives

Based on the identified research gaps, the following objectives have been proposed:

1) To develop a novel and effective generalized framework which can detect and perform well over unseen and different kinds of deepfake manipulations.
2) To propose a novel deepfake dataset with a large variety of video manipulation types and a comparative analysis of several state-of-the-art methods on the proposed dataset.
3) To develop a new framework for the localization of deepfake manipulation.

2.5 Research Contributions

This research thesis has made the following scientific contributions:

- Proposed a novel deepfake detection model called Tex-ViT utilizes gram-matrices as texture feature descriptors and incorporates a cross-attention mechanism from the vision transformer. The model integrates conventional ResNet characteristics with a texture module that functions simultaneously on segments of ResNet prior to each down-sampling process. This module functions as an input to the dual branch of the cross-attention vision transformer. The model's generalizability is demonstrated by experimentation conducted on several categories of FF++ and GAN dataset images in cross-domain contexts. The Celeb-DF, FF++, and DFDCPreview datasets were utilized for conducting experiments, employing several post-processing techniques like blurring, noise addition, and compression. The results highlighted the robustness of the models in many scenarios.
- Proposed Tex-Net is an alternative approach to detect deepfakes. It uses a combination of Gram matrices and Local Binary patterns to represent texture information. The rest of the architecture of Tex-Net is similar to that of Tex-ViT. The global texture is calculated during each down sampling operation of ResNet, and then, layer attributes are merged at many semantic levels. These properties consistently combine before being input into the dual-branch cross-attention-based vision transformer for classification. The model's ability to generalize was proved by conducting experiments on several categories of FF++ and GAN dataset images in a cross-manipulation context. Experiments were conducted on data samples from FF++, DFDCPreview, and Celeb-Df, which underwent different post-processing techniques such as blurring, noise addition, and compression. These experiments demonstrated the model's resilience.

- Additionally, a Div-DF dataset was introduced, which includes a wide range of video modifications such as face swapping, facial reenactment, and lip-syncing. The dataset has 150 genuine videos showcasing a diverse range of celebrities from various domains, along with 250 deepfake videos. The collection of deepfake videos comprises 100 videos featuring face-swapping, 100 videos showcasing facial reenactment, and 50 videos demonstrating lip-syncing. Deepfake videos are produced by employing sophisticated methods like the Face-Swap GAN (FSGAN) and the Wav2Lip approach.
- A novel deepfake video recognition model is provided, leveraging the pretrained Xception and LSTM models to enhance its sophistication. Xception utilizes depthwise separable convolution to capture the underlying spatial anomalies, while LSTM captures the variations among the altered sequences. The hybrid model assembly allows for the gathering of information regarding spatial and temporal distortions in several dimensions, making it a potent tool for detecting deepfakes. An assessment of the effectiveness of the suggested model and other advanced models on our Div-Df dataset demonstrates the superiority of the proposed model.
- A unique model is developed for localizing deepfake manipulation. The model has a dual-branch architecture that combines manually crafted feature noise with Convolutional Neural Networks (CNNs) as an Encoder-decoder (ED) system, bolstered by the attention mechanism. This model employs a dual-branch methodology, where one branch integrates noise characteristics and the other branch integrates RGB features. These characteristics are subsequently inputted into an ED architecture for the purpose of semantic learning. In addition, skip links are incorporated to maintain spatial information. A comprehensive investigation was carried out on the shallowfakes dataset, encompassing CASIA, COVERAGE, COLUMBIA, and NIST16, along with the deepfake dataset Faceforensics++ (FF++). The evaluation results demonstrate the model's excellent feature extraction capabilities.

The subsequent research studies serve as the foundation for this chapter.

1. **D. Dagar** and D. K. Vishwakarma, “A literature review and perspectives in deepfakes: generation, detection and applications” ***International Journal of Multimedia Information Retrieval***, vol. 11, June. 2022, doi: https://doi.org/10.1007/s13735-022-00241-w.

Chapter 3: Deepfake Detection in Images

3.1 Scope of this Chapter

This chapter focuses on the issue of deepfake detection in cross-domain settings where training and testing comes from different distribution. In order to achieve this objective, two innovative deep-learning architectures based on the texture feature and cross-attention mechanism are proposed. The two models differ only in the way texture computation is done. First architecture known as Tex-ViT, model collaborates conventional ResNet features with a texture module that runs parallel acts on parts of ResNet before every down-sampling operation and serves as an input to the dual branch of the cross-attention vision transformer. The architecture uses Gram matrices, which calculates the correlation between the features maps, for the computation of the texture features. The second architecture, Tex-Net uses the combination of Gram matrices and Local binary patterns as a texture descriptor and the rest of the architecture is same as of the first architecture. Experimentation done on the public deepfake dataset and GAN dataset images in the cross-domain settings and hence the model beat the score of various state-of-the-art models, proving texture as a feature that persist in various kinds of manipulations. Experimentation also done on the in-domain settings for post-processing operation like blurring, addition of noise and compression and once again the model established superiority over other models.

3.2 Tex-ViT: A Generalizable, Robust, Texture-based dual-branch cross-attention deepfake detector

3.2.1 Abstract

Deepfakes, which employ Generative Adversarial Networks (GANs) to produce highly realistic facial modification, are widely regarded as the prevailing method. Traditional Convolutional Neural Networks (CNNs) have been able to identify bogus media, but they struggle to perform well on different datasets and are vulnerable to adversarial attacks due to their lack of robustness. Vision transformers have demonstrated potential in the realm of image classification problems, but they require enough training data. Motivated by these limitations, this publication introduces Tex-ViT (Texture-Vision Transformer), which enhances CNN features by combining ResNet (Residual Networks) with a vision transformer. The model combines traditional ResNet features with a texture module that operates in parallel on sections of ResNet before each down-sampling operation. The texture module then serves as an input to the dual branch of the cross-attention vision transformer. It specifically focuses on improving

the global texture module, which extracts feature map correlation. Empirical analysis reveals that fake images exhibit smooth textures that do not remain consistent over long distances in manipulations. Experiments were performed on different categories of FaceForensics++ (FF++), such as Deepfakes (DF), Face2Face (f2f), Faceswap (FS), and Neural Texture (NT), together with other types of GAN datasets in cross-domain scenarios. Furthermore, experiments also conducted on FF++, DFDCPreview, and Celeb-DF dataset underwent several post-processing situations, such as blurring, compression, and noise. The model surpassed the most advanced models in terms of generalization, achieving a 98% accuracy in cross-domain scenarios. This demonstrates its ability to learn the shared distinguishing textural characteristics in the manipulated samples. These experiments provide evidence that the proposed model is capable of being applied to various situations and is resistant to many post-processing procedures.

3.2.2 Empirical Investigation for texture as a feature

Texture refers to the appearance of the surface characterized by the shape, size, density, and proportionate arrangement of its elementary parts. In computer vision terminology, it is the repeated occurrence of the grey pixel level in the space [114]. An empirical analysis of the fake and real images is done to reveal the differences between textural characteristics. Texturized images are generated for accurate and fake images using a texturized generating algorithm, and it can be seen from that fake images lack texturized details compared to authentic images. This could be because forming fake tampered data samples usually involves three steps, i.e., pre-processing, face generation, and post-processing operation. Post-processing operations are generally done to hide such texture defects; as a result, counterfeit images tend to have a smoother surface and fewer texture characteristics [13]. Also, looking more closely at Figure 3.2, one finds that manipulated images come from smoother surfaces. Hence, the lack of a texturized surface would be a potential clue for fake image detection.

Figure 3.1 Real and fake Images are shown with their texturized images. Texturized images are generated from the images using the texture-based algorithm.

Figure 3.2 Fake images on a closer look showing that fake images tend to have smoother surfaces

3.2.3 Proposed Methodology

The model comprises two components(Figure 3.3): texture Architecture and dual-branches cross-attention vision transformer. Texture architecture consists of two branches which serve as inputs to the parallel branches of the cross-attention vision transformer.

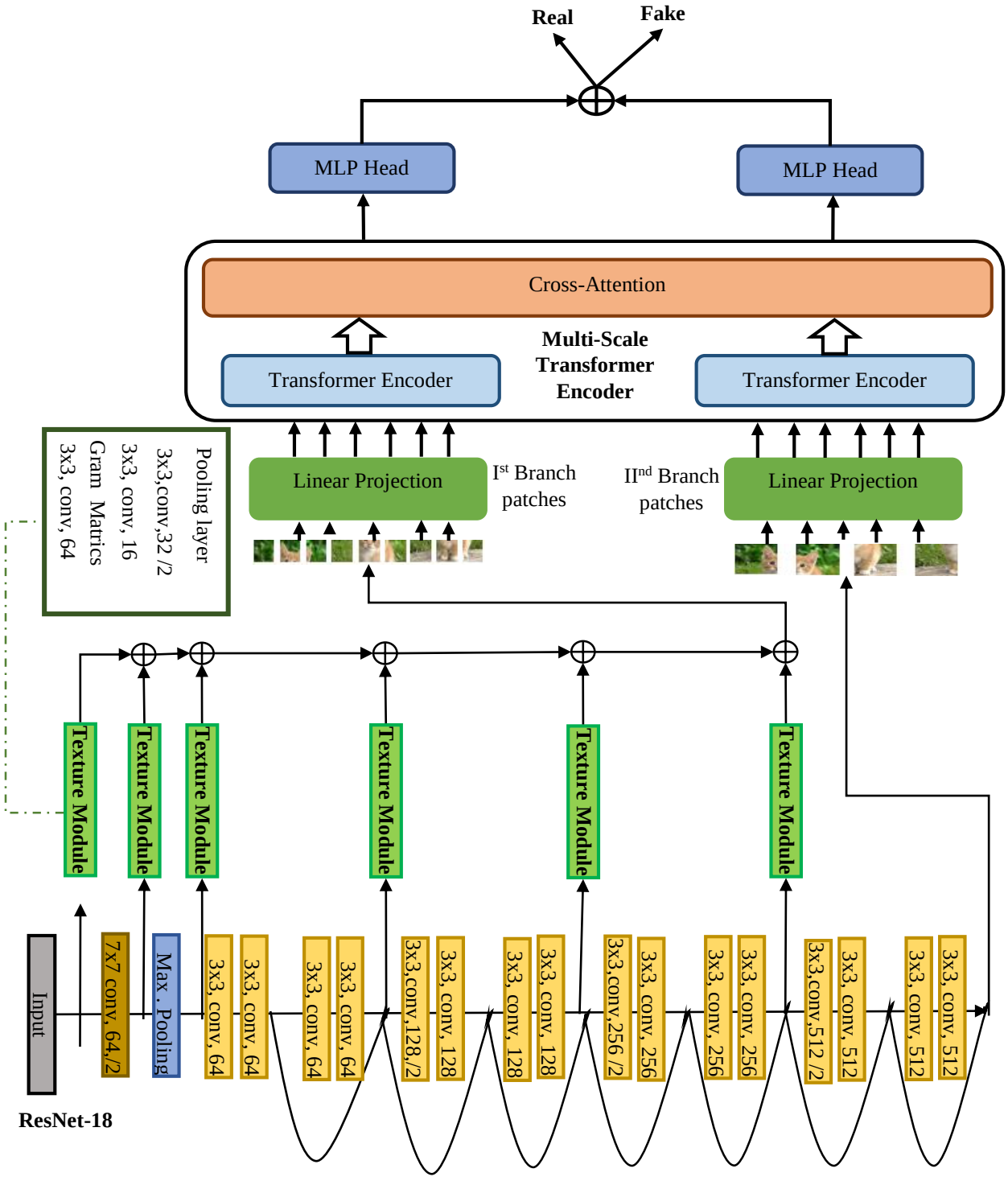


Figure 3.3: Proposed model consisting of texture module and ResNet serving as an input to dual-branch vision transformer with cross-attention mechanism

3.2.3.1 Texture Architecture

Texture architecture constitutes resnet-18 architecture as a backbone, and the texture block is computed at the input and before every down-sampling operation incorporating global texture at various levels. The texture block consists of convolutional layers and gram matrices. Gram matrices are used to extract texture correlation, while convolution layers is then applied to

enhance the representation, and pooling layers are used to align the computed features with the ResNet backbone features for the next level. The global Texture module is computed at multiple semantic levels before to each ResNet down sampling operation to model long-range texture features [115]. The main backbone of ResNet-18 learns the conventional features representation of the input images at various levels with a skip connection. It improves the gradient flow between the layers of the multi-scale features while the Texture block learns the global textures' semantic information at various scales.

Gram matrices as Texture features: Within a model, the texture is represented by the correlations among the features map responses in various model layers [116]. The Gram matrix quantifies the correlation between different feature map responses over different layers. These correlations, which are determined up to a constant factor, are represented by the gram matrices. Gram matrices $G^l \epsilon R^{N_l \times N_l}$ computes linear dependence between the layers:

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l \tag{3.1}$$

Above equation represents the gram matrix G_{ij}^l which is the inner product between the ith and jth feature of layer l, where F^l represents the lth feature map vectorized representation and F_{ik}^l represents the kth activation of the ith filter at position k in layer l. A texturized model, as defined, does not consider spatial information and is distinguished by the correlations among the feature maps. The texture is generated by computing gram matrices, which are calculated in the model prior to each downsampling operation of the ResNet. These matrices are then concatenated and used as input to the cross-attention mechanism of the vision transformer.

3.2.3.2 Dual branch Cross-Attention Vision Transformer

The vision transformer, free of inductive biases, is known for capturing long-range, global relationships between the pixels, courtesy of their self-attention mechanism and capacity for holding semantic information. The proposed architecture uses two parallel branches, and patches input into these branches are comparable in scale. The model takes the texture architecture's input, and then positional embedding is added into each patch, including the CLS token, to embed positional information into the model. Then, these tokens are passed through the stacked transformer encoder. Each transformer encoder contains a dual branch and is composed of Multi-headed self-attention(MSA) followed by the feed-forward Network(FFN) [117]. FFN includes two layers of the multi-layer perceptron, and the GELU non-linear layer is applied at the end of the first layers. Layer-Norm (LN) is used at the end of every block, with

residual skip-connection applied after every block. The input to the $\mathbf{x}_0$ ViT and l^{th} processing of the transformer encoder can be written as:

$$\mathbf{x}_0 = [\mathbf{x}_{\text{class}}||\mathbf{x}_{\text{patch}}\text{E}] + \text{E}_{\text{pos}} \qquad E \in R^{P^2.C\ x\ D}, E_{pos} \in R^{(N+1)\ x\ D} \tag{3.2}$$

$$\dot{\text{z}}_l = \dot{\text{z}}_{l-1} + MSA\big(LN(\mathbf{x}_{l-1})\big), \qquad l = 1. \ldots . . L \tag{3.3}$$

$$\mathbf{x}_l = \dot{\text{z}}_l + FFN\big(LN(\dot{\text{z}}_l)\big), \qquad l = 1 \ldots \ldots L \tag{3.4}$$

where is $\mathbf{x}_{clsemb} \in \text{R}^{1\times\text{C}}$, $\mathbf{x}_{patchemb} \in \text{R}^{\text{N}\times\text{C}}$ $\mathbf{x}_{posemb} \in \text{R}^{(\text{N}+1)\times\text{C}}$ and E are the cls, patch positional and embedding tokens, respectively(C, N and D are the embedding's dimension, the number of the tokens and the dimensions of the flattened tokens respectively). Afterwards, the CLS token of one branch, which has learned the abstract information, acts as a token query to interact with the patch tokens of the other branch through an attention mechanism resulting in multi-scale features. Similarly, the CLS token interacts with the patch tokens of the other branch. The cross-attention mechanism is represented in the subsequent equations where $\mathbf{x}$ is the input to MSA(Multi-headed self-attention module:

$$\text{ɣ}^1 = [\mathbf{x}^1_{cls}||\mathbf{x}^2_{patch}] \qquad \mathbf{x}^1 \in token\ I^{st} branch, \quad \mathbf{x}^2 \in tokens\ II^{nd}\ branch \tag{3.5}$$

$$q = \text{ɣ}^1_{cls} W_q, \qquad [k, v] = \text{ɣ}^1 W_{kv} \qquad W_q, W_{kv} \in \text{R}^{\text{D x 3D}_\text{h}} \tag{3.6}$$

$$A = softmax\left(\frac{qk^T}{\sqrt{D_h}}\right) \qquad A \in R^{NxN} \tag{3.7}$$

$$SA(\text{ɣ}^1) = Av \tag{3.8}$$

$$MSA(\text{ɣ}^1) = [SA_1(\text{ɣ}^1); SA_2(\text{ɣ}^1); \ldots . ; SA_k(\text{ɣ}^1)]W_{msa} \qquad W_{msa} \in R^{k.D_h xD} \tag{3.9}$$

$$\acute{\text{y}}^1_{cls} = \mathbf{x}^1_{cls} + MSA(LN([\mathbf{x}^1_{cls}||\mathbf{x}^2_{patch}])) \tag{3.10}$$

Where q, k, and v are the query, key, and value, respectively, n+1 is the number of patches, d is the model dimension, k is the number of heads, and Dh(d/k) is the head dimension. $\text{W}_{\text{q,}}$ $\text{W}_{\text{kv,}}$ and W_{msa} are the learnable parameters for the query, key, value, and MSA, respectively. Following fusion with other branch tokens, the CLS token at the next transformer encoder engages with its patch tokens once more. Here, it imparts knowledge from the other branch to its patch tokens, enhancing each patch token's representation. Then, these tokens are passed through the Layer Norm to MLP(Multi-Layer Perceptron) for parameter learning:

$$\ddot{\text{y}}^1 = \acute{\text{y}}^1_{cls} + \mathbf{x}^1_{patch} \tag{3.11}$$

$$\check{z} = MLP(LN(\ddot{y}^1)) \quad (3.12)$$

Finally, these classification tokens are concatenated for the final predictions.

Algorithms 1: Tex-ViT for Deepfake classification

Parameter Initialisation:

- Input: I = {I_1, I_2,…..I_n} be the set of images, and L = {0, 1} be the set of labels, 0 being the real and 1 being the deepfake image
- n is the size of the dataset
- Split I into three subsets for 70% training, 15% validation, and 15% testing.

1: **For 1 to 100 epochs, do**
2: Input image I into ResNet for feature extraction.
3: Compute the texture using texture block before every down sampling operation in ResNet and keep concatenating them.
4: ResNet CNNs and texture features calculated at step 1 and step 2 are fed into the dual branch of the vision transformer.
5: Split the features into patches (fixed sizes) and flatten them at each branch.
6: With these image patches flattened, create linear embeddings in lower dimensions.
7: Include positional embeddings with CLS token.
8: Feed the sequence into the transformer encoder at each branch.
9: Create tokens by querying the CLS token of the I^{st} branch with patch tokens of another branch and vice-versa.
10: Concatenate the tokens of both branches for classification.
11: Train the model end-to-end and update weights using the Adam optimizer.
12: Evaluate the validation set and save the weights of the model that performs well.
13: **end for**
14: Load the weights of the model saved at step 11.
15: Evaluate the performance on the test set

3.2.4 Experiments

This section will detail the choice of training hyper-parameters, different datasets, the choice of face extractor, and the different experiment scenarios conducted.

3.2.4.1 Experimental Settings

The starting learning rate is set to 0.01. Adam optimizer is used to update the model's parameters, as it is being widely used and took less time for parameter updation in my case. The batch size is taken as 64, due to the memory constraints at my computer systems. Each experiment is run for 100 epochs, as the model performance hits the saturation point after this specific number of epochs. The experiments are run for 24GB NVIDIA TITAN RTX GPUs.

3.2.4.2 Dataset Pre-Processing

Three deepfake datasets are used to evaluate the models: Celeb-DF, DFDC-Preview, and Faceforensics++, as these are being widely used and also the state-of-the-art dataset in the current scenarios. These datasets consist of short facial videos from which the frames are extracted utilizing the RetinaFaceResNet50 face extractor, as the RetinaFaceResNet50 face extractor has a lesser failure rate than MTCNN. One hundred frames are extracted from each video. For a DFDC and Celeb-DF dataset, most faces have aspect ratios of [1, 1.5] and heights between [151, 200] pixels. Lastly, an FF++ has a size of [151, 200] and an aspect ratio of [1,1.5](Table 3.1). Different GAN images are also used to evaluate the model. Fake images: ProGAN and StyleGAN images and real image datasets: CelebA-HQ, CelebA, and FFHQ are downloaded from their respective repositories. StarGAN and STGAN images are generated by executing the code from their GitHub repositories.

Table 3.1 Details for the training, validation, and testing dataset with their resolutions

Dataset	**Training Set**	**Validation set**	**Testing set**	**Image Resolution**
FF++(DeepFakes)	8k real, 8k fake image frames.	2k real, 2k fake image frames.	2k real, 2k fake image frames.	128x128
FF++(face2face)	8k real, 8k fake image frames.	2k real, 2k fake image frames.	2k real, 2k fake image frames.	128x128
FF++(Faceswap)	8k real, 8k fake image frames.	2k real, 2k fake image frames.	2k real, 2k fake image frames.	128x128
FF++(Neural Texture)	8k real, 8k fake image frames.	2k real, 2k fake image frames.	2k real, 2k fake image frames.	128x128
DFDCPreview	10k real, 10k fake image frames.	1.5k real, 1.5k fake image frames.	1.5k real, 1.5k fake image frames.	128x128
Celeb-DF	10k real, 10k fake image frames.	8k real, 8k fake image frames.	8k real, 8k fake image frames.	128x128
CelebA-HQ & ProGAN	10k(CelebA-HQ) & 10k(ProGAN)	1.5k(CelebA-HQ) & 1.5k(ProGAN)	1.5k(CelebA-HQ) & 1.5k(ProGAN)	1024x1024

CelebA-HQ& StyleGAN	10k(CelebA-HQ) & 10k(StyleGAN)	1.5k(CelebA-HQ) & 1.5k(StyleGAN)	1.5k(CelebA-HQ) & 1.5k(StyleGAN)	1024x1024
FFHQ and StyleGAN	10k(FFHQ) & 10k(StyleGAN)	1.5k(FFHQ) & 1.5k(StyleGAN)	1.5k(FFHQ) & 1.5k(StyleGAN)	1024x1024
CelebA & StarGAN	10k(CelebA) & 10k(StarGAN)	1.5k(CelebA) & 1.5k(StarGAN)	1.5k(CelebA) & 1.5k(StarGAN)	128x128
CelebA & STGAN	10k(CelebA) & 10k(StarGAN)	1.5k(CelebA) & 1.5k(StarGAN)	1.5k(CelebA) & 1.5k(StarGAN)	128x128

3.2.4.3 Data Augmentation and Scaling

Usually, a vision transformer needs a lot of data for training to perform at par with the CNN model, as shown by the original ViT model [118]. However, with the rich set of careful data-augmentation techniques, DeiT [119] has shown promising results with fewer data and comparable performance with the CNN model. For the proposed model, various data-augmentation techniques have been used, which include rand augmentation [120], cut mix [121], and mixup [122], along with the random-erasing [123] and drop path regularisation model techniques to improve the overall results of the classifications.

3.2.4.4 Experiments on the cross-manipulation settings for the Faceforensics++ dataset

Experimentation has been performed on the various categories of Faceforensics++ [124]. The model is trained on one class of FF++ and tested on the same as on other varieties of FF++. Weights of the models that perform well on the validation are saved for evaluation on the test dataset. Even though many state-of-the-art models have been introduced recently, it is still difficult to compare them fairly. This is partially because there is a dearth of publicly available codes for the models and training procedures that are unavailable to the research community. Consequently, we advocate for the community to embrace open-source software and for the generation strategies of large-scale datasets to be evaluated independently of the model's success. These kinds of actions are essential to maintaining equity and encouraging further developments in this area. For the comparison, four models are taken into consideration:

a) MesoInception-4 model [125].
b) Capsule Network [97].
c) Combining Vision Transformers and Efficient Net(E-ViT) [126].
d) UCF [127].
e) IID [128].
f) SIA[21].

g) UIA[22].

Code for these models has been taken from their GitHub repository and customized according to the dataset, and more evaluation metrics have been added for comprehensive evaluation. These models have been trained on one category of manipulation and tested on other categories of manipulation of FF++. These experiments are necessary to test the performance of the models against various manipulations, which is necessary to validate the detector's generalization abilities.

Table 3.2 represents the score of the various models when trained on the DeepFakes category and tested on the various categories of the FF++. Various models score very well for the same type of manipulation, and few even score perfectly, or it can be said that they are overfitting. These overfitted models' performance degraded heavily when asked to classify other categories of FF++. MesoNet and CapsuleNet, which are considered the most advanced models for detecting forgery, get an accuracy score of approximately 50% when it comes to identifying manipulated images in the FF++ dataset. This is due to their limited ability to learn just traditional features from convolutional neural networks (CNN), which is insufficient for effectively detecting cross-manipulation scenarios. Among the latest techniques, the Uia and Ucf approaches, which are renowned for identifying shared characteristics among different types of manipulation, do not meet the necessary criteria for an effective deepfake detector that can be applied universally. Our model outperforms the other models, with an accuracy of 72%, specifically for the DF category of the dataset. The majority of models struggle to accurately categorize the face-swap category in FF++, and a small number of models performed below 50% accuracy. Our model's performance demonstrates that texture is a consistent characteristic that remains present across different types of facial alterations.

Table 3.3 represents the scores for the models trained on the face2face categories and tested on the other categories of the FF++ dataset. Once again, the different models excessively suit the face2face category, and their effectiveness significantly declines when applied to other categories of manipulation. The CapsuleNet approach exhibits the poorest performance among all methods for cross-manipulation. When trained on the face2face category, all models perform significantly better than in the previous scenarios in cross-manipulation settings. This is likely because face2face involves facial re-enactment techniques that use video frames to create a highly detailed reconstruction of the face, taking into account different lighting conditions and facial emotions. These inherent characteristics enable the models to learn the implicit features of the manipulation, which in turn aids in cross-manipulation scenarios. Uia is a highly effective model with an accuracy score of approximately 70%. This model utilizes

an unsupervised technique and incorporates an inconsistency-aware module to detect discrepancies among the patch-level data. The Sia approach has poor performance due to its heavy reliance on the attention mechanism, which can occasionally miss tiny abnormalities in the manipulations. Furthermore, several models exhibited subpar performance without any notable improvements.

Conversely, Tex-ViT maintains the general characteristics, avoids overfitting when the training and testing data are from the same distribution, and generalizes well to alternative distribution categories. The manipulation achieved scores of 73% and 71% for the DF and NT categories, respectively. The FS category score for the manipulation has increased compared to the score in the preceding table.

Table 3.4 represents the performance of the model when trained in the FS category and tested on additional variations of FF++. All of the models exhibit overfitting for the same category and demonstrate inadequate performance for the other manipulation categories. This is mostly attributed to the Faceswap construction technique, which utilizes facial landmark points to generate a 3D template. This template is then projected onto the target shape in order to minimize the disparity between the projected shape and the landmark points. The meticulous process, which involves precise shape mixing and color correction, poses a greater challenge for the detector to identify accurately. The Ucf and Uia model exhibits poor performance, with an accuracy score of approximately 50% when trained on the face swap manipulation. This highlights the fragility of the model under different circumstances. Sia has poor performance, although IID demonstrates slightly higher performance compared to previous models. This improvement can be attributed to the model's capability to learn both implicit and explicit characteristics. However, the overall performance still falls well short of the desired score. MesoNet and CapsuleNet exhibit substandard performance; however, the transformer-based model E-ViT demonstrates significantly superior performance owing to its hybrid model structure, which combines CNN and ViT to capture long-range dependencies. Our model encounters difficulty in accurately identifying this category of manipulation, with an accuracy score ranging from 62-67%. This is because the face swap-generating mechanism incorporates intricate generation techniques that our model finds challenging to detect.

Table 3.5 represents the final category of manipulation, wherein the model is trained using the NT category of manipulation and subsequently tested on the remaining categories. The models trained in this specific category of manipulation exhibit a commendable level of performance when compared to other categories of manipulation. The NT generation mechanism utilizes the texturing technique to understand the inherent characteristics of the data sample, enabling it to

excel in the categories of manipulation. All the models exhibit comparable performance when trained on the NT category of manipulation, indicating that texture is one of the invariant properties that assist in their ability to perform in diverse types of manipulation. The majority of the models achieved a score above 70% when tested on the DF category of manipulation. With the exception of Ucf and IID, all models in the face2face category perform well and even outperform the former category. However, the manipulation category (FS) consistently shows poor performance. Furthermore, the issue of overfitting persists in the NT category. Once again, our model outperforms the other models. The state-of-the-art (SoTA) model achieves an accuracy score of 77% in face-to-face manipulation and 76% in Deepfake manipulation categories, demonstrating higher performance compared to other models.

So, other models score almost perfectly when training and testing come from the same distribution but fail to generalize well for other distributions. In contrast, based on the texture module and cross-attention mechanism, our model performs well in almost various manipulations, effectively proving that texture is a potential feature that persists among different manipulations. However, every performance suffers from the FS category of manipulation of FF++. Figure 3.4 represents the ROC curves of various models for images trained on face2face and testing on different types.

3.2.4.5 Experiments on the cross-domain settings for GAN dataset images

Extensive testing has been done on various GAN-generated images. Complete image synthesis, such as StyleGAN, ProGAN, and Attribute manipulation images of StarGAN and STGAN, has been considered for analysis. High-resolution authentic images are taken from the FFHQ, CelebA-HQ, while low-resolution images are taken from the CelebA. Five real and fake image datasets have been designed for the fair and comprehensive evaluation: CelebA-HQ ProGAN, CelebA-HQ StyleGAN, FFHQ StyleGAN, CelebA StarGAN, and CelebA STGAN. To compare the results of these datasets, four state-of-the-art models have been considered:

a) Xception with depth-wise separable convolution(Xception-Net) [129].
b) CNN's generated images are easy to spot now(CNN-Net) [130].
c) Efficient-Net [131].
d) UCF [127].
e) IID [128].
f) SIA[21]
g) UIA[22]

Once again, code has been extracted from the GitHub source and tailored to suit the dataset used for these models. Additionally, additional evaluation criteria have been incorporated to facilitate comparison evaluation. The model exhibits exceptional performance and surpasses the scores achieved by several state-of-the-art methodologies. Similarly, when it comes to FF++ models designed for cross-forgery, their current level of performance is still rather distant from the optimal score required for their practical implementation in real-world situations. Table 3.6 represents the score of these models on various datasets. It is evident that the scores of different models vary between datasets; they excel in one dataset but do poorly in another. It is evident that when the testing dataset contains both fake and actual images, models can promptly recognize them due to their training on these types of images, but they face difficulty in identifying the other class. For instance, in the initial row, a model that was trained on CelebA-HQ ProGAN and tested on CelebA-HQ StyleGAN, most models can accurately recognize the CelebA-HQ category, as indicated by the precision value. However, they struggle to correctly classify the other category, as indicated by their recall value. ProGAN and StyleGAN employ distinct modification techniques, leading to disparate feature spaces.
Consequently, models trained in one category exhibit suboptimal performance on the other. There is a noticeable similarity between the CelebA-HQ and FFHQ real datasets in terms of their characteristics, as a model trained on one dataset can achieve good performance on the other. For instance, a model that has been trained using the CelebAHQ StyleGAN has strong performance when combined with the FFHQ StyleGAN. Among these scenarios, the IID model has performed exceptionally well. This is because the datasets used are completely synthetic, and the IID model is particularly adept at identifying implicit inconsistencies within the feature space. Tex-ViT consistently beats other models across a range of settings. The evidence indicates that the model acquires the typical distinguishing characteristics, including the overall texture that remains consistent despite different types of modification, whether it is generating a full image or altering various qualities. The recall metric score indicates that all the models struggle to identify the ProGAN images accurately. Remarkably, nearly all the models awarded a flawless rating to the photos that were trained using StarGAN and assessed using STGAN, and vice versa. This could potentially elucidate the rationale behind the utilization of comparable counterfeiting techniques in their production. Figure 3.5 represents the ROC curves of various models for images trained and tested on different GAN image datasets.

3.2.4.6 Experiments on the in-domain settings for various post-processing operations of FF++, Celeb-DF, and DFDCPreview dataset

One of the limitations of the different models is that they are not robust enough for various post-processing operations like blurring, compression, the addition of noise, scaling, translation, etc. [132]. To demonstrate the model's robustness, the primary post-processing operations on the test dataset are blurring, compression, and addition of noise. For blurring the images, Gaussian blur PyTorch transformation has been used with a kernel size of 7x7 and sigma 25; for the addition of noise, zero mean and standard deviation of 0.2 have been designed, and finally, for the compression, quality of the images has been degraded by 3x times(Figure 3.6). Models have been trained on the regular images but tested on the images undergoing various post-processing operations. Three primary deepfake datasets have been considered for evaluation. Again, four models have been used for the comparative assessment:

a) MesoInception-4 model [125].
b) Capsule Network [97].
c) CNN's generated images are easy to spot now(CNN-Net) [130].
d) UCF [127].
e) IID [128].
f) SIA[21]
g) UIA[22]

Testing dataset does not undergo any post-processing operations. The first row of the table represents the results when the image has not undergone any processing operations; in that case, all the models have performed perfectly(Table 3.7). Without undergoing post-processing operations, every model in the FF++ dataset overfits and ultimately performs poorly for various post-processing activities. When comparing FF++ to DFDCPreview and Celeb-DF, FF++ models perform better. However, the score for the Celeb-DF dataset is lower, possibly because this dataset contains high-quality images that closely resemble genuine images, with constant lighting and texture, making it slightly more challenging. MesoNet has gotten the lowest amount compared to other competitive approaches, mostly due to its poor capacity to capture the traditional CNN features. The IID approach outperforms the other model, achieving a near-perfect score for the DFDCPreview dataset. The model's capacity to concentrate on the inconsistent characteristics connected to identity makes it particularly useful for face-swapping manipulations.

Testing dataset undergoes blurring operations: Here, the second row for each dataset represents the score for blurring operations. The performance has not degraded significantly

for the blurring operation, showing that the blurry images retain the manipulated artifacts of the non-blurry images. However, the images have been blurred to a significant extent. The performance of the models in the case of FF++ decreased by at least 12% when exposed to blur operations. For the DFDCPreview and Celeb-Df datasets, the performance showed a slight decline, around 3-6%, with a few exceptions for certain models. MesoNet exhibits a more pronounced decrease in performance for Celeb-DF and DFDCPreview because of their heavy reliance on traditional CNN capabilities. The IID model has seen a decline in performance, specifically for the FF++ dataset, but there is a slight reduction in performance for the other dataset. Sia and Uia exhibit superior resilience compared to other models when exposed to blurring operations. Other models are significantly impacted. Our model has improved its ability to withstand and recover from challenges, as indicated by a mere 1-2% decrease in performance for the DFDCPreview and Celeb-Df datasets. Moreover, almost 12% of the data samples continue to display discriminatory artifacts even after undergoing substantial blurring.

