DESIGN AND DEVELOPMENT OF FRAMEWORK TO DETECT MALICIOUS MANIPULATIONS IN MULTIMEDIA DATA

A Dissertation submitted to

DELHI TECHNOLOGICAL UNIVERSITY

for the Award of Degree of

DOCTOR OF PHILOSOPHY

In

INFORMATION TECHNOLOGY

By
Ankit Yadav

Under the Supervision of
Prof. Dinesh Kumar Vishwakarma
Information Technology

Department of Information Technology
Delhi Technological University
(Formerly Delhi College of Engineering)
Delhi, India
January, 2024

DECLARATION

I certify that the dissertation titled "***Design and Development of Framework to Detect Malicious Manipulations in Multimedia Data***" that I am submitting for the Doctor of Philosophy degree is solely my own and has not been previously submitted for any other degree or certification from any other academic institution. The research conducted in this thesis is original and has been independently carried out by me under the guidance of my supervisor.

Place:

Date:

This is to certify that the above statements made by the candidate are true.

Ankit Yadav
2K19/PHDIT/01

CERTIFICATE

This is to certify that the work contained in the thesis entitled "***Design and Development of Framework to Detect Malicious Manipulations in Multimedia Data***" submitted by Mr. Ankit Yadav (2K19/PHDIT/01) to Delhi Technological University, India, for the degree of Doctor of Philosophy is based on his unique research work conducted under my supervision.

He has met all the necessary criteria to submit the thesis in accordance with the specified standards. I affirm the authenticity of the work and verify that the thesis has not been used as the foundation for the approval of any degree or equivalent distinction.

Prof. Dinesh Kumar Vishwakarma
Supervisor
Head of Department,
Information Technology,
Delhi Technological University

ACKNOWLEDGEMENT

I am profoundly grateful to my esteemed PhD supervisor, **Prof. Dinesh Kumar Vishwakarma**, whose exceptional guidance and unwavering support have been instrumental in completing this thesis. His exemplary discipline, unyielding focus, and relentless work ethic have inspired me and set a high standard for academic excellence. I am fortunate to have benefited from his expertise and commendable commitment throughout this challenging yet rewarding journey. I thank **Mrs. Sushma Vishwakarma**, **Diya** and **Advika** for ensuring I always had a home away from home.

No man is complete without his family, who silently toil behind the scenes to help him fight for his dreams. I thank my parents, **Mr. Raj Kishor Yadav** and **Mrs. Vandana Yadav,** for being the best parents one could hope for. I thank my siblings **Shreya**, **Akansha**, and **Harshit** for doing what siblings do best, i.e., be loving in their own fun way. I am also grateful for the blessings of my late chacha ji, **Mr. Rajendra Yadav** and chachi ji, **Mrs. Kusum Lata**.

Finding great friends is like finding superpowers, and I have been blessed with several of them. I thank **Premdeep Singh, Rohit Arora, Harshit Bhatia** and **Pranjal Sharma** for the unwavering encouragement and shared laughter that provided the much-needed balance to the demands of academic life. Their companionship has served as a fortifying force, and I am thankful for the countless moments of camaraderie.

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. Chhavi Dhiman, Dr. Tej Singh, Dr. Ashima Yadav, Dr. Deepika Varshney** and **Dr. Priyanka Meel.**

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 **Deepak Dagar, 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.

Ankit Yadav
2K19/PHDIT/01

ABSTRACT

Due to the widespread usage of image and video editing tools, an alarming problem has emerged in an era characterised by the rapid spread of multimedia content on social media platforms. The combination of the simplicity and complexity of these innovations presents a substantial risk to the genuineness and reliability of the information included in multimedia files. This thesis emphasises the necessity to create robust systems for detecting dangerous alterations in multimedia data by using the potential of deep learning techniques. This is achieved by using the potential of deep learning algorithms. The susceptibility of audiovisual content to harmful changes has significantly increased, reaching unprecedented levels. This results from implementing modern technologies that facilitate the production of counterfeits with a high degree of authenticity. The objective of this study is to leverage the capabilities of deep learning to identify and mitigate such manipulations effectively. This research examines the incorporation of multimodal approaches, considering the many characteristics of multimedia material that are widespread in our present digital environment. Given that social media platforms are the main channels for sharing information, the suggested detection systems utilising deep learning aim to ensure the reliability and accuracy of multimedia material. As a result, this will enhance the establishment of a digital ecosystem characterised by increased reliability and credibility. This thesis tackles this manipulation detection challenge by proposing four novel deep-learning architectures and a novel image manipulation dataset that aids in training such forgery detection models.

The first two models, namely *MRT-Net* and *Face-NeSt* are dedicated to the problem of face manipulation detection. Facial manipulation is an extremely serious form of identity manipulation that can easily be used to mislead others and perform fraudulent activities. MRT-Net is a dual-branch architecture that extracts manipulation residuals and textural features to detect forgery in facial images. An auto-adaptive mechanism lets it dynamically choose the best proportion of the two features. Face-NeSt extracts the discriminative information from multiple scales of features extracted from a baseline model. Specifically, it extracts multi-scale attentional features fused adaptively, representing the best proportion of discriminative features. MRT-Net and Face-NeSt are evaluated on three public benchmark datasets: the FaceForensics ++ (FF++), DeepFake Detection Challenge (DFDC) and the CelebDF datasets. Experimental results prove that the proposed models are superior to the existing state-of-the-art methods.

The next two models are dedicated to the problem of detecting splice manipulation in images. The first framework has a dual-branch structure with a spatial and compression branch. The spatial branch leverages transfer learning to extract discriminative spatial clues without adding any significant computational cost. The second branch highlights inconsistencies in the DCT coefficient histograms caused by the splice forgery. The second model is a splice localization framework. It contains a unique "visually attentive multi-domain feature extractor" (VA-MDFE) that extracts attentional features from the RGB, edge and depth domains. Next, a "visually attentive downsampler" (VA-DS) is responsible for fusing and downsampling the multi-domain features. Finally, a novel "visually attentive multi-receptive field upsampler" (VA-MRFU) module employs multiple receptive field-based convolutions to upsample attentional features by focussing on different information scales. Experimental results conducted on the public benchmark dataset CASIA v2.0 prove the potency of the proposed model. A novel splice manipulation dataset has also been created from Python code and Adobe Photoshop software since the existing splice detection datasets have very few samples and are not ideally suited to train deep-learning models.

Lastly, the role of visual attention models is studied in the context of forgery detection. Specifically, five recently proposed visual attention mechanisms are integrated with a baseline convolutional neural network. The performance boost for each type of attention model is measured. Also, the increase in the computational cost for each type of attention is measured, and this tradeoff of performance vs complexity is presented.

LIST OF PUBLICATIONS

Publications Arising from Research Work in this Thesis

SCIE Journal Papers

1. **A. Yadav** and D. K. Vishwakarma, "MRT-Net: Auto-adaptive weighting of manipulation residuals and texture clues for face manipulation detection," **Expert Systems with Applications**, vol. 232, 2023.
2. **A. Yadav** and D. K. Vishwakarma, "AW-MSA: Adaptively weighted multi-scale attentional features for DeepFake detection," **Engineering Applications of Artificial Intelligence**, vol. 127 Part B, 2024.
3. **A. Yadav** and D. K. Vishwakarma, "Toward effective image forensics via a novel computationally efficient framework and a new image splice dataset," **Signal, Image and Video Processing**, 2024.
4. **A. Yadav** and D. K. Vishwakarma, "Datasets, Clues and State-of-the-Arts for Multimedia Forensics: An Extensive Review," **Expert Systems with Applications**, 2024.
5. **A. Yadav** and D. K. Vishwakarma, "A Visually Attentive Splice Localization Network with Multi-Domain Feature Extractor and Multi-Receptive Field Upsampler." Under Review in **IEEE Signal Processing Letters,** (https://arxiv.org/abs/2401.06995, 2024).

Conference Papers

1. **A. Yadav** and D. K. Vishwakarma, "Investigating the Impact of Visual Attention Models in Face Forgery Detection", in **International Conference on Applied Intelligence and Sustainable Computing** (ICAISC), Dharwad, Karnataka, 2023.
2. **A. Yadav** and D. K. Vishwakarma, "Recent Developments in Generative Adversarial Networks: A Review", in 2020 **IEEE Sixth International Conference on Multimedia Big Data** (BigMM), New Delhi, India, 2020.

Publications Arising from Research Work Outside this Thesis

SCIE Journal Papers

1. **A. Yadav,** D. Gupta and D. K. Vishwakarma, "Uncovering visual attention-based multi-level tampering traces for face forgery detection," **Signal, Image and Video Processing**, 2023.
2. V. Gupta, **A. Yadav** and D. K. Vishwakarma, "HumanPoseNet: An all-transformer architecture for pose estimation with efficient patch expansion and attentional feature refinement," **Expert Systems with Applications**, vol. 244, 2024.
3. **A. Yadav** and D. K. Vishwakarma, "Deep learning algorithms for person re-identification: sate-of-the-art and research challenges," **Multimedia Tools and Applications**, 2023.
4. **A. Yadav**, S. Aggarwal, D. K. Vishwakarma. "LiteFaceNet: Combining Light-Weight Computation with Dynamic Receptive Fields for Deepfake Detection." Under Review in **ACM Transactions on Multimedia Computing, Communications, and Applications,** 2023.
5. V. Gupta, **A. Yadav** and D. K. Vishwakarma, "FreqFaceNet: An Enhanced Transformer Architecture with Dual-Order Frequency Attention for Deepfake Detection." Under Review in **Journal of Visual Communication and Image Representation**, 2023**.**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

In today's digital era, the alteration of multimedia information poses a widespread danger in our society. Various tampering techniques, from basic modifications to advanced forgeries, present substantial obstacles to the validity and integrity of multimedia products. Common types of forgeries are image alterations, video editing, audio adjustments, and deepfake technologies, which may all be utilized to deceive, misinform, or influence individuals and groups. Detecting and detecting tampering is essential for protecting the reliability of media sources, maintaining the credibility of information, and defending the integrity of digital material when misinformation spreads quickly. The advancement of technology necessitates the creation of practical techniques to identify and prevent the manipulation of multimedia to lessen the adverse effects on society's understanding of truth and reality. This chapter presents the introductory study of malicious manipulation detection of multimedia content.

1.1 Growing Popularity of Social Media Platforms

The last decade has witnessed a tremendous rise in social media platforms. An extensive online presence has become a normal part of daily human lives. The number of active users on social media has grown tremendously, from just over 2 million active users at the beginning of 2015 to almost 4 million active users by the end of 2020 [1]. Also, the average person had about 8.6 social media accounts in 2020 [1]. It is an understatement to say that social media has become integral to everyday life. The importance of social media is discussed as follows:

- Social media connects people together.
- Social media provides a platform for sharing information, exchanging ideas, expressing opinions, etc.
- Social media also attracts a large number of passive information consumers. Users create and share multimedia data and view and explore data shared by other community users, group, organization, etc.
- Social media has an enormous impact on individuals' mental and emotional states.

1.2 Role of Big Data

With the growth of social media platforms, a massive amount of multimedia content is being created every hour. This gigantic and ever-increasing amount of multimedia data has been termed as '*Big Data'.* Users on different platforms freely share various aspects of their lives

via images, videos and text posts. The large amount of content, especially visual content with images and videos, creates a fast-changing, dynamic, and impactful impression on society as a whole. Some *critical aspects of big data* are:

- Big Data is a massive collection of multimedia content, including text, audio, images and videos.
- Such a massive collection of multimedia data has never been created before in human history and is largely due to the growing social media platforms.
- Big Data provides a clear picture of the personal lives of individuals, the functioning of organizations and the collective psyche of society as a whole.

1.3 Creation of Multimedia Manipulation Tools and Approaches

Several tools like Adobe Photoshop, Premier Pro, and Illustrator allow for modifying multimedia content, including images and videos. Such tools provide an extensive list of options to modify content and create enhanced and yet realistic manipulations. While these tools are primarily meant to modify multimedia content to improve the visual quality of samples, they can be easily used to harm individuals, groups or society. The same applies to the endless number of mobile applications targeted to modify and manipulate multimedia content.

Several recent state-of-the-art (SOTA) methods have been developed in the research domain to create realistic manipulations of images and videos. Manipulations such as deepfakes [2] provide serious identity manipulations that are so realistic that it is humanly difficult to distinguish between an original and a deepfake. Other manipulations include splicing [3], [4], copy-move [5], [6] and many more.

In this era of widespread social media popularity, the design and development of methods for malicious multimedia manipulation are proving harmful to society. While social media is the main engine behind producing massive amounts of multimedia data or big data, several malicious manipulation approaches can be enforced to use this unending source of images and videos to inflict harm upon individuals/organizations and promote the spread of misinformation.

1.4 Harmful Impacts of Multimedia Manipulation

Some harmful effects of manipulation approaches are discussed below:

- **Defamation:** An individual or an organization can be defamed by posting maliciously manipulated images/videos that create a false impression of wrong doings.
- **Frauds:** Facial manipulation methods facilitate faking identities. By pretending to be someone else in an image/video, fraudsters attempt to cause monetary losses to an unsuspecting individual.
- **Misdirection:** Manipulated multimedia can also be used to create misdirection and sway public opinion, often times to gain political advantage.
- **Fake News:** Maliciously editing images or repurposing an old multimedia sample to promote untrue news or rumours causes panic and distress in society.
- **Other Manipulations:** While the list of possible manipulations is endless, most of these can be used in some form or another to mislead, lie, manipulate and cause harm to individuals/organizations.

Fig. 1 clearly demonstrates the sensitive nature of the present time. Given the scenario of rising social media presence and the development of numerous malicious manipulation approaches for images and videos, undesirable communications such as hoaxes, scams, frauds, defamation, misdirection, etc., have become quite common and robust detection systems are required to prevent the damage caused to society through such manipulations. Hence, it is of paramount importance to develop novel manipulation detection approaches that are capable of detecting and finding tampered regions within multimedia images and videos [7], [8], [9], [10].

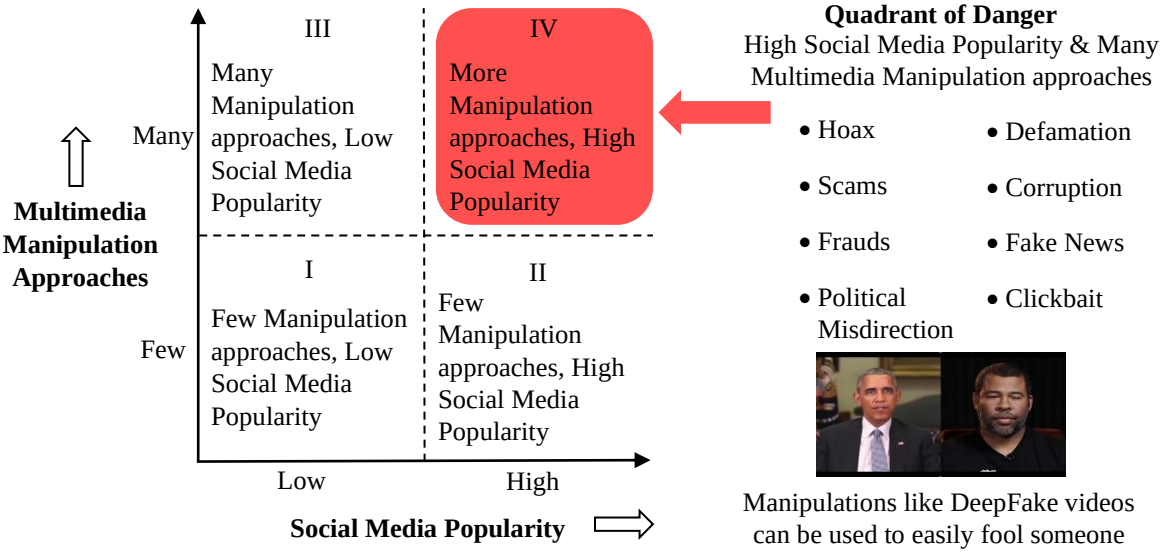


Fig. 1 Quadrant IV, with high social media popularity and the creation of numerous multimedia manipulation approaches, has given rise to a dangerous scenario in current times where it is easy to mislead, lie, defame and cause harm to an individual/organization.

1.5 Motivations for Detection of Multimedia Manipulation

There are several motivations behind proposing tamper detection methods:

- *Preventing Financial Fraud*: One of the key reasons for forgery detection systems in the financial sector is to avoid fraud. This includes identifying fake checks, counterfeit cash, and fraudulent credit card transactions.
- *Protection of Intellectual Property:* In the domain of intellectual property, forgery detection is used to safeguard copyrights, trademarks, and patents from being counterfeited or exploited.
- *Ensuring Document Authenticity:* Forgery detection technologies are critical in legal and government contexts for validating the validity of documents such as passports, driver's licences, birth certificates, and immigration paperwork.
- *Art Authentication:* In the art world, artwork authentication is vital to preventing art fraud. Forgery detection methods aid in determining if a work of art is genuine or a forgery.
- *Maintaining Data Integrity:* Forgery detection methods ensure data integrity in the digital era. Detecting falsified digital signatures, changed electronic documents, and modified photos or videos is part of this.
- *Protecting Brand Reputation:* Companies and brands use forgery detection to safeguard their reputation by identifying and blocking counterfeit items from entering the market.
- *Securing Identification and Access Control:* Forgery detection methods are used in security applications for biometric authentication (e.g., fingerprint, face recognition) and access control systems to prevent unauthorised access.
- *Ensuring Trust in Digital Transactions:* Forgery detection aids in establishing trust between parties in e-commerce and online financial transactions by confirming the legitimacy of digital identities and transactions.
- *Regulation Compliance:* Rules require various sectors to deploy forgery detection technologies as part of their compliance activities. Financial institutions, for example, are frequently required to implement anti-money laundering (AML) and know-your-customer (KYC) procedures.
- *Legal and Forensic Investigations:* Forgery detection procedures are used by law enforcement agencies and forensic professionals to gather evidence, create cases, and solve crimes involving forged papers, signatures, or identities.
- *Identity Theft Prevention:* Forgery detection is crucial in preventing identity theft, which occurs when people's personal information is faked or taken for fraud.

- *Improving Cybersecurity:* Forgery detection methods are used in the cybersecurity arena to identify and prevent many sorts of cyberattacks, such as email spoofing, phishing, and malware that attempt to fool or mimic people.

1.6 Types of Malicious Multimedia Manipulations

This section describes the common manipulation methods in images and videos.

Deepfakes: Image splicing combines parts, objects, or areas from many source pictures into a single composite image. These elements include people, objects, backdrops, and other visual components. Image splicing may be used for various objectives, ranging from artistic inventiveness and photo editing to generating deceiving or misleading images for nefarious goals such as disinformation dissemination or digital forgeries. Deep learning techniques, namely generative adversarial networks (GANs) and deep neural networks, enable deepfakes. GANs comprise two neural networks—a generator and a discriminator—that work in tandem to generate extremely realistic synthetic material. Deepfake technology enables the amazingly accurate modification of faces, sounds, or whole scenarios. This includes modifying facial expressions, swapping faces, adjusting lip-syncing in films, and more. Deepfakes aren't just for visual content. They may also be used to make false audio recordings or voiceovers by synthesising sounds that resemble a certain person's voice.

Splicing*:* Image splicing is a digital image alteration method that combines various bits or aspects from numerous source pictures to generate a new composite image. This method entails cutting or copying portions from one or more source pictures and pasting them into a destination image. Image splicing can be used for legal purposes like image editing and composition or for deceitful purposes like generating misleading or fraudulent visual information. Image splicing combines parts, objects, or areas from many source pictures into a single composite image. These elements include people, objects, backdrops, and other visual components. Image splicing may be used for various objectives, ranging from artistic inventiveness and photo editing to generating deceiving or misleading images for nefarious goals such as disinformation dissemination or digital forgeries.

Copy-Move*:* Copy-move forgery is a digital image forgery or manipulation in which a specific picture section is frequently copied and pasted within the same image to fool viewers or modify the content. This sort of forgery is especially prevalent in the digital arena, where it is used to produce duplicate or cloned objects or pieces inside an image. The duplicated piece

is generally pasted over another image area to hide or reproduce an item or scene, making the original information look intact. A part of the picture is reproduced in a copy-move fake. Copying an object, text, or any other visual element is an example of this. The copied section is then put into another part of the same picture. This frequently entails changing the cloned element's location, orientation, or size. The main purpose of copy-move forgery is to trick people into thinking the edited image is authentic and unaltered. It may be used to conceal or add things, eliminate undesired features, or change the image's composition.

Object Removal: In digital image processing and computer vision, object removal refers to removing or concealing certain objects or areas within an image while keeping the picture's visual coherence and consistency. This approach is extensively used in picture editing, image modification, and computer vision applications for various goals, including improving an image's attractiveness, deleting undesired items, and changing the content. It has applications in a variety of disciplines, but its usage in particular settings necessitates careful evaluation of the ethical implications.

Other Manipulations: Several other manipulations are also possible, including recolouring, resampling, seam carving, inpainting, shadow removal, etc.

Several research contributions have been proposed to counter these common manipulations, such as copy-move detection methods [11], [12], [5] [6], splice detection approaches [3] [4], facial manipulation detection contributions [13] [14], facial retouching detection [15] etc.

This study explores deep learning-based manipulation detection approaches in images and videos. Because of the explosive rise of social media platforms in recent years and the development of harmful manipulation techniques, it is now easier than ever to generate and change multimedia material. Deep learning-based approaches have proven superior to hand-crafted feature-designing methods in computer vision applications.

1.7 Sources of Research Works Studied

This section highlights the approach used to prepare this thesis. This thesis includes research papers from top journals, conferences and workshops of several popular repositories like IEEE Xplore, Science Direct, Springer, ACM and Google Scholar. Relevant publications were included using keyword searches for “forgery detection”, “manipulation detection”, “images”, “videos”, “deep”, “review”, “survey”, etc. High-quality journals such as ACM Transactions, IEEE Transactions and 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), International Conference on Computer Vision (ICCV) were prioritized while including research contributions.

Fig. 2 shows the year-wise distribution of contributions, demonstrating that the major contributions are from recent years.

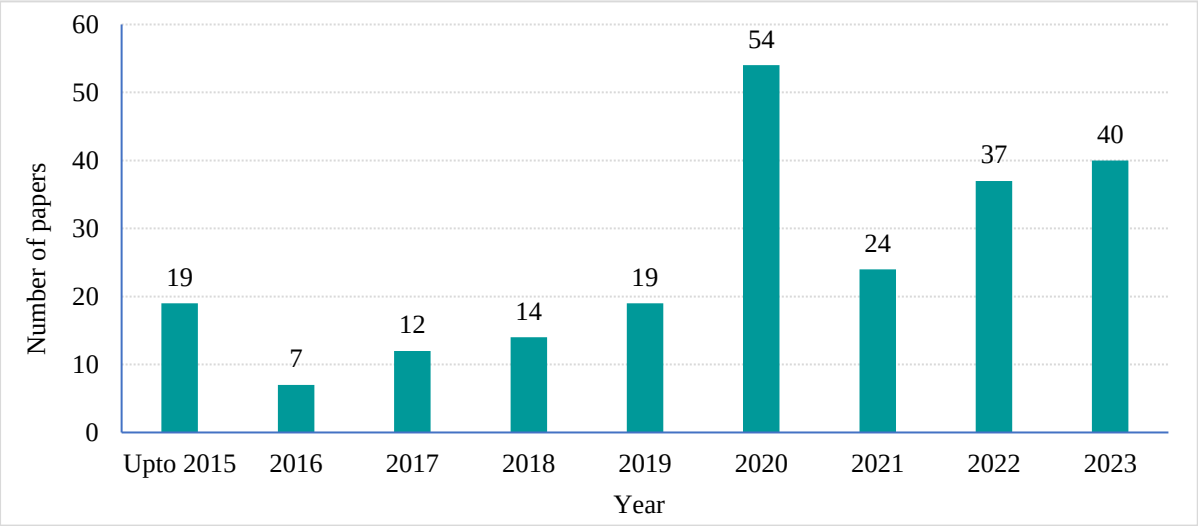
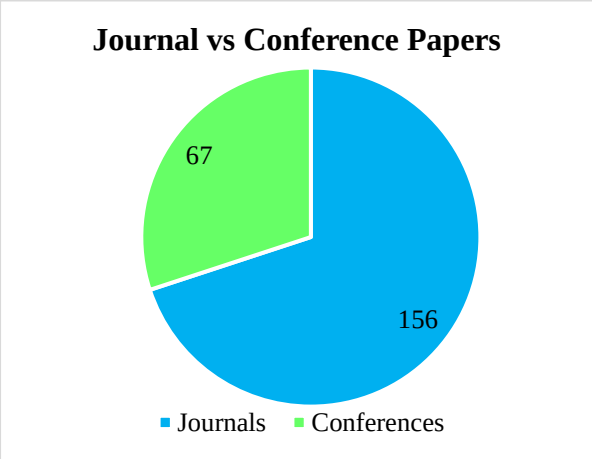
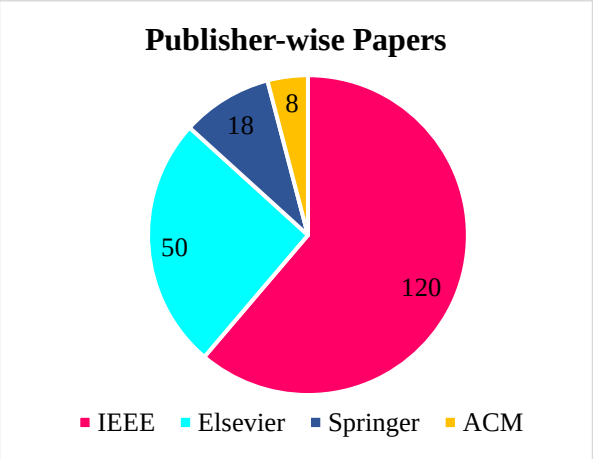


Fig. 2 Year-wise Papers of Manipulation Detection Literature





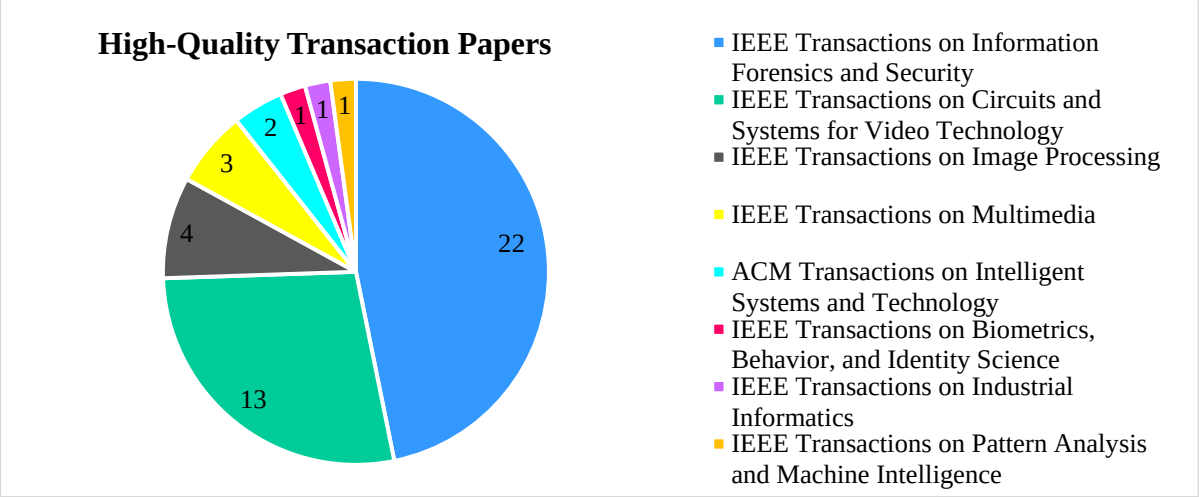


Fig. 3 The distribution of papers discussed in this thesis is presented in the above pie charts. The first graph gives a comparison of the number of conference and journal papers cited. The second graph shows the publisher-wise distribution of papers. The third graph shows the number of high-quality research papers from transaction journals.

Fig. 3 shows the distribution of papers cited in this thesis. The first graph presents the number of conference and journal papers cited in this thesis. Next, the second graph shows the publisher-wise distribution of papers. And last, the third pie chart shows the number of papers from high-quality transaction journals.

1.8 Thesis Overview

This dissertation contains six chapters.

Chapter 2 is dedicated to the review of the literature that highlights the existing state-of-the-art methods for detecting manipulation in multimedia content. Specifically, it presents the research works categorized by the type of manipulations they detect, the research gaps identified, the research objectives targeted and the research contributions made.

Chapter 3 is dedicated to the problem of face manipulation detection. Two novel deep-learning architectures are proposed for this purpose. The first model contains an auto-adaptive weighting mechanism that intelligently chooses the best proportion of manipulation residuals and textural features. The second model contains an intelligent multi-scale attentional module that fuses multi-scale features dynamically to detect facial manipulation in images.

Chapter 4 is dedicated to the problem of splice detection and localization in images. A novel splice detection dataset is proposed by creating spliced samples from Python code and Adobe Photoshop software. A novel light-weight splice detection framework is proposed that extracts discriminative features from spatial and compression domains. Another novel splice localization network is proposed to extract multi-domain features from the input images' RGB,

edge, and depth domains. It also upsamples features using attentional multi-receptive field convolution operation to localize the region of splice forgery in images.

Chapter 5 studies the impact of visual attention mechanisms in face forgery detection. Five recently proposed visual attention models are incorporated in a baseline CNN, and the tradeoff between performance and computational cost is studied.

Chapter 6 presents the conclusion of the research work done in this dissertation and possible future research directions.

Chapter 7 includes the references cited in this thesis.

Chapter 2: Literature Review

This chapter explores the existing literature on the problem of manipulation detection in multimedia content.

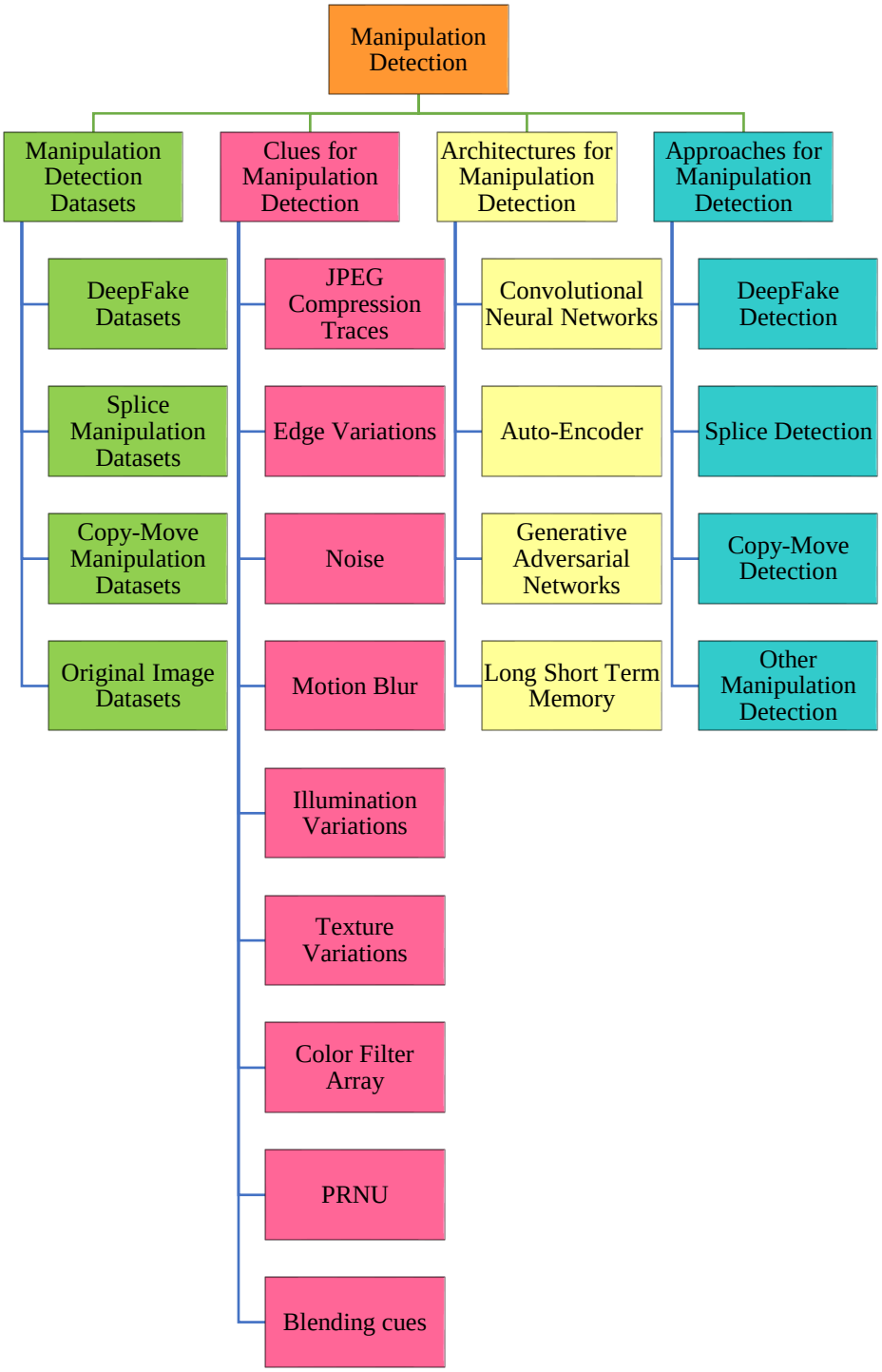


Fig. 4 Taxonomy of Malicious Manipulation Detection in Multimedia

This chapter explores the literature related to detecting malicious manipulation of multimedia data. Specifically, the detection methods have been categorized into DeepFake detection methods, splice detection methods, and copy-move detection methods. Then, the research gaps are elaborated in Section 2.5. Next, Section 2.6 describes the research objectives covered in this dissertation. Finally, Section 2.7 explains the research contributions made towards these research objectives.

2.1 DeepFake Detection Methods

Deepfake is any multimedia content synthesized using an artificially-intelligent approach [2], [16], [17]. Deepfakes are ultra-realistic identity manipulations that cannot be manually differentiated by a human [18], [19]. These manipulations commonly include swapping facial regions, transferring facial pose/expression or synthesizing a complete artificial face [20], [21], [22], [23].

Some contributions have used handcrafted feature-based methods to detect deepfake videos, such as texture analysis from Local Derivative Patterns on Three Orthogonal Planes [24]. While these methods claim to achieve good detection scores, they seriously lack localisation capabilities and require comprehensive manual feature designing, a classic drawback of handcrafted feature methods. The most effective deepfake detection/localization methods are based on deep architectures learning discriminative features automatically using a variety of novelties in input pre-processing, architectures or both.

The most common approach towards deepfake detection is to use visual information from images or video frames as input and employ novel deep architectures to learn discriminative features. Yang et al. [25] extract multi-scale textural features and demonstrate their high relevance in detecting deepfakes. Authors propose a novel “central difference convolution” (CDC) operator to compute texture difference from pixel gradient information. The texture difference is combined at multiple scales using “atrous spatial pyramid pooling” (ASPP). Based on the novel CDC and ASPP, CNN shows strong generalization capability and robustness to distorted test data. Some methods have targeted optical flow to detect deepfakes. Amerini et al. [26] propose to learn inter-frame dissimilarities from optical flow. Guo et al. [27] extract structure forgery clues, dividing faces into strong and weak correlation regions and highlighting potential tampering areas.

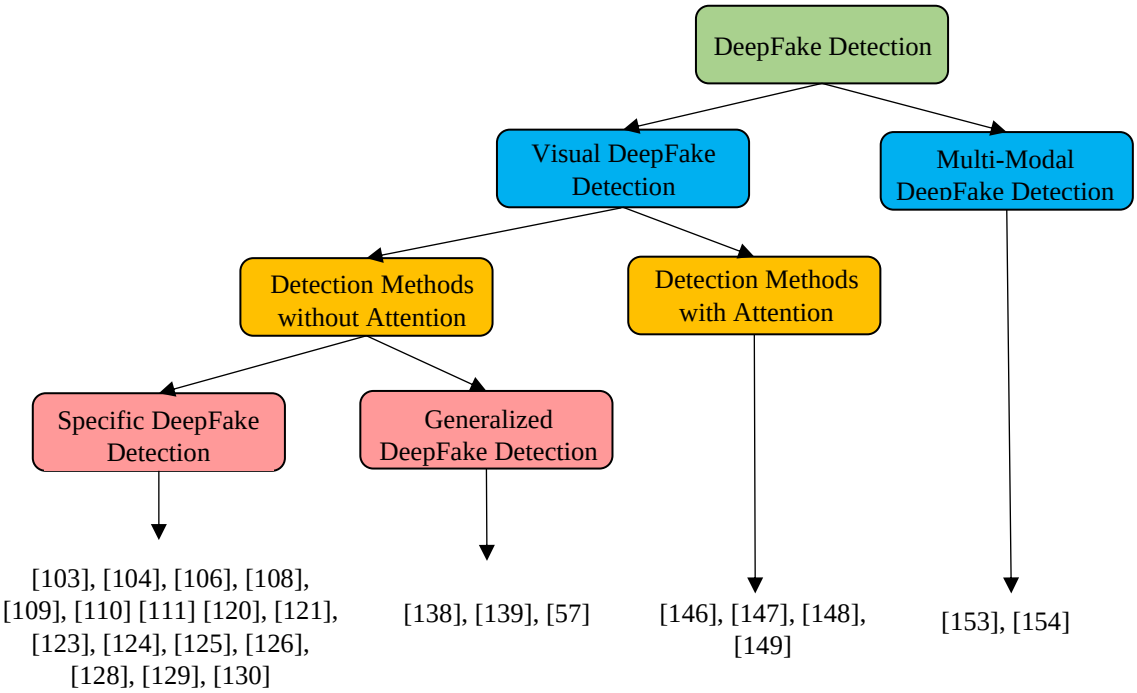


Fig. 5 Categories of DeepFake Detection Methods

VGG16 and ResNet50 architectures are trained on optical flow frames to classify input as original or fake, achieving 81.61% and 75.46% accuracy on the FaceForensics++ dataset. Xu et al. [28] consider faces from different video frames as a set and propose a novel set convolutional neural network that performs multi-frame feature aggregation to detect deepfakes. Kong et al. [29] utilize segmentation and noise maps to detect and localize facial manipulations. Tan et al. [30] propose a novel transformer-based architecture for feature compensation and aggregation, fusing global transformer and local convolutional features and reducing redundant feature learning. Chen et al. [31] use a spatiotemporal attention-based Xception-LSTM architecture for tamper detection. Ganguly et al. [32] utilise a transformer in one branch and Xception CNN in another branch to highlight face tampering artefacts. Pu et al. [33] combine frame-level and video-level inconsistencies to detect facial manipulation. Xia et al. [34] utilize textual statistical disparities between real and fake samples in each color channel and extract discriminative features from the co-occurrence matrix to detect deepfake manipulation. Kingra et al. [35] exploit LBP-based texture differences to detect manipulation.

Several approaches have targeted inconsistencies in biological clues such as visual lip movements [36], [37], [38], [39], eye blinking [40], heartbeat information [41], [42], face context [43] as an indicator for deepfake manipulations. Yang et al. [38] aim for speaker authentication by proposing a novel deep architecture based on novel lip feature representation.

A novel "Fundamental Lip Feature Extraction" (FFE-Net) subnet captures lip motion patterns, reducing the impact of static lip features such as lip shape and appearance. Another novel, "Representative Lip Feature Extraction and Classification" (RC-Net) subnet, captures a person's talking habits by extracting high-level lip features.

Several contributions have been made towards generalizations of deepfake detection. Wang et al. [44] use pixel-wise Gaussian blurring and a novel adversarial training practise to train models on adversarially crafted inputs to boost generalization capability. Korshunov et al. [45] propose to boost generalized deepfake detection by trying several data augmentation techniques, including a novel data farming approach. The authors also demonstrate the effectiveness of a few-shot tuning approaches to achieve the same. Wang et al. [46] prevent a drop in detection performance against compression degradation by training on a siamese network setup that processes input image and its degraded quality equivalent in pairs. In [47], the authors propose a Locality Aware Autoencoder (LAE) that uses a pixel-level mask to learn discriminative features from forged regions instead of finding superficial correlations. Hu et al. [48] use disentangled representation learning (DRL) to separate forgery-relevant information from other non-forgery-based noise features. Ablation study indicates that the disentanglement module plays a significant role in detecting deepfakes.

The recently proposed "attention-mechanism" has greatly enhanced the learning capability of deep models in detecting the manipulation of images/videos [49]. Several novel contributions have used attention to highlight discriminative regions within input that help to refine deepfake localization. Dang et al. [50] improve the binary classification capability of CNN by using an attention mechanism. A novel attention-layer is proposed that takes any high-dimensional CNN feature map $\mathbb{F}$ as input and produces an attention map $\mathbb{M}_{att}$ using a novel "manipulation appearance model" (MAM) and then performing channel-wise multiplication with $\mathbb{F}$ to produce $\widetilde{\mathbb{F}}$. Choi et al. [51] use attention to uncover key video frames that have a high impact on the final prediction score. A certainty-aware attention map is generated that computes the certainty of frame-level prediction from a video, and then certainty-attentive features are generated based on the previously learned attention map to produce a binary classification. Experimentation results suggest that the attention mechanism improves the AUC scores from 0.92 to 0.94 and the accuracy score from 0.89 to 0.92.

While most deepfake detection methods have focused on using visual data, some contributions include multi-modal approaches. Chugh et al. [52] infer that fake videos will have

dissimilarities in their audio and video channels. A two-branch architecture extracts features from visual and audio channels of 1-second videos. The two branches are trained individually on binary cross-entropy loss. The contrastive loss enforces the dissimilarities between audio and visual information of fake samples. A novel "Modality Dissonance Score" (MDS) measures the aggregate dissimilarity of visual-audio modality. Building on a similar idea, Mittal et al. [53] utilize the audio-visual channel and learn perceived emotions from the audio and visual channels to detect deepfakes. Chu et al. [54] extract facial expression representations and lip motion patterns using an Action Unit Transformer and Temporal Convolutional Network, respectively, to predict deepfake manipulation.

2.2 Splice Detection Methods

Splice manipulation involves copying and pasting one image's region(s) onto another. Fundamentally, all splice detection approaches rely on the simple idea that the pasted region and the original region of a spliced sample hold distinct properties, and any competent splice detection framework must highlight this difference. The most common splice detection clues include 1) Noise variations 2) Compression traces 3) Source camera property differences 4) Illumination inconsistencies.

Traditional Splice Detection Methods: Traditional splice detection methods primarily focused on designing handcrafted features that highlight discriminative differences between original and spliced samples. Some methods are based on *image characteristics* such as detecting sharp transitions of edges and corners [55], chroma information [56], etc. Methods based on *source device identification*, such as [57], [58] proved ineffective when the extracted camera signal was weak. Certain *hash-based methods* such as [59], [60] have also been attempted to solve splice manipulation but cannot be regarded as blind splice manipulation detection methods.

Deep Learning-based Splice Detection methods: Deep learning-based approaches for splice detection are divided into two categories, namely, 1) *Deep Spatial Splice Detection Methods* 2) *Deep Hybrid Splice Detection Methods*.

Deep Spatial Splice detection methods directly input pixel information from images/videos and employ architectural novelties to automatically extract discriminative features for manipulation detection and localization of spliced regions [61], [62]. Deep Hybrid Splice detection methods perform automatic feature extraction from a variety of input information,

including spatial information [63], CbCr channels [64], illumination maps [65], resampling features [66], DCT histograms [67], residual features [68], source device patterns [69] etc to obtain the robust classification of manipulated samples. Some approaches combine these distinct inputs with spatial pixel data to obtain higher metric scores [70], [66], [71] etc.

