DEVELOPMENT OF FRAMEWORK FOR IMAGE MANIPULATION DETECTION

A Thesis Submitted
In Fulfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

by

RAHUL THAKUR
(Roll No. 2K18/PhD/EC/01)

Under the Supervision of
Prof. RAJESH ROHILLA
Department of Electronics & Communication Engineering
Delhi Technological University

Department of Electronics & Communication Engineering

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

March, 2025

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE'S DECLARATION

I, **Rahul Thakur,** hereby certify that the research work being presented in the thesis entitled "**Development of Framework for Image Manipulation Detection**" in fulfillment of the requirements for the award of the Degree of Doctor of Philosophy, submitted in the Departmentof **Electronics and Communication Engineering, Delhi Technological University** is an authenticated record of my own work carried out during the period from **August 2018 to March 2025**under the supervision of **Prof. Rajesh Rohilla.**

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

Rahul 19/03/2025

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Rajesh 19.3.2025

Signature of the Supervisor(s)

Signature of External Examiner

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE

Certified that **Rahul Thakur (2K18/PhD/EC/01)** has carried out their search work presented in this thesis entitled **"Development of Framework for Image Manipulation Detection"** for the award of **Doctor of Philosophy** from Department of Electronics and Communication Engineering, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Rajesh

Prof. Rajesh Rohilla
Professor,
Department of ECE,
Delhi Technological
University

Date: 19-03-2025

Abstract

In an era where digital images are extensively disseminated and manipulated, the authenticity of the visual content has become progressively vulnerable to manipulation. Digital images are now a crucial source of information in social media, thanks to advancements in technology and the Internet. Modern futuristic image editing software and tools make it simple to tweak a digital image without leaving any visible clues. The widespread use of digital images in news and legal proceedings has raised worries about their validity, integrity, and reliability. Manipulated or tampered photos can mislead the public, harm a person's reputation or business, influence political opinions, or impact criminal investigations. Conventional image manipulation techniques include copy-move, splicing and inpainting, whereas recent developments in image manipulation include synthetically generated images such as deepfake. Passive image manipulation detection and localization of manipulated regions within an image remains challenging. The thesis is structured into comprehensive chapters, beginning with foundational aspects, moving through specific and multiple manipulation detection methodologies and culminating in a robust solution for recent advancements in manipulations such as deepfake detection.

The thesis laid the groundwork by introducing the fundamentals of image manipulation detection, including image manipulation categorization, basic terminologies, application of image manipulation, the challenges of image manipulation detection and the classification of forgery detection techniques. This foundational knowledge provided context for understanding the scope and complexity of the problem. Furthermore, motivation and problem statement, performance metrics and thesis organization are discussed.

The thesis comprehensively reviews existing state-of-the-art (SOTA) methods employed for image manipulation detection. Various methods are reviewed, including traditional handcrafted, machine learning and deep learning-based methods for image manipulation detection. This review also examines the

limitations of the existing techniques and identifies the research gaps, leading to the formulation of research objectives.

The thesis provides a targeted approach for specific types of manipulation detection, such as offline signature forgery detection (OfSFD) and copy-move forgery detection (CMFD). The thesis developed a robust and efficient method for writer-independent offline signature forgery detection (WIOfSFD). The technique presents a formulation that uses the pre-trained model to direct the feature learning process and uses the Siamese neural network (SNN) to distinguish between genuine and forged signatures. Also, a residual-based convolutional neural network has been developed for CMFD.

The thesis introduces two methodologies, namely MDLFormer and LFRViT, for detecting multiple forgeries using a single framework. MDLFormer used multi-modal data to exploit various inconsistencies present in a manipulated image, global context-based swin transformer (GCST) encoder to enhance the model's ability to aggregate, refine, and focus on critical global discrepancies between various patches and feature pyramid network (FPN) based decoder for manipulation detection and localization. In contrast, LFRViT uses a Laplacian filter residual (LFR) based vision transformer (ViT) for multiple forgery detection.

The thesis also presented a hybrid learning-based approach consisting of kernel principal component analysis (KPCA) for deepfake face manipulation detection. The method uses the EfficientNetV2-L model for the feature extraction topped up with KPCA for feature dimensionality reduction to have an effective and fast feature learning process. The method is robust to various facial manipulation techniques such as identity swap, expression swap, attribute-based manipulation, and entirely synthesized faces. Experimental results validate the method’s effectiveness and demonstrate its potential as a reliable tool for detecting synthetic manipulations, which are becoming more common in digital forensics. Finally, this thesis work is concluded and the future scope of image manipulation detection is discussed.

List of Publications

Journals

Published

1. R. Thakur and R. Rohilla, “Recent Advances in Digital Image Manipulation Detection Techniques: A brief Review,” Forensic Science International, vol. 312, p. 110311, May 2020, doi: https://doi.org/10.1016/j.forsciint.2020.110311. ***(SCI Indexed)***

2. R. Thakur and R. Rohilla, “An effective framework based on hybrid learning and kernel principal component analysis for face manipulation detection,” Signal, Image and Video Processing, Apr. 2024, doi: https://doi.org/10.1007/s11760-024-03117-0. ***(SCI Indexed)***

Under Review

3. R. Thakur and R. Rohilla, “eSNN: EfficientNet-based Siamese Neural Network for Offline Signature Verification and Forgery Detection” Multimedia Tools and Applications, Springer.

4. R. Thakur and R. Rohilla, “MDLFormer: Multi-modal Global Context-based Swin Transformer for Image Manipulation Detection and Localization” Signal, Image and Video Processing, Springer.

Conferences

5. R. Thakur and R. Rohilla, "Copy-Move Forgery Detection using Residuals and Convolutional Neural Network Framework: A Novel Approach," 2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC), Greater Noida, India, 2019, pp. 561-564, doi: 10.1109/PEEIC47157.2019.8976868.

6. R. Thakur and R. Rohilla, "LFRViT: Laplacian Filter Residual-based Vision Transformer for Multiple Image Forgery Detection," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-5, doi: 10.1109/ICCCNT61001.2024.10724100.

Acknowledgment

I am deeply grateful to Lord Shiva, whose guidance and power have allowed me to pursue this Ph.D. I thank the almighty for granting me the wisdom, health, and strength to undertake and complete this research. I pray for his continued blessings so I may always serve him in all he wills.

First and foremost, I would like to express my sincere gratitude to my supervisor, **Prof. Rajesh Rohilla**, a professor at the Department of Electronics and Communication Engineering, for his continuous support, inspiration, patience, and guidance throughout my research. Their insightful feedback and encouragement were invaluable in completing this thesis.

My sincere regards *to* ***Prof. Prateek Sharma, Vice-Chancellor, Delhi Technological University,*** for providing me with a platform for pursuing my PhD work. I express my gratitude to **Prof. Rajeshwari Pandey**, *DRC Chairman*, *Department of ECE*, and **Prof. O. P. Verma**, *Head, Department of ECE*, for their kind support and for providing the necessary facilities to undertake this research. Also, I am thankful to all my colleagues, lab mates and the non-teaching staff members of the ECE department for their cooperation.

I owe thanks to my mother, **Mrs. Anuradha Thakur**; my father, **Mr. Rashpal Thakur**; my beloved wife, **Mrs. Nancy Jaswal**; my brother, **Mr. Rohit Thakur** and all my friends who inspired me for their continued and unfailing love, support and understanding during my pursuit of Ph.D. degree that made the completion of thesis possible. I want to express my sincere regards to my late grandfather, who always inspired me to continue. I thank the Almighty for giving me the strength and patience to work through all these years so that I can stand proudly with my head held high.

Thank you all for making this journey possible.

Rahul Thakur

Table of Content

List of Figure

List of Tables

List of Symbols and Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
SNNs	Siamese Neural Networks
GANs	Generative Adversarial Networks
ViT	Vision Transformers
A	Accuracy
FAR	False Acceptance Rate
FRR	False Rejection Rate
EER	Equal Error Rate
P	Precision
R	Recall
ROC	Receiver Operating Characteristics
AUC	Area Under an ROC Curve
IoU	Intersection over Union
TPR	True Positive Rate
GT	Ground Truth
B	Predicted Binary Mask
TP	True Positive
FN	False Negative
FP	False Positive
CMFD	Copy-move Forgery Detection
MDLFormer	Multi-modal Global Context-based Swin Transformer
LFRViT	Laplacian Filter Residual-based Vision Transformer
SOTA	State-of-the-art
GCST	Global Context Swin Transformer
FPN	Feature Pyramid Network
KPCA	Kernel Principal Component Analysis
OfSFD	Offline Signature Forgery Detection
LBP	Local Binary Pattern
DCT	Discrete Cosine Transforms

DWT	Discrete Wavelet Transforms
SIFT	Scale-invariant Feature Transform
SURF	Speeded Up Robust Features
CNN	Convolutional Neural Networks
SVM	Support Vector Machine
KNN	K-nearest Neighbor
HMM	Hidden Markov Model
ORB	Oriented FAST and Rotated BRIEF
PCA	Principal Component Analysis
SRM	Spatial Rich Model
CKN	Convolutional Kernel Network
SFCN	Single-Task Fully Convolutional Network
MFCN	Multi-Task Fully Convolutional Network
FCN-CRF	Fully Convolutional Network and Conditional Random Field
AWGN	Additive White Gaussian Noise
FFT	Fast Fourier Transform
LSTM	Long Short-Term Memory
IML	Image Manipulation Localization
YOLO	You Only Look Once
LBPH	Linear Binary Pattern Histogram
DFFD	Diverse Fake Face Dataset
FF-LBPH DBN	Fisher Face LBPH using Deep Belief Network
OnSFD	Online Signature Forgery Detection
WD	Writer-Dependent
WI	Writer-Independent
WIOfSFD	Writer-Independent Offline Signature Forgery Detection
ReLU	Rectified Linear Units
FLOPs	Floating Point Operations per Second
SD-MFR	Second Difference Median Filter Residual
LFR	Laplacian Filter Residual
GCB	Global Context Block

IMDL	Image Manipulation Detection And Localization
$\otimes$	Matrix Multiplication
$\oplus$	Element-Wise Addition
LN	Layer Normalization

Chapter 1
Introduction

This chapter introduces the background of image manipulation detection, image manipulation categorization and image forgery detection techniques. Applications of image manipulation and evaluation metrics used are also discussed. Furthermore, motivation and contribution of this thesis are elaborated upon and thesis organization is outlined.

1.1 Background

In an era where digital images are extensively disseminated and manipulated, the authenticity of the visual content has become progressively vulnerable to manipulation. Digital images are now a crucial source of information in social media. A large number of images are being produced and with the ease of availability of computer software or mobile applications, one can easily manipulate an image. Image manipulation has become very convenient nowadays with the help of editing tools, such as Adobe Photoshop, image manipulation programs, Affinity Photo, Paintshop and many more [1]. Using image manipulation techniques can be both useful and harmful as well. Image manipulation techniques can glamorize an image using image filters. Image manipulation is also useful for commercial purposes, as it uses realistic effects in movies like Harry Potter, Twilight, and many more, allowing them to share their creative ideas. On the other hand, these techniques can be utilized to control the substance of the picture with a malignant goal. Given the ease and effectiveness of the image editing tools, it is extremely hard to distinguish a manipulated image. With the advancement of image manipulation techniques and post-processing methods, it is very difficult for the forensic detector to detect the type of manipulation and manipulated region [1]. A study also shows that humans have a very restricted capacity to distinguish between the original and manipulated image [2]. These manipulated images are shared and uploaded on social media to provoke people's sentiments. These

altered photographs can serve as evidence in criminal investigations and tarnish an individual's reputation. For example, as shown in *Fig. 1.1*, the famous fake photo of Senator Tydings talking with Earl Browder (left) is a composite of two distinctive photographs [3]. It is believed that this fake photo may have contributed to Senator Tydings's electoral defeat in 1950. Therefore, it necessitates robust image manipulation detection mechanisms.

(a) Manipulated image *(b) Original image*

Fig. 1.1 A well-known image manipulation example, the composite photo of Senator Millard Tyding and American Communist Party Leader Earl Browder (left) [3].

Image manipulation encompasses a variety of techniques used to manipulate images, ranging from traditional techniques like splicing, copy-move and inpainting/removal etc., to recent advanced methods like face swap and deepfake etc. *Fig. 1.2* shows examples of image manipulation using different techniques, including traditional and deep learning-based methods. Deep learning-based approaches have revolutionized image manipulation, to generate highly realistic manipulated images. Artificial intelligence (AI) powered fake images look so real that they can easily fool humans and these counterfeit images are a bigger threat than fake news as they are more convincing than the text [4]. Various famous applications such as deepfake and face swap are based on convolutional neural networks, deep learning and adversarial networks which are employed to produce deep fake images. For example, *Fig. 1.2 (b)* illustrate an example of Faceswap, where Angela Merkel's face has been replaced with that of Donald Trump. and *Fig. 1.2 (c)* shows the generated fake image of the famous

Hollywood actor Nicholas Cage. With the advent of these fake images, the images have lost their credibility. This has caused fraud and fear of privacy in people [4].

(a) Image Manipulations using traditional methods

(b) Face swap

(c) Deepfake

Fig. 1.2 Examples of image manipulations from DEFACTO [135] dataset.

1.2 Image Manipulation Categorization

Image manipulation is commonly categorized based on its purpose and the underlying approach to steganography, forgery and generating, as illustrated in *Fig. 1.3* [5]. Image manipulation is the common term that consists of any form of altering, editing or modifying an image. Table 1.1 gives the fundamental definitions used in image manipulation. These terms are interrelated and differ based on how these terms and concepts are defined. Image steganography [6] does not come under the category of image forgery. It hides some data by somewhat altering the pixels in the image. Image forgery alters an image maliciously, including methods like copy-move, splicing and inpainting/removal to deceive the facts that happened in the past, requiring robust detection systems to identify tampered regions. However, image tampering falls within the realm of image forgery as it alters the image's content or context, such as recoloration, image enhancement, blurring and adding noise. Image-generating

techniques driven by AI models to generate entirely synthetic images can be used for forgery or not necessarily be used for forgery. Generated images often have no original source and are used for various purposes, including artistic creation, gaming, and malicious applications such as realistic deepfakes. Each type poses unique detection challenges, demanding specialized forensic approaches.

Table 1.1: Fundamental terminologies used in image manipulation

Terminology	Definition
Image Manipulation	It refers to any form of altering, editing or modifying an image.
Image Forgery	It refers to intentionally and deliberately modifying or creating an image to deceive or mislead.
Image Tampering	It is a subset of image forgery in which the graphic content of the image is modified. It refers to a specific act of altering images.
Image Generating	These are computer-generated images, or some part of the image is computer-generated, which can be used to forge an image.
Image Steganography	It is used to hide some data in the image. It slightly alters some pixels in the image and embeds extra data in the image.
Copy-move	In this technique, the content is copied and moved to the other position in the original image. The new content copied is from the same source as the original image or from the original image itself.
Splicing	Splicing is generally used as a substitute for cut-paste, in which a composite image is made by cutting and joining the multiple images. It denotes the region duplication between two images.
Inpainting	It is the process of restoring and reconstructing the lost or corrupted part of the image.
Deepfake	It is a type of synthetic media in which AI, particularly deep learning techniques, is used to generate realistic but fake images, audio or videos that accurately mimic real people or events.