Testing dataset undergoes compression: The quality of the photographs was reduced by treble as a consequence of the compression. Compressing samples for the DFDCPreview and Celeb-DF datasets has a minimal effect on the models' performance, suggesting that the altered artifacts are not significantly affected by the reduction in size. The performance of models such as UCF, CNN-Net, Sia, and IID in the FF++ dataset has been significantly impacted, indicating that these models are not specifically designed for compression circumstances. Additionally, the dataset contains a variety of manipulations, which exacerbates the model's complication in comprehending the extensive distribution of features. The models' efficacy was minimally affected by the Celeb-DF dataset, while the DFDCPreview dataset had a slightly greater impact. Compression typically entails a reduction in the resolution of data sampling, which affects the smaller details and leads to distortions such as ringing, banding, blocking, and halo. The gradients in the smooth portions are significantly impaired by these distortions. Models that concentrate on specific artifacts encounter difficulties when confronted with a diverse array of intricate properties. Models that concentrate on a variety of artifacts at differing levels of detail are more likely to accurately classify intricate feature patterns. Our model prioritizes intricate attributes, commencing with the integration of conventional CNN features at multiple levels and texturing. It employs a cross-attention method to comprehend both global and local details by utilizing the capabilities of transformers. This enables the model to acquire nuanced, intricate characteristics at multiple levels with greater efficacy.

Addition of Noise to the testing dataset: The PyTorch transformation modifies the data samples with noise, which has a mean of zero and a standard deviation of 0.2. The results revealed a

substantial decrease in the scores of all models, with IID, MesoNet CNN-Net, Sia, and Uia scoring as low as 50%. This underscores the susceptibility of these detection methods to the presence of noise. The IID model, which is intended to detect face-swap, has been significantly impacted by the introduction of noise in the FF++ and DFDC Preview datasets. This has resulted in erroneous classification and has rendered the model susceptible to adversarial perturbations. MesoNet is susceptible to a variety of adversarial strategies as a result of a significant decrease in its classification score. The quality of images is reduced by the presence of noise, which masks anomalies or inconsistencies and introduces random variations and patterns. Consequently, accurate or complete feature extraction is impeded. In order to achieve optimal performance on a noisy dataset, a model must either employ the attention mechanism to leverage multi-scale features that can effectively capture both local and global features that are resilient to noise, or employ sophisticated data augmentation techniques that introduce noise to aid in the model's classification learning. By utilizing the latter approach, our model has been able to acquire intricate and resilient features that can withstand a variety of adversarial techniques. When the accuracy of the other model in the DFDCPreview dataset is less than 80%, the Tex_ViT model consistently outperforms it with an accuracy score of 98%. Nevertheless, the model remains susceptible to the introduction of disturbance to a certain extent.

Table 3.2 Models trained on deepfake dataset of FF++ and tested on its different categories. Here, bold values represent the highest score among competitive methods.

Test	Ucf					IID					MesoNet					CapsuleNet					E-ViT					Sia					Uia					Tex-ViT(ours)				
	Pr	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**
Df	0.9989	0.9985	0.9987	0.9987	0.9987	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.999	0.9994	**1.0**	0.9995	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.999	**1.0**	0.9995	**1.0**	0.9995	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.9829	0.982	0.9824	0.9985	0.9825
F2F	**0.791**	0.263	0.263	0.697	0.6023	0.7790	0.2715	0.4026	**0.7218**	0.5972	0.6667	0.1455	0.2389	0.6345	0.5365	0.6239	0.151	0.2431	0.6159	0.53	0.755	0.165	0.2708	0.6479	0.5557	0.6464	0.3455	0.4503	0.6345	0.5783	0.7286	0.192	0.3039	0.6508	0.5602	0.7242	**0.6126**	**0.6637**	0.7048	**0.6948**
FS	0.224	0.018	0.033	**0.711**	0.5396	0.4637	0.016	0.0309	0.5621	0.4987	0.2168	0.0155	0.0289	0.2962	0.4797	0.3764	0.0335	0.0615	0.4207	0.489	0.392	0.265	0.4964	0.4700	0.4927	0.3245	0.0865	0.1365	0.3999	0.4532	0.4652	0.0435	0.0709	0.5062	0.49625	**0.6952**	**0.040**	**0.07564**	0.6547	**0.6291**
NT	**0.8164**	0.238	0.368	0.6693	0.5923	0.8765	0.245	0.3829	0.7263	0.6025	0.7703	0.208	0.3275	0.6972	0.573	0.7132	0.25	0.3702	0.6639	0.5747	0.789	0.257	0.3877	0.6829	0.5942	0.6616	0.5025	0.5711	0.6707	0.6227	0.7794	0.304	0.4374	0.7097	0.609	0.7248	**0.6989**	**0.6541**	**0.7958**	**0.7028**

Table 3.3: Models trained on the face2face dataset of FF++ and tested on its different categories. Here, bold values represent the highest score among competitive methods.

Test	Ucf					IID					MesoNet					CapsuleNet					E-ViT					Sia					Uia					Tex-ViT(ours)				
	Pr	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**
Df	0.8531	0.3165	0.4617	0.7533	0.631	**0.8638**	0.387	0.5345	0.8117	0.663	0.7459	0.367	0.4919	0.7085	0.621	0.8161	0.253	0.3862	0.6304	0.598	0.6751	0.2775	0.3933	0.6733	0.572	0.6217	0.544	0.5802	0.6560	0.6065	0.7305	0.6305	0.6768	0.777	0.6989	0.7365	**0.664**	**0.6983**	**0.7958**	**0.7133**
F2F	0.9975	0.9985	0.9980	0.9986	0.998	0.9995	**1.0**	0.9997	**1.0**	0.9997	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.9960	0.9995	0.9977	0.9999	0.9977	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.9714	0.9875	0.9794	0.9986	0.9793
FS	0.644	0.124	0.2079	0.5282	0.5277	0.5780	0.1315	0.2143	0.6605	0.5177	0.5871	0.266	0.3662	0.6064	0.5395	0.6371	0.151	0.2441	0.566	0.5325	0.6399	0.2390	0.3480	0.5922	0.552	0.5676	0.5015	0.5325	0.6079	0.5597	**0.7120**	0.4315	0.5373	0.6473	0.6284	0.6952	**0.5982**	**0.6433**	**0.6987**	**0.6593**
NT	0.8092	0.314	0.4524	0.6827	0.62	0.7891	0.378	0.5111	0.7877	0.6384	0.6937	0.3375	0.4541	0.6843	0.5942	**0.8122**	0.2855	0.4224	0.6666	0.6097	0.7209	0.4855	0.5802	0.7106	0.649	0.6741	**0.694**	0.6839	0.7217	0.6792	0.7523	0.638	0.6904	0.7949	0.7139	0.7604	0.689	**0.7229**	**0.8152**	**0.7360**

Table 3.4: Models trained on the face swap dataset of FF++ and tested on its different categories. Here, bold values represent the highest score among competitive methods.

Test	Ucf					IID					MesoNet					CapsuleNet					E-ViT					Sia					Uia					Tex-ViT(ours)				
	Pr	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**	**Pr**	**Re**	**F1**	**AUC**	**Acc**
Df	0.371	0.0305	0.0564	0.4577	0.4894	0.7414	0.3885	0.5098	**0.7262**	0.6226	0.5892	**0.522**	**0.5535**	0.6321	0.579	0.5375	0.0895	0.1534	0.5415	0.5065	0.6872	0.311	0.4282	0.6712	0.5847	0.6432	0.274	0.3843	0.6093	0.5610	0.5309	0.253	0.3427	0.5789	0.5147	**0.7746**	0.2486	0.3764	0.6847	**0.6248**
F2F	0.7403	0.1625	0.2665	0.5369	0.5527	0.6943	0.4315	0.5322	**0.6964**	**0.6207**	0.6216	**0.5455**	**0.5811**	0.6354	0.6067	0.6383	0.1985	0.3028	0.5568	0.5430	0.6700	0.3005	0.4149	0.573	0.5762	0.7391	0.35	0.4751	0.6554	0.6132	0.6722	0.4605	0.5465	0.6516	0.618	**0.7546**	0.2283	0.3505	0.6446	0.6078
FS	0.9969	0.9949	0.9949	0.9949	0.995	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.9853	0.9755	0.9804	0.9982	0.9805
NT	0.6137	0.058	0.105	0.4929	0.5107	0.6113	0.2855	0.3892	0.5926	0.552	0.5903	0.539	0.5635	0.6205	0.5825	0.5342	0.1365	0.2174	0.5350	0.5087	0.6246	0.2355	0.3420	0.5574	0.547	0.5770	0.219	0.3175	0.5467	0.5292	0.5275	0.2585	0.3469	0.5181	0.5135	**0.7589**	**0.6378**	**0.6931**	**0.7088**	**0.6728**

Table 3.5: Models trained on NT dataset of FF++ and tested on its different categories. Here, bold values represent the highest score among competitive methods.

Test	Ucf					IID					MesoNet					CapsuleNet					E-ViT					Sia					Uia					Tex-ViT(Ours)				
	Pr.	**Re.**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**	**Pr.**	**Re**	**F1**	**AUC**	**Acc**	**Pr.**	**Re**	**F1**	**AUC**	**Acc**	**Pr.**	**Re**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**
Df	**0.8814**	0.613	0.7231	0.7860	0.7652	0.7941	0.729	0.7601	**0.8542**	**0.77**	0.7347	**0.8145**	**0.7725**	0.8287	0.7602	0.7619	0.616	0.6812	0.7801	0.7117	0.7894	0.6505	0.7133	0.8325	0.7385	0.7236	0.6335	0.6755	0.7699	0.6957	0.6966	0.6395	0.6668	0.7383	0.6805	0.7405	0.8045	0.7714	0.8394	0.7612
F2F	**0.7989**	0.379	0.5146	0.6384	0.642	0.7169	0.6245	0.6675	0.7843	0.6899	0.7276	**0.8615**	**0.7889**	0.8342	0.7695	0.7100	0.5045	0.5898	0.7265	0.6492	0.7583	0.739	0.7485	0.8374	0.7517	0.7369	0.7565	0.7466	0.8053	0.7432	0.7532	0.748	0.7506	0.8183	07514	0.7672	0.7995	**0.7830**	**0.8558**	**0.7785**
FS	0.2403	0.0465	0.078	0.4005	0.4497	0.3039	0.1295	0.1816	0.4045	0.4156	0.5013	0.3745	0.4287	0.5296	0.501	0.4557	0.1675	0.2449	0.4886	0.4837	0.4966	0.2235	0.3082	0.5057	0.4985	0.4798	0.304	0.3722	0.5189	0.4872	0.5543	0.429	0.4836	0.5606	0.542	**0.7072**	**0.5844**	**0.6399**	**0.8148**	**0.6408**
NT	0.9954	0.993	0.9945	0.9972	0.9945	0.9494	0.9955	0.9719	0.9999	0.9712	0.9985	**1.0**	0.9992	**1.0**	0.9992	**1.0**	0.9925	0.9959	0.9978	0.996	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.9645	0.9795	0.9719	0.9971	0.9718

Table 3.6: Models trained and tested on the GAN datasets. Here, bold values represent the highest score among competitive methods.

Train	Test	Xception					Ucf					IID					CNN-Net					Efficient-Net					Sia					Uia					Tex-ViT(ours)				
		Pr.	**Re.**	**F1**	**AUC**	**Acc.**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**	**Pr.**	**Re**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**
CelebA-HQ_ProGAN	CelebAHQ_StyleGAN	**1.0**	0.06	0.113	0.7899	0.53	0.9	0.006	0.0119	.4645	0.50266	**1.0**	0.0033	0.0066	0.6842	0.50166	0.9714	0.1913	0.3212	**0.9708**	0.5957	**1.0**	0.008	0.0158	0.8466	0.504	**1.0**	0.165	0.2837	0.9378	0.5826	0.8	0.0053	0.0105	0.8654	0.502	0.8807	**0.64**	**0.7413**	0.8925	**0.7767**
	FFHQ_StyleGAN	0.904	0.0506	0.0959	0.7546	0.5226	0.652	**0.997**	**0.7884**	0.7777	0.7169	0.8	0.0026	0.0053	0.6048	0.501	0.6098	0.3466	0.4465	0.6533	0.5703	0.7215	0.038	0.0721	0.7205	0.5116	0.7785	0.1546	0.2581	0.7254	0.5553	**0.9666**	0.0193	0.0379	0.7285	0.5093	0.7695	0.6386	0.6980	**0.8091**	**0.7237**
CelebA-HQ_StyleGAN	CelebAHQ_ProGAN	**1.0**	0.024	0.0468	0.6872	51.2	0.471	0.005	0.0105	0.4161	0.4966	0.5	0.5	0.5	0.5468	0.5	0.7889	0.0033	0.0066	0.8363	0.5013	**1.0**	0.006	0.0013	0.8093	0.5033	**1.0**	0.0053	0.0106	**0.8450**	0.5026	0.5	0.5	0.5	0.4708	0.5	0.7134	**0.5445**	**0.6176**	0.6818	**0.6065**
	FFHQ_StyleGAN	0.8625	**1.0**	0.9262	0.9886	0.9203	0.732	0.999	0.8452	0.7735	0.8169	0.8631	**1.0**	**0.9265**	**0.9998**	**0.9206**	**0.9931**	0.9986	0.8619	0.9943	0.84	0.7420	0.9993	0.8517	0.9972	0.8260	0.6105	**1.0**	0.7581	0.9783	0.6810	0.7022	**1.0**	0.8251	0.9965	0.788	0.8409	0.948	0.8912	0.945	0.8843
FFHQ_StyleGAN	CelebAHQ_ProGAN	**0.9806**	0.2366	0.3813	**0.8118**	0.616	0.5	**0.5**	0.5	0.5	0.5	0.6666	0.0013	0.0026	0.6027	0.5033	0.7727	0.012	0.023	0.7799	0.506	0.9032	0.0186	0.0365	0.7952	0.5083	0.6363	0.0046	0.0092	0.5773	0.501	0.9643	0.036	0.0694	0.7862	0.51733	0.8421	0.3626	**0.506**	0.7454	**0.6473**
	CelebAHQ_StyleGAN	0.9946	0.9993	0.9970	0.9998	0.9970	0.996	0.99	0.9933	0.9979	0.9933	1.0	**0.9993**	**0.9996**	**0.9999**	**0.9996**	0.9998	0.9963	0.9963	0.9999	0.9963	0.9963	0.9993	0.9993	0.9999	0.9993	0.9986	0.9973	0.9979	0.9999	0.998	0.9986	0.998	0.9983	0.9999	0.9983	0.9819	0.9793	0.9806	0.9968	0.9807
CelebA StarGAN	CelebA STGAN	0.9993	**1.0**	0.9996	1.0	0.9996	0.999	**1.0**	0.9996	1.0	0.9996	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.9775	0.927	0.9517	0.9882	0.9530
CelebA STGAN	CelebA StarGAN	0.9986	**1.0**	0.9993	0.9999	0.9993	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.9958	0.9493	0.9720	0.9961	0.9727

Table 3.7: Models trained on various datasets and tested under various conditions. Here, bold values represent the highest score among competitive methods.

ti	Testing Dataset	Ucf					IID					MesoNet					CapsuleNet					CNN-Net					Sia					Uia					Tex-ViT(ours)				
		Pr.	**Re.**	**F1**	**AUC**	**Acc.**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc.**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**	**Pr.**	**Re.**	**F1**	**AUC**	**Acc**
FF++	FF++	0.988	0.9915	0.9898	0.9968	0.9897	0.9897	0.9951	0.9923	0.9996	0.9923	0.8322	**0.9998**	0.9084	0.9968	0.8991	0.9925	0.9993	**0.9993**	0.9997	**0.9993**	**0.9998**	0.9996	0.9968	**0.9999**	0.9968	0.9973	0.9985	0.9979	0.9999	0.9979	0.9982	0.9992	0.9986	0.9999	0.9987	0.9368	0.9424	0.9396	0.9882	0.9395
FF++	FF++ Blurry	0.673	0.9282	0.7803	0.8297	0.7386	0.575	0.9975	0.7299	0.8699	0.6304	0.5305	**1.0**	0.6933	0.9819	0.5575	0.628	0.9975	0.7715	0.8788	0.7039	**0.9807**	0.9982	0.8329	0.9831	0.7998	0.7540	0.958	0.8438	0.9336	0.8227	0.7151	**1.0**	0.8338	**0.9854**	0.8007	0.7720	0.9894	**0.8673**	0.9644	**0.8486**
FF++	FF++ Noisy	0.642	0.6445	0.6433	0.6479	0.6415	0.5	0.5	0.5	0.5903	0.5	0.5235	**0.9928**	0.6854	0.7079	0.5445	0.6981	0.046	0.0092	0.5612	0.5013	0.5	0.5	0.5	0.5	0.5	0.5	0.0095	0.0186	0.4979	0.50	**0.7489**	0.1103	0.1923	0.6546	0.5366	0.6425	0.7839	**0.7061**	**0.7409**	**0.673**
FF++	FF++ Compression	**1.0**	0.0025	0.0044	0.5784	0.5011	**1.0**	0.0094	0.0186	0.912	0.5047	0.9262	**0.9868**	**0.9555**	**0.9940**	**0.9541**	**1.0**	0.0032	0.0064	0.6939	0.5016	0.9376	0.0764	0.1419	0.9300	0.5381	0.9973	0.231	0.3751	0.9002	0.6151	0.9992	0.6947	0.8196	0.9914	0.8471	0.9713	0.8651	0.9159	0.9839	0.9206
DFDCPreview	DFDCPreview	**1.0**	0.9996	0.9997	0.9997	0.9998	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.9900	0.9993	0.9947	0.9999	0.9946	**1.0**	0.9993	0.9996	0.9999	0.9996	0.9999	0.9986	0.9989	0.9999	0.999	**1.0**	0.9986	0.9993	**1.0**	0.9993	**1.0**	0.9993	0.9996	0.9999	0.9996	0.9926	0.9893	0.9909	0.9995	0.9910
DFDCPreview	DFDCPreviewBlurry	0.998	0.7926	0.8838	0.9287	0.8956	**1.0**	0.8493	0.9185	0.9940	0.9246	0.9985	0.8913	0.9418	0.9981	0.945	**1.0**	0.748	0.8558	0.9881	0.874	0.9967	0.72	0.8372	0.9965	0.86	**1.0**	0.8273	0.9055	0.9957	0.9136	**1.0**	0.9433	0.9708	0.9999	0.9716	0.9933	**0.9893**	**0.9913**	**0.9992**	**0.9913**
DFDCPreview	DFDCPreview Noisy	0.5	**1.0**	0.6666	0.5447	0.5	0.5	**1.0**	0.6666	0.50	0.5	0.5	**1.0**	0.6666	0.5	0.5	0.4999	0.9986	0.6662	0.7179	0.4996	0.5598	**1.0**	0.6666	0.6015	0.5	0.2727	0.01	0.019	0.4388	0.4916	0.4978	0.99	0.6625	0.6841	0.4956	**0.9788**	0.986	**0.9824**	**0.9985**	**0.982**
DFDCPreview	DFDCPreview Compression	0.757	0.9999	0.8619	0.7940	0.84	0.9375	**1.0**	0.9677	0.9999	0.9666	0.9695	0.9986	0.9839	0.9994	0.9836	0.5468	**1.0**	0.7070	0.9304	0.5856	0.9943	**1.0**	0.7587	0.9938	0.9939	0.9247	0.9986	0.9602	0.9989	0.9586	0.9621	0.9993	0.9803	0.9988	0.98	**0.9980**	0.9986	**0.9983**	**0.9999**	**0.9988**
Celeb-DF	Celeb-DF	0.954	**0.9543**	0.9544	0.9605	0.9544	0.9227	0.9451	0.9338	0.9773	0.9330	0.8222	0.9075	0.8628	0.9259	0.8556	0.9418	0.9462	0.9441	0.9741	0.9439	0.9848	0.9376	0.9406	0.9814	0.9408	0.8483	0.9008	0.8738	0.9431	0.8698	0.9235	0.9298	0.9266	0.9823	0.9264	**0.976**	0.9442	**0.9603**	**0.9928**	**0.9610**
Celeb-Df	Celeb-DF Blurry	0.830	0.9511	0.8865	0.8937	0.8782	0.7483	0.9397	0.8332	0.9174	0.8118	0.6291	**0.9757**	0.7649	0.8822	0.7002	0.8793	0.756	0.8130	0.8979	0.8261	**0.9600**	0.9421	0.8912	0.9574	0.885	0.8191	0.8792	0.8481	0.9189	0.8425	0.8717	0.9345	0.9020	0.9631	0.8985	0.9502	0.9258	**0.9379**	**0.9812**	**0.9387**
Celeb-Df	Celeb-DF Noisy	0.5	**1.0**	0.6666	0.5083	0.50	0.6241	0.5	0.5	0.5427	0.50	0.5	0.5	0.5	0.5267	0.50	0.5	0.5	0.5	0.5696	0.5	0.5	0.5	0.5	0.513	0.50	0.4766	0.09312	0.15558	0.4947	0.4954	0.6924	0.1513	0.2484	0.6218	0.5421	**0.6649**	0.7581	**0.7084**	**0.7365**	**0.6721**
Celeb-Df	Celeb-DF Compression	**0.973**	0.5248	0.6819	0.9124	0.7551	0.9048	0.7524	0.8216	0.9321	0.8366	0.9468	0.334	0.4938	0.9193	0.6576	0.9278	0.5146	0.6620	0.8951	0.7373	0.9344	0.534	0.6881	0.9328	0.758	0.7512	0.9086	0.8228	0.9093	0.8044	0.9192	0.778	0.8427	0.9481	0.8548	0.9203	**0.9252**	**0.9227**	**0.9793**	**0.9226**

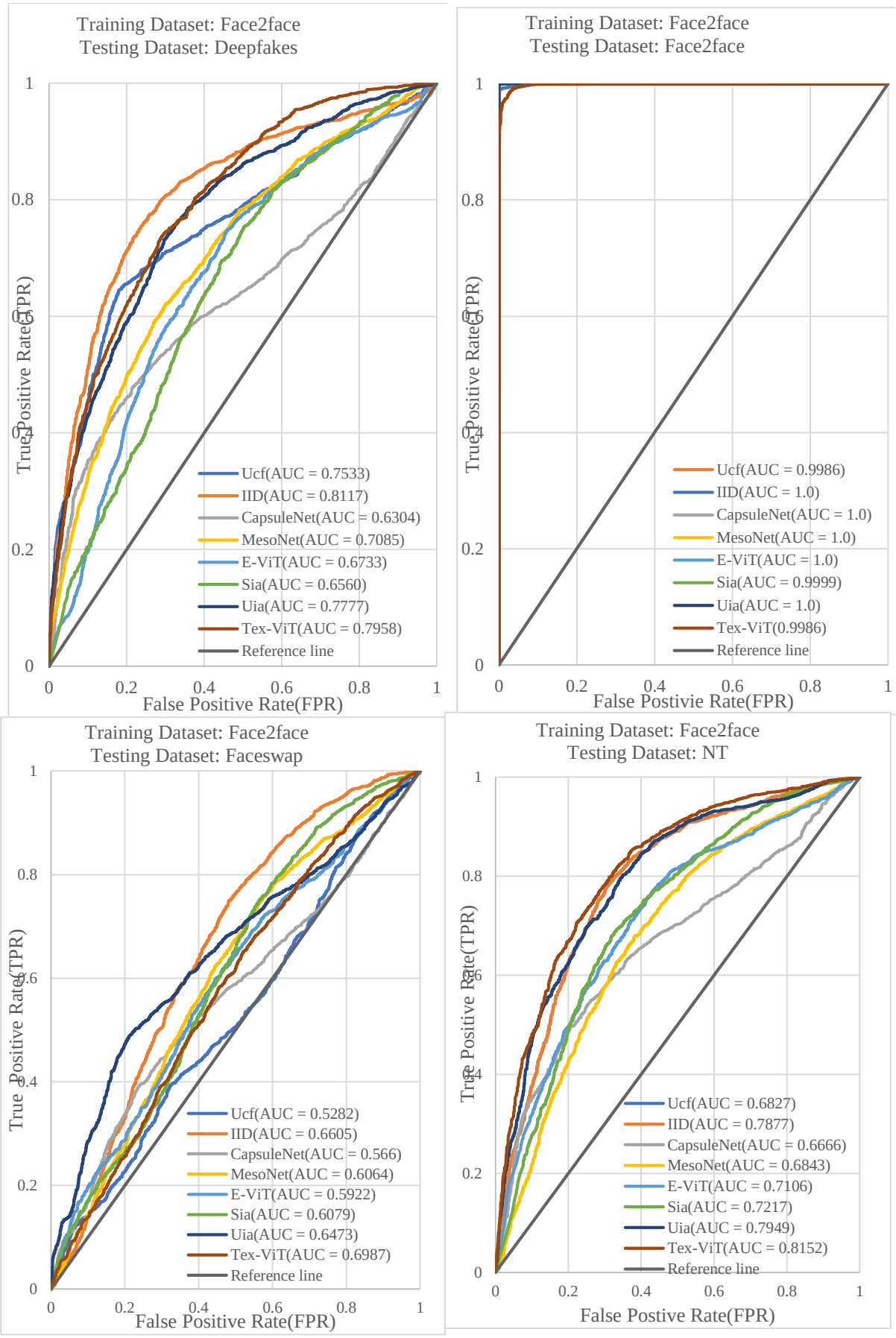


Figure 3.4 ROC curve for the model trained on Face2face dataset and tested on the other categories of FF++

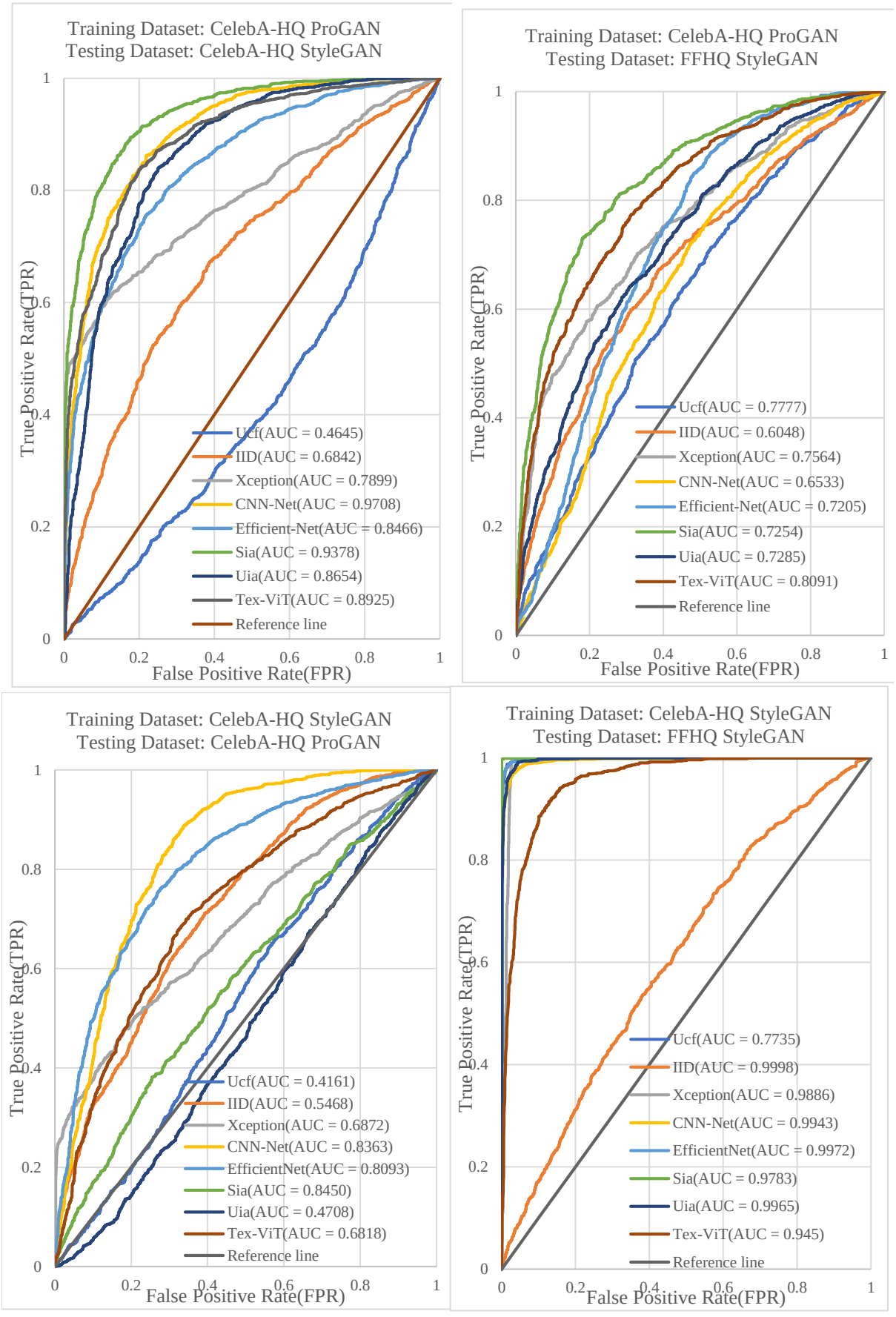


Figure 3.5 ROC curve for the model trained and tested on the different types of GAN images

Original Image Blurred Image Noisy Image Compressed Image

Figure 3.6 Image undergoes different post-processing operations

Table 3.8 : Tex-ViT's computational complexity is compared to well-known computer vision models. The parameter column indicates the accuracy score, number of trainable parameters, and GPU-CPU times taken for each input batch size.

Model	**Accuracy**	**Parameter (millions)**	**CPU time(sec)**	**GPU time(sec)**
Efficient_b7	75.49	66.34	4.07s	0.35s
ResNet152	79.3	58	3.03s	0.309s
ResNext	79.55	81.41	4.61s	0.338s
ConvNext	83.39	88.57	3.62	0.37s
Swin Transformer	75.4	59.96	4.92s	0.41s
XceptionNet	71.6	20.81	2.45s	0.66s
MesoInception	62.3	0.028	**2.1s**	0.72s
CNN-Net	52.47	23.51	1.99s	0.379s
EfficientNetV2	80.5	118.51	3.93s	0.40s
CapsuleNet	60.56	**1.5**	2.49s	**0.154**
Tex-ViT(Our Model)	**84.85**	43	3.06s	1.02s

3.2.5 Complexity Analysis of Tex-ViT

This section compares the computational complexity of the proposed Tex-ViT architecture to well-known computer vision models. The computational considerations are the number of trainable parameters, the accuracy attained on the DF (FF++) dataset, and the CPU and GPU inference times.

Table 3.8 presents the tabular view of the complexity analysis of Tex-ViT against the various standard model of computer vision models. It can be easily seen that Tex-ViT is a reasonably lightweight model compared to various computer vision models, including EfficientNetv1, and has also achieved more excellent performance in diverse scenarios. Figure 3.7 represents the accuracy score on the y-axis and the number of trainable parameters on the x-axis. Regarding the number of trainable parameters, Efficientv2 and ConvNext are on the higher sides and have an accuracy score of around 80%; owning a higher number of parameters tends to learn more discriminative information about the manipulation. Deepfake Detection models like MesoInceptionNet, Capsule Net, and CNN-Net are relatively lightweight, but their performance suffers in diverse scenarios. The execution time is measured in seconds for a batch

size of 32 during inference time. Heavy models take more execution time than lightweight models due to their number of parameters. Xception is a lightweight model with 20 million parameters and an accuracy score of 70%.

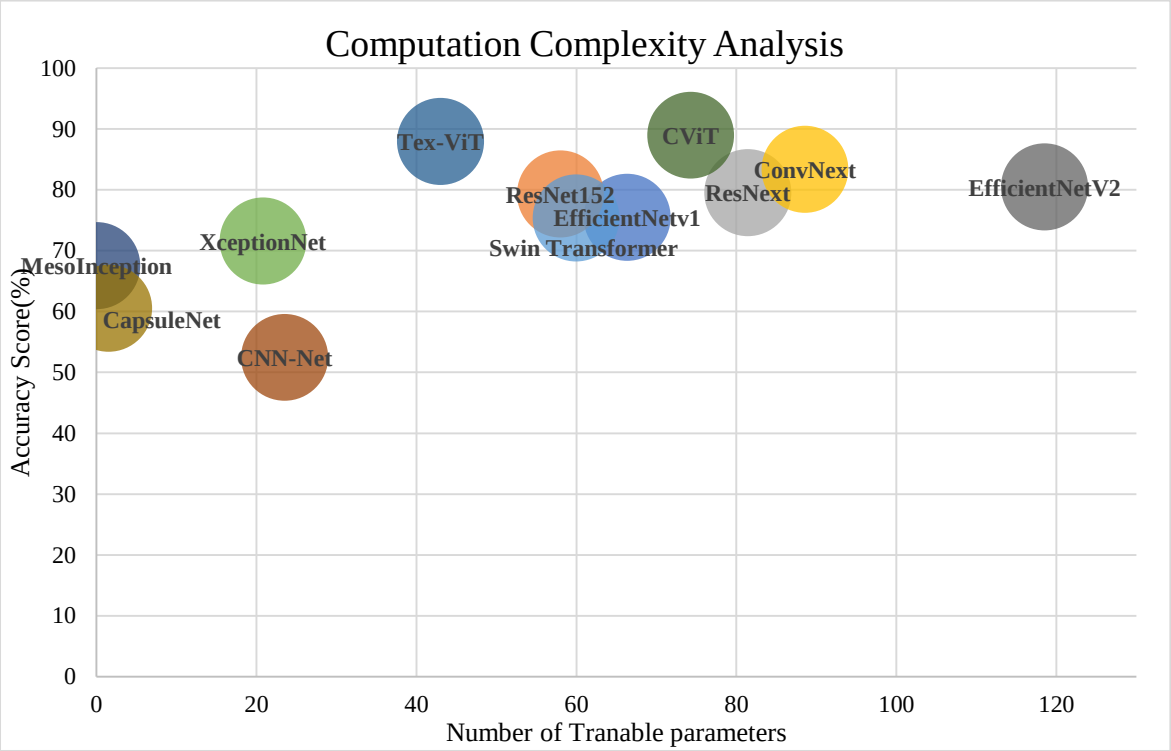


Figure 3.7: Tex-ViT's complexity analysis in comparison to well-known computer vision models. The number of millions of trainable parameters in each model is indicated on the horizontal axis. The accuracy of the DF(FF++) dataset is indicated on the vertical axis.