Traditional splice manipulation methods suffer from several drawbacks. *Image Characteristic* methods prove weak if forgery is followed by a post-processing operation. *Source Device Properties* methods fail if the signals extracted are dilute and provide very little discriminative information. *Hash methods* for splice detection require a hash of the original non-forged image, which defeats the purpose of blind splice manipulation detection. *Watermarking* methods like [72] also require original images, which presents the same problem as in the case of hash-based splice detection methods. Another serious drawback of traditional splice manipulation detection methods is that while these methods are able to classify original and spliced samples to some degree, they demonstrate very weak localization ability. Hence, the automatic feature extraction capability of deep learning proves paramount towards accurate splice detection and localization.

Splice manipulation leaves distinct compression artifacts, and several contributions have been targeted to exploit this [73], [70]. Specifically, if an original single-compressed image is spliced and recompressed a second time, the double compression leaves distinct traces. The DCT histograms of doubly compressed images obtain a distinct shape by exhibiting a higher frequency of missing values as compared to histograms from the original single compressed image [74]. Some contributions combine spatial and compression information to detect/localize splice manipulation. In [73], authors train a novel deep model by combining DCT coefficients and uncompressed pixel information for splice detection and beat traditional hand-crafted based splice detection methods. In [70], researchers prove that spatial and DCT compression information prove complimentary in detecting double jpeg compression that indicates splice forgery. A dual-branch deep architecture is trained on spatial and DCT features and attains high accuracy scores (93 to 99% accuracy) for cases when first compression quality (QF_1) is less than second compression quality (QF_2), i.e., $QF_1 < QF_2$. However, these model performances suffer for the case $QF_1 > QF_2$ due to small statistical differences. This is still a persistent research gap that needs to be addressed by upcoming tampering detection models.

Liu et al. [75] propose a fusion of noise and compression information for splice detection. Specifically, the proposed Fusion-Net contains two blocks of the novel DenseNet architecture.

A novel residual loss is proposed that enforces the network to learn forensic features of noise and compression, and a novel discrepancy loss is used to enhance the traces from multiple sources within an image patch. The two novel losses combined with classification loss help the proposed model to achieve 0.97 and 0.90 auc scores on the Columbia and Realistic Tampering datasets, respectively, making it highly robust for splice localization. Another similar method utilizing noise and compression features is proposed in [63]. Since splice manipulations copy an image region onto a different image, the spliced sample contains source device traces of multiple cameras. Several splice detection methods exploit this characteristic by judging if a given image contains patterns from multiple cameras, thereby indicating splice forgery. Bondi et al. [69] use a pre-trained CNN to extract features from non-overlapping image patches and then utilize a clustering algorithm to decide if each patch includes traces from single or multiple cameras. A patch confidence score indicating the contribution of a given patch in finding discriminative source camera information helps the clustering algorithm to choose correct patches for splice detection results, but it contributes little to the localization process. The proposed model achieves 0.91 accuracy for known camera images and 0.81 accuracy for unknown camera categories.

PRNU pattern is a popular source camera characteristic that aids in splice detection. However, estimating PRNU requires a large number of images from a given camera. Also, rich semantic image content interferes with PRNU estimation. Cozzolino et al. [76] propose a novel camera model fingerprint called *noiseprint* that outputs camera residual signals that are much stronger than PRNU. The main novelty of noiseprint lies in the fact that the uncovered camera patterns don't match the entirety of two images from the same camera, but only when the patches are from the same spatial regions within the images since camera artifacts vary spatially within images. Two CNNs with the same architecture and weights are trained to suppress image content and highlight discriminative noise residuals using a *distance-based logistic loss*. The proposed noise residual extraction method achieves splice detection and localization scores. However, experimental results suggest that the extracted pattern is not robust enough to train with uncompressed image data or resize operation on test images, as both scenarios lead to significant drops in performance. Another study combines noiseprint with PRNU for device source identification [77].

Wang et al. [78] propose an architecture based on a novel *weight combination module* that combines YCbCr, Edge and PRNU features in a weighted manner. Four such modules are

connected in serial, and the weight parameters are autotuned with backpropagation. The ablation study reveals that individual PRNU features are more discriminative than YCbCr or Edge features. However, the best results are obtained by a weighted combination of the three, scoring 99.45% accuracy on CASIA v1.0 and 99.32% on CASIA v2.0 for size 64 x 64.

2.3 Copy-Move Detection Methods

Copy Move is one of the most popular types of image tampering, in which a portion of a picture is copied onto one or more parts of the same image.

Traditional Copy-Move Detection Methods: Traditional copy-move manipulation detection methods primarily focused on handcrafted features such as discrete cosine transform (DCT) [79], chroma features [80], discrete wavelet transform (DWT) [81], principle component analysis (PCA) [82], Zernike moments [83], Blur moments [84], Local Binary Pattern (LBP) [85], Oriented Fast and Rotated Brief (ORB) [86], Speeded up Robust Features (SURF) [87], Scale-Invariant Feature Transform (SIFT) [88], Color Filter Array (CFA) [89]. The traditional copy-move detection approaches are categorized as *block-based* and *keypoint-based approaches*. In block-based detection approaches, an image is broken down into overlapping blocks. Then handcrafted features such as DWT, DCT, chroma features, PCA, etc., are extracted for each block, and finally, a block matching algorithm compares the uncovered features from each block. In keypoint algorithms, features are extracted to compare only high-entropy regions within images using local descriptors like SIFT, SURF and ORB.

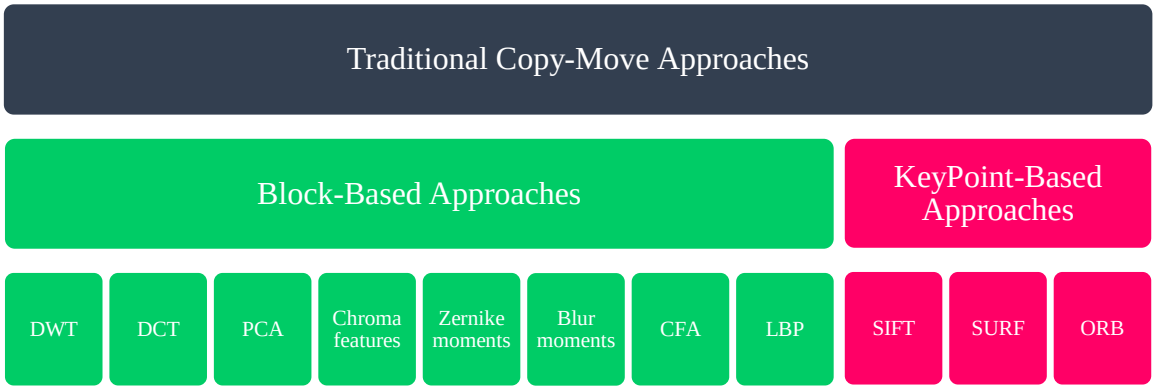


Fig. 6 Traditional Copy-Move Detection Approaches

Though effective, traditional block-based copy-move detection methods are computationally expensive and not robust to geometric transformations such as scaling pasted objects. [5]. Traditional keypoint-based copy-move detection methods are computationally efficient when compared to block-based methods since they avoid an exhaustive comparison

of all overlapping regions within an image and only aim to match extracted keypoint features. However, these methods demonstrate poor localization capabilities and cannot solve the smoothing manipulation snippet [90].

Deep learning-based Copy-Move Detection methods: Copy-move manipulation can be *plain, affine* or *complex* [91]. Plain copy-move comprises a simple copy-paste operation with no transformations and is easy to detect. Affine copy-move includes scaling and rotation transformation before pasting the object. Complex copy-move includes not only affine transformations but also utilizes extra image processing steps such as blending edges of pasted objects and color/brightness enhancements to suppress the manipulation artifacts. Affine copy-move requires advanced tools such as Adobe Photoshop.

BusterNet [91] is the first major end-to-end deep learning architecture to detect and localize copy-move manipulation. It comprises a dual-branch CNN network utilizing the first four VGG16 architectures. A 'manipulation detection' branch predicts regions of possible manipulation. A 'similarity detection' branch is responsible for finding copy-move regions using self-correlation by measuring region-wise similarity and percentile pooling for additional statistical analysis. Pretrained on the ImageNet dataset and fine-tuned on a synthetically prepared dataset with attacks including blending, rotation, scaling and translation, a three-stage training strategy ensures that the branches learn to maximize their feature extraction capability before training the model end-to-end. BusterNet outperforms the then state-of-the-arts, achieving a high image level auc score of 0.8 on the CASIA dataset. It proves robust against most attacks or postprocessing methods of the CoMoFoD dataset.

One key challenge in copy-move manipulation detection is identifying and distinguishing between original image regions with similar textural data and copy-move manipulated regions since both cases have identical visual information. Islam et al. [6] try to solve this problem by using a dual-attention-based architecture. The authors compute an *affinity matrix* with second-order statistics on features extracted from a CNN. Then, a first-order attention module highlights all similar regions within an image, and a second-order attention module separates similar-looking original regions from copy-moved regions. High values in off-diagonal elements indicate copy-move forgery. The proposed method is designed for adversarial training where a generator produces a copy-move forgery mask, and a discriminator is trained to differentiate generated masks from actual ground truths.

Zhu et al. [5] propose a novel *Adaptive Attention and Residual Refinement Network* (AR-Net) that utilizes positional and channel attention to highlight discriminative parts of features. Deep matching is used to learn self-correlation among feature maps, and atrous spatial pyramid pooling is used to obtain multi-scale features. Zhong et al. [90] propose the Dense InceptionNet network having a pyramid feature extractor (PFE) to extract multi-dimensional and multi-scale features, feature correlation matching (FCM) to learn the correlation of dense features and hierarchical post-processing (HPP) to improve training through a combination of entropies.

2.4 Other Manipulation Detection Methods

While copy move, splicing and facial tampering are the most common forms of manipulations, several other types of manipulation detection and localization approaches have also been proposed. Some of these manipulations are discussed below.

Nam et al. [92] tackle *seam carving* by proposing an ILFNet architecture containing five blocks to detect local artefacts caused by seam insertion or removal operation. Li et al. [93] handle *inpainting* manipulation through a C-based architecture with four ResNet blocks trained on image residuals to localize the inpainting region. Yan et al. [94] approach recolouring detection using a CNN with three feature extraction blocks and one feature fusion block. To identify recoloring, the picture is used as input, along with illumination consistency and inter-channel correlation Yarlagadda et al. [95] take the issue of *shadow removal detection* by training a cGAN to output localization mask of shadow removal region. Long et al. [96] perform *frame deletion detection* in videos by using 3D convolutions in the network that threshold the L2 distance of color histograms, optical flow and motion energy of two consecutive frames to detect deleted frames.

2.5 Research Gaps

On the basis of the literature presented in the above section, several research gaps have been identified.

- Several multi-branch approaches have proven highly effective for face manipulation detection. While multi-branch architectures are well suited to learning complementary features from multiple domains, they usually suffer from a common flaw at the feature fusion stage. The domain features are always fused *equally* in terms of proportion. This may not be true in most cases, as there may be more discriminative information in any

one of the domain features. Intelligent deep learning architectures that are capable of automatically choosing the best proportion of multi-domain features need to be designed.

- Recent contributions have focussed on extracting multi-scale features that uncover more discriminative information than single-scale feature extraction networks for face manipulation detection [97, 98, 99]. One key issue with the existing multi-scale feature extraction methods is the implicit assumption that each scale of features holds equal importance and is directly fused for the final prediction. However, this may not always be true as different scales of features may carry different levels of important information crucial for identifying face tampering. A better approach is to develop a dynamic weighting scheme that allows the model to weigh each scale of feature based on their relevance before making the final prediction.
- The size of splice detection datasets is very small for image splice detection, and there is an important need to develop larger splice detection datasets.
- Given the small size of splice detection datasets, there is a need to develop lightweight models for splice detection to prevent the problem of overfitting.
- Visual attention has played a key role in boosting the performance of deep learning models in various computer vision tasks. However, these attention mechanisms come at the expense of added computational costs. No research studies the tradeoff between performance and computational cost while utilizing these visual attention mechanisms in manipulation detection models.

2.6 Research Objectives

The research objectives for this thesis are:

- To develop a manipulation detection dataset with labelled samples (original/forged) and binary masks to train the state-of-the-art deep learning models for robust and accurate manipulation detection in multimedia information.
- To conduct a comparative performance analysis of several state-of-the-art methods for fraudulent multimedia detection on the proposed dataset.
- To propose novel manipulation detection approaches achieving high robustness against the most competent and prevailing approaches for creating fraudulent images or videos.
- To design manipulation localization approaches that perform detection and locate the regions of manipulation within data.

2.7 Research Contributions

The following research contributions have been made in this research thesis:

- Proposed *MRT-Net*, a novel end-to-end architecture for facial manipulation detection in deepfake videos that utilizes an auto-adaptive weighting mechanism of manipulation residuals and textural information to find the best proportion of the two features. The proposed model uses a recently proposed attention mechanism that aggregates features along two dimensions to capture long-range dependencies and accurate positional information. MRT-Net achieves high AUC scores of 0.9964 on DFDC, 0.9921 on CelebDF, 0.9910 on FF++(DeepFake), 0.9974 on FF++(Face2Face), 0.9942 on FF++(FaceShifter), 0.9933 on FF++(FaceSwap) and 0.9662 on FF++(NeuralTextures) datasets beating several state-of-the-art methods.
- Proposed *Face-NeSt*, a novel deepfake detection architecture that dynamically chooses a suitable proportion of multi-scale features to identify face manipulation. Face-NeSt contains a novel 'adaptively weighted multi-scale attentional' module that weighs multi-scale features according to their relevance before combining them for the final prediction. Four auto-adaptive β weight parameters are added to the computation graph of the proposed Face-NeSt model and help to dynamically control the proportion of multi-scale attentional features used in making the final prediction. The attention mechanism highlights important local and global feature regions across the channel and spatial dimensions. The AUC scores are 0.9823 on CelebDF, 0.9947 on DFDC, 0.9945 on DeepFake (FF++), 0.9905 on Face2Face (FF++), 0.9978 on FaceShifter (FF++), 0.9948 on FaceSwap (FF++) and 0.9548 on NeuralTextures (FF++) beating all state-of-the-arts.
- Proposed a novel splice detection dataset – *BiometricLab-DTU-Splice Dataset* is proposed. The proposed dataset has two variants. The first variant is autogenerated from code, while the second contains handmade spliced samples. Binary masks are available in both variants. A novel lightweight, dual-branch, information-preserving, spatial-compression modal splice detection framework is proposed to detect spliced jpeg images while restricting the computational complexity to a small fraction of the usual computational cost in deep learning. The proposed model contains a novel 'spatial branch' to extract discriminative spatial information for detecting image splicing. Transfer learning is used to leverage the strong classification capabilities of deep

models in the spatial domain at minimal computational costs. The proposed model contains a novel information-preserving 'compression branch' that uses original resolution compression data to extract double compression artifacts from spliced jpgs. Experimentations with several variants of the proposed spliced detection framework and comparisons with existing splice detection methods prove the potency of the proposed splice detection framework at minimum computational costs.

- Proposed a novel, visually-attentive splice localization model with multi-domain feature extractor and multi-receptive field upsampler. Specifically the "visually attentive multi-domain feature extractor" (VA-MDFE) extracts attentional features from the RGB, edge and depth domain of input images. Next, a "visually attentive downsampler" (VA-DS) is responsible for fusing the multi-domiain feature and downsampling them. Lastly, a "visually attentive multi-receptive field upsampler" (VA-MRFU) upsamples features using multiple receptive fields during the convolution operation. Experimental results clearly indicate the superiority on the proposed splice localization model against the existing state-of-the-art methods.
- Conducted an exhaustive study of recent visual attention models and demonstrated their effectiveness in detecting face forgery. Five attention models, namely, Coordinate Attention [100], Selective Kernel Attention [101], Triplet Attention [102], CoT Attention [103] and Shuffle Attention [104] have been studied. The attention modules are evaluated by measuring their performance on the popular face forgery dataset FaceForensics++. A study of the computational complexity of each type of visual attention has been conducted. The tradeoff between performance and computational cost associated with each attention mechanism is presented.

Chapter 3: Face Manipulation Detection in Images

3.1 Scope of this Chapter

This chapter is dedicated to the problem of face manipulation detection in images. To this end, two novel deep-learning architectures are proposed. The first model is *MRT-Net*, which uses an auto-adaptive weighting mechanism of manipulation residuals and textural information to find the best proportion of the two features crucial to detecting face forgery. MRT-Net is an end-to-end, multi-domain architecture that utilizes channel attention to capture long-range dependencies along the depth. The second model is *Face-NeSt*, a visual attention-based, multi-scale deepfake detection architecture that extracts visually attentive multi-scale features to detect face manipulation. Four auto-adaptive β parameters are added to the computation graph of Face-NeSt and help to dynamically control the proportion of multi-scale attentional features used in making the final prediction. The attention mechanism highlights important local and global feature regions across the channel and spatial dimensions. Experimental results on three public benchmark datasets for face manipulation detection prove that both MRT-Net and Face-NeSt models beat the existing state-of-the-art models comfortably, clearly establishing their superiority.

3.2 MRT-Net: Auto-Adaptive Weighting of Manipulation Residuals and Texture Clues for Face Manipulation Detection

3.2.1 Abstract

Due to the increasing prevalence of social media and the proliferation of deceptive manipulation techniques, it is now more effortless than ever to deceive and disrupt society by altering information on social media platforms. Therefore, it is imperative to develop robust manipulation detection systems. This publication presents an innovative architecture, MRT-Net, that extracts distinct "manipulation residuals" (MR) and "textural" (T) characteristics for the purpose of detecting face manipulation. The fusion step of most multi-branch designs commonly exhibits a fault, wherein it combines multi-domain properties in equal proportion. This proves detrimental as not all features may hold equal significance in determining the final prediction. MRT-Net addresses this issue by including an auto-adaptive weighting technique to determine the optimal balance of manipulation residual and textural information, which mutually enhance each other. More precisely, the proposed neural network incorporates two

weighting factors, denoted as $\propto_1$ and $\propto_2$, for the MR and T features. These parameters are dynamically updated by backpropagation, enabling MRT-Net to discover the optimal combination of residual and textural information. In addition, MRT-Net utilises a channel attention method to enhance its performance to a greater extent. MRT-Net demonstrates exceptional performance on three widely recognised benchmark datasets, namely Deep Fake Detection Challenge (DFDC), CelebDF, and FaceForensics++ (FF++). The AUC scores obtained are as follows: 0.9964 for DFDC, 0.9921 for CelebDF, 0.9910 for FF++(DeepFake), 0.9974 for FF++(Face2Face), 0.9942 for FF++(FaceShifter), 0.9933 for FF++(FaceSwap), and 0.9662 for FF++(NeuralTextures). The model also attains accuracy scores of 0.9760 on DFDC, 0.9815 on CelebDF, 0.9670 on FF++(DeepFake), 0.9767 on FF++(Face2Face), 0.9611 on FF++(FaceShifter), 0.9676 on FF++(FaceSwap), and 0.9025 on FF++(NeuralTextures). The outstanding outcomes clearly showcase the effectiveness of MRT-Net, as it surpasses other cutting-edge techniques for detecting face tampering with ease.

3.2.2 Proposed Methodology

This section presents an in-depth description of MRT-Net, highlighting its core components, explaining their role in the overall detection process and how they are integrated. Fig. 8 shows the architecture of the proposed model.

3.2.2.1 MRT-Net Overview

This section provides a visual overview of the working of the proposed model. Specifically, the the algorithm and the flowchart are presented here. Algorithm 1 presents the pseudo-code for training MRT-Net.

Algorithm 1 Pseudocode for training the proposed MRT-Net

Input:

$Dataset = \{\mathbb{X}_i, \mathbb{Y}_i\}_{i=1}^{n}$ such that $\mathbb{X}_i \in \mathbb{R}^{3\times128\times128}$ represents face images and $\mathbb{Y}_i \in \{0,1\}$ is the true label.
Model parameters θ
Batch Size $\mathcal{B}$
Total Batches $\mathscr{b}_{total}$
Epoch $\mathcal{E}$
Initial Learning Rate ℓr
Learning Rate Decay factor γ after every n epochs. $\gamma \in [0,1]$
Auto-Adaptive weights $\in \{\alpha_1, \alpha_2\}$

Output:

Trained MRTNet model for face manipulation detection

1.	Initialize θ and auto-adaptive weights $\{\alpha_1, \alpha_2\}$	
2.	**for** $e = 1,2,3 \ldots \mathcal{E}$ **do**	Train for $\mathcal{E}$ epochs.
3.	**for** $\mathscr{b} = 1,2,3 \ldots \mathscr{b}_{total}$ **do**	Loop through all batches in an epoch.
4.	$(\mathbb{X}, \mathbb{Y}) \sim \mathcal{S}$	Randomly sample one batch of size $\mathcal{B}$.
5.	$\mathbb{Y}_{MR} \leftarrow \mathcal{MR}(\mathbb{X})$	Extract residual features $\mathbb{Y}_{MR}$ from manipulation residual branch $\mathcal{MR}()$.

6.	$\mathbb{Y}_T \leftarrow \mathcal{T}(\mathbb{X})$	Extract textural features $\mathbb{Y}_T$ from texture branch $\mathcal{T}()$.
7.	$\mathbb{Y}_{adaptive} \leftarrow (\alpha_1 \times \mathbb{Y}_{MR}) \oplus (\alpha_2 \times \mathbb{Y}_T)$	Auto-Adaptive Weighting of features $\mathbb{Y}_{adaptive}$. $\oplus$ denotes concatenation.
8.	$\mathbb{Y}_{final} \leftarrow \mathcal{L}(\mathbb{Y}_{adaptive})$	Get the final prediction of the model $\mathbb{Y}_{final}$.
9.	$\theta \leftarrow \theta - \ell r \triangle_\theta \mathcal{L}_{CE}(\mathbb{Y}_{final}, \mathbb{Y})$	Update model parameters to minimize cross-entropy loss $\mathcal{L}_{CE}$ via backpropagation.
10.	$\alpha_1 \leftarrow \alpha_1 - \ell r \triangle_{\alpha_1} \mathcal{L}_{CE}(\mathbb{Y}_{final}, \mathbb{Y})$	Update α_1 and α_2 through backpropagation (auto-adaptive).
11.	$\alpha_2 \leftarrow \alpha_2 - \ell r \triangle_{\alpha_2} \mathcal{L}_{CE}(\mathbb{Y}_{final}, \mathbb{Y})$	
12.	**if** $e \,\%\, n = 0$ **then**	Decay learning rate after every 'n' epochs.
13.	$\ell r \leftarrow \ell r \times \gamma$	
14.	**end for**	
15.	**end for**	

Fig. 7 shows the flowchart of the proposed MRT-Net model.

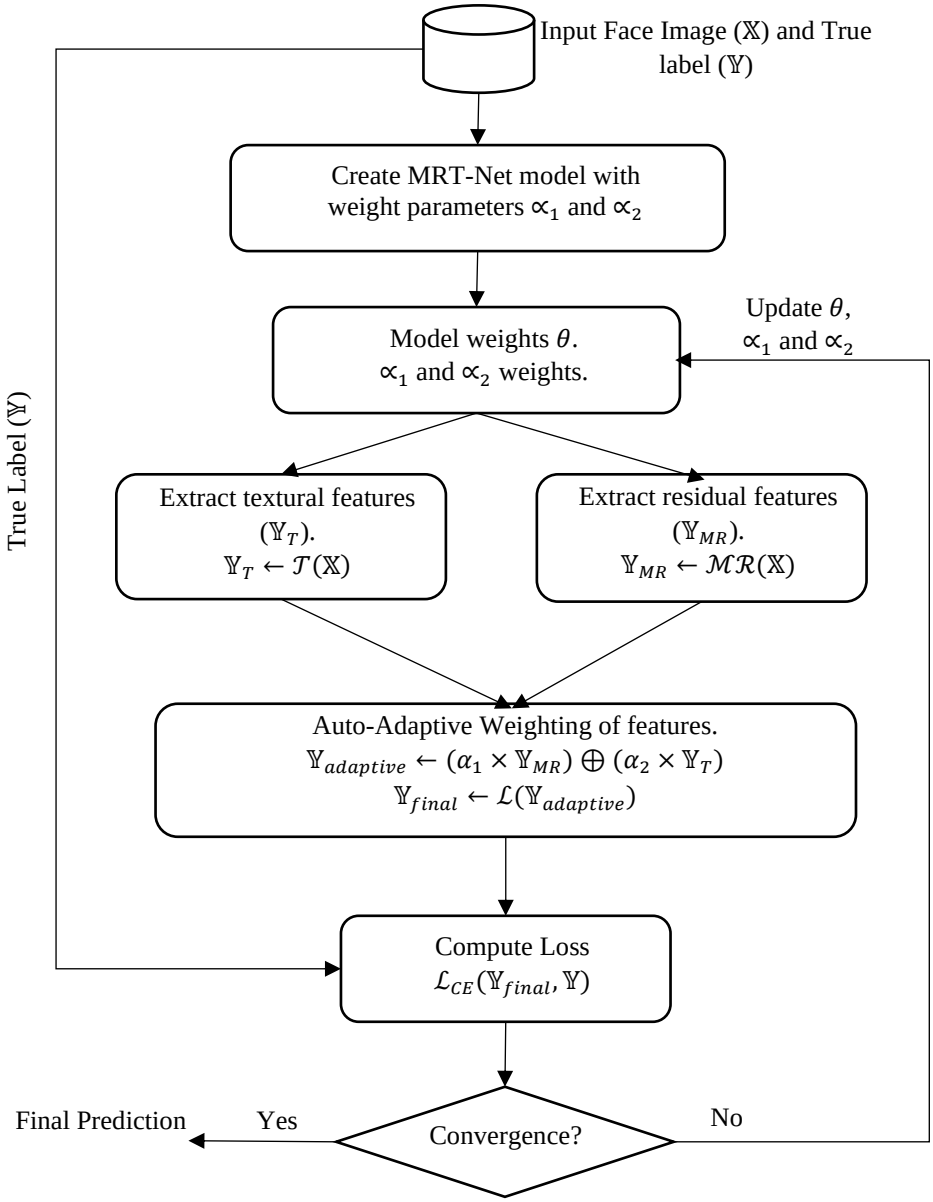


Fig. 7 Flowchart of the MRT-Net model

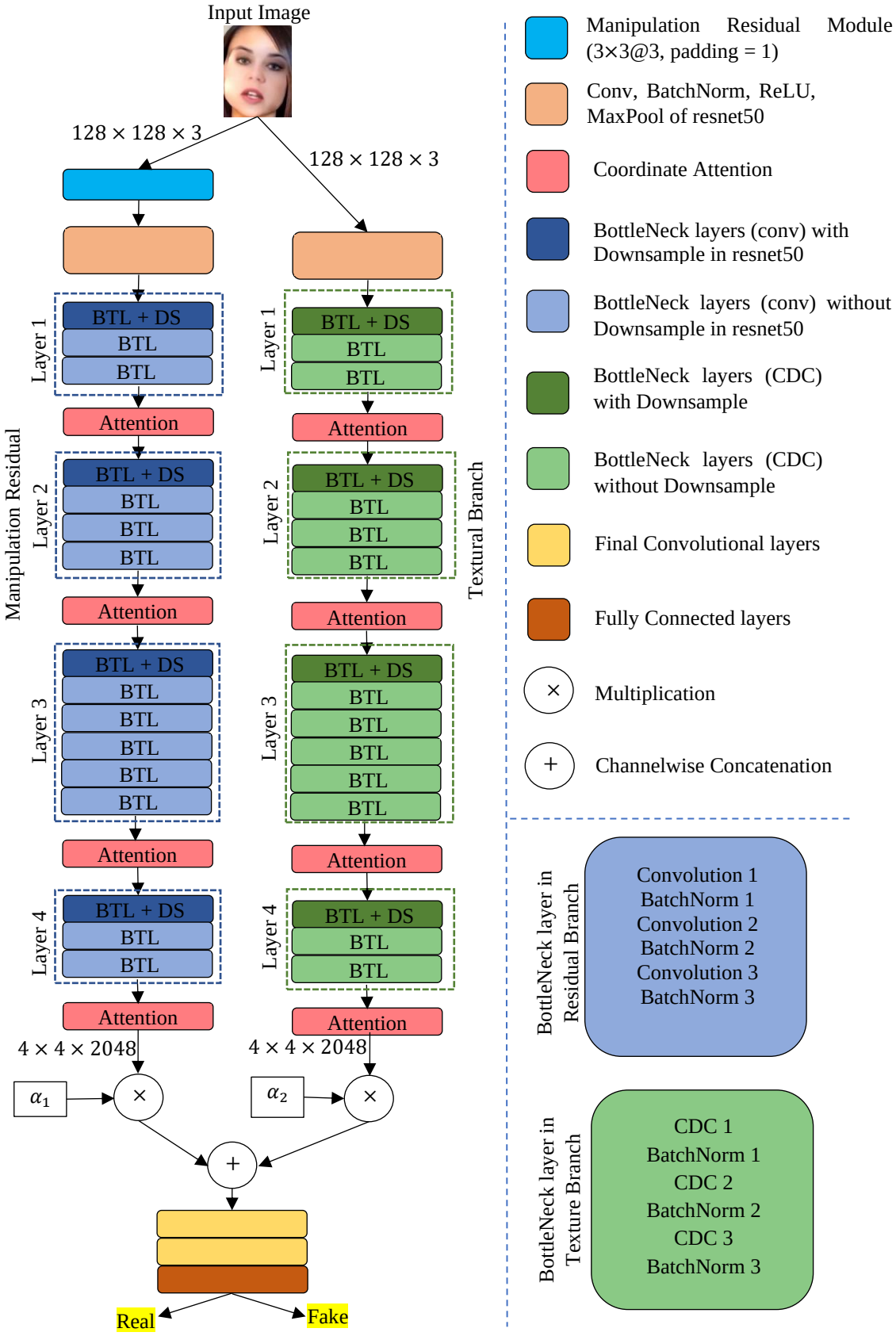


Fig. 8 Architecture of MRT-Net having manipulation residual and textural branch. BTL stands for BottleNeck layers, DS stands for Down-Sample layers and CDC stands for Central Difference Convolution.

3.2.2.2 Manipulation Residual Branch

Detecting manipulation in images requires uncovering manipulation traces. The semantic content of an image plays little role in this process and hence, it makes intuitive sense to suppress such semantic content. This approach allows deep neural nets to extract relevant features from manipulation residuals.

Several research contributions such as [105, 106, 107], aim to learn discriminative features from predicted manipulation traces. A special predictor operator $\emptyset(\cdot)$ predicts pixel values $\hbar_{output}$ from input $\hbar_{input}$ as shown in Eq. 1. $\hbar_{output}$ represents the features focussing on the semantic content of the input image. The manipulation residual can be computed by removing the semantic features $\hbar_{output}$ from the input image. Hence, the manipulation residual $\hbar_{manipulated}$ is computed by the operation described in Eq. 2.

$$\hbar_{output} = \emptyset(\hbar_{input}) \tag{1}$$

$$\hbar_{manipulated} = \hbar_{output} - \hbar_{input} \tag{2}$$

In the proposed model, the predictor $\emptyset(\cdot)$ has been designed as a single convolutional layer. Since deep models have convolutional kernels that are auto-adaptive, they learn the best kernel coefficient values via backpropagation, ensuring an iteratively improving manipulation residual extractor. The manipulation residual extractor convolution layer has three kernels of size 3×3 with padding value 1 to ensure same dimensions for $\hbar_{output}$ and $\hbar_{input}$ and is placed at the beginning of the manipulation residual branch as shown in Fig. 8.

The manipulation residual module is followed by a resnet50 architecture whose pooling and fully connected layers are trimmed. The resnet50 architecture contains blocks of three, four, six and three bottleneck layers respectively, and one attention module is added at the end of each bottleneck layer block. All kernels are initialized with pre-trained ImageNet weights except the manipulation residual extraction module. This approach significantly boosts model performance as demonstrated in the experiment section.

3.2.2.3 Textural Branch

Texture has proven to be a beneficial modality for facial manipulation detection [25, 24, 108]. Hence, the second branch of the proposed model extracts texture information from a facial image. Intuitively, this proves complementary to feature learning from the first branch which aims to suppress the semantic content of the input image. The idea of texture extraction

is derived from [109] by utilizing a novel operator called *central difference convolution* (CDC), as demonstrated in Fig. 9.

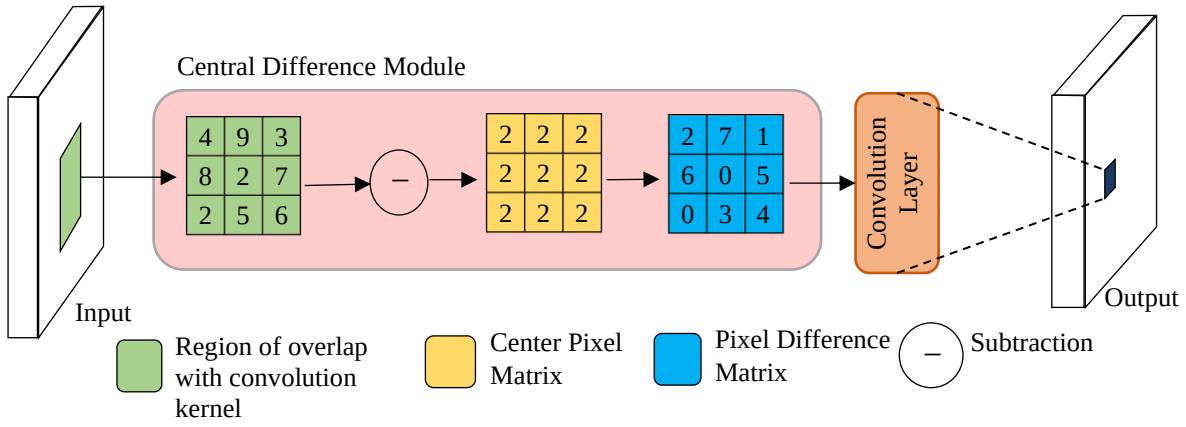


Fig. 9 Central Difference Convolution (CDC) [109]

Eq. 3 describes the standard vanilla convolution operation where the output feature map y is obtained for the current location ℓ_c. Specifically, a convolutional kernel is placed onto the input image with its center value overlapping with the input image at ℓ_c. ℓ_n denotes locations in the local receptive field $\mathcal{F}$ and is is used to calculated a weighted sum of the input pixel values with the corresponding weights in the convolution kernel. $\mathbb{x}$ represents the input matrix and $\mathbb{w}$ are the convolutional weights.

$$y(\ell_c) = \sum_{\ell_n \in \mathcal{F}} \mathbb{w}(\ell_n) \cdot \mathbb{x}(\ell_c + \ell_n) \tag{3}$$

$$y(\ell_c) = \varphi \left(\underbrace{\sum_{\ell_n \in \mathcal{F}} \mathbb{w}(\ell_n) \cdot (\mathbb{x}(\ell_c + \ell_n) - \mathbb{x}(\ell_c))}_{CDC} \right) + (1 - \varphi) \left(\underbrace{\sum_{\ell_n \in \mathcal{F}} \mathbb{w}(\ell_n) \cdot \mathbb{x}(\ell_c + \ell_n)}_{vanilla\ convolution} \right) \tag{4}$$

Eq. 4 describes the CDC operation which is a modified version of the vanilla convolution. When the CDC kernel is placed onto the input image, instead of taking the weighted sum of input pixels and convolutional kernel weights directly, a center matrix is computed by copying the center value from the region of overlap of the input image. This center value is then subtracted from each input pixel value which is then convolved with the kernel weights. The subtracted pixel values are represented by the $\mathbb{x}(\ell_c + \ell_n) - \mathbb{x}(\ell_c)$ term in equation 4. φ is a hyperparameter in the range [0,1], specifying the degree of tradeoff between intensity and gradient information [109]. A higher value of φ specifies more emphasis on extracting texture difference information.

The texture branch is created by replacing all convolution kernels of resnet50 architecture with CDC except in the downsample layers. The pooling and fully connected layers are pruned, just like in the manipulation residual branch. Weights are initialized randomly in this branch.

3.2.2.4 Attention Module

Several attention-based research methods have established the importance of the attention mechanism in facial manipulation detection [50, 51, 110]. A recently proposed *Coordinate Attention* [100] mechanism has been used in the proposed model to boost the classification capability. Coordinate attention employs a lightweight computation model explicitly designed for mobile networks. It maps channel attention from two direction-aware features, learning long-range dependencies and precise positional information. Fig. 10 demonstrates the structure of coordinate attention. The proposed model contains two resnet50 branches. A single coordinate attention block has been employed after each bottleneck layer block in both branches as shown in Fig. 8.

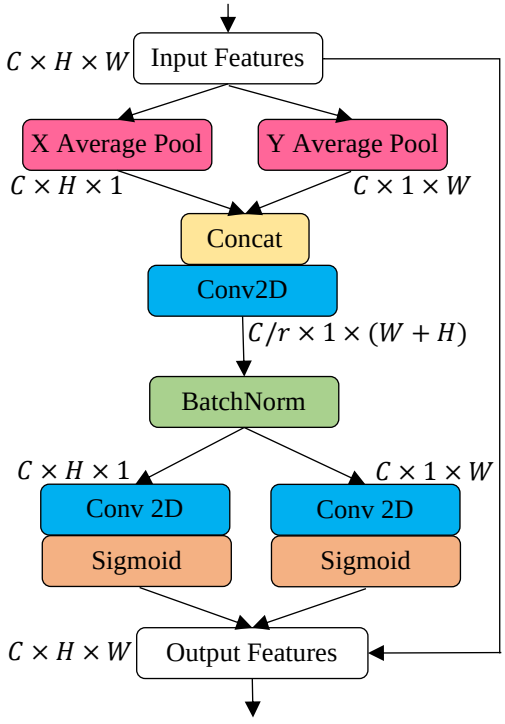


Fig. 10 Structure of Coordinate Attention [100]

3.2.2.5 Auto-Adaptive Weighted Fusion

The auto-adaptive weighted fusion is the main novelty of the proposed model. Unlike other multi-branch approaches that directly fuse multi-domain features in equal proportion, this mechanism allows MRT-Net to choose an ideal proportion for feature mixing. Specifically,

MRT-Net can decide what proportion of manipulation residual and textural information is best suited for detecting facial manipulation.

The proposed model contains two branches based on resnet50 architecture. The shape of output features from both branches is $4 \times 4 \times 2048$. Two weight parameters namely, $\propto_1$ and $\propto_2$ are added to introduce a weighted fusion of the features $\mathbb{f}_1$ and $\mathbb{f}_2$ from manipulation residual and texture branch, respectively. The fused features $\mathbb{f}_3$ is obtained by Eq. 5:

$$\mathbb{f}_3 = (\propto_1 \times \mathbb{f}_1) \oplus (\propto_2 \times \mathbb{f}_2) \tag{5}$$

Here, $\times$ represents the arithmetic multiplication and $\oplus$ is channel-wise concatenation operation. Other fusion techniques are explored in the ablation study. Specifically, sum and mean fusion approaches were also tried. However, channel-wise concatenation yielded the best results. Fused feature $\mathbb{f}_3$ having 4096 channels is then passed through two convolutional layers to reduce the dimensionality of features.

The main novelty of the proposed model is the auto-adaptive updating of $\propto_1$ and $\propto_2$. These two weights are added to the computation graph of MRT-Net. This means that they are added to the weight parameters of MRT-Net. Hence, they are updated automatically via backpropagation and the model finds the ideal proportion of feature fusion in the training phase. To experiment with different proportions of manipulation residual and textural features, $\propto_1$ and $\propto_2$ are initialized with different value combinations, as mentioned in Table VIII.

3.2.3 Experimental Setup

This section presents the experimental settings used for the training and evaluation of the proposed MRT-Net model.

3.2.3.1 Datasets

FaceForensics++ (FF++): The FaceForensics++ (FF++) [111] dataset is a widely used collection of data for detecting face manipulations. It includes several types of manipulations such as Deepfakes [112], FaceSwap [113], Face2Face [114], FaceShifter [115], and Neural Textures [116]. The dataset has a total of 1000 authentic videos and 5000 altered videos, encompassing all five modification categories. The dataset comprises samples of three distinct qualities: raw, high (c23), and poor quality (c40).

The Deepfake Detection Challenge (DFDC): The DFDC dataset [117] comprises 5214 videos generated by two unknown modification algorithms including 66 people. The ratio of manipulated samples to original samples is 1:0.28.

Celeb-DF: The Celeb-DF [118] is a large dataset of deepfakes. The dataset comprises 590 authentic video samples captured from 59 performers, with 5639 deepfake videos. The videos that have been tampered with in this collection exhibit a remarkably authentic level of alteration, which poses a challenge for identification. Videos are recorded at a frame rate of 30 frames per second (fps) and have an average duration of 13 seconds.

3.2.3.2 Train, Validation and Test Splits

This section defines the allocation of the dataset into distinct subsets for the purposes of training, validation, and testing of the proposed model. The suggested model was trained, validated, and tested using the number of face images shown in Table I.

Table I Details of train, validation and test split.

Dataset	Train Split	Validation Split	Test Split
Deepfakes (FF++)	48000	6400	9600
Face2Face (FF++)	48000	6400	9600
FaceShifter (FF++)	48000	6400	9600
FaceSwap (FF++)	48000	6400	9600
NeuralTextures (FF++)	48000	6400	9600
Celeb DF	160000	26432	22400
DFDC	112000	22400	27424

3.2.3.3 Evaluation Metrics

This section defines the evaluation metrics employed to quantify the classification proficiency of the proposed model, as seen in Table II.

Table II Classification metrics used in this experiment.

Metric	Formula	Range	Acronym
Accuracy	$\frac{\mathbb{TP}+\mathbb{TN}}{\mathbb{TP}+\mathbb{TN}+\mathbb{FP}+\mathbb{FN}}$	[0,1]	ACC
Precision	$\frac{\mathbb{TP}}{\mathbb{TP}+\mathbb{FP}}$	[0,1]	P
Recall	$\frac{\mathbb{TP}}{\mathbb{TP}+\mathbb{FN}}$	[0,1]	R
F1 score	$2 * \frac{Precision * Recall}{Precision + Recall}$	[0,1]	F1
Area Under Curve	-	[0,1]	AUC
Mathews Correlation Coefficient	$\frac{\mathbb{TP} * \mathbb{TN} - \mathbb{FP} * \mathbb{FN}}{\sqrt{(\mathbb{TP}+\mathbb{FP})(\mathbb{TP}+\mathbb{FN})(\mathbb{TN}+\mathbb{FP})(\mathbb{TN}+\mathbb{FN})}}$	[-1,1]	MCC

3.2.3.4 Face Extraction

RetinaFace [119] has been used to extract face images from video frames. It is superior to other face extraction models such as dlib [24, 28, 120] or MTCNN [121, 110, 122].

3.2.3.5 Preprocessing and Data Augmentation

In this experiment, each facial image has been resized to dimensions of 128×128. The pixel values are scaled to fit inside the range of [0,1]. Random horizontal and vertical flip augmentations are applied to the face images.