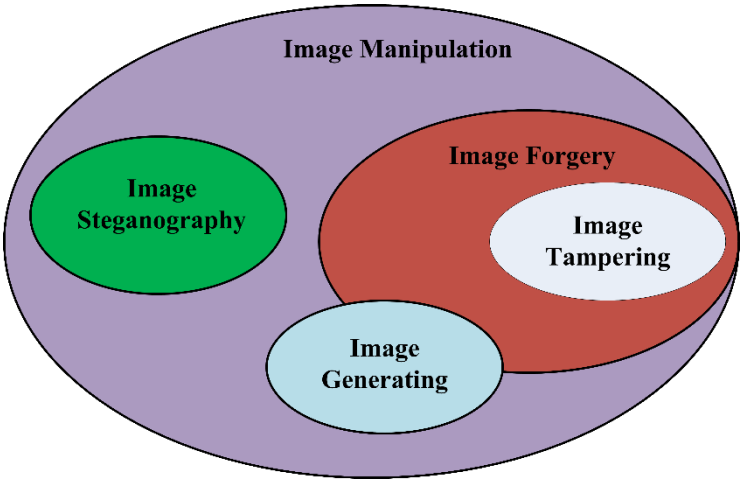


Fig. 1.3: Categorical representation of image manipulation.

1.3 Image Forgery Detection Techniques

Image forgery detection techniques can be classified, as shown in *Fig. 1.4*, into two categories: active and passive forgery methods [5]. Active methods include embedding additional data, such as watermarks or digital signatures, into the image during its creation. This embedded data helps detect Active methods consisting of digital signature and watermarking [7]. Active methods are utilized to identify the image's integrity and whether the image is authentic or tampered with. However, their effectiveness relies on the prior existence of such embedded information inside the image. On the contrary, passive methods, also known as blind methods, do not require any prior knowledge about the image. Passive methods analyze intrinsic irregularities and inconsistencies in the image or trace the artifacts left by the tampering operation to identify manipulation. Passive methods are categorized into intrinsic regularities & inconsistencies, tampering operations and natural & computer graphic images. Further, based on dependent or independent techniques, tampering operations are categorized into specific forgery detection techniques such as copy-move, splicing, JPEG compression, retouching and light inconsistencies. Active methods are highly reliable on embedded data or prior information when used, while passive methods are ubiquitous and versatile, making them indispensable in modern digital forensic

applications [8]. However, passive methods can be more challenging due to the increasing sophistication of image manipulation tools.

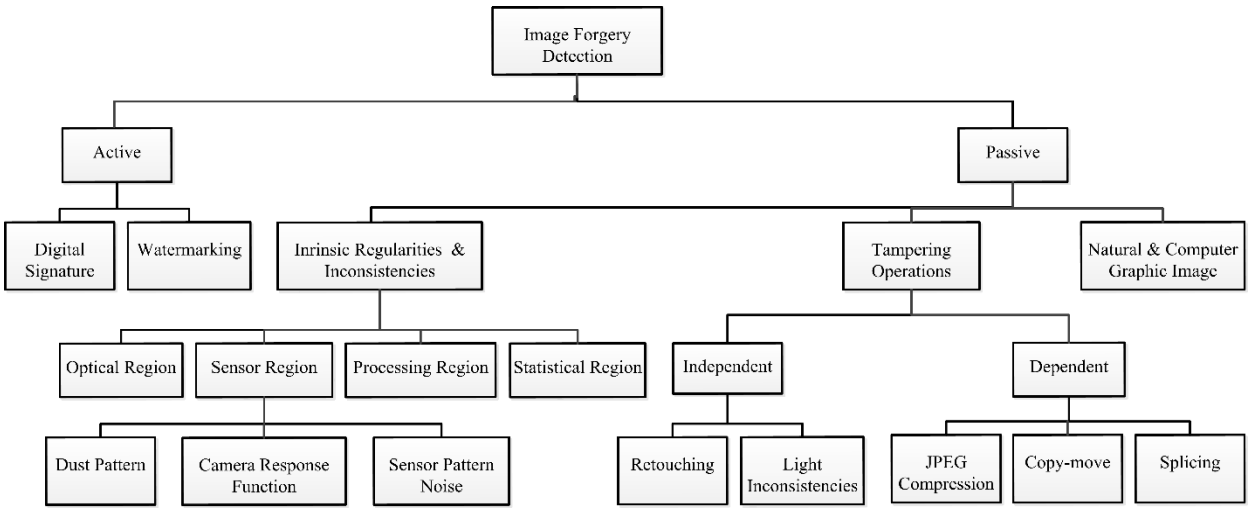


Fig. 1.4 Classification of image forgery detection techniques

1.4 Applications of Image Manipulation Detection

Image manipulation detection plays an essential role across multiple domains where the integrity and authenticity of visual content are crucial. As digital tampering methods become more sophisticated, detecting such manipulations has become a priority in law enforcement, media, finance, and even personal security applications. From signature verification to deepfake detection, the applications of image manipulation detection are broad and continue to grow with advances in deep learning and computer vision techniques. Some prominent areas include offline signature verification, copy-move forgery detection (CMFD), splicing, inpainting, and deepfake detection, each addressing unique forms of image tampering.

Offline Signature Forgery Detection is essential in financial, legal, and governmental settings, where authentic signatures are required for identity verification [9]. Manipulation detection frameworks analyze signature attributes such as stroke pattern, pressure, and spatial alignment to differentiate genuine signatures from forgeries. Techniques based on deep learning architectures like Siamese Neural Networks (SNN) provide high accuracy in detecting forgery in offline signatures, thus enhancing fraud prevention and authentication.

Copy-Move Forgery Detection: Copy-move forgery involves duplicating a portion of an image and moving it to another location within the same image, often to hide or replicate specific content [10]. This type of forgery is widely used in digital tampering, and detecting it requires robust analysis of duplicated patterns and slight inconsistencies in texture and lighting. CNN-based frameworks are particularly effective in identifying copy-move forgeries, leveraging residual features and spatial correlations to pinpoint duplicated regions accurately.

Splicing Detection: Splicing is a manipulation technique that combines elements from multiple images to create a single composite image, often with the intent to mislead. Splicing detection aims to identify inconsistencies in texture, lighting, and boundaries between the combined elements [10]. Advanced detection methods use deep learning algorithms to analyze subtle differences at the pixel and boundary levels, effectively identifying where one image has been fused with another.

Inpainting Detection: Inpainting refers to the process of filling in missing or undesired regions within an image, typically used to remove objects or alter backgrounds [11]. In forensics, inpainting detection is crucial for identifying areas that have been artificially reconstructed. Detection frameworks leverage models trained to recognize unnatural textures and pixel arrangements that indicate inpainting, ensuring that image integrity is preserved in contexts where accuracy is critical.

Deepfake Detection has become increasingly important due to the rise in realistic yet synthetic images and videos created using generative adversarial networks (GANs) [12]. These forgeries can be used to impersonate individuals, spread misinformation, or manipulate opinions. Deepfake detection algorithms focus on recognizing inconsistencies in facial expressions, eye movements, and other subtle features that indicate tampering. Techniques utilizing Vision Transformers (ViT) [13] are particularly effective in detecting these artificial creations by focusing on both spatial and temporal artifacts. Deepfake detection focuses on identifying and mitigating the risks associated with manipulated media, particularly videos and images where individuals' faces or voices are convincingly altered using AI. This is a critical area of

research due to the potential misuse of deepfake technology in fraud, misinformation, and other malicious activities.

1.5 Multiple Manipulation Detection

Nowadays, detecting manipulated images is challenging, as the complexity of manipulations can vary, ranging from single manipulation to more sophisticated multiple manipulations. Image manipulation detection can be broadly classified into two categories: single manipulation detection and multiple manipulation detection [11]. A forger can manipulate an image by employing a variety of image manipulation techniques [14]. There has been a significant interest in developing a universal/multiple image forgery detection approach to detect multiple manipulation operations [15], [16]. Single manipulation detection focuses on identifying and analyzing a particular type of manipulation. These techniques are designed to target a particular manipulation such as OfSFD, copy-move, splicing, inpainting or enhancement. Single manipulation detection methods often use specialized feature extraction techniques or deep learning models to achieve good performance for that specific task. Despite their effectiveness, single manipulation detection methods struggle to generalize across multiple manipulation types or recognize the combination of manipulations in an image. Contrarily, multiple manipulation detection techniques seek to detect multiple manipulations even when many manipulations coexist. Multiple manipulation detection goal is to apply a unified approach to identify and distinguish various types of forgeries present in a manipulated image. Multiple manipulation detection systems require more complex models capable of handling various manipulation operations and robust feature representations that generalize across various manipulations. Multiple manipulation detection methods are more challenging to build yet essential for practical forensics applications. Many researchers have focused on detecting manipulated images by employing machine and deep learning techniques. Some of the prominent forms of manipulations covered in this thesis are digital OfSFD, copy-move, splicing, inpainting/removal and deepfake.

1.6 Evaluation Metrics

Evaluation metrics are quantitative measures employed to assess the efficacy and performance of the proposed model. This section discusses several assessment criteria employed for assessing the performance of the proposed image manipulation detection methods in the subsequent chapters. Evaluation parameters indicate how well a system operates in the presence of a test dataset. Because of the imbalanced nature of the dataset, some of the few parameters likely produce good results while others do not, i.e., accuracy. To address the issue of imbalanced datasets, Precision, Recall, and F1 measures should be used to provide the true performance of the evaluated algorithms [17]. The approaches' performance is evaluated using two types of evaluation parameters: image-based and pixel-based. Pixel-based evaluating parameters rely on ground truth images in the dataset for evaluation and are considered practical and accurate. The following metrics are frequently used to assess different manipulation detection methods: Accuracy (A), False Acceptance Rate (FAR), False Rejection Rate (FRR), Equal Error Rate (EER), Precision (P), Recall (R), F1 score, Receiver Operating Characteristics (ROC), Area Under an ROC curve (AUC), Intersection over Union (IoU), etc. Depending on the requirement, these metrics can be applied to evaluation at the pixel or image level. The definitions of the metrics used in this dissertation are provided below.

1. ***Accuracy (A)***: It measures the level of accurately identifying the manipulated and authentic images and is calculated using Eqn. 1.1.

$$A = \frac{TP+TN}{TP+TN+FP+FN} \tag{1.1}$$

2. ***False Acceptance Rate (FAR)***: It measures the likelihood that a method incorrectly identifies an authentic image (negative class) as manipulated (positive class) and is calculated using Eqn. 1.2.

$$FAR = \frac{FP}{FP+TN} \tag{1.2}$$

3. ***False Rejection Rate (FRR)***: It measures how well a method incorrectly rejects a manipulated image (positive class) as an authentic image (negative class) and is calculated using Eqn. 1.3.

$$FRR = \frac{FN}{FN+TP} \tag{1.3}$$

4. ***Equal Error Rate (EER):*** An error value where FAR equals the FRR. To calculate the EER, one can plot the FAR and FRR on a Receiver Operating Characteristic (ROC) curve and identify the point where the two curves intersect. If there is not a number where FAR and FRR are equal, choose a number that falls between the two as given in Eqn. 1.4

$$ERR = \frac{FAR+FRR}{2}, if\ FAR \neq FRR \tag{1.4}$$

5. ***Precision (P):*** It measures how well a method correctly identifies a manipulated image (positive class) among all positive predictions. Precision is computed by dividing the number of correct positive predictions (true positives) by the total number of positive predictions (sum of true positives and false positives) and is calculated using Eqn. 1.5.

$$P = \frac{TP}{TP+FP} \tag{1.5}$$

6. ***Recall (R) or True Positive Rate (TPR) or Sensitivity:*** It measures how well a method correctly identifies all manipulated images (positive class). It is computed by dividing the number of correct positive predictions (true positives) by the total number of real positive cases (sum of true positives and false negatives) and is calculated using Eqn. 1.6.

$$R = \frac{TP}{TP+FN} \tag{1.6}$$

7. ***F1 score:*** It measures Precision (P) and Recall (R) harmonic mean and is calculated using Eqn. 1.7.

$$F1 = \frac{2\times P\times R}{P+R} = \frac{2\times TP}{2\times TP+FN+FP} \tag{1.7}$$

8. ***Intersection over Union (IoU):*** It calculates the amount of overlap between the predicted binary mask (B). and the ground truth mask (GT). It is a metric used to localize an image's manipulated region. It is calculated by dividing the area of overlap between B and GT region by the combined area of both regions, and it is calculated using Eqn. 1.8.

$$IoU = \frac{GT \cap B}{GT \cup B} = \frac{TP}{Tp+FN+FP} \tag{1.8}$$

Where True Positive (TP) refers to the manipulated class accurately identified as manipulated, while False Negative (FN) represents the manipulated class incorrectly classified as authentic. False Positive (FP) indicates the authentic class is misclassified as manipulated.

1.7 Motivation

Image manipulation is one of the most preliminary and prevalent modification attacks on digital image forensics. Digital images have become increasingly vulnerable to manipulation with the advancement of image editing tools and digital media creation. Consequently, many image manipulation tools and software have been developed that can be further used for malicious activities like mob agitation and fake news spreads. Manipulation techniques like copy-move forgery, splicing, inpainting, signature forgery, and deepfake generation are now more accessible than ever, allowing the manipulation of images with high fidelity and subtlety. While these manipulations allow for creative applications, they threaten information integrity and authenticity. The widespread use of manipulated content has increased disinformation, identity theft and weakening trust in digital media. As a result, the need for reliable and effective image manipulation detection methods has become crucial to finding traces of manipulation in images and hence, successfully classifying them as authentic or manipulated. This thesis explores, evaluates, and develops advanced methods for detecting image manipulations, contributing to digital forensics, security and media authenticity verification.

1.8 Contribution

The thesis presents novel methodologies and frameworks to improve the detection and localization of various image manipulations, addressing existing challenges in digital forensics, security applications and content verification. The study contributes to various key areas of image manipulation detection, including offline signature verification, CMFD, multiple forgery detection and localization, and deepfake face manipulation detection. Various machine learning and deep learning methods, including diverse features, have been used. The datasets that are made publicly available are included and their characteristics and parameter settings are tabled. In this thesis, the key contribution includes a writer-independent offline signature verification model based on a pre-trained EfficientNet model used for feature extraction in the twin network of SNN, a residual-based CNN model for CMFD, a Multi-modal Global Context-based Swin Transformer (MDLFormer) model for multiple forgery detection and localization. Furthermore, the thesis incorporates a Laplacian Filter Residual-based Vision Transformer (LFRViT) framework for multiple forgery detection, leveraging ViT architecture and Laplacian filters to capture subtle tampering artifacts indicative of tampering. Lastly, an effective framework is proposed based on hybrid learning for deepfake face manipulation detection. These models collectively enhance image manipulation detection by providing adaptable, robust solutions across various manipulation types. Consequently, they support critical media integrity, digital forensics, and authentication applications. Various standard datasets have been utilized to validate the efficacy of the model. A comprehensive description of the proposed approaches has been addressed in the subsequent section.

1.9 Thesis Organization

This thesis is organized into six chapters. The brief outlines are given below:

Chapter 1: This chapter provides the fundamentals concerning image manipulation detection. This involves image manipulation categorization, basic terminologies, categorization of techniques for image forgery detection and application of image

manipulation detection. This chapter also discussed the evaluation metrics, motivation, contribution of this thesis and thesis organization.

Chapter 2: This chapter explains the challenges in the existing SOTA methods employed for image manipulation detection. The standard overall design framework of the image manipulation detection system based on the traditional handcrafted feature extraction-based approach is discussed. A review of several methods used for image manipulation detection is done. This helped to discover the research gaps in existing solutions in image manipulation detection. Finally, the research objective has been formulated based on the research gaps addressed in this thesis.

Chapter 3: This chapter explains the proposed methodologies used for detecting specific manipulations such as offline signature forgery and CMFD. A detailed description of the problem statement, dataset, feature extraction process, and the methodology adopted has been provided in this chapter. Experiments on standard datasets validates the effectiveness of the proposed method and a comparison study of the results is also provided.

Chapter 4: This chapter incorporates two different methods to detect multiple manipulations. The first method, MDLFormer, consists of multi-modal input, Global Context Swin Transformer (GCST) encoder and Feature Pyramid Network (FPN)-based decoder to detect and localize the manipulation. The second method, LFRViT, is a Laplacian filter residual-based vision transformer for multiple manipulation detection. In the thesis, the methodologies concerning each of the given methods have been discussed in detail. Additionally, the results of the proposed methods are obtained on standard datasets and compared with existing SOTA methods.