3.2.1 Conclusion

In this paper, empirical analysis has been done on the human visual and CNN results to demonstrate that human vision is based on the shapes and CNN layers that use texture to identify objects. This finding indicates that texture is one crucial indicator for deepfake detection. Furthermore, it has been seen that texture correlation is not preserved in the case of fake images and that images tend to have smoother surfaces. Inspired by that, the proposed Tex-ViT uses the conventional CNN features using ResNet and texture modules using the features of ResNet. Then, the output of these two parallel branches serves as an input to the dual-branch vision transformers operating on patches with the cross-attention mechanism. Experimental results show that the model performs well in the cross-manipulation categories of FF++ datasets. The evaluation shows that texture is an invariant feature that persists among various manipulation methods, and learning such a feature would eventually result in a good performance for the model. However, the model has a low score in the FS category of FF++. Experimentation is also done on the different types of GAN image datasets and outperforms

the other state-of-the-art models. It again shows the model's superior learning abilities for different feature spaces. Experimentation is done on various post-processing image scenarios, and it was found that the model is robust enough for different adversarial operations. However, the model needs to improve its score for compressed scenarios, but its score is still better than the other SoTA models. These experiments show that the model learns the common discriminative features that persist along several fake images.

Future work would involve improving the model's accuracy for the FS manipulation category of FF++, which could incorporate additional modules to enhance the learning of features. Also, improving the model's robustness against the compressed data samples would be one of the futuristic works.

3.3 Tex-Net: Texture-based parallel branch cross-attention generalized robust deepfake detector

3.3.1 Abstract

In recent years, artificial faces generated using Generative Adversarial Networks (GANs) and Variational Auto-encoders(VAEs) have become more lifelike and difficult for humans to distinguish. Deepfake refers to highly realistic and impressive media generated using deep learning technology. Convolutional Neural Networks (CNNs) have demonstrated significant potential in computer vision applications, particularly identifying fraudulent faces. However, if these networks are trained on insufficient data, they cannot effectively apply their knowledge to unfamiliar datasets, as they are susceptible to inherent biases in their learning process, such as translation, equivariance, and localization. The attention mechanism of vision transformers has effectively resolved these limits, leading to their growing popularity in recent years. This work introduces a novel module for extracting global texture information and a model that combines data from CNN (ResNet-18) and cross-attention vision transformers. The model takes in input and generates the global texture by utilizing Gram matrices and local binary patterns at each down sampling step of the ResNet-18 architecture. The ResNet-18 main branch and global texture module operate simultaneously before inputting into the visual transformer's dual branch's cross-attention mechanism. Initially, the empirical investigation demonstrates that counterfeit images typically display more uniform textures that are inconsistent across long distances. The model's performance on the cross-forgery dataset is demonstrated by experiments conducted on various types of GAN images and Faceforensics++ categories. The results show that the model outperforms the scores of many state-of-the-art techniques, achieving an accuracy score of up to 85%. Furthermore, multiple tests are performed on

different data samples (FF++, DFDCPreview, Celeb-Df) that undergo post-processing techniques, including compression, noise addition, and blurring. These studies validate that the model acquires the shared distinguishing characteristics (global texture) that persist across different types of fake picture distributions, and the outcomes of these trials demonstrate that the model is resilient and can be used in many scenarios.

3.3.2 Model framework

This section will describe the proposed model and various components of the architecture. The proposed model comprises two major components: the Global texture block architecture and the dual branch cross-attention-based vision transformer (Figure 3.9).

3.3.2.1 Global Texture Module

This model component utilizes ResNet-18 as its foundational architecture. The texture block is calculated before each down-sampling process, integrating the texture information from the input samples into the architecture. Two branches run simultaneously, with one network component utilizing ResNet-18 layers to compute the traditional feature representation, while the other branches focus on computing the texture information. The texture block consists of convolutional layers to align the dimensions of the layers. Gram-matrices and LBP layers extract the texture correlation and the binary features, followed by the convolution to refine the representation and pooling layers to align the computed features with the ResNet backbone features to be forwarded to the next level. The global Texture module is computed at various semantic levels (before every down-sampling operation of the ResNet) for long-range modelling of the texture features. Texture feature extraction is done at two layers, i.e. using Gram matrices and LBP.

Gram matrices as a descriptor of textural characteristics: The texture is represented by the correlation among the features map of the layers; such texturized information remains independent of the spatial information and is represented by the correlation [116]. Gram matrices $G^l \epsilon\ R^{N_l \times N_l}$ represent the correlation or linear dependence between the layers and are computed to the constant of proportionality, given by the formula:

$$\mathcal{G}_{ij}^{l} = \sum_{k} \mathcal{F}_{ik}^{l} \mathcal{F}_{jk}^{l} \tag{3.13}$$

Above equation represents the gram matrix. G_{ij}^{l} which is the inner product between the ith and jth feature of layer l, where F^l represents the l^{th} feature map vectorized representation and F_{ik}^{l} represents the k^{th} activation of the i^{th} filter at position k in layer l.

Local Binary Patterns as a Texture Feature Descriptor: Local binary patterns typically serve as feature descriptors in computer vision for face recognition. Varying the methods to choose neighboring pixels leads to distinct texture patterns in Local Binary Patterns (LBP). Typically, it requires two parameters: the quantity of dots and the radius of the receptive field. This is a textural feature descriptor that compares the value of the center pixel with the values of its neighboring pixels, creating a binary feature. The central pixel is compared to its neighbouring pixel values. If the central value is less than the neighbouring values, it is assigned a value of "0". If the central value exceeds the neighboring values, it is assigned a value of "1". A binary integer is generated and assigned to the center pixel value, forming a grid of binary vectors. The mathematical representation of LBP is as follows:

$$LBP = \sum_{j=0}^{N-1} p(n_i - G_c)\, 2^j \quad , p(x) = \begin{cases} 1, & if\ x > 0 \\ 0, & otherwise \end{cases} \tag{3.14}$$

Where c is the centre pixel, n_i denotes the i^{th} surrounding pixel, and N is the total number of n eighbourhood pixels. Figure 3.8 demonstrates how an image is transformed into an LBP matrix. neighbourhood pixels.

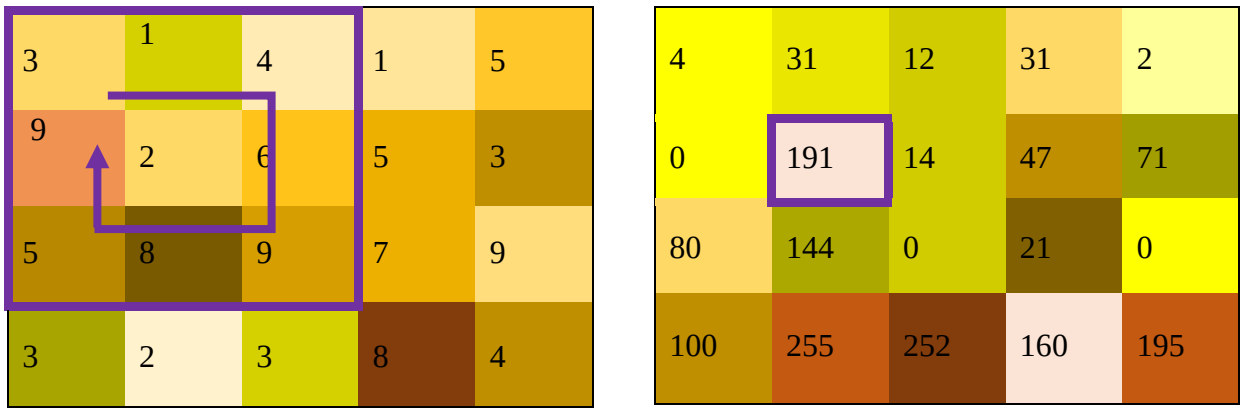


Figure 3.8: Example of how an image is converted to an LBP matrix.

The architecture of this model involves calculating the LBP matrix for each layer of the Gram Matrix. Convolutional and pooling layers then refine the output of the LBP matrix to enhance the representation before being transferred to the next level. Using LBP layers provides a more abstract depiction of the overall texture features representation.

3.3.2.2 Cross-Attention Vision Transformer with parallel branches

The parallel-branches vision consists of two branches that receive patches of different sizes as input. The first branch turns feature maps into a sequence of token patches. In addition, a classification (CLS) token is included in the group of tokens for categorization. As the self-attention mechanism does not consider positional information, the vision transformer

incorporates positional embedding into the token patches, which includes CLS tokens. Next, the token embedding is processed by the transformer encoder, which comprises a series of blocks. Each block contains Multi-headed self-attention (MSA) and a feed-forward network. The feed-forward network comprises two layers of multi-layer perceptron with hidden layers. The GELU non-linearity is applied after the first layer, and Layer Norm (LN) is applied before every block. There is a residual block skip link between the blocks. The input to the vision transformer and processing of the k[th] transformer encoder can be written as:

$$x_0 = [\mathfrak{X}_{clsemb} || \mathfrak{X}_{patchemb}\, \mathcal{E}] + \mathfrak{E}_{pos} \qquad \mathcal{E} \in \mathcal{R}^{\mathcal{P}^2.\mathcal{C} \times \mathcal{D}}, \mathfrak{E}_{pos} \in \mathcal{R}^{(\mathcal{N}+1) \times \mathcal{D}} \tag{3.15}$$

$$z_k = z_{k-1} + \mathcal{MSA}\big(\mathcal{LN}(x_{k-1})\big) \qquad k = 1 \ldots \ldots \ldots \mathcal{L} \tag{3.16}$$

$$x_k = z_k + FFN\left(LN\big(z_{k-1} + \mathcal{MSA}\big(\mathcal{LN}(x_{k-1})\big)\big)\right) \quad k = 1 \ldots \ldots \ldots \mathcal{L} \tag{3.17}$$

where is $\mathfrak{X}_{clsemb} \in \mathcal{R}^{1 \times \mathcal{C}}$, $\mathfrak{X}_{patchemb} \in \mathcal{R}^{\mathcal{N} \times \mathcal{C}}$ and $\mathfrak{X}_{posemb} \in \mathcal{R}^{(\mathcal{N}+1) \times \mathcal{C}}$ are the CLS, patch and positional embedding tokens, respectively($\mathcal{C}$ and $\mathcal{N}$ are the embedding's dimension and the number of the tokens). In order to enhance efficiency and effectiveness in fusing multi-scale characteristics, a CLS token at each branch is used as an agent to share information among the patch tokens from the other branch before projecting the information back to the branch. Finally, the CLS token, which has learned the abstract information of tokens, interacts or serves as a query to the patch to another branch, enabling the fusion of multi-scale information. The CLS token interacts with its patch tokens upon merging with other branch tokens at the subsequent transformer encoder. Here, it can transfer the acquired knowledge from the other branch to its patch tokens, enhancing the patch token representations(Figure 3.10). Similarly, the CLS of another branch interacts with patch tokens of the first branch, enabling the fuse of multi-scale information into the model.

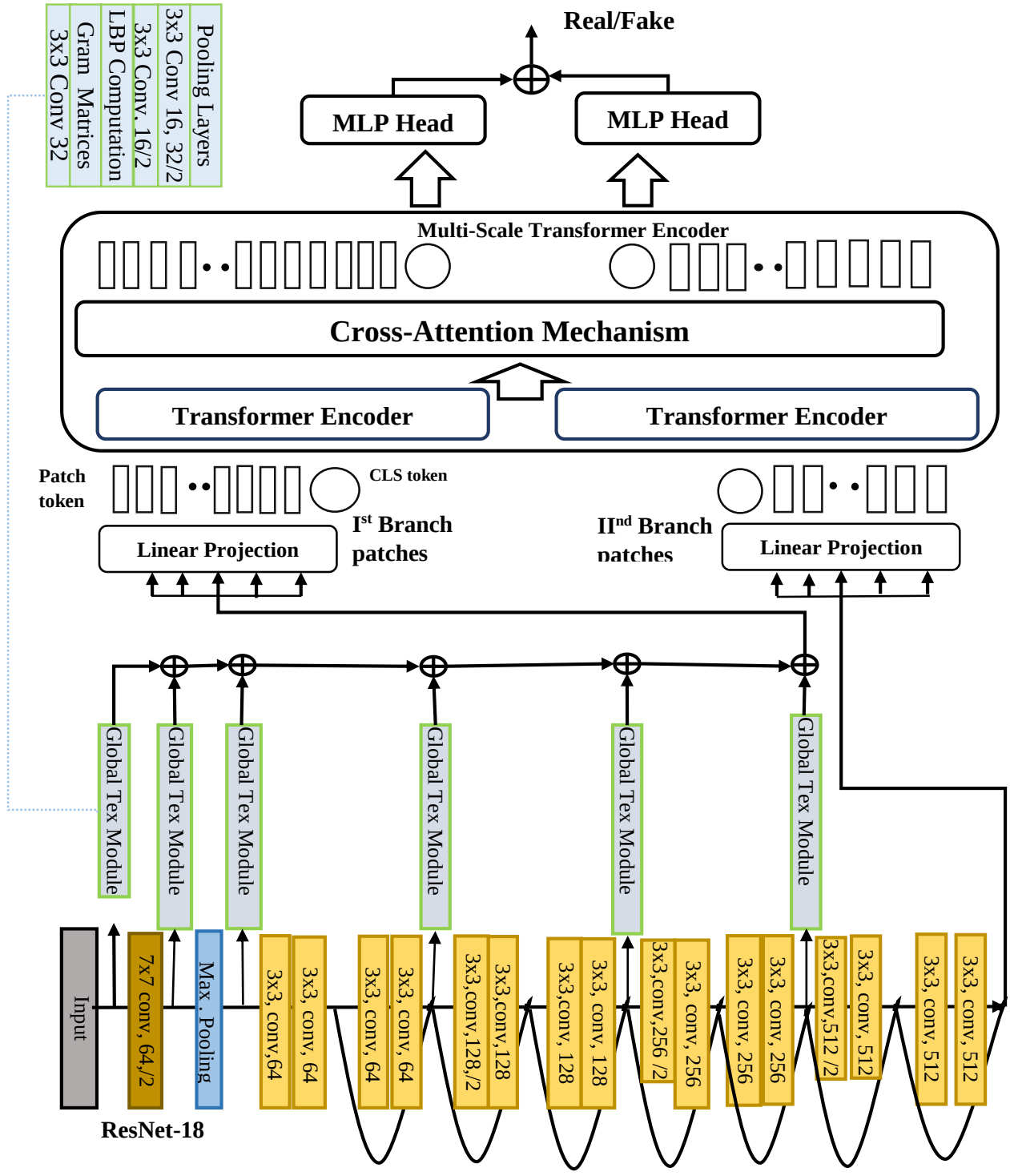


Figure 3.9: The proposed model consists of a global texture module and ResNet serving as inputs to a dual-branch vision transformer with a cross-attention mechanism.

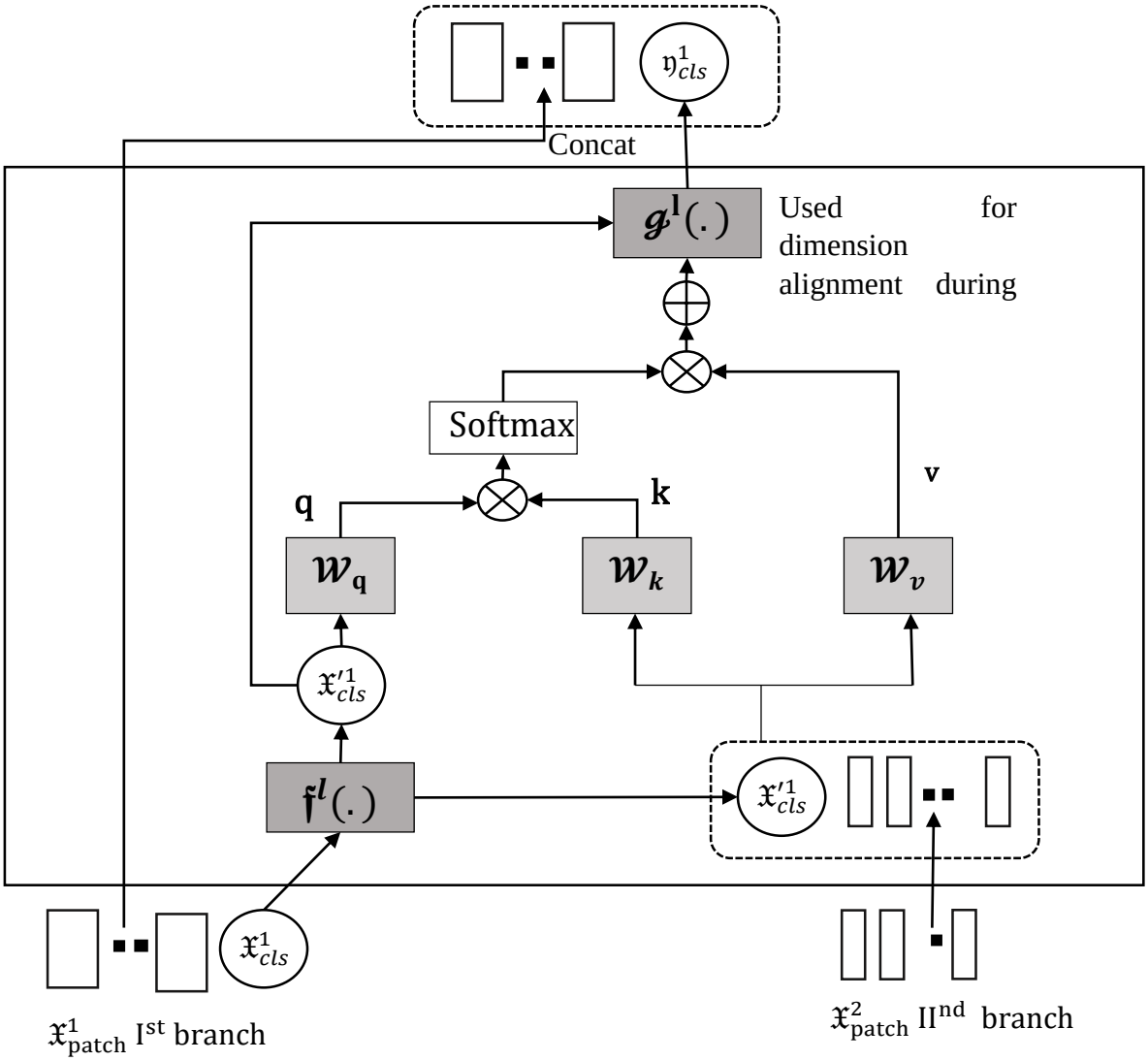


Figure 3.10: Cross attention mechanism where the CLS toke of the I[st] branch acts as a query token and communicates with the patch token of the II[nd] branch

Mathematical equations at the transformer encoder can be written as:

$$\mathfrak{X}^1 = [[\mathfrak{T}^1_{cls} || \mathfrak{T}^2_{ptch}]] \quad \mathfrak{T}^1 \in \text{tokens I}^{\text{st}} \text{ branch}, \quad \mathfrak{T}^2 \in \text{tokens II}^{\text{nd}} \text{ branch} \tag{3.18}$$

$$\mathfrak{X}'^1 = [\mathfrak{f}^1(\mathfrak{X}^1_{cls}) || \mathfrak{X}^2_{patch}] \tag{3.19}$$

$$\mathfrak{Q} = \mathfrak{X}'^1_{\text{cls}} \mathcal{W}_{\text{q}}, \quad \mathfrak{K} = \mathfrak{X}'^1 \mathcal{W}_k \quad \mathcal{V} = \mathfrak{X}'^1 \mathcal{W}_v \quad \mathcal{W}_q, \mathcal{W}_k, \mathcal{W}_v \in \mathfrak{R}^{\mathcal{C} \text{ x } (\mathcal{C}/\mathcal{h})} \tag{3.20}$$

$$\mathfrak{A} = \mathbb{S}\text{oftmax}\left(\frac{\mathfrak{Q}\mathcal{k}^T}{\sqrt{\mathfrak{D}_{\mathfrak{h}}}}\right) \qquad \mathfrak{A} \in \mathfrak{R}^{\mathbb{N} \times \mathbb{N}} \tag{3.21}$$

$$\mathcal{CA}(\mathfrak{X}^1) = \mathfrak{A}\mathfrak{v} \tag{3.22}$$

$$\mathfrak{M}\mathcal{CA}(\mathfrak{X}^1) = [\mathcal{CA}_1(\mathfrak{X}^1); \mathcal{CA}_2(\mathfrak{X}^1); \ldots\ldots; \mathcal{CA}_{\mathcal{k}}(\mathfrak{X}^1)]\mathcal{W}_{\text{msa}} \qquad \mathcal{W}_{\text{msa}} \in \mathfrak{R}^{\mathcal{k}.\mathfrak{D}_{\mathfrak{h}} \times \mathfrak{D}} \tag{3.23}$$

$$\mathfrak{y}^1_{cls} = \mathfrak{T}^1_{cls} + \mathfrak{M}\mathcal{CA}\langle \mathcal{LN}\{[[\mathfrak{T}^1_{cls} || \mathfrak{T}^2_{ptch}]]\}\rangle \tag{3.24}$$

Where the number of patches is $\mathfrak{n}+1$, the model dimension is $\mathcal{D}$, the number of heads is $\mathcal{k}$, and the head dimension is $\mathfrak{D}_h(d/\mathcal{k})$ and $\mathfrak{f}^l(.)$ is a dimension alignment operator. The query, key, and value are represented by the variables $\mathcal{q}$, $\mathcal{k}$, and $\mathcal{v}$, respectively. As for the query, key, value, and MSA, respectively, the learnable parameters are $\mathcal{W}_{\mathcal{q}}, \mathcal{W}_{\mathcal{kv}}$, and $\mathcal{W}_{\mathfrak{msa}}$. At the subsequent transformer encoder, the CLS token interacts with its patch tokens again after merging with other branch tokens. This improves the representation of each patch token by transferring knowledge from the other branch to its own. Afterwards, these inputs are sent to an MLP (Multi-Layer Perceptron) for parameter learning after passing via the Layer Norm:

$$\mathbb{Z} = \mathcal{MLP}\{\mathcal{LN}\langle \mathfrak{y}_{\mathfrak{cls}}^1 + \mathfrak{X}_{\mathfrak{patch}}^1 \rangle\} \tag{3.25}$$

These embedding's from the two branches are finally concatenated for the final prediction.

Algorithms 1: LBP-ViT for Deepfake Detection

Setting the initial parameter:

- Input: $\mathbb{I} = \{\mathrm{I}_1, \mathrm{I}_2, \mathrm{I}_3, \ldots\ldots, \mathrm{I}_n\}$ represents the set of data image samples,
- $\mathbb{L} = \{0, 1\}$ represent the set of labels, with 0 denoting the real image and 1 the deepfake one.
- n is the dataset's size.
- Divide $\mathbb{I}$ into three subsets: 70% for training, 15% for validation, and 15% for testing.

1: **Perform for 1 to 100 epochs:**

2: Input set of images $\mathcal{J} \subseteq \mathbb{I}$ to ResNet module for feature extraction.

3: Compute the texture features using the Global texture block before each ResNet down sampling operation and continue concatenating them.

4: ResNet computed and texture features at steps 1 and 2 are fed into the parallel branches of the vision transformer.

5: At each branch, flatten the features into patches of fixed sizes.

6: Create linear embedding's in smaller dimensions with flattened image patches $\mathfrak{T}_{pch}^{1||2}$ using linear projection module.

7: Add positional embedding together with the CLS token.

8: Input the sequence into each branch's transformer encoder.

9: Query the CLS token of the Ist branch with patch tokens of another branch to

10: create tokens and vice versa for the cross-attention mechanism.

Sophisticated features are further passed to Multi-Layer Perceptron(MLP) for latent feature learning.

11: Concatenate the features of both branches for classification.

12: Update the weights using the Adam optimizer and train the model end-to-end.

13: Evaluate the validation set and save the weights of the well-performing model.

14: **end for**

15: Load the model's weights saved at step 13.

16: Evaluate the model on the test dataset.

3.3.3 Experimentation

This section of the study will analyze the selection of training parameters, various datasets, the choice of face extractor, and the diverse tests undertaken for the model.

3.3.3.1 Experimental Settings

The studies employ NVIDIA TITAN RTX GPUs equipped with 24GB of RAM. The initial learning rate is established as 0.01, while the batch size is defined as 64. The Adam optimizer is used to update the parameters of the model. Each experiment consists of running one hundred epochs, as it has been determined that the system's performance reaches a saturation point after this number of epochs.

3.3.3.2 Dataset and its pre-processing

The Faceforensics++ dataset was utilized to assess the performance of the model. The Faceforensics++ dataset is divided into four categories: Deepfakes, Face2face, Face swap, and Neural Textures. This dataset comprises brief facial videos from which various frames have been extracted using the RetinaFaceResNet50 face extractor, chosen for its lower failure rates than MTCNN. FF++ frames have a size of [151, 200] and an aspect ratio of [1, 1.5]. In addition, various GAN images are employed to assess the model. Artificial images generated by ProGAN, StyleGAN, STGAN, and StarGAN, as well as authentic image datasets such as CelebA-HQ, CelebA, and FFHQ, are obtained from their respective online sources(Table 3.9).

Table 3.9: Details for the training, validation and testing dataset with their resolutions

Dataset	**Training Set**	**Validation set**	**Testing set**	**Image Resolution**
FF++(DeepFakes)	8k real & 8k fake image frames.	2k real & 2k fake image frames.	2k real & 2k fake image frames.	128x128

FF++(face2face)	8k real & 8k fake image frames.	2k real & 2k fake image frames.	2k real & 2k fake image frames.	128x128
FF++(Faceswap)	8k real & 8k fake image frames.	2k real & 2k fake image frames.	2k real & 2k fake image frames.	128x128
FF++(Neural Texture)	8k real & 8k fake image frames.	2k real & 2k fake image frames.	2k real & 2k fake image frames.	128x128
CelebA-HQ & ProGAN	10k(CelebA-HQ) & 10k(ProGAN)	1.5k(CelebA-HQ) & 1.5k(ProGAN)	1.5k(CelebA-HQ) & 1.5k(ProGAN)	1024x1024
CelebA-HQ& StyleGAN	10k(CelebA-HQ) & 10k(StyleGAN)	1.5k(CelebA-HQ) & 1.5k(StyleGAN)	1.5k(CelebA-HQ) & 1.5k(StyleGAN)	1024x1024
FFHQ and StyleGAN	10k(FFHQ) & 10k(StyleGAN)	1.5k(FFHQ) & 1.5k(StyleGAN)	1.5k(FFHQ) & 1.5k(StyleGAN)	1024x1024
CelebA & StarGAN	10k(CelebA) & 10k(StarGAN)	1.5k(CelebA) & 1.5k(StarGAN)	1.5k(CelebA) & 1.5k(StarGAN)	128x128
CelebA & STGAN	10k(CelebA) & 10k(StarGAN)	1.5k(CelebA) & 1.5k(StarGAN)	1.5k(CelebA) & 1.5k(StarGAN)	128x128

3.3.3.3 Data Augmentation

Vision transformers are data-hungry; hence, they need data to train them before making good predictions, as shown by the initial ViT model [118]. However, the DeIT model [119], with careful and rich data-augmentation techniques, can perform better and beat the scores of various state-of-the-art models. Different data-augmentation techniques of the DeIT model, like cut mix [121], mixup [122], and rand augmentation [120], along with the drop path regularization model, have been used to improve the results.

3.3.3.4 Experimentation on various categories of FF++ in cross-domain settings

The Faceforensics++ [15] dataset consists of four distinct categories: Deepfakes (DF), Face2face (f2f), Face swap (FS), and Neural Textures (NT). The model is trained on a single category of FF++ and then evaluated on both the same category and other categories of FF++. The model undergoes training and subsequent validation on the validation set. Despite establishing multiple SoTA models in recent studies, comparing them equally remains challenging. This is partly because there are no publicly accessible codes for the models and training techniques the broader research community may use. Hence, the models with accessible codes online have been selected for comparison, totaling seven models.

1) Capsule Network [97].
2) MesoInception-4 model [125].

3) Combining Efficient Net and Vision Transformers for Video Deepfake Detection(E-ViT) [126].
4) Recce [133].
5) IID [128].
6) SBI [134].
7) UCF [108].

The code for these models has been sourced from their respective GitHub repositories and tailored to suit the dataset. Additionally, additional evaluation metrics have been incorporated to provide a more comprehensive assessment of the model. The models are trained on a specific category of manipulation in the FF++ dataset and then evaluated on the remaining categories of the same dataset.

The results of the several models tested on the remaining FF++ categories after being trained on the DeepFakes category are shown in Table 3.10. Except for MesoNet and Recce, every model achieves a flawless score, indicating the presence of overfitting in the models. This implies that they perform inadequately when tested on a different dataset with a distinct distribution but perform well when trained and tested on the same dataset. Most models achieve an accuracy score of around 50% for the deepfake face swap category. The performance of our model surpassed that of the other models by a substantial margin, achieving a score of over 70%. This demonstrates that texture is a feasible feature that can be used in many variations. In addition, current state-of-the-art (SoTA) techniques, specifically Recce [64] and IID [65], do not perform optimally, indicating that there is still a significant gap between the desired outcome and the effectiveness of various manipulations.

Table 3.11 displays the model comparison between the face2face training data and the testing data from different categories of FF++. All models exhibited exceptional performance across all dataset categories, surpassing the scores presented in the table above. MesoNet and CapsuleNet, considered state-of-the-art models in the past, struggled to achieve accuracy beyond 60%. These models utilized conventional convolution layers to learn the features of the training dataset, leading to inferior performance when applied to unseen datasets. Based on the transformer architecture, the E-VIT model has outperformed earlier models. This demonstrates the vision transformer's ability to learn global feature space, resulting in a modest improvement compared to CNN-based approaches. The reconstruction-based method, Recce, has subpar performance and scores even lower than previous methods. In many circumstances, the model's reconstruction abilities do not contribute to its performance. The IID technique, which incorporates both implicit and explicit identities, has partially uncovered the distinction,

leading to somewhat improved performance compared to previous models. Our model has outperformed the other models and achieved an accuracy score of 80%, demonstrating the strong discriminatory capabilities of the texture and cross-attention mechanism of ViT.

Table 3.12 displays the score for the model trained on the Faceswap category and then tested on all other categories of FF++. Most models exhibit subpar performance when trained on FS but excel when tested within the same category. This is partially because of the Faceswap formation procedure that utilizes facial landmark points. These points are used to create a 3D template, which is then projected onto the target shape to minimize any differences between the projected shape and the landmark points. The IID model created explicitly for identifying face swap faces, performs relatively better than other models but still performs poorly. The UCF model, although designed to uncover standard features through feature disentanglement and method-specific approaches, does not perform as well as other models. The SBI model, known for its ability to perform generalization and robustness, has unfortunately fallen short compared to other models. Recce and MesoNet models do not perform well in diverse scenarios. However, the E-VIT model performs slightly better than the other models. Our model has achieved an average score of approximately 65%, surpassing other models' performance. However, there is room for improvement in the category of manipulation. The feature space for this particular category differs from other manipulations, which challenges our model's performance in this area.

Table 3.13 depicts the model trained on NT and subsequently verified on the other categories of FF++. All categories of models exhibit comparable performance to one another. The generation of NT data samples utilizes the neural texture of the targeted individual and employs a rendering-based approach. This texture is also a common feature found in various manipulations. Training on such examples allows the model to perform well on other manipulations. Most models have an accuracy score of approximately 70% or higher. All models in the FS category exhibit subpar performance. The accuracy score of Capsule-Net is significantly lower compared to other models. DF has the highest level of accuracy compared to all other categories of cross-manipulation, with the face2face category ranking second. The model is constructed using the texturing technique, allowing it to comprehend the intrinsic attributes of the data sample. Consequently, it demonstrates exceptional proficiency in managing many forms of manipulation. The performance of our model has been potent, with an accuracy score that has reached 85% in cross-manipulation scenarios.

Most models achieve near-perfect scores when trained and tested on the same distribution but struggle to generalize effectively to different distributions. Thanks to the texture module and

cross-attention mechanism, our model demonstrates strong performance over a wide range of manipulations. This confirms that texture is a consistent and valuable feature across diverse manipulations. Nevertheless, every performance is adversely affected by the FS category of manipulation of FF++. Figure 3.11 depicts the Receiver Operating Characteristic (ROC) scores of different models trained on the Deepfake category and then evaluated on the other categories of FF++. The superiority of the LBP-ViT model over other models is readily apparent. Additionally, the performance of all models decreases in the Face swap category. his decline may be attributed to face-swap manipulation inconsistencies, which have a limited range and may not be effectively captured by the model's texture analysis.

3.3.3.5 Experiments on the different GAN images in cross-domain settings

The GAN images have been known to be highly realistic and believable to the naked eye since the breakthrough research paper by Ian Goodfellow [135] in 2014. Different GAN image generation mechanisms, like ProGAN, StyleGAN, StarGAN, and STGAN, have been developed over the years to cover various types of manipulation of images. High-resolution images like ProGAN and StyleGAN and low-resolution images like StarGAN and STGAN were used to evaluate the model performance in different image settings. Five real and fake image datasets have been designed for fair and comprehensive evaluation: CelebA-HQ ProGAN, CelebA-HQ StyleGAN, FFHQ StyleGAN, CelebA StarGAN, and CelebA STGAN. Models have trained on one set and tested on the other set. Seven state-of-the-art models have been considered for comparison against the proposed model:

1) Residual-Net [136].
2) CNN's generated images are surprisingly easy to spot now(CNN-Net) [130].
3) Efficient-Net [137].
4) Recce [133].
5) IID [128].
6) SBI [134].
7) UCF [108].