3.2.3.6 Training Settings & Hardware

The initial learning rate is set to 0.01. 32 consecutive frames are sampled from each video. The batch size is set to 128 face images. Face images are shuffled within a batch for the randomness of the input sequence. SGD optimizer is used to update the weights of the neural network. A linear learning rate scheduler decays the learning rate by 10% after every 2 epochs. All training operations are run for 50 epochs. All experiments are run on two 24 GB NVIDIA TITAN RTX GPUs running in parallel.

3.2.4 Experimental Results & Analysis

This section presents the experimental results achieved by the proposed MRT-Net model.

3.2.4.1 Performance of MRT-Net on Benchmark Datasets

This section presents the accuracy (ACC), precision (P), recall (R), F1, AUC and MCC scores of MRT-Net on the FF++, DFDC and CelebDF datasets.

Table III Performance of MRT-Net on FF++, CelebDF and DFDC datasets.

Datasets	**Alpha Initial**		**Alpha Final**		**ACC**	**P**	**R**	**F1**	**AUC**	**MCC**
	$\propto_1$	$\propto_2$	$\propto_1$	$\propto_2$						
Deepfakes (FF++)	0.75	0.25	0.7890	0.2110	0.9670	0.9867	0.9504	0.9682	0.9910	0.9347
Face2Face (FF++)	0.75	0.25	0.8150	0.1850	0.9767	0.9704	0.9857	0.9780	0.9974	0.9534
FaceShifter (FF++)	0.75	0.25	0.8147	0.1853	0.9611	0.9589	0.9627	0.9608	0.9942	0.9223
FaceSwap (FF++)	0.75	0.25	0.8348	0.1652	0.9676	0.9671	0.9697	0.9684	0.9933	0.9353
NeuralTextures (FF++)	0.75	0.25	0.8160	0.1840	0.9025	0.8689	0.9232	0.8952	0.9662	0.8056
Celeb DF	0.75	0.25	0.7550	0.2450	0.9815	0.9833	0.9959	0.9896	0.9921	0.9084
DFDC	0.75	0.25	0.7920	0.2080	0.9760	0.9805	0.9891	0.9848	0.9964	0.9279

Table III displays the classification scores achieved by MRT-Net. The model scores above 0.96 accuracy for most cases signifying excellent classification capability. MRT-Net achieves high F1 scores of 0.96 and above for most cases, measuring the balance between precision and recall. MRT-Net also achieves excellent AUC scores of 0.99 and above for most cases as shown in Fig. 11. This proves that MRT-Net makes high-confidence predictions and can clearly identify manipulated samples.

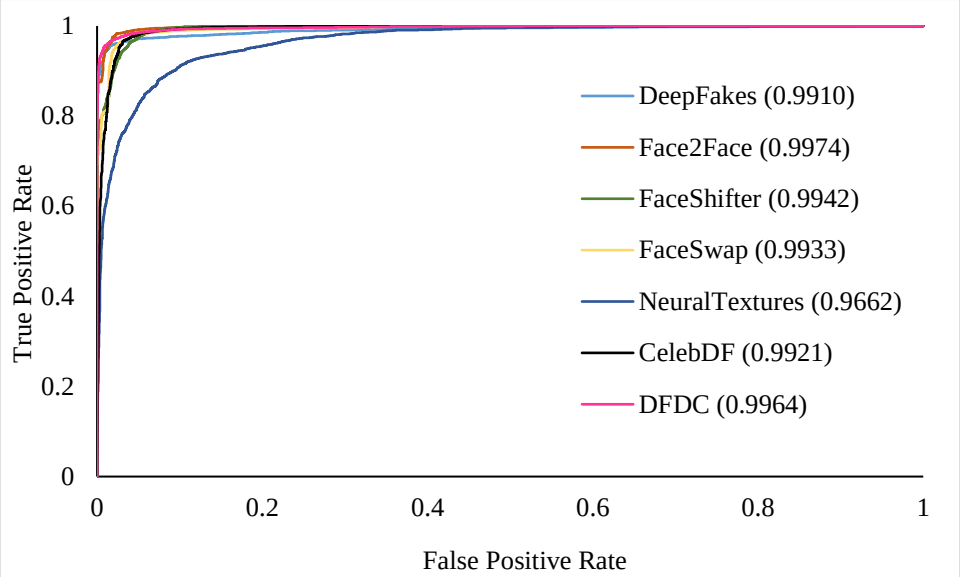


Fig. 11 ROC curves for MRT-Net on FaceForensics++, CelebDF and DFDC datasets.

Most manipulation detection methods specify the performance in terms of accuracy, precision, recall, F1 and AUC scores but these metrics are asymmetric. However, MCC is a robust metric for measuring the model's ability to identify both classes instead of just the positive class. Table III shows high MCC scores above 0.90 in most cases. This proves that MRT-Net is equally good at identifying original face images and facial manipulations.

3.2.4.2 Comparison Against Existing State-of-the-Arts

This section compares the performance of MRT-Net against the recent state-of-the-art methods for face manipulation detection.

Table IV Result comparison of MRT-Net with the state-of-the-art methods on the FF++ dataset.

Methods	Year	DeepFake		Face2Face		FaceShifter		FaceSwap		NT		Average	
		ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Yang et al. [123]	2023	-	-	-	-	-	-	-	-	-	-	-	0.8709
Guo et al. [124]	2023	-	-	-	-	-	-	-	-	-	-	-	0.9755
Lin et al. [97]	2023	-	-	-	-	-	-	-	-	-	-	0.9074	0.9486
Guo et al. [27]	2023	-	-	-	-	-	-	-	-	-	-	-	0.9879
Yang et al. [125]	2023	-	-	-	-	-	-	-	-	-	-	0.9382	0.9827
Xu et al. [126]	2023	-	-	-	-	-	-	-	-	-	-	-	0.9034
Yang et al. [127]	2022	-	-	-	-	-	-	-	-	-	-	-	0.7888
Nirkin et al. [128]	2022	0.9450	-	0.8030	-	-	-	0.8450	-	0.7400	-	-	-
Liu et al. [129]	2021	0.9348	-	0.8602	-	-	-	0.9226	-	0.7678	-	0.8713	-
*AMTENnet [107]	2021	0.9285	0.9801	0.9204	0.9756	0.9273	0.9837	0.8995	0.9665	0.7728	0.8455	0.8897	0.9502
*Coordinate-Attention [100]	2021	0.8675	0.9430	0.5932	0.6295	0.7536	0.8276	0.7562	0.8441	0.6162	0.6514	0.7173	0.7791
Shang et al. [130]	2021	0.9563	-	0.9015	-	-	-	0.9493	-	0.8001	-	-	-

Methods	Year	DeepFake		Face2Face		FaceShifter		FaceSwap		NT		Average	
		ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Chen et al. [131]	2021	-	0.9595	-	-	-	-	-	0.9787	-	-	-	-
Hu et al. [132]	2021	0.9464	0.9800	0.8648	0.9400	-	-	0.8527	0.9400	0.8005	0.9000	-	-
* CDCN++ [109]	2020	0.9074	0.9727	0.8715	0.9458	0.9081	0.9716	0.9086	0.9601	0.7639	0.8422	0.8719	0.9384
Qian et al. [133]	2020	0.9597	-	0.9532	-	-	-	0.9653	-	0.8332	-	0.9278	-
Baek et al. [134]	2020	0.7180	-	0.6860	-	-	-	0.6310	-	0.7070	-	-	-
Zi et al. [135]	2020	0.9210	-	0.8390	-	-	-	0.9250	-	0.7820	-	-	-
Rössler et al. [111]	2019	0.7450	-	0.7590	-	-	-	0.7090	-	0.7330	-	-	-
Amerini et al. [26]	2019	-	-	0.8161	-	-	-	-	-	-	-	-	-
Afchar et al. [136]	2018	0.8727	-	0.5620	-	-	-	0.6117	-	0.4067	-	0.6132	-
MRT-Net (Proposed)	-	**0.9670**	**0.9910**	**0.9767**	**0.9974**	**0.9611**	**0.9942**	**0.9676**	**0.9933**	**0.9025**	**0.9662**	**0.9550**	**0.9884**

Table IV shows the performance comparison of MRT-Net against the recent state-of-the-arts on the FF++ dataset. MRT-Net achieves better scores across all different manipulations of FF++. MRT-Net comfortably beats all the state-of-the-arts including CDCN++, AMTENnet and CoordinateAttention network, clearly proving its superiority. Additionally, Fig. 12 presents a visual comparison of the ACC, AUC and MCC scores of MRT-Net against CDCN++, AMTENnet and CoordinateAttention networks on the FF++ (DF) dataset category.

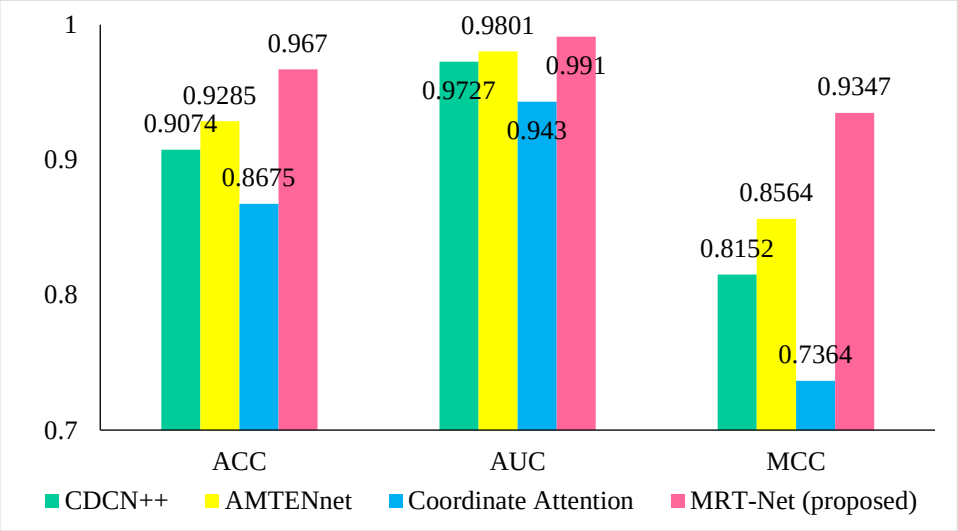


Fig. 12 Comparison of MRT-Net against the base papers on the FF++ (DF) dataset.

Table V presents results achieved by MRT-Net obtained on the CelebDF dataset are reported and compared against the state-of-the-art methods. Once again, MRT-Net proves its superiority against CDCN++, AMTENnet and CoordinateAttention network and all the other state-of-the-arts.

Table V Result comparison of MRT-Net with the state-of-the-art methods on the CelebDF dataset.

Methods	Year	ACC	AUC
Asha et al. [137]	2023	0.8800	0.8400
Yang et al. [123]	2023	-	0.9108
Guo et al. [124]	2023	-	0.6743
Ke et al. [138]	2023	0.8892	
Guo et al. [27]	2023	-	0.6950
Xu et al. [126]	2023	-	0.9332
Li et al. [139]	2023	0.7169	0.7659
Nadimpalli et al. [140]	2022	0.6200	0.6700
Yang et al. [127]	2022	-	0.8561
Nirkin et al. [128]	2022	-	0.6600
Li et al. [141]	2021	-	0.7600
Hu et al. [48]	2021	-	0.7884
Trinh et al. [142]	2021	-	0.7176
Luo et al. [143]	2021	0.7435	0.9234
Chen et al. [144]	2021	0.9575	-
Chen et al. [131]	2021	-	0.8765
Hu et al. [132]	2021	0.8074	0.8700
*AMTENnet [107]	2021	0.9254	0.8804
*CoordinateAttention [100]	2021	0.8904	0.8026
Qian et al. [133]	2020	0.8706	0.8148
*CDCN++ [109]	2020	0.9180	0.8982
Dang et al. [50]	2020	-	0.7120
Choi et al. [51]	2020	0.9200	0.9400
Li et al. [145]	2020	-	0.7476
Afchar et al. [136]	2018	0.6750	0.6681
MRT-Net (Proposed)	**-**	**0.9815**	**0.9921**

Table VI Result comparison of MRT-Net with the state-of-the-art methods on the DFDC dataset.

Methods	Year	ACC	AUC
Deng et al. [146]	2023	0.9216	0.9784
Mohiuddin et al. [147]	2023	0.7534	0.8567
Asha et al. [137]	2023	0.8700	0.9100
Guo et al. [124]	2023	-	0.9597
Ke et al. [138]	2023	0.9108	-
Lin et al. [97]	2023	-	0.8847
Guo et al. [27]	2023	-	0.9827
Yu et al. [148]	2023	0.9593	0.9896
Zhao et al. [149]	2023	0.9210	-
Yang et al. [125]	2023	-	0.9911
Heo et al. [150]	2023	-	0.9780
Xu et al. [126]	2023	-	0.8037
Ganguly et al. [151]	2022	0.7321	0.8632
Ganguly et al. [152]	2022	0.7520	0.8359
Nadimpalli et al. [140]	2022	0.9100	0.8600
*AMTENnet [107]	2021	0.9139	0.9402
*CoordinateAttention [100]	2021	0.8917	0.8830
Shang et al. [130]	2021	-	0.9778
Xu et al. [28]	2021	0.8453	0.9347
Li et al. [141]	2021	-	0.7600
Luo et al. [143]	2021	0.8841	0.9496
Li et al. [145]	2020	-	0.8090
*CDCN++ [109]	2020	0.9174	0.9492
Li et al. [153]	2020	0.8511	-
Qi et al. [42]	2020	0.6410	-
Mittal et al. [53]	2020	-	0.8440
Chugh et al. [52]	2020	-	0.9160
Montserrat et al. [17]	2020	0.9188	-

Methods	Year	ACC	AUC
MRT-Net (Proposed)	**-**	**0.9760**	**0.9964**

Table VI shows the results of MRT-Net on the DFDC dataset. Here again, it is observed that the proposed model outperforms all the state-of-the-art methods including base paper methods CDCN++, AMTENnet and CoordinateAttention network.

3.2.4.3 Complexity Analysis of MRT-Net

This section presents the computational complexity of the proposed MRT-Net architecture and compares it against the popular computer vision models. The computational factors under consideration include the number of trainable parameters, the number of 'Multiply-Accumulate' (MACs) operations, accuracy achieved on the DF (FF++) dataset, inference time on CPU and GPU.

In the MAC metric, "multiply" means to perform the multiplication operation on two numbers or elements, usually elements found in matrices within deep learning. "Accumulate" means to sum together the outcomes of several multiplication processes. A single multiply-accumulate process entails multiplying two integers and adding the product to an accumulating total. The MAC measure quantifies the number of multiply-accumulate operations needed to calculate the neural network's output.

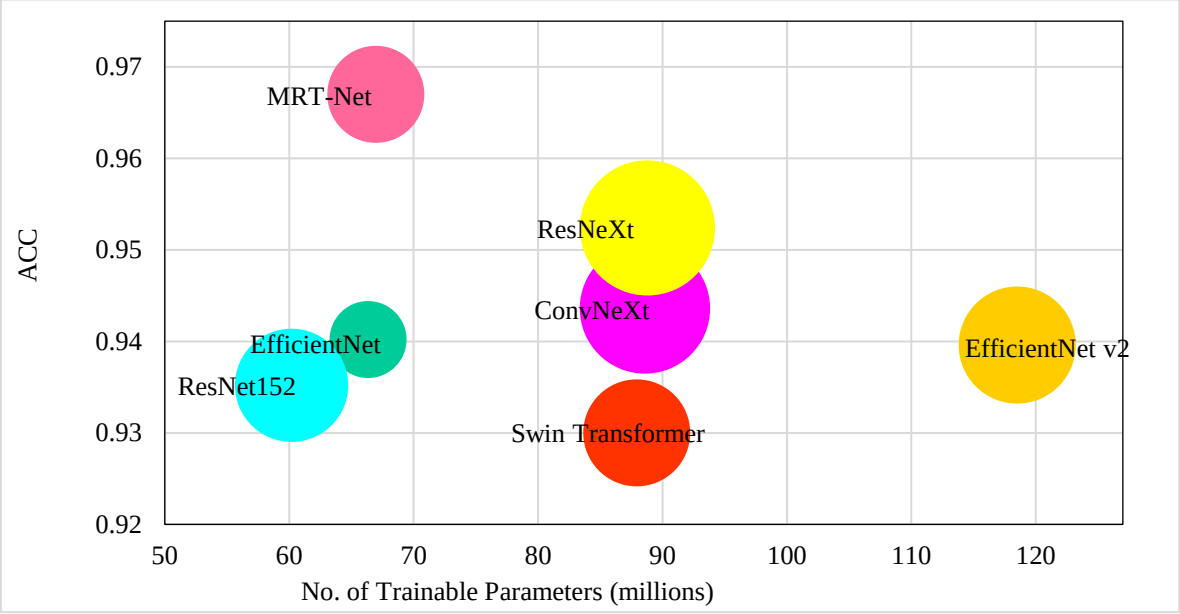


Fig. 13 Complexity analysis of MRT-Net against popular computer vision models.

The intricacy of a deep learning model is indicated by its MAC count, as a larger MAC count often signifies that the model demands more computing resources, such as CPU cycles

or GPU RAM, to carry out the required computations. Models with larger MAC counts are usually more intricate and need robust hardware for efficient training and deployment.

Fig. 13 presents a visual complexity analysis of MRT-Net against popular computer vision models. The vertical axis specifies the model accuracy achieved on the FF++ (DF) dataset. The horizontal axis specifies the number of trainable parameters in millions. The size of each circle represents the number of 'multiply-accumulate' operations per input image. Larger circles have higher MAC values. The MRT-Net circle is at the top left corner of the diagram, which signifies high performance. EfficientNet and Resnet152 have a similar number of trainable parameters but achieve less accuracy. ResNeXt, ConvNeXt, Swin Transformer, and EfficientNet v2 are larger models regarding trainable parameters, yet MRT-Net achieves better accuracy than all.

Table VII Comparison of Computational Complexity of MRT-Net against popular computer vision models.

Model	**ACC**	**Parameter (millions)**	**MACs ($\times 10^9$)**	**CPU time (s)**	**GPU time (s)**
ConvNeXt	0.9436	88.59	5.02	4.95	1.18
EfficientNet	0.9402	66.34	1.75	7.27	1.43
EfficientNet v2	0.9396	118.51	4.05	6.61	0.96
Swin Transformer	0.9300	87.93	3.38	6.69	0.93
ResNet152	0.9352	60.19	3.79	4.41	0.32
ResNeXt	0.9524	88.79	5.40	4.75	0.87
CDCN++	0.9074	2.38	13.08	18.16	1.28
AMTENnet	0.9285	1.93	0.13	0.65	0.07
CoordinateAttention	0.8675	2.67	0.11	1.33	0.43
MRT-Net (proposed)	0.9670	66.96	2.78	5.74	0.51

Table VII presents a tabular view of MRT-Net complexity against similar-sized computer vision models as well as the three base papers, CDCN++, AMTENnet and CoordinateAttention network. MRT-Net beats all models in terms of the accuracy score. In terms of trainable parameters, MRT-Net lies in between, where EfficientNet v2 is the largest model with 118.51 million parameters. The base papers CDCN++, AMTENnet and CoordinateAttention network are fairly small-sized networks with just 2.38, 1.93 and 2.67 million parameters respectively. These models are fairly lightweight when compared to MRT-Net having 66.96 million parameters.

The CDCN++ architecture has an unusually high number of MAC operations per image causing the highest inference time per input batch on both CPU and GPU as compared to other models.

3.2.4.4 Ablation Study of MRT-Net

This section conducts ablation study to establish the benefits of its individual components.

Classification by Dual Branch

This section features a dual-branch architecture incorporating joint learning from spatial and textural domains. ResNet50 acts as the backbone architecture for both branches. The average pool and fully connected layers are trimmed from the end of resnet50 in both branches. Therefore, the output shape from both branches is $4 \times 4 \times 2048$, as is the case for resnet50. Features from both branches are fused in a weighted manner as mentioned in the proposed architecture section. Table VIII presents the results obtained from joint feature learning with different initial values of $\propto_1$ and $\propto_2$.

Table VIII Classification results on dual branch-architecture having a fusion of color and texture modality.

Initial		**Final**		**ACC**	**P**	**R**	**F1**	**AUC**	**MCC**
$\propto_1$	$\propto_2$	$\propto_1$	$\propto_2$						
without adaptive weights				0.7562	0.7865	0.7413	0.7632	0.8377	0.5136
0.5	0.5	0.4681	0.5319	0.7627	0.7395	0.8327	0.7833	0.8497	0.5276
0.6	0.4	0.5048	0.4952	0.7770	0.7776	0.7651	0.7713	0.8752	0.5539
0.4	0.6	0.4133	0.5867	0.7911	0.7738	0.7823	0.7780	0.8739	0.5808
0.75	0.25	0.5526	0.4474	**0.7928**	0.7814	**0.8370**	**0.8083**	**0.8796**	**0.5851**
0.25	0.75	0.3195	0.6805	0.7847	**0.7933**	0.7871	0.7902	0.8684	0.5691

Table VIII shows the performance due to joint feature learning architecture. The model achieves the worst scores when there is no adaptive weighting. Alternatively, the model when trained with adaptive weighting achieves accuracy in the range 76% to 80% for different initial values of $\propto_1$ and $\propto_2$. Also, it can be inferred that color and texture features yield the best results when fused with 75% and 25% weighted composition.

Choice of Weight Initialization Strategy

This section aims to improve model performance by changing the weight initialization strategy. Both branches were initialized with random weights in the previous section. In this section, the color branch is initialized with ImageNet pre-trained weights, while the texture branch is randomly initialized. Experimental results obtained from this new weight initialization strategy are shown in Table IX.

The results demonstrate the superiority of the new weight initialization approach. For the FF++ (DF) dataset, the best accuracy with random weight initialization was 0.7928, which increased to 0.9568 after initializing the color branch with pre-trained ImageNet weights. The same is true for all classification metrics reported. AUC increased from 0.8796 to 0.9913, F1 increased from 0.8083 to 0.9561, and the MCC score is boosted from 0.5851 to 0.9137. This presents substantial evidence of the benefits of using such a weight initialization.

Table IX Classification results on dual branch architecture having a fusion of color and texture modality with imagenet weights in color branch.

	Initial		**Final**		**ACC**	**P**	**R**	**F1**	**AUC**	**MCC**
	$\propto_1$	$\propto_2$	$\propto_1$	$\propto_2$						
DF	without adaptive weights				0.9276	0.9169	0.9319	0.9243	0.9768	0.8550
	0.5	0.5	0.6937	0.3063	0.9432	**0.9631**	0.9238	0.9430	0.9850	0.8873
	0.6	0.4	0.7516	0.2484	0.9506	0.9549	0.9476	0.9512	0.9887	0.9013
	0.4	0.6	0.6226	0.3774	0.9490	0.9460	0.9442	0.9451	0.9847	0.8976
	0.75	0.25	0.7977	0.2023	**0.9568**	0.9613	**0.9511**	**0.9561**	**0.9913**	**0.9137**
	0.25	0.75	0.5381	0.4619	0.9394	0.9559	0.9193	0.9372	0.9856	0.8795
F2F	without adaptive weights				0.9264	0.9441	0.9101	0.9268	0.9781	0.8535
	0.5	0.5	0.6214	0.3786	0.9465	0.9582	0.9270	0.9423	0.9766	0.8929
	0.6	0.4	0.6559	0.3441	0.9563	0.9701	0.9436	0.9567	**0.9861**	0.9130
	0.4	0.6	0.5977	0.4023	0.9379	0.9489	0.9295	0.9391	0.9847	0.8760
	0.75	0.25	0.7017	0.2983	**0.9584**	**0.9729**	**0.9507**	**0.9616**	0.9850	**0.9165**
	0.25	0.75	0.5183	0.4817	0.9484	0.9473	0.9485	0.9479	0.9843	0.8968
FaceSwap	without adaptive weights				0.9278	0.9129	0.9336	0.9231	0.9731	0.8553
	0.5	0.5	0.6336	0.3664	0.9445	0.9501	0.9337	0.9419	0.9787	0.8890
	0.6	0.4	0.6744	0.3256	0.9474	0.9548	**0.9510**	**0.9529**	0.9804	0.8935
	0.4	0.6	0.6100	0.3900	0.9356	0.9350	0.9338	0.9344	0.9781	0.8711
	0.75	0.25	0.6884	0.3116	**0.9479**	**0.9649**	0.9386	0.9515	**0.9824**	**0.8956**
	0.25	0.75	0.5139	0.4861	0.9395	0.9235	0.9349	0.9292	0.9748	0.8590
DFDC	without adaptive weights				0.9075	0.9172	0.9684	0.9421	0.9507	0.7199
	0.5	0.5	0.6209	0.3791	0.9246	0.9364	0.9703	0.9530	0.9578	0.7651
	0.6	0.4	0.6719	0.3281	0.9373	**0.9534**	0.9680	0.9607	0.9758	0.8068
	0.4	0.6	0.5742	0.4258	0.9230	0.9355	0.9686	0.9518	0.9552	0.7645
	0.75	0.25	0.7341	0.2659	**0.9441**	0.9532	**0.9775**	**0.9652**	**0.9763**	**0.8249**
	0.25	0.75	0.5119	0.4881	0.9189	0.9421	0.9579	0.9499	0.9500	0.7365

Fig. 14 plots the variation of $\propto_1$ and $\propto_2$ values for the two weight initialization approaches. Graphs in the first column of Fig. 14 represent random weight initialization with different initial values for $\propto_1$ and $\propto_2$. The second column represents ImageNet weight initialization in the color branch. In row a) of Fig. 14, having $\propto_1$ (blue) and $\propto_2$ (orange) initialized to 0.5 and 0.5, the random weight initialization causes the texture coefficient $\propto_2$ to increase to 0.5319 while $\propto_1$ decreases to 0.4681.

A similar pattern of $\propto_1$ decreasing and $\propto_2$ increasing can be observed in the first column for rows b), c), d) and e) for the random weight initialization.

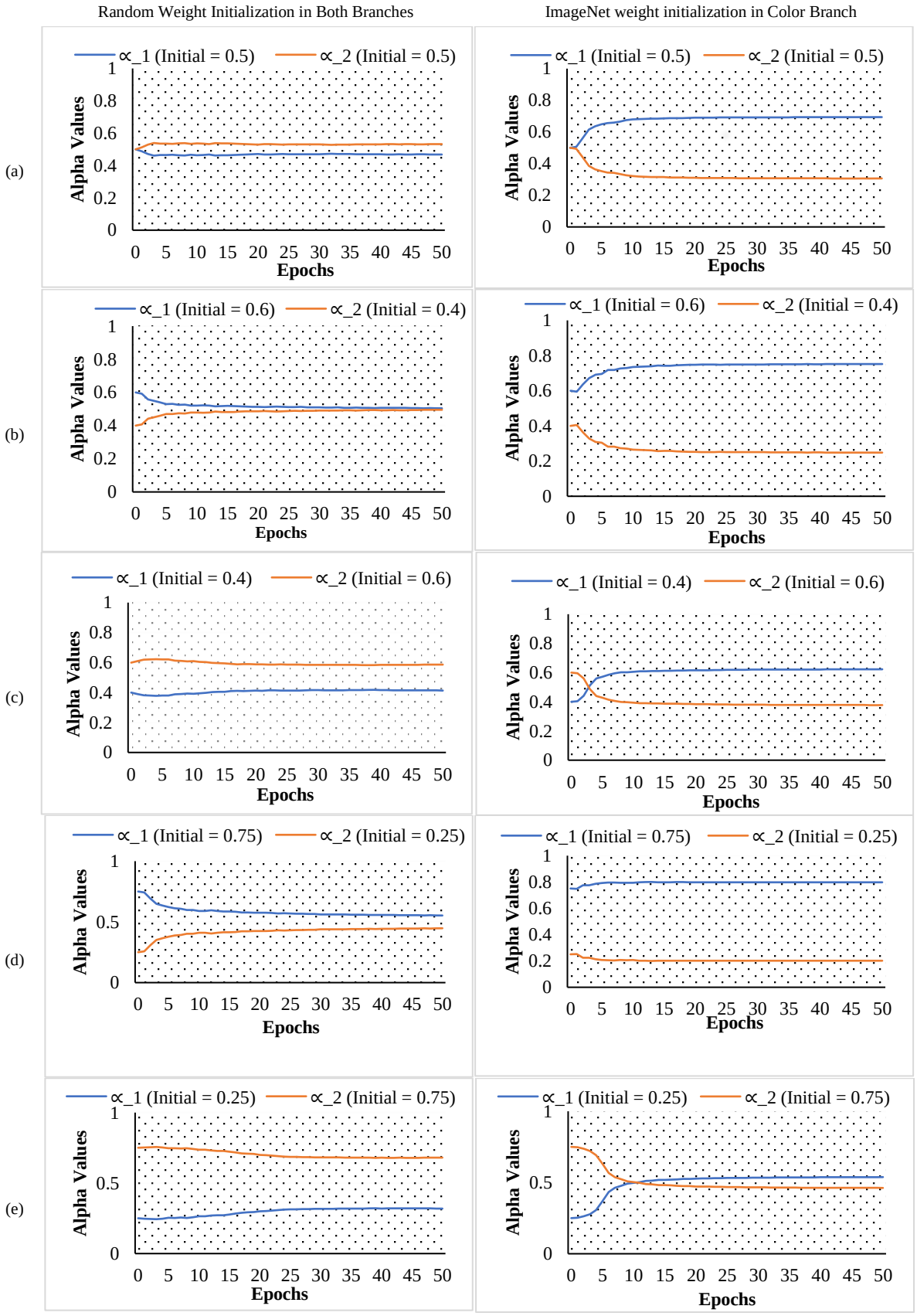


Fig. 14 Comparison of changes in values of $\propto_1$ and $\propto_2$ for random weight initialization in both branches (first column) and ImageNet weights initialization in the color branch (second column).

The values for $\propto_1$ and $\propto_2$ change from 0.6 and 0.4 to 0.5048 and 0.4952 respectively in row b), from 0.4 and 0.6 to 0.4133 and 0.5867 respectively in row c), from 0.75 and 0.25 to 0.5526 and 0.4474 respectively in row d) and lastly, from 0.25 and 0.75 to 0.3195 and 0.6805 respectively in row e). This means that the model gives more weightage to textural features from the second branch when both branches are initialized randomly. However, the accuracy metrics achieved from such weight initialization (Table VIII) lie roughly in the range of 0.76 to 0.80, which is not impressive compared to existing state-of-the-art methods.

The second column of Fig. 14 represents the ImageNet weight initialization of the color branch. An opposite trend is observed in this column where $\propto_1$ increases and $\propto_2$ decreases for different initial values. The values for $\propto_1$ and $\propto_2$ change from 0.5 and 0.5 to 0.6937 and 0.3063 respectively in row a), from 0.6 and 0.4 to 0.7516 and 0.2484 respectively in row b), from 0.4 and 0.6 to 0.6226 and 0.3774 respectively in row c), from 0.75 and 0.25 to 0.7977 and 0.2023 respectively in row d) and lastly, from 0.25 and 0.75 to 0.5381 and 0.4619 respectively in row e). This clearly indicates that when the color branch is initialized with ImageNet weights, the model gives more weightage to color features learned from the first branch. But the main advantage is the massive jump in classification scores. The accuracy scores of the model now reach roughly in the range of 0.93 to 0.95 (Table IX), which is a significant improvement. Similarly, results on F2F, FaceSwap and DFDC datasets confirm that the best initial values for $\propto_1$ and $\propto_2$ are 0.75 and 0.25. Additionally, it can be seen that the performance with no adaptive weighting is worse than all cases of adaptive weighting, clearly highlighting the importance of the auto-adaptive weighting mechanism.

Hence, moving forward, the color branch is initialized with ImageNet weights and initial values of $\propto_1$ and $\propto_2$ are set to 0.75 and 0.25, respectively, since that case produces the best results.

Choice of Fusion Strategy

This section explores the fusion strategy for merging color and texture modality. So far, the fusion process involved concatenating features along the depth. In this section, the sum and mean operations are tried as the fusion strategy. Initial value of $\propto_1$ and $\propto_2$ are set to 0.75 and 0.25, since they yielded the best results in the previous section.

Table X Comparison of feature fusion strategies – concatenation, sum and mean.

	Fusion Strategy	ACC	P	R	F1	AUC	MCC
DF	Concatenation	**0.9568**	0.9613	0.9511	**0.9561**	**0.9913**	**0.9137**

	Fusion Strategy	ACC	P	R	F1	AUC	MCC
	Sum	0.9557	0.9415	**0.9667**	0.9539	0.9905	0.9115
	Mean	0.9454	**0.9743**	0.9204	0.9466	0.9834	0.8925
F2F	Concatenation	**0.9584**	**0.9729**	**0.9507**	**0.9616**	**0.9850**	**0.9165**
F2F	Sum	0.9483	0.9529	0.9390	0.9459	0.9828	0.8965
F2F	Mean	0.9426	0.9611	0.9253	0.9429	0.9778	0.8859
FaceSwap	Concatenation	**0.9479**	**0.9649**	0.9386	**0.9515**	**0.9824**	**0.8956**
FaceSwap	Sum	0.9355	0.9513	0.9184	0.9345	0.9780	0.8715
FaceSwap	Mean	0.9301	0.9264	**0.9417**	0.9340	0.9617	0.8599
DFDC	Concatenation	**0.9441**	**0.9532**	**0.9775**	**0.9652**	**0.9763**	**0.8249**
DFDC	Sum	0.9267	0.9413	0.9650	0.9530	0.9653	0.7871
DFDC	Mean	0.9152	0.9338	0.9602	0.9468	0.9498	0.7396

Results from Table X clearly show that concatenation fusion performs best. Henceforth, concatenation fusion is treated as the default choice for feature fusion.

Impact of Manipulation Residual Module

This section studies the impact of extracting manipulation residuals on facial manipulation detection. Three experiments are designed to extract manipulation residuals from the color, texture, or both branches.

Table XI Comparison of Manipulation Residual Extraction Module in color branch only, texture branch only and both branches.

	Branch with MR	ACC	P	R	F1	AUC	MCC
DF	Color Only	**0.9606**	0.9580	**0.9648**	**0.9614**	0.9911	**0.9212**
DF	Texture Only	0.9491	**0.9636**	0.9366	0.9499	0.9894	0.8987
DF	Both	0.9527	0.9560	0.9507	0.9534	**0.9918**	0.9055
F2F	Color Only	**0.9664**	**0.9860**	0.9446	**0.9648**	**0.9912**	**0.9335**
F2F	Texture Only	0.9224	0.9451	0.9116	0.9280	0.9776	0.8446
F2F	Both	0.9300	0.8946	**0.9494**	0.9212	0.9817	0.8596
FaceSwap	Color Only	**0.9574**	**0.9751**	**0.9391**	**0.9568**	**0.9848**	**0.9156**
FaceSwap	Texture Only	0.9202	0.9057	0.9410	0.9230	0.9753	0.8409
FaceSwap	Both	0.9330	0.9319	0.9339	0.9329	0.9759	0.8661
DFDC	Color Only	**0.9656**	**0.9757**	**0.9793**	**0.9775**	**0.9916**	**0.9041**
DFDC	Texture Only	0.9350	0.9489	0.9687	0.9587	0.9718	0.8069
DFDC	Both	0.9358	0.9510	0.9692	0.9600	0.9741	0.7989

Table XI results show that the manipulation extraction module in the color branch works best for all datasets evaluated.

Impact of Attention Module

In this section, three recently proposed novel attention mechanisms, namely, *Triplet Attention* [102], *Shuffle Attention* [104] and *Coordinate Attention* [100] are integrated with the dual

branch architecture. Table XII presents the results obtained by trying the abovementioned attention mechanisms.

Table XII Comparison of three state-of-the-art attention modules: Triplet attention [102], Shuffle attention [104] and Coordinate attention [100]

	Attention Type	ACC	P	R	F1	AUC	MCC
DF	Triplet Attention	0.9451	0.9701	0.9226	0.9458	0.9830	0.8915
	Shuffle Attention	0.9381	0.9504	0.9175	0.9336	0.9805	0.8762
	Coordinate Attention	**0.9670**	**0.9867**	**0.9504**	**0.9682**	**0.9910**	**0.9347**
F2F	Triplet Attention	0.9555	0.9698	0.9348	0.9520	0.9863	0.9112
	Shuffle Attention	0.9379	0.9567	0.9160	0.9359	0.9787	0.8765
	Coordinate Attention	**0.9767**	**0.9704**	**0.9857**	**0.9780**	**0.9974**	**0.9534**
FaceSwap	Triplet Attention	0.9452	0.9393	0.9516	0.9454	0.9788	0.8906
	Shuffle Attention	0.9263	0.9202	0.9262	0.9232	0.9699	0.8524
	Coordinate Attention	**0.9676**	**0.9671**	**0.9697**	**0.9684**	**0.9933**	**0.9353**
DFDC	Triplet Attention	0.9236	0.9479	0.9530	0.9505	0.9642	0.7839
	Shuffle Attention	0.9177	0.9239	0.9761	0.9493	0.9562	0.7403
	Coordinate Attention	**0.9760**	**0.9805**	**0.9891**	**0.9848**	**0.9964**	**0.9279**

Table XII demonstrate that coordinate attention works best with the proposed model. A detailed description of coordinate attention is mentioned in the proposed architecture section.

Final Model – MRT-Net

Based on the ablation study so far, the characteristics of the proposed model MRT-Net, are as follows. It contains two resnet50 backbone networks for spatial and textural feature learning. It is initialized with ImageNet weights in the spatial branch. It fuses auto-adaptive weighted spatial and textural features using depth-wise concatenation. It implements a manipulation trace extraction module in the color branch. Finally, it utilizes coordinate attention in both branches to boost classification capability.

Fig. 15 and Fig. 16 present a visual demonstration of the increase in performance due to the addition of the manipulation residual and attention modules on the FF++ (F2F) and DFDC datasets respectively.

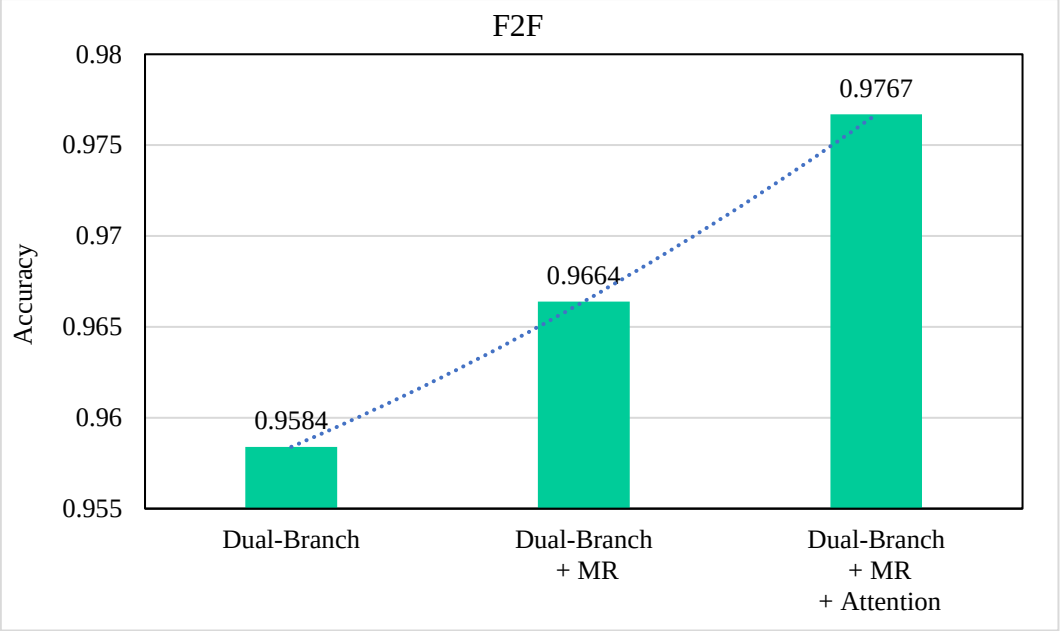


Fig. 15 Increase in model accuracy on the F2F dataset by adding the MR and Attention modules.

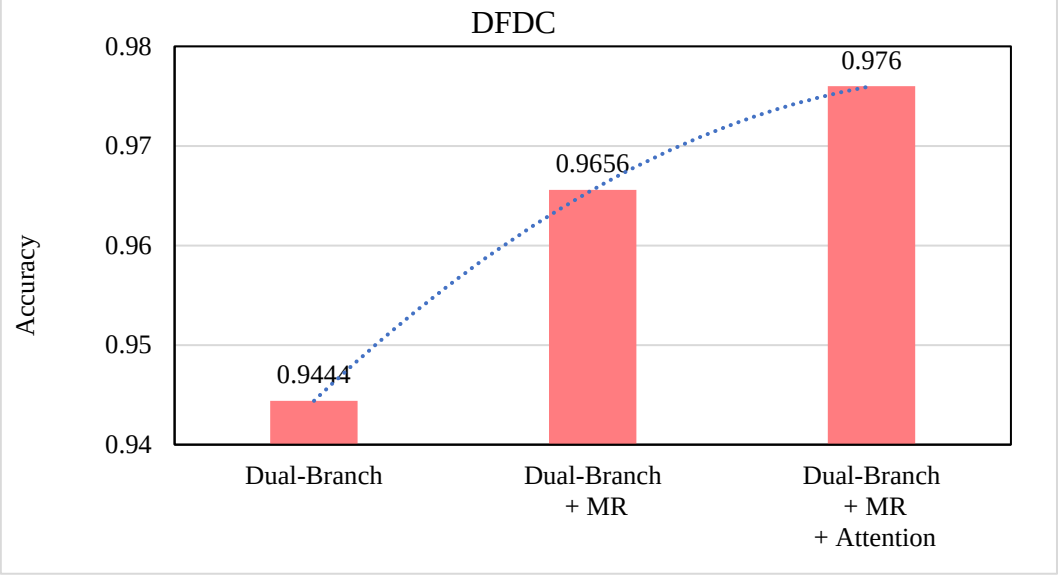


Fig. 16 Increase in model accuracy on the DFDC dataset by adding the MR and Attention modules.

3.2.4.5 Qualitative Analysis of MRT-Net

This section presents a qualitative analysis of the MRT-Net model. Specifically, the visualization diagram demonstrating the regions of focus for MRT-Net have been plotted. Several methods exist that predict class activation maps (CAM) for a given CNN. LayerCAM [154], a recently proposed CNN visualization method, is used to produce reliable CAM maps for different layers of MRT-Net as shown in Fig. 17.

3.2.5 Conclusion

This section proposed a novel facial manipulation detection network MRT-Net. The proposed network is a dual-branch architecture learning discriminative features from manipulation residuals and textural information. MRT-Net enjoys an auto-adaptive weighting strategy for fusing features learned from the two branches. A recently proposed coordinate attention boosts MRT-Net's classification capabilities by highlighting important feature channels. Experimental results on FF++, CelebDF and DFDC datasets clearly prove the superiority of the proposed model achieving high scores in terms of accuracy, precision, recall and F1 scores. MRT-Net achieves above 0.99 AUC score in most cases as shown in Fig. 11. MRT-Net also attains over 0.90 MCC scores in most cases, indicating that the proposed model not only learns to identify the positive class (manipulated facial images) with high confidence but also identifies the negative class (original face images) with certainty. Comparison results from Table IV, Table VI and Table V prove that MRT-Net is superior to the existing state-of-the-art facial manipulation detection methods.

The ablation study presented in Table IX demonstrates the superiority of the auto-adaptive weighting mechanism of the two features as compared to the direct fusion of features from the two branches. This is primarily because MRT-Net can choose the ideal proportion of manipulation residual and textural features due to this adaptive weighting mechanism.