Chapter 5: This chapter presents a novel approach based on hybrid learning and kernel principal component analysis (KPCA) for deepfake face manipulation detection. The result and discussion section explains the effectiveness of the proposed approach on standard datasets and comparative analysis of obtained results is also included.

Chapter 6: This chapter summarizes proposed works, significant findings, contributions and limitations. This chapter also suggests some potential future directions in this area and social impact on society beyond academic circles.

Chapter 2

Literature Review

This chapter comprehensively reviews existing methods employed for image manipulation detection. Reviewed various methods, including traditional handcrafted, machine learning models and deep learning-based approaches for image manipulation detection. This review also examines the limitations of the existing methods and identifies the research gaps in image manipulation detection. Finally, the research objective has been formulated based on the research gaps addressed in this thesis.

This chapter reviews various methods, focusing on techniques for digital OfSFD, CMFD, multiple forgery detection and localization, and deepfake face manipulation detection. In the past, methods for detecting image manipulation were limited to a single kind of manipulation [18]. The image was manipulated using single manipulation technique and the type of manipulation was then identified by analyzing the distinct trace that was left behind. Simple feature extraction techniques are utilized, followed by classification to identify certain kinds of image manipulation.[19].

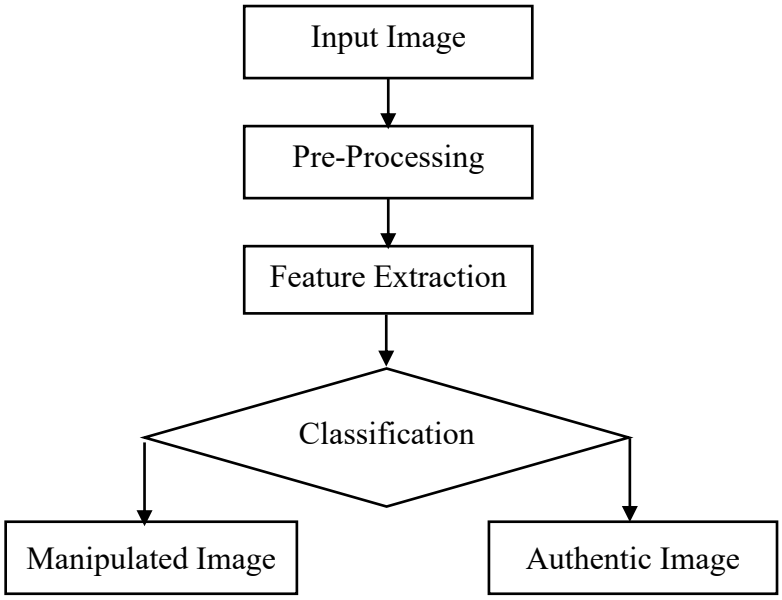


Fig. 2.1: The general structure of the image manipulation detection system based on the handcrafted feature extraction method.

Fig. 2.1 shows the image manipulation detection system's general structure based on handcrafted feature extraction methods.

For any research problem, five aspects must be explored: datasets, pre-processing tasks, feature extraction techniques, methodology/models and performance evaluation criteria. Several pre-processing techniques were used, such as binarization, normalization, thinning, bounding box, inversion, and noise removal techniques, to remove the inconsistencies present in the image and enhance the image's quality for further processing [18]. A few of the feature extractors that are used for manipulation detection are local binary pattern (LBP) [20], discrete cosine transforms (DCT) [21], discrete wavelet transforms (DWT) [22], scale-invariant feature transform (SIFT) [23], speeded up robust features (SURF) [24]. These techniques accomplish pre-processing following feature extraction. The images are then classified using some thresholding criteria on the extracted features or matching technique is used to classify the images. But now, images are manipulated using multiple tampering operations to make them realistic, so they cannot be viewed as manipulated images. With the advancement in editing tools, detecting the manipulation and the manipulated region in the image is not easy. Later, some new methods were created to identify multiple-image manipulations in images, but these methods were limited to some constraints and could not detect multiple-image manipulations in images [25]. In a real-world scenario, an image is manipulated using multiple image manipulation techniques. Consequently, there is a requirement for multiple image manipulation detection techniques to authenticate an image as an authentic or a manipulated image. Nowadays, numerous research scholars have created techniques based on deep learning models to detect image manipulation in images and have produced better results and outperform the hand-crafted feature extraction techniques [26], [27], [28].

Deep learning models have been proven to be the best technique for feature learning and classification. In the past decade, deep learning methods have been used extensively in every field, and the use of these models has increased rapidly. Deep learning approaches are also being used in image manipulation detection as well. In deep learning-based image manipulation detection methods, many real and manipulated images are given to the models for manipulation detection. A good

training model is used to capture the underlying features of the images. This chapter will study various models used to detect manipulation in the image. Image manipulation detection techniques can be broadly represented into two categories: handcrafted feature representation and learned feature representation. The following taxonomy has been designed for image manipulation detection methods; see *Fig. 2.2*.

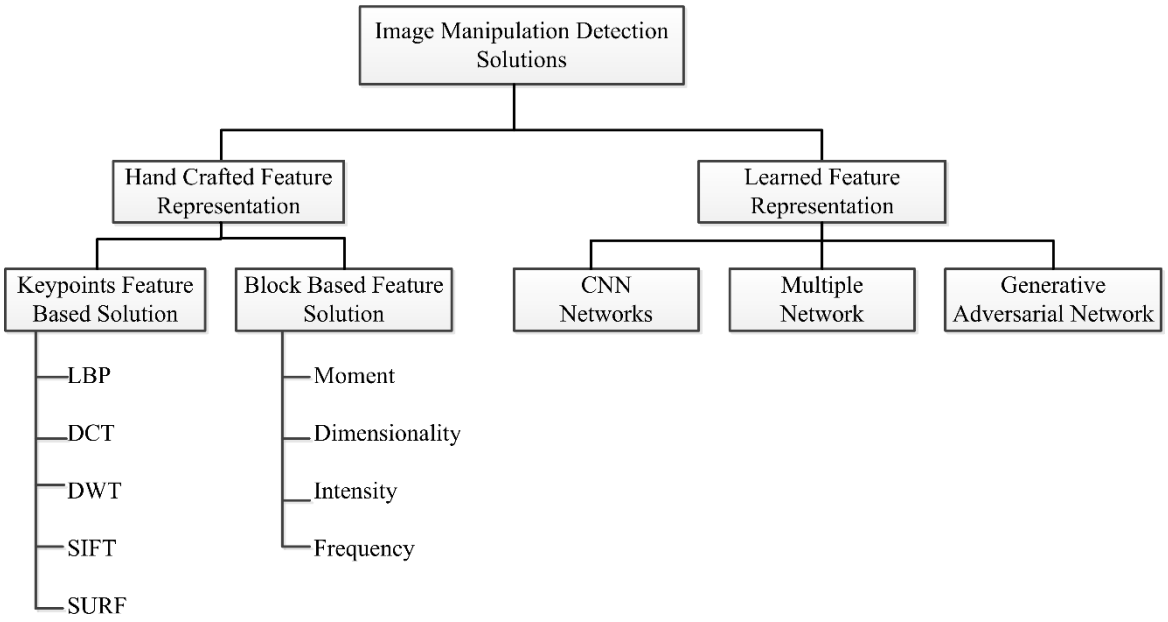


Fig. 2.2: Taxonomy of Image Manipulation Detection Methods.

2.1 Digital Offline Signature Forgery Detection

Signature forgery detection tasks have been proposed for a variety of hand-crafted features. Many consider global features using block codes, wavelets and Fourier series [29]. Other approaches consider local features such as location, tangent direction, curvature, blob structure and connected components with geometrical and topological properties [30]. For OfSFD, projection and contour-based approaches are also widely used in signature verification [31]. Additionally, a few structural methods that examine the relationships between local attributes are examined. Manual feature extraction techniques consist of structural, geometric, texture, statistical and global methods, whereas automatic feature extraction techniques consist of deep learning methods such as convolutional neural networks (CNN), autoencoders and deep sparse networks, etc., [32]. Unlike human feature extraction techniques, techniques for automatically

extracting features (such as deep learning) do not require domain knowledge to extract discriminative features for classification [33]. CNN and deep sparse approaches are two examples of deep learning-based methods widely used for automatic feature extraction. These techniques enable the automatic learning of features with minimal or no pre-processing [34]. These techniques can find the optimum pattern for enhancing verification performance by extracting intricate information from the raw signature image utilizing several abstraction layers. CNN has drawn much attention as a very effective feature extraction and identification approach since it is an automatic feature learning method [34]. It is significant to highlight that CNN is capable of self-taught, automatically extract characteristics and offer probabilistic predictions for each class label. Therefore, integrating automated feature extraction with prediction or classification for OfSFD detection provides an effective and reliable solution [35]. These deep learning-based methods can automatically extract complicated features with few or no pre-processing techniques and require no prior domain knowledge to extract discriminative features for classification [36]. This makes automatic feature extraction techniques more popular and well-known for automatic feature extraction.

Various methods or models have been adopted for OfSFD systems, which can be divided into models based on template matching, machine learning models, and deep learning models. In template matching methods, a template is matched with the offline signature image to find the maximum match pixels using several similarity measure techniques such as the Euclidean distance, graph edit distance, the cosine similarity measure and fuzzy similarity measure [37][38]. Machine learning-based models have significantly improved over traditional template-matching models in classifying genuine and forged signature images [39]. Researchers have embraced several machine learning-based models, including the support vector machine (SVM), the k-nearest neighbor (KNN), the decision tree, and the Gentle AdaBoost. The verification of signatures has utilized a wide range of machine-learning techniques [32]. For instance, Fang et al. [40] examined tracking characteristics and pen stroke locations for signature verification, but they also observed a FAR of 16.7%. A signature verification method was created by Alaei et al. [41] utilizing a fuzzy similarity measure and an interval symbolic representation of offline images of signatures. Using the SVM, the hidden Markov model (HMM) and Euclidean

classifiers, Ferrer et al. [42] analyzed the geometric aspects of signatures. With different degrees of success and often Average Error Rates exceeding 10%, several additional papers have investigated variants of features and classifiers [43][44][45]. Nowadays, deep learning-based methods are quite popular and are used in every field of research due to their ease of use, automatic learning capability, and generalization capability. Deep learning methods like CNNs have been used to detect forgeries by automatically learning features from offline digital signatures. Literature has also shown the use of several deep learning-based models in OfSFD systems. Various deep learning-based models like deep neural networks [45], shallow neural networks [33], CNNs [46], pre-trained neural networks and SNN are used for offline signature verification systems [47].

2.2 Copy-move Forgery Detection

CMFD is among the most prevalent problems in multimedia forensics. This type of forgery obscures or copies a few elements or portion of the image. The act of copying a portion of the image and inserting it within the same image is known as copy-move forgery. In contrast, splicing involves copying and pasting a portion of an image onto another image that is not same as the source image. Numerous CMFD-related work is primarily based on the two techniques: (i) Key- point-based feature matching [23][24] and (ii) Block based feature matching [48][49][50][51]. Key point-based methods extract and compare key features within the image. Various feature extractors are used, including SIFT, SURF and ORB (Oriented FAST and Rotated BRIEF) etc., because of their robustness to rotation and scaling. The block-based methods divide the image into overlapping regions, which are matched to detect similarities. Commonly utilized techniques include the DCT, Principal Component Analysis (PCA), and SIFT. Nonetheless, these approaches come with a significant computational cost, along with number of others inherent drawbacks. Therefore, several works incorporated adaptive over-segmentation [52][53] to divide the image into non-overlapping patches to reduce the computational complexity and perform feature matching to detect forgery. However, rather than feature-matching parts of images and detecting copy-move forgery, focus on detecting the traces of operations performed after copy-move and

splicing to blend it with the original image. In the literature, much work has been done to detect the traces left by image tampering post-processing operations like median filtering [54], re-compression [55] and contrast enhancement [56]. Such operations are employed to make the forgery look more convincing, with median filtering being the most widely used. Current techniques use deep learning-based methods to automatically learn complicated features from training data, resulting in higher accuracy and robustness than handcrafted-based methods. Deep learning-based approaches are now used in every field of research because of their automatic learning features capability and achieving high accuracy in classification. Various deep learning-based approaches were also used to detect an image's tempering and prove better results. Generally, in a deep learning model, images are directly given as input to the network layer, and the network automatically learns the features based on the image's content. However, in the case of image tempering detection, instead of learning the content-based features, the traces left after the tempering operation performed on the image are learned and used to classify the image as authentic or tempered. To learn the traces left after the tempering operation, preprocessing such as filtering is done and these filtering residuals are fed to the first convolutional layer. Yang et al. [57] used Laplacian filter before passing the image through CNN for edge sharpening and hence image enhancement thereby reducing the blurring effects. Deep learning-based techniques have recently been used for detecting splicing and/or copy-move forgery type of manipulation. Deep CNNs are particularly excellent at identifying copy-move forgeries because they recognize unique patterns that indicate manipulation. Hybrid frameworks combining traditional methods with deep learning techniques have produced promising results by employing handcrafted and deep-learned features to improve detection accuracy. Rao and Ni present a approach for detecting splicing and copy-move forgery [58]. Their approach involves a supervised CNN that learns the hierarchical features of the manipulated input RGB color image. Instead of initializing the weights randomly, as in conventional CNN, a high pass channel set is used to estimate any remaining mappings in the spatial rich model (SRM). To hide the image content and detect the subtle artifact caused by the tampering operations, the first layer uses kernel weights based on 30 high-pass filters. The 10 layers make up the CNN architecture that is used to automatically learn the features. The final discriminative

Table 2.1: Copy-move forgery detection methods.

Author (s)	Methodology	Details	Performance	Dataset
Liu *et al.*, [52]	CKN	A data-driven local descriptor, GPU-based adaptive over-segmentation, robust to post-processing, noise, brightness change, gaussian blurring and transformations	F1 = 0.5997	CoMoFoD
Salloum *et al.*, [59]	Edge-enhanced SFCN and MFCN	Multi-task learning, based on VGG-16, robust to noise, gaussian blurring and JPEG compression	F1 = 0.6117	Columbia
Cozzolino *et al.*, [60]	Constrained CNN, Residual feature extraction	Small training set, robust to median filtering, gaussian blurring, noise, resizing and JPEG compression	Accuracy = over 90%	Synthetic
Ouyang *et al.*, [61]	Transfer learning using ImageNet	Uses a pre-trained model that isn't realistically robust to copy-move forgery	Error = 2.32%	Oxford
Wu *et al.*, [61]	CNN feature extractor using VGG16 model	End-to-end Deep CNN solution, poor in a pure texture image	F1 = 0.7572	CASIA v2.0
Liu *et al.*, [63]	FCN-CRF	Pixel-to-pixel forgery detection, scale-invariant, and optimization error exist.	TPR = 82.6%	CASIA v2.0

features are obtained by combining the results of a pre-trained CNN's dense patch-based feature extraction from the test image with the feature fusion approach. Lastly, the SVM classifier is used to make binary classification (authentic/forged). In another method, Liu et al.[52] use the convolutional kernel network (CKN), a data-driven local descriptor. This technique uses adaptive GPU-based over-segmentation based on the convolutional-oriented boundaries (COB) method to produce multiscale-oriented contours and region hierarchies. In addition, the segmented picture is subjected to key point identification, CKN feature extraction, patch matching, and transform estimation computation. In [59], two approaches were employed for image splicing localization problems: a single-task fully convolutional network (SFCN) and a multi-task fully convolutional network (MFCN). A lot of other methods are used for copy-move and splicing forgery detection, like Cozzolino et al., [60] uses constrained CNN based on residual feature extraction, Ouyang et al., [61] uses transfer learning method, Wu et al., [62] uses CNN feature extractor using VGG16 model, Liu et al., [63] uses fully convolutional network and conditional random field (FCN-CRF) method for pixel to pixel-based forgery detection and Wu et al., [64] introduces an end-to-end deep neural network called BusterNet for CMFD and localization. Various copy-move and splicing manipulation detection techniques are mentioned in *Table 2.1*. *Table 2.1* consists of the authors, the methodology used, brief details about the method, performance parameters and the dataset on which the evaluation is done.