Similarly, for these models, code has been taken from their GitHub repositories and customized for the GAN dataset, and more evaluation metrics have been added for a more robust and comprehensive evaluation. Table 3.14 represents the models' results on various GAN images. Other models seem to be under-fitting to the cross-forgery detection and overfitting to the same data distribution settings. Another interesting observation is that almost all the models do not perform well when trained on CelebA-HQ StyleGAN and tested on the CelebA-HQ ProGAN

images. It is evident that when the testing dataset contains both fake and real images, models can swiftly identify them due to their training but struggle to identify the other class. In the first row, the model trained on CelebA-HQ ProGAN and tested on CelebA-HQ StyleGAN shows high precision in identifying the CelebA-HQ class. However, the lower recall value indicates it struggles with classifying the other category. ProGAN and StyleGAN utilise distinct manipulation techniques, leading to the creation of distinct feature spaces. Consequently, models trained on one category may not yield satisfactory performance on the other. The CelebA-HQ and FFHQ real datasets exhibit similar feature space, allowing models trained on one to perform effectively on the other. For instance, a model trained on the CelebAHQ StyleGAN can achieve impressive results when combined with the FFHQ StyleGAN. In other words, state-of-the-art(SoTA) methods excel in accurately classifying authentic images but struggle when identifying fake images. This leads to an average accuracy score of 50%, indicating a weak performance in detecting samples from different datasets. The ideal score for CelebA StarGAN and CelebA STGAN images indicates their common data distribution space. Our model effortlessly surpassed the scores of several cutting-edge models and significantly elevated the standards of evaluation criteria. Therefore, once again, this confirms that the model has the capability to acquire diverse distinguishing characteristics and textures that appear to endure across multiple types of counterfeit picture distributions. Nevertheless, the model encounters challenges when trained on StyleGAN images and tested on ProGAN images. Despite being synthetic, it is essential to note that these images come from different distribution spaces, which poses a challenge for the model to identify. Figure 3.12 represents the ROC curves of the models trained and tested on various GANs. The AUC scores of different models are shown and apparently our model has the highest score in comparison to the other models.

3.3.3.6 Experiments on various post-processing operations of FF++, Celeb-DF, and DFDCPreview dataset

Images or data samples on the web undergo processing operations like blurring, compression, translation, rotation, up sampling or down sampling, and manipulators. A constraint of the diverse models is their insufficient resilience to diverse post-processing techniques such as noise addition, scaling, translation, blurring, compression, and so on [132]. Three post-processing procedures (blurring, compression, and noise addition) are performed on the test dataset to show the model's resilience. The images were blurred using the Gaussian blur PyTorch transformation with a kernel size of 7x7 and sigma 25; noise was added using a zero mean and 0.2 standard deviation; and the images' quality was reduced by three times due to

compression (Figure 3.13). The regular images were used to train the models, but several post-processing methods were applied to the images for testing. Three primary deepfake datasets have been taken into consideration for analysis. Once more, the comparative evaluation was conducted using four models:

a) CViT [138].
b) MesoInception-4 model [125].
c) CNN's generated images are easy to spot now(CNN-Net) [130].
d) Recce [133].
e) IID [128].
f) SBI [134].
g) UCF [108].

Testing dataset without undergoing any post-processing operations: The initial row of each dataset exhibits the outcomes without any post-processing interventions applied to the image. All models in this scenario have exhibited outstanding performance, as shown in (Table 3.15),. The accuracy for the FF++ and DFDCPreview datasets is approximately 99%, highlighting the impressive capabilities of these state-of-the-art models in performing well within their designated domains. The IID achieved a flawless score on the DFDCPreview dataset. The accuracy score for Celeb-Df is approximately 95% for these models, as the dataset contains realistic diversified content that includes subtle modification artifacts that are challenging for the model to identify. MesoNet exhibits a marginal decline in performance for the FF++ and DFDCPreview datasets, potentially attributed to its reliance on conventional CNN layers, which may not correctly learn all the manipulations present in the FF++ dataset. The Recce approach exhibits a marginal decline in performance when used to the DFDCPreview dataset, mainly because it is not adept at accurately detecting the minor anomalies present in this dataset.

Testing dataset undergoes blurring operations: Models are trained on data samples without blurring operation while tested on the blurred data samples. As stated, The images were blurred using the Gaussian blur PyTorch transformation with a kernel size of 7x7 and sigma 25. The models had a minimum drop of 14% in performance when subjected to blur operations in the case of FF++. In the case of the DFDCPreview and Celeb-Df datasets, the performance decreased marginally, approximately 4-6%, with a few exceptions for specific models. MesoNet has a more significant decline in performance for Celeb-DF and DFDCPreview, mainly because these datasets heavily depend on conventional CNN features. The IID model has demonstrated a loss in performance for the FF++ dataset, while there is a modest decrease

in performance for the other dataset. Another Recce model is significantly affected by the blurring operation due to its firm reliance on the training dataset. As a result, it exhibits inadequate generalization and robustness when faced with unseen data during testing. Other models are likewise affected by the blurring procedure, although to a lesser degree. Our model demonstrates enhanced resilience and robustness, as evidenced by a minimal reduction in performance of only 1-2% for the DFDCPreview and Celeb-Df datasets. Additionally, approximately 14% of the data samples still exhibit discriminative artifacts even after being subjected to significant blurring.

Noise addition to the test data samples: The data samples are augmented with noise using a PyTorch transformation, where the noise has a mean of zero and a standard deviation of 0.2. The results revealed a substantial decrease in the scores for all the models, with scores as low as 50% for CviT, MesoNet, and CNN-Net. This highlights the susceptibility of these detection approaches to the introduction of noise. The hybrid model of CViT has been significantly impacted by the introduction of noise in the FF++ and DFDC Preview datasets, resulting in susceptibility to adversarial perturbations and subsequent inaccurate classification. MesoNet experiences a significant decrease in its classification score, rendering it vulnerable to different adversarial techniques. The UCF model, renowned for identifying shared characteristics through feature disentanglement and multi-task learning, faces significant challenges in achieving robustness when exposed to noise. Another approach, i.e. Recce, utilizing reconstruction learning and a bi-partite graph, focuses on generalization but struggles to perform effectively when noise features are introduced. In order to achieve high performance on a dataset with much noise, a model must either employ advanced data augmentation techniques that introduce noise to help the model learn how to classify it or utilize the attention mechanism to leverage multi-scale features that can effectively capture both local and global features that are robust to noise. Our model has used the latter strategy, enabling it to acquire intricate and resilient characteristics that withstand different adversarial procedures. When the accuracy of the other model in the DFDCPreview dataset does not exceed 80%, the LBP-ViT model consistently exceeds it with an accuracy score of 98%. Nevertheless, the model is still susceptible to the introduction of noise to some degree.

Testing dataset undergoes compression: The images' quality was reduced by three times due to compression. The models' performance does not significantly decrease when samples are compressed for the DFDCPreview and Celeb-DF datasets, showing that the size reduction has minimal effect on the modified artifacts. The performance of models such as UCF, CNN-Net, Recce, and IID in the FF++ dataset has been significantly affected, indicating that these models

are not explicitly designed for compression scenarios. Additionally, the dataset contains multiple categories of manipulation, making it even more challenging for the models to learn the diverse distribution of features. The Celeb-DF dataset had the most negligible impact on the performance of the models, followed by the DFDCPreview dataset. Compression typically includes decreasing the resolution of data samples, affecting the finer features and leading to blocking, halo, ringing, and banding distortions. These artifacts specifically damage the gradients in the smooth sections. The models that concentrate on certain artifacts struggle to perform well in the presence of diverse and complex characteristics. Models that concentrate on several types of artifacts at different levels of detail are more likely to categories complicated feature patterns accurately. Our model prioritizes intricate characteristics, starting with textures and incorporating traditional CNN features at various scales. It leverages transformer capabilities to grasp global and local details through a cross-attention mechanism. This enables the model to effectively learn subtle, complex features at multiple scales for improved performance.

Table 3.10 Models trained on the Deepfakes category of FF++ and tested on its other categories

Test	Ucf					MesoNet					CapsuleNet					E-ViT					Recce					IID					Sbi					LBP-ViT(ours)				
	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc
Df	0.9989	0.9985	0.9987	0.9987	0.9987	**1.0**	0.999	0.9994	**1.0**	0.9995	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.8195	0.9007	0.9949	0.9097	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
F2F	**0.791**	0.263	0.263	0.697	0.6023	0.667	0.1455	0.2389	0.6345	0.5365	0.6239	0.151	0.2431	0.6159	0.53	0.7551	0.165	0.2708	0.6479	0.5557	0.629	0.084	0.148	0.6417	0.5172	0.7790	0.2715	0.4026	0.7218	0.5972	0.6641	0.1355	0.2250	0.6167	0.5334	0.7500	**0.6411**	**0.6913**	**0.7978**	**0.7198**
FS	0.224	0.018	0.033	0.711	0.5396	0.2168	0.0155	0.0289	0.2962	0.4797	0.3764	0.0335	0.0615	0.4207	0.489	0.3925	0.0265	0.0496	0.4700	0.4927	0.2147	0.0175	0.0323	0.4177	0.4767	0.4637	0.016	0.0309	0.5621	0.4987	0.2318	0.016	0.0299	0.3806	0.4815	**0.7151**	**0.045**	**0.0847**	**0.6673**	**0.6478**
NT	0.8164	0.238	0.368	0.6693	0.5923	0.7703	0.208	0.3275	0.6972	0.573	0.7132	0.25	0.3702	0.6639	0.5747	0.7895	0.257	0.3877	0.6829	0.5942	0.8696	0.13	0.2262	0.7158	0.5555	**0.8765**	0.245	0.3829	0.7263	0.6025	0.7890	0.2525	0.3825	0.7318	0.5925	0.7382	**0.6555**	**0.6944**	**0.7982**	**0.7201**

Table 3.11: Models trained on the Face2face category of FF++ and tested on its other categories

Test	Ucf					MesoNet					CapsuleNet					E-ViT					Recce					IID					Sbi					LBP-ViT(ours)				
	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc
Df	0.8531	0.3165	0.4617	0.7533	0.631	0.7459	0.367	0.4919	0.7085	0.621	0.8161	0.253	0.3862	0.6304	0.598	0.6751	0.2775	0.3933	0.6733	0.572	**0.8918**	0.1815	0.3016	0.8082	0.5795	0.8638	0.387	0.5345	0.8117	0.663	0.7985	0.5015	0.6160	0.794	0.6875	0.8876	**0.6435**	**0.7461**	**0.8619**	**0.781**
F2F	0.9975	0.9985	0.9980	0.9986	0.998	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.9989	0.9875	0.9932	0.9995	0.9932	0.9995	**1.0**	0.9997	1.0	0.9997	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
FS	**0.644**	0.124	0.2079	0.5282	0.5277	0.5871	0.266	0.3662	0.6064	0.5395	0.6371	0.151	0.2441	0.566	0.532	0.6399	0.2390	0.3480	0.5922	0.5525	0.6	0.0945	0.163	0.6507	0.5157	0.5780	0.1315	0.2143	0.6605	0.5177	0.5997	0.2915	0.3923	0.6106	0.5485	0.6340	**0.5839**	**0.6079**	**0.6657**	**0.6512**
NT	0.8092	0.314	0.4524	0.6827	0.62	0.6937	0.3375	0.4541	0.6843	0.5942	0.8122	0.2855	0.4224	0.6666	0.609	0.7209	0.4855	0.5802	0.7106	0.6487	0.8472	0.2385	0.3722	0.7480	0.5977	0.7891	0.378	0.5111	0.7877	0.6384	0.7618	0.579	0.6579	0.7871	0.6989	**0.8515**	**0.777**	**0.8125**	**0.8804**	**0.8207**

Table 3.12: Models trained on the Faceswap category of FF++ and tested on its other categories

Test	Ucf					MesoNet					CapsuleNet					E-ViT					Recce					IID					Sbi					LBP-ViT(ours)				
	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc	Pr	Re	F1	AUC	Acc
Df	0.371	0.0305	0.0564	0.457	0.4894	0.5892	**0.522**	**0.5535**	0.6321	0.579	0.5375	0.0895	0.1534	0.5415	0.5065	0.6872	0.311	0.4282	0.6712	0.5847	0.5466	0.1495	0.2347	0.5438	0.5127	0.7414	0.3885	0.5098	**0.7262**	0.6265	0.5380	0.092	0.157	0.5759	0.5065	**0.8075**	0.2677	0.4021	0.6984	**0.6314**
F2F	0.7403	0.1625	0.2665	0.5369	0.5527	**0.6216**	**0.5455**	**0.5811**	0.6354	**0.6067**	0.6383	0.1985	0.3028	0.5568	0.5430	0.6700	0.3005	0.4149	0.573	0.5762	0.6783	0.3005	0.4164	0.6157	0.579	0.6943	0.4315	0.5322	**0.6964**	0.6207	0.6758	0.1845	0.2898	0.5824	0.5480	0.6902	0.3075	0.4254	0.6316	0.5847
FS	0.9969	0.9949	0.9949	0.9949	0.995	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.9856	0.996	0.9907	0.9990	0.9907	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	1.0	0.9997	0.9997	0.9997	0.9997
NT	0.6137	0.058	0.105	0.4929	0.5107	0.5903	**0.539**	**0.5635**	0.6205	0.5825	0.5342	0.1365	0.2174	0.5350	0.5087	0.6246	0.2355	0.3420	0.5574	0.547	0.5686	0.1635	0.2539	0.5367	0.5197	0.6113	0.2855	0.3892	0.5926	0.552	0.6513	0.085	0.1503	0.5267	0.5197	**0.8514**	0.0999	0.1788	**0.6434**	**0.6596**

Table 3.13: Models trained on the NT category of FF++ and tested on its other categories

Test	Ucf					MesoNet					CapsuleNet					E-ViT					Recce					IID					Sbi					LBP-ViT(ours)				
	Pr.	Re.	F1	AUC	Acc.	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc
Df	**0.8814**	0.613	0.7231	0.7860	0.7652	0.7347	0.8145	0.7725	0.8287	0.7602	0.7619	0.616	0.6812	0.7801	0.7117	0.7894	0.6505	0.7133	0.8325	0.7385	0.7941	0.729	0.7601	0.8542	0.77	0.8006	0.532	0.6392	0.8331	0.6997	0.7913	0.711	0.7490	0.8538	0.7617	0.8285	**0.858**	**0.843**	**0.9166**	**0.8402**
F2F	0.7989	0.379	0.5146	0.6384	0.642	0.7276	**0.8615**	0.7889	0.8342	0.7695	0.7100	0.5045	0.5898	0.7265	0.6492	0.7583	0.739	0.7485	0.8374	0.7517	0.7169	0.6245	0.6675	0.7843	0.6899	0.7794	0.5335	0.6334	0.8213	0.6912	0.7703	0.689	0.7273	0.8212	0.7417	**0.8593**	0.849	**0.8541**	**0.9360**	**0.855**
FS	0.2403	0.0465	0.078	0.4005	0.4497	0.5013	**0.3745**	**0.4287**	0.5296	0.501	0.4557	0.1675	0.2449	0.4886	0.4837	0.4966	0.2235	0.3082	0.5057	0.4985	0.3039	0.1295	0.1816	0.4045	0.4156	0.3761	0.1025	0.1611	0.4985	0.4662	0.4234	0.177	0.2496	0.4720	0.4680	**0.7482**	0.0299	0.0572	**0.6763**	**0.6676**
NT	0.9954	0.993	0.9945	0.9972	0.9945	0.9985	**1.0**	0.9992	**1.0**	0.9992	**1.0**	0.9925	0.9959	0.9978	0.996	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.9494	0.9955	0.9719	0.9999	0.9712	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.9985	**1.0**	0.9992	**1.0**	0.9992

Table 3.14: Models trained and tested on different types of GAN datasets

Train	Test	Ucf					Res-Net					CNN-Net					Efficient-Net					Recce					IID					Sbi					LBP-ViT(ours)				
		Pr.	Re.	F1	AUC	Acc.	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc
CelebA-HQ_ProGAN	**CelebAHQ_StyleGAN**	0.9	0.006	0.0119	.4645	0.50266	**1.0**	0.1166	0.2089	0.9598	0.5583	0.9714	0.1913	0.3212	0.9708	0.5957	**1.0**	0.008	0.0158	0.8466	0.504	**1.0**	0.0013	0.0026	0.6608	0.5066	**1.0**	0.0033	0.0066	0.6842	0.50166	0.9655	0.0186	0.0366	0.6814	0.509	0.9881	**0.662**	**0.7982**	**0.9721**	**0.827**
	FFHQ_StyleGAN	0.652	**0.997**	**0.7884**	0.7777	0.7323	0.6590	0.1726	0.2736	0.6253	0.5416	0.6098	0.3466	0.4465	0.6533	0.5703	0.7215	0.038	0.0722	0.7205	0.5116	0.6666	0.0013	0.0026	0.6572	0.5003	0.8	0.0026	0.0053	0.6048	0.501	**0.9**	0.018	0.035	0.623	0.508	0.7724	0.776	0.7741	**0.8317**	**0.7736**
CelebA-HQ_StyleGAN	**CelebAHQ_ProGAN**	0.471	0.005	0.0105	0.4161	0.4966	0.01	0.01	0.01	0.5354	0.4996	0.7889	0.0033	0.0066	**0.8363**	0.5013	**1.0**	0.006	0.0013	0.8093	0.5033	**1.0**	0.0066	0.0013	0.497	0.503	0.5	**0.5**	**0.5**	0.5468	0.5	0.6292	0.084	0.1482	0.6417	0.5172	**0.7635**	0.254	0.3811	0.7299	**0.5876**
	FFHQ_StyleGAN	0.732	0.999	0.8452	0.7735	0.8169	0.8156	**1.0**	0.8985	0.9979	0.887	**0.9931**	0.9986	0.8619	0.9943	0.84	0.7420	0.9993	0.8517	0.9972	0.8260	0.8862	0.966	0.9244	0.9812	0.9210	0.8631	1.0	0.9265	0.9998	0.9206	0.7689	0.9986	0.8689	0.9952	0.8493	**0.9565**	**0.9546**	**0.9556**	**0.9901**	**0.9556**
FFHQ_StyleGAN	**CelebAHQ_ProGAN**	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.01	0.567	0.50	0.7727	0.012	0.023	0.7799	0.506	**0.9032**	0.0186	0.0365	0.7952	0.5083	0.75	0.002	0.0039	0.6090	0.5006	0.6666	0.0013	0.0026	0.6027	0.5033	0.6102	0.024	0.0462	0.6001	0.5043	0.7404	**0.7226**	**0.7314**	**0.8048**	**0.7346**
	CelebAHQ_StyleGAN	0.996	0.99	0.9933	0.9979	0.9933	**1.0**	0.9973	0.9986	0.9999	0.9987	0.9998	0.9963	0.9963	0.9999	0.9963	0.9963	0.9993	0.9993	0.9999	0.9993	**1.0**	0.848	0.9177	0.9965	0.924	**1.0**	0.9993	0.9996	0.9999	0.9996	0.9881	0.998	0.9930	0.9997	0.993	**1.0**	**0.9986**	**0.9993**	**0.9999**	**0.9993**
CelebA_StarGAN	**CelebA_STGAN**	0.999	**1.0**	0.9996	**1.0**	0.9996	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**0.5**	0.006	0.0118	0.9885	0.50	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
CelebA_StarGAN	**CelebA_STGAN**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**0.5**	0.5	0.5	0.5	0.5	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Table 3.15: Models trained on different datasets and tested on various post-processing scenarios.

Training Dataset	Testing Dataset	CViT					MesoNet					UCF					CNN-Net					Recce					IID					Sbi					LBP-ViT(ours)				
		Pr.	Re.	F1	AUC	Acc.	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc	Pr.	Re.	F1	AUC	Acc
FF++	**FF++**	0.9412	0.979	0.9596	0.994	0.9997	0.8322	0.9998	0.9084	0.9968	0.8991	0.988	0.9915	0.9898	0.9968	0.9897	**0.9998**	0.9996	0.9968	0.9999	0.9968	0.9898	0.9142	0.9505	0.9942	0.9524	0.9897	0.9951	0.9923	0.9996	0.9923	0.9973	0.9972	0.9973	0.9991	0.9973	0.9975	**0.9982**	**0.9977**	**0.9999**	**0.9998**
FF++	**FF++ Blurry**	0.7590	0.996	0.8615	0.966	0.8399	0.5305	**1.0**	0.6933	0.9819	0.5575	0.673	0.9282	0.7803	0.8297	0.7386	**0.9807**	0.9982	0.8329	**0.9831**	0.7998	0.7059	0.8767	0.7821	0844	0.7558	0.575	0.9975	0.7299	0.8699	0.6304	0.7194	0.9637	0.8238	0.9157	0.7939	0.7974	0.9502	**0.8671**	0.9380	**0.8544**
FF++	**FF++ Noisy**	0.5344	0.728	0.6163	0.568	0.5469	0.5235	**0.9928**	0.6854	0.7079	0.5445	**0.642**	0.6445	0.6433	0.6479	0.6415	0.5	0.5	0.5	0.5	0.5	0.6137	0.5668	0.5893	0.6164	0.5904	0.5	0.5	0.5	0.5903	0.5	0.6241	0.5742	0.5981	0.643	0.5971	0.6406	0.8187	**0.7188**	**0.7392**	**0.6797**
FF++	**FF++ Compression**	0.9391	0.972	**0.9553**	**0.991**	**0.9545**	0.9262	**0.9868**	0.9555	0.9940	0.9541	**1.0**	0.0025	0.0044	0.5784	0.5011	0.9376	0.0764	0.1419	0.9300	0.5381	**1.0**	0.0498	0.095	0.8811	0.5249	**1.0**	0.0094	0.0186	0.912	0.5047	0.9974	0.3949	0.5657	0.9614	0.6969	0.9778	0.8788	0.9257	0.9858	0.9295
DFDCPreview	**DFDCPreview**	**1.0**	0.999	0.9996	0.999	0.9996	0.9900	0.9993	0.9947	0.9999	0.9946	**1.0**	0.9996	0.9997	0.9997	0.9998	0.9999	0.9986	0.9989	0.9999	0.999	0.9989	0.9985	0.9987	0.9997	0.9905	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.9993	0.9996	0.9999	0.9999	**1.0**	0.9986	0.9993	0.9993	0.9993
DFDCPreview	**DFDCPreview_Blurry**	0.9688	0.851	0.9063	0.969	0.912	0.9985	0.8913	0.9418	0.9981	0.945	0.998	0.7926	0.8838	0.9287	0.8956	0.9967	0.72	0.8372	0.9965	0.86	0.6050	0.9773	0.7474	0.8770	0.6696	**1.0**	0.8493	0.9185	0.9940	0.9246	0.9805	0.9433	0.9616	0.9950	0.9623	0.9993	**0.99**	**0.9946**	**0.9998**	**0.9946**
DFDCPreview	**DFDCPreview Noisy**	0.7944	0.703	0.7461	0.839	0.7606	0.5	**1.0**	0.6666	0.5	0.5	0.5	1.0	0.6666	0.5447	0.5	0.5598	**1.0**	0.6666	0.6015	0.5	0.875	0.0188	0.0365	0.5644	0.508	0.5	**1.0**	0.6666	0.50	0.5	0.5	**1.0**	0.6666	0.6096	0.50	**0.9813**	0.9846	**0.9846**	**0.9988**	**0.983**
DFDCPreview	**DFDCPreview Compression**	**0.9986**	0.998	**0.9986**	0.999	0.9986	0.9695	0.9986	0.9839	0.9994	0.9836	0.757	0.9999	0.8619	0.7940	0.84	0.9943	**1.0**	0.7587	0.9938	0.9939	0.5008	0.9953	0.6663	0.6042	0.5016	0.9375	**1.0**	0.9677	0.9999	0.9666	0.9816	0.9986	0.99	0.9999	0.99	0.9973	0.9993	0.9983	**0.9997**	**0.9983**
Celeb-DF	**Celeb-DF**	0.8997	0.941	0.9197	0.978	0.9178	0.8222	0.9075	0.8628	0.9259	0.8556	0.954	0.9543	0.9544	0.9605	0.9544	**0.9848**	0.9376	0.9406	0.9814	0.9408	0.7878	**0.9673**	0.8684	0.9676	0.8534	0.9227	0.9451	0.9338	0.9773	0.9330	0.9227	0.9653	0.9435	0.9897	0.9422	0.9741	0.9395	**0.9564**	**0.9848**	**0.9572**
Celeb-Df	**Celeb-DF Blurry**	0.8075	0.930	0.8645	0.937	0.8542	0.6291	**0.9757**	0.7649	0.8822	0.7002	0.830	0.9511	0.8865	0.8937	0.8782	0.9600	0.9421	0.8912	0.9574	0.885	0.6978	0.9328	0.7984	0.911	0.7644	0.7483	0.9397	0.8332	0.9174	0.8118	0.8938	0.8618	0.8775	0.9578	0.8797	**0.9965**	0.9432	**0.9547**	**0.9804**	**0.9553**
Celeb-Df	**Celeb-DF Noisy**	0.6041	0.564	0.5832	0.643	0.5971	0.5	0.5	0.5	0.5267	0.50	0.5	**1.0**	0.6666	0.5083	0.50	0.5	0.5	0.5	0.5135	0.50	0.6233	0.5642	0.5923	0.6603	0.6082	0.6241	0.5	0.5	0.5427	0.50	0.5	0.5	0.5	0.4683	0.5	**0.6431**	0.8386	**0.7279**	**0.7431**	**0.6867**
Celeb-Df	**Celeb-DF Compression**	0.8280	0.891	0.8581	0.938	0.8528	0.9468	0.334	0.4938	0.9193	0.6576	**0.973**	0.5248	0.6819	0.9124	0.7551	0.9344	0.534	0.6881	0.9328	0.758	0.8666	0.7687	0.8147	0.9186	0.8252	0.9048	0.7524	0.8216	0.9321	0.8366	0.9378	0.7607	0.84	0.9558	0.8552	0.895	**0.9295**	**0.9120**	**0.9734**	**0.9104**

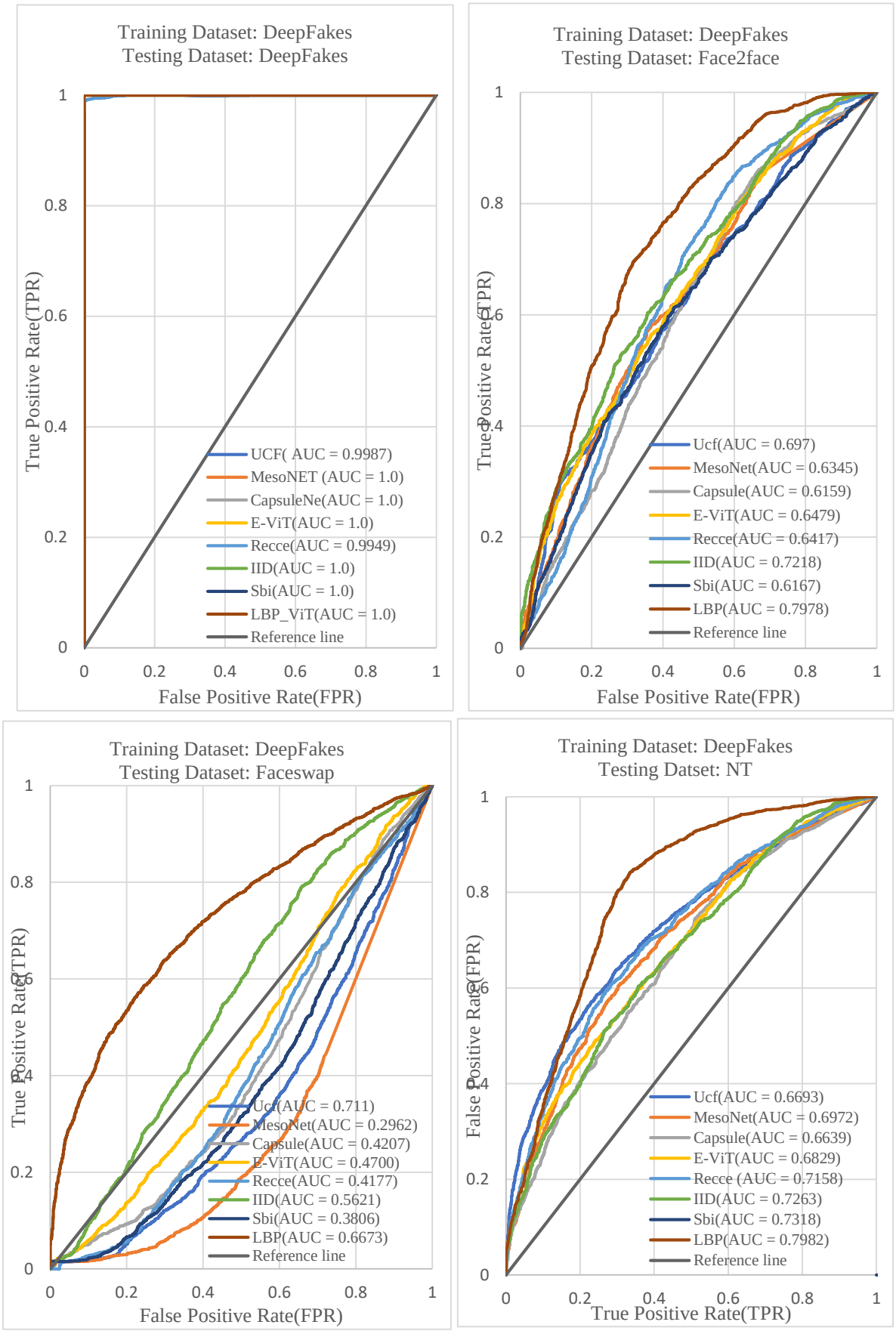


Figure 3.11 ROC curves for the model trained on the DeepFakes dataset and tested on other categories of FF++

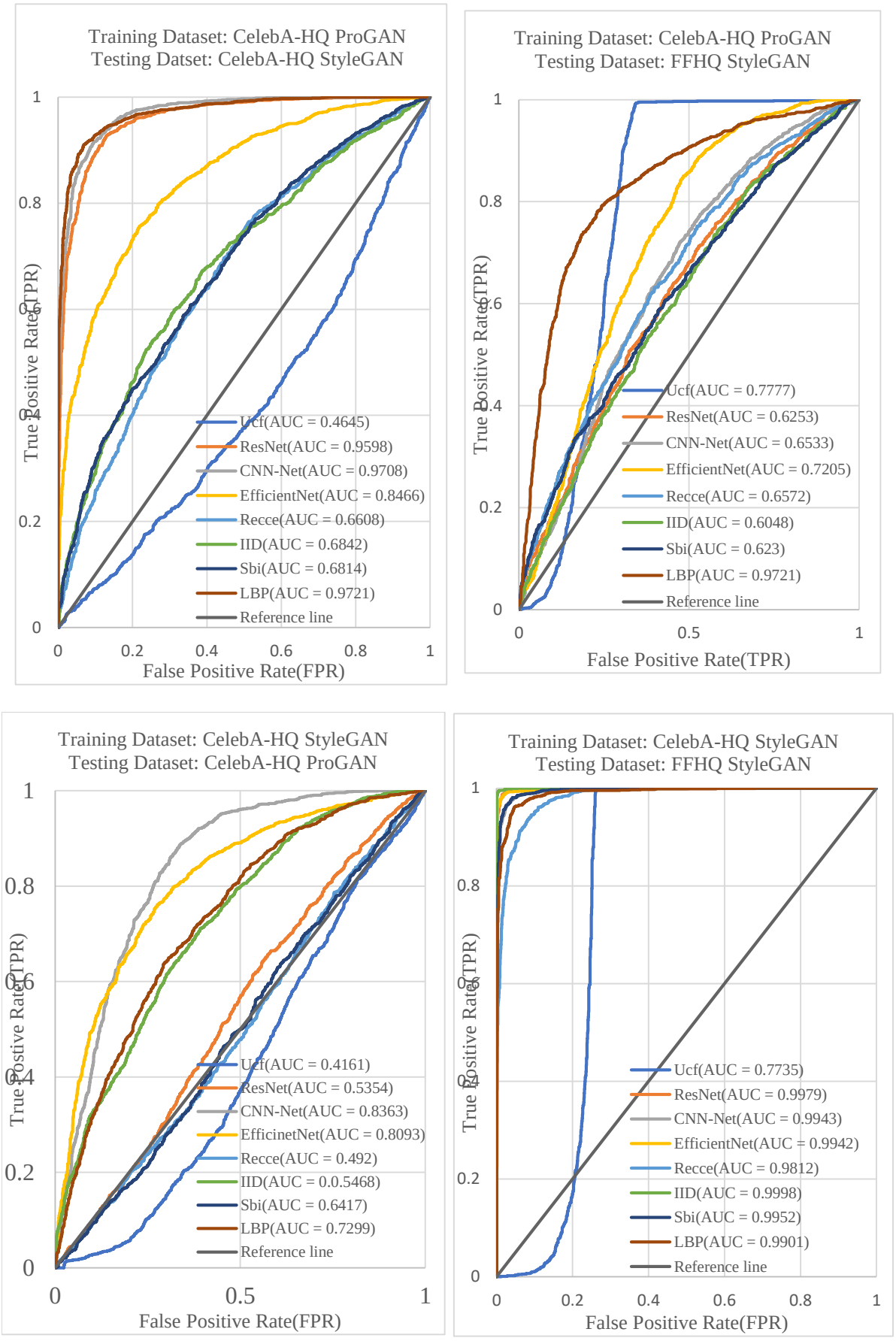


Figure 3.12 ROC curves of the model trained and tested on various sets of GAN images

Original Image **Blurred Image** **Noisy Image** **Compressed Image**

Figure 3.13 Images undergo different post-processing operations

3.3.4 Ablation Studies

Ablation studies are performed for each component of LBP-ViT outlined in this section to confirm its validity. This section employs different categories of the Faceforensics++ dataset to conduct experiments. The components undergo an iterative training process on three FF++ categories and are then tested on a fourth category.

3.3.4.1 Model without the transformer encoder

In this experimentation, the vision transformer encoder has been removed along with the cross-attention (as attention mechanism is a part of transformer encoder), to study the relevance of ResNet and the global texture module. The ResNet18 architecture serves as the basis for this component, and the global texture module is calculated prior to each down-sampling operation in the main branch. Furthermore, these texture features combine until they join with the primary branch(Figure 3.14). Subsequently, these textural features are combined with the main branch and transmitted to the fully linked and sigmoid layer for final classification. depicts the schematic illustration of the model. The classification results are displayed in Table 3.16. The accuracy score ranges from 70% to 85% for several categories of FF++ (except the Face swap category), indicating that global texture features captured at different semantic levels enables it to detect hidden altered artifacts. If we compare the accuracy score of the overall model with the current module, there has been a substantial decrease in the score for DF and NT category, showing the greater importance of visual transformer with its cross-attention mechanism. Transformer encoder enables further learning and refinement of features that has been calculated at different scale to attend to the global and local details with great attention of detailing. Absence of transformer would lead to a substantial decrease in the score of various categories of FF++. Nevertheless, the model continues to experience difficulty accurately categorizing images for face-swap categories. This could be attributed to the limited ability of the model to successfully capture and manipulate textures within a narrow range of swap

boundaries. The model demonstrates its ability to efficiently capture artifacts for several categories, indicating that the texture feature plays a significant role in classification.