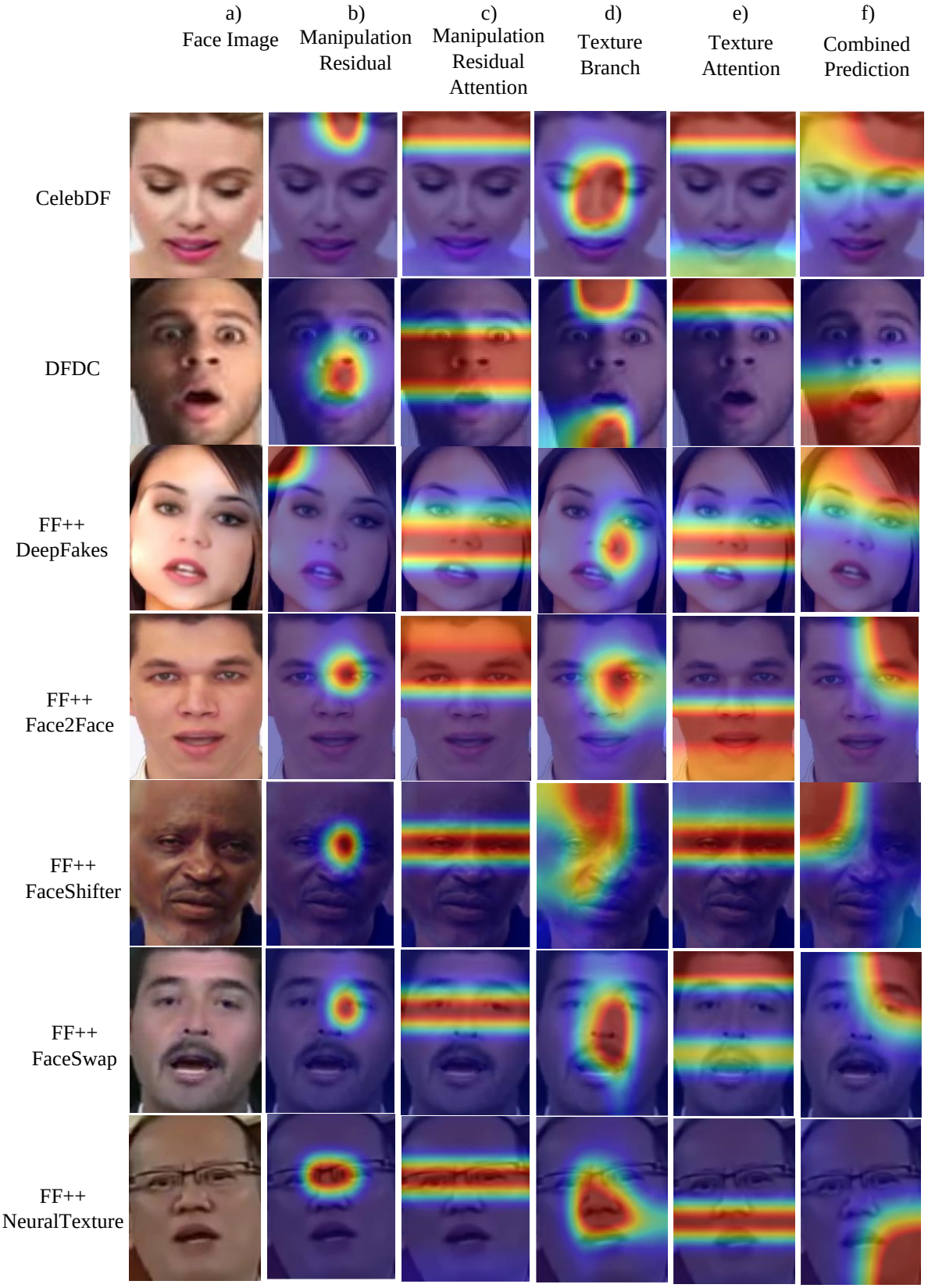


Fig. 17 MRT-Net's region of focus from the perspective of b) Manipulation Residual Branch c) Manipulation Residual Attention d) Texture Branch e) Texture Attention f) Combined Overall Prediction of MRT-Net.

3.3 AW-MSA: Adaptively Weighted Multi-Scale Attentional Features for DeepFake Detection

3.3.1 Abstract

With the recent rise of realistic face manipulation methods, building robust face tampering detection methods has become more critical than ever before. Several research works have focussed on extracting multi-scale features to enhance the feature learning process. However, most of such works suffer from a design flaw of combining multiple scale information in equal proportion. This is not the best approach, as a feature from one scale could be more important than other scale features. To this end, a novel deepfake detection architecture, *Face-NeSt* has been proposed. Face-NeSt has the unique ability to choose an ideal proportion of multi-scale features best suited for the final prediction. Specifically, Face-NeSt employs a novel 'adaptively weighted multi-scale attentional' (AW-MSA) module that is capable of choosing the best proportion of multi-scale features. Face-NeSt uses an attention mechanism that allows it to highlight important feature regions along the spatial and channel dimensions, both locally and globally. Unlike the popular computer vision models of recent times, Face-NeSt is designed to be computationally light-weight. Face-NeSt performs admirably on three publicly available benchmark datasets: FaceForensics++ (FF++), CelebDF and Deep Fake Detection Challenge (DFDC). The AUC scores are **0.9823** on CelebDF, **0.9947** on DFDC, **0.9945** on DeepFake (FF++), **0.9905** on Face2Face (FF++), **0.9978** on FaceShifter (FF++), **0.9948** on FaceSwap (FF++) and **0.9548** on NeuralTextures (FF++). These excellent findings highlight Face-NeSt's efficacy since it easily outperforms all state-of-the-art (SOTA) approaches for facial tampering detection.

3.3.2 Proposed Architecture

This section describes the proposed model Face-NeSt and its novel components. Fig. 18 presents a visual overview of the proposed architecture Face-NeSt.

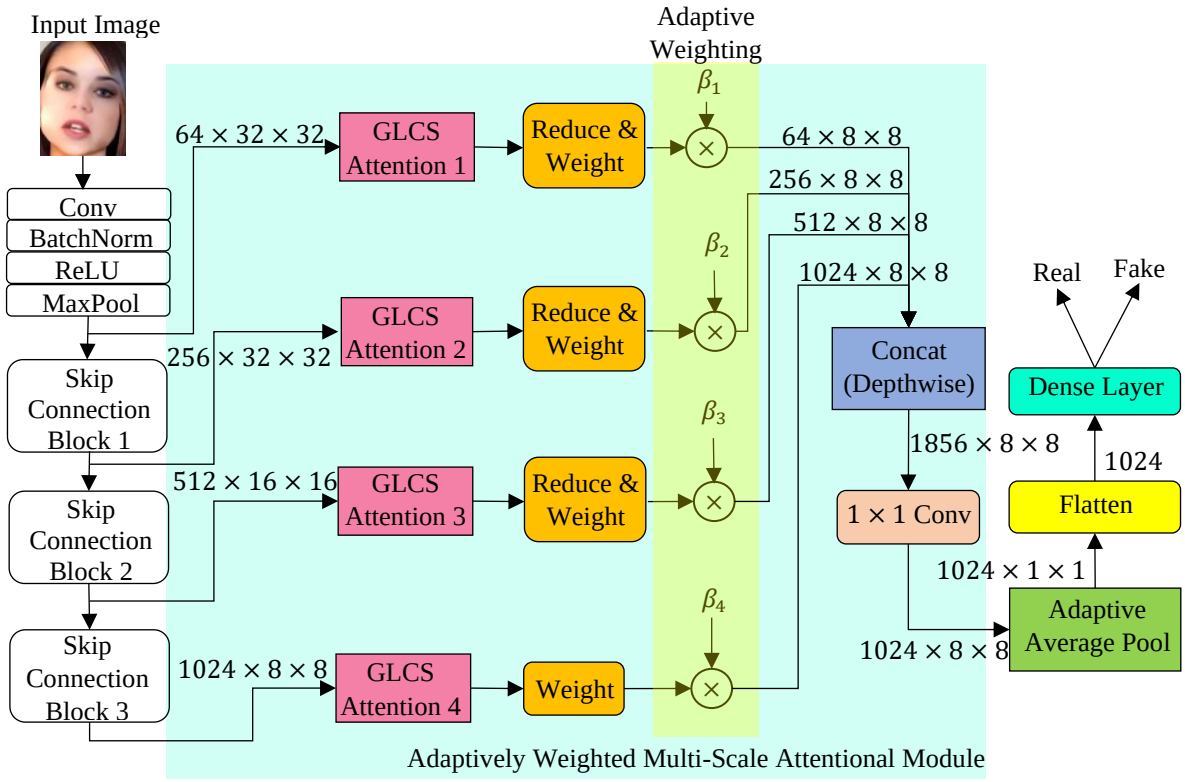


Fig. 18 The architecture of the proposed Face-NeSt model. GLCS stands for Global Local Channel Spatial attention.

Algorithm 2 Pseudocode for Face-NeSt

Input:

$Dataset = \{\mathbb{X}_i, \mathbb{Y}_i\}_{i=1}^{n}$ having facial images $\mathbb{X}_i \in \mathbb{R}^{3\times128\times128}$ and labels $\mathbb{Y}_i \in \{0,1\}$.
Trainable parameters θ
Size of each batch $\mathcal{B}$
No. of batches $\mathscr{b}_{total}$
Total number of epochs $\mathcal{E}$
Initial Learning Rate ℓr
Learning Rate Decay factor γ
Factors for Adaptive weighting $(\beta) \in \{\beta_1, \beta_2, \beta_3, \beta_4\}$

Output:

Trained Face-NeSt model.

1. Initialize θ and the adaptive weights β.
2. **for** $e = 1,2,3 \ldots \mathcal{E}$ **do** — Training epochs loop.
3. **for** $\mathscr{b} = 1,2,3 \ldots \mathscr{b}_{total}$ **do** — Reading batch loop.
4. $(\mathbb{X}, \mathbb{Y}) \sim \mathcal{S}$ — Randomly select one batch.
5. **for** feature $\mathcal{F}_i$ at scale 'i' **do** — Loop through features at each scale $\mathcal{F}_i$
6. $\mathcal{F}_{i,attentional} = \emptyset(\mathcal{F}_i)$ — Compute the attentional features from each scale $(\mathcal{F}_{i,attentional})$
7. $\mathcal{F}_{i,weighted-attentional} = \mathbb{MUL}(\mathcal{F}_{i,attentional}, \beta_i)$ — Adaptively weight attentional features from each scale $(\mathcal{F}_{i,weighted-attentional})$
8. **end for**
9. $\mathcal{F}_{fused} = \sum_{i=1}^{4}(\mathcal{F}_{i,weighted-attentional})$ — Fuse the adaptively weighted multi-scale attentional features $(\mathcal{F}_{fused})$. $\sum(.)$ represents feature fusion along the depth.
10. $\mathcal{F}_{final} \leftarrow \mathcal{L}(\mathcal{F}_{fused})$ — The final prediction $(\mathcal{F}_{final})$.
11. $\theta \leftarrow \theta - \ell r \triangle_\theta \mathcal{L}_{CE}(\mathcal{F}_{final}, \mathbb{Y})$ — Improvise weights to minimize loss $\mathcal{L}_{CE}$ via backpropagation.

12.	**for** β_i in β **do**	Loop through each β weighting parameter
13.	$\beta_i \leftarrow \beta_i - \ell r \triangle_{\beta_i} \mathcal{L}_{CE}(\mathcal{F}_{final}, \mathbb{Y})$	Update the β weighting parameters automatically via backpropagation.
14.	**end for**	
15.	**if** $e \,\%\, n = 0$ **then**	Decay learning rate after every 'n' epochs.
16.	$\ell r \leftarrow \ell r \times \gamma$	
17.	**end for**	
18.	**end for**	

Algorithm 2 presents the training process for the Face-NeSt model.

3.3.2.1 Baseline Architecture

The ResNet architecture [155], with its famous skip connection blocks, has proven extremely powerful for image classification tasks. Several improvements to ResNet have been proposed recently [156, 157, 158]. The architecture in [158] is taken as the baseline architecture for this experiment due to its combined strength of multi-path and split-attentional feature representation capability.

Specifically, a feature map is divided into G groups, given by the 'cardinality' hyperparameter K, and each group is further divided into split attentional subgroups given by the 'radix' hyperparameter R. The relation of G with K and R is given by $G = KR$. Split attention within a cardinal group [158] comprises global average pooling s_k^k as shown in Eq. 6.

The weighted fusion of a cardinal group V^k as shown in Eq. 7 with soft attention along the channel dimension as shown in Eq. 8.

$$s_c^k = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \widehat{U}_c^k\,(i,j) \tag{6}$$

$$V_c^k = \sum_{i=1}^{R} a_i^k(c) U_{R(k-1)+i} \tag{7}$$

$$a_i^k(c) = \begin{cases} \dfrac{e^{G_i^c(s^k)}}{\sum_{j=1}^{R}\left(e^{G_j^c(s^k)}\right)}, if\ R > 1 \\ \dfrac{1}{1+e^{-G_i^c(s^k)}}, if\ R = 1 \end{cases} \tag{8}$$

A single block contains the cardinal groups concatenated along the channel dimension $Concat(V^1, V^2, V^3 \ldots V^k)$ and then finally integrated as a shortcut connection $Y = V + X$.

Choice of Multi-Scale Features: The baseline architecture used in this experiment [158] forms the basis for selecting the type and number of multi-scale features. Specifically, this

baseline produces different scales of features from its distinct layers. For an input image of size $3 \times 128 \times 128$, the initial layers having convolution, batch norm and ReLU activation produce an output shape of $64 \times 32 \times 32$ which is the first scale of features. Next, the three skip connection modules produce output shapes of $256 \times 32 \times 32$, $512 \times 16 \times 16$ and $1024 \times 8 \times 8$ respectively, thereby forming three more feature scales. The fourth skip connection block of the baseline is discarded in the proposed model. Hence, based on the feature size produced by the baseline model, these four scales are used as the multi-scale features in the proposed model.

3.3.2.2 Adaptive Weighting of Multi-Scale Attentional Features

The main novelty of Face-NeSt is its ability to select an ideal proportion of multi-scale attentional features automatically. This section describes the 'adaptively weighted multi-scale attentional' module. The light-green box in Fig. 18 represents this novel module.

Global-Local-Channel-Spatial (GLCS) Attention

The research in computer vision has been largely boosted by attention mechanisms in recent years. However, most attention-based implementations focus only on a small subset of the input. The attention mechanisms either focus on channel or spatial attention across feature dimensions. Similarly, the attention mechanism is either used locally or globally.

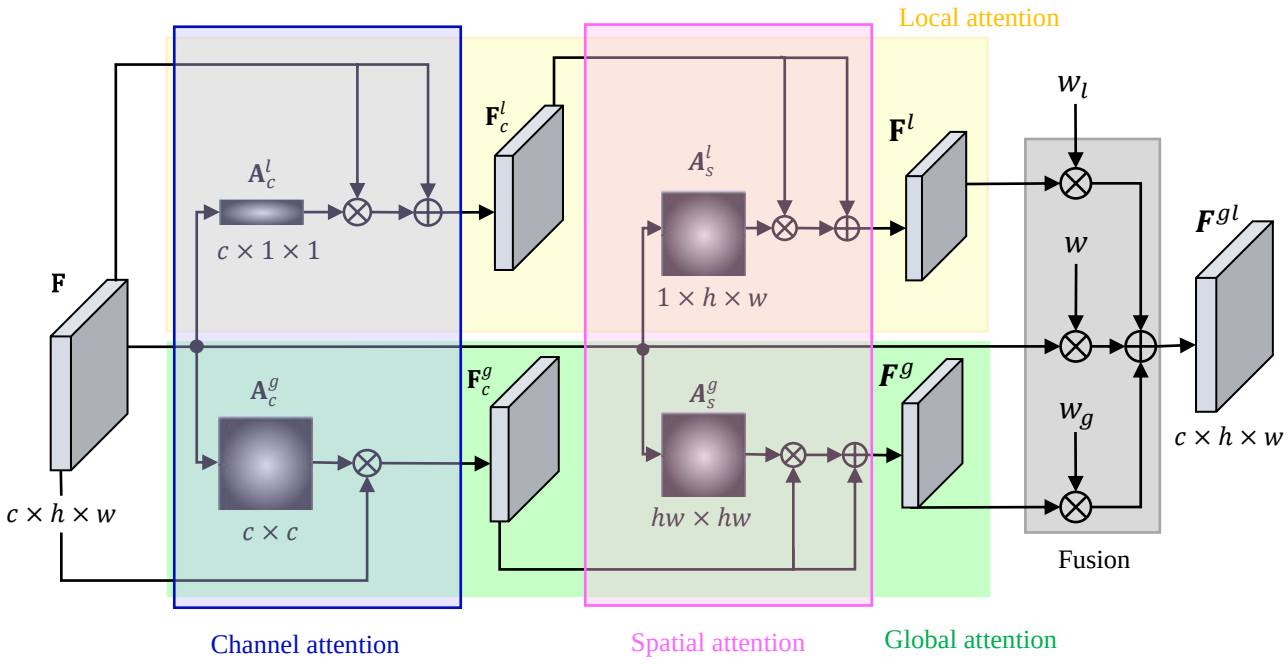


Fig. 19 Global Local Channel Spatial Attention Block [159]

The attention mechanism from [159] has been used in Face-NeSt to aggregate spatial and channel attention both locally and globally. Fig. 19 shows how input features are transformed using four forms of attention: channel-wise, spatially, locally, and globally. The size of input feature (F) is $c \times h \times w$.

Firstly, a local channel attention map (A_c^l) is computed from (F) using the global average pooling $GAP(.)$, 1×1 convolution $Conv_{1\times1}(.)$ and sigmoid $Sig(.)$ operation as shown in Eq. 9. Then using this local channel attention map (A_c^l), the local channel attentional features (F_c^l) are computed as shown in Eq. 10.

$$\mathrm{A}_c^l = Sig(Conv_{1\times1}(GAP(\mathrm{F})) \tag{9}$$

$$\mathrm{F}_c^l = \mathrm{F} + (\mathrm{F} \times \mathrm{A}_c^l) \tag{10}$$

Secondly, a global channel attention map (A_c^g) is computed from (F) using the same functions as shown in Eq. 11. The global channel attentional features (F_c^g) are computed as shown in Eq. 12.

$$\mathrm{A}_c^g = \mathrm{F} \times (Softmax(Sig(Conv_{1\times1,1}\big(GAP(\mathrm{F})\big) \times Sig(Conv_{1\times1,2}\big(GAP(\mathrm{F})\big))) \tag{11}$$

$$\mathrm{F}_c^g = \mathrm{F} \times \mathrm{A}_c^g \tag{12}$$

Thirdly, the local spatial attention map (A_s^l) is computed from (F) using the following equations. This is then used to produced local spatial attentional features (F^l).

$$\mathrm{F}' = Conv_{1\times1}(\mathrm{F}) \tag{13}$$

$$\mathrm{A}_s^l = Conv_{1\times1}(\sum(Conv_{3\times3}(\mathrm{F}'), Conv_{5\times5}(\mathrm{F}'), Conv_{7\times7}(\mathrm{F}'))) \tag{14}$$

$$\mathrm{F}^l = \mathrm{F}_c^l + (\mathrm{F}_c^l \times \mathrm{A}_s^l) \tag{15}$$

Fourthly, the global spatial attention map (A_s^g) is computed to calculate the global spatial attentional features (F^g) as shown in the following equations.

$$\mathrm{A}_s^g = Conv_{1\times1}(Conv_{1\times1}(\mathrm{F}) \times \Big(Softmax\big(Conv_{1\times1}(\mathrm{F}) \times Conv_{1\times1}(\mathrm{F})\big)\Big)) \tag{16}$$

$$\mathrm{F}^g = \mathrm{F}_c^g + (\mathrm{F}_c^g \times \mathrm{A}_s^g) \tag{17}$$

Finally, the local attentional features (F^l) and global attentional features (F^g) are multiplied with weighted factors and fused together to compute the final attentional features attended both locally and globally across channels as well as the spatial dimension (F^{gl}) as shown in Eq. 18.

$$\mathrm{F}^{gl} = (w \times \mathrm{F}) + (w_l \times \mathrm{F}^l) + (w_g \times \mathrm{F}^g) \quad (18)$$

When integrated with the proposed novel multi-scale attentional block, this attention mechanism produces multi-scale attentional feature extraction capabilities. The local channel attention uses global average pooling and 1D convolution operation to generate a local channel attention map A_c^l. Local spatial attention uses convolutional filters of size 3×3, 5×5 and 7×7 to extract spatial information at different scales (A_s^l). The global channel and spatial attention both utilize non-local filtering to obtain a global attention map in the channel and spatial dimension. Output features are a weighted average of the original, local, and global branch features, which have undergone summation operation. This GLCS module is employed within the multi-scale attentional block of the proposed model.

Adaptively Weighted Multi-Scale Attentional Module

Face-NeSt's main novelty lies in its ability to extract an ideal proportion of multi-scale attentional features automatically. To this end, a novel 'adaptively weighted multi-scale attentional' module is intended to teach the optimal percentage of discriminative characteristics at multiple scales. (Fig. 18). The GLCS attention module is utilised to improve the model's performance even further. Four features of different spatial resolutions are extracted from the baseline architecture and input to this block. Each scale feature is then passed through individual GLCS attention modules to highlight the important channel and spatial regions both locally and globally.

The 'Reduce & Weight' operation enforces spatial reduction followed by the adaptive weighting mechanism for features from each scale, as shown in (19).

$$\mathcal{F}_{i,weighted-attentional} = \mathbb{MUL}(\emptyset(\mathcal{F}_i), \beta_i) \quad (19)$$

Here, $\mathcal{F}_{i,weighted-attentional}$ represents the adaptively weighted attentional feature at a given scale 'i'. $\mathcal{F}_i$ represents the input feature at scale 'i'. $\emptyset(.)$ represents the spatial reduction operation implemented as a two-dimensional MaxPool operation. β_i is the weighting parameter for feature at scale 'i', which is used to extract a certain proportion of features. $\mathbb{MUL}(.)$ represents the multiply operation.

Specifically, the features extracted at four scales $[\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4]$ are first passed through separate GLCS attentional modules to find important spatial and channel regions both locally and globally. Next, the multi-scale features are proportioned with the weighting parameters $[\beta_1, \beta_2, \beta_3, \beta_4]$ giving weighted attentional features from multiple scales.

The weighting parameters are added to the computation graph of the proposed model and hence, are updated automatically via backpropagation. This automatic updation mechanism allows Face-NeSt to change the weighting parameters, allowing for the dynamic selection of the multi-scale attentional features. Each weighting parameter is initialized to 0.25 and as the network is trained, the individual β parameters get updated according to the importance of attentional features from each scale.

Fig. 18 shows the multi-scale features extracted from the baseline architecture. β_1 is the weighting factor for features of size $64 \times 32 \times 32$, β_2 is the weighting factor for features of size $256 \times 32 \times 32$, β_3 is the weighting factor for features of size $512 \times 16 \times 16$ and β_4 is the weighting factor for features of size $1024\times 8 \times 8$.

Finally, the 'adaptively weighted multi-scale attentional features' are concatenated along the channel dimension and passed through a 1×1 convolutional layer to reduce the feature map depth. Algorithm 2 presents the pseudocode of Face-NeSt with its novel adaptive weighting mechanism.

3.3.2.3 Reduced Computational Complexity

The baseline architecture contains the initial layers including a convolution layer, followed by a batch norm layer, relu activation, and maxpooling. Then there are four bottleneck blocks implementing skip connections with multi-path and split-attentional feature extraction. Finally, the adaptive average pooling and dense layers are attached. The total number of trainable parameters in this four-block model is 25.43 million. The last bottleneck block has been removed from the proposed model. This reduces the model size by removing more than half of the total trainable parameters making the proposed model computationally light-weight. This three-bottleneck architecture, along with the adaptively weighted multi-scale attentional block contains only 11.82 million parameters, making it light-weight compared to recent computer vision models.

3.3.2.4 Layer Details of Face-NeSt

This section details the layer-wise configuration of the proposed Face-NeSt architecture. Fig. 20 presents the layers in the proposed model. The input size is $32 \times 3 \times 128 \times 128$ for a batch of 32 images. The output shape and the number of parameters are given for each layer. The novel "Adaptively Weighted Multi-Scale Attentional Block" contains 2,545,528 parameters.

Layers	**Output shape** $(batch \times channel \times height \times width)$	**Number of Parameters**
Conv 2D	$32 \times 32 \times 64 \times 64$	864
BatchNorm 2D	$32 \times 32 \times 64 \times 64$	64
ReLU	$32 \times 32 \times 64 \times 64$	0
Conv 2D	$32 \times 32 \times 64 \times 64$	9,216
BatchNorm 2D	$32 \times 32 \times 64 \times 64$	64
ReLU	$32 \times 32 \times 64 \times 64$	0
Conv 2D	$32 \times 64 \times 64 \times 64$	18,432
BatchNorm 2D	$32 \times 64 \times 64 \times 64$	128
ReLU	$32 \times 64 \times 64 \times 64$	0
MaxPool 2D	$32 \times 64 \times 32 \times 32$	0
Skip Connection Block 1	$32 \times 256 \times 32 \times 32$	235,296
Skip Connection Block 2	$32 \times 512 \times 16 \times 16$	1,320,704
Skip Connection Block 3	$32 \times 1024 \times 8 \times 8$	7,696,640
Adaptively Weighted Multi-Scale Attentional Block	$32 \times 1024 \times 8 \times 8$	2,545,528
GlobalAvgPool2d	32×1024	0
Linear	32×2	2,050
	Total Parameters	**11,828,986**

Fig. 20 Layer details of the proposed Face-NeSt model.

3.3.3 Experimental Setup

This section explains the experimental steps taken to establish the validity of the proposed model.

3.3.3.1 Datasets and Classification Metrics

This section examines the most current datasets that are publicly available.

Celeb-DF Dataset: CelebDF [118] contains 590 original and 5639 deepfake videos having highly realistic manipulation quality. Original videos are recorded using 59 actors.

Deepfake Detection Challenge Dataset: The Deepfake Detection Challenge (DFDC) dataset [117] contains videos recorded by 66 actors, which are then processed by two unknown manipulations. There are a total of 5214 videos, and the ratio of original to tampered videos is 0.28:1.

FaceForensics++: The FaceForensics++ (FF++) [111] contains several manipulations such as Deepfakes [112], FaceSwap [113], Face2Face [114], FaceShifter [115], and Neural Textures [116]. Each manipulation category contains 1000 videos created from 1000 original samples. Videos are available in raw, high (c23), and low (c40) compression levels. In this experiment, c23 samples are utilised.

WildDeepFake: The WildDeepFake [135] dataset contains 7314 facial video clips derived from 707 deepfake videos collected from diverse sources across the web. This dataset is very challenging due to its diverse scenes and a rich variety of facial expressions.

Table XIII Classification metrics used to evaluate the proposed model.

Metric Name	Formula	Value Range
Accuracy (ACC)	$\frac{\mathbb{TP}+\mathbb{TN}}{\mathbb{TP}+\mathbb{TN}+\mathbb{FP}+\mathbb{FN}}$	[0,1]
Precision (P)	$\frac{\mathbb{TP}}{\mathbb{TP}+\mathbb{FP}}$	[0,1]
Recall (R)	$\frac{\mathbb{TP}}{\mathbb{TP}+\mathbb{FN}}$	[0,1]
F1 score (F1)	$2 * \frac{Precision * Recall}{Precision + Recall}$	[0,1]
Area Under Curve (AUC)	--	[0,1]
Mathews Correlation Coefficient (MCC)	$\frac{\mathbb{TP} * \mathbb{TN} - \mathbb{FP} * \mathbb{FN}}{\sqrt{(\mathbb{TP}+\mathbb{FP})(\mathbb{TP}+\mathbb{FN})(\mathbb{TN}+\mathbb{FP})(\mathbb{TN}+\mathbb{FN})}}$	[-1,1]

Classification Metrics: The above table lists the categorization metrics utilized in this experiment.

3.3.3.2 Hardware, Preprocessing, Hyperparameters and Weight Initialization

Hardware Specifications: All experimental tests are run in parallel on two NVIDIA GPUs, namely A5000 having 24GB memory each. System RAM is 128GB.

Preprocessing: This section describes the preprocessing steps followed in this experiment.

- Face Extraction: Popular deepfake detection algorithms have primarily used the dlib library [24, 28, 120] or MTCNN [121, 110, 122] for face detection and extraction. RetinaFace [34] is used in this experiment to extract facial images from video frames, given its low failure rate compared to MTCNN [119].

- Resizing, Normalization, and Data Augmentation: Facial images cropped from video frames are resized to 128×128. Pixel values are normalized to the range $[0,1]$. Facial images are flipped randomly in vertical and horizontal directions with a flipping probability of 0.5.

Hyperparameters and Training Conditions: All experiments are run for 30 epochs. The batch size is set to 4. The Adam optimizer is used to update the model weights. The initial learning

rate is set to 0.01. After every two epochs, the learning rate is decayed linearly by 10%. Table XIV presents size of train, validation and test sets in terms of the number of face images.

Table XIV Train, validation and test split size used in this experiment.

Dataset	Train Split	Validation Split	Test Split
Deepfakes (FF++)	48000	6400	9600
Face2Face (FF++)	48000	6400	9600
FaceShifter (FF++)	48000	6400	9600
FaceSwap (FF++)	48000	6400	9600
NeuralTextures (FF++)	48000	6400	9600
Celeb DF	160000	26432	22400
DFDC	112000	22400	27424

Model Weight Initialization: Deep models outperformed random weight initialization on classification tasks when trained with pre-learned ImageNet weights. Hence, the model weights in this experiment are initialized with ImageNet pre-trained weights for the ResNeSt architecture.

3.3.4 Experimental Results & Analysis

This section presents the experimental results for the proposed Face-NeSt model.

3.3.4.1 Face-NeSt Performance on the Benchmark Datasets

In this section, the performance scores achieved by the Face-NeSt model on the benchmark datasets has been presented. Table XV shows the performance on the CelebDF, FF++, WildDeepFake and DFDC benchmark datasets.

Table XV Results of Face-NeSt on three publicly available datasets, namely FF++, CelebDF and DFDC

Datasets		ACC	R	P	F1	AUC	MCC
Deepfakes	FF++ [111]	0.9805	0.9830	0.9769	0.9800	0.9945	0.9610
FaceShifter		0.9854	0.9761	0.9943	0.9851	0.9978	0.9711
Face2Face		0.9760	0.9844	0.9681	0.9762	0.9905	0.9510
FaceSwap		0.9779	0.9760	0.9812	0.9786	0.9948	0.9557
NeuralTextures		0.9128	0.9230	0.8907	0.9066	0.9548	0.8254
Celeb DF [118]		0.9612	0.9984	0.9585	0.9780	0.9823	0.8247
DFDC [117]		0.9742	0.9946	0.9728	0.9836	0.9947	0.9243
WildDeepFake [135]		0.9087	0.9148	0.9357	0.9251	0.9689	0.8745

Face-NeSt achieves more than 0.95 accuracy scores for most manipulation categories of FF++ on CelebDF and the DFDC dataset, indicating high classification capabilities for tampered facial images. Regarding AUC scores, Face-NeSt achieves more than 0.98 for all cases except NeuralTextures which is a challenging facial manipulation technique. These high AUC scores also imply that Face-NeSt performance is not vulnerable to class imbalance problems.

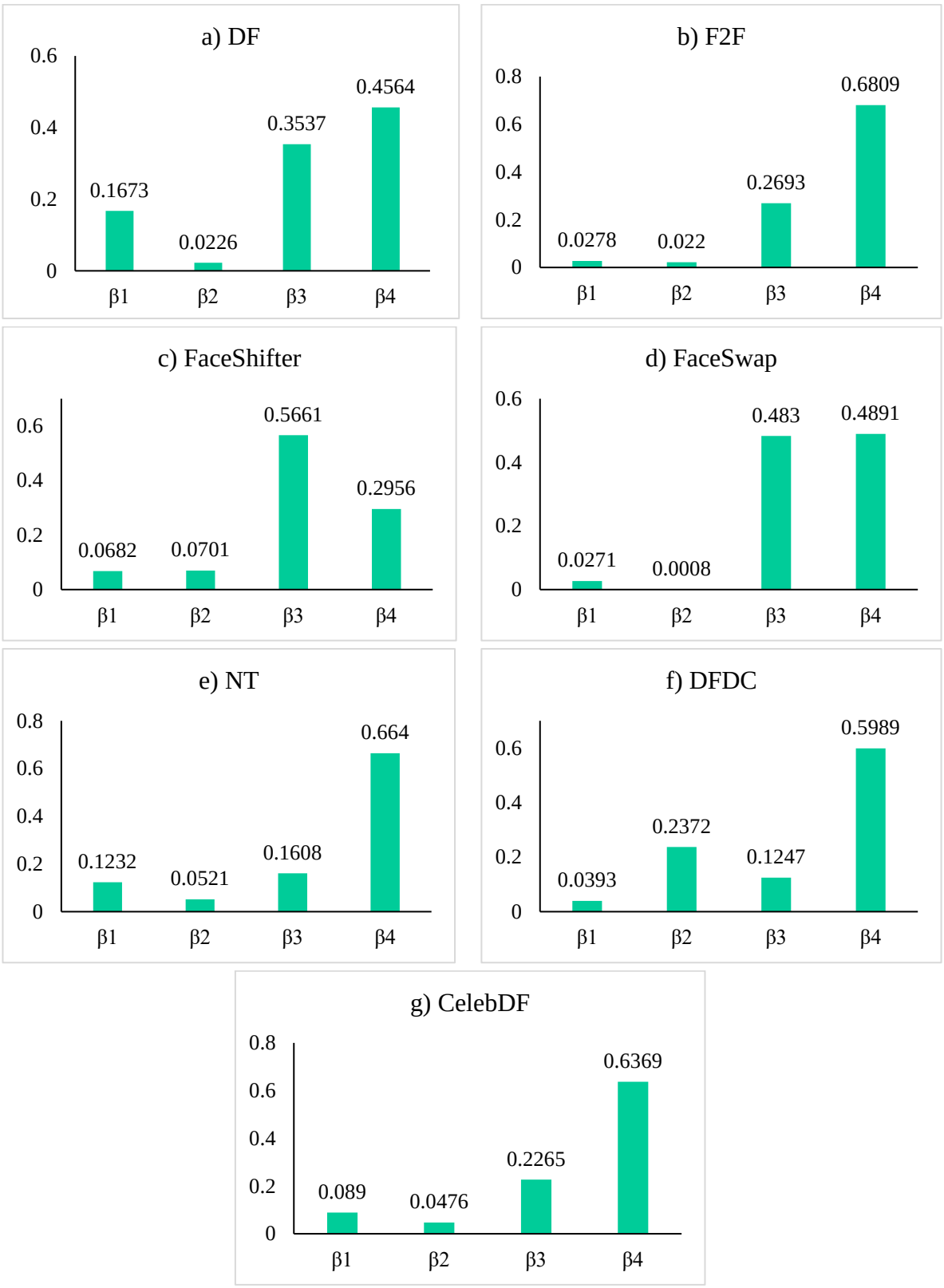


Fig. 21 The final β values for Face-NeSt on the benchmark datasets, a) DF b) F2F c) FaceShifter d) FaceSwap e) NT f) DFDC g) CelebDF.

MCC score measures the model's capability to identify both the positive as well as the negative classes. Except for the NeuralTextues and CelebDF datasets, Face-NeSt scores more

than 0.90 MCC, showing that it can recognise not just the positive class (tampered face photos), but also both classes.

Face-NeSt's adaptive weighting technique allows it to determine the appropriate proportion of multi-scale attentional characteristics. Each of the four weighting factors β_i are initialized to 0.25 at the beginning of the training procedure. As the model is trained, the β values get adjusted automatically via backpropagation. Fig. 21 presents the final β values on each benchmark dataset. The changed value of each β_i clearly indicates that the features from different scales hold different levels of importance to the final prediction.

In most cases, the value of $\beta_1$and β_2 is less than that of β_3 and β_4. This means that features corresponding to β_3 and β_4 are more important for the final prediction. It also implies that features extracted from deeper layers of the baseline network are more important than those from the initial layers.

Another observation is that β_4 obtains the maximum value (except in the case of FaceShifter) signifying the highest contribution in detecting facial manipulation. Fig. 21 demonstrates the dynamic nature of such a weighting mechanism that allows Face-NeSt to extract maximum discriminative information from multiple scales of features.

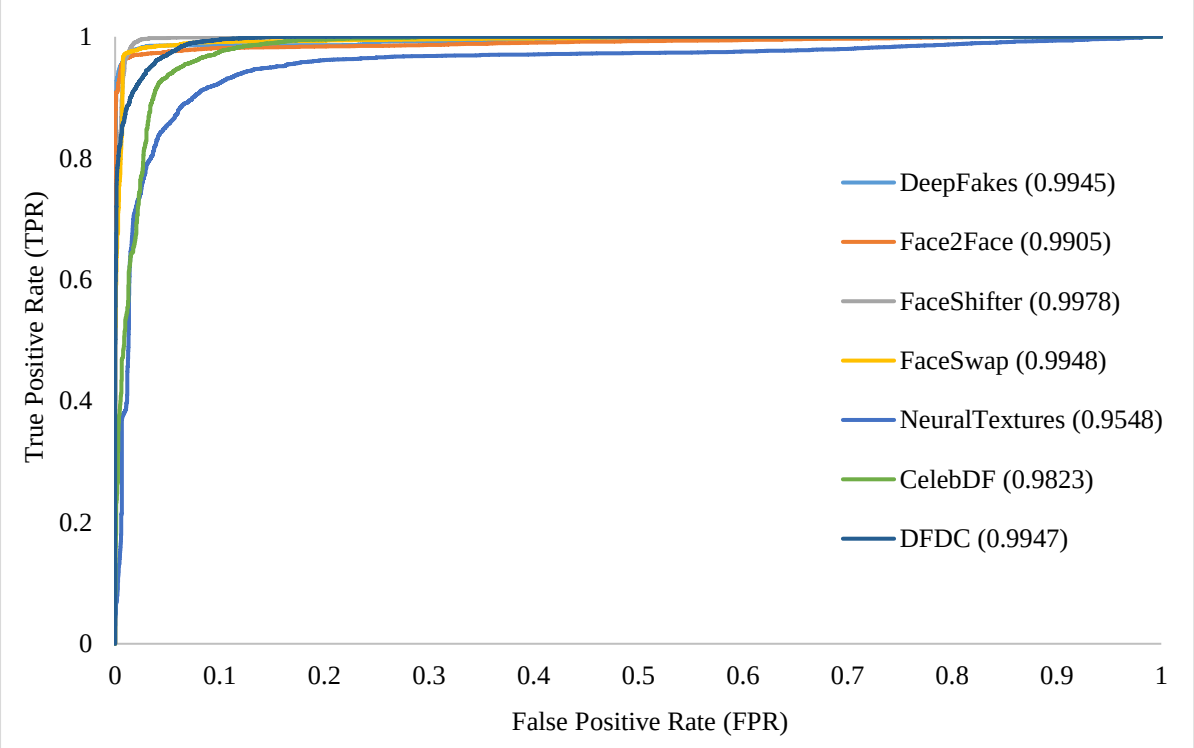


Fig. 22 AUC-ROC curves for Face-NeSt on the FF++, CelebDF, and DFDC datasets.

Fig. 22 shows the AUC-ROC curve for Face-NeSt on the DFDC, CelebDF and the FF++ datasets, indicating high values of True Positive Rate (TPR) and low False Positive Rate (FPR). This implies that Face-NeSt has high confidence in its predictions.

3.3.4.2 Comparison of Face-NeSt Against the Existing State-of-the-Art Approaches

The performance of Face-NeSt for face tampering detection is compared against the recent state-of-the-arts in this section.

Table XVI presents a comparison of Face-NeSt against several recent state-of-the-arts in face tampering identification on the CelebDF dataset. Face-NeSt beats the state-of-the-art methods by scoring the highest accuracy and AUC scores.

Table XVI Face-NeSt result comparison on the CelebDF dataset.

Methods	Year	ACC	AUC
Guo et al. [124]	2023	-	0.6743
Ke et al. [138]	2023	0.8892	-
Guo et al. [27]	2023	-	0.6950
Xu et al. [126]	2023	-	0.9332
Li et al. [139]	2023	0.7169	0.7659
Nirkin et al. [160]	2022	-	0.6600
*AMTENnet [107]	2021	0.9254	0.8804
Li et al. [141]	2021	-	0.7600
Hu et al. [48]	2021	-	0.7884
Chen et al. [131]	2021	-	0.8765
Chen et al. [144]	2021	0.9575	-
Luo et al. [143]	2021	0.7435	0.9234
Trihn et al. [142]	2021	-	0.7176
Dang et al. [50]	2020	-	0.7120
Choi et al. [51]	2020	0.9200	0.9400
Hu et al. [161]	2021	0.8074	0.8700
Li et al. [145]	2020	-	0.7476
Face-NeSt (Proposed)	**-**	**0.9612**	**0.9823**

Table XVII shows the superiority of Face-NeSt on the DFDC dataset. It easily outperforms all the recent SOTA approaches of facial manipulation detection.

Table XVII Face-NeSt result comparison on the DFDC dataset.

Methods	Year	ACC	AUC
Guo et al. [124]	2023	-	0.9597
Yu et al. [148]	2023	0.9593	0.9896
Zhao et al. [149]	2023	0.9210	-
Ke et al. [138]	2023	0.9108	-
Yang et al. [125]	2023	-	0.9911
Lin et al. [97]	2023	-	0.8847
Guo et al. [27]	2023	-	0.9827
Heo et al. [150]	2023	-	0.9780
Xu et al. [126]	2023	-	0.8037
*AMTENnet [107]	2021	0.9139	0.9402

Methods	Year	ACC	AUC
Guo et al. [124]	2023	-	0.9597
Yu et al. [148]	2023	0.9593	0.9896
Zhao et al. [149]	2023	0.9210	-
Ke et al. [138]	2023	0.9108	-
Yang et al. [125]	2023	-	0.9911
Lin et al. [97]	2023	-	0.8847
Guo et al. [27]	2023	-	0.9827
Heo et al. [150]	2023	-	0.9780
Li et al. [141]	2021	-	0.7600
Luo et al. [143]	2021	0.8841	0.9496
Xu et al. [28]	2021	0.8453	0.9347
Qi et al. [42]	2020	0.6410	-
Li et al. [153]	2020	0.8511	-
Mittal et al. [53]	2020	-	0.8440
Montserrat et al. [17]	2020	0.9188	-
Chugh et al. [52]	2020	-	0.9160
Face-NeSt (Proposed)	-	**0.9742**	**0.9947**

Table XVIII shows that Face-NeSt outperforms all recent state-of-the-arts on different manipulation types of the FF++ dataset. Scores are compared based on accuracy and AUC metric.

Table XVIII Face-NeSt result comparison on the FF++ dataset.