2.3 Multiple Forgery Detection

Multiple image manipulation detection methods are employed to detect the various tampering operations carried out on the image. Detecting manipulated images is challenging, as the complexity of manipulations can vary, ranging from single manipulation to more sophisticated multiple manipulations. A forger can manipulate an image by employing a variety of image manipulation techniques. A significant interest has been observed in developing a universal image forgery detection approach to detect multiple tampering operations. Universal image manipulation detection techniques are usually focused on identifying the traces that are left over after the post-processing operations. Many researchers have focused on detecting manipulated

images by employing machine and deep learning techniques. A variety of deep learning-based techniques are employed to identify multiple image manipulations. *Table 2.2* briefly details the various multiple image manipulation detection techniques. *Table 2.2* consists of the authors, the methodology used, brief details about the methods, performance parameters and the dataset on which the evaluation is done.

Bayar and Stamm [16] gave a deep learning-based universal image manipulation detection approach. The technique uses a novel convolutional layer that is distinct from the standard layers of a CNN. This technique uses a novel convolutional layer that automatically suppresses the image's content and records the traces left by the tampering operation, whereas previously pre-processing or preselected features were needed to detect image manipulation. While hiding the image's content, the modified attributes that were taken from the layer include the relationship between the pixel and its immediate vicinity. The constrained layer is the first layer, where the prediction error filters are learned using convolutional filters. After giving each filter, a weight at random, the constraint is applied to each filter and iteration. This completes the task of using the tampered image to learn the altered features. The following four tampering procedures were taken into consideration: resampling, AWGN (additive white Gaussian noise), median filtering, and Gaussian blurring. Both binary and multi-class classification were tested in the experiment. Two neurons make up the output layer of the binary classification method, which is used to categorize both original and altered images. Five neurons make up the problem output layer in multi-class classification, which is used to categorize various forms of image forgeries. The accuracy of the approach is high, at about 99.10%. Furthermore, a data-driven strategy was used by, Bayar and Stamm [65] in another paper to provide a manipulation parameter estimator. This method is independent of the individual study of the estimator for each form of manipulation. In [66], two techniques for identifying and locating image alteration are employed. The first approach classifies tampered images using a deep neural network and manually created features such as Fast Fourier Transform (FFT), Laplacian, and Radon. The second approach used a long short-term memory (LSTM) network to learn the boundary transformation or correlation between the current block of resampling characteristics and the neighbouring blocks. This gives the SoftMax classifier the discriminative features it needs to classify the data. The technique successfully classify

Table 2.2: Multiple image manipulation detection methods

Author (s)	Methodology	Details	Performance	Dataset
Zang *et al.*, [25]	Stack Autoencoder	Detect the tampered region accurately, applicable to both JPEG and TIFF formats, BMP image format is not included	Accuracy = 87.51%	CASIA
Bayar and Stamm [65]	CNN	Manipulation detection using data-driven parameter estimation, four different tampering operations are detected: JPEG compression, median filtering, gaussian blurring and resampling	Accuracy = 90% to 99%	Dresden based synthesized
Bunk *et al.*, [66]	CNN and LSTM	Detect and localize manipulation using resampling features and deep learning, involves JPEG quality, rescaling, rotation and shearing	Accuracy = 94.86%	NIST Nimble 2016
Bappy *et al.*, [67]	LSTM-EnDec	Manipulation localization is done using resampling features, LSTM cells and encoder-decoder network, low-resolution feature map, fit for restricting controls at a pixel level.	Accuracy = Over 71%	Synthesized using NIST'16, IEEE FC, COVERAGE
Mazumdar *et al.*, [15]	Deep siamese CNN	Instead of classification, the method discriminates based on the same or different processing operations they have gone through	Accuracy = 95.24%	Dresden based dataset

the manipulated images with an accuracy of 92.64%. Because the recompression of modified images leaves evidence behind, the JPEG image format is typically employed for forgery localization. However, in [25], the stack autoencoder was employed to

detect tampering actions in various image formats using contextual information and feature learning. Numerous other multiple image manipulations detection techniques were also employed, including Bappy et al. [67] where they used LSTM-EnDec and Mazumdar et al., [15], used Deep Siamese CNN.

2.4 Image Manipulation Localization

Image manipulation detection is ascertaining whether an image has been manipulated from its original state. On the other hand, image manipulation localization (IML) takes it a step further by identifying that an image has been manipulated and precisely determining the manipulated region within the image. Detection and localization are vital in diverse domains, including forensics, journalism, medical imaging and digital media authentication. Detection is useful for identifying potentially manipulated images, whereas localization provides additional information about the scope and characteristics of the manipulation, allowing for informed decisions regarding the image's authenticity and integrity. For many years, the field of media forensics has been established to identify fraudulent activities. Early research focuses on projecting images straight into binary label space (authentic/manipulated) using conventional features [8]. Localizing multiple image manipulations at the pixel level is difficult because of the tampered region's features, which include various scales, uneven shapes, hazy boundaries, and strong intrinsic resemblance to chaotic backdrop objects. Conventional IML techniques rely on hand-crafted features, such as self-consistency, point matching, and Markov features, which have a weak generalization capacity and strictly rely on the domain expertise of human experts [68]. Deep learning-based IML techniques may automatically extract discriminative features using deep neural networks and have a more significant learning ability for complicated scenarios [69].

The IML task, which aims to uncover and magnify the forgery traces concealed in the altered image, merely needs segmenting out the fabricated region instead of semantic segmentation. Progressive deep learning-based IML techniques may automatically extract discriminative features using deep neural networks and have a more significant learning ability for complicated scenarios than standard techniques

[70]. The boundary supervision techniques [71][72] and the two branches [73][74] are the major tools used in successful deep learning-based IML models. By combining RGB spatial data with noise view or frequency domain features, the two-branch-based models aim to increase the detection accuracy. The noise view perspective detects tampered sections by utilizing the information that the new parts added through splicing or removal differ from the pristine part in terms of noise distribution. This allows it to capture traces of image forgeries. A predetermined high-pass filter or limited convolution layer is used to construct the noise map given an input image. This noise map is then supplied to a deep neural network either separately [75] or in combination [76] with the input image. The image manipulation traces are improved by the noise inconsistencies derived from these noise streams. This approach is not very effective for identifying copy-move without introducing new elements. On the other hand, discrete cosine transform or rapid Fourier transform are primarily used to extract frequency information to make it easier to capture small indications of forgery that are no longer evident in the RGB domain [14]. The frequency modality that has been added on top of RGB information can strengthen the model's resistance to several image compression techniques. Nevertheless, only the high-frequency information was investigated in the majority of the models that were already in use; the frequency information was not extensively utilized. Methods based on border supervision have been presented consecutively to capture the forged traces around the tampered area. Empirically, the boundary artifact placement information was also somewhat beneficial for detecting the tampered regions [59]. For instance, the Sobel filter was employed by MVSS-Net [77] and its enhanced versions MVSS-Net++ [59] to construct an edge-supervised branch, which produced more targeted feature responses close to the forged regions. Additionally, Zhou et al. [78] used a discriminative generator and uniformly concatenated the backbone characteristics from various layers as the input of the auxiliary branch to segment and correct the boundary artifacts produced during the picture tampering process.

For many years, the field of media forensics has been established to identify fraudulent activities. Early research focuses on projecting images straight into binary label space (real/manipulated) using conventional features. Detecting manipulation at

the pixel level (IML) is the subject of a few works [79]. Conventional IML techniques rely on hand-crafted features, such as self-consistency [79], point matching [53] and Markov features [80], among others, which have a weak generalization capacity and strictly rely on the domain expertise of human experts. Nevertheless, several techniques—such as splicing [81], copy-move [82] and inpainting[75] —have only been studied to identify a single particular kind of alteration and are hence unsuitable for the general localization of image forgeries. A new generation of frameworks is desperately needed to address the challenges mentioned above to achieve more refined outcomes at the pixel level for more semantically complex and perceptually compelling images in the real world.

Many deep learning-based techniques have been presented in the recent few years to tackle the IML problem for the three common tampering procedures mentioned above, and they have demonstrated considerable potential. Bappy et al. [83] used the LSTM-based patch comparison method to identify the border around tampered sections. They also suggested general solutions for the hybrid encoder-decoder structure to enhance the algorithm's performance. Before the end-to-end framework with three high-pass filters, Wu et al. [84] used the steganalysis-rich model to investigate the noise inconsistencies between the tampered and clean regions. For the pixel-level IML challenge, however, the previously mentioned approaches remain far from useful in terms of resilience, feature generalization capacity, and segmentation accuracy. In order to do this, Hu et al. [85] developed a spatial pyramid attention network that builds on local self-attention to describe the link between multi-scale visual blocks accurately, hence improving detection accuracy. More recently, Wang et al. [68] introduced ObjectFormer, which uses learnable object prototypes based on attention and frequency attributes to detect tampering artifacts.

In the realm of natural language processing, architectures based on self-attention mechanisms, particularly the Transformer framework [86], have emerged as the top option due to their strong capacity to model long-range context information [87]. Dosovitskiy et al., [13] presented the ViT model, which eliminated the need for CNNs and worked best on the ImageNet classification dataset as a way to apply transformers to computer vision problems. The Pyramid ViT [88] and Swin

Transformer [89] were created to solve the challenges of porting the Transformer to multiple dense prediction applications, in contrast to the ViT, which was exclusively designed for image categorization. Robust hybrid Transformer architectures like TransFuse [90] and NestedFormer [90] include the Transformer into CNN to improve medical picture segmentation in the interim. NestedFormer [90] explicitly investigated multi-modal MRIs' intra- and inter-modality relationships to segment brain tumors. TransFuse [90] enhanced the effectiveness of modeling global contexts while keeping a good grasp of low-level details by combining Transformer and CNN in parallel.

2.5 Deepfake Detection

Digital picture editing has become more common in recent years. Therefore, it is challenging to confirm the authenticity and integrity of photos because it is so simple to manipulate an image. Deepfake detection methods use sophisticated machine learning algorithms to spot artificial manipulations, guaranteeing the integrity and authenticity of digital content. Deepfake detection focuses on identifying and mitigating the risks associated with manipulated media, particularly videos and images where individuals' faces or voices are convincingly altered using artificial intelligence. Numerous facial manipulation detection techniques have been put forth. The initial attempts relied on handcrafted features that were derived from irregularities and artifacts in the process of creating fake images. Deep learning has been widely used in recent techniques to extract salient and discriminative features to detect facial manipulations automatically. Although facial manipulation detection techniques have advanced significantly, they still have certain challenges and disadvantages. The diversity of training data affects the efficacy of these techniques. A lack of diversity in the training data may make it difficult for the model to identify more recent and advanced deep fake faces. Academics and industry professionals are actively addressing these issues, and continuous developments in data gathering, model architectures, and technology are intended to increase the robustness and dependability of facial manipulation detection techniques. A few noteworthy facial manipulation detection studies have been examined and considered.

In related studies focusing on facial manipulation detection, authors provided a combined approach that integrated You Only Look Once (YOLO) and Linear Binary Pattern Histogram (LBPH) [91]. The approach demonstrated the effective use of the YOLO-LBPH face detector for identifying facial regions in video frames, while feature extraction was performed using EfficientNet-B5. The precision (P) score of 88.9% and recall (R) score of 93.76% were achieved on the Diverse Fake Face Dataset (DFFD). In the other work [92], the Fisher Face Linear Binary Pattern histogram using the Deep Belief Network (FF-LBPH DBN) classifier method achieved an impressive accuracy rate of 97.82% on the DFFD dataset. Certain research work [93] explored a variety of deep learning and machine learning-based models for detecting GAN-based manipulation. It is shown that DenseNet-121 can detect artificially generated anomalies in medical imaging with an accuracy of 80.4%. Also, a deep learning-based approach [94] was used for detecting deepfakes, aiming to aid cyber security professionals in combating deep fake-related cybercrimes by accurately identifying manipulated content. The study employed and compared with several neural network models. The approach achieved an impressive accuracy of 94%. Different models' classification accuracy was examined in a separate study [95]. When applying pre-processing techniques, the CNN-only model without PCA achieved 63.86% accuracy, while the CNN model with PCA classifier achieved 74.26% accuracy. Without any pre-processing stages, the CNN-only model achieved 93.16% accuracy and the CNN model with PCA achieved 90.76% accuracy. Furthermore, increasing the number of samples used for training and testing in the CNN network resulted in the highest accuracy of 98.04% for image classification. In another study [96], spatio-temporal information was extracted using a convolutional neural network to detect facial manipulation. It has been observed that researchers have developed methods to identify a particular type of facial manipulation, but they are not robust enough to detect multiple facial manipulation techniques. Also, the models that were developed are very complex and computationally expensive.

2.5 Research Gap

The following research gaps are identified for future work based on the literature survey.

- Lack of comprehensive, systematic reviews that holistically consolidate recent advancements, datasets, and methodologies. This gap presents a strong case for a review article, as no unified source sufficiently captures the breadth and depth of developments in this rapidly evolving area.
- Hand-crafted features-based approaches have not proven good enough for manipulation detection as they require more human intervention, are not automatic, and are not robust, so there is always scope for other deep learning-based approaches.
- There is a lack of specialized systems tailored to specific forensic applications, such as face or offline signature verification, which require highly accurate and context-sensitive approaches. Research focused on application-specific manipulation detection frameworks could yield more targeted solutions for fields like biometric security, document verification, and media forensics, which have unique requirements and constraints.
- A robust framework is needed to detect multiple forgeries in a single model. Most existing methods focus on detecting specific types of forgeries. However, limited robustness is observed across multiple forgery detection methods.
- Methods developed for manipulation detection perform well as they have to do the binary classification (detecting whether an image is manipulated) but struggle to localize the manipulated region in the image accurately. Localization is a bit difficult compared to detection. Existing methods perform well for image-level manipulation detection but often lack pixel-level image manipulation localization.
- Image manipulation methods rapidly evolve with the advancements in generative AI, creating a gap between new manipulation techniques and existing detection methods. Developing adaptive and robust models capable of handling evolving manipulation techniques like deepfake is crucial.

2.6 Research Objectives

Based on the literature review, the main objectives of this thesis work are as follows:

- To review state-of-the-art handcrafted and deep learning approaches, image manipulation datasets, existing solutions and their limitations.
- To develop an effective approach for image manipulation detection.
- Design a manipulation detection system for various specific forensics applications such as face verification.
- Multiple forgery detection (universal forgery detection) and localization of forgery in a tampered image.
- To study and formulate the various deep learning-based approaches for image manipulation detection systems, including deepfake detection.

Chapter 3

Image Manipulation Detection for specific forgery

This chapter explained the proposed methodology for detecting specific manipulations, such as OfSFD and CMFD. A detailed description of the problem statement, dataset, feature extraction process and methods adopted are provided in this chapter. The effectiveness of the proposed approach is explained and validated through experiments on standard datasets and a SOTA comparison study of the results is provided.