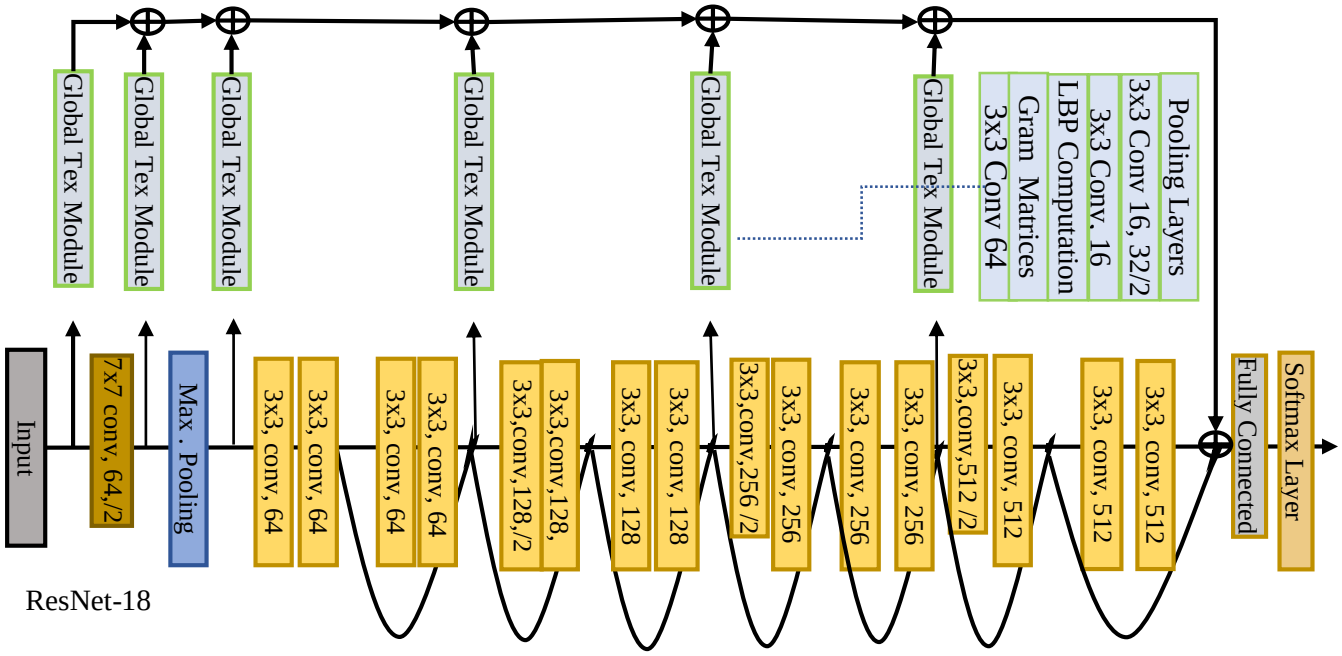


Figure 3.14 Model consisting of ResNet18 with Global texture block where features from both branches are concatenated for classification.

3.3.4.2 Parallel branches cross-attention vision transformer

The component is exhibited in Figure 3.15. In this scenario, two Convolutional Neural Networks (CNNs) located in different branches are utilized to process the input data and generate feature maps of different sizes. Subsequently, these maps are transmitted to the transformer encoder to provide a cross-attention process. The extracted features are subsequently inputted into a multi-layer perceptron, and the resulting output is combined to obtain the final prediction. Figure 3.15 displays the categorization outcome of this component. The transformer encoder learns the dependencies between the entire input image while calculating the relevance of the different pixels of the input image concerning each other, owing to the self-attention mechanism of the transformer encoder. So, the Global contextual relationship of a transformer includes a broader range of commonality of patterns over longer distances, including various patterns like intensities, brightness or spatial characteristics of an image. This method captures relationships between different pixels regardless of location by giving each one a "global view" of the entire image. Therefore, incorporating the global perspective of spatial attributes in their feature space may improve performance in the final classification task compared to the abovementioned ResNet design. The outcome is an

improvement in accuracy scores compared to the previous component, ranging from 60% to 70%. This suggests that regular CNN features with cross-attention cannot effectively classify various changed images. The attention mechanism is crucial in cases where texture features lack discriminability, as seen by the face swap examples, where the score surpasses the prior component. The primary inference from this component is that effectively capturing the essential underlying characteristics that consistently exist across several modified samples necessitates more than a transformer equipped with cross-attention.

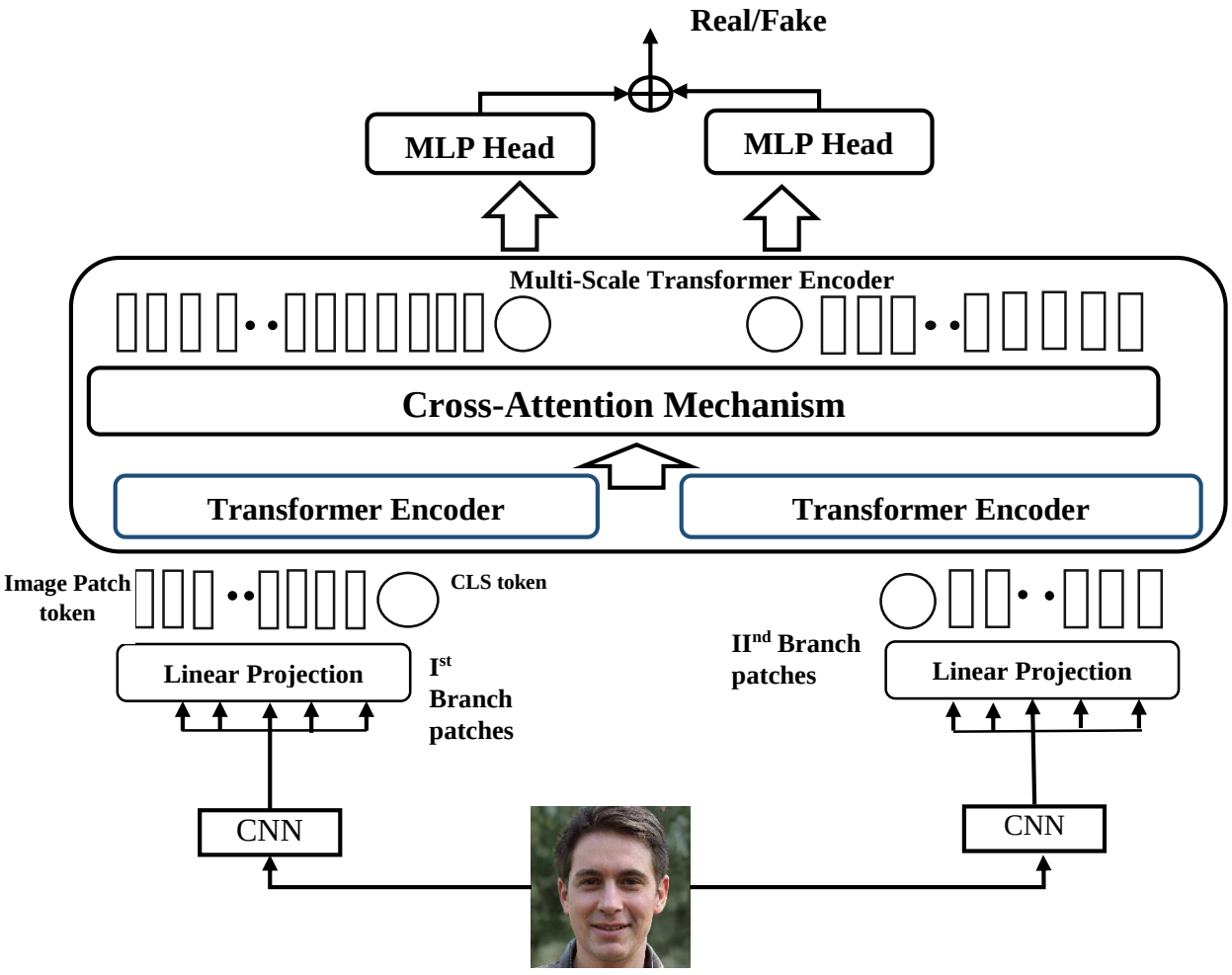


Figure 3.15: Model diagram of parallel-branch cross-attention vision transformer where the image is initially passed through CNNs to create feature maps of different sizes.

3.3.4.3 Model without the cross-attention mechanism

A simple concatenation strategy has been employed, where the tokens from both branches are combined without regard for their scale and branch. Therefore, the fusion equation will resemble the following:

$$\hat{y} = [f^1(\mathfrak{X}^1_{patch\ \&\ cls}) || f^2(\mathfrak{X}^2_{patch\ \&\ cls})] \tag{3.26}$$

$$\gamma = \hat{y} + \mathfrak{MCA}\langle \mathcal{LN}(\hat{y}) \rangle \tag{3.27}$$

Here, $f(.)$ is used for the dimension alignment. In Table 3.16, part III displays the score for the tested model without the cross-attention mechanism. The score across all categories of FF++ decreases by approximately 2-4% compared to the overall model with the cross-attention mechanism. The cross-attention mechanism enforces the self-attention mechanism to concentrate on the specific patches within and across smaller regions of varying scales, enabling the model to improve the feature representation of local and global features. Incorporating multi-scale features into the analysis enabled more effective processing of local and global characteristics, resulting in enhanced performance. The lack of an attention mechanism results in a decrease in performance by 2-4%. It is worth noting that the performance of the NT category of FF++ is lower by around 1%. This is likely because the NT dataset utilizes a texture mechanism in its development, which is readily obtained via the Global Tex module. As a result, there is less reliance on the cross-attention mechanism compared to other models.

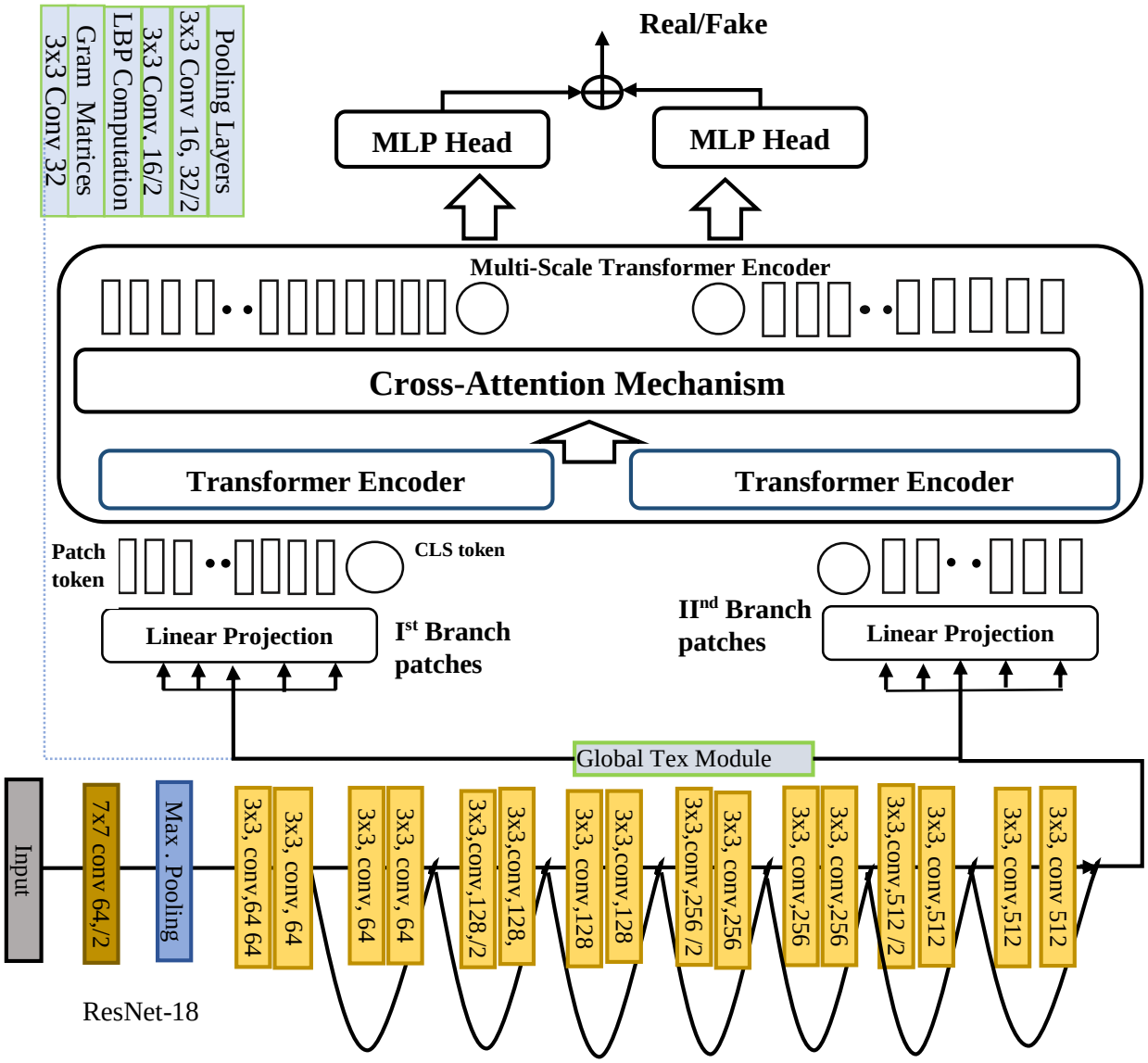


Figure 3.16: Model without the multi-layer aggregation of the texture. A single global texture module is computed and then fed into the dual branch of the vision transformer.

3.3.4.4 Model without the multi-layer aggregation of global texture

Figure 3.16 illustrates the model's diagram, where a single global texture is applied after the ResNet layer and inputted into the vision transformer's dual branch. The results are displayed in Table 3.16 in section 3. There is a noticeable decrease in performance, approximately 6-8% compared to the overall model. One possible explanation for the performance issue is that CNN has a limited receptive field, which means it struggles to capture dependencies over longer distances. Additionally, the texture calculated on this feature map may not effectively capture long-range feature modelling, leading to a decrease in the model's performance. In addition, calculating the texture at various semantic levels makes it possible to capture detailed information such as edges, corners, shapes, and objects. This leads to a more comprehensive and enhanced feature representation. In addition, harnessing texture

information from various scales also enhances gradient flow during back-propagation, enabling models to adapt and process intricate scenes.

3.3.4.5 LBP-ViT (ResNet with Global Texture Module + Cross-Attention Transformer)

The model described in this paper integrates the cross-attention transformer and texture block with ResNet. The component above suggests that texture can be a promising characteristic for detecting deepfakes, even when subjected to various sorts of deepfake manipulation. An inductive bias-free cross-attention vision transformer focuses on the manipulation details at the same time. The cross-attention mechanism enhances, concentrates, and enhances the conventional CNN features after the texture has been calculated and combined. Consequently, this leads to a more powerful feature representation and enhances the model's overall performance. The Global relation of a transformer may include a broader range of commonality of patterns over longer distances, which may include global texture and other patterns like intensities, brightness or spatial characteristics of an image. So, their feature space includes all sum-Bonam of all spatial characteristics; hence, texture information may get subdued, and the final classification task may give better results. While explicitly computing global texture features(as the texture is one of the potential features for discrimination) and then getting fed to the transformer would enable the entire architecture to focus on such features, resulting in better classification results, as shown by Table 3.16. The table shows that this potent combination produces more than 83% accuracy for Face2face and Deepfake data, 78% for NT samples, and more than 66% for face-swap samples. This indicates that the model can learn discriminative characteristics that hold across various sample types.

Table 3.16 displays the classification results for each component. Components are trained on three categories and tested on the fourth category of FF++ recursively.

Train	Test	Precision	Recall	F1 Score	AUC	Acc
Model without the transformer encoder						
F2F+FS+NT	DF	0.8352	0.646	0.7285	0.8703	0.75
DF+FS+NT	F2F	0.9009	0.769	0.8297	0.9275	0.84
DF+F2F+NT	FS	0.7989	0.379	0.5146	0.6489	0.642
DF+F2F+FS	NT	0.7833	0.6075	0.6843	0.7972	0.71
Dual-branch transformer with Cross-Attention Mechanism						
F2F+FS+NT	DF	0.668	0.7875	0.7231	0.7639	0.6985
DF+FS+NT	F2F	0.6650	0.691	0.6778	0.7378	0.6715
DF+F2F+NT	FS	0.644	0.5475	0.5920	0.6628	0.6227
DF+F2F+FS	NT	0.6943	0.66	0.6767	0.7324	0.68475

Model without multi-layer aggregation of feature						
F2F+FS+NT	DF	0.8425	0.8735	0.8577	0.9142	0.8575
DF+FS+NT	F2F	0.8501	0.7508	0.7977	0.88752	0.8107
DF+F2F+NT	FS	0.5863	0.6135	0.5995	0.6525	0.6425
DF+F2F+FS	NT	0.7334	0.8642	0.79344	0.8674	0.7724
Model without cross-attention Mechanism						
F2F+FS+NT	DF	0.7988	0.8465	0.8219	0.8766	0.8195
DF+FS+NT	F2F	0.8035	0.7288	0.7643	0.8556	0.7739
DF+F2F+NT	FS	0.5776	0.6079	0.5924	0.6475	0.6356
DF+F2F+FS	NT	0.6988	0.8216	0.7187	0.8266	0.7475
Tex-ViT(ResNet+Texture+ Cross-Attention Transformer)						
F2F+FS+NT	DF	**0.8642**	**0.9005**	**0.8819**	**0.9425**	**0.8795**
DF+FS+NT	F2F	**0.8708**	**0.7785**	**0.8221**	**0.9245**	**0.8315**
DF+F2F+NT	FS	**0.6070**	**0.6365**	**0.6214**	**0.6725**	**0.6642**
DF+F2F+FS	NT	**0.7446**	**0.879**	**0.8062**	**0.8771**	**0.7887**

3.3.5 Visualization outcomes of the LBP-ViT's predictions

This section visually illustrates the specific area of interest for the LBP-ViT prediction. The model is trained using all the images in the dataset, and the GradCAM class activation maps are used to determine the specific regions of focus for the classifier during prediction. The model's forecast zone is illustrated in Figure 3.17. The model focuses on specific texture regions to get the visualization results.

FF++ (DF and NT)

Celeb -DF & DFD CPre view

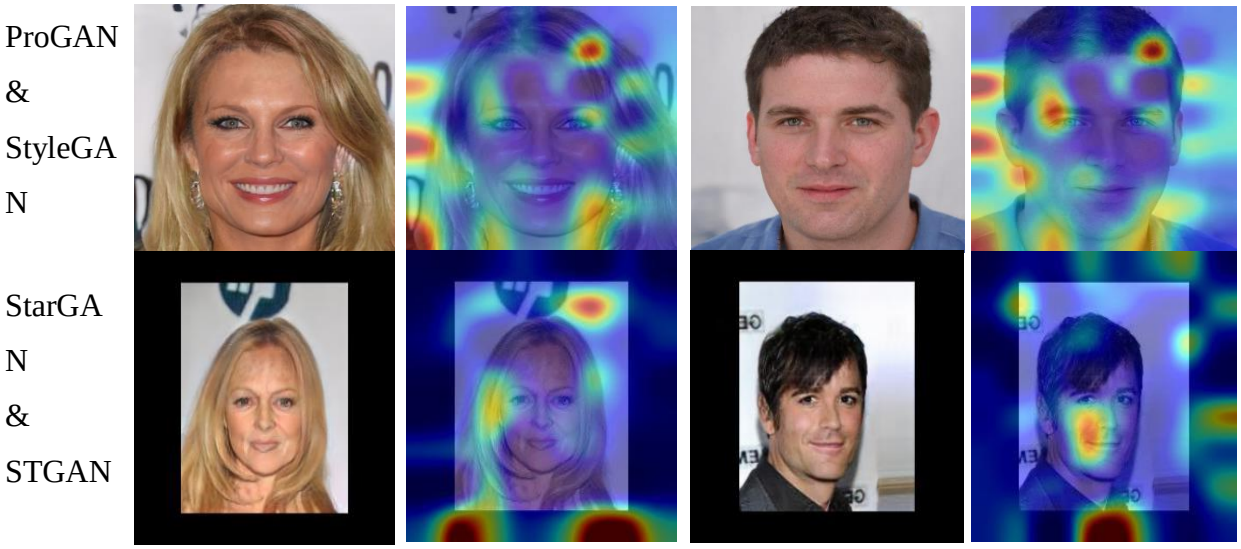


Figure 3.17: LBP-ViT's model's region of focus for various datasets

3.3.6 Conclusion

Empirical tests in this research validate that counterfeit facial images have a more even texture, and this overall information is not maintained throughout a significant spatial range. These investigations demonstrate that texture statistics are crucial and serve as a shared characteristic in the distribution of false face images. The proposed model utilizes ResNet-18 as the underlying architecture for extracting conventional convolution features. Additionally, it incorporates a textures module that computes the global utilizing gram matrices, together with Local binary patterns operation, before each down-sampling operation of ResNet-18. The texture information is continuously accumulated layer-by-layer and then combined with the usual features of ResNet-18 to produce the input for the dual-branch cross-attention-based vision transformer. An experiment was conducted to detect cross-forgery using several categories of Faceforensics++ and other GAN images. The results indicate that the model performed exceptionally well and outperformed numerous state-of-the-art models. In order to demonstrate the resilience of the model across many situations, we conducted experiments using the FF++, DFDCPreview, and Celeb-DF datasets. These experiments involved performing operations such as blurring, adding noise, and compression. This experiment demonstrates the model's ability to generalize to different types of unseen images and its resilience to various post-processing methods.

3.4 Significant outcome of this Chapter

The following are the significant results of this chapter:

- Proposed two novel architecture based on texture and cross-attention mechanism i.e. Tex-ViT and Tex-Net for generalized deepfake detection.

- Tex_ViT uses gram-matrices for texture computation. The model collaborates conventional ResNet features with a texture module that runs parallel acts on parts of ResNet before every down-sampling operation and serves as an input to the dual branch of the cross-attention vision transformer.
- Tex-Net model architecture uses the combination of Gram-matrices and Local Binary Patterns(LBP) for texture computation. The global texture is calculated at each down sampling operation of ResNet, and then layer features are aggregated at multiple layers. These features continue to combine before being fed into the dual-branch cross-attention-based vision transformer for the classification.
- Experimentation was conducted on different categories of FF++ in a cross-manipulation setting and different GAN dataset images in cross-domain settings to demonstrate the model's generalization ability of both models. Both model demonstrates superior performance compared to other SoTA models, providing further evidence of the strength of texture features.
- Through experimentation on FF++, DFDCPreview, and Celeb-Df, data samples were subjected to different post-processing operations such as blurring, noise addition, and compression. The results showed that the both models performed exceptionally well and demonstrated strong robustness against adversarial operations.

The subsequent research studies serve as the foundation for this chapter.

1. **D. Dagar** and D. K. Vishwakarma, “Tex-ViT: A Generalizable, Robust, Texture-based dual-branch cross-attention deepfake detector,” Under Review in ***Journal of Information Security and Applications*** (Pub: Elsevier), https://arxiv.org/abs/2408.16892
2. **D. Dagar** and D. K. Vishwakarma, “Tex-Net: Texture-based parallel branch cross-attention generalized robust deepfake detector” vol 30, article number 233, ***Multimedia Systems***, 2024 (Pub: Springer), doi: https://doi.org/10.1007/s00530-024-01424-7

Chapter 4: Deepfake video Dataset and framework for deepfake video detection

4.1 Scope of this Chapter

This chapter is dedicated to the problem of proposing a video dataset named, Div-Df, having variety of manipulation. This dataset is composed of 150 real videos of different celebrities of different professions and 250 deepfake videos (100 face-swap videos, 100 facial enactment videos, and 50 lip-sync videos). The dataset consists of real and fake videos of various famous personality's speeches and interviews. Additionally, proposed a deepfake video detection model that combines Xception and LSTM pretrained models with channel and spatial attention mechanisms (CBAM). The latent spatial artifacts are captured by Xception using depthwise separable convolution, while the differences between the altered sequences are captured by LSTM. This hybrid model assembly allows for the learning of the spatial and temporal distortions along various dimensions and is an effective tool for deepfake identification. Benchmarking is done using the proposed framework and various state-of-the-art methods on the proposed dataset which shows the superiority of the proposed model against various state-of-the-art models.

4.2 Div-Df: A Diverse Manipulation Deepfake Video Dataset

4.2.1 Abstract

Recent advances in image and video manipulation have given rise to grave concerns. Deepfake technology employs deep learning techniques to produce astoundingly lifelike content. Deepfakes are risky since they have the ability to counterfeit someone's identity by replacing their face with that of another person or generating random noise in the mouth area. Additionally, with just a few seconds of audio, AI-based deep learning models can replicate any person's voice. Detecting such videos is the only promising defense against such fraudulent data. Several deepfake datasets have been made available to help in deepfake detector training and testing, including DF-TIMIT [139], FaceForensics++ [15], Celeb-DF [140], DFDC [141], Deeperforensics1.0 [142], etc. Even though this has significantly improved deepfake detection methods, they are still unable to capture real-world scenarios entirely, as most of the dataset is face-swap manipulation. To bridge this gap, we have proposed a Div-DF dataset containing various types of video manipulation like face swap, facial reenactment, and lip-sync. This dataset is composed of 150 real videos of different celebrities of different professions and 250 deepfake videos (100 face-swap videos, 100 facial reenactment videos, and 50 lip-sync videos).

Deepfake videos are synthesized using state-of-the-art Face-Swap GAN(FSGAN) and the Wav2Lip method. The dataset contains high-quality samples of face-swapped and lip-sync videos, while the samples of face-re-enactment are of average quality. We have tested state-of-the-art detection and image classification models to standardize our dataset's baseline evaluation of various detection methods. We have done a comprehensive assessment along different metrics and found that our dataset is challenging and represents real-world samples.

4.2.2 Proposed Diverse Video Deepfake Dataset(Div-DF)

Div.-DF consists of varied video manipulation types, including Face swap, face reenactment, and lip-sync. Existing datasets do not include such variety in their dataset; they majorly focus on the face swap. Table 4.1 focuses on a variety of existing datasets. There are 400 videos in Div-DF, 150 of which are genuine and 250 of which are deep fakes (which include 100 face-swap videos, 100 face-reenact videos, and 50 lip-sync videos). Each video is typically 15 seconds long, runs at 30 frames per second, and has a minimum resolution of 480 pixels.

Table 4.1 A Comparison of DIV-DF with the existing Deepfake video dataset on various parameters

Dataset	**Manipulation Types**	**#Actors**	**Real Video Source**	**#Real Videos**	**#Fake Videos**	**Resolution**
UADFV [66]	Faceswap	49	Youtube	49	49	294x500
Df-TIMIT [139]	FaceSwap	32	VidTIMIT	640	320	64x64(LQ), 128x128(HQ)
FF++ [15]	Face Swap, Reenactment	977	Youtube	1000	4000	480p, 720p, 1080p
DFDC [141]	Faceswap	960	Volunteer Actor	23654	104500	1080x1920
DFDC Preview [143]	Faceswap	66	Volunteer Actor	1131	4113	1080x1920
Google DFD [144]	FaceSwap	28	-	363	3068	1080x1920
Celeb-DF [140]	FaceSwap	59	Youtube	590	5639	256x256
DeeperForensics 1.0 [142]	FaceSwap	100	Videos Shoot	50000	10000	1920x1080
WildDeepfake [145]	Faceswap	-	Internet	3805	3509	-
FakeAVCeleb [146]	FaceSwap, Lip-Sync, VC	600+	VoxCeleb 2.	500	20000	-
Div-DF(Ours)	**FaceSwap, LipSync, Facial reenact.**	**30**	**Youtube**	**150**	**250**	**360p, 480p, 720p, 1080p**

4.2.2.1 Data Collection

The authentic videos are selected from publicly accessible YouTube videos that correlate to speeches and interviews given by different celebrities who are of different ages, genders, and ethnicities. These videos are downloaded using the yt-dlp software, and then a portion of a video is extracted using FFmpeg software. Audios (English language) for the lip-sync manipulation are also extracted using the FFmpeg software. Five videos are collected for each subject, and there are 30 such subjects which constitute 150 real videos. The real dataset comprises 60% males and 40% females. Figure 4.1 presents the statistics of our real dataset sequences along various dimensions like profession, geographical location, age group, and resolution. We can see that 7% of the Identities lie in the age group of 18-35 years, 36% in the age group of 35-60 years, and more than 37% lie in the age group more significant than 60 years, and 20% have already died. The different professions of identities are politician followed by sportsman, actor, and others. The resolution of most sequences is excellent, around 1080p and 720p. A few videos are also of the resolution 480p and 360p, especially of the dead identities. The collected dataset is diverse in identities, illumination conditions, poses, and expressions.

4.2.2.2 Synthesizing methods

Our fake videos have three kinds of manipulation, i.e. Face-Swap, Facial reenactment and Lip-Sync. For generating the videos of these manipulations, we have used two well-known methods:

FSGAN [147]:: The Framework uses a unique recurrent neural network to do facial reenactment and face swapping for two identities. The model is identity-agnostic and applicable for both images or a video sequence. For video sequences based on reenactment, Delaunay triangulation, and barycentric coordinates, the model employs continuous face view interpolation. Afterward, a face blending network for maintaining skin tone, poses, and lighting conditions was used to combine the faces seamlessly. Model is subject-agnostic means it does not require training of the targeted faces.

Wav2Lip [148]: A lip-sync technique that syncs the mouth region according to arbitrary audio. The authors identified the weakness of earlier lip-sync methods, proposed a different loss function, and designed a powerful discriminator that produces highly realistic lip-sync samples.

FSGAN model generates face-swap and facial reenactment samples without requiring training on the input samples. Face-Swap-generated examples are highly realistic, while the

reenactment-generated samples are of the average medium in quality. The Wav2lip method is used to generate lip-sync samples of the dataset, developed are of good quality and hence challenging to spot as false with the naked eye. Figure 4.2 represents the visual samples of the dataset.

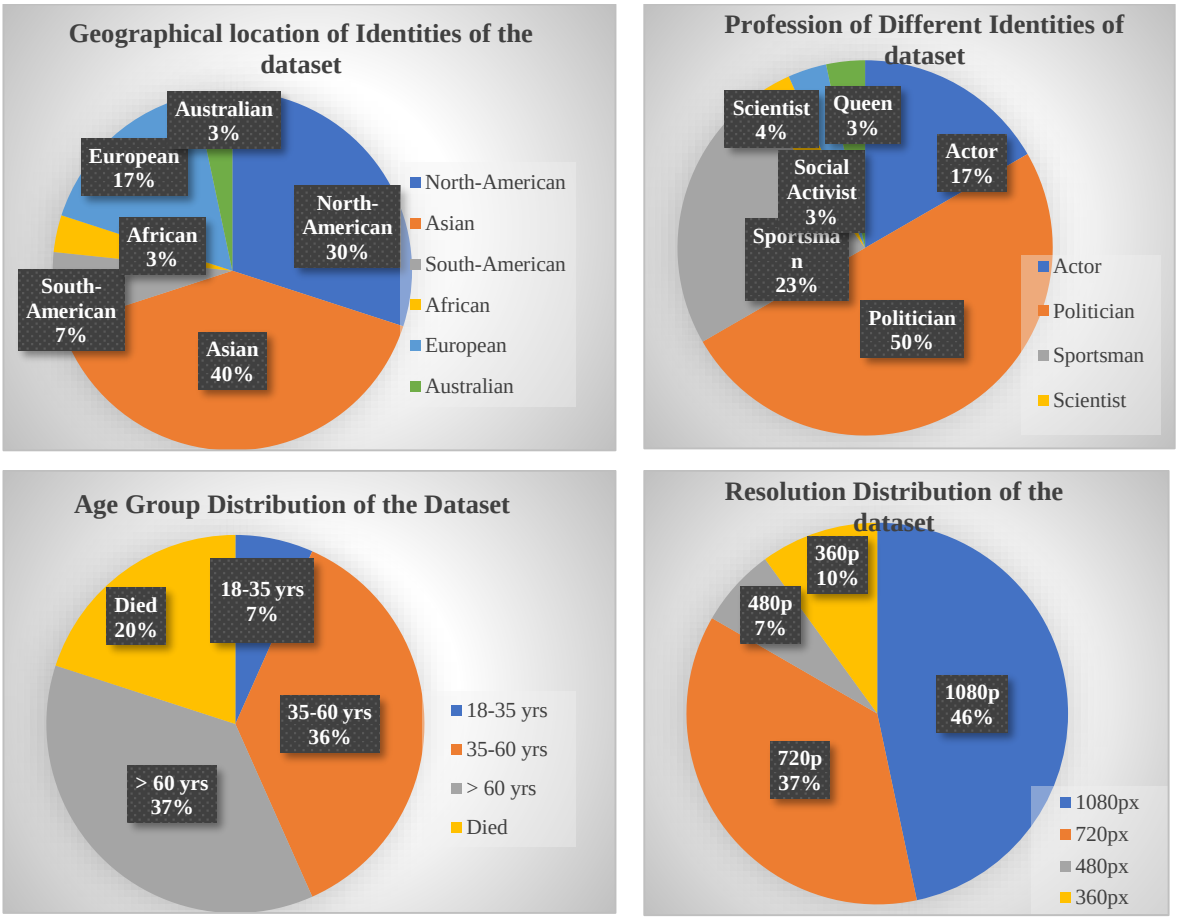


Figure 4.1 Statistics of Div-Df dataset along different dimensions

Figure 4.2 Samples of various categories of the Div-Df dataset

4.3 Deepfake video detection using a hybrid Xception-LSTM model with spatial and channel attention

4.3.1 Abstract

Recent advances in image and video manipulation have given rise to grave concerns. Deepfake technology employs deep learning techniques to produce astoundingly lifelike content. Deepfakes are dangerous since they have the ability to spoof someone's identity by swapping out their face for another person's or generating random noise from the mouth area. The only promising defense against such fake data is the detection of such videos. We have developed a deepfake detection model that combines Xception and LSTM pretrained models with channel and spatial attention mechanisms (CBAM) to counter the user's malevolent intent. The latent spatial artifacts are captured by Xception using depthwise separable convolution, while the differences between the altered sequences are captured by LSTM. This hybrid model assembly allows for the learning of the spatial and temporal distortions along various dimensions and is an effective tool for deepfake identification. The evaluation is conducted using our recently suggested Div-DF dataset, which includes various forms of video alteration such face swap, facial reenactment, and lip syncing. The evaluation reveals that our model performs well on a variety of datasets and can outperform a number of state-of-the-art deepfake detection and image classification models.