Methods	Year	DeepFake		Face2Face		FaceShifter		FaceSwap		NT		Average	
		ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Yang et al. [123]	2023	-	-	-	-	-	-	-	-	-	-	-	0.8709
Guo et al. [124]	2023	-	-	-	-	-	-	-	-	-	-	-	0.9755
Lin et al. [97]	2023	-	-	-	-	-	-	-	-	-	-	0.9074	0.9486
Yang et al. [125]	2023	-	-	-	-	-	-	-	-	-	-	0.9382	0.9827
Xu et al. [126]	2023	-	-	-	-	-	-	-	-	-	-	-	0.9034
Yang et al. [127]	2022	-	-	-	-	-	-	-	-	-	-	-	0.7888
Nirkin et al. [160]	2022	0.9450	-	0.8030	-	-	-	0.8450	-	0.7400	-	-	-
Liu et al. [129]	2021	0.9348	-	0.8602	-	-	-	0.9226	-	0.7678	-	0.8713	-
Shang et al. [130]	2021	0.9563	-	0.9015	-	-	-	0.9493	-	0.8001	-	-	-
Chen et al. [131]	2021	-	0.9595	-	-	-	-	-	0.9787	-	-	-	-
Hu et al. [161]	2021	0.9464	0.9800	0.8648	0.9400	-	-	0.8527	0.9400	0.8005	0.9000	-	-
Qian et al. [133]	2020	0.9597	-	0.9532	-	-	-	0.9653	-	0.8332	-	0.9278	-

Methods	Year	DeepFake		Face2Face		FaceShifter		FaceSwap		NT		Average	
		ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Baek et al. [134]	2020	0.7180	-	0.6860	-	-	-	0.6310	-	0.7070	-	-	-
Zi et al. [135]	2020	0.9210	-	0.8390	-	-	-	0.9250	-	0.7820	-	-	-
Rössler et al. [111]	2019	0.7450	-	0.7590	-	-	-	0.7090	-	0.7330	-	-	-
Amerini et al. [26]	2019	-	-	0.8161	-	-	-	-	-	-	-	-	-
Afchar et al. [136]	2018	0.8727	-	0.5620	-	-	-	0.6117	-	0.4067	-	0.6132	-
Face-NeSt (Proposed)	-	**0.9805**	**0.9945**	**0.9760**	**0.9905**	**0.9854**	**0.9978**	**0.9779**	**0.9948**	**0.9128**	**0.9548**	**0.9665**	**0.9864**

Table XIX Face-NeSt result comparison on the WildDeepFake dataset.

Methods	Year	ACC	AUC
Zhao et al. [162]	2023	0.8332	-
Liu et al. [163]	2023	0.8559	-
Wang et al. [164]	2023	0.8441	0.9257
Shi et al. [165]	2023	0.8453	0.9327
Jin et al. [166]	2023	0.7855	0.8641
Khormali et al. [167]	2023	0.8137	0.8124
Sun et al. [168]	2023	-	0.8355
Sun et al. [169]	2023	0.8339	0.9040
Hu et al. [170]	2022	0.7588	0.8138
Gu et al. [171]	2022	0.8414	0.9162
Cao et al. [172]	2022	0.8325	0.9202
Qian et al. [133]	2020	0.8066	0.8753
Face-NeSt (Proposed)	-	**0.9087**	**0.9689**

Table XIX presents a comparison of Face-NeSt's performance against state-of-the-arts on the WildDeepFake dataset. As usual, Face-NeSt comfortably beats all the existing approaches for face manipulation detection.

3.3.4.3 Complexity Analysis of Face-NeSt

This section presents an analysis of the computational complexity of Face-NeSt and compares it against popular computer vision models. Complexity analysis has been done by comparing the inference time on GPU and CPU, the accuracy achieved on the DeepFake, the number of 'multiply-accumulate' (MACs) operations for each image input and the count of trainable parameters (millions).

Fig. 23 compares Face-NeSt against popular computer vision models. All models are trained on the Deepfake (FF++) dataset. The vertical axis represents the accuracy scores. The horizontal axis shows the number of trainable parameters in each network. Each circle's size indicates the MAC value for each model. Larger circles have a higher MAC value. Face-NeSt

achieves the highest accuracy scoring 0.9805 accuracy. Additionally, it is light-weight with just 11.82 million parameters.

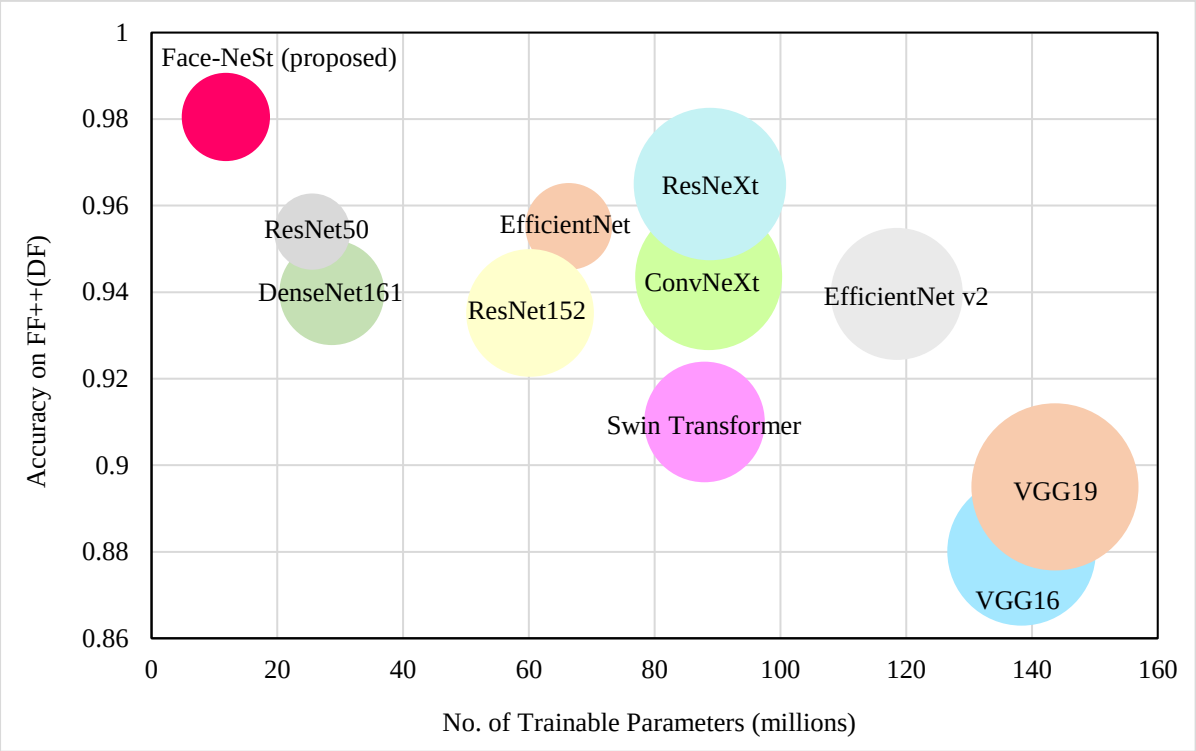


Fig. 23 Face-NeSt's complexity is compared to that of common computer vision models.

Table XX Face-NeSt's computational complexity is compared to prominent computer vision models.

Architectures	**Parameters (millions)**	**ACC**	**MACs operations ($\times 10^9$)**	**CPU inference time (seconds)**	**GPU inference time (seconds)**
EfficientNet [173]	66.34	0.9552	1.75	7.27	1.43
ConvNeXt [174]	88.59	0.9436	5.02	4.95	1.18
EfficientNet v2 [175]	118.51	0.9396	4.05	6.61	0.96
Swin Transformer [176]	87.93	0.9100	3.38	6.69	0.93
ResNet152 [155]	60.19	0.9352	3.79	4.41	0.32
VGG16 [177]	138.35	0.8801	5.13	3.50	0.35
VGG19 [177]	143.66	0.8950	6.49	4.25	0.37
DenseNet161 [178]	28.68	0.9399	2.56	3.65	0.56
ResNet50 [155]	25.55	0.9540	1.35	1.83	0.14
ResNeXt [156]	88.79	0.9650	5.40	4.75	0.87
Face-NeSt (Proposed)	11.82	0.9805	1.81	11.63	0.48

Face-NeSt easily outperforms the other heavier networks such as ResNeXt (88.79 million parameters), EfficientNet-v2 (118.51 million parameters), VGG16 (138.35 million parameters) and VGG19 (143.66 million parameters). Table XX presents the inference time of Face-NeSt on CPU and GPU for one batch of facial images.

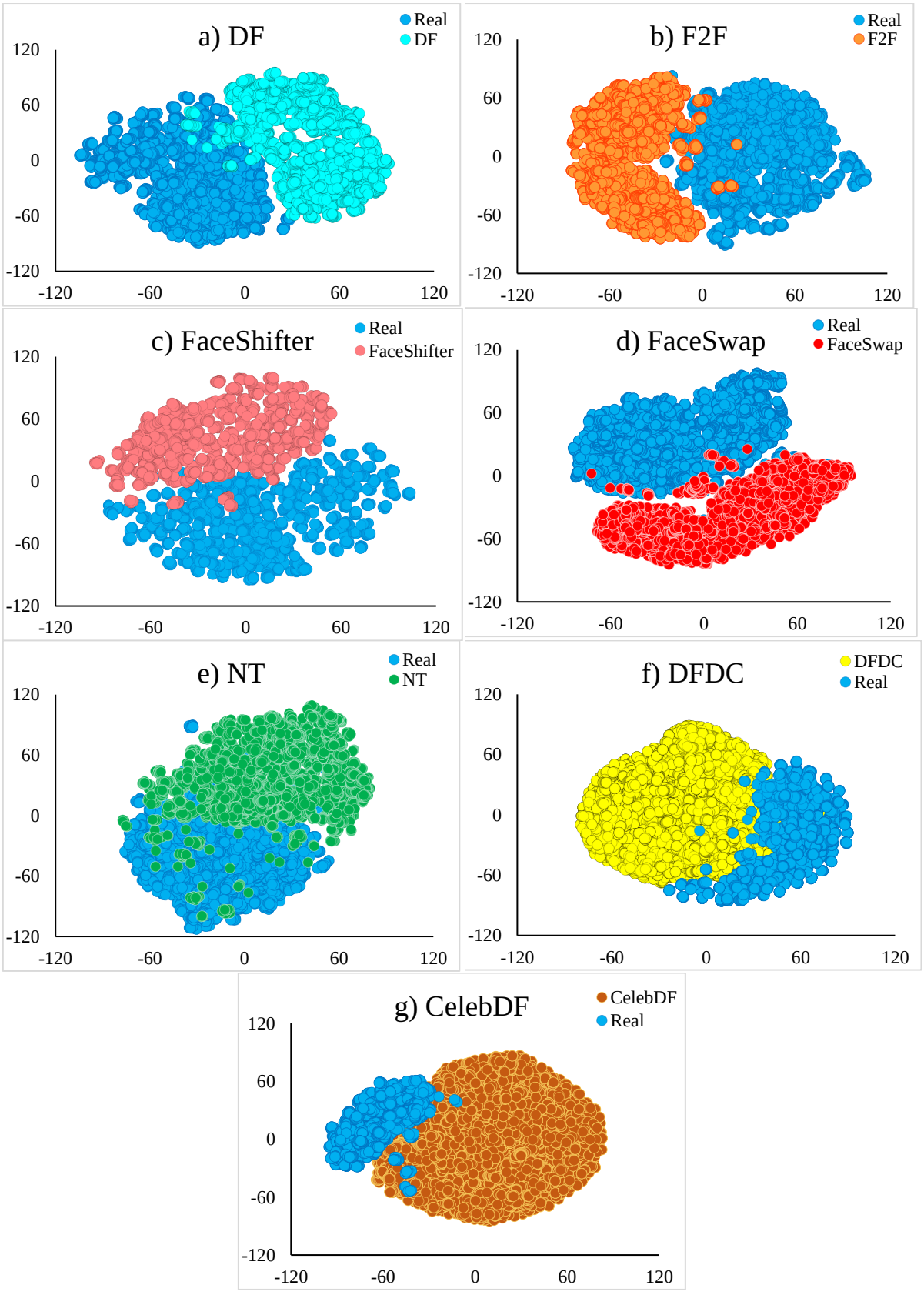


Fig. 24 The t-SNE visualizations for the Face-NeSt model on the benchmark datasets a) DF b) F2F c) FaceShifter d) FaceSwap e) NT f) DFDC g) CelebDF

3.3.4.4 Qualitative Analysis of Face-NeSt

This section presents a qualitative visualization of the Face-NeSt predictions by showing the t-SNE plots (Fig. 24) and the class activation map visualizations (Fig. 25).

Specifically, t-SNE visualization presents the discriminative feature extraction capabilities of the Face-NeSt model as shown in Fig. 24. It can be seen clearly that Face-NeSt extracts discriminative features for each manipulation type such that the plotted features from the test samples exist closely within their respective class clusters. There is some overlap only in the case of the Neural Texture (NT) dataset of FF++ which is a challenging case with realistic face tampering. Notably, for the DFDC and CelebDF datasets, the number of manipulated faces is far more than real face images, making them highly imbalanced datasets, which is challenging. However, Face-NeSt demonstrates strong classification capabilities on these datasets as well.

Fig. 25 shows the focus region for Face-Nest for the multi-scale features extracted by the baseline architecture (bottleneck 1, 2, and 3 layers) and the ‘adaptively weighted multi-scale attentional’ module. The maps are generated via the LayerCAM method [154].

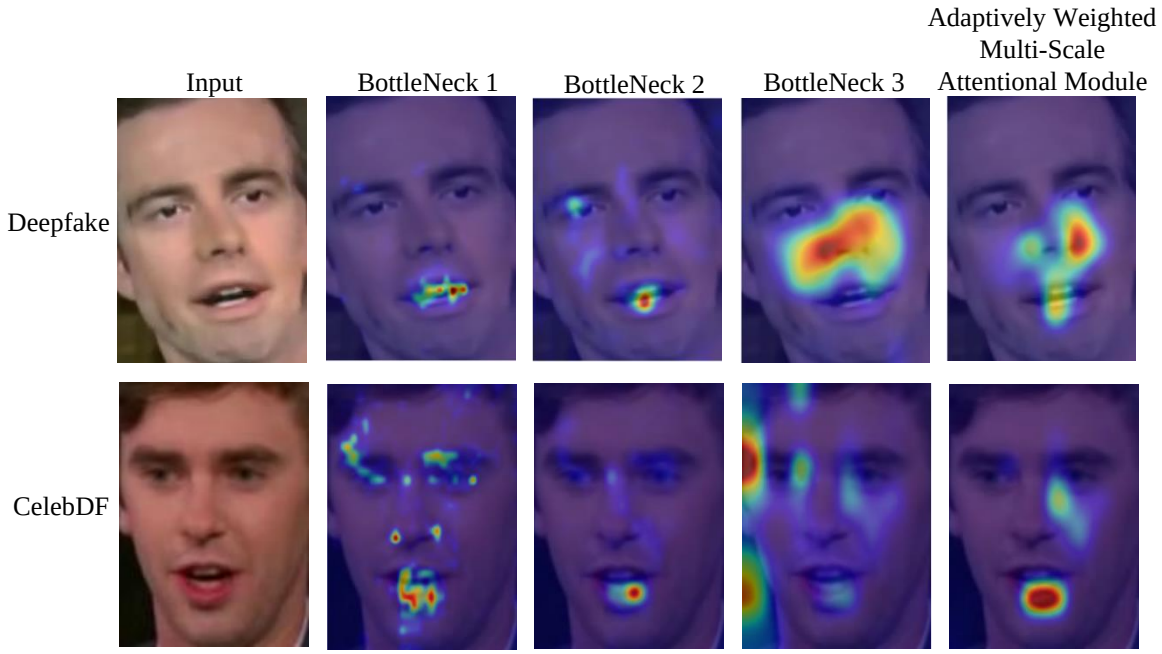


Fig. 25 The region of focus for Face-NeSt.

3.3.4.5 Generalization Study of Face-NeSt

This section evaluates the generalization ability of the proposed Face-NeSt model. Specifically, the five manipulation categories of the FF++ dataset, DeepFake (DF), Face2Face

(F2F), FaceShifter (FSh), FaceSwap (FSw) and NeuralTextures (NT) are used for the cross-dataset evaluation.

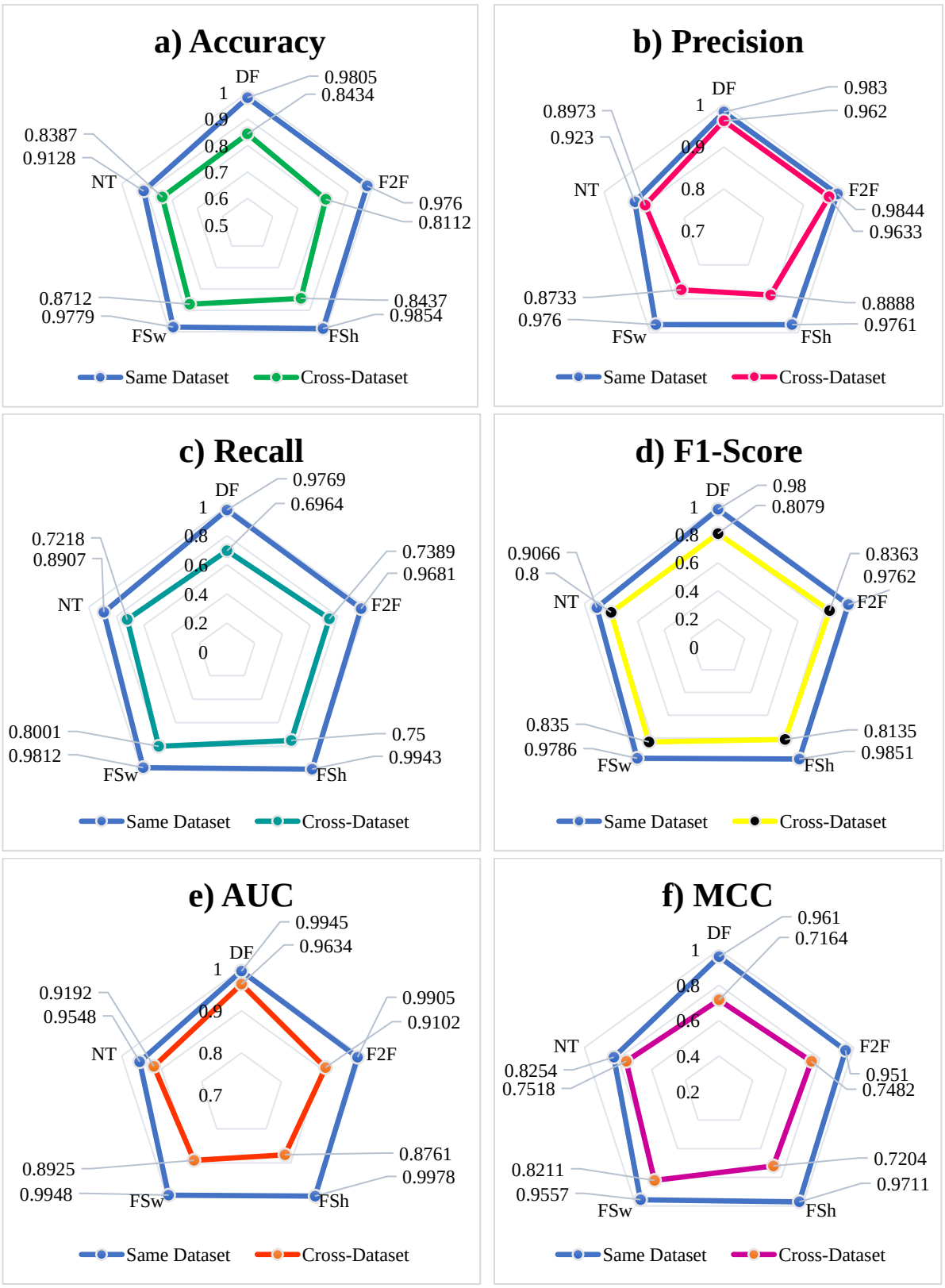


Fig. 26 Comparison of Face-NeSt performance for same and cross-dataset evaluation on metrics a) Accuracy b) Precision c) Recall d) F1-score e) AUC f) MCC score.

Table XXI Generalization Study of DenseTrace-Net on the FF++ dataset.

Train Datasets	Test Datasets	ACC	P	R	F1	AUC	MCC
F2F + FSh + FSw + NT	DF	0.8434	0.9620	0.6964	0.8079	0.9634	0.7164
DF + FSh + FSw + NT	F2F	0.8112	0.9633	0.7389	0.8363	0.9102	0.7482
DF + F2F + FSw + NT	FSh	0.8437	0.8888	0.7500	0.8135	0.8761	0.7204
DF + F2F + FSh + NT	FSw	0.8712	0.8733	0.8001	0.8350	0.8925	0.8211
DF + F2F + FSh + FSw	NT	0.8387	0.8973	0.7218	0.8000	0.9192	0.7518

Any four categories are selected as the combined training set while the fifth unseen category is used to test Face-NeSt's performance on unseen data. Table XXI presents the cross-dataset evaluation scores and Fig. 26 compares Face-NeSt's performance for the same and cross-dataset evaluation.

3.3.4.6 Ablation Study of Face-NeSt

This section includes an ablation investigation to validate the adaptive weighting mechanism's contribution to multi-scale attentional aspects in the Face-NeSt model. Specifically, Face-NeSt is trained on the DF and F2F categories of the FF++ dataset. The following cases are evaluated:

- Case A: The model is trained only on baseline architecture with no multi-scale feature learning.
- Case B: The model is trained with multi-scale feature learning.
- Case C: The model is trained with multi-scale feature learning boosted by the attention mechanism.
- Case D: The model is trained with non-adaptive weighting of attentional multi-scale features. This means that each of the β_i is set to a fixed value of 0.25 that does not change during training.
- Case E: (proposed model): The model is trained with an adaptive weighting of attentional multi-scale feature learning. This means that the β_i parameters are initialized to 0.25 at the beginning of the training and their values are adaptively changed during training due to backpropagation.

Fig. 27 shows the impact of each enhancement made. The increasing accuracy and MCC scores demonstrate the obvious benefit of extracting adaptively weighted multi-scale attentional features for the final prediction.

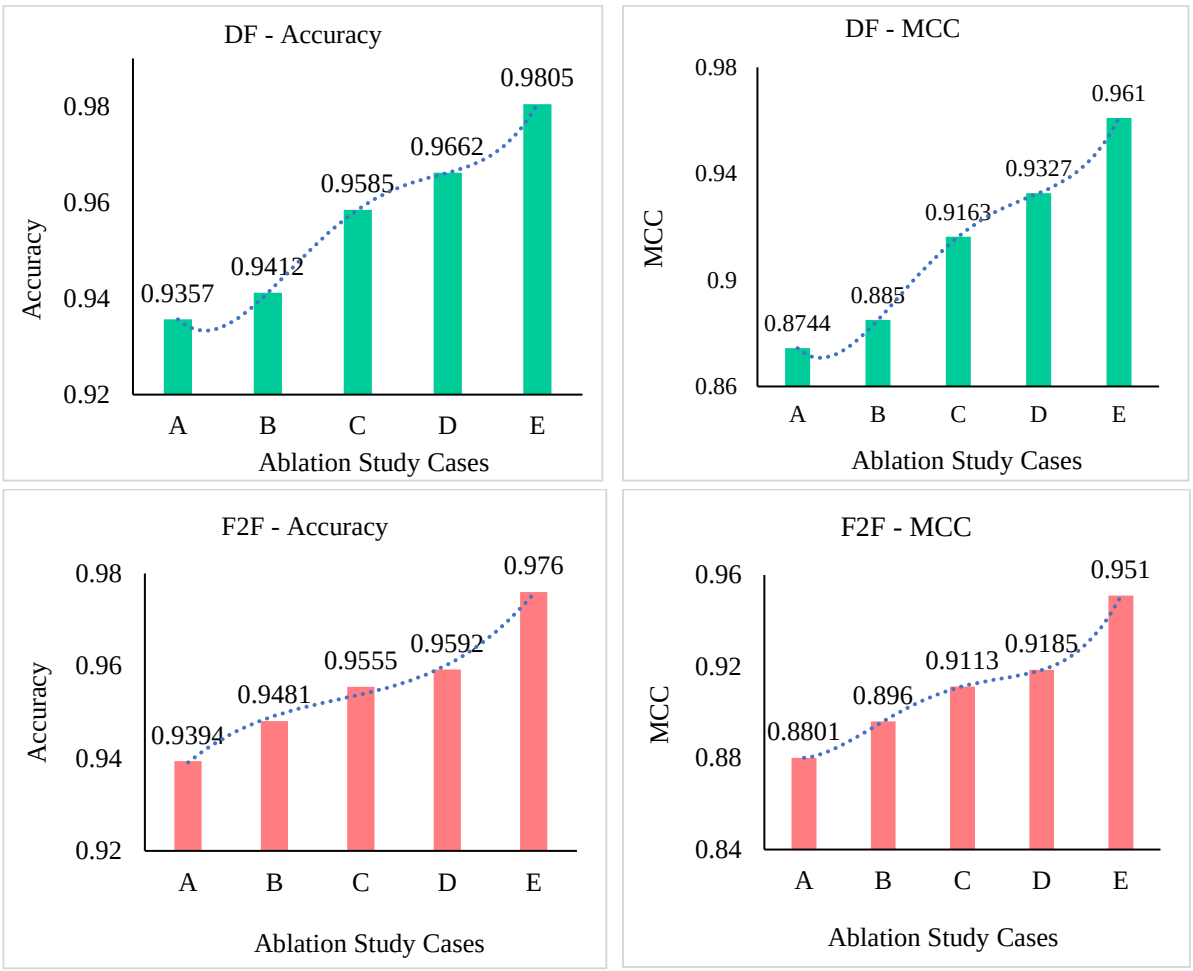


Fig. 27 Ablation study results for Face-NeSt.

Table XXII Ablation study scores for Face-NeSt model.

	Ablation Study Cases	Case	ACC	P	R	F1	AUC	MCC
DF [111]	Without Multi-Scale Features	A	0.9357	0.9024	**0.9848**	0.9418	0.9906	0.8744
	With Multi-Scale Features	B	0.9412	0.9108	0.9795	0.9439	0.9885	0.8850
	With Multi-Scale Attentional Features	C	0.9585	0.9516	0.9597	0.9556	0.9906	0.9163
	With Non-Adaptively Weighted Multi-Scale Attentional Features	D	0.9662	0.9523	0.9793	0.9656	0.9926	0.9327
	With Adaptively Weighted Multi-Scale Attentional Features	E	**0.9805**	**0.9830**	0.9769	**0.9800**	**0.9945**	**0.9610**
F2F [111]	Without Multi-Scale Features	A	0.9394	0.9763	0.8884	0.9303	0.9869	0.8801
	With Multi-Scale Features	B	0.9481	0.9493	0.9514	0.9503	0.9876	0.8960
	With Multi-Scale Attentional Features	C	0.9555	0.9468	0.9675	0.9570	0.9877	0.9113
	With Non-Adaptively Weighted Multi-Scale Attentional Features	D	0.9592	0.9625	0.9536	0.9581	0.9882	0.9185
	With Adaptively Weighted Multi-Scale Attentional Features	E	**0.9760**	**0.9844**	**0.9681**	**0.9762**	**0.9905**	**0.9510**

Table XXII presents detailed scores of the conducted ablation study. Case D containing adaptive weighting of multi-scale attentional features outperforms all the other cases.

3.3.5 Conclusion

Face-NeSt, a revolutionary face tampering detection model, is proposed in this work. The Face-NeSt model's key innovation is its capacity to adaptively weight multi-scale attentional characteristics in proportion to their value to the final prediction. Face-NeSt contains a novel 'adaptively weighted multi-scale attentional' module that performs this dynamic weighting of multi-scale features. An attention mechanism helps to capture important spatial and channel regions of multi-scale features both locally and globally. Face-NeSt performs highly on three public benchmark datasets, FF++, CelebDF, and DFDC. Face-NeSt achieves excellent AUC of 0.9947 on DFDC, 0.9823 on CelebDF, 0.9945 on DeepFake (FF++), 0.9905 on Face2Face (FF++), 0.9978 on FaceShifter (FF++), 0.9948 on FaceSwap (FF++) and 0.9548 on NeuralTextures (FF++) beating all recent state-of-the-art face tampering detection methods.

3.4 Significant Outcomes of this Chapter

The significant outcomes of this chapter are as follows:

- Proposed two novel deep-learning architectures, *MRT-Net* and *Face-NeSt*, for face manipulation detection in images.
- MRT-Net contains an intelligent auto-adaptive mechanism that automatically chooses the best proportion of manipulation residuals and textural features to detect facial manipulation. It is also aided by a channel attention mechanism.
- Face-NeSt is an adaptive multi-scale feature extractor model that chooses different scales of features in proportion to their relevance to the final prediction. It contains a novel 'adaptively weighted multi-scale attentional' (AW-MSA) module.
- Experimental results on three public benchmark face manipulation datasets, namely, FaceForensics++, CelebDF and DFDC, prove that both MRT-Net and Face-NeSt easily outperform existing state-of-the-art methods.

The following research works form the basis of this chapter:

- **A. Yadav** and D. K. Vishwakarma, "MRT-Net: Auto-adaptive weighting of manipulation residuals and texture clues for face manipulation detection," **Expert Systems with Applications**, vol. 232, 2023.
- **A. Yadav** and D. K. Vishwakarma, "AW-MSA: Adaptively weighted multi-scale attentional features for DeepFake detection," **Engineering Applications of Artificial Intelligence**, vol. 127 Part B, 2024.

Chapter 4: Splice Manipulation Detection and Localization in Images

4.1 Scope of this Chapter

This chapter is dedicated to the problem of detecting splice manipulation in images. To this end, two contributions have been proposed. In the first research work, a novel image splice detection dataset, *BiometricLab-DTU-Splice Dataset* is proposed. The proposed dataset contains two variants of spliced samples generated from Python code and Adobe Photoshop software, respectively. Binary masks are also provided in the proposed dataset. Additionally, a novel lightweight, dual-branch, information-preserving, spatial-compression modal splice detection framework is proposed to detect spliced jpeg images while restricting the computational complexity. The second research work is dedicated to the localization of splice manipulation in images. Specifically, a novel, visually attentive splice localization model with a multi-domain feature extractor and multi-receptive field upsampler is proposed. A "visually attentive multi-domain feature extractor" (VA-MDFE) extracts attentional features from the RGB, edge and depth domain of input images. Next, a "visually attentive downsampler" (VA-DS) is responsible for fusing the multi-domain features and downsampling them. Lastly, a "visually attentive multi-receptive field upsampler" (VA-MRFU) upsamples features using multiple receptive fields during the convolution operation. Experimental results clearly indicate the superiority of the proposed splice localization model against the existing state-of-the-art methods.

4.2 Towards Effective Image Forensics via A Novel Computationally Efficient Framework and A New Image Splice Dataset

4.2.1 Abstract

Splice detection models are the need of the hour since splice manipulations can be used to mislead, spread rumours and create disharmony in society. However, there is a severe lack of image-splicing datasets, which restricts the capabilities of deep learning models to extract discriminative features without overfitting. This research work presents two-fold contributions toward splice detection. Firstly, a novel splice detection dataset with two variants is proposed. The two variants include spliced samples generated from code and through manual editing.

Spliced images in both variants have corresponding binary masks to aid localization approaches. Secondly, a novel *Spatio-Compression Lightweight Splice Detection Framework* is proposed for accurate splice detection with minimum computational cost. The proposed dual-branch framework extracts discriminative spatial features from a lightweight spatial branch. It uses original resolution compression data to extract double compression artifacts from the second branch, thereby making it 'information preserving.' Several CNNs are tested in combination with the proposed framework on a composite dataset of images from the proposed dataset and the CASIA v2.0 dataset. The best model accuracy of 0.9382 is achieved, beating all state-of-the-art methods and demonstrating its superiority.

4.2.2 Proposed Splice Detection Dataset

One of the critical challenges in splice detection is the lack of large-scale splice datasets. Table XXIII provides a list of existing splice detection datasets. Most of the existing splice datasets are limited in terms of the number of samples, and not all include binary masks for localization implementations. Training deep models on small datasets inevitably presents the problem of overfitting.

Table XXIII Comparison of proposed splice detection dataset with existing splice datasets.

Ref.	Year	Dataset	Tampering Type	Original Samples	Spliced Samples	Resolution	Format	Splice Masks
[179]	2004	Columbia Gray	Splicing	933	912	128 x 128	BMP	
[180]	2006	Columbia Color	Splicing	183	180	757 x 568, 1152 x 768	TIFF	Yes
[181]	2009	CASIA v1.0	Splicing	800	921	384 x 256	JPG	No
[181]	2009	CASIA v2.0	Splicing	7491	5123	240 x 160, 900 x 600	TIFF, BMP, JPG	No
[182]	2013	DSO-I	Splicing	100	100	2048 x 1536	PNG	-
[182]	2013	DSI-I	Splicing	25	25	Variable	-	-
[183]	2014	Image Forensic Dataset Challenge	Splicing	144	144	2018 x 1536	PNG	-
[184]	2015	SYSU-OBJFORG dataset	Copy Move, Splicing	100	100	1280 x 720 @ 25fps	H.264 / AVC	-
[185]	2015	GRIP	Splicing	80	80	1024*768	JPG	Yes
-	-	**Biometric-Lab-DTU Splice Dataset – automatic (proposed)**	Splicing	8156	8156	3008 x 2000, 4288 x 2848, 4928 x 3264	JPG	Yes
-	-	**Biometric-Lab-DTU Splice Dataset – manual (proposed)**	Splicing	3106	3106	3008 x 2000, 4288 x 2848, 4928 x 3264	JPG	Yes

To this end, a novel splice detection dataset having two variants is proposed – BiometricLab-DTU-Splice Dataset . The first variant (automatic) is autogenerated from python code. The

second variant contains spliced images prepared in Adobe Photoshop Software. Fig. 28 demonstrates the proposed dataset's original, spliced, and binary mask images.

Fig. 28 Samples from the proposed BiometricLab-DTU Splice dataset

4.2.2.1 BiometricLab-DTU-Splice Dataset (Automatic)

The 'automatic' variant of the BiometricLab-DTU Splice dataset is autogenerated through Python code. 8156 high-resolution uncompressed images from the RAISE dataset [186] are used as source images. Several existing splice detection approaches have prepared datasets by

compressing images with quality factors of step size 5 [70], [67]. The jpeg images of the proposed dataset are produced with a random integer quality factor which ensures a richer distribution of compression information compared to the above-described case.

Dataset Generation of Automatic Spliced Images: A pre-trained RCNN model trained on the MS-COCO dataset extracts objects from the original images of the RAISE dataset. This is achieved from the instance_segmentation() function of the PixelLib python library. This function extracts different objects detected as the result of segmentation. Next, the binary masks are generated for each object by converting the object images to grayscale and thresholding into binary images. The extracted objects are tampered with several random manipulations, including rotation, scaling, flipping, contrast changes, brightness variations and sharpness modifications before pasting onto another original image. The degree of scaling, rotation and other manipulations is random as shown in Fig. 29. Finally, the extracted objects are pasted onto original images using the paste() function of the Python's PILLOW library. Lastly, the spliced images are saved as jpeg images with random compression quality factor as shown in the "second compression" column of Fig. 29. The image reading and color-scale modifications are done using the open-cv library, while the rotation, scaling, flipping, contrast and sharpness enhancements are achieved with the PILLOW library. The binary masks produced from code can aid in future splice localization methods.

name	first compression	second compression	object name	scale	rotation	flip	contrast	brightness	sharpness
r00816405t_spliced.jpg	94	73	object16_r0fb0a690	1	28	0	1.73	1.29	1.81
r00869422t_spliced.jpg	53	61	object2_rccf44639t	0.87	132	1	1.81	1.16	1.81
r00879054t_spliced.jpg	87	75	object2_rb6c0981e	0.94	109	1	1.82	1.29	1.55
r01058910t_spliced.jpg	63	53	object11_r854fa219	0.94	110	0	1.81	1.32	1.75
r01170470t_spliced.jpg	35	53	object3_r39a8b2f2t	1	69	1	1.74	1.35	1.69
r01474187t_spliced.jpg	51	34	object3_r61184440	1	176	1	1.58	1.16	1.95
r0150031ft_spliced.jpg	74	63	object5_r319e1687	1	149	1	1.65	1.27	1.86

Fig. 29 Parameters used during creation of spliced samples.

8156 high-resolution uncompressed images from the RAISE dataset were used to produce 8156 original and 8156 spliced images, which form the BiometricLab-DTU Splice dataset (automatic). Fig. 28 shows some samples from the proposed automatic dataset variant.

4.2.2.2 BiometricLab-DTU-Splice Dataset (Manual)

The second variant of the proposed dataset has been prepared manually. A total of 3106 images were spliced manually along with the same number of the original counterparts to formulate a balanced dataset of 6212 images. Spliced samples from the manual variant of the proposed dataset are more realistic visually than the automatic version due to the random

manipulations on pasted objects of computer-generated spliced images in the automatic variant dataset. Table XXIII compares the proposed BiometricLab-DTU Splice dataset against existing splice datasets. Most existing datasets fall short in terms of the total number of samples available, have smaller resolution images, and usually don't have splice masks to aid splice localization approaches when compared against the proposed dataset variants.

Dataset Generation of Manually Spliced Images: The source and target images are opened in the Adobe Photoshop software. An object is selected from the source image using the 'Quick Selection Tool'. The selected object is pasted onto the target image as a new layer. Resizing, rotation, flipping, contrast and brightness modifications are made randomly to the object layer. The pasted object is selected again and the 'Layer Mask Tool' in Photoshop generates the binary mask layer. Next, a 'Photoshop Action' is created that automates the process of saving the spliced image and its corresponding binary mask into separate folders. Specifically, the spliced image is saved as a jpeg image, while the binary mask is generated from the mask layer formed by the layer mask tool. For each image, opening the source and target images, selecting the object to be pasted and pasting operations are done manually. Then, the Photoshop action helps to save the spliced images and corresponding binary masks automatically.

4.2.3 Proposed Splice Detection Framework

The design of the proposed splice detection framework is motivated by several factors. Firstly, while spatial features are the most common type of deep features extracted, they do not provide the most discriminative information about spliced samples. Splice detection in different modalities, including frequency domain, noise domain, etc., has proven effective. Secondly, the small number and size of publicly available splice datasets (Table XXIII) provide a clear incentive to avoid heavy deep learning architectures having millions of trainable parameters. A deep model that is too complex for a given dataset will overfit and memorize the training samples. Thirdly, since deep learning architectures require fixed-size inputs, images are mainly resized to smaller resolutions, and hence there is 'information loss' due to the reduction of high-dimensional images. Hence, a splice detection framework is proposed with the following novel characteristics:

- **Dual-branch for Multi-Modal Feature Learning** – Different modalities besides the spatial domain have proven effective for splice detection. This proposed framework combines the spatial domain with a 'compression branch' that learns discriminative compression artifacts indicating image splicing.

- **Information Preserving** – The compression branch of the proposed framework extracts compression artifacts from original resolution image data. Hence, no information is lost due to resizing.
- **Lightweight** – The design of the proposed splice detection framework restricts the number of trainable parameters in the context of deep learning. None of the proposed framework variants have more than 100,000 trainable parameters, whereas standard deep networks can easily have up to millions of trainable parameters. Fewer trainable parameter leads to significantly lower computational cost for the proposed splice detection framework.
- **Futuristic** – A novel framework is proposed instead of building a fixed architecture for splice detection. The proposed framework supports a variety of existing deep architectures and is also capable of utilizing novel ideas from future research. This plug-and-play characteristic ensures that the proposed framework stays relevant for years.

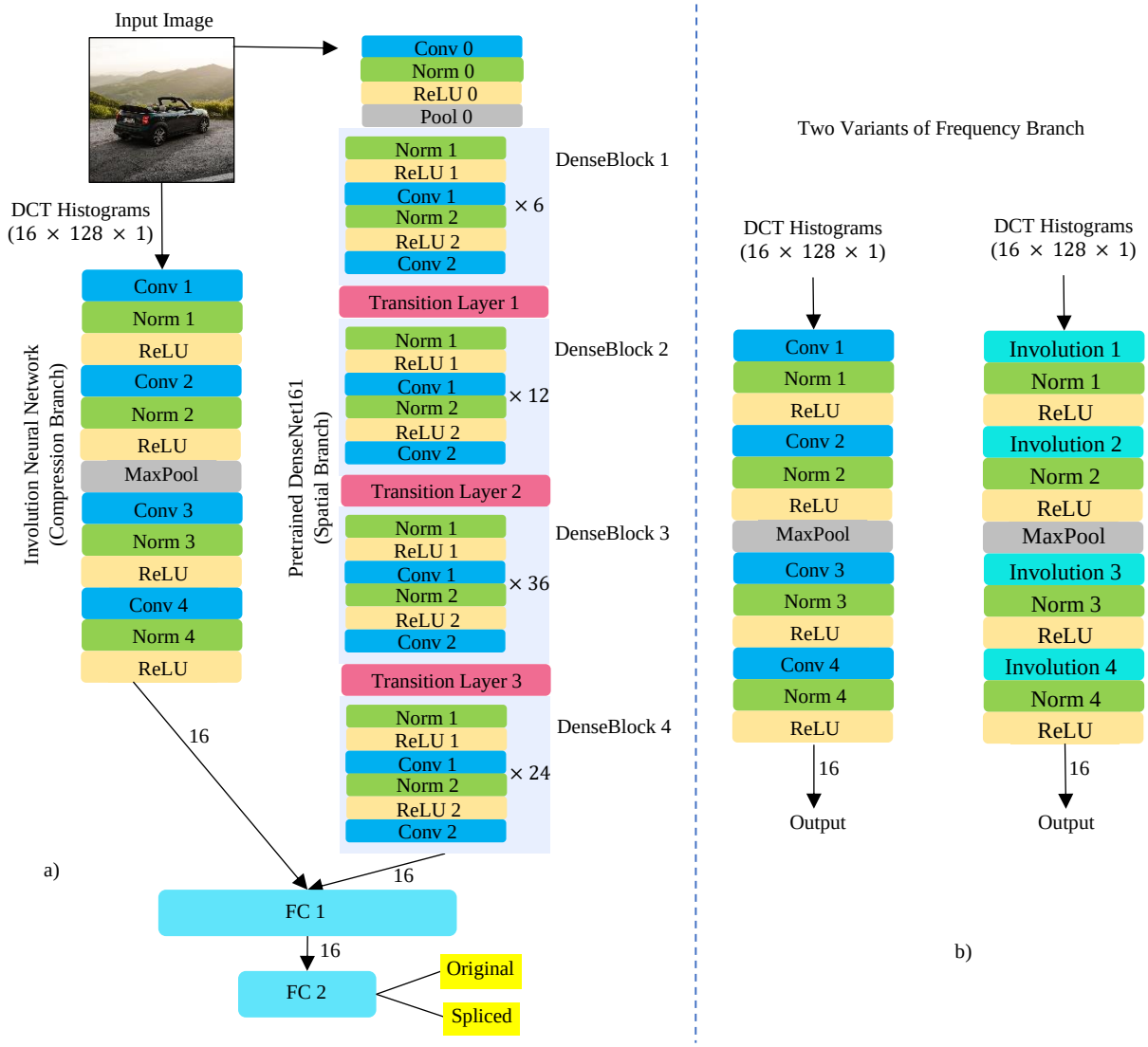


Fig. 30 a) DenseNet-CNN variant of the proposed Splice Detection Framework containing a pre-trained DenseNet161 for the spatial branch and a Convolution Neural Network (CNN) for the compression branch. b) The frequency branch of the proposed splice detection framework has two variants based on convolution and involution operators.

4.2.3.1 Spatial Branch

As the name suggests, the spatial branch extracts spatial features from input image samples. Several convolutional neural network architectures have proven highly effective for image classification problems. The spatial branch is designed to use transfer learning via deep networks pre-trained on the ImageNet dataset to inherit the proposed framework's lightweight and' futuristic characteristics. Designing the spatial branch in this manner has the following advantages. Firstly, several architectures ([177], [187], [155], [178]) that have proven to possess high image classification capabilities can be leveraged to extract discriminative spatial features for splice detection. Secondly, transfer learning ensures that the spatial branch remains 'lightweight,' i.e., only the last layer is trainable. Deep architectures of any size can be used for the spatial branch as long as it is pre-trained on ImageNet, and all layers are frozen except the last layer.