3.1 Offline Signature Forgery Detection

3.1.1 Introduction

Signature is one of several biometric traits commonly used for user verification, including fingerprints, palm geometry, face, retina, iris, and voice. Handwritten signatures are regarded as a reliable biometric attribute since they are unique to each individual and difficult to reproduce. Signatures are often employed as authentication hallmarks because they are simple, socially and legally acceptable to legitimate entities. Because the signature is the primary means of validation and approval in legitimate transactions, it is crucial to anticipate its authenticity. Signatures have long been regarded as the most widely accepted and logical methods of user verification, notwithstanding their vulnerability to expert forgers. Several attempts have been made to address the vulnerability related to the manual authentication scheme. Manual signature verification of various documents is time-consuming and requires human carefulness, expertise, and mastery to differentiate and detect forged signatures. A robust automated framework for OfSFD is needed. As machine learning techniques advance, researchers create various machine learning-based signature forgery detection approaches [32], as discussed in Section 2.1.

Signature forgery detection is classified into two categories based on their data acquisition approach: online signature forgery detection (OnSFD) and the OfSFD method [97]. In online or dynamic methods, a signature is obtained using an electronic device such as a tablet, smartphone, or electronic writing pad with a stylus pen. Each stage in the signing process provides information about the pen's location, inclination angle, stroke order, writing speed, and pressure [98]. In contrast, offline or static approaches collect signatures by scanning a handwritten signature on a document and converting it to a digital image [99].

In OfSFD systems, the process begins with digitally obtaining and preserving the individual's signatures. In signature forgery detection systems, strong features are taken from the training set signature image and compared to those extracted from the test image. The OfSFD system is the most well-known individual authentication technique for banking or business [100]. *Fig. 3.1* depicts a general workflow for an OfSFD system. The database contains registered and query signer signature images. The images are then preprocessed to extract appropriate features. A model is trained to do classification based on a score to determine if a signature image is genuine or forged.

Furthermore, the OfSFD employs two distinct approaches: writer-dependent (WD) and writer-independent (WI) [101]. The WD strategy trains the model for each writer, requiring a distinct classifier for each writer, whereas the WI approach requires a single global classifier for all writers. The WI approach uses a broad model, making it more practical and popular than the WD signature forgery detection approaches. As the number of users in the WD method grows, each user requires a separate classifier, increasing complexity and computing cost. A WI approach is more practical and user-friendly because it uses a single global classifier for all users [101].

Offline signatures are collected after they have been written on a document, then scanned and displayed as a digital image. Because of this, dynamic information about the signature, such as the location and speed of the pen over time, is lost, making OfSFD a difficult task. Offline signatures can be forged in three ways: simply, randomly, or skillfully. Simple forgery occurs when the forger is unaware of the real

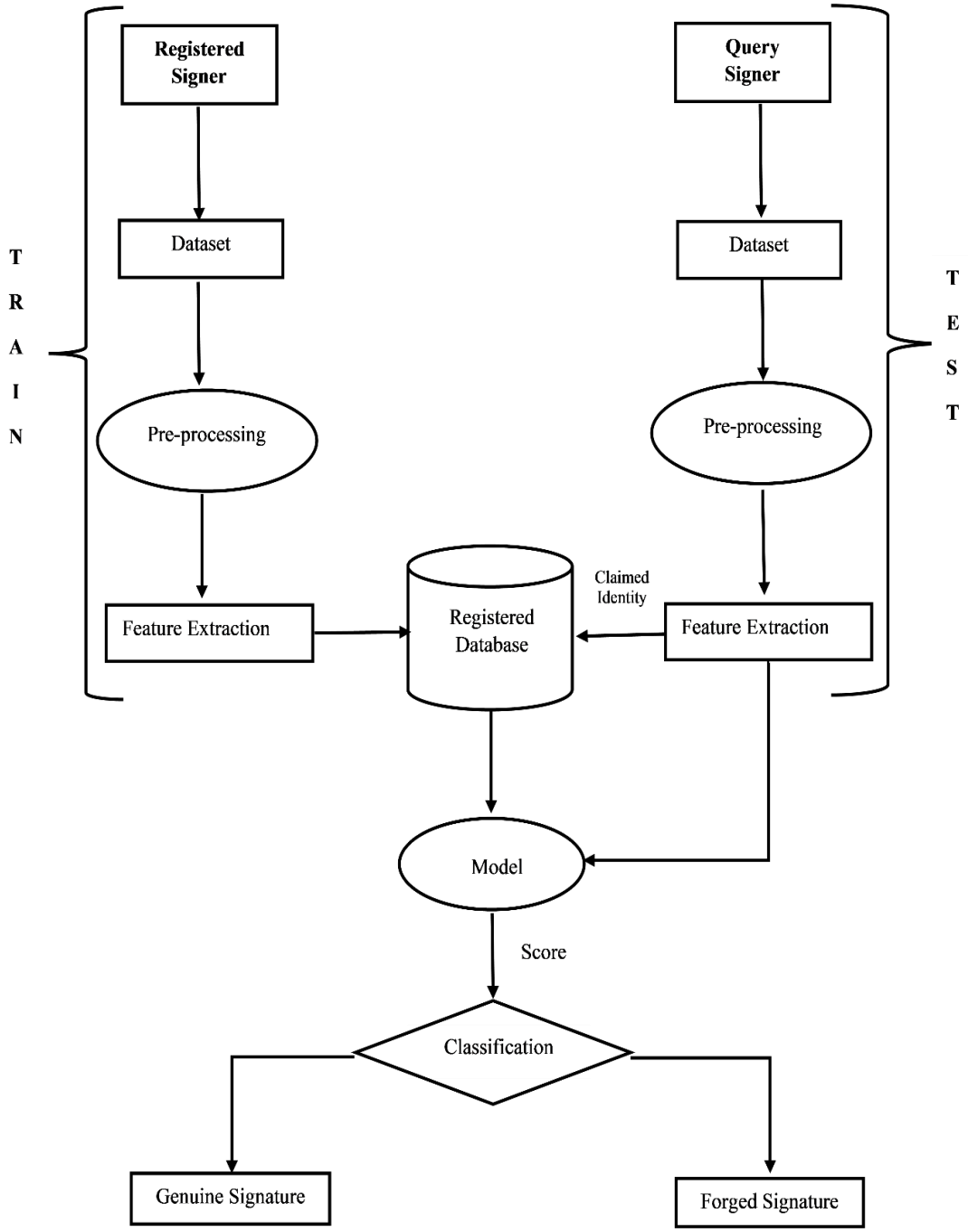


Fig. 3.1: General process of an offline signature verification system.

signature. However, they are aware of the signer's name, whereas in the case of random forgery, the forger substitutes their signature for a genuine signature, and in the case of skillful forgery, the forger is aware of the signer's name and genuine signature and attempts to emulate the signer's signature [102]. OnSFD methods outperform offline

equivalents due to data availability such as the pen location, inclination angle, stroke order, writing speed, and pressure. However, OnSFD techniques require specialized equipment, raising the framework cost and limiting real-world application scenarios [101]. There are numerous situations where authenticating an offline signature is the only choice, such as check transactions and document verification. As a result of its more extensive application region, this work centers on developing an automatic writer-independent offline signature forgery detection system (WIOfSFD). An EfficientNet-based Siamese neural network (eSNN) is proposed to discriminate between forged and genuine signatures.

It is observed that skilfully done forgeries closely resemble genuine signatures. Here are a few challenges encountered when developing feature extractors for these genuine signatures: (1) Genuine samples of signatures occasionally have completely distinct shapes. For such genuine samples, the feature extractor used would have produced significantly different feature vectors. (2) In some cases, the character shapes can differ greatly. Research focusing on how individual letters appear will produce poor results. (3) Directional-based descriptors (such as HOG or D-PDF) may be impacted by significant flourish fluctuation. (4) Some users find it difficult to distinguish between the traits of two signatures, even after thoroughly studying the data. (5) The available signature dataset is insufficient to cover all the signature characteristics. Handcrafted feature engineering will not be a suitable solution for this signature verification problem. Using deep convolutional neural networks as feature extractors could solve the problem, but due to the limited availability of signature datasets, it would not be easy to get discriminative features for genuine and forged signatures. Therefore, a popular pre-trained EfficientNet-B7 model is used as a feature extractor. This gives a dynamic, robust, and efficient feature extractor to solve the problem of signature forgery detection.

3.1.2 Efficient Siamese Neural Network for WIOfSFD

An eSNN is proposed for WIOfSFD. The main task is to classify whether the individual signatures are forged or genuine. eSNN is used to compensate for convolutional networks' shortcomings in detecting and differentiating between

discovering spatial component variations and alterations in image components while still utilizing their strengths in non-manual learnable identification and extraction of features. The complete architecture of the proposed network is shown in *Fig. 3.2*. The method is based on the Siamese network and EfficientNet model.

The Siamese network has two subnetworks, which give the feature vector representation of the respective input sample image. A subnetwork consists of an EfficientNetB7 pre-trained model on ImageNet followed by a flattening layer and two dense layers to finally get the feature vector representation for the input sample images. Both subnetworks are combined using a loss function that calculates the Euclidean distance between the two feature vectors obtained from the two subnetworks. The similarity score between two input sample images in the joint space is computed using the Euclidean distance function. However, preprocessing is done on the raw image before passing the image to the respective networks. Images in datasets range in size from 304×240 to 798×482; therefore, all the images are resized to 224×224 and preprocessed to ensure consistency. The outputs of the sub-networks are then compared using a loss function, typically through a distance metric that is Euclidean distance, to produce a similarity score. The contrastive loss [103] is one such loss function that is frequently employed in SNN and is calculated as follows:

$$L(e_1, e_2, Y) = \frac{1}{2}(1 - Y)D_w^2 + \frac{1}{2}Y \max(0, 1 - D_w)^2 \quad (3.1)$$

where e_1 and e_2 are two feature vectors, Y is a binary indicator 0 for the same class and 1 if the signature samples are from a different class. $D_w = \|f(e_1) - f(e_2)\|$ is the Euclidean distance calculated in the embedded feature space, and f is an embedding function from sub-networks that translates a signature image to real vector space. The feature vectors obtained from the sub-networks are compared by the contrastive loss function that computes the Euclidean distance between the two feature vectors in the embedded space. The Siamese network attempts to push the output feature vectors away if the input pairs are dissimilar and to push the feature vectors closer for input pairings that are tagged as similar, in contrast to conventional techniques that assign binary similarity labels to pairs. The resulting space will have the characteristic that images belonging to the same class (a genuine signature for a specific signer) will be

closer to each other than images belonging to dissimilar class (signatures of different signers) as a result of the loss function given by Eqn. 3.1. The next step is to establish a threshold value (t_h) for the distance between two images to assess whether they belong to the same class (genuine, genuine) or a distinct class (genuine, forged). Finally, the query signature is accepted for classification if the similarity score is less than the chosen threshold and rejected otherwise. Eqn. 3.2 is used to verify each user.

$$\sigma(X|S) = \begin{cases} Genuine, \ if\ S < t_h \\ Forged, \ otherwise \end{cases} \tag{3.2}$$

Where X is the query user and t_h is the distance threshold value. The user is considered genuine if the similarity score (S) between two input sample images is less than t_h.

The eSNN takes the offline signature pair from two different users (for example, users 1 and 2) and generates an output based on how similar the two pairs are, as illustrated in *Fig. 3.2*. In a Siamese network, the sub-networks share the same weights and biases, so they are "identical". This allows the network to learn a common representation of the input data and apply it to both inputs. The signature of the genuine user 1 is passed to the first subnetwork and the signature of user 2 produced while attempting a forgery of user 1's signature, is passed to the second subnetwork. The signature image is passed to the subnetwork to perform the feature extraction using the pre-trained model EfficientNetB7.

3.1.2.1 Siamese Neural Networks

For autonomous feature extraction and classification problems, deep learning architectures utilizing CNN approaches have gained popularity. However, it is well acknowledged that these techniques call for a lot of labeled data, which may not be feasible (or desired) to verify signatures. As a result, a new strategy is required that can be extended to new users without retraining the model and that can be trained on smaller data samples. One possible strategy is one-shot learning, which can be carried out with a Siamese network of twin sister networks with identical weights.

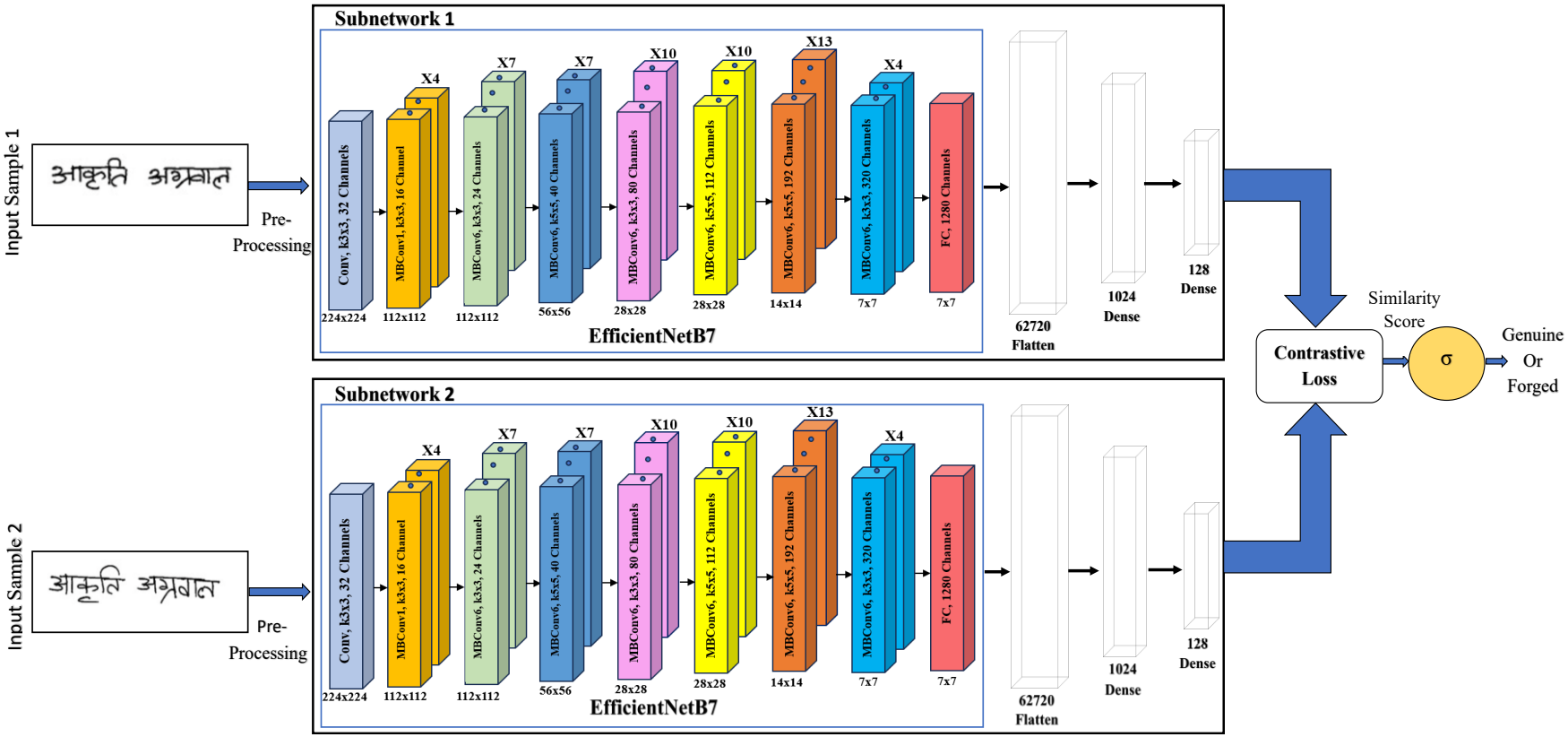


Fig. 3.2: eSNN: the proposed architecture for signature forgery detection

A Siamese network is an artificial neural network, sometimes called a twin neural network, which involves pairwise learning. Siamese network architecture generally consists of two subnetworks having the same configuration, such as identical parameters and shared weights [104]. Here in the SNN, feature vectors of the two input images are obtained from the subsequent subnetworks. Then, these feature vectors are compared using Euclidean distance, which computes the similarity score or distance from each class. The Euclidean distance is small when the two input images belong to the same class, such as genuine-genuine (genuine signatures from the same user) and is large for the dissimilar class, such as genuine-forged (forged or signatures from the other user).