4.3.2 Proposed Framework

The model comprises three major components: CBAM, XceptionNet, and LSTM (Long short-term Memory)(Figure 4.3).

CBAM [149]: Channel attention exploits the relationship between the inter-channel of different modules, computed by squeezing the channel information along various channel axes. Then, the spatial information of the feature maps is aggregated using a single hidden layer with a shared multi-layer perceptron network after max-pooling and average-pooling operations with a single hidden layer. Channel attention focuses on the ‘what’ part, while spatial attention focuses on the ‘where’ information part of the feature maps.

XceptionNet [129]: The “extreme” version of Inception V3, where the spatial correlation and cross-channel correlation are completely separated. The design is a linear stack of 14 modules with residual connections in between 36 depth-wise separable convolutional layers. This novel decoupling of correlation resulting in a substantial reduction in the number of parameters and, hence, less overhead of computations.

LSTM [150]: LSTM captures the long-term dependencies between the input samples. It is mainly used for sequence data or temporal data. LSTM removes vanishing gradient problems, leaving the training model unaltered and handling dispersed representations, continuous values, and noise, which are used to bridge long lags in some problems. Also, LSTMs do not require keeping a limited number of prior states. For our model, LSTM is employed to capture the temporal discrepancies between the frame of the video samples.

Input video samples are broken into the frames fed to the CBAM model to refine the representation of the feature maps. Then, these representations are passed to the XceptionNet pre-trained model to capture intrinsic and latent spatial artifacts. Then, such representations are passed to the LSTM model to learn the temporal discrepancies between the frames.

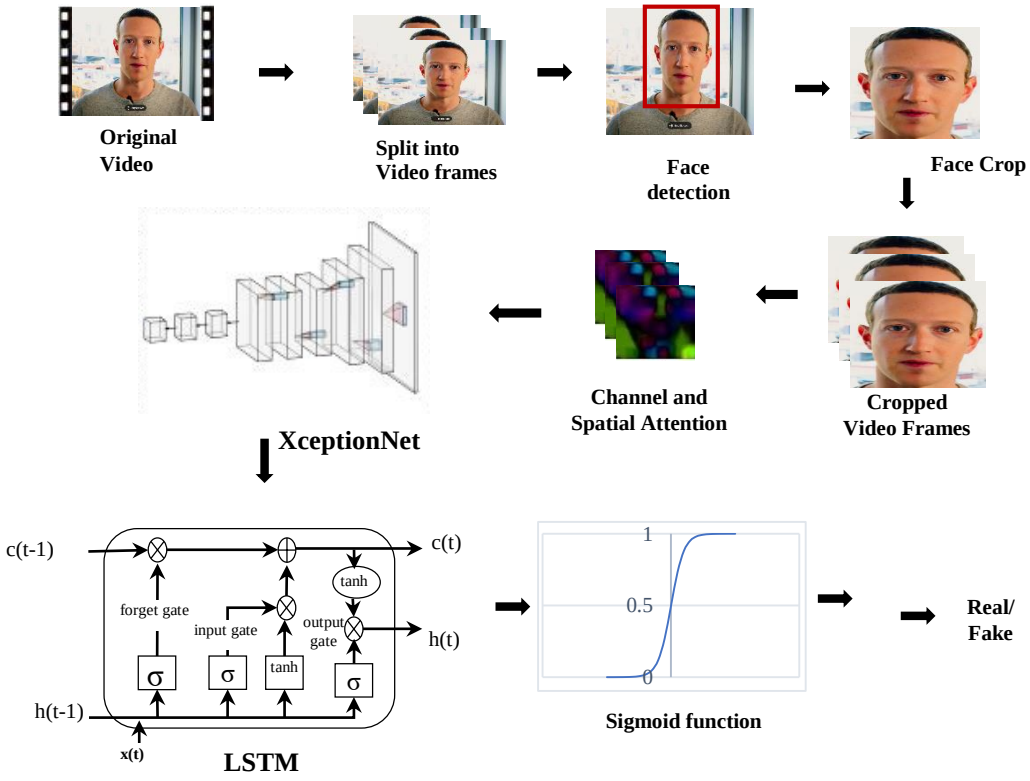


Figure 4.3: The proposed workflow for the deepfake detection where the original video was divided into frames and faces were detected using MTCNN face detection. The cropped faces are first fed into the CBAM module, and then the refined representation is passed to passed to the combination of XcpetionNet and LSTM to learn artifacts and then to the softmax function for prediction.

4.3.3 Experimentation

The selection of training hyper-parameters, various datasets, the choice of the face extractor, and all of the experiment scenarios will be covered in this section.

4.3.3.1 Experimental settings:

The starting learning rate is set at 0.1. 64 is taken as the batch size. Using the Adam optimizer, the model's parameters will be updated. Each experiment is run for 100 epochs, and the experiments are run for 24GB NVIDIA TITAN RTX GPUs.

4.3.3.2 Dataset Pre-Processing

Our algorithm as well as different deepfake detection and image classification models have been tested on the Div.-DF dataset. There are 400 videos in Div-DF, 150 of which are genuine and 250 of which are deepfakes(which include 100 face-swap videos, 100 face-reenact videos, and 50 lip-sync videos). Each video is typically roughly 15 seconds long and has a frame rate of 30fps, and the resolution of the videos is 480px or more. Videos are initially divided into frames, and their faces are extracted using the MTCNN face extractor. Around 300-400 frames are extracted for the videos. The training set, validation set, and testing set were each given a portion of the dataset that was split into three sets with a ratio of 70%:15%:15%.

4.3.3.3 Evaluation

We tested out our model in the experiment. For comparison with our model, a number of deepfake and image classification models are taken into consideration. Models are initialized with the pre-trained weights if they are available. The model is trained using the training dataset, and it is then evaluated against the validation dataset; weights from models that perform well during validation are kept for use on the test dataset.

Various models that are considered for the evaluation are as follows:

- MesoInception-4 model [125].
- Residual-Net50 [136].
- CNN-Net [130].
- Efficient-Net [137].
- CViT [151].
- Capsule Network [152].
- Global Texture [153].
- E-ViT [126].

Code for these models was pulled from their GitHub repository, modified for the dataset, and given more assessment metrics for a more thorough evaluation. Table 4.2 contains the benchmark score of different deepfake detection models on our Div-Df dataset. The models' average accuracy is about 85 per cent, showing the vulnerability of the models in a diversified scenario. Our model can beat the score of various other models, showing our model's best performance in diversified scenarios. MesoNet, vision transformer and Efficient_B0 have great difficulty functioning in such diversified conditions. Figure 4.4 highlighted the ROC, i.e., different models' probability curves or detection capability. Almost all the model has an AUC score of more than 90% but less than 95%, demonstrating the typical performance of these contemporary models on this variety of datasets.

Table 4.2: Benchmark score of our model and different model on the Div-DF dataset.

Models	**Precision**	**Recall**	**F1-Score**	**AUC**	**Accuracy**
Capsule-Net [154]	0.9340	0.8496	0.8898	0.9405	0.8680
CNN-Net [130]**.**	**0.9766**	0.8865	0.9019	0.9544	0.8791
CViT [151]	0.8148	0.8888	0.8502	0.9196	0.8201
EfficientNet_B0 [131]	0.6753	**0.9936**	0.8040	0.7394	0.6962
E-ViT [126]	0.8762	0.8605	0.8683	0.9106	0.8357
Gram-Net [115]	0.8835	0.9318	0.9070	0.9647	0.88
MesoNet [125]**.**	0.7840	0.6994	0.7393	0.7862	0.6907
ResNet50 [155]**.**	0.8827	0.8839	0.8831	0.9328	0.8533
Proposed Model	0.9555	0.9364	**0.9458**	**0.9855**	**0.9306**

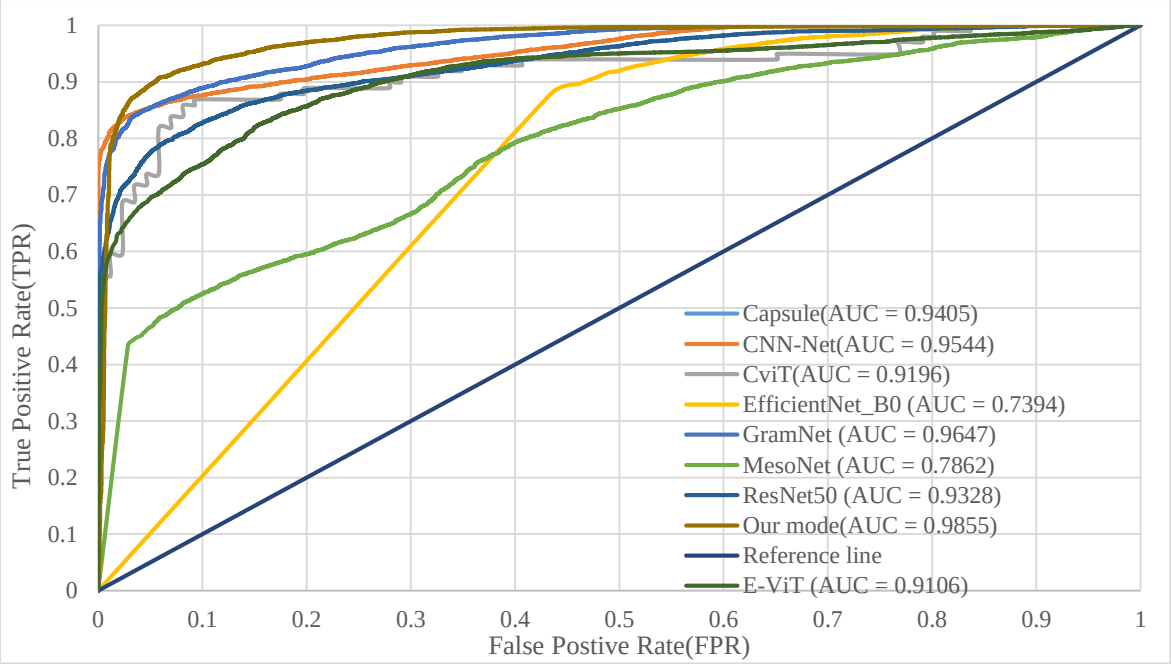


Figure 4.4: ROC curve of various deepfake detection models

4.3.4 Conclusion

In this study, we introduced a dataset called Div.-DF that contains various deepfake video manipulation, i.e. Face-Swap, facial reenactment and Lip-sync. The dataset comprises 150 real videos and 250 deepfake videos, divided into 100 faces-swap, 100 facial reenactments, and 50 lip-sync videos. The data samples are of high visual quality, especially face-swap and lip-sync videos. We have also proposed deepfake detection method that captures the latent, intrinsic spatial and temporal discrepancies among the manipulated samples. The Xception pre-trained model is employed to capture the spatial artefacts, and to catch long-term differences among the artificial samples, we used LSTM. The input samples are initially passed through the CBAM module to refine the representation. We have standardized the benchmark evaluation of our model and compared it against different deepfake detection and image classification methods. Models perform well on the dataset that represents the diversified scenarios of manipulation.

4.4 Significance outcome of this chapter

The significance outcome of this chapter are as follows:

- Proposed a novel diverse manipulation dataset named Div-Df and a framework for deepfake video detection.

- Div-DF dataset consists of real and various manipulated videos like Face Swap, Facial reenactment, and lip-sync. YouTube is used to collect real videos, while fake videos are created using FSGAN and Wav2Lip techniques.
- Framework for deepfake video detection is proposed consisting of Xception model with spatial and channel attention to capturing spatial artifacts and LSTM to capture long-term dependencies or time discrepancies among the manipulated samples.
- Evaluated the performance of the proposed model and various state-of-the-art methods on Div-Df dataset and found that suggested model performs when videos are subjected to various manipulations, with performance on par with other SoTA models.

The subsequent research studies serve as the foundation for this chapter.

1. **D. Dagar** and D. K. Vishwakarma "Div-Df: A Diverse Manipulation Deepfake Video Dataset" ***IEEE Conference: Global Conference on Information Technologies and Communications(GCITC)***, Bengaluru. (2023), doi: 10.1109/GCITC60406.2023.10426446.
2. **D. Dagar** and D. K. Vishwakarma **"**A Hybrid Xception-LSTM model with channel and Spatial Attention for Deepfake Video Detection**"** ***IEEE Conference: International Conference on Mobile Networks and Wireless Communications,*** Tumakur, Karnataka. (2023), doi: 10.1109/ICMNWC60182.2023.10435983.

Chapter 5: Localization for Deepfake Manipulation

5.1 Scope of this Chapter

This chapter is dedicated to the problem of deepfake manipulation in visual data. To solve this problem, a novel framework. To address this issue, developed a dual-branch model that integrates handmade feature noise with Convolutional Neural Networks (CNNs) as an Encoder-decoder (ED) system enhanced by the attention mechanism. This model utilises a dual-branch approach, where one branch incorporates noise features and the other branch incorporates RGB features. These branches are then combined and fed into an ED architecture for the purpose of semantic learning. Additionally, skip connections are employed to preserve spatial information. The shallowfakes dataset (CASIA, COVERAGE, COLUMBIA, NIST16) and deepfake dataset Faceforensics++ (FF++) were extensively tested to showcase their exceptional ability to extract features and outperform various baseline models.

5.2 Shallowfake and Deepfake Image Manipulation Localization using Noise and RGB-based Dual Branch method

5.2.1 Abstract

The reliability of multimedia is being progressively tested by sophisticated Image Manipulation localization(IML) methods, which has led to the creation of the IML domain. A good manipulation model requires extracting non-semantic differences features between manipulated and authentic regions to exploit artifacts, which calls for explicit comparisons between the two areas. Existing models either use handcrafted-based feature methods, convolutional neural networks (CNNs), or a combination of both. Handcrafted feature methods assume the tampering beforehand, limiting their capabilities for diverse tampering operations, while CNNs model semantic information, which is not enough for the manipulation artifact. To improve these limitations, we have designed a dual-branch model that combines handcrafted feature noise and CNNs as an Encoder-decoder(ED) powered by the attention mechanism. This dual-branch model uses noise features on one branch and RGB on the other before feeding to an ED architecture for semantic learning and skip connection deployed to retain spatial information. Furthermore, this architecture uses channel spatial attention to strengthen further and refine the features' representation. Extensive experimentation on the shallowfakes dataset (CASIA, COVERAGE, COLUMBIA, NIST16) and deepfake datasets Faceforensics++(FF++) to demonstrate the superior feature extraction capabilities and

performance to various baseline models with AUC score even reaching 99%. Also, it is one of the first methods to perform localization on the deepfake dataset. The model is relatively lighter, has 38 million parameters, and easily outperforms other State-of-the-Art(SoTA) models.

5.2.2 Proposed Methodology

The proposed model consists of two parallel branches; one uses an RGB image as an input, while the other uses noise/residual features as an input feature determined by the Bayar convolution and SRM convolution filters. These kernel filters suppress semantic content and enhance low-level manipulation traces(Figure 5.1). The channel refines RGB and noise feature representation of manipulation and spatial attention maps generated by their respective modules. The input features are multiplied with the attention maps to strengthen discriminative features. ED architecture is employed where the encoder translates the intermediate features into discriminative feature maps, which are then further processed and recovered by the decoder to generate classification predictions down to the pixel level. Skip connections improve the propagation capabilities of encoder-decoder network features. A dual attention network further improves feature representation by adapting global dependencies along the spatial and channel axes and local semantic characteristics, resulting in a more precise manipulation of localized features. The resulting features from both branches are concatenated for a more accurate final manipulation prediction.

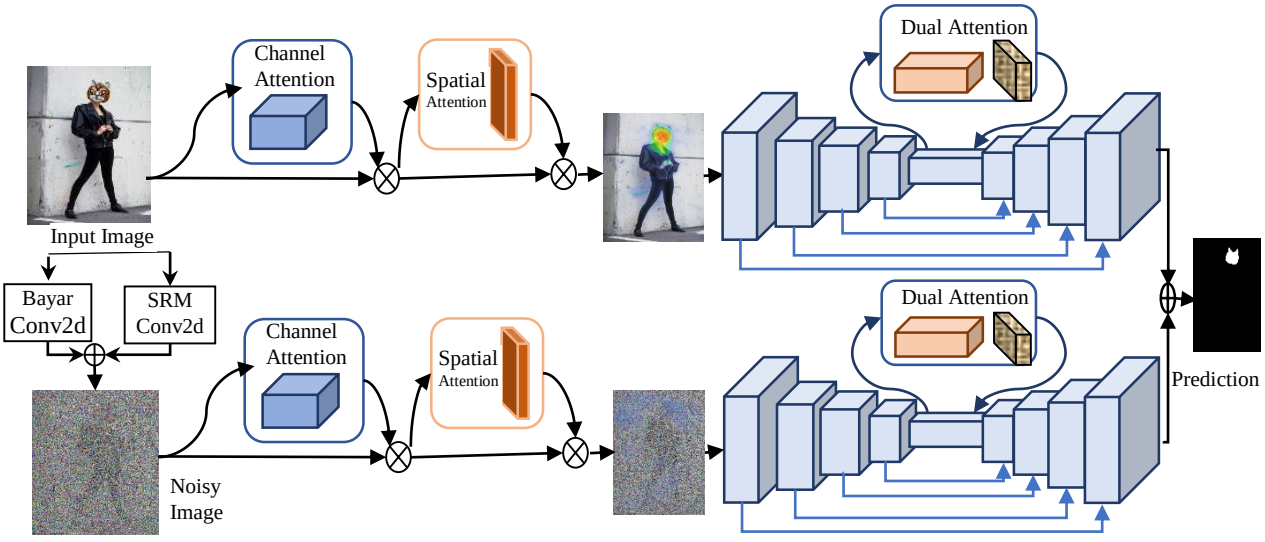


Figure 5.1: Overview of the proposed model consisting of a dual branch consisting of RGB and noise branch followed by ED architecture

Three components are used in the above model: Residual/Noise filters, ED with skip connection, and visual Attention modules.

5.2.2.1 Noise Inconsistencies

An actual image has uniform noise distribution throughout the image. The intuition behind noise residual as a feature is that when an object is removed from one image/section (of an image) and copied/pasted onto another/section of an image, the noise features of the two are less likely to match. In this configuration, the noise residual is the disparity between a pixel's actual value and its estimated value, which is derived via interpolating the values of nearby pixels, which serves as the model for noise. Bayer and SRM filters are the most standard filters that successfully capture the low-level noise residual features.

5.2.2.1.1 Bayar Convolution or Constrained CNN

Using data, constrained CNN can learn the modifications brought up by image manipulation methods into local pixel relationships. Hence, this approach can suppress image-level content and subsequently learn the latent traces of image manipulation [156]. Constrained CNN's primary purpose is to learn prediction error filters, producing feature maps utilized as low-level forensic traces since they offer superior robustness and universality. To force the CNN to learn the low-level traces, the following constraints are enforced on the weights of the kernel of CNN:

$$\begin{cases} \omega_k^{(l)}(0,0) = -1, \\ \sum_{m,n \neq 0} \omega_k^{(l)}(m,n) = 1, \end{cases} \tag{5.1}$$

Abvoe equation represents the constraints enforced on the filter of the kernel. The superscript indicates the CNN layer. l^{th}, the kth convolutional filter within a layer is indicated by the subscript k, and for a CNN filter, the central value is represented by the spatial index (0,0).

5.2.2.1.2 Steganalysis Features

Another method used to extract features from an image's noise residuals is SRM (Spatial Rich Models) filters. Fridrich et al. [157] first introduced the concept of SRM. It was primarily created for steganalysis, extracting latent or hidden features from an image's noisy residuals by applying a set number of high-pass filters. Subsequently, those features are combined and sent to ensemble classifiers. It is a specially designed method that essentially calculates the statistics required to extract specific characteristics from the noise residuals surrounding the neighborhood of pixels in an image. This approach yields a feature that can be considered a local noise descriptor.

5.2.2.2 Encoder-Decoder with Skip connection

The architecture used by most semantic segmentation algorithms today is the ED structure. Our encoder network consists of a Vgg-16 network with 13 convolutional layers (w/o full connected layers), followed by max-pooling layers and is divided into five stages wherein, at each stage, the spatial resolution is halved at every stage, and the channel dimension keeps doubling. This approach allows the model to acquire complex hierarchical representations of visual characteristics, resulting in more reliable and precise predictions. The rationale behind using VGG16(w/o FC layers) as an encoder network is that it has fewer parameters that enable the powerful representation of discriminative visual features. Hence, an encoder module gathers more semantic information while decreasing the feature mappings and increasing channel dimensions. The decoder network enables up-sampling by mapping low-resolution encoded feature maps to full-scale input-resolution feature maps. The proposed architecture uses the transpose convolution layers for the decoder module to enhance the coarse feature mapping of the full-resolution segmentation map. ED employs skip connection to enhance the neural network feature propagation abilities and to prevent gradient vanishing and exploding gradients [158].

5.2.2.3 Attention Mechanism

The attention mechanism uses input-dependent weights to replace the traditional learnable fixed weights, allowing the CNN to learn input-aware relationships that help it emphasize the critical features. Two attention modules are used in the model to encode discriminative features more effectively, which are as follows:

5.2.2.3.1 Channel Attention

A channel attention map is created by utilizing the inter-channel relationship of features. For manipulated localized features, channel attention emphasizes “what" is meaningful for the end task [159]. The module first models the spatial features of various feature maps to generate context descriptors using average and max pooling operations. Subsequently, both descriptors are transmitted to a Multi-layer perceptron (MLP) [159]. Next, the output feature vectors are combined using element-wise summation, and the resulting vector is then normalized using the sigmoid function.

$$\boldsymbol{M_c}(\boldsymbol{F}) = Sigmoid(MLP\big(AvgPool(\boldsymbol{F})\big) + MLP\big(MaxPool(\boldsymbol{F})\big)) \quad (5.2)$$

Where $\boldsymbol{M_c}(\boldsymbol{F})$ Denotes the feature map that captures channel attention.

5.2.2.3.2 Spatial Attention

The inter-spatial interaction among features generates the spatial attention map. It is complementary to channel attention and differs in that it concentrates on "where" manipulation localization is presented. Initially, the average-pooling and max-pooling operations are performed on the channel axis to calculate the spatial attention. Subsequently, these results are combined to provide a suitable feature descriptor [159]. The concatenated feature descriptor is subjected to a convolution layer to generate a spatial attention map delineating regions where desired features are prioritized over undesirable ones.

Below are the equations of the spatial attention map.

$$\boldsymbol{M_c}(\boldsymbol{F}) = Sig(f^{7\times7}\left([AvgPool(\boldsymbol{F})\right);\ MLP(MaxPool(\boldsymbol{F})])) \tag{5.3}$$

5.2.2.4 Dual Attention Mechanism

The Dual Attention network incorporates a self-attention mechanism identifying spatial and channel feature dependencies.

5.2.2.4.1 Position Attention Module

The purpose of the self-attention mechanism in the position attention module is to capture the spatial relationships between any two positions in the feature map [160]. A local feature $\alpha \in \mathcal{R}^{C\times H\times W}$ applied into the layer of convolution, resulting in the generation of two additional feature maps., $\beta \in \mathcal{R}^{C\times H\times W}$ and $\gamma \in \mathcal{R}^{C\times H\times W}$, respectively. Next, reshape them $\mathcal{R}^{C\times N}$ where $N = H \times W$ denotes the number of pixels of a layer. The spatial attention map $\check{S} \in \mathcal{R}^{N\times N}$ which is then computed by performing a matrix multiplication between the transpose of β and γ and applying a softmax layer:

$$s_{ji} = Sigmoid(\beta_i, \gamma_i)\ \ where\ i = 1,2,\dots,N \tag{5.4}$$

where s_{ji} quantifies the impact of i^{th} position is on j^{th} position. Subsequently, passes the feature α through convolution layers to generate feature $\delta \in \mathcal{R}^{C\times H\times W}$e and reshape it to a size of $\mathcal{R}^{C\times N}$. Next, calculate the matrix multiplication of δ and the transpose of $\mathbb{S}$ and reshape the resulting matrix to have dimensions $\mathcal{R}^{C\times H\times W}$. Eventually, we determine the product of the input by a scaling factor ρ and aggregate it with the characteristics α using an element-wise summation to obtain the final result. $\theta \in \mathcal{R}^{C\times H\times W}$Moreover, it is described below:

$$\theta_j = \rho \sum_{i=1}^{N} \left(s_{ji}\delta_i\right) + \alpha_i, \tag{5.5}$$

The outcome characteristic θ is obtained by taking a weighted sum of the initial characteristics and all the features at each point. Consequently, it selectively combines different contexts based on the spatial attention map and provides a global contextual view.

5.2.2.4.2 Channel Attention Module

By considering the connections between channel maps, it is feasible to emphasize the interdependence of feature maps and improve the representation of features in a specific semantic domain. Consequently, the channel attention module directly represents channel interdependencies [160]. As opposed to the position attention module, the channel attention map $\acute{Z} \in \mathcal{R}^{C\times H\times W}$ directly from the original features $\mu \in \mathcal{R}^{C\times H\times W}$. To be more precise, we reshape μ to $\mathcal{R}^{C\times N}$and then multiply μ by its transposition in a matrix. Eventually, a softmax layer is employed to produce the channel attention map $\acute{Z} \in \mathcal{R}^{C\times c}$:

$$z_{ji} = Sigmoid(\mu_i, \mu_j) \ \ where\ i = 1,2, \dots, C \tag{5.6}$$

Where z_{ji} measures the influence of the i^{th} position's influence on j^{th} position. Furthermore, we apply a matrix multiplication using the transpose of $\acute{Z}$ and μ, transforming the outcome into $\mu \in \mathcal{R}^{C\times H\times W}$. The final output, $F \in \mathcal{R}^{C\times H\times W}$, is then obtained by multiplying the result by a scaling parameter ϑ and using an element-wise sum operation with μ.

$$F_j = \vartheta \sum_{i=1}^{C} (x_{ji}\mu_i) + \mu_i \tag{5.7}$$

Above equation demonstrates the final feature of each channel, which represents the long-range semantic linkages between feature maps and is computed as a weighted sum of the features from all channels and the original features. It improves the ability to discriminate between different features [160].

5.2.3 Experiments

This section aims to evaluate the performance of the proposed methodology by validating it on various benchmark datasets and comparing it with various SoTA manipulation localization methods.

5.2.3.1 Datasets

5.2.3.1.1 Shallowfake dataset

A massive data hunger characterizes deep learning network training. There are insufficient images in the usual datasets used for image manipulation detection today to support deep neural network training. Furthermore, a standard dataset's altered images may not be sufficient for

training because they have fewer artifacts. The model undergoes pre-training using CASIAv2, subsequently fine-tuning with additional datasets, and testing is done on them. Table 5.1 contains the details of the split of the training-testing dataset with their kind of manipulation.

Table 5.1 Training and testing split of the various benchmark datasets. S means splicing, C means copy-move, and R means removal.

Datasets	Training Set	Testing Set	Total Samples	Types of manipulation
CASIA V2.0 [161]	5063	-	5063	S, C
CASIA V1.0 [161]	-	920	920	S, C
COLUMBIA [162]	130	50	180	S
COVERAGE [163]	75	25	100	C
NIST16 [164]	414	150	564	S, C, R

5.2.3.1.2 Deepfake Dataset

No deepfake image dataset currently contains a ground truth mask for the manipulated regions. Zhang et al. [165] have built their dataset of Faceforensics++ [32], the only deepfake dataset containing masks for most of its videos. Famous FF++ contains 1000 videos and 5000 fake videos manipulated methods (Deepfakes, Face2Face, Faceshifter, Face-Swap and Neural-Textures). Four manipulations are considered for the frames extracted as 1000 face shifter videos contain no ground truth mask from the videos. Two frames are extracted for each video, and due to some Accessibility concerns prevented us from downloading some legitimate and bogus videos. We have obtained a total of 8,449 genuine frames and 7,330 counterfeit frames.

5.2.3.2 Experimental setup

We use the Vgg-Net16 [166] as a backbone for encoder-decoder architecture, pre-trained on ImageNet [167], to develop our model using the PyTorch framework. The model is executed using two NVIDIA RTX A5000 GPUs, and the image is scaled to 256×256. The model is optimized using the Adam optimizer with a batch size 32 during the training and testing phase. The starting learning rate is 1e-4, which decays every 10^{th} step with a decay rate of 0.8.

5.2.4 Quantitative Analysis

We have quantitatively assessed the performance of the shallowfake and deepfake datasets.

5.2.4.1 Shallowfake dataset

Following the approach [168], the model was trained using the CASIA2 dataset and fine-tuned with standard shallowfakes datasets like Nist16, Coverage, Columbia, and CASIA1. The AUC and F_1 scores of these datasets are recorded in Table 5.2. For comparison, two categories of models are considered, i.e. unsupervised and DNN models. The table shows that the model outperforms the unsupervised models by a significant margin. The performance of handcrafted

features intended for the specific type of manipulation is severely constrained, and all of these traditional methods capture certain tampering artifacts with limited information for detection. Our method's scores are comparable and outperform the other DNN-based methods at various datasets. For the CASIA and Columbia, our model performs significantly well with an AUC score even reaching 96.72%, and on the Nist16 dataset, it has a comparable score to other models (AUC~99.9%).

In comparison, the model performs poorly on the COVERAGE dataset, with an AUC score reaching 77.38% owing to fewer images and the dataset containing copied/moved objects of similar appearance. Our technique captures the broader range of features like RGB features, noise inconsistencies, and global context rather than the neighboring pixels, which helps gather more information for manipulation classification. Various complex CNN models cannot perform well; this could be attributed to most DNN-based methods that model the network by superimposing multiple CNN networks or adding complex branches. For instance, TDA-Net combines three CNN streams, such as end-to-end training for a complex network, which makes it harder for the network to train and requires more computing power. Also, few models are so much simpler that they focus only on the semantic features and hence fail to locate the tampered segments accurately. On the other hand, our model is relatively less complex, easily captures non-semantic features and does not require significant training data to achieve comparable performance.

Table 5.2 Evaluation experiments results of various models on the shallowfake dataset. "-" means unknown score.

Category	Method	**NIST16**		**COLUMBIA**		**COVERAGE**		**CASIA**	
		AUC	**F1**	**AUC**	**F1**	**AUC**	**F1**	**AUC**	**F1**
Unsupervised	ELA [169]	0.429	0.236	0.581	0.470	0.583	0.222	0.613	0.214
	NOI1 [170]	0.487	0.285	0.546	0.574	0.587	0.269	0.612	0.263
	CFA1 [171]	0.501	0.174	0.720	0.467	0.485	0.190	0.522	0.207
DNN-based model	MFCN [172]	-	0.571	-	0.612	-	-	-	0.541
	RGB-N [173]	0.937	0.722	0.858	0.697	0.817	0.474	0.795	0.582
	J-LSTM [174]	0.764	-	-		0.712	-	-	-
	LSTM-EnDec [175]	0.794	-		-	0.712	-	-	-
	CR-CNN [176]	0.992	0.927	0.861	0.790	0.939	0.757	0.789	0.475
	ManTra-Net [177]	0.795	-	0.824	-	0.819	-	0.817	-
	TDA-Net [168]	0.948	0.756	0.892	0.735	0.864	0.474	0.831	0.582
	GSR-Net [178]	0.945	0.736			0.768	0.489	0.796	0.574
	SPAN [179]	0.961	0.582	0.936	0.815	0.937	0.558	0.838	0.382
	PSCC-Net [180]	0.996	0.819	-	-	0.941	0.723	0.875	0.554
	MVSS-Net++ [181]	0.976	0.854	-	-	0.897	0.753	0.844	0.546
	ObjectFormer [182]	0.996	0.824	-	-	0.957	0.7580	0.882	0.579

	TA-Net [183]	**0.997**	0.865	-	-	**0.978**	**0.782**	0.893	0.614
	Transforensics [184]	-	-	-	-	0.884	0.674	0.850	0.627
	Our Model	0.9919	**0.9910**	**0.9672**	**0.9556**	0.7738	0.4362	**0.9494**	**0.8765**

5.2.4.2 Deepfake dataset

Ten models are considered for evaluation on the deepfake dataset. Six are the SoTA image manipulation models; the others are the standard image-segmentation models. Codes of the models that are available on GitHub are considered for comparison. Pre-trained Pytorch models are used for image segmentation models, which are further fine-tuned. Three evaluation metrics are used for a more comprehensive evaluation. Table 5.3 shows the experimental results of the deepfake on the various models. All the models performed decently except MantraNet [177] on the various categories of the Faceforensics++ dataset. It could be attributed to the fact that most of the manipulations are of the face, and there is a single entity that occupies the entire frame, and all the models are powerful enough to capture such apparent artifacts. MantraNet performs poorly on all categories of the deepfake, possibly because images/frames are not high-resolution and contain blurriness and noise, which the model cannot process and learn the latent discriminative features. NedB-Net [185] is another model that has performed well but has a relatively lower score than other SoTA models, which may be due to the low-quality images and the more significant manipulated regions; the paper's authors also highlight this problem. DL-Net [186] performs reasonably well in the FF++ dataset, with the F1 score reaching 96%, owing to its capability to capture high and low-level cues using noise level segmentation map prediction, which constraints the model to focus on the manipulated regions. However, the model has a low score for Face-swap manipulation among the four categories of FF++ manipulation. Another method [187] used for deepfake localization uses a weak supervision framework and uses three methods, i.e. GradCAM, Patches and Attention, for results illustrations. We have used GradCAM methods for score comparison. The method performs outstanding well in the weak supervision setting, showing the powerful discriminative capabilities of the model. However, like the earlier model, the model has a drop in performance for the FS category of manipulation, which could be the network's inability to make accurate predictions at the edges. DADF method [188] performs better than most models, which use multi-scale adapters to capture short and long-range forgeries and guided attention mechanisms, enhancing rich forgery clues. Their scores are at par with other methods and a proper State-of-the-art method for comparison. Another method [165] used also performed well, with the F1 score reaching 98%. Their method is built on top of existing UperNet and

uses Bayar convolution methods to trace noise clues. Despite being these State-of-the-art models, all models score more than 90%, which could be attributed to the fact that most of the manipulation has been done on the face, which is easy for the models to recognize. Our model has performed considerably well in the various categories of the FF++ dataset and outperformed the scores of other standard models. These evaluations showed that different modules designed well to work in tandem resulted in the robust learning of the discriminative for varied manipulation.