Table XXIV Several Deep Architectures are used for the spatial and compression branch of the proposed framework.

Ref	Model	Short Form	Branch	Input Size	Key Idea
[177]	VGG16	VGG	Spatial	256 x 256 x 3	Small convolutional kernels
[187]	GoogleNet	-	Spatial	256 x 256 x 3	Inception module, parallel convolution
[155]	ResNet18	ResNet	Spatial	256 x 256 x 3	Skip Connections
[178]	DenseNet161	DenseNet	Spatial	256 x 256 x 3	Feature Reuse
[188]	Vision Transformer	ViT	Spatial	384 x 384 x 3	Self-Attention on Images
-	Convolutional Neural Network	CNN	Compression	16 x 128 x 1	Channel-Specific and Spatial-Agnostic
[189]	Involution Neural Network	INN	Compression	16 x 128 x 1	Spatial-Specific and Channel-Agnostic

The spatial branch's last layer in the proposed framework is trainable and modified to produce features of 16 dimensions. Fig. 30 shows a variant of the proposed splice detection framework that uses an ImageNet pre-trained 'Vision Transformer' [188] in the spatial branch. Table XXIV shows several pre-trained deep architectures used for the spatial branch of the proposed spliced detection framework. All models receive color images as input. It is restated that such a design enjoys strong classification capabilities in the spatial domain while staying lightweight regardless of the size of the deep architecture used.

4.2.3.2 Compression Branch

The Discrete Cosine Transform (DCT) is a mathematical method that converts spatial domain data, such as images, into frequency domain data. In JPEG compression, the main

concept of utilizing DCT is based on the fact that most of the image's energy is focused on a few low-frequency coefficients. In contrast, high-frequency coefficients capture finer details and have less impact on the overall visual quality. JPEG compression achieves compression by quantizing and deleting higher frequency components while maintaining visual quality deemed acceptable to the human eye. The Discrete Cosine Transform (DCT) is a crucial element of JPEG compression. It facilitates the reduction of picture data by converting it into the frequency domain and eliminating less significant high-frequency details.

Several existing works highlight the presence of distinct compression artifacts in spliced jpg images [190], [70]. Specifically, an original jpg image (without splicing) is compressed once. However, a spliced image containing an object pasted from another image undergoes a second jpg compression. This dual compression leaves distinct artifacts in DCT coefficient histograms. The DCT coefficients include 1 DC and 63 AC coefficients. Different works have considered different AC coefficients in zig-zag order with different histogram ranges. [190] highlights the artifacts by using 9 DCT coefficients (zig-zag order) and a histogram range of $[-5,5]$ to constitute features of size 99×1. Similarly, [70] considers 9 DCT coefficients with a histogram range of $[-50,50]$ to formulate a feature size of 909.

Visual analysis is conducted on several original-spliced image pairs, and histograms are plotted to evaluate the ideal range of features. Fig. 31 shows the histograms from an original (green) and corresponding spliced (red) image. Plotting the zeroth, fourth, eighth, and fifteenth AC coefficient (zig-zag order) demonstrates the presence of distinct maximum values and a varying number of non-zero histogram bins in the range of $[-63,64]$ between the original and spliced image. Hence, 16 AC zig-zag coefficients (0 to 15) are selected, and the histogram range is set to $[-63,64]$. The input size for the compression branch is set to $16 \times 128 \times 1$.

The histogram data input is extracted from original resolution images; hence, the compression branch is 'information preserving,' i.e., there is no loss of information from input resizing. Regardless of the input image dimensions, the compression branch receives compression information from original resolution images having a size $16 \times 128 \times 1$. Compression features extracted from original resolution images are standardized to have a mean value of 0 and a standard deviation value of 1 (zero-centered input) to aid in faster model convergence. The input size of $16 \times 128 \times 1$ helps keep the proposed framework 'lightweight,' i.e., the architecture processing input of this size need not be massive.

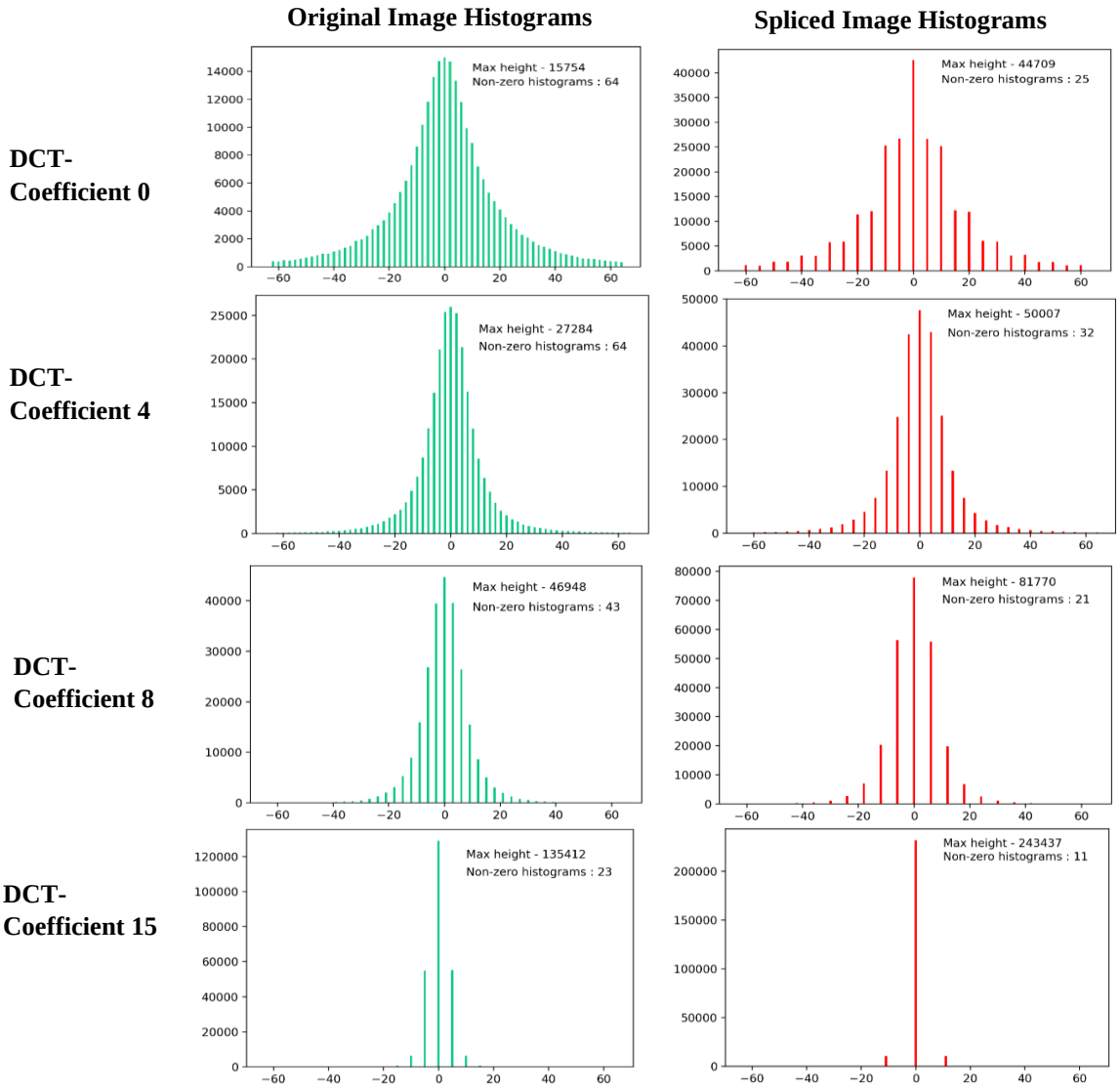


Fig. 31 Double compression artifacts in DCT coefficient histograms of spliced jpg images.

Two different architectures are employed as the compression branch in several variants of the proposed framework. Besides trying a simple CNN, an involution neural network (INN) containing the novel involution kernel [189] (introduced in CVPR 2021) is also used as the compression branch. The novel involution-based neural network can achieve competent classification results at a lesser computational cost than CNN compression branch models.

The convolution involves applying a kernel (filter) to an input picture by sliding it across the image and calculating the weighted sum of pixel values inside the kernel's receptive field at each place. Involution is the reverse process of convolution. Involution calculates the weighted sum of kernel values based on the pixel values of the input image rather than computing a weighted sum of pixel values. Involution is convolving a kernel with an input picture by sliding it across the image and calculating a weighted total, with the kernel values

being fixed and the input image values being adjusted. Involution has been suggested as a substitute for convolution in some deep learning structures, asserting enhanced efficiency and performance in specific tasks. Fig. 32 demonstrates the feature extraction methodology of the two kernels.

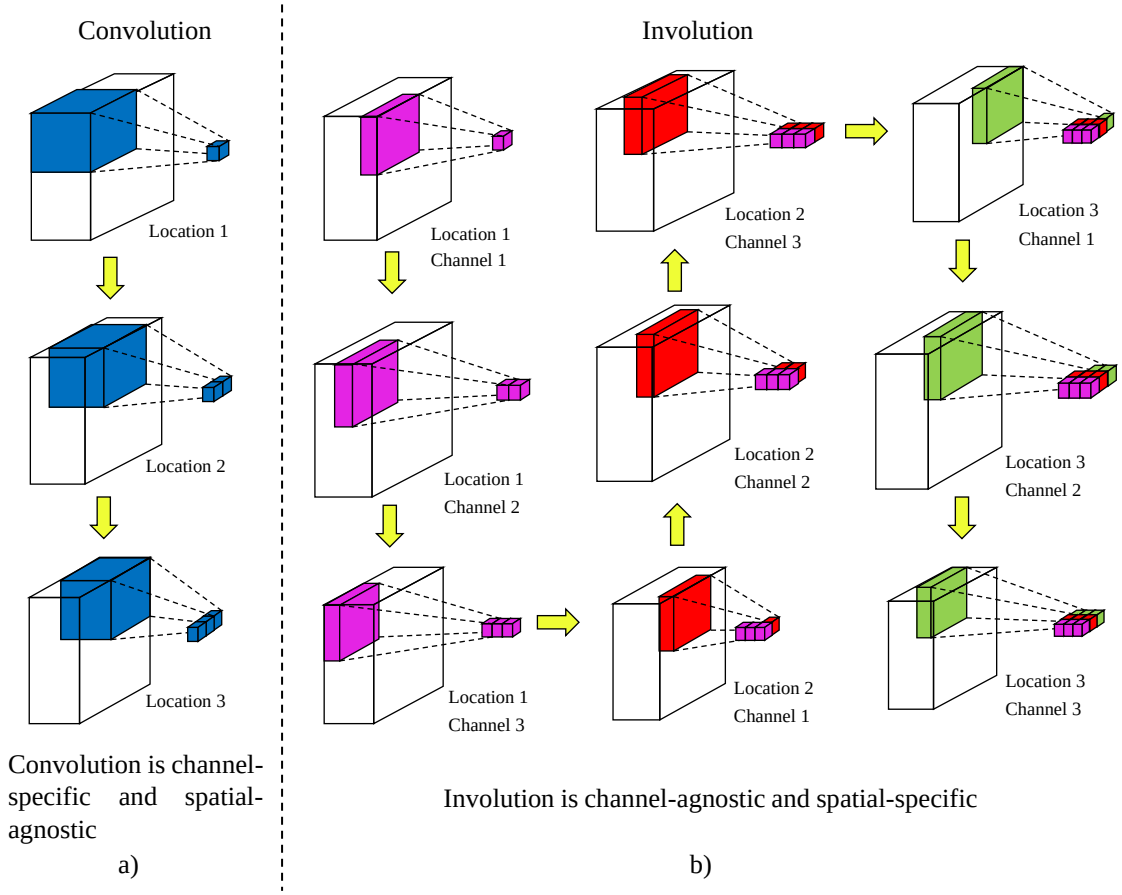


Fig. 32 Comparison of Convolution and Involution kernels.

Both the CNN and INN variants contain four convolution/involution layers. Each layer is accompanied by batch normalization. Max Pooling is used to reduce the dimensionality of feature maps. ReLU activation function is used. INN models specialize in reducing the number of trainable parameters while achieving competent results. Despite the fewer parameters in the INN compression branch, it can achieve competent classification results. The compression branch is also designed to produce the final features of size 16.

4.2.3.3 Final Framework

The final model fuses the 16 features each spatial and compression branch produces. It utilizes two fully connected layers to scale down the fused features to the final two features

representing binary classification scores. ReLU activation and batch normalization are applied to the first fully connected layer output.

4.2.4 Experimental Setup

4.2.4.1 Datasets

This experiment is conducted using two image splicing datasets. The first dataset is the newly proposed BiometricLab-DTU-Splice Dataset already discussed in Section 4.2.2. The second dataset is the CASIA v2.0 dataset.

The CASIA v2.0 is a challenging image tampering dataset containing 7491 original and 5123 tampered images [181]. However, over half of the tampered images are uncompressed (TIFF or BMP format). Since the proposed splice detection framework utilizes compression artifacts of image samples, the uncompressed format images were compressed with a random quality factor. This modified variant of the CASIA v2.0 dataset is used for experimentation.

Since all three datasets are novel (two proposed and one modified publicly available dataset), a comparison of the proposed framework with existing splice detection approaches is conducted by training all architectures (proposed and existing models) on these new datasets instead of merely comparing with metric figures mentioned in published works. 5123 original images are chosen from the original CASIA v2.0 dataset for experimentation to maintain class balance.

Table XXV Details of Proposed Datasets

Dataset	Type	Original Samples	Spliced Samples	Total Samples	Splice Mask
BiometricLab-DTU-Splice dataset (automatic)	Proposed	8156	8156	16312	Yes
BiometricLab-DTU-Splice dataset (manual)	Proposed	3106	3106	6212	Yes
CASIA v2.0 (modified)	Publicly Available	5123	5123	10246	-

Table XXV demonstrates the number of samples in each dataset used in this experiment.

4.2.4.2 Hardware Resources and Evaluation Metrics

All experiments have been implemented using the Pytorch Library. All experiments are run on a system with 128 GB RAM and a 24 GB NVIDIA TITAN RTX graphic card. The metrics used to evaluate the proposed splice detection framework are *Accuracy* (ACC), *Precision* (P), *Recall* (R), *F1-score* (F1), *Area Under Curve* (AUC), and *Mathews Correlation Coefficient* (MCC).

Correctly identified spliced samples are counted as true-positive ($\mathbb{TP}$), original images correctly classified as original are counted as true-negative ($\mathbb{TN}$), original images misclassified as spliced are counted as false-positive ($\mathbb{FP}$) and spliced images misclassified as original are counted as false-negative ($\mathbb{FN}$). The various metrics are defined using Eq. (20)-(24).

$$Accuracy = \frac{\mathbb{TP}+\mathbb{TN}}{\mathbb{TP}+\mathbb{TN}+\mathbb{FP}+\mathbb{FN}} \tag{20}$$

$$Precision = \frac{\mathbb{TP}}{\mathbb{TP}+\mathbb{FP}} \tag{21}$$

$$Recall = \frac{\mathbb{TP}}{\mathbb{TP}+\mathbb{FN}} \tag{22}$$

$$\mathbb{F}1\ Score = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{23}$$

$$MCC = \frac{\mathbb{TP}*\mathbb{TN}-\mathbb{FP}*\mathbb{FN}}{\sqrt{(\mathbb{TP}+\mathbb{FP})(\mathbb{TP}+\mathbb{FN})(\mathbb{TN}+\mathbb{FP})(\mathbb{TN}+\mathbb{FN})}} \tag{24}$$

4.2.4.3 Pre-processing and Data Augmentation

Images are resized to 256×256 for all spatial models except Vision Transformer, which expects an input size of 384×384. Images are standardized to have a mean value of 0 and a standard deviation of 1 (channel-wise) and hence are 'zero-centered.' Three data augmentations are used for experimentation: random horizontal flip, random vertical flip, and random rotation from 0 to 180 degrees.

The datasets mentioned in the previous section are combined, as shown in Table XXVI, to produce a larger pool of diverse input data. The AMC variant combines the proposed and modified CASIA datasets to produce one large dataset of 32770 images. The AM variant combines both variants of the proposed dataset to make a more extensive set of 22524 images.

Table XXVI Dataset Variants that are used for the experimentation.

Dataset Variant	Code	Total Samples	Train & Validation Samples	Test Samples
BiometricLab-DTU-Splice dataset (automatic + manual)	AM	22524	90%	10%
All three datasets combined	AMC	32770		

All variants of the proposed splice detection framework and existing splice detection architectures are trained and evaluated on these variants to ensure comprehensive training and evaluation in this experiment.

4.2.4.4 Model Variants

The models specified in Table XXIV are combined and evaluated to verify the validity of the proposed splice detection framework. The combined architectures are named after their constituent branch models.

The ResNet-CNN variant uses a pre-trained resnet18 in the spatial branch and a simple CNN for the frequency branch. The GoogleNet-CNN variant has a GoogleNet architecture as its spatial branch and CNN as its frequency branch. Similarly, VGG-CNN, DenseNet-CNN, and ViT-CNN include VGG, DenseNet161, and Vision Transformer models in their spatial and CNN-based frequency branches.

The involution operator is also used in the frequency branch with fewer trainable parameters than the simple CNN-based frequency branch. Hence each of the above-mentioned spatial branch models is combined with an involution-based frequency branch (INN) to produce five more variants denoted by ResNet-INN, GoogleNet-INN, VGG-INN, DenseNet-INN, and ViT-INN.

Hence, ten model variants of the proposed splice detection framework are trained and evaluated in this experiment. The spatial branch of each variant is loaded with ImageNet weights and frozen except for the last layer.

4.2.4.5 Hyperparameter Settings

All experiments are run for 30 epochs. The batch size is 256 for all models except Vit-CNN and the train-test splits are set to 90% and 10%, respectively. Five-fold cross-validation is implemented to ensure comprehensive training. Several optimizers are tried, and the Adam optimizer consistently produces the best results. The learning rate (LR) is decayed linearly. Table XXVII shows the hyperparameter settings for each model variant to obtain the best results.

The CNN frequency branch-based architectures achieved their best results with an initial learning rate of 0.001 and decayed by 50% after every one or two epochs. The INN frequency branch-based architectures showed their best results when the learning rate was initialized to a higher value and decayed by a smaller factor. Specifically, the initial learning rate for INN-based models was 0.01 and decayed by 10% only (except for DenseNet-INN) after every one or two epochs.

Table XXVII Hyperparameter setting for various variants of the proposed Splice Detection Framework.

Models	Epochs	Batch Size	Initial LR	LR Decay Factor	Step Size of LR Decay (epochs)	Optimizer
VGG-CNN	30	256	0.001	0.5	1	Adam
VGG-INN	30	256	0.01	0.9	1	Adam
ResNet-CNN	30	256	0.001	0.5	2	Adam
ResNet-INN	30	256	0.01	0.9	1	Adam
GoogleNet-CNN	30	256	0.001	0.5	2	Adam
GoogleNet-INN	30	256	0.01	0.9	1	Adam
DenseNet-CNN	30	256	0.001	0.5	2	Adam
DenseNet-INN	30	256	0.01	0.5	2	Adam
ViT-CNN	30	576	0.001	0.5	1	Adam
ViT-INN	30	256	0.01	0.9	1	Adam

4.2.5 Experimental Results & Analysis

This section presents the results of all model variants of the proposed splice detection framework.

4.2.5.1 Performance of Proposed Splice Detection Framework

ACC, P, R, F1, AUC, and MCC scores obtained by model variants of the proposed framework are mentioned in Table XXVIII**.**

Table XXVIII Performance of the proposed Splice Detection Framework.

Proposed Model Variants	Dataset	Input Size (Spatial Branch)	Trainable Parameter Count	ACC	P	R	F1	AUC	MCC
VGG-CNN	AMC	256 x 256 x 3	99,538	0.9305	0.9471	0.9116	0.9290	0.9749	0.8617
VGG-INN	AMC	256 x 256 x 3	76,854	0.9018	0.9125	0.8847	0.8984	0.9640	0.8038
ResNet-CNN	AMC	256 x 256 x 3	42,194	0.9379	0.9510	**0.9208**	0.9357	0.9749	0.8761
ResNet-INN	AMC	256 x 256 x 3	19,510	0.8571	**0.9814**	0.7328	0.8391	0.9472	0.7402
GoogleNet-CNN	AMC	256 x 256 x 3	50,386	0.9342	0.9458	**0.9208**	0.9331	0.9781	0.8687
GoogleNet-INN	AMC	256 x 256 x 3	27,702	0.9006	0.9384	0.8574	0.8961	0.9647	0.8042
DenseNet-CNN	AMC	256 x 256 x 3	69,330	**0.9382**	0.9578	0.9185	**0.9378**	**0.9802**	**0.8772**
DenseNet-INN	AMC	256 x 256 x 3	46,646	0.8779	0.9105	0.8278	0.8672	0.9431	0.7577
ViT-CNN	AMC	384 x 384 x 3	46,290	0.9376	0.9571	0.9144	0.9353	0.9790	0.8759
ViT-INN	AMC	384 x 384 x 3	23,606	0.8733	0.9209	0.8174	0.8661	0.9463	0.7516

The training loss and validation loss plots converge smoothly towards 0 and validation accuracy towards increases close to 1. The confusion matrix obtained after training each model variant confirms the strong classification capabilities of the proposed framework model variants.

4.2.5.2 Result Analysis

This section discusses the results obtained by model variants in the previous section. Table XXVIII presents the number of trainable parameters in each model.

The largest model is VGG-CNN with 99538 trainable parameters, while the lightest model is the ResNet-INN variant with only 19510 trainable parameters. This proves the lightweight

nature of the proposed framework since deep learning models of moderate size easily contain a few million trainable parameters. In contrast, all variants of the proposed splice detection framework are incredibly lightweight.

The CNN frequency branch-based models consistently score more than 0.93 ACC, more than 0.92 F1-score, more than 0.97 AUC, and more than 0.86 MCC scores. The INN frequency branch-based models score more than 0.87 ACC (except for ResNet-INN), more than 0.86 F1-score F1 (except for ResNet-INN), more than 0.94 AUC, and more than 0.74 MCC. It is clear from the above scores that the reduced number of trainable parameters in the INN models resulted in a slight performance drop.

The DenseNet-CNN model achieved the best scores with 0.9382 ACC, 0.9378 F1, 0.9802 AUC, and 0.8772 MCC. ResNet-INN scored the highest precision score of 0.9814. ResNet-CNN and GoogleNet-CNN share the highest recall score of 0.9208.

Fig. 33 shows the ROC curves for each model variant of the proposed splice detection framework.

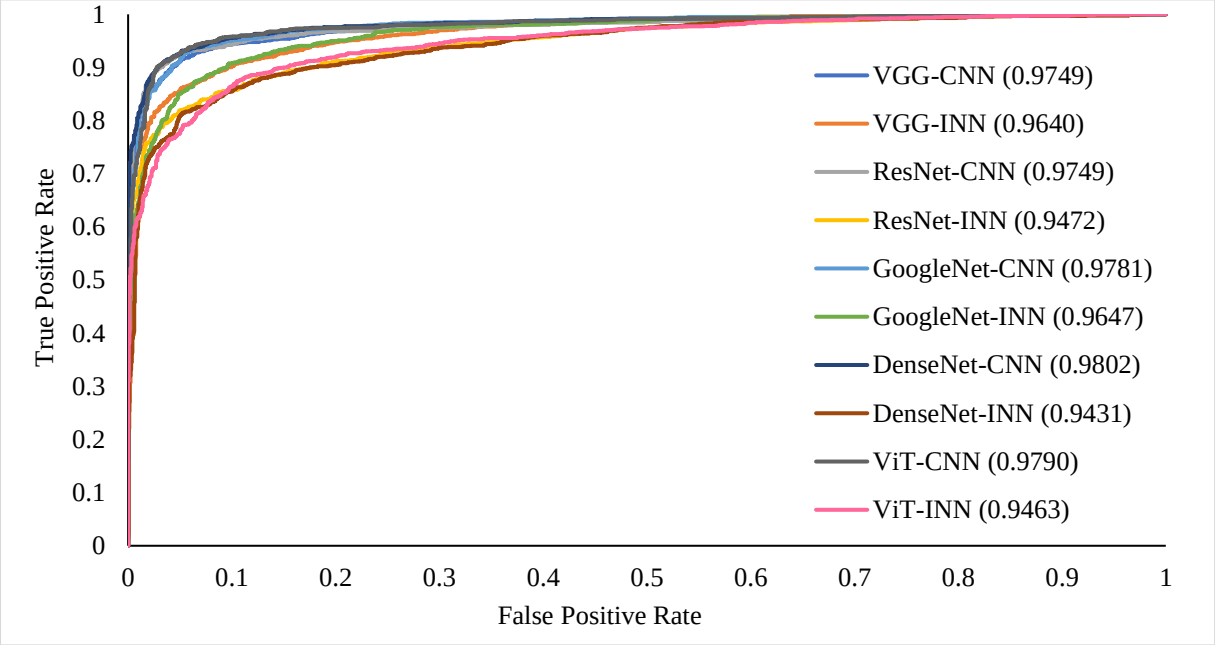


Fig. 33 ROC curves for different variants of the proposed framework.

4.2.5.3 Ablation Study

An ablation study of the proposed framework is conducted to confirm the validity of the contributions made by the spatial and compression branches. The spatial branch and frequency branch are trained and evaluated individually. Resnet18 is the spatial branch, loaded with ImageNet weights, and contains frozen layers except the last.

Table XXIX Ablation study performance for the individual branches of the proposed Splice Detection Framework

Models	ACC	P	R	F1	AUC	MCC
Spatial Branch Only (ResNet)	0.6957	0.7106	0.6495	0.6787	0.7602	0.3922
Frequency Branch Only (CNN)	**0.7966**	**0.8376**	0.7251	**0.7773**	**0.8718**	**0.5971**
Frequency Branch Only (INN)	0.6146	0.5786	**0.8033**	0.6727	0.7296	0.2523

Table XXIX presents the results obtained from the ablation study, where each branch is trained on the same dataset as the proposed model. The best accuracy of 0.7966 is achieved by the CNN-based frequency branch, which is still very low compared to the performance of the proposed model variants. Similarly, the best F1, AUC, and MCC scores of 0.7773, 0.8718, and 0.5971, respectively, are very low. The scores in Table XXIX indicate that the individual branches do not have high classification capability compared to the proposed model variants.

4.2.6 Comparison with Existing Splice Detection Methods

A comparison of the performance of the proposed splice detection framework with existing state-of-the-art approaches is presented here. The comparison is based on classification metric scores and the size of models under comparison in terms of the number of trainable parameters.

4.2.6.1 Comparison based on Classification Metrics

This section presents the comparison of the proposed splice detection framework with the existing state-of-the-art methods for splice detection.

Table XXX presents a comparison of the proposed splice detection framework against existing state-of-the-art methods on the CASIA v2 dataset. The proposed framework outperforms all the mentioned approaches.

Table XXX Comparison of Existing Splice Detection Approaches against the proposed splice detection framework on the CASIA dataset.

Model	ACC	P	R	F1	AUC	MCC
Zhang et al. **[191]**	-	-	-	0.7653	-	-
Sun et al. **[192]**	-	-	-	0.6805	-	-
Yan et al. **[193]**	-	0.8090	0.7460	0.7350	-	-
Bi et al. **[62]**	-	0.6780	0.5860	0.5860	-	-
Wu et al. **[194]**	-	0.6310	0.6730	0.6510	-	-
Chen et al. **[195]**	-	-	-	0.7388	-	-
Liu et al. **[196]**	-	-	-	0.5232	-	-
Salloum et al. **[197]**	-	-	-	0.6675	-	-
Wu et al. **[198]**	-	-	-	0.5770	-	0.5590
Chen et al. **[199]**	-	-	-	0.6097	-	-
Zhang et al. **[200]**	-	-	-	0.6286	-	-
Xu et al. **[201]**	-	-	-	0.4601	0.8191	-
Chen et al. **[202]**	-	0.6616	0.7548	0.7051	-	-
Zhou et al. **[203]**	-	0.5044	0.6575	0.5709	-	-
Wei et al. **[204]**	-	0.5202	0.6642	0.5834	-	-
(Proposed) DenseNet-CNN	0.8125	0.7823	0.7646	0.7733	0.9126	0.6136
(Proposed) ResNet-CNN	**0.8851**	**0.8123**	**0.9415**	**0.8664**	**0.9596**	**0.7744**

Table XXXI presents a comparison of the proposed framework on the combined dataset. Here again, the proposed framework performs better than other similar approaches. Four existing splice detection approaches have been implemented and evaluated in this section to demonstrate a fair comparison and illustrate the superiority of the proposed splice detection framework. The architectures implemented and the training process followed are the same as the research manuscripts mentioned.

Table XXXI Comparison of Existing Splice Detection Approaches against the ResNet-CNN variant of the proposed splice detection framework.

Ref.	Model	Trainable Parameters	Dataset	ACC	P	R	F1
[205]	(Existing) Dense CNN	48,818	AMC	0.5850	0.5923	0.5498	0.5703
[206]	(Existing) DCT + Quantization Table	11,104,706	AMC	0.9273	0.9919	0.8951	0.9409
[70]	(Existing) Multi-Domain CNN	8,693,322	AMC	0.4693	0.4742	0.6413	0.5452
-	(Proposed) DenseNet-CNN	69,330	AMC	0.9382	0.9578	0.9185	0.9378
-	(Proposed) ResNet-CNN	42,194	AMC	0.9379	0.9510	0.9208	0.9357
[78]	(Existing) Weighted Feature Fusion	4,111,490	AM	0.5682	0.5837	0.4758	0.5245
-	(Proposed) ResNet-CNN	42,194	AM	0.9197	0.9508	0.9259	0.9382

The architecture in [205] contains four dense blocks with four, two, and two dense layers, respectively. Transition layers include 1×1 convolutions and Max pooling. Input images are converted to grayscale and resized to 256×256. Normalization is also done to the range of [0,1]. The model is trained for 50 epochs with an initial learning rate of 0.001, which is decayed by 10% every 1/3rd of an epoch. The optimizer used is SGD, and the batch size is 32.

The best architecture of [206], combining a quantization table to the last pooling and two fully connected layers, is implemented for comparison. The histogram range of Y channel DCT coefficients is [-60,60]. Train and test images are split by 90% and 10%, respectively. The model is trained for 50 epochs with a learning rate is 0.001 and an Adam optimizer. All convolution operations are accompanied by batch normalization.

The multi-domain CNN proposed in [70] is implemented and repurposed towards binary classification for image splice detection. The histogram range for DCT coefficients is [-50,50]. Train, validation, and test sets have sizes of 90%, 5%, and 5%, respectively. Both spatial and frequency branch use dropout and their respective inputs are applied with normalization to the range [0,1]. The model is trained for 50 epochs with the AdaDelta optimizer, with an initial learning rate of 0.01 is reduced by 10% every epoch.

The weighted feature combination architecture in [78], having four weight combination modules to combine YCbCr, Edge, and PRNU features, is implemented. The weight

parameters $\propto_a$, $\propto_b$ and $\propto_c$ for each of the four weight modules are added to the computational graph and list of trainable parameters for automatic tuning during backpropagation. Cross-validation training is implemented with SGD optimizer, and the initial learning rate is 0.001, which decays by 10% every 10 epochs. Initial values for $\propto_a$, $\propto_b$ and $\propto_c$ are set to 0.3, 0.3 and 0.4, respectively.

PRNU features can be calculated from flat-field images of the source camera, or they can be estimated from a large number of natural images captured by a given camera device [207]. It is unclear how the authors computed PRNU features for CASIA dataset images since the dataset paper does not include source device information [181]. Hence, to alleviate this problem, the implemented weighted feature combination architecture is evaluated on the AM dataset variant, which includes the proposed Biometric-DTU-Splice dataset (automatic + manual). All images of the proposed Biometric-DTU-Splice dataset are derived from the RAISE dataset's uncompressed images whose camera device information is available. 50 images from each camera model ($N = 50$) are used to compute the PRNU factor $\widehat{\mathrm{K}}$ using Eq. 25 for each camera device, as done in [207].

$$\widehat{\mathrm{K}} = \frac{\sum_{k=1}^{N}(\mathcal{W}_k \mathcal{I}_k)}{\sum_{k=1}^{N}(\mathcal{I}_k)^2} \tag{25}$$

Here $\mathcal{I}_k$ is one of the multiple images from the source camera and $\mathcal{W}_k = \mathcal{I}_k - \widehat{\mathcal{I}_k}$ represents image noise residual. The implemented architecture is compared against the ResNet-CNN variant of the proposed framework, which is trained for a second time on the AM dataset variant for a fair comparison (Table XXXI).

The results from Table XXXI indicate the superiority of the proposed 'lightweight dual-branch information preserving spatio-compression modal splice detection framework.' Only [206] of the existing splice detection methods can match the proposed framework's accuracy, precision, recall, and f1-scores. But it is an extremely heavy architecture with more than 11 million trainable parameters.

4.2.6.2 Comparison based on Size (Number of Trainable Parameters)

One of the design principles of the proposed splice detection framework is to make it 'lightweight.' This idea is ideally suited when the usual splice detection datasets are small and deep architectures are prone to overfitting. The proposed model reduces the number of trainable parameters by using transfer learning in the spatial branch and processing DCT features of size

$16 \times 128 \times 1$ in the compression branch through extremely lightweight neural networks. The number of trainable parameters in each variant of the proposed splice detection framework is presented in Table XXVIII and that of existing splice methods in Table XXXI. Fig. 34 presents a size comparison of all the proposed splice detection framework variants (pink) and some existing splice detection methods (blue). The size difference (no. of trainable parameters) between the proposed and existing methods is so significant that a 'log-scale' is used to plot the size difference. Fig. 34 indicates the 'lightweight' nature of the proposed splice detection framework variants.

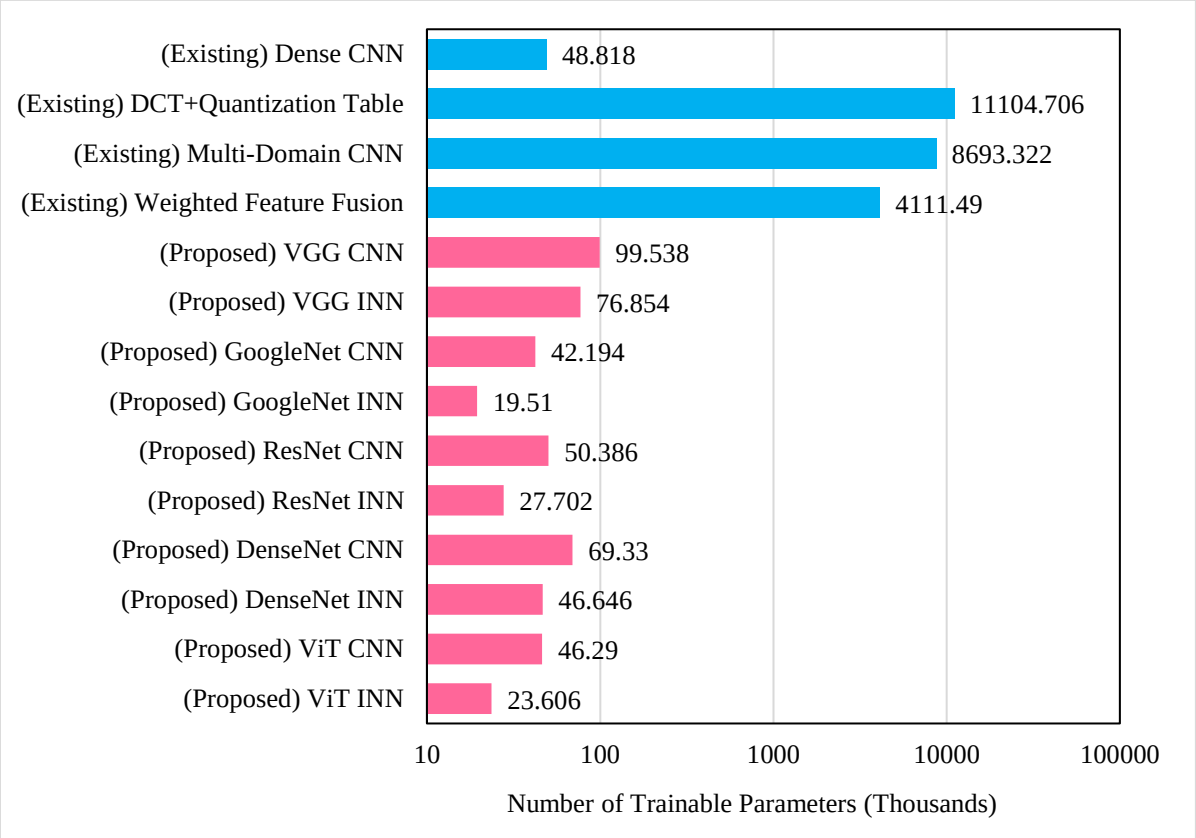


Fig. 34 Size comparison of the proposed Splice Detection Model Variants (pink) against Existing State-of-the-Arts (blue).

4.2.7 Conclusion

This research work makes two-fold contributions towards splice detection in jpg images. Firstly, a novel splice detection dataset, the *'BiometricLab-DTU Splice dataset,'* is proposed. The proposed splice dataset has two variants: the first is autogenerated from code, and the second contains handmade spliced samples. The proposed splice detection dataset significantly adds to the existing small-scale splice datasets. Secondly, a novel 'Lightweight Dual-branch Information Preserving Spatio-Compression Modal Splice Detection Framework' is proposed

that incorporates design principles consistent with today's splice detection research scenario (small-scale splice datasets). Several variants of the proposed framework are trained on images from the proposed splice dataset and a modified CASIA v2.0 dataset. Experimental results prove the superiority of the proposed splice detection framework compared to existing methods without requiring millions of trainable parameters in the neural network. A similar design principle can be used for future work to build a splice localization framework.

4.3 A Visually Attentive Splice Localization Network with Multi-Domain Feature Extractor and Multi-Receptive Field Upsampler

4.3.1 Abstract

Image splice manipulation presents a severe challenge in today's society. With easy access to image manipulation tools, it is easier than ever to modify images that can mislead individuals, organizations or society. In this work, a novel, "Visually Attentive Splice Localization Network with Multi-Domain Feature Extractor and Multi-Receptive Field Upsampler", has been proposed. It contains a unique "visually attentive multi-domain feature extractor" (VA-MDFE) that extracts attentional features from the RGB, edge and depth domains. Next, a "visually attentive downsampler" (VA-DS) is responsible for fusing and downsampling the multi-domain features. Finally, a novel "visually attentive multi-receptive field upsampler" (VA-MRFU) module employs multiple receptive field-based convolutions to upsample attentional features by focussing on different information scales. Experimental results conducted on the public benchmark dataset CASIA v2.0 prove the potency of the proposed model. It comfortably beats the existing state-of-the-arts by achieving an IoU score of 0.851, pixel F1 score of 0.9195 and pixel AUC score of 0.8989.

4.3.2 Proposed Architecture

The details of the proposed architecture are presented in this section.

4.3.2.1 Visually Attentive Multi-Domain Feature Extractor (VA-MDFE)

This section describes the visually attentive multi-domain feature extractor of the proposed model. Specifically, a baseline model aided with visual attention extracts features from the input image's RGB, edge and depth domain.

The design proposed in [178] serves as the foundational architecture in this research work. The system promotes the reuse of features by allowing a layer to access feature maps from all preceding levels, hence enhancing the overall flow of information throughout the network., as shown in Eq. 26:

$$\mathfrak{y}_L = \emptyset_L([\mathfrak{y}_0, \mathfrak{y}_1, \mathfrak{y}_2, \mathfrak{y}_3 \dots \dots \mathfrak{y}_{L-1}]) \quad (26)$$

The baseline contains four dense blocks and three transition blocks. The last two dense blocks and transition blocks are discarded for computational efficiency.

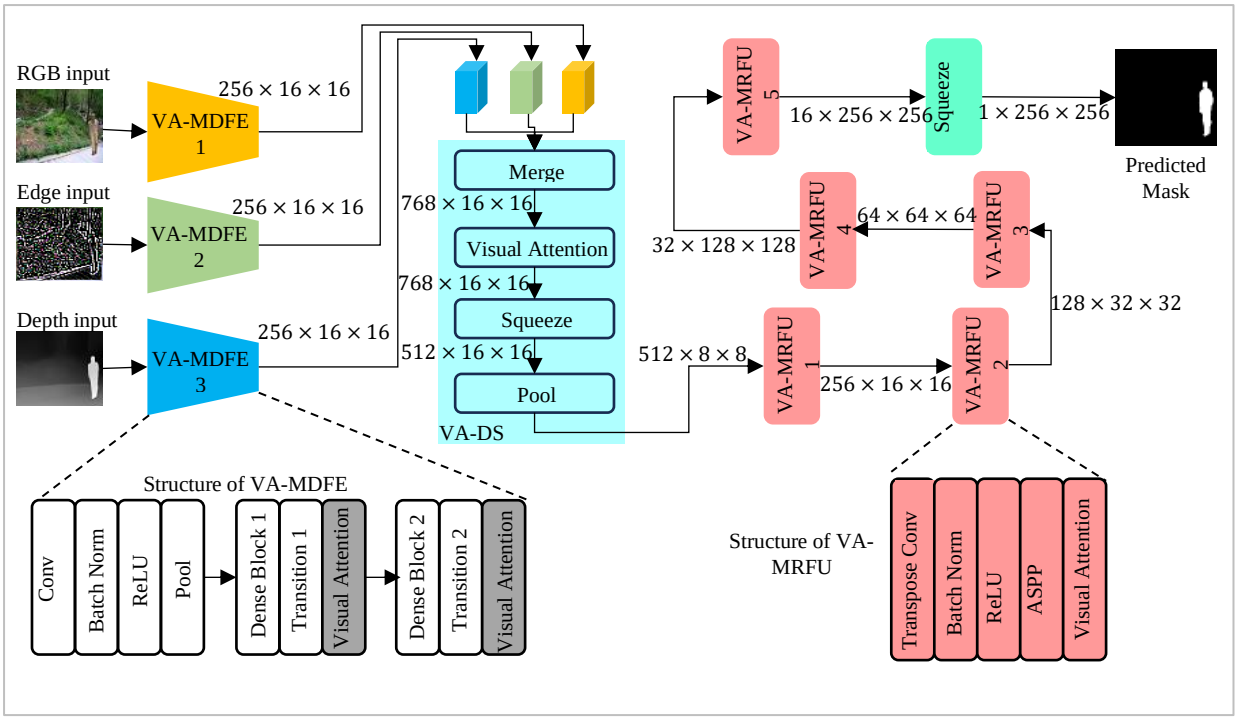


Fig. 35 The structure of the proposed splice localization model.

Each of the two remaining dense blocks of this baseline is appended with a visual attention layer [102] that employs a three-branch design using a 'rotational' module to detect distinct elements in three different orientations. This approach ensures efficient computation with a minimal parameter count of just 300.Two branches handle the input of shape $\mathcal{X} \in \mathbb{R}^{H \times W \times C}$ and employ a distinct Z-pool mechanism to extract important channel characteristics along the height H and W dimensions. Meanwhile, the a traditional spatial attention is computed in the third branch. Three instances of VA-MDFE is used to create a three-branch architecture for multi-domain feature extraction as shown in Fig. 35. The three input domains are RGB, edge and depth.