In contrast to conventional neural networks, SNNs are more robust in an imbalanced dataset. The advantage of using Siamese networks is that they can effectively learn the similarities and differences between two inputs, even if the inputs are from different domains or have different distributions. This makes them useful for tasks like signature forgery detection, where defining a clear similarity metric is difficult. However, Siamese networks are slower than conventional classifying neural networks. Therefore, in the proposed model, the subnetwork gives the feature vector as output for the input sample image based on the weights of the pre-trained model (EfficientNet) instead of learning from scratch.

3.1.2.2 Feature Learning

Feature learning is performed on the preprocessed images. In feature learning, several features are extracted to distinguish between different signatures. The eSNN takes a pair of inputs consisting of a signature sample. Now, the preprocessed images are passed to the respective sub-networks. The sub-network consists of EfficientNet-B7, where features are extracted using transfer learning from each signature sample. In the proposed method, a pre-trained model is used to make the process more robust, typically a deep neural network EfficientNet-B7 trained on a large, general-purpose dataset, ImageNet, which contains millions of images. This model's pre-trained weights and biases are used as initialization for fine-tuning, which helps reduce the

training data and computational resources required to solve the problem. The pre-trained model extracts high-level features from the input sample images in feature extraction. Due to limited labeled training data for WIOfSFD, transfer learning from a pre-trained model can leverage the knowledge from large amounts of data and improve performance.

EfficientNet is a family of convolutional neural networks designed by a team of researchers at Google AI in 2019 [105]. EfficientNet is known for its efficiency in terms of both accuracy and computational resources, making it a popular choice for signature forgery detection tasks. EfficientNet has achieved SOTA performance on several computer vision benchmarks, including ImageNet, COCO object detection, and PASCAL VOC segmentation. Therefore, EfficientNet-B7 is used as a feature extractor in the SSN's subnetworks as it performs best over other ConvNets.

EfficientNet is a pretty large network consisting of many learnable parameters (approximately 66 million) obtained after training the network. EfficientNet is trained on a large dataset (ImageNet, which contains over 15 million images); thus, a considerable computational asset would be required for training. This could be a problem because accessing such a high computational machine is difficult whenever you train the network. In particular, for huge image datasets, it has been observed that the low-level features learned from the initial layers of the network are generally the same irrespective of the dataset. Therefore, pre-trained weights trained on ImageNet obtained from the EfficientNet-B7 model can be used to initialize the other network. This helps reduce training time and makes the model more robust, which results in lower generalization errors.

3.1.3 Experiment

The experiment adheres to the specified design to examine the effectiveness of the proposed method for signature forgery detection: (1) Load data and generate pairs of similar (genuine, genuine) signatures and dissimilar (genuine, forged) signature classes. (2) Preprocessing is done to have greyscaled, binary, noise-free, sharpened, and normalized images. (3) Each dataset is split into three separate datasets for

training, validation, and testing purposes. After examining the dataset structure, each part is allotted 60%, 20%, and 20% of the whole dataset accordingly (except the ICDAR 2011 SigComp dataset, where the train and test set is predefined). (4) After loading the pre-trained model, add a flattening layer and two dense layers to create the final subnetwork. (5) Train the network on different datasets separately to help the network learn the weights and reduce the loss function. The loss function gives the similarity score based on the Euclidean distance between the two feature vectors. The validation dataset determines the ideal threshold value to provide the anticipated class labels during training. (6) Calculate the evaluation parameters for each set of test data. The final performance of the model is assessed using test data in accordance with the assessment criteria.

3.1.3.1 Preprocessing

Preprocessing aims to prepare all the signatures for further operations and make learning more feasible. It is a general understanding that signature images have intra-class variance. Different examples of the same individual signature will vary due to fluctuations in a person's mood, state of mind, etc., and a lack of space on the writing surface. These reasons explain why various samples of the same signature frequently differ in height, width, skewness, etc. Several preprocessing steps have been carried out in the current analysis to eliminate these intra-class variances. *Fig. 3.3* illustrates the preprocessing procedure for the raw signature images from the datasets. The raw images are resized to 224×224, as the default fixed size of the image is taken for training. The resized images undergo three essential steps: (1) Gray Scaling: Images are converted from the RGB image into the grayscale image. To build a 3-channel input image for the pre-trained model, the one-channel image created by the grey scale operation is stacked to create three levels of equal pixel values. (2) Binarization: The image is transformed to a binary image with only black and white pixel values using Otsu thresholding [106] to reduce noise.

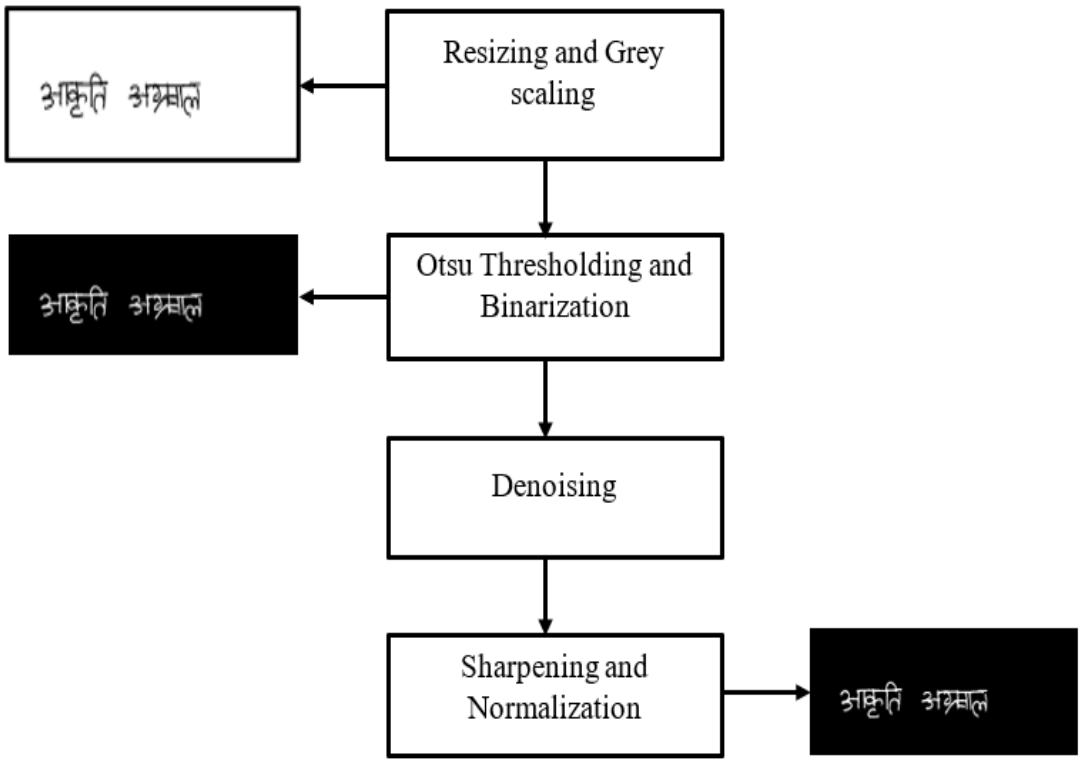


Fig. 3.3: Preprocessing procedures for the raw signature samples from the datasets

3.1.3.2 Dataset

Various datasets with unique structures and signature characteristics evaluate the eSNN performance. The datasets taken into consideration are GPDS-Synthetic [107], MCYT-75 [108][109], CEDAR [29], BHSig260 [110], ICDAR 2011 Signature Verification Competition [37] and UTSig [111]. Some examples of real and forged signatures from each dataset are presented in *Fig. 3.4* to help the viewer comprehend the signatures gathered in each dataset. Three real signatures from the same individual in the dataset are displayed in each row, along with a forged signature image of the same user. Six different datasets are employed in this analysis. *Table 3.1* shows the details of the datasets.

Table 3.1: OfSFD Datasets Description

Dataset	**Script**	**Signers**	**Signature Samples**	**Signature Samples (Genuine/ Forged)**	**Signatures per Signer (Genuine/ Forged)**	**Description**
GPDS-Synthetic [107]	English	4000	216000	96000 /120000	24/30	600 dpi, JPG format
MCYT-75 [108][109]	English	75	2250	1125/1125	15/15	600 dpi, greyscale, BMP format
CEDAR [29]	English	55	2624	1320/1320	24/24	300 dpi, greyscale, PNG format
BHSig260 (Hindi) [110]	Hindi	160	8640	3840/4800	24/30	300 dpi, greyscale, TIF format
BHSig260 (Bengali) [110]	Bengali	100	5400	2400/3000	24/30	300 dpi, greyscale, TIF format
ICDAR 2011 SigComp (Chinese) [37]	Chinese	10 + 10	1178	235+236 /340+367	(21 to 24) /(23 to 36)	400 dpi, PNG
ICDAR 2011 SigComp (Dutch) [37]	Dutch	10 + 54	2297	240+1296 /123+638	(23 to 24) /(8 to 16)	400 dpi, PNG
UTSig [111]	Persian	115	8280	3105/5175	27/42	600 dpi, greyscale, TIF format

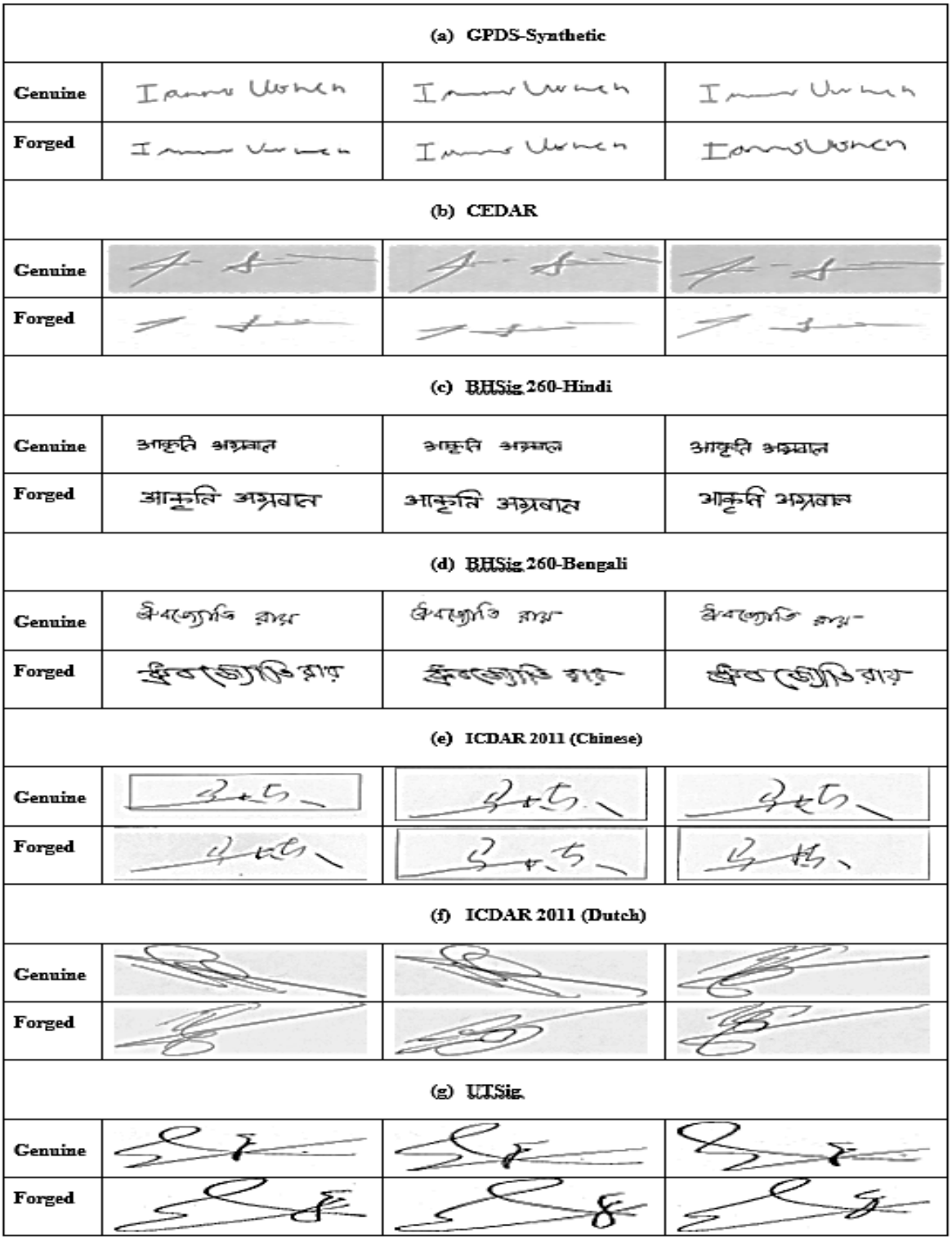


Fig. 3.4: Examples of genuine and forged signatures samples from different datasets (a) GPDS-Synthetic, (b) CEDAR, (c) BHSig260 (Hindi), (d) BHSig260 (Bengali), (e) ICDAR 2011 (Chinese), (f) ICDAR 2011 (Dutch), (g) UTSig. Three real signatures from the same individual in the dataset are displayed in each row, along with a forged signature image of the same user.

3.1.3.3 Experimental Setup

A pre-trained model, EfficientNet, is used from the Keras Application API in the subnetwork. The subnetwork comprises a pre-trained network (EfficientNet) whose layers are freezed followed by flattening and adding two dense layers. The first dense layer consists of 1024 neurons with a dropout rate of 0.5, whereas the second dense layer consists of 128 neurons. At the output of the dense layers, the Rectified Linear Units (ReLU) activation function is used. This network receives input pairs, and the resulting feature embeddings are provided to a distance function to determine similarity. When given the estimated distance, a loss function modifies the parameters to reduce the distance between pairs of (genuine, genuine) signatures and raise the distance between pairs of (genuine, forged) signatures. A Euclidean distance function determines the distance between the two output image encodings from the twin sub-networks. Adam optimizer is used to train this Siamese network for 15 epochs with contrastive loss, momentum rate of 0.9, the initial learning rate is set to 10^{-4}, and batch size of 64. The model was trained using a 12GB Nvidia Quadro K4200 and Tesla K40C GPU card. The eSNN model has 1.44M trainable parameters and requires around 393M FLOPs, ensuring efficient performance in resource-constrained environments.

3.1.3.4 Performance Evaluation

To determine if the signature pair $(i; j)$ belongs to a similar or dissimilar class, a threshold 't_h' is set for the distance measure $D(x_i; x_j)$, produced by the model. Refer to all signature pairs $(i; j)$ with different identities as "$\rho_{dissimilar}$", whereas all pairs with the same identity are referred to as "$\rho_{similar}$". The set of all *TP* at the threshold 't_h' can, therefore, be defined as

$$TP(t_h) = \{(i,j)\epsilon\rho_{similar}, with\ D(x_i; x_j) \leq t_h\} \tag{3.3}$$

Where $\rho_{similar}$ is the number of similar signature pairs.

Likewise, the set of all *TN* at 't_h' can be defined as

$$TN(t_h) = \{(i,j)\epsilon\rho_{dissimilar}, with\ D(x_i; x_j) \geq t_h\} \tag{3.4}$$

Where $\rho_{dissimilar}$ is the number of dissimilar signature pairs.