Table 5.3 Evaluation experiments results of various models on categories of F++ dataset. IMD means Image Manipulation Detection models, and IS means Image Segmentation Models.

Types	Methods	DeepFakes			Face2Face			FaceSwap			Neural Texture		
		IoU	AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU	AUC	F1
IMD models	**MVSS-Net** [189]	0.9558	0.9996	0.9787	0.9788	0.9997	0.9893	0.9570	0.9989	0.9780	0.9382	0.998	0.968
	Mantra Net [177]	0.3469	0.9523	0.5151	0.3553	0.9159	0.5243	0.3394	0.8888	0.5068	0.3664	0.9706	0.5363
	NedB_Net [185]	0.8811	0.9767	0.9368	0.8432	0.9662	0.9149	0.8522	0.9700	0.9221	0.8827	0.9783	0.9377
	DL-Net [186]	0.8750	0.9952	0.9337	0.9108	0.9946	0.9533	0.8976	0.9976	0.9460	0.9262	0.9979	0.9617
	Weakly_Super_Gradcam [187]	0.9787	0.9990	0.9897	0.8753	0.9795	0.9336	0.9575	0.9791	0.9231	0.9861	0.9990	0.9868
	DADF [188]	0.9453	0.9939	0.9677	0.9621	0.9899	0.9786	0.9599	0.9896	0.9655	0.9236	0.9788	0.9586
	Shallow Deepfake_local [165]	0.9617	0.999	0.9713	0.9801	0.9898	0.9799	0.9485	0.9856	0.9666	0.9365	0.9989	0.9689
IS Models	**DeepLab** [190]	0.9428	0.9981	0.9706	**0.9840**	**0.9999**	**0.9919**	0.9769	0.9986	0.9883	0.9704	0.9998	0.9640
	FCN [191]	0.9701	0.9967	0.9848	0.9834	0.9988	0.9786	0.9470	0.9991	0.9728	0.9591	0.9984	0.9728
	LRASPP [192]	0.9114	0.9992	0.9536	0.9445	0.9998	0.9714	0.9106	0.9995	0.9532	0.9399	0.9997	0.9690
Our Model		**0.9736**	**0.9940**	**0.9866**	0.9820	0.9964	0.9909	**0.9810**	**0.9983**	**0.9904**	**0.9710**	**0.9913**	**0.9853**

5.2.5 Qualitative Analysis

This section compares our method with the two most competitive methods, i.e. MantraNet and MVSSS, to give some specific qualitative outcomes on both shallowfakes and deepfake datasets. Figure 5.2 shows the visualization results of image manipulation detection. Our method achieves better localization results than other methods, as the other methods generate

many false positives. In the case of the shallowfake dataset, the method has superior localization performance for CASIA, Columbia and Nist16 datasets. For the deepfake dataset, localization seems much easier, as most image manipulation is done on the face, making it easy for localization. MantraNet exhibits a significant departure from the ground truth, whereas MVSSNET exhibits a significant rate of false positives in regions that have not been altered. The primary cause of this outcome is that during the training phase, MVSSNET used a lot of natural images, which could have a clear negative impact on network training. Additionally, MVSSNET dramatically increases false positives due to frequently responding to altered and non-manipulated portions of the image. On the contrary, our method focuses on low-level features like noise and high-level contextual features, leading to better localization of manipulated artifacts.

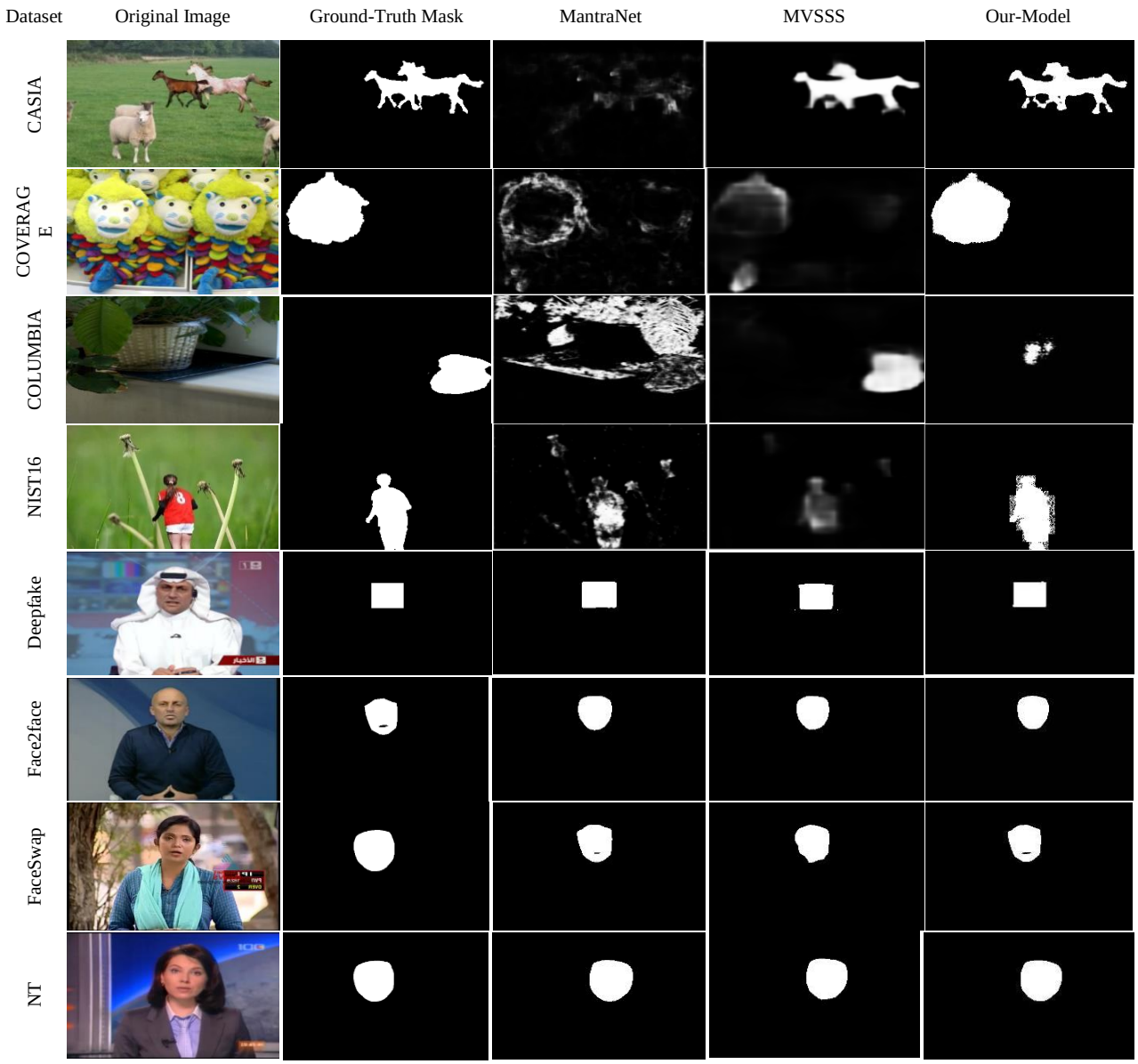


Figure 5.2 Qualitative Visualization results of the manipulation localization for shallow fakes and deepfake dataset images for different methods

5.2.6 Ablation Studies

We assess the proposed network in various settings with the components introduced one at a time to study the impact of each component. All the components were trained on the CASIA2 dataset and then evaluated on the different shallowfakes datasets, i.e. NIST16, Columbia, Coverage and CASIA1. Table 5.4 shows the results of the ablations experiment. The different experimental settings are discussed below:

Case A: Model without Channel Attention: Channel attention modules have been removed from both branches to study their impact. AUC and F_1 scores decrease slightly, showing inter-relation along the channel axis containing potential features that assisted in the detection task. Two things worth noting: for the Columbia and Nist16 dataset, scores decreased very slightly, nearly ~2%(AUC score), while for the Coverage dataset, it decreased significantly by nearly ~9%, showing that where data samples are less, attention mechanism plays a pivotal role.

Case B: Model without Spatial Attention: From scores, it perceives that spatial attention contributes less in comparison to channel attention to the detection features. It could be attributed to spatial attention focusing on high-level or semantic features, which is less critical for manipulation. Also, one of the trivial observations is that scores declined slightly and uniformly across all the datasets.

Case C: Model without Channel and Spatial Attention: In this case, channel and spatial have been removed from both branches to study their relevance. Scores have decreased significantly, nearly ~14%, across all the datasets except the Coverage dataset. It shows that these modules work better together to capture informative features and global correlations along every axis and emphasize such critical features.

Case D: Model without Dual Attention: Dual attention modules are axed from the Encoder-decoder architecture to investigate their importance. This dual attention contributes less to the overall detection task of either the spatial or channel attention module. Also, the dual attention works almost uniformly for every dataset, showing its ability to capture long-range contextual information for varied manipulations.

Case E: Model without Noise branch: Here, the noise branch, consisting of the RGB branch, is completely removed from the model. This means that RGB high-level or semantic features are primarily used for detection. Here, the performance decreased drastically by nearly ~8%, confirming that low-level features like noise consistencies are crucial indicators in the overall detection task. For the CASIA model, scores have decreased to a greater extent, around ~19%,

as it varied manipulation like splicing and copy-move, showing the relevance of residual noise feature to varied manipulation.*Case F: Model without RGB branch:* The RGB branch is completely removed from the overall model, and the noise branch is used for manipulation detection. In this scenario, the AUC score is decreased to a smaller extent, around 4%, again confirming that the low-level noise features have more relevance than high-level semantic features for the detection task. For the Coverage dataset, the performance has decreased drastically, meaning that noise branches need ample datasets to learn noise inconsistencies for detection tasks.

Table 5.4: Ablation experiment of different components where the model trained on CASIA2 and tested on other datasets.

Cases	NIST16		COLUMBIA		COVERAGE		CASIA	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1
Case A	0.9246	0.9688	0.9516	0.8758	0.6891	0.3623	0.8868	0.8078
Case B	0.9327	0.9896	0.9626	0.9285	0.7208	0.3953	0.9123	0.852
Case C	0.8055	0.8302	0.8328	0.7497	0.7373	0.3946	0.8033	0.4254
Case D	0.9087	0.9291	0.93	0.8398	0.7307	0.4049	0.9210	0.8560
Case E	0.8688	0.8015	0.8873	0.7333	0.7261	0.3871	0.7543	0.4852
Case F	0.9239	0.9488	0.9435	0.8569	0.6052	0.2879	0.8863	0.8562
Overall Model	**0.9441**	**0.9952**	**0.9641**	**0.8988**	**0.7542**	**0.4272**	**0.9313**	**0.8650**

5.2.7 Computational complexity Analysis

In this section, we assessed the complexity of three distinct deep networks: MantraNet, SPAN, MVSSNET, Nedb_Net [185], DeepLabv3_ResNet50 [190], FCN_ResNet50 [193] and LRASPP_MobileNet [192]. All experimental analysis is done on two NVIDIA RTX A5000 GPUs. Table 5.5 shows the computational analysis of different models compared to ours. Regarding the number of parameters, the model is relatively lighter, with 38.86 million parameters compared to MVSS-Net and Deeplab. MantraNet and LRASPP are many relatively lighter models than other models. Another metric was calculated in CPU and GPU time for one epoch for a batch size of 32 images of the CASIA dataset. The model runs on the GPU and CPU to measure their time in seconds. GPU time is always less time for different models. Our model takes less time both on GPU and CPU.

Table 5.5: Computation complexity analysis of different models. GPU and CPU time are measured for an epoch of batch size of 32.

Methods	**Parameters (Millions)**	**CPU time (sec)**	**GPU time (sec)**
MantraNet	3.80	8.06	0.14
MVSS-Net	142.782	9.66	1.13
SPAN	4.06	**8.08**	**0.278**
Nedb-Net	45.085	10.11	1.171

DeepLabv3_ResNet50	39.63	9.97	1.17
FCN_ResNet50	32.946	10.20	1.14
LRASPP_MobileNet	**3.21**	9.01	1.01
Our_Model	38.86	9.09	0.94

5.2.8 Conclusion

This paper proposes a novel dual-branch architecture consisting of Noise Residual extraction modules at one branch and RGB information at the other branch powered by the attention mechanism before feeding to the ED architecture for precise IML prediction. The model effectively captures the low-level inconsistencies critical for IML tasks and additional semantic features. Extensive experiments on the shallowfakes and deepfake datasets have shown that the model is able to capture subtle traces of manipulation and achieves SoTA results. Future work may involve checking the model's generalizability on the unseen dataset and robust evaluation of various compression scenarios.

5.3 Significance outcome of this chapter

The significance outcome of this chapter are as follows:

- Developed a unique dual-branch architecture that employs RGB information on one branch and noise features on the other. In order to more accurately represent artifacts, both branches implemented channel and spatial attention, which was subsequently followed by ED architecture. A dual attention module is employed to learn semantic interdependencies in the spatial and channel domains for down-sampled features.
- The discriminative capabilities of the model are demonstrated by the deepfake dataset FF++, which outperforms other models, and the shallowfakes dataset, which includes CASIA, Columbia, Coverage, and NIST16.
- Ablation studies are conducted to investigate the significance of various components within the overall model.
- The complexity computation analysis was conducted to demonstrate that the model with 38 million parameters is comparatively less complex than the various SoTA models.

This chapter is based on the following research works:

1. **D. Dagar** and D. K. Vishwakarma, “Shallowfake and Deepfake Image Manipulation Localization using Noise and RGB-based dual branch method” ***Signal, Image and Video Processing***, vol 18, pages 7065-7077, 2024, doi: https://doi.org/10.1007/s11760-024-03376-x

Chapter 6: Conclusion and Future Scope

6.1 Conclusion

This chapter concludes the research conducted in this thesis. Overall, four innovative architectures based on deep learning are proposed for the detection of deepfake manipulation in multimedia content. The first two models are specifically designed to address the issue of Deepfake detection in images. A Diverse manipulation video dataset and along with a framework for deepfake video detection is also proposed. Finally, the method to localize the deepfake manipulation is proposed. The details of the proposed approaches are as follows:

- A novel deepfake detection model i.e Tex-ViT which uses gram-matrices as texture feature descriptor and cross-attention mechanism of vision transformer. The model combines traditional ResNet features with a texture module that operates in parallel on sections of ResNet before each down-sampling operation. This module then serves as an input to the dual branch of the cross-attention vision transformer. Experimentation done on the various categories of FF++ and GAN dataset images in cross-domain settings to demonstrate the model's generalizability. Experiments were conducted on the Celeb-DF, FF++, and DFDCPreview datasets using various post-processing techniques such as blurring, noise addition, and compression. The results demonstrated the resilience of the models in varied settings.
- Another method for deepfake detection named Tex-Net which uses the combination of Gram matrices and Local Binary patterns as a texture features representation and rest of the architecture is same as of Tex-ViT. The global texture is computed during each down sampling operation of ResNet, and subsequently, layer characteristics are consolidated at many layers. These characteristics persistently merge prior to being inputted into the dual-branch cross-attention-based vision transformer for classification. The model's generalization capacity was demonstrated by conducting experimentation on several categories of FF++ and GAN dataset images in a cross-manipulation setting. Experimentation also done on the data samples from FF++, DFDCPreview, and Celeb-Df that were subjected to various post-processing techniques, including blurring, noise addition, and compression which demonstrated the model's robustness.
- Presented a Div-DF dataset comprising diverse forms of video modification such as face swapping, facial reenactment, and lip-syncing. The dataset consists of 150 authentic videos featuring various celebrities from different fields, together with 250 deepfake videos. The deepfake videos include 100 face-swap videos, 100 facial reenactment videos, and 50 lip-

sync videos. Deepfake films are created by utilizing advanced techniques such as the Face-Swap GAN (FSGAN) and the Wav2Lip approach.

- A sophisticated deepfake video detection model is proposed which combines the pretrained Xception and LSTM models. Xception employs depthwise separable convolution to capture the latent spatial artefacts, whereas LSTM captures the discrepancies among the modified sequences. The hybrid model assembly enables the acquisition of knowledge about spatial and temporal distortions across many dimensions, making it a powerful tool for identifying deepfakes. Evaluation of the efficacy of the proposed model and various state-of-the-art models on our Div-Df which shows the superiority of the proposed model.
- A novel model for deepfake manipulation localization is proposed. The model uses a dual-branch model that integrates handmade feature noise with Convolutional Neural Networks (CNNs) as an Encoder-decoder (ED) system, enhanced by the attention mechanism. This model utilizes a dual-branch approach, where one branch incorporates noise characteristics and the other branch incorporates RGB features. These features are then fed into an ED architecture for semantic learning. Additionally, skip connections are included to preserve spatial information. Extensive research was conducted on the shallowfakes dataset, which includes CASIA, COVERAGE, COLUMBIA, and NIST16, as well as the deepfake dataset Faceforensics++ (FF++). The evaluation results proves the exceptional feature extraction capabilities of the model.

6.2 Future Scope

Extensive research has been carried out in recent years to identify deepfake manipulation in multimedia content. Although the performance in detecting or localizing these manipulations has consistently improved, there are still several promising research directions that need to be addressed.

- **Generalization to unknown dataset:** Existing deepfake detection methods perform well on the seen dataset, but their performance degrades on the unseen dataset. Lots of methods have worked on the generalization but the performance is not satisfactory which makes them unfit for their deployment in real-world scenarios. Moreover, its absence gives the upper hand to the anti-social elements to misuse the technology at their whims and fancies. That’s why generalization is one of the most crucial indicators of the performance of the methods and future work would definitely needs to raise the bar of performance of the generalization of detection methods.

- **Lack of Interpretability of detection methods:** The main issue with detection approaches is the lack of interpretability. These methods typically rely on neural networks, which suffer from a fundamental problem of being black-box in nature, making it difficult to explain their results. In real-world situations, it is necessary to provide an explanation that can be easily understood by humans for any forensic methodologies used. For example, let's say a deepfake detection technology is used in the trial to identify evidence. If that is the situation, it may be necessary to provide a rationale or clarification for different portions of it being a deepfake. Hence, detection methods should prioritize the comprehensibility of the detection outcomes, which unquestionably continue to be a matter of concern for future consideration.
- **Robustness of various Adversarial perturbation:** Recent research suggest that deep-learning models, although they have improved in detecting manipulation, are highly susceptible to adversarial attacks. Injecting noise into the input pixel values can significantly alter the predictions made by a trained model. Enhancing the resilience of deep-learning models against adversarial attacks is an imperative area for future research.
- **Deployment in real world scenarios:** Existing methods tend to perform well in the controlled environment, where we have a dataset that hardly represents real-world scenarios. For real-world data, which contains various noises and manipulation, their performance degrades as the detection methods are designed to identify specific types of artifacts. Further research should be focused on addressing deployment challenges for end-users through the development of applications or web-based frameworks. The growing capability of modern hardware enables the utilization of complex computational models on mobile devices.
- **Scarcity of a large and quality dataset**: A dataset is crucial for detection algorithms because it enables the learning of distinct sets of features needed to recognize different artifacts [10]. Other proposed datasets encounter issues that adversely affect the performance of detection algorithms. Some of the prominent concerns are as follows:
    1) The dataset is small in size and does not adequately reflect different types of modification.
    2) Facial features exhibit inconsistency and blurriness.
    3) Video frames exhibiting flickering or jitteriness.
    4) Inconsistent lighting in pictures.

5) Insufficient availability of a diverse audio sample with background noise to accurately depict real-world scenarios.
6) Absence of obstructing objects in the images.
7) Images or video frames of poor quality.

Future work needs to focus on the creation of quality dataset which addresses above issues.

Chapter 7: References

[1] B. Dean, “Social Network Usage & Growth Statistics: How Many People Use Social Media in 2021?,” BackLINKO, 01 09 2021. [Online]. Available: https://backlinko.com/social-media-users#global-social-media-growth-rates.

[2] “Most popular social networks worldwide as of July 2021,” Statistics Research Department, July 2021. [Online]. Available: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/.

[3] D. Güera and E. J. Delp, “Deepfake Video Detection Using Recurrent Neural Networks,” in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS),* Auckland, 2018.

[4] E. STRICKLAND, “Facebook AI Launches Its Deepfake Detection Challenge,” IEEE, December 2019. [Online]. Available: https://spectrum.ieee.org/facebook-ai-launches-its-deepfake-detection-challenge.

[5] Y. Mirsky and W. Lee, “The Creation and Detection of Deepfakes: A Survey,” *ACM Computing Surveys,* vol. 54, no. 1, 2021.

[6] R. Chesney and D. K. Citron, “Deep Fakes: A Looming Challenge for Privacy, Democracy, and National Security,” p. 68, 2018.

[7] A. Jaiman, “Positive uses of Deepfakes,” towards data science, 15 Aug 2020. [Online]. Available: https://towardsdatascience.com/positive-use-cases-of-deepfakes-49f510056387. [Accessed 11 April 2021].

[8] A. Jaiman, “Deepfakes Harms and Threat Modeling,” 19 Aug 2020. [Online]. Available: https://towardsdatascience.com/deepfakes-harms-and-threat-modeling-c09cbe0b7883. [Accessed 14 April 2021].

[9] “Faceswap,” [Online]. Available: https://faceswap.dev/. [Accessed 6 April 2021].

[10] “FakeApp,” [Online]. Available: https://www.fakeapp.com/. [Accessed 6 April 2021].

[11] T. Karras, T. Aila, S. Laine and J. Lehtinen, “Progressive Growing of GANs for Improved Quality, Stability, and Variation,” in *International Conference on Learning Representations(ICLR),* Vancouver, 2018.

[12] T. Karras, S. Laine and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* Long Beach, 2019.

[13] Y. Nirkin, L. Wolf, Y. Keller and T. Hassner, “DeepFake Detection Based on Discrepancies Between Faces and their Context,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2021.

[14] “deepfakes/Faceswap,” github, 2016. [Online]. Available: https://github.com/deepfakes/faceswap.

[15] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies and M. Niessner, “FaceForensics++: Learning to Detect Manipulated Facial Images,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, 2019.

[16] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik and H. Pfister, “Video Face Replacement,” *ACM Transactions on Graphics*, vol. 30, no. 6, pp. 1-10, 2011.

[17] L. Li, J. Bao, H. Yang, D. Chen and F. Wen, “Advancing High Fidelity Identity Swapping for Forgery Detection,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, 2020.

[18] C. Chan, S. Ginosar, T. Zhou and A. Efros, “Everybody Dance Now,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, 2019.

[19] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt and M. Nießner, “Face2Face: Real-time Face Capture and Reenactment of RGB Videos,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 2016.

[20] J. Thies, M. Zollhöfer and M. Nießner, “Deferred neural rendering: image synthesis using neural textures,” *ACM Transactions on Graphics*, vol. 38, no. 4, p. 66, 2019.

[21] L. Liu, W. Xu, M. Zollhöfer, H. Kim, F. Bernard, M. Habermann, W. Wang and C. Theobalt, “Neural Rendering and Reenactment of Human Actor Videos,” *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 1-14, 2019.

[22] M. Christos Doukas, M. R. Koujan, V. Sharmanska, A. Roussos and . S. Zafeiriou, “Head2Head++: Deep Facial Attributes Re-Targeting,” *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 1, pp. 31-43, 2021.

[23] E. Zakharov, A. Shysheya, E. Burkov and V. Lempitsky, “Few-Shot Adversarial Learning of Realistic Neural Talking Head Models,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, 2019.

[24] T.-C. Wang, . M.-Y. Liu, A. Tao, G. Liu, J. Kautz and B. Catanzaro, “Few-shot Video-to-Video Synthesis,” in *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, 2019.

[25] O. Gafni, O. Ashual and L. Wolf, “Single-Shot Freestyle Dance Reenactment,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, 2021.

[26] J. Zhang, X. Zeng, . Y. Pan, Y. Liu, Y. Ding and C. Fan, “FaceSwapNet: Landmark Guided Many-to-Many Face Reenactment,” in *arXiv:1905.11805v1*, 2019.

[27] Y. Zhang, S. Zhang, Y. He, C. Li, C. C. Loy and Z. Liu, “One-shot Face Reenactment,” in *arXiv:1908.03251v1*, 2019.

[28] K. Gu, Y. Zhou and T. Huang, “FLNet: Landmark Driven Fetching and Learning Network for Faithful Talking Facial Animation Synthesis,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, Hilton New York Midtown, 2020.

[29] J. Lee, D. Ramanan and R. Girdhar, “MetaPix: Few-shot video retargeting,” in *International Conference on Learning Representations*, 2020.

[30] E. Sanchez and M. Valstar, “A recurrent cycle consistency loss for progressive face-to-face synthesis,” in *IEEE International Conference on Automatic Face and Gesture Recognition*, Buenos Aires, 2020.

[31] S. Tripathy, J. Kannala and E. Rahtu, “FACEGAN: Facial Attribute Controllable rEenactment GAN,” in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, 2021.

[32] S. SUWAJANAKORN, S. M. SEITZ and I. KEMELMACHER-SHLIZERMAN, “Synthesizing Obama: Learning Lip Sync from Audio,” *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1-14, 2017.

[33] O. Fried, A. Tewari, M. Zollhöfer, A. Finkelstein, E. Shechtman, D. B. Goldman, K. Genova, Z. Jin, C. Theobalt and M. Agrawala, “Text-based Editing of Talking-head Video,” *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1-14, 2019.

[34] A. Lahiri, V. Kwatra, C. Frueh, J. Lewis and C. Bregler, “LipSync3D: Data-Efficient Learning of Personalized 3D Talking Faces from Video using Pose and Lighting Normalization,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, 2021.

[35] Z. Zhang, L. Li, Y. Ding and C. Fan, “Flow-guided One-shot Talking Face Generation with a High-resolution Audio-visual Dataset,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, 2021.

[36] A. Jamaludin, J. S. Chung and . A. Zisserman, “You said that?: Synthesising talking faces from audio,” *International Journal of Computer Vision*, vol. 127, pp. 1767-1779, 2019.

[37] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim and J. Choo, “StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 2018.

[38] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu and F. Moreno-Noguer, “GANimation: One-Shot Anatomically Consistent Facial Animation,” *International Journal of Computer Vision*, vol. 128, pp. 698-713, 2019.

[39] M. Liu, Y. Ding, M. Xia, X. Liu, E. Ding, W. Zuo and S. Wen, “STGAN: A Unified Selective Transfer Network for Arbitrary Image Attribute Editing,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, 2019.

[40] H. Liang, X. Hou and L. Shen, “SSFlow: Style-guided Neural Spline Flows for Face Image Manipulation,” in *Proceedings of the 29th ACM International Conference on Multimedia*, New York, 2021.

[41] R. Wang, J. Chen, G. Yu, . L. Sun, C. Yu, C. Gao and N. Sang, “Attribute-specific Control Units in StyleGAN for Fine-grained Image Manipulation,” in *Proceedings of the 29th ACM International Conference on Multimedia*, New York, 2021.

[42] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen and T. Aila, “Analyzing and Improving the Image Quality of StyleGAN,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* Seattle, 2020.

[43] Y. Shen, J. Gu, X. Tang and B. Zhou, “Interpreting the Latent Space of GANs for Semantic Face Editing,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* Seattle, 2020.

[44] R. Chen, X. Chen, B. Ni and Y. Ge, “SimSwap: An Efficient Framework For High Fidelity Face Swapping,” in *Proceedings of the 28th ACM International Conference on Multimedia*, Seattle, 2020.

[45] M. Masood, M. Nawaz, . K. M. Malik, A. Javed and . A. Irtaza, “Deepfakes Generation and Detection: State-of-the-art, open challenges, countermeasures, and way forward,” in *arXiv:2103.00484v1*, 2021.

[46] F. Juefei-Xu, R. Wang, Y. Huang, Q. Guo, L. Ma and Y. Liu, “Countering Malicious DeepFakes: Survey, Battleground, and Horizon,” in *arXiv:2103.00218v1,* 2021.

[47] H. Li, B. Li, S. Tana and J. Huang, “Identification of deep network generated images using disparities in color components,” *Signal Processing*, vol. 174, 2020.

[48] P. Chen, J. Liu, T. Liang, C. Yu, S. Zou, . J. Dai and J. Han, “DLFMNet: End-to-End Detection and Localization of Face Manipulation Using Multi-Domain Features,” in *IEEE International Conference on Multimedia and Expo (ICME),* Shenzhen, 2021.

[49] S. McCloskey and . M. Albright, “Detecting GAN-generated Imagery using Color Cues,” in *arXiv:1812.08247v1*, 2018.

[50] N. Yu, L. Davis and M. Fritz, “Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, 2019.

[51] M. Koopman, A. M. Rodriguez and Z. Geradts, “Detection of Deepfake Video Manipulation,” in *Irish Machine Vision and Image Processing conference(IMVIP)*, Belfast, 2018.

[52] Y. Li and S. Lyu, “Exposing DeepFake Videos By Detecting Face Warping Artifacts,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops,*, Long Beach, 2019.

[53] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen and B. Guo, “Face X-ray for More General Face Forgery Detection,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* Seattle, 2020.

[54] F. Matern, C. Riess and M. Stamminger, “Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations,” in *IEEE Winter Applications of Computer Vision Workshops (WACVW)*, Waikoloa, 2019.

[55] Y. Zhao, W. Ge, W. Li, R. Wang, L. Zhao and J. Ming, "Capturing the Persistence of Facial Expression Features for Deepfake Video Detection," in *International Conference on Information and Communications Security,* Beijing, 2019.

[56] X. Li, K. Yu, S. Ji, Y. Wang, C. Wu and H. Xue, "Fighting Against Deepfake: Patch&Pair Convolutional Neural Networks (PPCNN)," in *Companion Proceedings of the Web Conference 2020,* New York, 2020.

[57] S. Lee, S. Tariq, Y. Shin and S. S. Woo, "Detecting handcrafted facial image manipulations and GAN-generated facial images using Shallow-FakeFaceNet," *Applied Soft Computing,* vol. 105, 2021.

[58] Z. Shang, H. Xie, Z. Zha, L. Yu, Y. Li and Y. Zhang, "PRRNet: Pixel-Region relation network for face forgery detection," *Pattern Recognition,* vol. 116, 2021.

[59] S. Agarwal, H. Farid, O. Fried and M. Agrawala, "Detecting Deep-Fake Videos from Phoneme-Viseme Mismatches," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* Seattle, 2020.

[60] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera and D. Manocha, "Emotions Don't Lie: An Audio-Visual Deepfake Detection Method using Affective Cues," in *ACM International Conference on Multimedia*, New York, 2020.

[61] K. Chugh, P. Gupta, A. Dhall and R. Subramanian, "Not made for each other- Audio-Visual Dissonance-based Deepfake Detection and Localization," in *ACM International Conference on Multimedia,* New York, 2020.

[62] P. Yu, Z. Xia, J. Fei and Y. Lu, "A Survey on Deepfake Video Detection," *IET Biometrics,* 2021.

[63] B. C. Hosier and M. C. Stamm, "Detecting Video Speed Manipulation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* Seattle, 2020.

[64] I. Amerini, L. Galteri, R. Caldelli and A. D. Bimbo, "Deepfake Video Detection through Optical Flow based CNN," in *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW),* Seoul, 2019.

[65] R. Caldelli, L. Galteri, I. Amerini and A. D. Bimbo, "Optical Flow based CNN for detection of unlearnt deepfake manipulations," *Pattern Recognition Letters,* vol. 146, 2021.

[66] X. Yang, Y. Li and S. Lyu, "Exposing Deep Fakes Using Inconsistent Head Poses," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Brighton, 2019.

[67] Y. Li, M.-C. Chang and S. Lyu, "In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking," in *IEEE International Workshop on Information Forensics and Security (WIFS),* Hong Kong, 2018.

[68] H. Qi, . Q. Guo, . F. Juefei-Xu, X. Xie2, L. Ma, W. Feng, Y. Liu and J. Zhao, "DeepRhythm: Exposing DeepFakes with Attentional Visual Heartbeat Rhythms," in *ACM International Conference on Multimedia*, New York, 2020.

[69] U. A. Ciftci, I. Demir and L. Yin, "FakeCatcher: Detection of Synthetic Portrait Videos using Biological Signals," *IEEE Transactions on Pattern Analysis and Machine Intelligence (Early Access),* 2020.

[70] J. Hernandez-Ortega, R. Tolosana, J. Fierrez and A. Morales, "DeepFakesON-Phys: DeepFakes Detection based on Heart Rate Estimation," in *arXiv:2010.00400v3,* 2020.

[71] R. Yasrab, W. Jiang and . A. Riaz, "Fighting Deepfakes Using Body Language Analysis," *Forecasting, MDPI, Open Access Journal,* vol. 3, no. 2, pp. 1-19, 2021.

[72] H. Khalid and S. S. Woo, "OC-FakeDect: Classifying Deepfakes Using One-class Variational Autoencoder," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops,* Seattle, 2020.

[73] X. Xuan, B. Peng, W. Wang and J. Dong, "On the Generalization of GAN Image Forensics," in *Chinese Conference on Biometric Recognition,* Zhuzhou, 2019.

[74] P. Zhou, X. Han, V. I. Morariu and L. S. Davis, "Two-Stream Neural Networks for Tampered Face Detection," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* Honululu, 2017.

[75] H. Jeon, Y. Bang and S. S. Woo, "FakeTalkerDetect: Effective and Practical Realistic Neural Talking Head Detection with a Highly Unbalanced Dataset," in *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW),* Seoul, 2019.

[76] X. Wu, Z. Xie, Y. Gao and Y. Xiao, "SSTNet: Detecting Manipulated Faces Through Spatial, Steganalysis and Temporal Features," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Barcelona, 2020.

[77] S. Tariq, S. Lee, H. Kim, Y. Shin and S. S. Woo, "GAN is a Friend or Foe? A Framework to Detect Various Fake Face Images," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing,* Cyprus, 2019.

[78] S. J. Sohrawardi, A. Chintha, B. Thai, S. Seng, A. Hickerson, R. Ptucha and M. K. Wright, "Poster: Towards Robust Open-World Detection of Deepfakes," in *ACM SIGSAC Conference on Computer and Communications Security,* London, 2019.