4.3.2.2 Visually Attentive Downsampler (VA-DS)

The proposed model contains a novel "visually attentive downsampler" (VA-DS). VA-DS is responsible for aggregating features from each domain. VA-DS has two stages namely 'merge' and 'downsample'. The following equation describes the operations of VA-DS:

$$f_{down} = \mathcal{P}(\mathcal{Sq}(\mathcal{VA}(\mathcal{M}(f_i)))) \tag{27}$$

Here f_i refers to the input features from each of the three domains. $\mathcal{M}(.)$ stands for the merge operation and it is achieved by concatenating features along the channel dimension. $\mathcal{VA}(.)$ refers to the visual attention layer applied after merging to highlight important regions within the features fused from multiple domains. $\mathcal{Sq}(.)$ means 'squeeze' operation, which reduces the

number of channels via 1×1 convolution. $\mathcal{P}(.)$ is the pool operation to reduce the spatial resolution of features.

4.3.2.3 Visually Attentive Multi-Receptive Field Upsampler (VA-MRFU)

The proposed model contains a novel "visually attentive multi-receptive field upsampler" (VA-MRFU). VA-MRFU is responsible for upsampling the multi-domain features extracted from VA-MDFE. The following equation can describe this module:

$$f_{up,i} = \mathcal{VA}(\mathcal{ASPP}(\mathcal{R}e(\mathcal{BN}(\mathcal{TC}(f_{multi})))) \quad (28)$$

Here $f_{up,i}$ are the upsampled features from one VA-MRFU module. f_{multi} represents the multi-domain features extracted from VA-MDFE. $\mathcal{TC}(.)$ stands for transpose convolution with kernel size and stride taken as 2. $\mathcal{BN}(.)$ and $\mathcal{R}e(.)$ represent batch normalization and ReLU activation, respectively.

The multi-receptive field mechanism has been implemented via the atrous spatial pyramid pooling module represented here as $\mathcal{ASPP}(.)$. Specifically, it is a three-branch architecture that performs convolution over the input with varying receptive fields. This is achieved by varying the 'dilation' parameter of the convolution operation to change the receptive fields without increasing the computational cost. Dilation rates of 2, 3 and 4 are used in this experiment in each VA-MRFU module.

Finally, $\mathcal{VA}(.)$ represents the attention layer similar to the ones used in the above modules.

4.3.3 Experimental Setup

This part outlines the specific experimental parameters employed in this study to assess the efficacy of the proposed approach.

4.3.3.1 Dataset

The CASIA v2.0 is a challenging image tampering dataset containing 7491 original and 5123 tampered images [181]. The manipulated images are created through various techniques like copy-move, splicing, and removal. The manipulated images aim to simulate real-world tampering scenarios, providing researchers with comprehensive data to develop and evaluate algorithms for detecting and analyzing image forgeries.

4.3.3.2 Preprocessing, Hyperparameters, Hardware & Loss Function

All images are resized to the resolution of 256×256. Each pixel value is normalized to the range of [0,1].

All experiments are run for 20 epochs. Adam optimizer is used for weight updation. The learning rate is initialized to 0.0001 and is decayed linearly by 10% after every epoch.

Two 24GB NVIDIA RTX A5000 GPUs are run in parallel for this experiment.

The 'focal loss' from [208] has been used to train the proposed model. This loss is ideally suited for the background-foreground class imbalance problem in object detection scenarios.

4.3.3.3 Evaluation Metrics

The following metrics have been to measure the localization capabilities of the proposed model.

The *Intersection over Union* (IoU) quantifies the degree of overlap between the predicted and ground truth masks by calculating the ratio of the intersection area to the union area of these areas. Higher iou scores suggest superior alignment and accuracy in localization tasks, where a value of 1 signifies complete overlap.

Pixel-level accuracy measures the correctness of localised image modifications by determining the proportion of properly identified pixels out of the total number of pixels in a picture. Greater pixel-level accuracy ratings suggest more alignment and precision in localised alterations.

Pixel-level F1 score is a quantitative measure employed to assess the precision and memory of localised picture alterations at the individual pixel level. The F1 score quantifies the trade-off between correctly identified manipulated pixels (true positives) and the accuracy of the localised manipulation compared to the ground truth annotations. Higher F1 scores indicate better overall accuracy in capturing manipulated regions.

Pixel-level AUC, also known as Area Under the Curve, is a quantitative metric utilised to evaluate the effectiveness of detecting localised image modification at different thresholds. It assesses the balance between the rate of correctly identified manipulated areas and the rate of incorrectly identified manipulated regions at the individual pixel level. A greater pixel-level AUC signifies superior differentiation between manipulated and non-manipulated areas, demonstrating the efficacy of the detection system.

4.3.4 Experimental Results & Analysis

This section presents the experimental results obtained for the proposed model on the benchmark datasets.

4.3.4.1 Performance on Benchmark Dataset CASIA v2.0

Fig. 36 presents the performance on the CASIA v2.0 dataset in terms of the IoU, pixel accuracy, pixel F1 and pixel AUC scores. The proposed model achieves excellent scores, highlighting its strong localization capability.

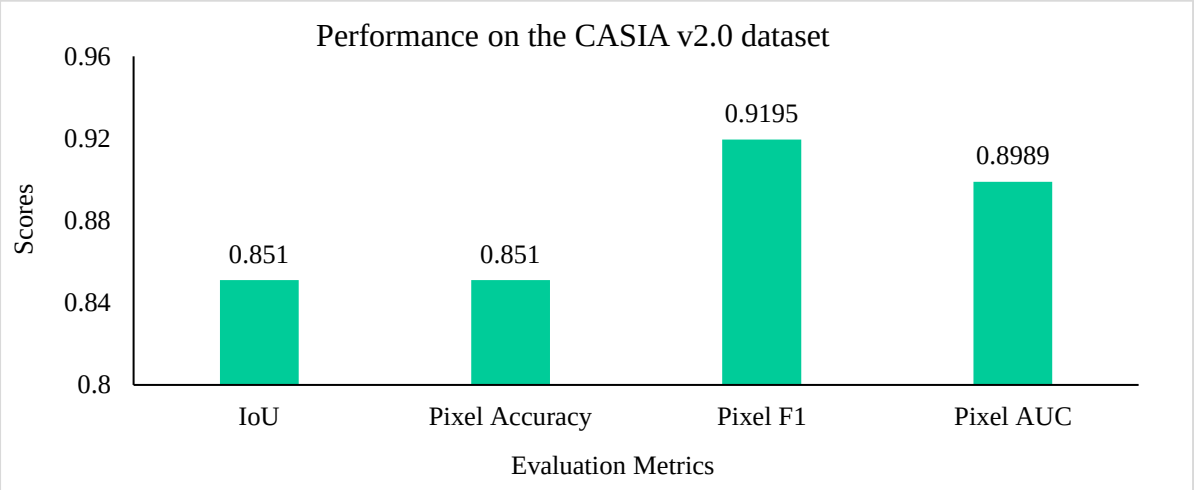


Fig. 36 Performance of the proposed model on CASIA v2.0 dataset.

4.3.4.2 Comparison Against the State-of-the-Arts

Table XXXII compares the proposed model against the existing state-of-art methods. The proposed model easily outperforms other methods, demonstrating its superiority.

Table XXXII Comparison of the Proposed Model against existing state-of-the-art methods.

Methods	IoU	Pixel F1	Pixel AUC
Zhang et al. [191]	0.7139	0.7653	-
Sun et al. [192]	0.5157	0.6805	-
Huang et al. [209]	-	0.6100	0.749
Nazir et al. [210]	-	0.8469	-
Yin et al. [211]	-	0.5840	0.8950
RRU-Net [62]	0.4752	0.5333	-
ManTra-Net [194]	0.1261	0.2009	-
Chen et al. [199]	0.4386	0.6097	-
Proposed Model	**0.8510**	**0.9195**	**0.8989**

4.3.4.3 Qualitative Analysis

This section presents a visual comparison of the forgery masks predicted by the proposed model.

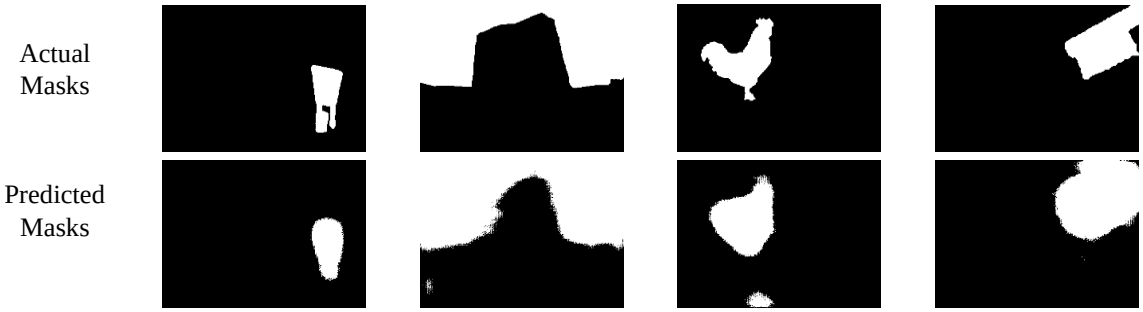


Fig. 37 A comparison of the actual and predicted masks from the proposed model.

Fig. 37 shows that the proposed splice localization model can locate the region of manipulation with precision.

4.3.4.4 Ablation Study

This section presents an ablation study of the proposed model. Specifically, each of the individual domains are evaluated separately. This means that the proposed model is compared against single domain feature extractors with RGB, edge and depth images as input.

Table XXXIII Comparison of single-domain vs the proposed multi-domain model.

Methods	**IoU**	**Pixel Accuracy**	**Pixel F1**	**Pixel AUC**
RGB Only	0.5494	0.7245	0.6201	0.637
Edge Only	0.5164	0.614	0.652	0.5497
Depth Only	0.4633	0.6681	0.6332	0.5804
Proposed Model	**0.851**	**0.851**	**0.9195**	**0.8989**

Table XXXIII presents the performance of single-domain approaches against the proposed model. Specifically, each domain from the RGB, edge and depth is used as an individual feature extractor in the single domain models.

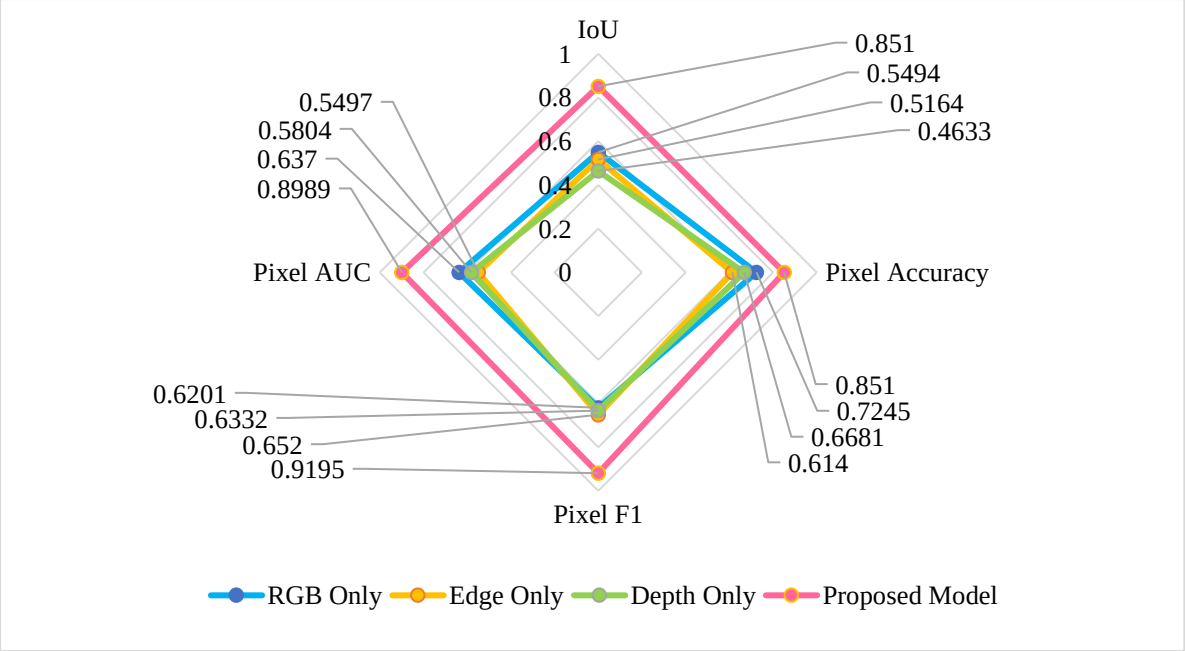


Fig. 38 A visual comparison of single domain (RGB, edge and depth) against the proposed multi-domain feature extractor.

Fig. 38 visually compares the ablation study conducted where single domain architectures are compared against the proposed model. The proposed model (pink) outperforms the individual domains of RGB (blue), edge (orange) and depth (green) across all metrics, namely IoU, pixel accuracy, pixel F1 and pixel AUC. This clearly states the benefit of having a multi-domain feature extractor.

4.3.5 Conclusion

In this research work, a novel image splice localization network is proposed. The proposed model contains a novel "visually attentive multi-domain feature extractor" that extracts attentional features from the RGB, edge and depth domain. A novel "visually attentive multi-receptive field upsampler" is responsible for the upsampling of features using multiple receptive field-based convolution operation. Experimental results on the CASIA v2.0 public benchmark dataset prove the potency of the proposed model as it easily beats the existing research approaches of splice localization.

4.4 Significant Outcomes of this Chapter

The significant outcomes of this chapter are as follows:

- Proposed a novel image splice detection dataset, *BiometricLab-DTU Splice dataset* having spliced samples generated from Python code and Adobe Photoshop software.

- Proposed a novel, light-weight dual-branch splice detection framework having a spatial and compression branch. The spatial branch extracts features from the RGB domain, while the compression branch highlights the irregularities in DCT coefficients caused by the splice operation in jpg images.
- Proposed a novel Visually Attentive Splice Localization Network with Multi-Domain Feature Extractor and Multi-Receptive Field Upsampler. It contains a novel "visuall attentive multi-domain feature extractor" (VA-MDFE), "visually attentive downsampler" (VA-DS) and "visually attentive multi-receptive field upsampler" (VA-MRFU). VA-MDFE extracts attentional features from the RGB, edge and depth domain of the input image. VA-DS is responsible for fusing multi-domain features and downsampling them. VA-MRFU upsamples the features using convolution operation with multiple receptive fields.

The following research works form the basis of this chapter:

- **A. Yadav** and D. K. Vishwakarma, "Toward effective image forensics via a novel computationally efficient framework and a new image splice dataset," **Signal, Image and Video Processing**, 2024.
- **A. Yadav** and D. K. Vishwakarma, "A Visually Attentive Splice Localization Network with Multi-Domain Feature Extractor and Multi-Receptive Field Upsampler." Under Review in **IEEE Signal Processing Letters,** (https://arxiv.org/abs/2401.06995, 2024).

Chapter 5: Role of Visual Attention in Manipulation Detection

5.1 Scope of this Chapter

This chapter studies the tradeoff between performance and computational complexity for different visual attention mechanisms in a face manipulation detection model. Specifically, five recently proposed visual attention models are integrated with a baseline deep learning model, and their relative performance and computational costs are evaluated. Experimental results clearly indicate that an increase in the computational cost of the visual attention mechanism does not necessarily predict a similar increase in the performance in detecting facial manipulation.

5.2 Investigating the Impact of Visual Attention Models in Face Forgery Detection

5.2.1 Abstract

With the recent rise of realistic face manipulation methods, building robust face tampering detection methods has become more important than ever before. Visual attention has played an important role in highlighting discriminative regions within input, which is important for making accurate predictions. This research work presents a comparative study of several recently proposed visual attention models for the problem of face forgery detection. Specifically, five visual attention models, namely, coordinate, selective kernel, triplet, CoT, and shuffle attention, have been tested by integrating with a baseline deep learning model. The modified visually attentive architectures are trained and tested on the popular public benchmark dataset FaceForensics++. The experimental results achieved by different attention approaches are compared. Additionally, the computational costs involved in each type of attention have also been discussed specifying the accuracy and computation tradeoff. Experimental results prove that Triplet Attention performs best by achieving accuracy scores of 0.9543 and 0.7190 on the DF and NT categories of the FF++ dataset. Triplet attention is also extremely lightweight with only 1200 trainable parameters compared to the other attention modules under study.

5.2.2 Methodology

This section describes several attention modules and the baseline model used to evaluate them.

5.2.2.1 Shuffle Attention

Shuffle attention [104] proposes an efficient way to fuse attention mechanisms across the spatial and channel dimensions of features without increasing the computational cost. Specifically, a given input $\mathcal{X} \in \mathbb{R}^{H \times W \times C}$, is sub-divided into $\mathcal{G}$ sub-groups along the channel dimension having a shape $\mathcal{X}_k \in \mathbb{R}^{H \times W \times C/\mathcal{G}}$. Each $\mathcal{X}_k$ is further sub-divided into 2-branches having shape $H \times W \times C/2\mathcal{G}$. The two branches compute spatial and channel attention respectively and the reduced computation is enforced by the subdivision of input along the channel dimension as shown in Fig. 39.

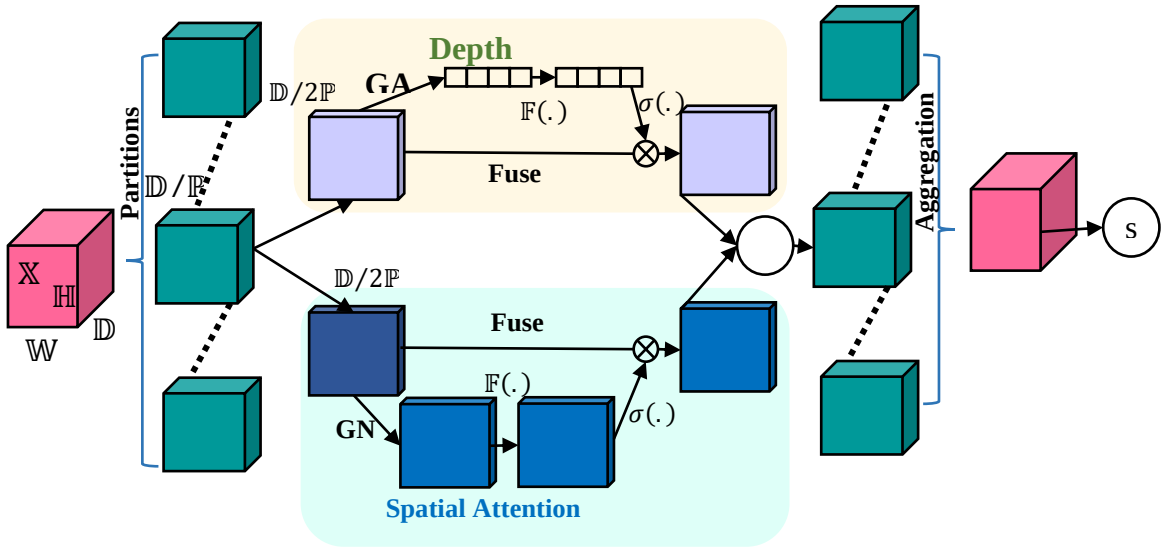


Fig. 39 Structure of Shuffle Attention

5.2.2.2 Selective Kernel (SK) Attention

Selective Kernel Attention [101] is a three-stage method that uses multiple-sized convolutional kernels for processing inputs with adaptive receptive fields. Specifically, it uses 3×3 and 5×5 kernels to extract multi-scale features in the 'split' stage. For input $\mathcal{X} \in \mathbb{R}^{H' \times W' \times C'}$ the method uses two convolutional kernels to produce $(\tilde{\mathcal{U}} \in \mathbb{R}^{H \times W \times C})$ and $(\hat{\mathcal{U}} \in \mathbb{R}^{H \times W \times C})$ feature maps. The 'fuse' stage fuses discriminative information from each receptive field branch using the summation operation and followed by global average pooling $\mathcal{GP}(.)$ along the channel dimension as given by Eq. 29:

$$\mathcal{U} = \mathcal{GP}(\tilde{\mathcal{U}} + \hat{\mathcal{U}}) \tag{29}$$

Finally, the 'select' operation employs the soft attention mechanism across feature channels to highlight important channels.

5.2.2.3 CoT Attention

CoT attention [103] or Contextual Transformer attention improves the transformer-based self-attention mechanism by searching for important modeling dependencies among neighboring keys. Specifically, the proposed attention model combines self-attention learning with contextual feature mining by finding relevant relationships among neighboring keys for each query-key pair.

For input $\mathcal{X} \in \mathbb{R}^{H\times W\times C}$ and keys ($\mathcal{K} = \mathcal{X}$), query ($\mathcal{Q} = \mathcal{X}$) and value ($\mathcal{V} = \mathcal{X}\mathcal{W}_v$), the CoT block uses $k \times k$ grouped convolution to learn contextual keys ($\mathcal{K}^1 \in \mathbb{R}^{H\times W\times C}$) representing the relationship of neighboring keys. Finally, the attention matrix is derived by Eq 30.:

$$\mathcal{A} = [\mathcal{K}^1, \mathcal{Q}]\mathcal{W}_\theta\mathcal{W}_\delta \tag{30}$$

The final attentional feature maps are given by ($\mathcal{K}^2$) by Eq. 31:

$$\mathcal{K}^2 = \mathcal{V} \circledast \mathcal{A} \tag{31}$$

5.2.2.4 Triplet Attention

Triplet attention [102] is a three-branch structure that uses the 'rotation' operation to capture discriminative regions within input in three directions while keeping minimal computation overheads. The triplet attention contains three branches. Two branches are responsible for processing the input of shape $\mathcal{X} \in \mathbb{R}^{H\times W\times C}$ and using a novel Z-pool layer to mine important channel features along the H and W dimension while the third branch works like a traditional spatial attention module.

The highlight of the triplet attention module is that it is extremely lightweight and its computation does not increase with increased input dimensions.

5.2.2.5 Coordinate Attention

Coordinate Attention [100] is a novel channel attention mechanism that captures long-range feature dependencies along one spatial dimension and also preserves exact positional encodings along another spatial dimension. Unlike common channel attention mechanisms that use 2D global pooling, coordinate attention uses two 1D kernels. For an input $\mathcal{X} \in \mathbb{R}^{H\times W\times C}$, two 1D

pooling kernels of shape $\mathbb{R}^{H\times 1}$ and $\mathbb{R}^{1\times W}$ are used to encode discriminative features along the height and width in each channel thereby yielding direction-aware feature maps.

5.2.2.6 Baseline Architecture

To perform an effective comparison of the abovementioned attention mechanisms the baseline architecture from [155] has been used. Attention layers are attached after each bottleneck layer of this architecture.

5.2.3 Experimental Setup

This section explains the experimental steps taken to establish the validity of the proposed model.

5.2.3.1 Datasets

FaceForensics++: The FaceForensics++ (FF++) [111] contains multiple face tampering examples such as Deepfakes (DF) [112], FaceSwap (FS) [113], Face2Face (F2F) [114], FaceShifter [115] and Neural Textures (NT) [116]. Each manipulation category contains 1000 videos created from 1000 original samples. Videos are provided in three qualities: raw, high (c23) and low (c40) compression. The c23 samples are used in this experiment. In this research work, experiments are conducted on the DF and NT categories of FF++ dataset.

5.2.3.2 Classification Metrics

The classification metrics used in this experiment are given in the table below.

Table XXXIV Classification metrics used in this experiment

Metric	Formula	Range
Accuracy (ACC)	$\frac{\mathbb{TP}+\mathbb{TN}}{\mathbb{TP}+\mathbb{TN}+\mathbb{FP}+\mathbb{FN}}$	[0,1]
Precision (P)	$\frac{\mathbb{TP}}{\mathbb{TP}+\mathbb{FP}}$	[0,1]
Recall (R)	$\frac{\mathbb{TP}}{\mathbb{TP}+\mathbb{FN}}$	[0,1]
F1 score (F1)	$2*\frac{Precision*Recall}{Precision+Recall}$	[0,1]
Area Under Curve (AUC)	-	[0,1]
Mathews Correlation Coefficient (MCC)	$\frac{\mathbb{TP}*\mathbb{TN}-\mathbb{FP}*\mathbb{FN}}{\sqrt{(\mathbb{TP}+\mathbb{FP})(\mathbb{TP}+\mathbb{FN})(\mathbb{TN}+\mathbb{FP})(\mathbb{TN}+\mathbb{FN})}}$	[-1,1]

These metrics are accuracy, precision, recall, F1 score, AUC score and MCC score.

5.2.3.3 Hardware Specifications

All experiments are executed on two 24 GB NVIDIA A5000 GPUs in parallel. The system memory is 128GB.

5.2.3.4 Preprocessing

This section describes the preprocessing steps followed in this experiment.

Face Extraction: Popular deepfake detection algorithms have mostly used the dlib library [24, 28, 120] or MTCNN [121, 110, 122] for face detection and extraction. RetinaFace [34] is used in this experiment to extract facial images from video frames given its low failure rate compared to MTCNN [119].

Resizing, Normalization and Data Augmentation: Facial images that are cropped from video frames are resized to 128×128. Pixel values are normalized to the range $[0,1]$. Facial images are flipped randomly in both vertical and horizontal directions with a flipping probability of 0.5.

5.2.3.5 Hyperparameters and Training Conditions

All experiments are run for 30 epochs. The batch size is set to 4. The Adam optimizer is used to update the model weights. The initial learning rate is set to 0.01. A linear learning rate decay is employed that reduces the learning rate by 10% after every 2 epochs.

5.2.3.6 Model Weight Initialization

Deep models when initialized with pre-trained ImageNet weights have performed better on classification tasks as compared to random weight initialization. Hence, the model weights in this experiment are initialized with ImageNet pre-trained weights provided on the Pytorch framework website.

5.2.3.7 System & Software Requirements

The proposed model requires at least 16GB memory of NVIDIA graphic card. The secondary storage required for storage of the FF++ dataset is 10GB. All coding experiments are conducted in the Python language. PyTorch framework is used to design and train the neural network. Jupyter Notebook IDE has been used to write the code.

5.2.4 Experimental Results & Analysis

This section presents the results obtained by each type of visual attention mechansims under study in this experiment.

5.2.4.1 Performance Analysis of Visual Attention Models

Table XXXV shows the performance of each attention mechanism on the DF (FF++) dataset. The triplet attention mechanism performs best with 0.9543 accuracy and 0.9874 AUC score. The shuffle attention achieves the highest Recall score and Coordinate attention is best in terms of the F1 score.

Table XXXV Results of each attention module on the DF (FF++) dataset.

Attention Modules	ACC	P	R	F1	AUC	MCC
Coordinate	0.9408	0.9167	0.9589	**0.9373**	0.9842	0.8822
Selective Kernel	0.8901	0.9263	0.8532	0.8882	0.9649	0.7831
Triplet	**0.9543**	**0.9415**	0.9566	0.9490	**0.9874**	**0.9077**
CoT	0.9214	0.9224	0.9015	0.9118	0.9717	0.8411
Shuffle	0.8480	0.7656	**0.9798**	0.8596	0.9560	0.7248

Table XXXVI presents the performance of the various attention mechanisms on the NT (FF++) dataset. Here again, Triplet attention has the best scores against other attention mechanisms. Coordinate attention gets the highest precision score of 0.7511 while triplet attention scores highest on all other metrics.

Table XXXVI Results of each attention module on the NT (FF++) dataset.

Attention Modules	ACC	P	R	F1	AUC	MCC
Coordinate	0.7104	**0.7511**	0.6763	0.7112	0.7917	0.4248
Selective Kernel	0.6479	0.6815	0.6269	0.6531	0.7008	0.2980
Triplet	**0.7190**	0.6486	**0.9347**	**0.7658**	**0.8592**	**0.4897**
CoT	0.6409	0.5952	0.8579	0.7028	0.7548	0.3165
Shuffle	0.6556	0.6149	0.7747	0.6856	0.7093	0.3262

The NT dataset is more challenging and hence the scores achieved are lower than those achieved on the DF dataset.

Fig. 40 and Fig. 41 present the comparison of accuracy scores of various attention mechanisms on the DF and NT datasets of FF++ respectively. In both cases, Triplet Attention outperforms all other attention mechanisms while coordinate attention comes second in terms of performance.

Fig. 42 and Fig. 43 demonstrate the AUC-ROC curve for attention mechanisms on the DF and NT datasets of FF++ respectively.

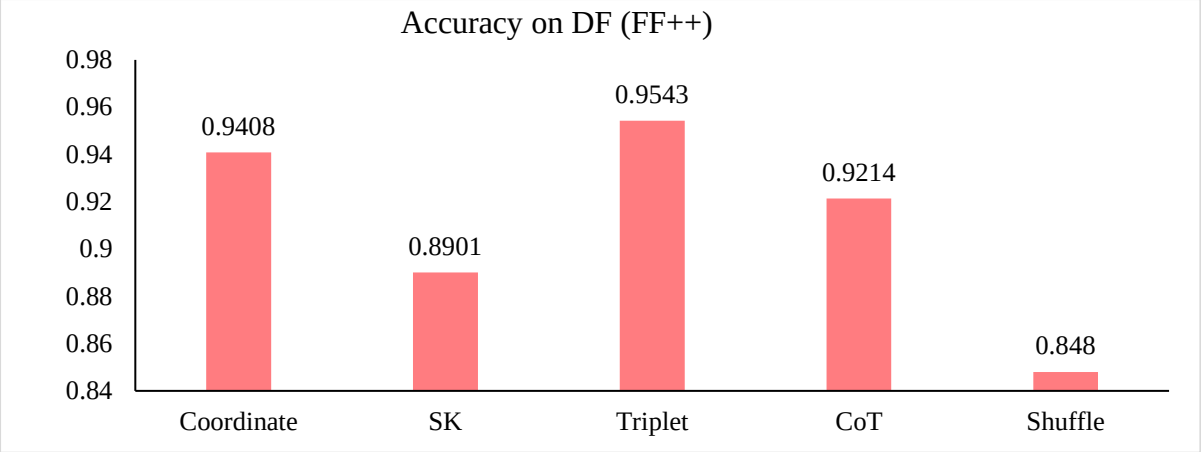


Fig. 40 Accuracy comparison of attention modules on the DF (FF++) dataset.

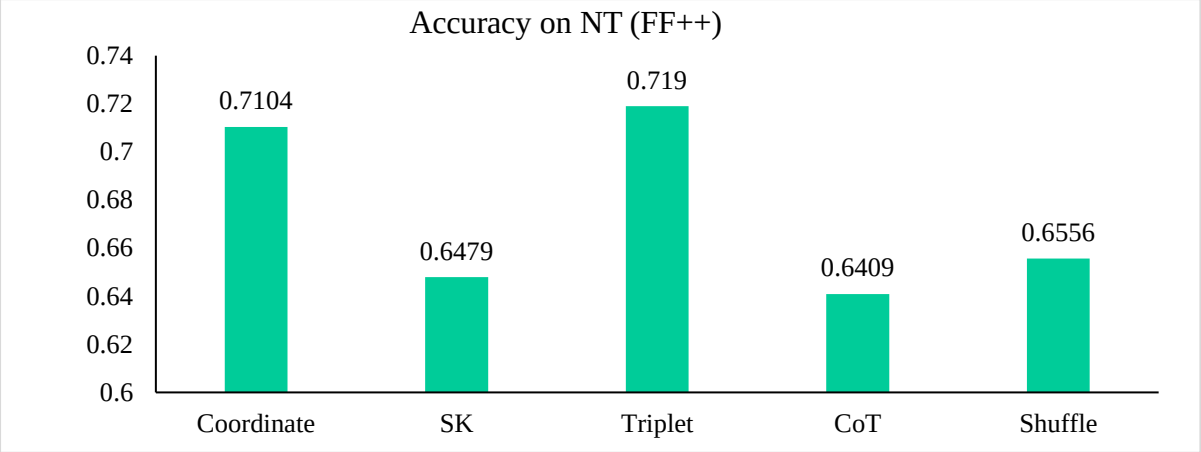


Fig. 41 Accuracy comparison of attention modules on the NT (FF++) dataset

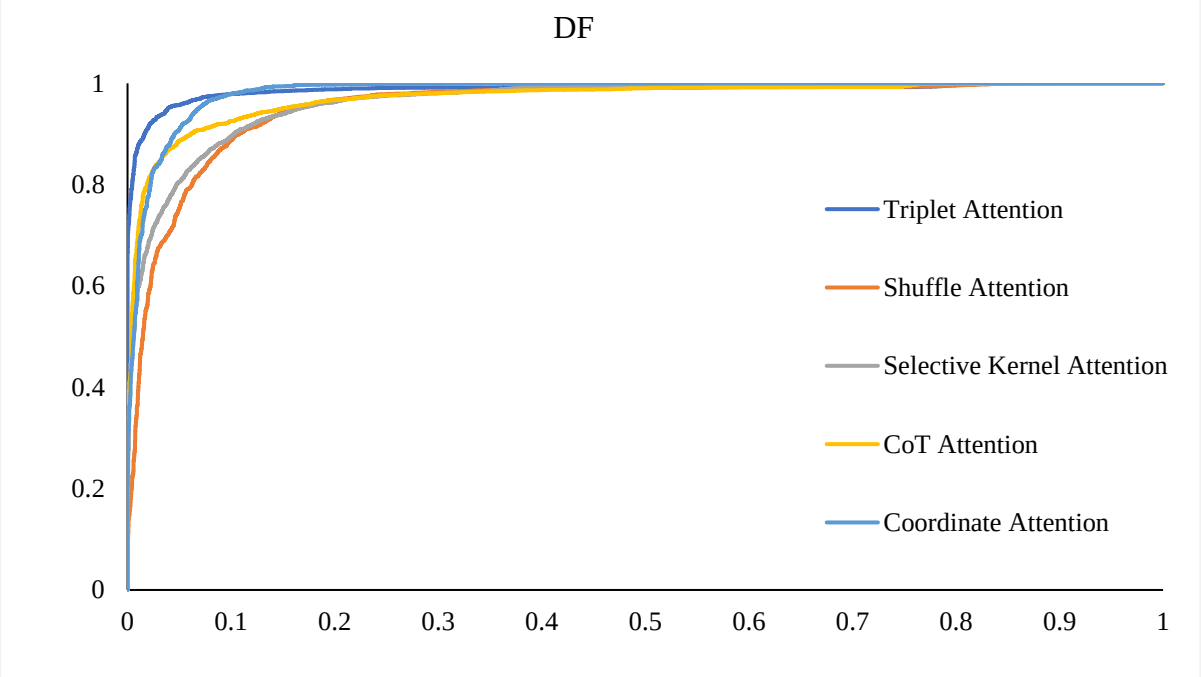


Fig. 42 AUC curves for various attention modules on the DF (FF++) dataset.

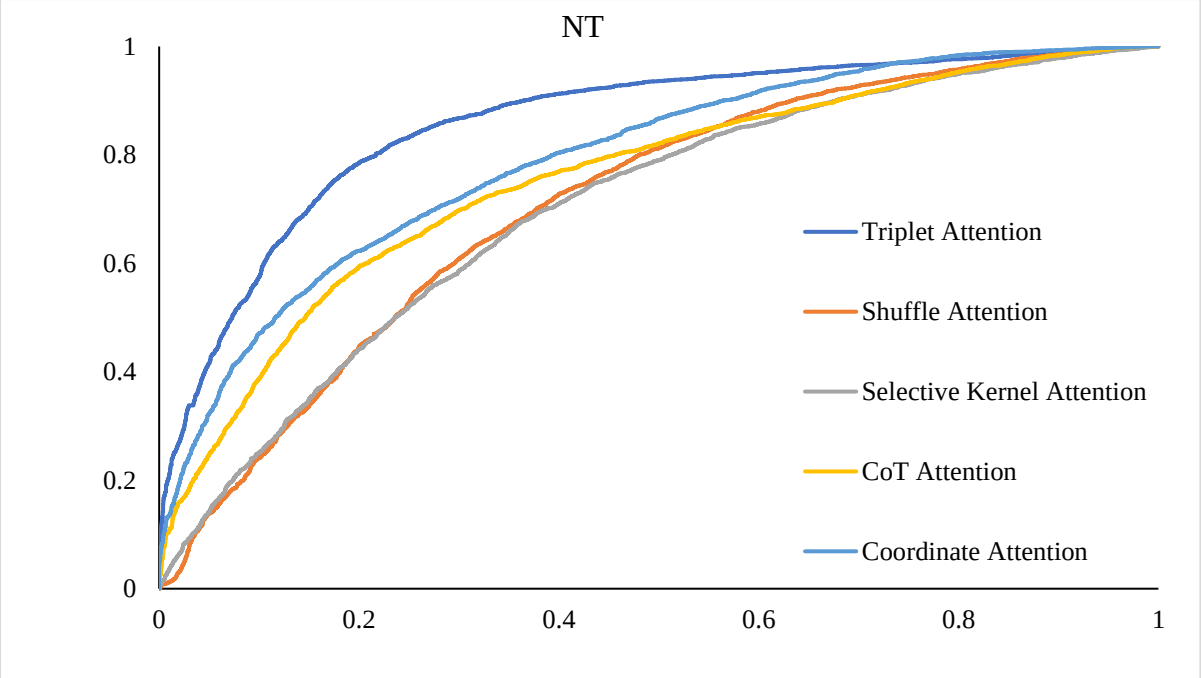


Fig. 43 AUC curves for various attention modules on the NT (FF++) dataset

The Triplet attention achieves the highest AUC score of 0.9874 on the DF dataset and 0.8592 on the NT dataset, clearly stating that the model is highly confident in its predictions.

5.2.4.2 Complexity Analysis of Visual Attention Models

This section presents a discussion of the computational overheads involved in implementing each attention mechanism. As discussed earlier, each attention mechanism has been integrated by using four attention layers, one for each skip connection block of the baseline model.

Fig. 44 presents a comparison of the number of parameters for each type of attention mechanism. The baseline model without any attention layers has 23.5 million parameters (blue bar). Triplet, Coordinate and Shuffle attention layers (green bars) are extremely lightweight having only 530280, 1200 and 1440 parameters respectively. Selective Kernel (SK) and CoT attention (green bars) are computationally heavy having 61.3 million and 48.7 million parameters respectively.

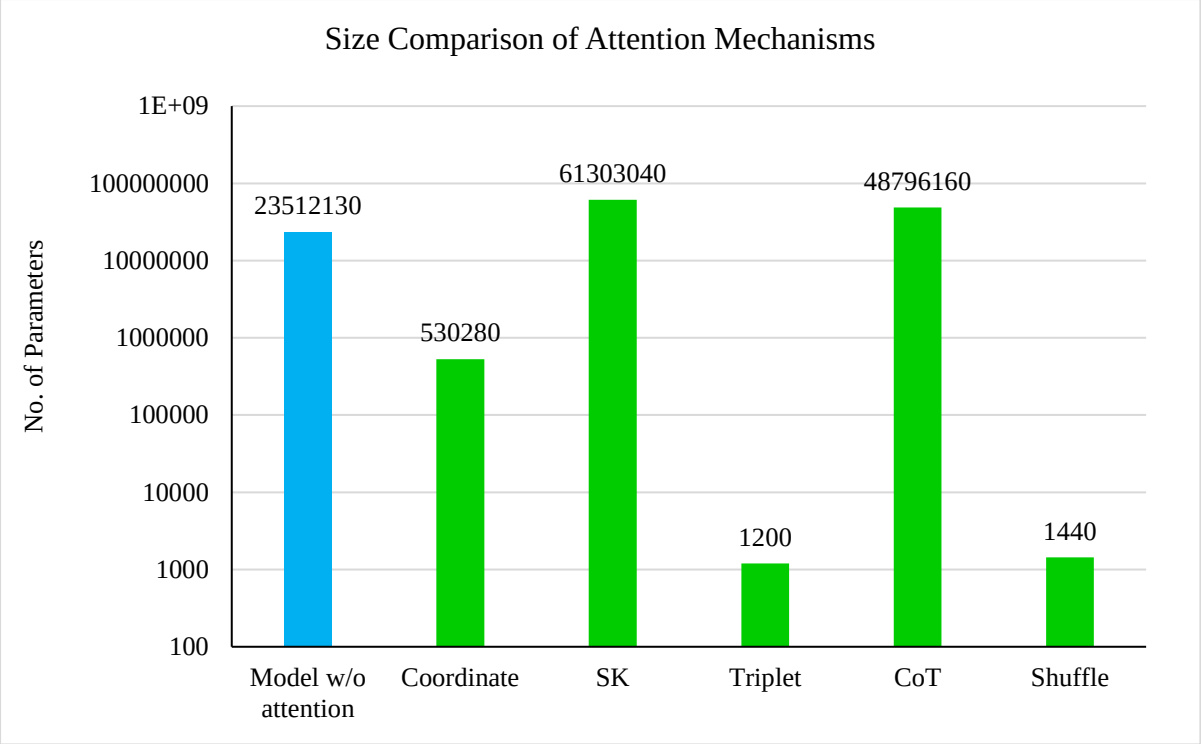


Fig. 44 Size comparison of attention mechanisms in terms of th number of parameters.

Fig. 45 shows the percentage distribution of parameters for each type of attention mechanism. The Coordinate, Triplet and Shuffle attention are lightweight occupying only 2.2056%, 0.0051% and 0.0061% of total model parameters respectively. The Selective Kernel attention and CoT attention are heavier attention modules having 72.2784% and 67.4835% of the total model parameters.

5.2.5 Conclusion

This experiment compares the recently proposed visual attention models for face forgery detection. Specifically, a baseline CNN has been enhanced with several recently proposed attention modules and their impact has been studied. The study has been conducted regarding the performance on the FaceForensics++ dataset. The computational complexity of different attention mechanisms has also been analyzed and compared visually. It is clear from the conducted experiment that different attention modules employ different computational overheads to existing CNN parameters to highlight important spatial regions of input.