Now, for a given signature, *the TPR* (t_h) and the *TNR* (t_h) are defined as

$$TPR(t_h) = \frac{|TP(t_h)|}{|\rho_{similar}|}, TNR(t_h) = \frac{|TN(t_h)|}{|\rho_{dissimilar}|}, \tag{3.5}$$

Therefore, the maximum accuracy can be calculated by varying $t_h \epsilon D$ with a 0.01 step size from *D*'s lowest value to its highest value. The test accuracy is determined by iterating through various threshold levels and counting the number of image pairs that were properly identified. The accuracy of each system was calculated using the best thresholds discovered for it.

$$Accuracy = \underbrace{max}_{t_h \epsilon D} \frac{1}{2}(TPR(t_h) + TNR(t_h)) \tag{3.6}$$

In classification problems, accuracy is not a sufficient requirement by itself. For this reason, FAR, FRR, EER and F1-score values are determined for each class. These metrics can demonstrate how effective a verification system is in discriminating between genuine and forged identities.

3.1.4 Result and Discussion

Table 3.2 compares proposed eSNN and cutting-edge techniques on the various datasets. The proposed method performed better on all the datasets, including the GPDS Synthetic, MCYT-75, CEDAR, BHSig 260, ICDAR 2011 and UTSig datasets. The proposed method outperforms the cutting-edge approaches with respect to Accuracy (A), EER, FAR, FRR and F1 Score criteria. The proposed method, eSNN, performs flawlessly on the CEDAR dataset and is comparable to the other two top approaches, Signet [43] and Compact Correlated Features (Dutta et al. [112]). On the BHSig260 database, the performance of the eSNN is not superior to the most effective method currently available (CBCapsNet [37]). Suggested eSNN approaches significantly outperform the other techniques in all the other datasets. On the ICDAR 2011 SigComp (Chinese) dataset, shown in *Table 3.1*, the accuracy of the eSNN exceeds 95%, while the accuracy of the other methods does not exceed 88%. On the CEDAR dataset, the eSNN accuracy is 100%, and the FAR and FRR are equal to zero. Noteworthy is the fact that the performance of the eSNN is also excellent on the UTSig dataset, which has the lowest EER value. In conclusion, the proposed model has a significant advantage over other SOTA methods in five of the six datasets.

Table 3.2: Comparison between the proposed eSNN and cutting-edge techniques.

Dataset	Methods	Accuracy	EER	FAR	FRR	F1 Score
GPDS-Synthetic	Dutta et al. [112]	73.67	-	28.34	27.62	-
	SigNet [43]	77.76	-	22.24	22.24	-
	LS2Net [113]	96.91	-	-	-	-
	Inception-v1 [114]	77	0.22	-	-	0.753
	CBCapsNet [37]	90.87	-	9.45	8.81	-
	Yapici et al. [115]	-	12.34	8.66	10.41	0.88
	eSNN	98.23	2.265	3.01	2.52	0.86
MCYT-75	LS2Net [113]	96.41	-	-	-	0.97
	Ooi [116]	-	9.87	-	-	-
	Soleimani et al. [117]	-	9.86	-	-	-
	Alonso et al. [118]	-	29.62	26.84	32.4	-
	Hezil et al. [119]	-	7.78	6.23	9.33	-
	Bhunia et al. [120]	-	6.10	6.00	6.20	-
	Maergner et al.[121]	-	3.91	-	-	-
	Sima et al.[122]	-	5.46	-	-	-
	Masoudnia et al.[123]	-	5.85	-	-	-
	Yapici et al. [115]	-	2.58	2.66	1.33	0.97
	eSNN	97.82	2.54	2.54	2.54	0.98
CEDAR	Kalera et al. [29]	78.50	-	19.50	22.45	-
	Chen and Srihari [124]	83.60	-	16.30	16.60	-
	Chen and Srihari [125]	92.10	-	8.20	7.7	-
	Kumar et al. [126]	91.67	8.33	8.33	8.33	-
	Dutta et al. [112]	100.0	0.00	0.00	0.00	-
	SigNet [43]	100.0	0.00	0.00	0.00	-
	LS2Net [113]	98.30	-	-	-	0.99
	CBCapsNet [37]	100	0.00	0.00	0.00	-
	Maergner et al.[121]	-	5.91	-	-	0.97
	Sima et al.[122]	-	4.94	-	-	-
	eSNN	100	0.00	0.00	0.00	0.99
BHSig260 (Hindi)	Pal et al. [110]	75.53	24.47	24.47	24.47	-
	Dutta et al. [112]	85.90		13.10	15.09	-
	SigNet [43]	84.64	15.36	15.36	15.36	-
	CBCapsNet [37]	100	0.00	0.00	0.00	-
	eSNN	89.28	10.72	10.72	10.72	0.99
BHSig260 (Bengali)	Pal et al. [110]	66.18	33.82	33.82	33.82	-
	Dutta et al. [112]	84.90	NA	15.78	14.43	-
	SigNet [43]	86.11	13.89	13.89	13.89	-
	CBCapsNet [37]	94.3	NA	5.11	6.29	-
	eSNN	88.69	11.30	11.28	11.32	0.98
ICDAR 2011 SigComp (Chinese)	Liwicki et al.[127]	80.04	NA	19.62	21.01	-
	Alvarez et al. [128]	88	NA	8.2	18.2	-
	eSNN	96.16	4.01	3.84	4.17	0.91
ICDAR 2011 SigComp (Dutch)	Liwicki et al.[127]	97.67	NA	2.19	2.47	-
	Alvarez et al. [128]	94	NA	13.32	3.13	-
	eSNN	97.88	2.09	2.02	2.1	0.99
UTSig	Maergner et al.[121]	-	14.09	-	-	-
	Sima et al.[122]	-	12.88	-	-	-
	Masoudnia et al.[123]	-	7.02	-	-	-
	eSNN	98.39	2.39	2.58	2.53	0.97

'-' represents not available.

The ROC curve shown in *Fig. 3.5* is used to evaluate the performance of the eSNN across several threshold values. ROC is a probability curve that indicates the level or amount of separability. It indicates how well the method can discriminate between classes. The method works better if the AUC is high. It is evident from *Fig. 3.5* that the CEDAR dataset area under the curve is much higher, indicating that the proposed method performed better on the CEDAR dataset than the other datasets.

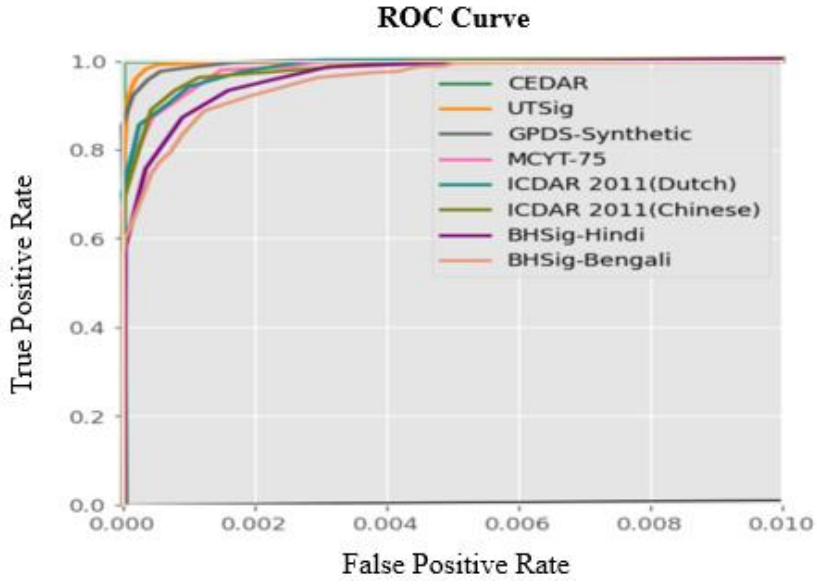


Fig. 3.5: ROC curve of the proposed method for different datasets

3.2 Copy-Move Forgery Detection

3.2.1 Introduction

CMFD is a specialized area within digital image forensics that focuses on detecting a particular sort of image manipulation. Copy-Move Forgery encompasses copying a part of the image and pasting it within the same image. CMFD is one of the most popular techniques because of its subtlety and ease of manipulating images with minimal visible inconsistencies. Copy-move forgeries are difficult to detect by conventional image manipulation detectors, which generally look for anomalies caused by external objects or lighting inconsistencies. However, copy-move forgeries

might cause noticeable distortion, such as minor edges, textures, or color inconsistencies.

CMFD encounters various challenges due to the diversity and complexity of forging tactics. Forged regions can undergo modifications such as rotation, scaling, and blurring, which alter visual and spatial properties, making detection difficult. JPEG compression and noise addition complicate detection by reducing image quality, making it difficult for computers to identify duplicate parts reliably. Furthermore, recurring patterns in images, such as textures in foliage or building facades, produce false positives since actual similarities can seem replicated portions. Furthermore, minor or subtle forgeries, particularly well-blended ones, necessitate extremely sensitive detection methods to distinguish tampered content from authentic patterns. As a result, developing strong, accurate, and efficient CMFD for various manipulation scenarios remains a serious research issue.

3.2.2 Residual-based CNN Method for CMFD

The proposed method uses the Second Difference Median Filter Residual (SD-MFR) and the Laplacian filter residual (LFR) to suppress image content and only explore the inconsistencies left behind after the tampering operation. These two residuals act as input to a robust CNN architecture to detect the traces in tampered images and classify them as so. The complete process flow is given in *Fig. 3.6.*

The proposed CMFD technique uses the SD-MFR and LFR residuals as combined input to the novel CNN network to classify images as authentic or tampered. SD-MFR is used to capture the median filter residuals, and LFR is used to capture the blurring features. Instead of directly feeding the image as input to the CNN, some preprocessing is performed on the image.

3.2.2.1 Preprocessing

Images are resized to 128×128 and converted to grayscale before finding their filtering residuals. First, calculate the SD-MFR median of the image given by Eqn. 3.7, then

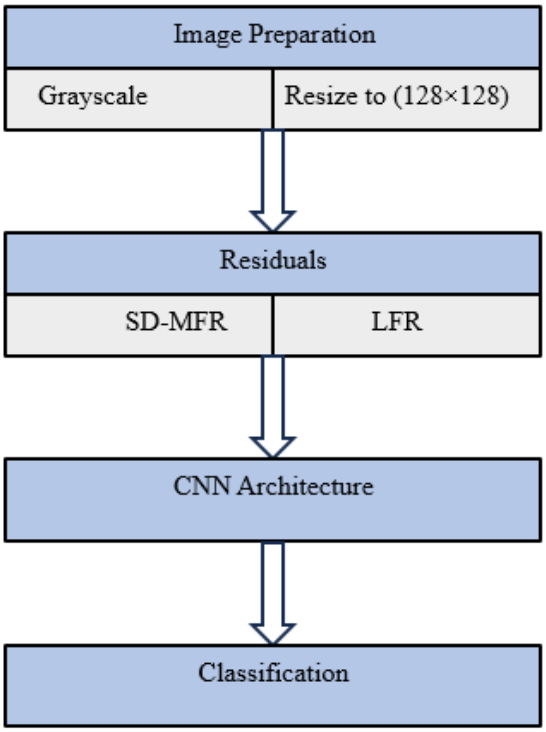


Fig. 3.6: The proposed model flow for CMFD

the median of the median is given by Eqn. 3.8 and finally, the MFR is calculated using Eqn. 3.9 by subtracting Eqn. 3.7 from Eqn. 3.8.

$$Y_{i,j} = med_w(X_{i,j}) \tag{3.7}$$

$$Z_{i,j} = med_w(Y_{i,j}) \tag{3.8}$$

$$SD - MFR_{i,j} = z_{i,j} - X_{i,j} \tag{3.9}$$

where $X_{i,j}$ is the pixel's intensity at the i^{th} and j^{th} pixel and 'w' represents a 5×5 window for median filtering.

Now, to calculate the LFR, a Laplacian filter mask is used, as shown by Eqn. 3.10.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.10}$$

Further, the Laplacian of the image is obtained by Eqn. 3.11 and then the LFR is calculated by subtracting the image from the Laplacian of the image as given by Eqn. 3.12. The LFR is given as:

$$L_{i,j} = Laplacian(x_{i,j}) \tag{3.11}$$

$$LFR_{i,j} = L_{i,j} - X_{i,j} \tag{3.12}$$

3.2.2.2 CNN Architecture

The proposed CNN architecture is inspired by VGGNet [22]. The convolution layer is the core building block of the CNN. It comprises a set of independent filters individually convolved with the input image. They use randomly initialized filters that further become parameters to be trained. Convolution layers are extremely effective in extracting relevant feature maps from images. Let $Z_{i,j}$ give the convolution over an image at the i[th] and j[th] pixel, provided by Eqn. 3.13.

$$Z_{i,j} = \Phi(\sum_{l=0}^{L}\sum_{m=0}^{M} w_{l,m}X_{i+l,j+m} + w_b) \tag{3.13}$$

Where $X_{i,j}$ is the intensity of the pixel at the location *i, j* of an input image and $w_{l,m}$ denotes weight, w_b is the bias, Φ denotes the activation function and the L×M is the size of the kernel.

The proposed network consists of six convolutional layers with ReLU activations. The ReLU activation function is used to get nonlinearity in the network. The ReLU activation function is based on the thresholding operation and is expressed in Eqn. 3.14.

$$\Phi(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{3.14}$$

The first convolution layer is for dimension reduction and has 64 kernels of size 1×1×2. The second layer consists of 64 filters of 3×3 kernel size. The third layer consists of 128 kernels of size 3x3. The fourth and fifth layers have 256 filters of 3x3 kernel size, whereas the sixth convolution layer consists of 512 filters having 3×3 kernel size each. All the convolutional layers are followed by a max-pooling layer except the first layer to reduce the feature size. It performs downsampling by dividing

the input into rectangular pooling regions and taking the maximum value from each pooling region. A pool of 3×3 with a stride of 2 is used. Just like the input is scaled before feeding into the input layer of a network, batch normalization scales the output from the activation of each convolution layer so that the next layer receives scaled input, thereby increasing performance and speed. This layer is used after each convolution layer in the model. Flattening allows changing a high-dimensional tensor's shape into a single dimension so that the dense layer can interpret it. It removes all of the dimensions except one. It is used after the fifth max pooling layer. The dropout layer is used before a dense layer to avoid over-fitting the training data, as the network is trained on a small dataset. It randomly drops out some nodes during an epoch of training so that responsibility for the input is shared equally among the nodes. This layer is used before both the fully connected layers in the proposed framework. A dense layer is a layer in which each neuron accepts the input from all the neurons that were in the previous layer. The proposed method uses two dense layers with 2048 neurons, each with ReLU activations initialized with He initializers and regularized with L2 regularization and an output layer with softmax activation and two neurons. The complete architecture of the proposed network is shown in *Fig. 3.7*.