[79] T. Fernando, C. Fookes, S. Denman and . S. Sridharan, "Exploiting Human Social Cognition for the Detection of Fake and Fradulent Faces via Memory Networks," in *arXiv:1911.07844v1,* 2019.

[80] X. Sun, B. Wu and W. Chen, "Identifying Invariant Texture Violation for Robust Deepfake Detection," in *arXiv:2012.10580v1,* 2020.

[81] X. Ding, Z. Raziei, E. C. Larson, E. V. Olinick, P. Krueger and . M. Hahsler, "Swapped face detection using deep learning and subjective assessment," *EURASIP Journal on Information Security,* vol. 6, 2020.

[82] A. Kumar, A. Bhavsar and R. Verma, “Detecting Deepfakes with Metric Learning,” in *International Workshop on Biometrics and Forensics (IWBF)*, Porto, 2020.

[83] M. S. Rana and A. H. Sung, “DeepfakeStack: A Deep Ensemble-based Learning Technique for Deepfake Detection,” in *IEEE International Conference on Cyber Security and Cloud Computing*, New York, 2020.

[84] X. Zhou, Y. Wang and P. Wu, “Detecting Deepfake Videos via Frame Serialization Learning,” in *IEEE 3rd International Conference of Safe Production and Informatization (IICSPI)*, Chongqing City, 2020.

[85] X. H. Nguyen, . T. S. Tran, . V. T. Le, . K. D. Nguyen and D.-T. Truong, “Learning Spatio-temporal features to detect manipulated facial videos created by the Deepfake techniques,” *Forensic Science International: Digital Investigation*, vol. 36, 2021.

[86] Z. Xu, . J. Liu, W. Lu, B. Xu, X. Zhao, B. Li and J. Huang, “Detecting facial manipulated videos based on set convolutional neural networks,” *Journal of Visual Communication and Image Representation*, vol. 77, 2021.

[87] Z. Chen and H. Yang, “Attentive Semantic Exploring for Manipulated Face Detection,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, 2021.

[88] J. Zhang, J. Ni and H. Xie, “DeepFake Videos Detection Using Self-Supervised Decoupling Network,” in *IEEE International Conference on Multimedia and Expo (ICME)*, Shenzhen, 2021.

[89] Z. Gu, Y. Chen, T. Yao, S. Ding, J. Li, F. Huang and L. Ma, “Spatiotemporal Inconsistency Learning for DeepFake Video Detection,” in *Proceedings of the 29th ACM International Conference on Multimedia*, New York, 2021.

[90] Y. Tu, Y. Liu and X. Li, “Deepfake Video Detection by Using Convolutional Gated Recurrent Unit,” in *International Conference on Machine Learning and Computing*, Shenzhen, 2021.

[91] Y.-X. Zhuang and C.-C. Hsu, “Detecting Generated Image Based on a Coupled Network with Two-Step Pairwise Learning,” in *IEEE International Conference on Image Processing (ICIP)*, Taipei, 2019.

[92] O. d. Lima, S. Franklin, . S. Basu, B. Karwoski and A. George, “Deepfake Detection using Spatiotemporal Convolutional Networks,” in *arXiv:2006.14749v1* , 2020.

[93] Y. Lang, X. Li, Y. Chen, X. Mao, Y. He, S. Wang, H. Xue and Q. Lu, “Sharp Multiple Instance Learning for DeepFake Video Detection,” in *Proceedings of the 28th ACM International Conference on Multimedia*, Seattle WA, 2020.

[94] B. Chen, X. Ju, B. Xiao, W. Ding, Y. Zheng and V. H. C. d. Albuquerque, “Locally GAN-generated face detection based on an improved Xception,” *Information Sciences*, vol. 572, pp. 16-28, 2021.

[95] H.-S. Chen, M. Rouhsedaghat, H. Ghani, S. Hu, S. You and C.-C. J. Kuo, "DefakeHop: A Light-Weight High-Performance Deepfake Detector," in *IEEE International Conference on Multimedia and Expo (ICME)*, Shenzhen, 2021.

[96] S. Das, S. Seferbekov, A. Datta, M. S. Islam and M. R. Amin, "Towards Solving the DeepFake Problem : An Analysis on Improving DeepFake Detection using Dynamic Face Augmentation," in *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, 2021.

[97] H. H. Nguyen, F. Fang, J. Yamagishi and I. Echizen, "Multi-task Learning for Detecting and Segmenting Manipulated Facial Images and Videos," in *IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Tampa, 2019.

[98] M. Du, S. K. Pentyala, Y. Li and X. Hu, "Towards Generalizable Deepfake Detection with Locality-aware AutoEncoder," in *ACM International Conference on Information & Knowledge Management*, Virtual Event Ireland, 2020.

[99] P. He, H. Li and H. Wang, "Detection of Fake Images Via The Ensemble of Deep Representations from Multi Color Spaces," in *IEEE International Conference on Image Processing (ICIP)*, Taipei, 2019.

[100] Z. Guo, G. Yang, J. Chen and X. Sun, "Fake face detection via adaptive manipulation traces extraction network," *Computer Vision and Image Understanding*, vol. 204, 2021.

[101] R. Wang, F. Juefei-Xu, L. Ma, X. Xie, Y. Huang, J. Wang and Y. Liu, "FakeSpotter: A Simple yet Robust Baseline for Spotting AI-Synthesized Fake Faces," in *International Joint Conference on Artificial Intelligence(IJCAI)*, Yokohama, 2020.

[102] S. A. Khan and H. Dai, "Video Transformer for Deepfake Detection with Incremental Learning," in *Proceedings of the 29th ACM International Conference on Multimedia*, New York, 2021.

[103] Y. Lin, H. Chen, B. Li and J. Wu, "Towards Generalizable DEEPFAKE Face Forgery Detection with Semi-Supervised Learning and Knowledge Distillation," in *IEEE International Conference on Image Processing (ICIP)*, Bordeaux, France, 2022.

[104] H. Chen, Y. Lin, B. Li and S. Tan, "Learning Features of Intra-Consistency and Inter-Diversity: Keys Toward Generalizable Deepfake Detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 3, pp. 11468-1480, 2023.

[105] Z. Hou, Z. Hua, K. Zhang and Y. Zhang, "CDNet: Cluster Decision for Deepfake Detection Generalization," in *IEEE International Conference on Image Processing (ICIP)*, Kuala Lumpur, Malaysia, 2023.

[106] W. Guan, W. Wang, J. Dong and B. Peng, "Improving Generalization of Deepfake Detectors by Imposing Gradient Regularization," *IEEE Transactions on Information Forensics and Security*, vol. 19, 2024.

[107] Z. Guo, L. Wang, W. Yang, G. Yang and K. Li, "LDFnet: Lightweight Dynamic Fusion Network for Face Forgery Detection by Integrating Local Artifacts and Global Texture Information," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 2, pp. 1255 - 1265, 2024.

[108] Z. Yan, Y. Zhang, Y. Fan and B. Wu, "UCF: Uncovering Common Features for Generalizable Deepfake Detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France, 2023.

[109] P. Yu, J. Fei, X. Zhihua, Z. Zhili and J. Weng, "Improving Generalization by Commonality Learning in Face Forgery Detection," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 547-558, 2022.

[110] J. Li, Z. Yu, G. Luo and Y. Zhu, "CodeDetector: Revealing Forgery Traces with Codebook for Generalized Deepfake Detection," in *Proceedings of the 2024 International Conference on Multimedia Retrieval*, 2024.

[111] J. Frank, T. Eisenhofer, . L. Schonherr, A. Fischer, D. Kolossa and T. Holz, "Leveraging Frequency Analysis for Deep Fake Image Recognition," *Proceedings of Machine Learning*, vol. 119, pp. 3247-3258, 2020.

[112] R. Durall, M. Keuper, F.-J. Pfreundt and J. Keuper, "Unmasking DeepFakes with simple Feature," in *arXiv:1911.00686v3*, 2020.

[113] I. Masi, A. Killekar, R. M. Mascarenha, S. P. Gurudatt and W. AbdAlmageed, "Two-Branch Recurrent Network for Isolating Deepfakes in Videos," in *European Conference on Computer Vision*, Glasgow, 2020.

[114] J. Yang , A. Li, S. Xiao, W. Lu and X. Gao, "MTD-Net: Learning to Detect Deepfakes Images by Multi-Scale Texture Difference," *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, vol. 16, pp. 4234 - 4245, 2021.

[115] B. Liu and C.-M. Pun, "Exposing splicing forgery in realistic scenes using deep fusion network," *Information Sciences*, vol. 526, pp. 133-150, 2020.

[116] L. A. Gatys, A. S. Ecker and M. Bethge, "Texture synthesis using convolutional neural networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Montreal, Quebec, Canada, 2015.

[117] C.-F. (. Chen, Q. Fan and R. Panda, "CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification," in *Computer Vision and Pattern Recognition*, Nashville, Tennessee, 2021.

[118] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, . X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *2021*, In International Conference on Learning Representations.

[119] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[120] E. D. Cubuk, B. Zoph, J. Shlens and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, 2020.

[121] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo and J. Choe, “CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea, 2019.

[122] H. Zhang, . M. Cisse, Y. N. Dauphin and D. Lopez-Paz, “mixup: Beyond Empirical Risk Minimization,” in *In International Conference on Learning Representations*, 2018.

[123] Z. Zhong, L. Zheng, G. Kang, S. Li and Y. Yang, “Random Erasing Data Augmentation,” in *Proceedings of the AAAI Conference on Artificial Intelligence,*, 2020.

[124] J. Hu, X. Liao, W. Wang and Z. Qin, “Detecting Compressed Deepfake Videos in Social Networks Using Frame-Temporality Two-Stream Convolutional Network,” *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 32, no. 3, pp. 1089 - 1102, 2021.

[125] D. Afchar, V. Nozick, J. Yamagishi and I. Echizen, “MesoNet: a Compact Facial Video Forgery Detection Network,” in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Hong Kong, China, 2019.

[126] D. Coccomini, . N. Messina, . C. Gennaro and F. Falchi, “Combining EfficientNet and Vision Transformers for Video Deepfake Detection,” in *International Conference on Image Analysis and Processing*, 2022.

[127] Z. Yang, J. Liang, Y. Xu, X.-Y. Zhang and R. He, “Masked Relation Learning for DeepFake Detection,” *IEEE Transactions on Information Forensics and Security,* vol. 18, pp. 1696 - 1708, 2023.

[128] B. Huang, Z. Wang, J. Yang, J. Ai, Q. Zou, Q. Wang and D. Ye, “Implicit Identity Driven Deepfake Face Swapping Detection,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada, 2023.

[129] F. Chollet, “Xception: Deep Learning with Depth-wise Separable Convolutions,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017.

[130] S.-Y. Wang, O. Wang, R. Zhang, A. Owens and A. A. Efros, “CNN-Generated Images Are Surprisingly Easy to Spot… for Now,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020.

[131] M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in *36th International Conference on Machine Learning*, 2019.

[132] D. Dagar and D. K. Vishwakarma, “A literature review and perspectives in deepfakes: generation, detection, and applications,” *International Journal of Multimedia Information Retrieval,* vol. 11, pp. 219-289, 2022.

[133] J. Cao, C. Ma, T. Yao, S. Chen, S. Ding and X. Yang, “End-to-End Reconstruction-Classification Learning for Face Forgery Detection,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 2022.

[134] K. Shiohara and T. Yamasaki, “Detecting Deepfakes with Self-Blended Images,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 2022.

[135] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, “Generative Adversarial Nets,” in *Advances in Neural Information Processing Systems*, Montreal Canada, 2014.

[136] K. He, X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016.

[137] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in *International Conference on Machine Learning*, 2019.

[138] D. Wodajo and S. Atnafu, “Deepfake Video Detection Using Convolutional Vision Transformer,” in *Computer Vision and Pattern Recognition*, Nashville, TN, 2021.

[139] P. Korshunov and S. Marcel, “DeepFakes: a New Threat to Face Recognition? Assessment and Detection,” in *arXiv:1812.08685v1*, 2018.

[140] Y. Li, X. Yang, P. Sun, H. Qi and S. Lyu, “Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, 2020.

[141] B. Dolhansky, . J. Bitton, B. Pflaum, J. Lu, R. Howes, . M. Wang and C. C. Ferrer, “The DeepFake Detection Challenge (DFDC) Dataset,” in *arXiv:2006.07397v4*, 2020.

[142] L. Jiang, R. Li, W. Wu, C. Qian and C. C. Loy, “DeeperForensics-1.0: A Large-Scale Dataset for Real-World Face Forgery Detection,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, 2020.

[143] B. Dolhansky, R. Howes, B. Pflaum, N. Baram and C. C. Ferrer, “The Deepfake Detection Challenge (DFDC) Preview Dataset,” in *arXiv:1910.08854v2*, 2019.

[144] “Contributing Data to Deepfake Detection Research,” 2019. [Online]. Available: https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html.

[145] B. Zi, . M. Chang, J. Chen, X. Ma and Y.-G. Jiang, “WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection,” in *Proceedings of the 28th ACM International Conference on Multimedia*, Seattle, 2020.

[146] H. Khalid, S. Tariq, M. Kim and S. Woo, “FakeAVCeleb: A Novel Audio-Video Multimodal Deepfake Dataset,” in *Neural Information Processing systems(NIPS)*, South Korea, 2021.

[147] Y. Nirkin, Y. Keller and T. Hassner, “FSGAN: Subject Agnostic Face Swapping and Reenactment,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, 2019.

[148] “Rudrabha/Wav2Lip,” github, [Online]. Available: https://github.com/Rudrabha/Wav2Lip.

[149] S. Woo, . J. Park, J.-Y. Lee and I. S. Kweon, “CBAM: Convolutional Block Attention Module,” in *European Conference on Computer Vision*, Munich, Germany, 2018.

[150] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[151] H. Wu, B. Xiao, N. Codella, L. Mengchen, D. Xiyang, L. Yuan and L. Zhang, “CvT: Introducing Convolutions to Vision Transformers,” in *Computer Vision and Pattern Recognition*, Nashville, TN, 2021.

[152] H. H. Nguyen, J. Yamagishi and I. Echizen, “Capsule-forensics: Using Capsule Networks to Detect Forged Images and Videos,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019.

[153] Z. Liu, X. Qi and H. S. P. Torr, “Global Texture Enhancement for Fake Face Detection In the Wild,” in *Conference on Computer Vision and Pattern Recognition(CVPR)*, Virtual, 2020.

[154] Y.-f. Hsu and S.-f. Chang, “Detecting Image Splicing using Geometry Invariants and Camera Characteristics Consistency,” in *IEEE International Conference on Multimedia and Expo*, Toronto, 2006.

[155] K. He, X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 2016.

[156] B. Bayar and M. C. Stamm, “Constrained Convolutional Neural Networks: A New Approach Towards General Purpose Image Manipulation Detection,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2691 - 2706, 2018.

[157] J. Fridrich and J. Kodovsky, “Rich Models for Steganalysis of Digital Images,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868-882, 2012.

[158] Z. Lai, H. Sun, R. Tian, N. Ding, Z. Wu and Y. Wang, “Rethinking Skip Connections in Encoder-decoder Networks for Monocular Depth Estimation,” in *10.48550/arXiv.2208.13441*, 2022.

[159] S. Woo, J. Park, J.-Y. Lee and I. S. Kweon, “CBAM: Convolutional Block Attention Module,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, 2018.

[160] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang and H. Lu, “Dual Attention Network for Scene Segmentation,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019.

[161] J. Dong, W. Wang and T. Tan, “CASIA Image Tampering Detection Evaluation Database,” in *IEEE China Summit and International Conference on Signal and Information Processing*, Beijing, China, 2013.

[162] T.-T. Ng, J. Hsu and S.-F. Chang, “Columbia Image Splicing Detection Evaluation Dataset,” in *DVMM Laboratory of Columbia University*, 2009.

[163] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen and S. Winkler, "COVERAGE—A novel database for copy-move forgery detection," in *IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, 2016.

[164] H. Guan, M. Kozak, E. Robertson, Y. Lee, A. N. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, J. Smith and J. Fiscus, "MFC Datasets: Large-Scale Benchmark Datasets for Media Forensic Challenge Evaluation," in *IEEE Winter Applications of Computer Vision Workshops (WACVW)*, Waikoloa, HI, USA, 2019.

[165] J. Zhang, H. Tohidypour, Y. Wang and P. Nasiopoulos, "Shallow- and Deep- fake Image Manipulation Localization Using Deep Learning," in *International Conference on Computing, Networking and Communications (ICNC)*, Honolulu, HI, USA, 2023.

[166] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Kuala Lumpur, Malaysia, 2015.

[167] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009.

[168] S. Li, S. Xu, W. Ma and Q. Zong, "Image Manipulation Localization Using Attentional Cross-Domain CNN Features," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 5614 - 5628, 2023.

[169] Neal Krawetz and Hacker Factor Solutions, "A picture's worth…digital image Analysis and Forensics," *Hacker Factor Solutions*, vol. 2, no. 2, p. 2, 2007.

[170] B. Mahdian and S. Saic, "Using noise inconsistencies for blind image forensics," *Image and Vision Computing*, vol. 27, no. 10, pp. 1497-1503, 2009.

[171] P. Ferrara, T. Bianchi, A. D. Rosa and A. Piva, "Image Forgery Localization via Fine-Grained Analysis of CFA Artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1566-1577, 2012.

[172] R. Salloum, Y. Ren and C.-C. J. Kuo, "Image Splicing Localization using a Multi-task Fully Convolutional Network (MFCN)," *Journal of Visual Communication and Image Representation*, vol. 51, pp. 201-209, 2018.

[173] P. Zhou, X. Han, V. I. Morariu and L. S. Davis, "Learning Rich Features for Image Manipulation Detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018.

[174] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj and B. Manjunath, "Exploiting Spatial Structure for Localizing Manipulated Image Regions," in *IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017.

[175] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath and A. K. Roy-Chowdhury, "Hybrid LSTM and Encoder–Decoder Architecture for Detection of Image Forgeries," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3286-3300, 2019.

[176] C. Yang, H. Li, F. Lin, B. Jiang and H. Zhao, "Constrained R-Cnn: A General Image Manipulation Detection Model," in *IEEE International Conference on Multimedia and Expo (ICME)*, London, UK, 2020.

[177] Y. Wu, W. AbdAlmageed and P. Natarajan, "ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries With Anomalous Features," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019.

[178] P. Zhou, B.-C. Chen, X. Han, M. Najibi, A. Shrivastava, S.-N. Lim and L. Davis, "Generate, Segment, and Refine: Towards Generic Manipulation Segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, USA, 2020.

[179] X. Hu, Z. Zhang, Z. Jiang, S. Chaudhari, Z. Yang and R. Nevatia, "SPAN: Spatial Pyramid Attention Network for Image Manipulation Localization," in *European Conference on Computer Vision*, Glasgow, United Kingdom, 2020.

[180] X. Liu, Y. Liu, J. Chen and X. Liu, "PSCC-Net: Progressive Spatio-Channel Correlation Network for Image Manipulation Detection and Localization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7505-7517, 2022.

[181] C. Dong, X. Chen, R. Hu, J. Cao and X. Li, "MVSS-Net: Multi-View Multi-Scale Supervised Networks for Image Manipulation Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3539-3553, 2022.

[182] J. Wang, Z. Wu, J. Chen, X. Han, A. Shrivastava, S.-N. Lim and Y.-G. Jiang, "ObjectFormer for Image Manipulation Detection and Localization," in *Computer Vision and Pattern Recognition*, New Orleans, 2022.

[183] Z. Shi, H. Chen and D. Zhang, "Transformer-Auxiliary Neural Networks for Image Manipulation Localization by Operator Inductions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 9, pp. 4907-4920, 2023.

[184] J. Hao, Z. Zhang, S. Yang, D. Xie and S. Pu, "TransForensics: Image Forgery Localization with Dense Self-Attention," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021.

[185] Z. Zhang, Y. Qian, Y. Zhao, X. Zhang, L. Zhu, J. Wang and J. Zhao, "Noise and Edge Based Dual Branch Image Manipulation Detection," in *International Conference on Computing, Networks and Internet of Things*, New York, 2023.

[186] C. Kong, B. Chen, H. Li, S. Wang, A. Rocha and S. Kwong, "Detect and Locate: Exposing Face Manipulation by Semantic- and Noise-Level Telltales," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1741-1756, 2022.

[187] D. Tantaru, E. Oneata and D. Oneata, "Weakly-supervised deepfake localization in diffusion-generated images," in *IEEE Workshop on Applications of Computer Vision (WACV)*, Hawai, 2024.

[188] Y. Lai, . Z. Luo and Z. Yu, "Detect Any Deepfakes: Segment Anything Meets Face Forgery Detection and Localization," in *Biometric Recognition*, Singapore, 2023.

[189] X. Chen, C. Dong, J. Ji, J. Cao and X. Li, “Image Manipulation Detection by Multi-View Multi-Scale Supervision,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC Canada, 2021.

[190] L.-C. Chen, G. Papandreou, F. Schroff and H. Adam, “Rethinking Atrous Convolution for Semantic Image Segmentation,” in *arxiv.org/abs/1706.05587*, 2017.

[191] E. Shelhamer, J. Long and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640 - 651, 2016.

[192] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam and Q. Le, “Searching for MobileNetV3,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019.

[193] E. Shelhamer, J. Long and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640 - 651, 2017.

PROOF OF PUBLICATIONS

SCIE Journal Paper 1:

❖ **D. Dagar** and D. K. Vishwakarma, "A literature review and perspectives in deepfakes: generation, detection and applications" ***International Journal of Multimedia Information Retrieval***, vol. 11, June. 2022, doi: https://doi.org/10.1007/s13735-022-00241-w.

SPRINGER LINK

Account

Find a journal | Publish with us | Track your research | Search | Cart

Home > International Journal of Multimedia Information Retrieval > Article

A literature review and perspectives in deepfakes: generation, detection, and applications

Trends and Surveys | Published: 23 July 2022

Volume 11, pages 219–289, (2022) Cite this article

Download PDF | Access provided by Delhi Technological University

International Journal of Multimedia Information Retrieval

Aims and scope →

Submit manuscript →

Deepak Dagar & Dinesh Kumar Vishwakarma

3786 Accesses | 11 Citations | 1 Altmetric | Explore all metrics →

Use our pre-submission checklist →

Avoid common mistakes on your manuscript.

Abstract

In the last few years, with the advancement of deep learning methods, especially Generative Adversarial Networks (GANs) and Variational Auto-encoders (VAEs), fabricated content has become more realistic and believable to the naked eye. Deepfake is one such emerging technology that allows the creation of highly realistic, believable synthetic content. On the one hand, Deepfake has paved the way for highly advanced applications in various fields like advertising, creative arts, and film productions. On the other hand, it poses a threat to various Multimedia Information Retrieval Systems (MIPR) such as face recognition and speech recognition systems and has more significant societal implications in spreading misleading information. This paper aims to assist an individual in understanding the deepfake technology (along with its application), current state-of-the-art methods and gives an idea about the future pathway of this technology. In this paper, we have presented a comprehensive literature survey on the application of deepfakes, followed by discussions on state-of-the-art methods for deepfake generation and detection for three media: Image, Video, and Audio. Next, we have extensively discussed the architectural components and dataset used for various methods of deepfakes. Furthermore, we discuss the various limitations and open challenges of deepfakes to identify the research gaps in this field. Finally, discuss the conclusion and future directions to explore the potential of this technology in the coming years.

Sections | Figures | References

Abstract
Introduction
Deepfake techniques
Deepfake datasets
Architectural components and tools for deepfake
Current limitations and open challenges
Conclusion and future work
Notes
References
Author information
Ethics declarations
Additional information
Rights and permissions
About this article

Advertisement

SCIE Journal Paper 2:

❖ **D. Dagar** and D. K. Vishwakarma, “Tex-Net: Texture-based parallel branch cross-attention generalized robust deepfake detector” vol 30, article number 233, ***Multimedia Systems***, 2024 (Pub: Springer), doi: https://doi.org/10.1007/s00530-024-01424-7.

SPRINGER LINK

Account

Find a journal | Publish with us | Track your research | Search | Cart

Home > Multimedia Systems > Article

Tex-Net: texture-based parallel branch cross-attention generalized robust Deepfake detector

Regular Paper | Published: 01 August 2024

Volume 30, article number 233, (2024) Cite this article

Download PDF

Access provided by Delhi Technological University

Multimedia Systems

Aims and scope →

Submit manuscript →

Deepak Dagar & Dinesh Kumar Vishwakarma

77 Accesses Explore all metrics →

Use our pre-submission checklist →

Avoid common mistakes on your manuscript.

Abstract

In recent years, artificial faces generated using Generative Adversarial Networks (GANs) and Variational Auto-encoders (VAEs) have become more lifelike and difficult for humans to distinguish. Deepfake refers to highly realistic and impressive media generated using deep learning technology. Convolutional Neural Networks (CNNs) have demonstrated significant potential in computer vision applications, particularly identifying fraudulent faces. However, if these networks are trained on insufficient data, they cannot effectively apply their knowledge to unfamiliar datasets, as they are susceptible to inherent biases in their learning process, such as translation, equivariance, and localization. The attention mechanism of vision transformers has effectively resolved these limits, leading to their growing popularity in recent years. This work introduces a novel module for extracting global texture information and a model that combines data from CNN (ResNet-18) and cross-attention vision transformers. The model takes in input and generates the global texture by utilizing Gram matrices and local binary patterns at each down sampling step of the ResNet-18 architecture. The ResNet-18 main branch and global texture module operate simultaneously before inputting into the visual transformer’s dual branch’s cross-attention mechanism. Initially, the empirical investigation demonstrates that counterfeit images typically display more uniform textures that are inconsistent across long distances. The model’s performance on the cross-forgery dataset is demonstrated by experiments conducted on various types of GAN images and Faceforensics + + categories. The results show that the model outperforms the scores of many state-of-the-art techniques,

Sections | Figures | References

Abstract

Introduction

Related work

Empirical analytical observation

Model framework

Experimentation

Ablation studies

Complexity analysis of tex-ViT

Visualization outcomes of the LBP-ViT’s predicti...

Conclusion

Data availability

References

Author information

Ethics declarations

Additional information

Rights and permissions

SCIE Journal Paper 3:

❖ **D. Dagar** and D. K. Vishwakarma, "Shallowfake and Deepfake Image Manipulation Localization using Noise and RGB-based dual branch method" ***Signal, Image and Video Processing***, vol 18, pages 7065-7077, 2024, doi: https://doi.org/10.1007/s11760-024-03376-x

SPRINGER LINK

Account

Find a journal | Publish with us | Track your research | Search | Cart

Home > Signal, Image and Video Processing > Article

Shallowfake and deepfake image manipulation localization using noise and RGB-based dual branch method

Original Paper | Published: 25 June 2024

(2024) Cite this article

Download PDF | Access provided by Delhi Technological University

Signal, Image and Video Processing

Aims and scope →

Submit manuscript →

Deepak Dagar & Dinesh Kumar Vishwakarma

90 Accesses | Explore all metrics →

Use our pre-submission checklist →

Avoid common mistakes on your manuscript.

Abstract

The reliability of multimedia is being progressively tested by sophisticated Image Manipulation localization (IML) methods, which has led to the creation of the IML domain. A good manipulation model requires extracting non-semantic differences features between manipulated and authentic regions to exploit artifacts, which calls for explicit comparisons between the two areas. Existing models either use handcrafted-based feature methods, convolutional neural networks (CNNs), or a combination of both. Handcrafted feature methods assume the tampering beforehand, limiting their capabilities for diverse tampering operations, while CNNs model semantic information, which is not enough for the manipulation artifact. To improve these limitations, we have designed a dual-branch model that combines handcrafted feature noise and CNNs as an Encoder-decoder(ED) powered by the attention mechanism. This dual-branch model uses noise features on one branch and RGB on the other before feeding to an ED architecture for semantic learning and skip connection deployed to retain spatial information. Furthermore, this architecture uses channel spatial attention to strengthen further and refine the features' representation. Extensive experimentation on the shallowfakes dataset (CASIA, COVERAGE, COLUMBIA, NIST16) and deepfake datasets Faceforensics + + (FF + +) to demonstrate the superior feature extraction capabilities and performance to various baseline models with AUC score even reaching 99%. Also, it is one of the first methods to perform localization on the deepfake dataset. The model is relatively lighter, has 38 million parameters, and easily outperforms other State-of-the-Art(SoTA) models.

Sections | Figures | References

Advertisement

Conference Paper 1:

❖ **D. Dagar** and D. K. Vishwakarma "Div-Df: A Diverse Manipulation Deepfake Video Dataset" ***IEEE Conference: Global Conference on Information Technologies and Communications(GCITC)***,Bengaluru. (2023), doi: 10.1109/GCITC60406.2023.10426446.

Conferences > 2023 Global Conference on Inf...

Div-Df: A Diverse Manipulation Deepfake Video Dataset

Publisher: IEEE | Cite This | PDF

Deepak Dagar ; Dinesh Kumar Vishwakarma All Authors

46 Full Text Views

Abstract

Document Sections

I. Introduction

II. Related work

III. Diverse Video Deepfake Dataset(DIV-DF)

IV. Benchmark Evaluation

V. Conclusion

Authors

Figures

References

Keywords

Metrics

Abstract:

Recent advances in image and video manipulation have given rise to grave concerns. Deepfake technology employs deep learning techniques to produce astoundingly lifelike content. Deepfakes are risky since they have the ability to counterfeit someone's identity by replacing their face with that of another person or generating random noise in the mouth area. Additionally, with just a few seconds of audio, AI-based deep learning models can replicate any person's voice. Detecting such videos is the only promising defense against such fraudulent data. Several deepfake datasets have been made available to help in deepfake detector training and testing, including DF-TIMIT [1], FaceForensics++ [2], Celeb-DF [3], DFDC [4], Deeperforensics1.0 [5], etc. Even though this has significantly improved deepfake detection methods, they are still unable to capture real-world scenarios entirely, as most of the dataset is face-swap manipulation. To bridge this gap, we have proposed a Div-DF dataset containing various types of video manipulation like face swap, facial reenactment, and lip-sync. This dataset is composed of 150 real videos of different celebrities of different professions and 250 deepfake videos (100 face-swap videos, 100 facial reenactment videos, and 50 lip-sync videos). Deepfake videos are synthesized using state-of-the-art Face-Swap GAN(FSGAN) and the Wav2Lip method. The dataset contains high-quality samples of face-swapped and lip-sync videos, while the samples of face-re-enactment are of average quality. We have tested state-of-the-art detection and image classification models to standardize our dataset's baseline evaluation of various detection methods. We have done a comprehensive assessment along different metrics and found that our dataset is challenging and represents real-world samples.

Published in: 2023 Global Conference on Information Technologies and Communications (GCITC)

Date of Conference: 01-03 December 2023

DOI: 10.1109/GCITC60406.2023.10426446

Date Added to IEEE *Xplore*: 18 April 2024

Publisher: IEEE

Conference Paper 2:

❖ **D. Dagar** and D. K. Vishwakarma **"**A Hybrid Xception-LSTM model with channel and Spatial Attention for Deepfake Video Detection**"** ***IEEE Conference: International Conference on Mobile Networks and Wireless Communications,*** Tumakur, Karnataka. (2023), doi: 10.1109/ICMNWC60182.2023.10435983.

Conferences > 2023 3rd International Confer...

A Hybrid Xception-LSTM Model with Channel and Spatial Attention Mechanism for Deepfake Video Detection

Publisher: IEEE | Cite This | PDF

Deepak Dagar ; Dinesh Kumar Vishwakarma All Authors

139 Full Text Views

Abstract

Document Sections

I. Introduction

II. Related Works

III. Methods

IV. Results and Discussion

V. Conclusion

Authors

Figures

References

Keywords

Abstract:

The great strides taken in recent times in image and video manipulation have raised serious concerns. Deepfake technology uses deep learning approaches to create highly realistic, astonishing content. Detecting such videos is the only promising defense against such fraudulent data. To counter the malicious intent of the user, a deepfake detection model is proposed that employs channel and spatial attention mechanisms(CBAM) along with Xception and LSTM pretrained models. Xception uses depthwise separable convolution to capture the latent spatial artifacts. LSTM captures the discrepancies among the manipulated sequences; hence, this hybrid ensembling of models allows the learning of powerful features. The evaluation is performed on the recently proposed Div-DF dataset consisting of varied video manipulation like face swap, facial reenactment, and lip-sync. It shows that the model works well (Accuracy~ 93 % & AUC ~ 0.98) on the diversified dataset and easily beats the score of various state-of-the-art deepfake detection and image classification models.

Published in: 2023 3rd International Conference on Mobile Networks and Wireless Communications (ICMNWC)

Date of Conference: 04-05 December 2023

DOI: 10.1109/ICMNWC60182.2023.10435983

Date Added to IEEE *Xplore*: 22 February 2024

Publisher: IEEE

ISBN Information:

Conference Location: Tumkur, India

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis: DEVELOPMENT OF FRAMEWORK FOR DEEPFAKE DETECTION IN MULTIMEDIA DATA

Total Pages: 135

Name of the Scholar: Deepak Dagar

Supervisor: Prof. Dinesh Kumar Vishwakarma

Department: Information Technology

This is to report that the above thesis was scanned for similarity detection. The process and outcome are given below:

Software used: Turnitin Similarity Index: 5% Word Count: 41351 Words

Date: 10/08/2024

Candidate's Signature **Signature of Supervisor**

PLAGIARISM REPORT

PAPER NAME
Deepak Dagar_Thesis Final Copy SAFE VERSION - Copy.docx

AUTHOR
Deepak Dagar

WORD COUNT
41351 Words

CHARACTER COUNT
237059 Characters

PAGE COUNT
135 Pages

FILE SIZE
15.7MB

SUBMISSION DATE
Sep 4, 2024 12:12 AM GMT+5:30

REPORT DATE
Sep 4, 2024 12:15 AM GMT+5:30

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 1% Internet database
- 4% Publications database
- Crossref database
- Crossref Posted Content database
- 1% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 10 words)
- Manually excluded sources

Author Biography

Deepak Dagar received his B.Tech degree in Software Engineering from Delhi Technological University (DTU), Delhi, in 2014 and M.Tech from Netaji Subhas Institute of Technology (NSIT), Delhi, in 2016. He has worked in Bombardier Transportation Private Limited from 2016- 2019 as software testing engineer and software test lead. He is currently a part-time research scholar at Delhi Technological University, Delhi. The topic of his dissertation is Development of framework for Deepfake Detection in Multimedia Data. His current interests include, deep learning, computer vision, deepfake multimedia generation and detection.