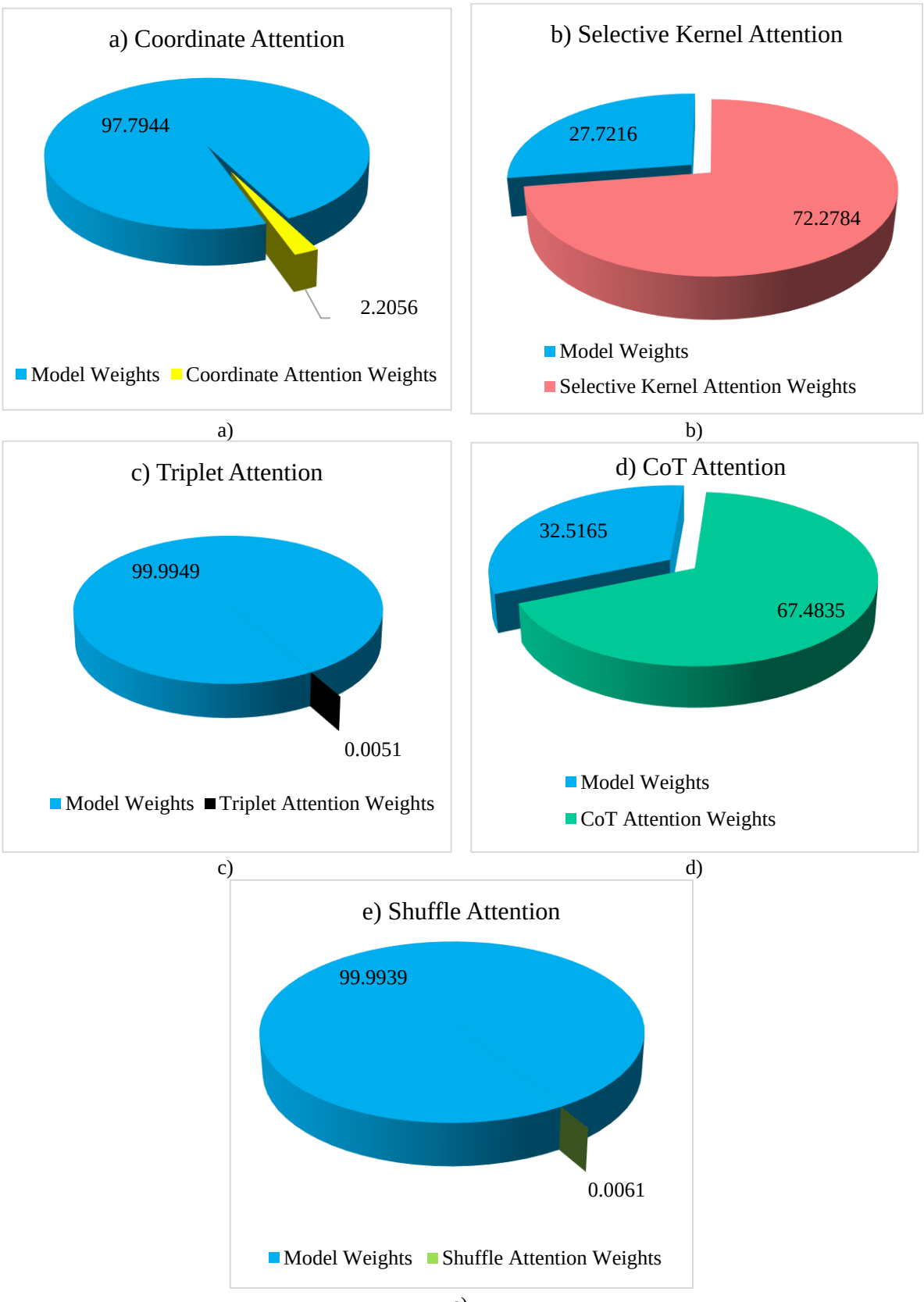


Fig. 45 The parameter distribution of each attention mechanisms, a) coordinate attention b) selective kernel attention c) triplet attention d) CoT attention e) shuffle attention.

Considering the performance of each of the five attention models and their corresponding computational cost, the Triplet Attention is the best attention model in the context of forgery

detection. It achieves a high accuracy of 0.9543 on the DF dataset while only adding 1200 additional trainable parameters to the ResNet architecture. Other attention models, such as CoT and Selective Kernel Attention, are computationally expensive as they add many parameters but do not significantly boost performance compared to the Triplet Attention mechanism.

5.3 Significant Outcomes of this Chapter

The significant outcomes of this chapter are as follows:

- Studied the role of visual attention models in face forgery detection.
- A baseline CNN is combined with five recently proposed attention mechanisms, namely, coordinate, selective kernel, triplet, CoT, and shuffle attention.
- The relative performance of these five attention models is compared. The performance vs computational cost tradeoff for each of these attention mechanisms is presented.

The following research works form the basis of this chapter:

- **A. Yadav** and D. K. Vishwakarma, "Investigating the Impact of Visual Attention Models in Face Forgery Detection", in **International Conference on Applied Intelligence and Sustainable Computing** (ICAISC), Dharwad, Karnataka, 2023.

Chapter 6: Conclusion & Future Scope

6.1 Conclusion

This chapter concludes the research work done in this thesis. Overall, four novel deep learning-based architectures are proposed for manipulation detection in multimedia content. The first two models are dedicated to the problem of face manipulation detection. The other two architectures detect and localize image splice manipulation. A novel image splice dataset is also proposed. The details are as follows:

- A novel face manipulation detection model, MRT-Net, is proposed to combine the manipulation residual and textural features extracted from its dual-branch design to predict face forgery. It contains an auto-adaptive weighting mechanism that allows it to dynamically choose the best proportion of the two features for the final prediction. Experimental results on the FF++, DFDC, and CelebDF datasets clearly establish the superiority of the proposed model.
- Another novel deepfake detection network, called Face-NeSt, is proposed. Face-NeSt leverages multi-scale features extracted from different depths of a standard baseline convolutional neural network. It contains a novel adaptively weighted multi-scale attentional module that inputs four scales of features at different scales, applies visual attention, and aggregates them together according to their degree of relevance in the final prediction. Experimental results on the FF++, DFDC and CelebDF datasets show that Face-NeSt outperforms the existing deepfake detection models.
- A novel image splice detection dataset has been proposed. It contains spliced samples generated from Python code as well as the Adobe Photoshop software. A novel dual-branch splice detection framework is proposed to detect splicing in images. It contains a spatial branch that leverages transfer learning to detect spatial clues of manipulation without adding any significant computational costs. It contains a compression branch that tracks inconsistencies in the DCT coefficients of JPG images caused by the splice manipulation operation. Experimental results establish the benefits of the proposed framework and the proposed dataset.
- A novel splice localization network is proposed to find regions of forgery within images. A "visually attentive multi-domain feature extractor" (VA-MDFE) extracts attentional features from the RGB, edge and depth domains. Next, a "visually attentive

downsampler" (VA-DS) is responsible for fusing and downsampling the multi-domain features. Finally, a novel "visually attentive multi-receptive field upsampler" (VA-MRFU) module employs multiple receptive field-based convolutions to upsample attentional features by focussing on different information scales. Experimental results conducted on the public benchmark dataset CASIA v2.0 prove the potency of the proposed model.

- The role of visual attention models is studied in detecting face forgery. Five recently proposed visual attention models are integrated with a baseline convolutional neural network. The performance boost is attained due to the study of each attention mechanism. The computational cost added due to the integration of each type of attention layer is also presented. Finally, a tradeoff between the performance boost and computational cost overhead is presented for each type of visual attention in this study.

6.2 Future Scope

In recent years, extensive research has been conducted to detect manipulation in multimedia content. While the performance has consistently improved in detecting or localizing these manipulations, several promising research directions need to be addressed.

- **Explainable AI:** Deep learning has mostly been used as a black-box tool where the model predicts or localizes the manipulation. However, interpreting why the model predicts the given output remains a mystery. Some tools help to understand the relative significance of the learned weights. These include plotting the class activation maps (CAMs). More research work needs to be done in this direction to improve the explainability of deep models.
- **Robustness to Adversarial Attacks:** While the performance of deep-learning models has consistently increased in detecting manipulation, recent studies indicate that these models are highly prone to adversarial attacks. Introducing noise in the input pixel values can easily vary the predictions of a trained model. Improving the robustness of deep-learning models against adversarial attacks is a crucial future work direction.
- **Deployment:** While the theoretical research has gained leaps and bounds in detecting manipulation, deploying these deep-learning models remains a challenge, given their high computational costs. More research work needs to be dedicated towards deployment issues for the end-user in the form of an application or web-based

framework. The increasing capacity of recent hardware facilitates the use of such heavy computational models on mobile devices.

- **Multi-modal Approaches:** Multi-modal approaches have performed better than single-modality models due to the complementary feature of learning from multiple modalities. However, this comes at the additional computational cost of having multi-branch architectures with more parameters than a single-branch model. More research needs to be done to consistently use the benefits of the multi-modal approaches without significantly adding to the associated computational cost.

References

[1] B. Dean, "Social Network Usage & Growth Statistics: How Many People Use Social Media in 2020?," BackLinko, August 2020. [Online]. Available: https://backlinko.com/social-media-users.

[2] Y. Mirsky and W. Lee, "The Creation & Detection of Deepfakes - A Survey," *ACM Computing Surveys,* vol. 54, pp. 1-41, 2022.

[3] J. Horváth, D. M. Montserrat, H. Hao and E. J. Delp, "Manipulation Detection in Satellite Images Using Deep Belief Networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, 2020.

[4] P. Johnston, E. Elyan and C. Jayne, "Video tampering localisation using features learned from authentic content," *Neural Computing and Applications,* vol. 32, p. 12243–12257, 2020.

[5] Y. Zhu, C. Chen, G. Yan, Y. Guo and Y. Dong, "AR-Net: Adaptive Attention and Residual Refinement Network for Copy-Move Forgery Detection," *IEEE Transactions on Industrial Informatics,* vol. 16, no. 10, pp. 6714 - 6723, 2020.

[6] A. Islam, C. Long, A. Basharat and A. Hoogs, "DOA-GAN: Dual-Order Attentive Generative Adversarial Network for Image Copy-Move Forgery Detection and Localization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* Seattle, 2020.

[7] S. Tyagi and D. Yadav, "A Comprehensive Review on Image Synthesis with Adversarial Networks: Theory, Literature, and Applications," *Archives of Computational Methods in Engineering,* vol. 29, p. 2685–2705, 2022.

[8] S. Kaur, S. Singh, M. Kaur and H.-N. Lee, "A Systematic Review of Computational Image Steganography Approaches," *Archives of Computational Methods in Engineering,* vol. 29, p. 4775–4797, 2022.

[9] S. Bharathiraja, B. R. Kanna and M. Hariharan, "A Deep Learning Framework for Image Authentication: An Automatic Source Camera Identification Deep-Net," *Arabian Journal for Science and Engineering,* 2022.

[10] S. Suratkar and F. Kazi, "Deep Fake Video Detection Using Transfer Learning Approach," *Arabian Journal for Science and Engineering,* 2022.

[11] M. M. A. Alhaidery, A. H. Taherinia and H. I. Shahadi, "A robust detection and localization technique for copy-move forgery in digital images," *Journal of King Saud University - Computer and Information Sciences,* vol. 35, no. 1, pp. 449-461, 2023.

[12] S. L. Abdulwahid , "The detection of copy move forgery image methodologies," *Measurement: Sensors ,* vol. 26, 2023.

[13] 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.

[14] 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.

[15] K. Sharma, G. Singh and P. Goyal, "IPDCN2: Improvised Patch-based Deep CNN for facial retouching detection," *Expert Systems with Applications,* vol. 211, 2023.

[16] 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.

[17] D. M. Montserrat, H. Hao, S. K. Yarlagadda, S. Baireddy, R. Shao, J. Horváth, E. Bartusiak, J. Yang, D. Güera, F. Zhu and E. J. Delp, "Deepfakes Detection with Automatic Face Weighting," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, 2020.

[18] M. S. Rana, M. N. Nobi, B. Murali and A. H. Sung, "Deepfake Detection: A Systematic Literature Review," *IEEE Access,* vol. 10, pp. 25494 - 25513, 2022.

[19] F. Chamot, Z. Geradts and E. Haasdijk, "Deepfake forensics: Cross-manipulation robustness of feedforward- and recurrent convolutional forgery detection methods," *Forensic Science International: Digital Investigation,* vol. 40, 2022.

[20] A. Malik, M. Kuribayashi, S. M. Abdullahi and A. N. Khan, "DeepFake Detection for Human Face Images and Videos: A Survey," *IEEE Access,* vol. 10, pp. 18757 - 18775, 2022.

[21] A. H. Khalifa, N. A. Zaher, A. S. Abdallah and M. W. Fakhr, "Convolutional Neural Network Based on Diverse Gabor Filters for Deepfake Recognition," *IEEE Access,* vol. 10, pp. 22678 - 22686, 2022.

[22] M. Mustak, J. Salminen, M. Mäntymäki, A. Rahman and Y. K. Dwivedi, "Deepfakes: Deceptions, mitigations, and opportunities," *Journal of Business Research,* vol. 154, 2023.

[23] A. Yadav and D. K. Vishwakarma, "MRT-Net: Auto-adaptive weighting of manipulation residuals and texture clues for face manipulation detection," *Expert Systems with Applications,* vol. 232, 2023.

[24] M. Bonomi, C. Pasquini and G. Boato, "Dynamic texture analysis for detecting fake faces in video sequences," *Journal of Visual Communication and Image Representation,* vol. 79, 2021.

[25] 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.

[26] 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.

[27] Z. Guo, G. Yang, D. Wang and D. Zhang, "A data augmentation framework by mining structured features for fake face image detection," *Computer Vision and Image Understanding,* vol. 226, 2023.

[28] 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.

[29] 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.

[30] Z. Tan, Z. Yang, C. Miao and G. Guo, "Transformer-Based Feature Compensation and Aggregation for DeepFake Detection," *IEEE Signal Processing Letters,* vol. 29, pp. 2183 - 2187, 2022.

[31] B. Chen, T. Li and W. Ding, "Detecting deepfake videos based on spatiotemporal attention and convolutional LSTM," *Information Sciences,* vol. 601, pp. 58-70, 2022.

[32] S. Ganguly, A. Ganguly, S. Mohiuddin, S. Malakar and R. Sarkar, "ViXNet: Vision Transformer with Xception Network for deepfakes based video and image forgery detection," *Expert Systems with Applications,* vol. 210, 2022.

[33] W. Pu, J. Hu, X. Wang, Y. Li, S. Hu, B. Zhu, R. Song, Q. Song, X. Wu and S. Lyu, "Learning a deep dual-level network for robust DeepFake detection," *Pattern Recognition,* vol. 130, 2022.

[34] Z. Xia, T. Qiao, M. Xu, N. Zheng and S. Xie, "Towards DeepFake video forensics based on facial textural disparities in multi-color channels," *Information Sciences,* vol. 607, pp. 654-669, 2022.

[35] S. Kingra, N. Aggarwal and N. Kaur, "LBPNet: Exploiting texture descriptor for deepfake detection," *Forensic Science International: Digital Investigation,* Vols. 42-43, 2022.

[36] H. Wang, Z. Liu and S. Wang, "Exploiting Complementary Dynamic Incoherence for DeepFake Video Detection," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 33, no. 8, pp. 4027 - 4040, 2023.

[37] J. Ma, S. Wang, A. Zhang and A. W.-C. Liew, "Feature Extraction For Visual Speaker Authentication Against Computer-Generated Video Attacks," in *IEEE International Conference on Image Processing (ICIP)*, Abu Dhabi, 2020.

[38] C.-Z. Yang, J. Ma, S. Wang and A. W.-C. Liew, "Preventing DeepFake Attacks on Speaker Authentication by Dynamic Lip Movement Analysis," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1841-1854, 2020.

[39] K. Lin, W. Han, S. Li, H. Zhao, J. Ren, L. Zhu and J. Lv, "IR-Capsule: Two-Stream Network for Face Forgery Detection," *Cognitive Computation*, vol. 15, pp. 13-22, 2023.

[40] A. Fogelton and W. Benesova, "Eye blink completeness detection," *Computer Vision and Image Understanding*, Vols. 176-177, pp. 78-85, 2018.

[41] S. Fernandes, S. Raj, E. Ortiz, L. Vintila, M. Salter, G. Urosevic and S. Jha, "Predicting Heart Rate Variations of Deepfake Videos using Neural ODE," in *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Seoul, 2019.

[42] H. Qi, Q. Guo, F. Xu, X. Xie, L. Ma, W. Feng, Y. Liu and J. Zhao, "DeepRhythm: Exposing DeepFakes with Attentional Visual Heartbeat Rhythms," in *28th ACM International Conference on Multimedia*, Lisboa, 2020.

[43] 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.

[44] Z. Wang, Y. Guo and W. Zuo, "Deepfake Forensics via an Adversarial Game," *IEEE Transactions on Image Processing*, vol. 31, pp. 3541 - 3552, 2022.

[45] P. Korshunov and S. Marcel, "Improving Generalization of Deepfake Detection With Data Farming and Few-Shot Learning," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 3, pp. 386 - 397, 2022.

[46] J. Wang, Y. Sun and J. Tang, "LiSiam: Localization Invariance Siamese Network for Deepfake Detection," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2425 - 2436, 2022.

[47] M. Du, S. Pentyala, Y. Li and X. Hu, "Towards Generalizable Deepfake Detection with Locality-aware AutoEncoder," in *29th ACM International Conference on Information & Knowledge Management*, Atlanta, 2020.

[48] J. Hu, S. Wang and X. Li, "Improving the Generalization Ability of Deepfake Detection via Disentangled Representation Learning," in *IEEE International Conference on Image Processing (ICIP)*, Anchorage, 2021.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Long Beach, 2017.

[50] H. Dang, F. Liu, J. Stehouwer, X. Liu and A. K. Jain, "On the Detection of Digital Face Manipulation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, 2020.

[51] D. H. Choi, H. J. Lee, S. Lee, J. U. Kim and Y. M. Ro, "Fake Video Detection With Certainty-Based Attention Network," in *IEEE International Conference on Image Processing (ICIP)*, Abu Dhabi, 2020.

[52] K. Chugh, P. Gupta, A. Dhall and R. Subramanian, "Not made for each other- Audio-Visual Dissonance-based Deepfake Detection and Localization," in *28th ACM International Conference on Multimedia*, Lisboa, 2020.

[53] 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 *28th ACM International Conference on Multimedia*, Lisboa, 2020.

[54] B. Chu, W. You, Z. Yang, L. Zhou and R. Wang, "Protecting World Leader Using Facial Speaking Pattern Against Deepfakes," *IEEE Signal Processing Letters* , vol. 29, pp. 2078 - 2082, 2022.

[55] W. Chen, Y. Q. Shi and . W. Su, "Image splicing detection using 2-D phase congruency and statistical moments of characteristic function," in *Security, Steganography, and Watermarking of Multimedia Contents IX*, San Jose, 2007.

[56] X. Zhao, J. Li, S. Li and S. Wang, "Detecting Digital Image Splicing in Chroma Spaces," in *International Workshop on Digital Watermarking*, Seoul, 2010.

[57] 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.

[58] H. Gou, A. Swaminathan and M. Wu, "Noise Features for Image Tampering Detection and Steganalysis," in *IEEE International Conference on Image Processing*, San Antonio, 2007.

[59] Z. Tang, X. Zhang, X. Li and S. Zhang, "Robust Image Hashing With Ring Partition and Invariant Vector Distance," *IEEE Transactions on Information Forensics and Security,* vol. 11, no. 1, pp. 200-214, 2016.

[60] X. Wang, K. Pang, X. Zhou, Y. Zhou, L. Li and J. Xue, "A Visual Model-Based Perceptual Image Hash for Content Authentication," *IEEE Transactions on Information Forensics and Security* , vol. 10, no. 7, pp. 1336 - 1349, 2015.

[61] X. Cun and C.-M. Pun, "Image Splicing Localization via Semi-global Network and Fully Connected Conditional Random Fields," in *European Conference on Computer Vision (ECCV)*, Munich, 2018.

[62] X. Bi, Y. Wei, B. Xiao and W. Li, "RRU-Net: The Ringed Residual U-Net for Image Splicing Forgery Detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, 2019.

[63] B. Liu and C.-M. Pun, "Deep Fusion Network for Splicing Forgery Localization," in *European Conference on Computer Vision (ECCV)*, Munich, 2018.

[64] Z. Zhang, Y. Zhang, Z. Zhou and J. Luo, "Boundary-based Image Forgery Detection by Fast Shallow CNN," in *IEEE International Conference on Pattern Recognition (ICPR)*, Beijing, 2018.

[65] T. Pomari, G. Ruppert, E. Rezende, A. Rocha and T. Carvalho, "Image Splicing Detection Through Illumination Inconsistencies and Deep Learning," in *IEEE International Conference on Image Processing (ICIP)*, Athens, 2018.

[66] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath and A. K. R.-. 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.

[67] C. Deng, Z. Li, X. Gao and D. Tao, "Deep Multi-scale Discriminative Networks for Double JPEG Compression Forensics," *ACM Transactions on Intelligent Systems and Technology,* vol. 10, no. 2, 2019.

[68] R. Zhang and J. Ni, "A Dense U-Net with Cross-Layer Intersection for Detection and Localization of Image Forgery," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, 2020.

[69] L. Bondi, S. Lameri, D. Güera, P. Bestagini, E. J. Delp and S. Tubaro, "Tampering Detection and Localization Through Clustering of Camera-Based CNN Features," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, 2017.

[70] I. Amerini, T. Uricchio, L. Ballan and R. Caldelli, "Localization of JPEG Double Compression Through Multi-domain Convolutional Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, 2017.

[71] M. Saddique, K. Asghar, U. I. Bajwa, M. Hussain, H. A. Aboalsamh and Z. Habib, "Classification of Authentic and Tampered Video Using Motion Residual and Parasitic Layers," *IEEE Access,* vol. 8, pp. 56782 - 56797, 2020.

[72] C. Podilchuk and E. Delp, "Digital watermarking: algorithms and applications," *IEEE Signal Processing Magazine* , vol. 18, no. 4, pp. 33-46, 2001.

[73] B. Li, H. Zhang, H. Luo and S. Tan, "Detecting double JPEG compression and its related anti-forensic operations with CNN," *Multimedia Tools and Applications,* vol. 78, p. 8577–8601, 2019.

[74] Q. Wang and R. Zhang, "Double JPEG compression forensics based on a convolutional neural network," *EURASIP Journal on Information Security,* 2016.

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

[76] D. Cozzolino and L. Verdoliva, "Noiseprint: A CNN-Based Camera Model Fingerprint," *IEEE Transactions on Information Forensics and Security,* vol. 15, pp. 144 - 159, 2020.

[77] D. Cozzolino, F. Marra, D. Gragnaniello, G. Poggi and L. Verdoliva, "Combining PRNU and noiseprint for robust and efficient device source identification," *EURASIP Journal on Information Security,* 2020.

[78] J. Wang, Q. Ni, G. Liu, X. Luo and S. K. Jha, "Image splicing detection based on convolutional neural network with weight combination strategy," *Journal of Information Security and Applications,* vol. 54, 2020.

[79] S. Ye, Q. Sun and E.-C. Chang, "Detecting Digital Image Forgeries by Measuring Inconsistencies of Blocking Artifact," in *IEEE International Conference on Multimedia and Expo,* Beijing, 2007.

[80] D. Cozzolino, G. Poggi and L. Verdoliva, "Efficient Dense-Field Copy–Move Forgery Detection," *IEEE Transactions on Information Forensics and Security,* vol. 10, no. 11, pp. 2284-2297, 2015.

[81] M. Bashar, K. Noda, N. Ohnishi and K. Mori, "Exploring Duplicated Regions in Natural Images," *IEEE Transactions on Image Processing,* 2010.

[82] D.-Y. Huang, C.-N. Huang, . W.-C. Hu and C.-H. Chou , "Robustness of copy-move forgery detection under high JPEG compression artifacts," *Multimedia Tools and Applications,* vol. 76, p. 1509–1530, 2017.

[83] S.-J. Ryu, M.-J. Lee and H.-K. Lee, "Detection of Copy-Rotate-Move Forgery Using Zernike Moments," in *International Workshop on Information Hiding*, Calgary, 2010.

[84] B. Mahdian and S. Saic, "Detection of copy–move forgery using a method based on blur moment invariants," *Forensic Science International,* vol. 171, no. 2-3, pp. 180-189, 2007.

[85] I. Chingovska, A. Anjos and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *International Conference of Biometrics Special Interest Group (BIOSIG)*, Darmstadt, 2012.

[86] Y. Zhu, X. Shen and H. Chen, "Copy-move forgery detection based on scaled ORB," *Multimedia Tools and Applications,* vol. 75, p. 3221–3233, 2016.

[87] B. Shivakumar and S. Baboo, "Detection of region duplication forgery in digital images using SURF," *International Journal of Computer Science,* vol. 8, no. 4, 2011.

[88] A. Costanzo, I. Amerini, R. Caldelli and M. Barni, "Forensic Analysis of SIFT Keypoint Removal and Injection," *IEEE Transactions on Information Forensics and Security,* vol. 9, no. 9, pp. 1450 - 1464, 2014.

[89] Q. Bammey, R. G. v. Gioi and J.-M. Morel, "An Adaptive Neural Network for Unsupervised Mosaic Consistency Analysis in Image Forensics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, 2020.

[90] J.-L. Zhong and C.-M. Pun, "An End-to-End Dense-InceptionNet for Image Copy-Move Forgery Detection," *IEEE Transactions on Information Forensics and Security,* vol. 15, pp. 2134 - 2146, 2020.

[91] Y. Wu, W. A.-. Almageed and P. Natarajan, "BusterNet: Detecting Copy-Move Image Forgery with Source/Target Localization," in *European Conference on Computer Vision (ECCV)*, Munich, 2018.

[92] S.-H. Nam, W. Ahn, I.-J. Yu, M.-J. Kwon, M. Son and H.-K. Lee, "Deep Convolutional Neural Network for Identifying Seam-Carving Forgery," *IEEE Transactions on Circuits and Systems for Video Technology ,* 2020.

[93] H. Li and J. Huang, "Localization of Deep Inpainting Using High-Pass Fully Convolutional Network," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, 2019.

[94] Y. Yan, W. Ren and X. Cao, "Recolored Image Detection via a Deep Discriminative Model," *IEEE Transactions on Information Forensics and Security ,* vol. 14, no. 1, pp. 5-17, 2019.

[95] S. K. Yarlagadda, D. Güera, D. M. Montserrat, F. M. Zhu, E. J. Delp, P. Bestagini and S. Tubaro, "Shadow Removal Detection and Localization for Forensics Analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Brighton, 2019.

[96] C. Long, E. Smith, A. Basharat and A. Hoogs, "A C3D-Based Convolutional Neural Network for Frame Dropping Detection in a Single Video Shot," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, 2017.

[97] H. Lin, W. Huang, W. Luo and W. Lu, "DeepFake detection with multi-scale convolution and vision transformer," *Digital Signal Processing,* vol. 134, 2023.

[98] H. Chen, Y. Li, D. Lin, B. Li and J. Wu, "Watching the BiG artifacts: Exposing DeepFake videos via Bi-granularity artifacts☆," *Pattern Recognition,* vol. 135, 2023.

[99] F. Dong, X. Zou, J. Wang and X. Liu, "Contrastive learning-based general Deepfake detection with multi-scale RGB frequency clues," *Journal of King Saud University - Computer and Information Sciences,* vol. 35, no. 4, pp. 90-99, 2023.

[100] Q. Hou, D. Zhou and J. Feng, "Coordinate Attention for Efficient Mobile Network Design," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021.

[101] X. Li, W. Wang, X. Hu and J. Yang, "Selective Kernels Network," in *Conference on Computer Vision and Pattern Recognition*, Long Beach, California, 2019.

[102] D. Misra, T. Nalamada, A. U. Arasanipalai and Q. Hou, "Rotate to Attend: Convolutional Triplet Attention Module," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, 2021.

[103] Y. Li, T. Yao, Y. Pan and T. Mei, "Contextual Transformer Networks for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 45, no. 2, pp. 1489-1500, 2023.

[104] Q.-L. Zhang and Y.-B. Yang, "SA-Net: Shuffle Attention for Deep Convolutional Neural Networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, 2021.

[105] 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.

[106] T. Pevny, P. Bas and J. Fridrich, "Steganalysis by Subtractive Pixel Adjacency Matrix," *IEEE Transactions on Information Forensics and Security,* vol. 5, no. 2, pp. 215-224, 2010.

[107] 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.

[108] J. Yang, S. Xiao, A. Li, G. Lan and H. Wang, "Detecting fake images by identifying potential texture difference," *Future Generation Computer Systems,* vol. 125, pp. 127-135, 2021.

[109] Z. Yu, C. Zhao, Z. Wang, Y. Qin, Z. Su, X. Li, F. Zhou and G. Zhao, "Searching Central Difference Convolutional Networks for Face Anti-Spoofing," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020.

[110] C. Lu, B. Liu, W. Zhou, Q. Chu and N. Yu, "Deepfake Video Detection Using 3D-Attentional Inception Convolutional Neural Network," in *IEEE International Conference on Image Processing (ICIP)*, Anchorage, 2021.

[111] 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, Korea (South), 2019.

[112] "DeepFakes," GitHub, 14 August 2020. [Online]. Available: https://github.com/deepfakes/faceswap. [Accessed 08 July 2022].

[113] "FaceSwap," GitHub, 19 June 2016. [Online]. Available: https://github.com/MarekKowalski/FaceSwap. [Accessed 08 July 2022].

[114] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt and M. Nießner, "Face2Face: real-time face capture and reenactment of RGB videos," *Communications of the ACM,* vol. 62, no. 1, pp. 96-104, 2019.

[115] L. Li, J. Bao, H. Yang, D. Chen and F. Wen, "FaceShifter: Towards High Fidelity And Occlusion Aware Face Swapping," *https://arxiv.org/abs/1912.13457*, 2019.

[116] J. Thies, M. Zollhöfer and M. Nießner, "Deferred neural rendering: image synthesis using neural textures," *ACM Transactions on Graphics,* vol. 38, no. 4, pp. 1-12, 2019.

[117] B. Dolhansky, R. Howes, B. Pflaum, N. Baram and C. C. Ferrer, "The Deepfake Detection Challenge (DFDC) Preview Dataset," *https://arxiv.org/abs/1910.08854,* 2019.

[118] 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, WA, USA, 2020.

[119] J. Deng, J. Guo, E. Ververas, I. Kotsia and S. Zafeiriou, "RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020.

[120] J. Guo and Y. Liu, "Facial parts swapping with generative adversarial networks," *Journal of Visual Communication and Image Representation,* vol. 78, 2021.

[121] Y. Zhou, A. Luo, X. Kang and S. Lyu, "Face Forgery Detection Based On Segmentation Network," in *IEEE International Conference on Image Processing (ICIP)*, Anchorage, AK, USA, 2021.

[122] 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.

[123] G. Yang, A. Wei, X. Fang and J. Zhang, "FDS_2D: rethinking magnitude-phase features for DeepFake detection," *Multimedia Systems,* 2023.

[124] Z. Guo, G. Yang, D. Zhang and M. Xia, "Rethinking gradient operator for exposing AI-enabled face forgeries," *Expert Systems with Applications,* vol. 215, 2023.

[125] 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.

[126] K. Xu, G. Yang, X. Fang and J. Zhang, "Facial depth forgery detection," *Multimedia Tools and Applications,* 2023.

[127] G. Yang, K. Xu, X. Fang and J. Zhang, "Video face forgery detection via facial motion-assisted capturing dense optical flow truncation," *The Visual Computer,* 2022.

[128] 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,* vol. 44, no. 10, pp. 6111 - 6121, 2022.

[129] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang and N. Yu, "Spatial-Phase Shallow Learning: Rethinking Face Forgery Detection in Frequency Domain," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021.

[130] 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.

[131] 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.

[132] 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.

[133] Y. Qian, G. Yin, L. Sheng, Z. Chen and J. Shao, "Thinking in Frequency: Face Forgery Detection by Mining Frequency-Aware Clues," in *European Conference on Computer Vision,* 2020.

[134] J.-Y. Baek, Y.-S. Yoo and S.-H. Bae, "Generative Adversarial Ensemble Learning for Face Forensics," *IEEE Access,* vol. 8, pp. 45421 - 45431, 2020.

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

[136] 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, 2018.

[137] S. Asha, P. Vinod and V. G. Menon, "A defensive framework for deepfake detection under adversarial settings using temporal and spatial features," *International Journal of Information Security,* 2023.

[138] J. Ke and L. Wang, "DF-UDetector: An effective method towards robust deepfake detection via feature restoration," *Neural Networks,* vol. 160, pp. 216-226, 2023.

[139] G. Li, X. Zhao and Y. Cao, "Forensic Symmetry for DeepFakes," *IEEE Transactions on Information Forensics and Security,* vol. 18, pp. 1095 - 1110, 2023.

[140] A. V. Nadimpalli and A. Rattani, "On Improving Cross-dataset Generalization of Deepfake Detectors," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, New Orleans, LA, USA, 2022.

[141] G. Li, Y. Cao and X. Zhao, "Exploiting Facial Symmetry to Expose Deepfakes," in *IEEE International Conference on Image Processing (ICIP)*, Anchorage, 2021.

[142] L. Trinh, M. Tsang, S. Rambhatla and Y. Liu, "Interpretable and Trustworthy Deepfake Detection via Dynamic Prototypes," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, 2021.

[143] Z. Luo, S.-I. Kamata and Z. Sun, "Transformer And Node-Compressed Dnn Based Dual-Path System For Manipulated Face Detection," in *IEEE International Conference on Image Processing (ICIP)*, Anchorage, 2021.

[144] Z. Chen and H. Yang, "Attentive Semantic Exploring for Manipulated Face Detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, 2021.

[145] 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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, 2020.

[146] L. Deng, J. Wang and Z. Liu, "Cascaded Network Based on EfficientNet and Transformer for Deepfake Video Detection," *Neural Processing Letters ,* 2023.

[147] S. Mohiuddin, K. H. Sheikh, S. Malakar, J. D. Velásquez and R. Sarkar, "A hierarchical feature selection strategy for deepfake video detection," *Neural Computing and Applications,* p. 9363–9380, 2023.

[148] Y. Yu, X. Zhao, R. Ni, S. Yang, Y. Zhao and A. C. Kot, "Augmented Multi-Scale Spatiotemporal Inconsistency Magnifier for Generalized DeepFake Detection," *IEEE Transactions on Multimedia,* vol. Early Access, pp. 1-13, 2023.

[149] C. Zhao, C. Wang, G. Hu, H. Chen, C. Liu and J. Tang, "ISTVT: Interpretable Spatial-Temporal Video Transformer for Deepfake Detection," *IEEE Transactions on Information Forensics and Security,* vol. 18, pp. 1335 - 1348, 2023.

[150] Y. J. Heo, . W.-H. Yeo and . B.-G. Kim, "DeepFake detection algorithm based on improved vision transformer," *Applied Intelligence,* vol. 53, p. 7512–7527, 2023.

[151] S. Ganguly, A. Ganguly, S. Mohiuddin, S. Malakar and R. Sarkar, "ViXNet: Vision Transformer with Xception Network for deepfakes based video and image forgery detection," *Expert Systems with Applications,* vol. 210, 2022.

[152] S. Ganguly, S. Mohiuddin, S. Malakar, E. Cuevas and R. Sarkar, "Visual attention-based deepfake video forgery detection," *Pattern Analysis and Applications,* vol. 25, p. 981–992, 2022.

[153] X. Li, Y. Lang, 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 USA, 2020.

[154] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng and Y. Wei, "LayerCAM: Exploring Hierarchical Class Activation Maps for Localization," *IEEE Transactions on Image Processing ,* vol. 30, pp. 5875 - 5888, 2021.

[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] S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He, "Aggregated Residual Transformations for Deep Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017.

[157] J. Xu, Y. Pan, X. Pan, S. Hoi, Z. Yi and Z. Xu, "RegNet: Self-Regulated Network for Image Classification," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1-6, 2022.

[158] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li and A. Smola, "ResNeSt: Split-Attention Networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, New Orleans, LA, USA, 2022.

[159] C. H. Song, H. J. Han and Y. Avrithis, "All the Attention You Need: Global-Local, Spatial-Channel Attention for Image Retrieval," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, 2022.

[160] 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*, vol. 44, no. 10, pp. 6111 - 6121, 2022.

[161] 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.

[162] H. Zhao, T. Wei, W. Zhou, W. Zhang, D. Chen and N. Yu, "Multi-attentional Deepfake Detection," in *Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021.

[163] D. Liu, Z. Zheng, C. Peng, Y. Wang, N. Wang and X. Gao, "Hierarchical Forgery Classifier On Multi-modality Face Forgery Clues," *IEEE Transactions on Multimedia*, pp. 1 - 12, 2023.

[164] Y. Wang, K. Yu, C. Chen, X. Hu and S. Peng, "Dynamic Graph Learning with Content-guided Spatial-Frequency Relation Reasoning for Deepfake Detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada, 2023.

[165] Z. Shi, H. Chen, L. Chen and D. Zhang, "Discrepancy-Guided Reconstruction Learning for Image Forgery Detection," in *International Joint Conferences on Artificial Intelligence*, 2023.

[166] X. Jin, . X.-Y. Mu and J. Xu, "Searching for the Fakes: Efficient Neural Architecture Search for General Face Forgery Detection," *https://arxiv.org/abs/2306.08830*, 2023.

[167] A. Khormali and J.-S. Yuan, "Self-Supervised GraphTransformer for DeepfakeDetection," *https://arxiv.org/abs/2307.15019*, 2023.

[168] K. Sun, S. Chen, T. Yao, X. Sun, S. Ding and R. Ji, "Towards General Visual-Linguistic Face Forgery Detection," *https://arxiv.org/abs/2307.16545*, 2023.

[169] K. Sun, S. Chen, T. Yao, X. Sun, S. Ding and R. Ji, "Continual Face Forgery Detection via Historical Distribution Preserving," *https://arxiv.org/abs/2308.06217*, 2023.

[170] J. Hu, X. Liao, J. Liang, W. Zhou and Z. Qin, "FInfer: Frame Inference-Based Deepfake Detection for High-Visual-Quality Videos," in *AAAI Conference on Artificial Intelligence*, 2022.

[171] Q. Gu, S. Chen, T. Yao, Y. Chen, S. Ding and R. Yi, "Exploiting Fine-Grained Face Forgery Clues via Progressive Enhancement Learning," in *AAAI Conference on Artificial Intelligence*, 2022.

[172] 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.

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

[174] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell and S. Xie, "A ConvNet for the 2020s," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[175] M. Tan and Q. Le, "EfficientNetV2: Smaller Models and Faster Training," in *Proceedings of Machine Learning Research*, 2021.

[176] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021.

[177] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations (ICLR)*, San Diego, 2015.

[178] G. Huang, Z. Liu, L. V. D. Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 2017.

[179] T.-T. Ng, J. Hsu and S.-F. Chang, "Columbia Image Splicing Detection Evaluation Dataset," Columbia University, 2004. [Online]. Available: https://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSplicedDataSet.htm.

[180] 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.

[181] 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, 2013.

[182] T. J. d. Carvalho, C. Riess, E. Angelopoulou, H. Pedrini and A. d. R. Rocha, "Exposing Digital Image Forgeries by Illumination Color Classification," *IEEE Transactions on Information Forensics and Security,* vol. 8, no. 7, pp. 1182 - 1194, 2013.

[183] "Image Forensics Challenge Dataset," 2014. [Online]. Available: https://signalprocessingsociety.org/newsletter/2014/01/ieee-ifs-tc-image-forensics-challenge-website-new-submissions.

[184] S. Chen, S. Tan, B. Li and J. Huang, "Automatic Detection of Object-Based Forgery in Advanced Video," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 26, no. 11, pp. 2138 - 2151, 2016.

[185] D. Cozzolino, G. Poggi and L. Verdoliva, "Efficient Dense-Field Copy–Move Forgery Detection," *IEEE Transactions on Information Forensics and Security,* vol. 10, no. 11, pp. 2284 - 2297, 2015.

[186] D.-T. D. Nguyen, C. Pasquini, V. Conotter and G. Boato, "RAISE: a raw images dataset for digital image forensics," in *6th ACM Multimedia Systems Conference*, Portland, 2015.

[187] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, 2015.

[188] 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 *International Conference on Learning Representations (ICLR)*, Austria, 2021.

[189] D. Li, J. Hu, C. Wang, X. Li, Q. She, L. Zhu, T. Zhang and Q. Chen, "Involution: Inverting the Inherence of Convolution for Visual Recognition," in *Computer Vision and Pattern Recognition (CVPR)*, Nashville, 2021.

[190] Q. Wang and R. Zhang, "Double JPEG compression forensics based on a convolutional neural network," *EURASIP Journal on Information Security,* vol. 23, 2016.

[191] Y. Zhang, G. Zhu, L. Wu, S. Kwong, H. Zhang and Y. Zhou, "Multi-Task SE-Network for Image Splicing Localization," *IEEE Transactions on Circuits and Systems for Video Technology ,* vol. 32, no. 7, pp. 4828 - 4840, 2022.

[192] Y. Sun, R. Ni and Y. Zhao, "ET: Edge-Enhanced Transformer for Image Splicing Detection," *IEEE Signal Processing Letters,* vol. 29, pp. 1232 - 1236, 2022.

[193] C. Yan, S. Li and H. Li, "TransU2-Net: A Hybrid Transformer Architecture for Image Splicing Forgery Detection," *IEEE Access,* vol. 11, pp. 33313 - 33323, 2023.

[194] 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, 2019.

[195] B. Chen, X. Qi, Y. Wang, Y. Zheng, H. J. Shim and Y.-Q. Shi, "An Improved Splicing Localization Method by Fully Convolutional Networks," *IEEE Access,* vol. 6, pp. 69472 - 69480, 2018.

[196] B. Liu and C.-M. Pun, "Locating splicing forgery by fully convolutional networks and conditional random field," *Signal Processing: Image Communication,* vol. 66, pp. 103-112, 2018.

[197] 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.

[198] Y. Wu, Y. Wo and G. Han, "Joint manipulation trace attention network and adaptive fusion mechanism for image splicing forgery localization," *Multimedia Tools and Applications,* vol. 81, p. 38757–38780, 2022.

[199] X. Chen, C. Dong, J. Ji, J. Cao and X. Li, "Image Manipulation Detection by Multi-View Multi-Scale Supervision," in *IEEE/CVF International Conference on Computer Vision*, Montreal, 2021.

[200] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi and A. Agrawal, "Context Encoding for Semantic Segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 2018.

[201] D. Xu, X. Shen, Y. Huang and Z. Shi, "RB-Net: integrating region and boundary features for image manipulation localization," *Multimedia Systems,* 2022.

[202] H. Chen, . Q. Han, Q. Li and X. Tong , "Digital image manipulation detection with weak feature stream," *The Visual Computer,* vol. 38, p. 2675–2689, 2022.

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

[204] X. Wei, Y. Wu, F. Dong, J. Zhang and S. Sun, "Developing an Image Manipulation Detection Algorithm Based on Edge Detection and Faster R-CNN," *Symmetry,* vol. 11, 2019.

[205] Y. Chen, X. Kang, Y. Q. Shi and Z. J. Wang, "A multi-purpose image forensic method using densely connected convolutional neural networks," *Journal of Real-Time Image Processing,* vol. 16, pp. 725-740, 2019.

[206] J. Park, D. Cho, W. Ahn and H.-K. Lee, "Double JPEG Detection in Mixed JPEG Quality Factors using Deep Convolutional Neural Network," in *European Conference on Computer Vision (ECCV)*, Munich, 2018.

[207] M. Chen, J. Fridrich, M. Goljan and J. Lukas, "Determining Image Origin and Integrity Using Sensor Noise," *IEEE Transactions on Information Forensics and Security,* vol. 3, no. 1, pp. 74-90, 2008.

[208] T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 42, no. 2, pp. 318 - 327, 2020.

[209] Y. Huang, S. Bian, H. Li, C. Wang and K. Li, "DS-UNet: A dual streams UNet for refined image forgery localization," *Information Sciences,* vol. 610, pp. 73-89, 2022.

[210] T. Nazir, M. Nawaz, M. Masood and A. Javed, "Copy move forgery detection and segmentation using improved mask region-based convolution network (RCNN)," *Applied Soft Computing,* vol. 131, 2022.

[211] Q. Yin, J. Wang, W. Lu and X. Luo, "Contrastive Learning based Multi-task Network for Image Manipulation Detection," *Signal Processing,* vol. 201, 2022.

Author Biography

Ankit Yadav received Bachelor of Computer Applications degree in 2013 and Master of Computer Applications degree in 2016 from Guru Gobind Singh Indraprastha University, Delhi, India. He is a senior research scholar (UGC-SRF) at the Department of Information Technology, Delhi Technological University, Delhi. The topic of his doctoral dissertation is Design and Development of a Framework to Detect Malicious Manipulations in Multimedia Data. His research interests include deep-learning based computer vision problems such as deepfake detection, image forgery detection, etc.