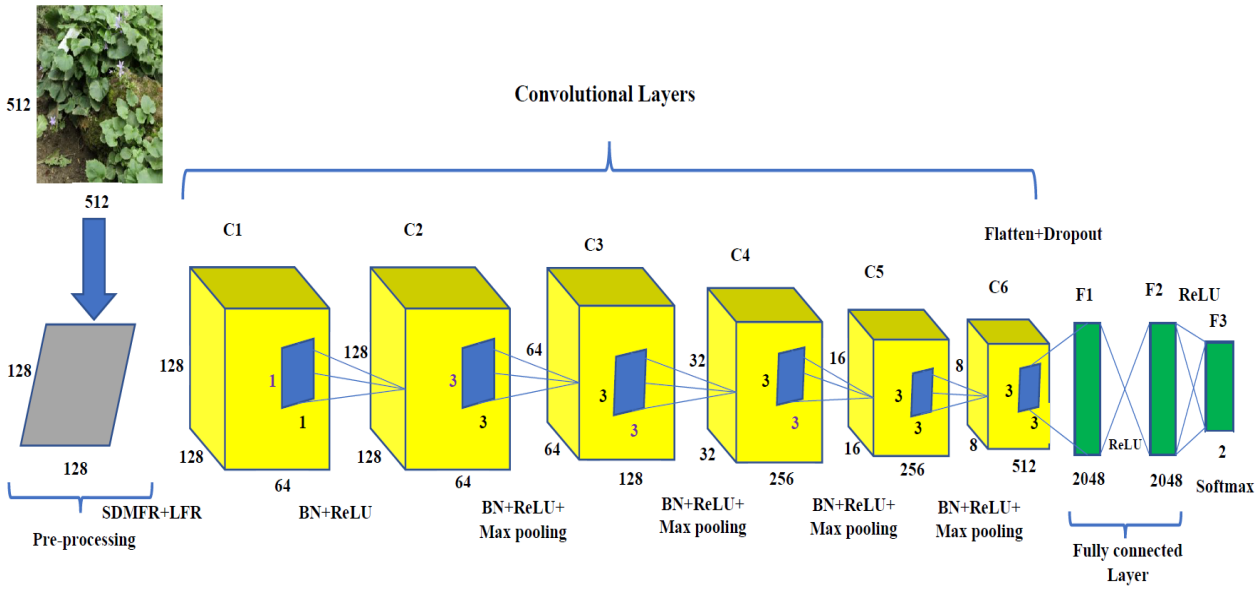


Fig. 3.7: CNN-based framework for CMFD

3.2.3 Results

The proposed network is tested on the CoMoFoD dataset [129], which consists of 4800 original and 4800 forged images. The training and validation images are divided into a ratio of 70:30. Therefore, the network is trained on 6720 images, and the validation set contains 2800 images. The network achieves an accuracy of 95.97% on the validation set. The plot of training and validation accuracy and training and validation loss of the proposed method on the CoMoFoD dataset is in *Fig. 3.8.*

Table 3.3: Performance of the proposed method for CMFD

Dataset	**Training Set**		**Validation Set**		**Accuracy**
	Authentic	Tampered	Authentic	Tampered	
CoMoFoD	3360	3360	1440	1440	95.97%
BOSSBase	7000	3000	7000	3000	94.26%

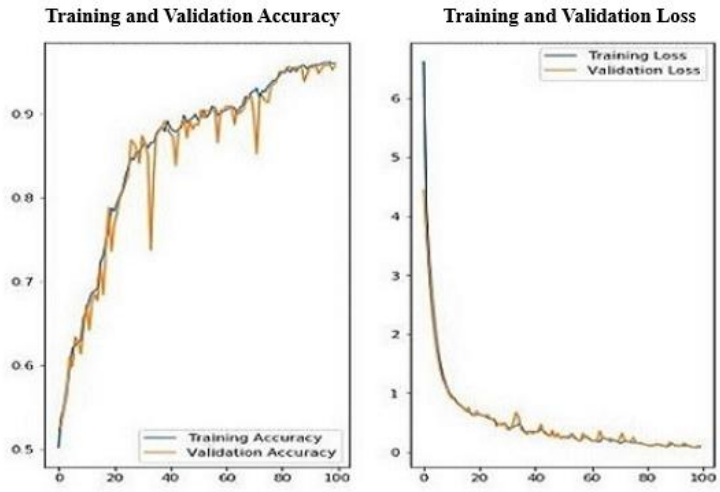


Fig. 3.8: Training and validation metrics of the proposed method on the CoMoFoD dataset.

The proposed method is also tested on the BOSSBase dataset [130] containing 10,000 raw images. Median filtered versions of each image is generated and then train

and test the model on these 20,000 images. The training and validation set is divided into the 70:30 split. The network achieves an accuracy of 94.26% on the validation set. The experimental results show the proposed method's effectiveness and achieve high accuracy.

3.3 Summary

This chapter delves into the targeted approach for detecting specific types of manipulation, particularly emphasizing WIOfSFD and CMFD. An eSNN method is specifically designed to address the challenges of authenticating handwritten signatures for WIOfSFD. eSNN approach uses transfer learning to describe a framework for OfSFD based on Siamese networks and WI feature learning. Unlike the previous approaches, this approach does not rely on handcrafted feature engineering; instead, it learns its features from data in a writer-independent scenario. The performance of the eSNN method was evaluated based on six popular signature datasets. The eSNN was designed to learn spatial features from the pre-trained EfficientNet in each sub-network of the SNN for WIOfSFD. The contrastive loss function generated a similarity score between two pairs based on the Euclidean distance. A decision was made based on the similarity score. The comparison between the performance of the eSNN and SOTA methods was done using various evaluation parameters. The result shows that eSNN had a significant advantage over the SOTA methods on five out of six datasets.

In addition, the chapter proposes a CMFD method using an SDMFR and LFR residual-based CNN framework. This method targets copy—move forgery created by duplicating parts of an image in order to disguise or manipulate content. The method is designed to capture the traces left by postprocessing operations like median filtering and image blurring to detect the discrepancies between copied and authentic regions. The method achieves high detection accuracy for the CoMoFoD and BOSSBase datasets. However, the system is not robust enough to detect tampering where no postprocessing operation has been applied. Hence, detecting such forgeries still has much potential for further research work. The eSNN and residual-based CNN methods

demonstrate advanced, domain-specific solutions for detecting specific forgery types, thereby contributing to improved integrity and security in digital media.

Chapter 4

Multiple Forgery Detection and Localization

This chapter incorporates two different methods to detect multiple manipulations. The first method, MDLFormer, consists of multi-modal input, GCST encoder and FPN-based decoder to detect and localize the manipulation. The second method, LFRViT, is a Laplacian filter residual-based vision transformer for multiple manipulation detection. In this chapter, the methodologies concerning each of the given methods have been discussed in detail. Further, the classification results of the proposed approaches are validated on standard datasets and compared with existing state-of-the-art methods.

4.1 MDLFormer Method for Multiple Forgery Detection and Localization

4.1.1 Introduction

Real-world manipulated images often exhibit multiple forgery operations. Multiple forgery detection refers to the recognition of various types of manipulation. These manipulations may include copy-move, splicing and inpainting forgeries. Detecting multiple forgeries is challenging compared to single forgery detection techniques due to the diversity of manipulation operations, variation in manipulation patterns and the subtlety involved in editing operations. Recent multiple forgery detection methods leverage deep learning models such as CNN and ViT, hybrid frameworks, to analyze the intricate attributes of the image and detect irregularities that indicate manipulation, such as inconsistencies in texture, lighting, color and structure, while maintaining robustness across different manipulations. The development of an effective multiple forgery detection approach must enhance the detection accuracy and aid applications in journalism, law enforcement and digital media authentication, where the verification of the integrity of multimedia content is crucial.

Image manipulation detection emphasizes the existence of manipulations, whereas image manipulation localization seeks to pinpoint and map the precise locations of these changes. Image manipulation detection determines whether an image has been manipulated from its original form. On the other hand, image manipulation localization takes it a step further by identifying that an image has been altered and precisely identifying the specific region within the image that has been manipulated. The localization process typically employs algorithms that analyze the image in detail, detecting areas where manipulation has occurred and marking those regions for further investigation. Detection and localization are vital in diverse domains, including forensics, journalism, medical imaging, and digital media authentication. Detection is useful for identifying potentially manipulated images, whereas localization provides information about the scope and characteristics of the manipulation, allowing for informed decisions regarding the image's authenticity and integrity. Localizing operations makes it possible to make more accurate corrections or adjustments. Image manipulation detection alone can determine if an image has been manipulated. However, localization provides additional detail and context, which is highly important for applications requiring precision, reliability, and accountability.

Despite the SOTA IML solutions stated in Section 2.4 above, two issues still require attention. The primary motivating factors behind this work are these two problems. Problem 1: During feature extraction, attention-based encoding-decoding networks and their derivatives are prone to losing some global context information. Any meddling behavior will somewhat destroy the integrity of the intrinsic features of the original image data itself. Problem 2: Because of edge disruption or body outline concealment, existing approaches frequently struggle to accurately and comprehensively identify the structure and characteristics of fabricated regions, leading to inaccurate predictions with imprecise or incomplete object bounds. Taking this into account, this study proposes a model, i.e., MDLFormer, which consists of multi-modal input that exploits various inconsistencies present in the manipulated image, GCST encoder to capture long-range dependencies as well as local artifacts and FPN based decoder for IMDL. This GCST encoder combines the strong ViT with the classical Global Context block (GCB). GCST encodes richer features using a widely

used Swin Transformer rather than a conventional CNN. Adding the GCB to the Swin Transformer can significantly enhance the model's performance as it simulates the global context efficiently and is lightweight. Finally, the FPN decoder is used to get the predicted mask with the same size as the input. Leveraging these well-designed modules, the proposed MDLFormer performs better image manipulation detection and localization (IMDL) tasks by utilizing multi-modal volumetric data and features extracted through Swin Transformer and the supplementary global context information from the GCB. The proposed IMDL scheme addresses both image-level and pixel-level manipulations. Comprehensive experiments are conducted on diverse standard datasets. The experimental findings confirm that the suggested MDLFormer significantly outperforms the current SOTA IMDL techniques in widely used evaluation metrics.

4.1.2 MDLFormer Model

A manipulation detection and localization model, namely MDLFormer, is proposed to help capture detailed information in manipulated images while overcoming receptive field limitations. The proposed IMDL scheme addresses both image-level and pixel-level manipulations. Three sections comprise the overall architecture of MDLFormer, as shown in *Fig. 4.1*

The primary goal is to detect the manipulated images and localize the image's manipulated regions. As illustrated in *Fig. 4.1*, an end-to-end architecture, which consists of an encoder/decoder known as MDLFormer, is employed to accomplish this goal. An encoder-decoder network is a traditional architecture for dense prediction tasks, producing output results that are identical in size to the inputs. This article uses the proposed GCST as the encoder and FPN as the basis for the decoder. The Swin Transformer in GCST is Swin-B, which contains 2, 2, 18, and 2 Swin Transformer Blocks in its four levels. The GCB is added in stages three and four. The encoder's primary responsibility is extracting high-level feature vectors by obtaining context information through convolution, activation, and normalizing algorithms.

.

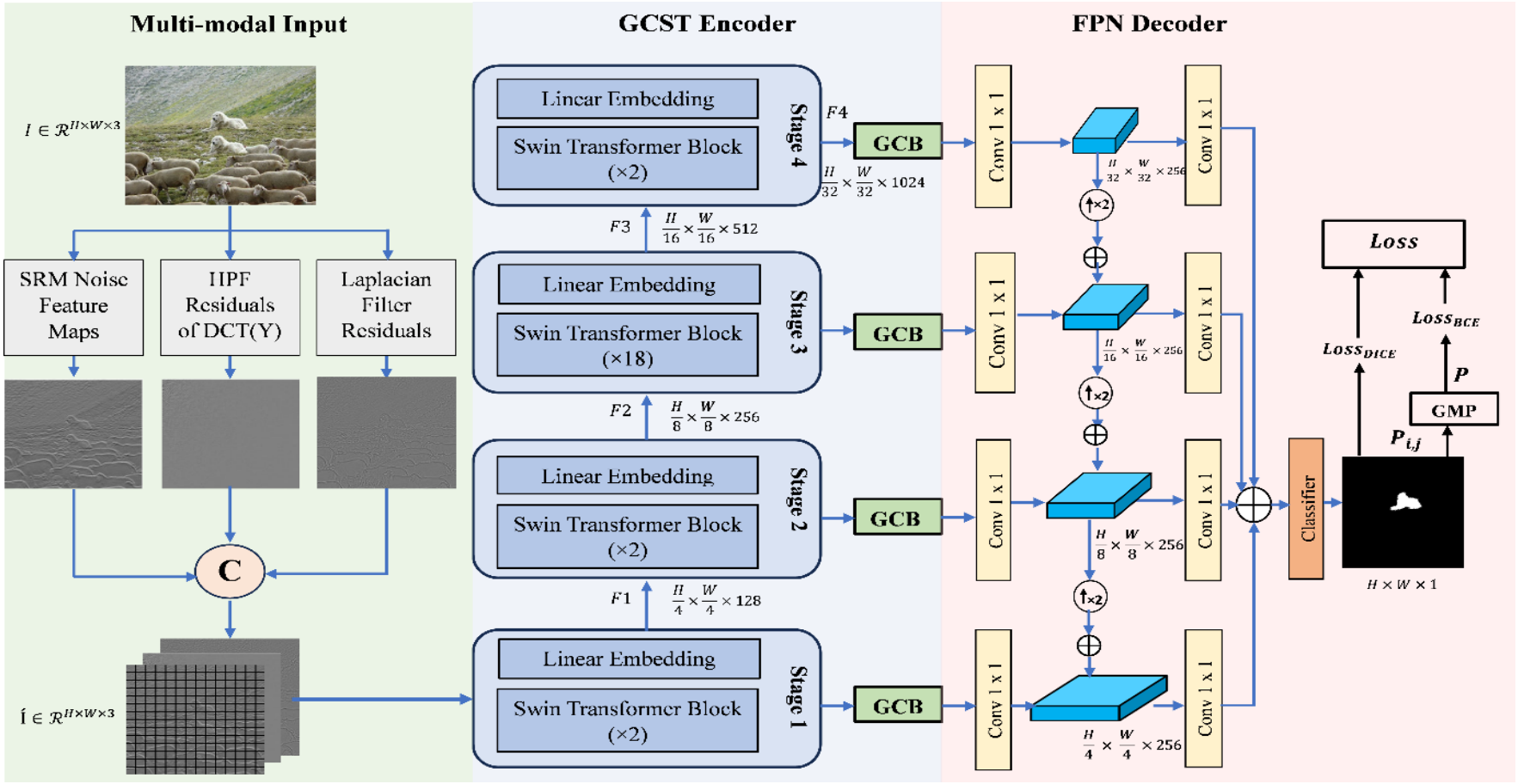


Fig. 4.1: Overview of proposed MDLFormer model architecture for image manipulation detection and localization. A detailed discussion of the three regions, i.e., multi-modal input (green region), GCST encoder (blue region) and FPN decoder (pink region).

Nonetheless, the decoder concatenates (adds) two inputs (one from the encoder's symmetrical layer and the other from its preceding layer) to determine the spatial information placement (up sampling). Proposed MDLFormer structure has a three-step pipeline, as seen in *Fig. 4.1*. The feature maps are displayed by their dimensions, such as H×W×C $\otimes$ represents matrix multiplication while $\oplus$ represents broadcast element-wise addition. Conv 1×1 operation is used to change the feature map dimension to the desired feature map dimension. Three different types of information are formed as the encoder input in the first step, known as multi-modal input. Although noise inconsistency is the most used modality, ablation research demonstrates that different modalities improve performance. The second phase consists of a GCST encoder, which extracts discriminative features to help classify between manipulated and authentic images. The final stage, manipulation localization, makes pixel-level localization possible with the FPN decoder. The following are MDLFormer's detailed processing steps.

4.1.2.1 Multi-modal Input

Typical image manipulations, which are usually undetectable to human eyes, may result in some changes between a pristine portion and a tampered part by splicing, removal, copy-move, and other post-processing operations to hide artifacts. To enable the encoder to learn forgery traces instead of image contents for an image $I \in R^{H\times W\times 3}$, three different sources of information are used as input. Among these inputs are: 1) Noise feature maps $I_1 \in R^{H\times W\times 3}$, is obtained to identify the noise discrepancy between genuine and tampered regions. The noise characteristics were taken from an SRM filter layer [131]. The idea behind the use of the SRM filter layer is that noise features between the source and target images are unlikely to match when an object is removed from one and pasted into the other (the target). The RGB image is converted into the noise domain to use the local noise features as input for the encoder. Noise features can be extracted from an image using various methods. SRM filter kernels is used to create the noise features and use them as the input channel to the GCST encoder based on previous work on SRM for manipulation classification [131]. SRM filters are employed to extract the local noise features from RGB images as the input to the GCST

encoder, inspired by recent advances in SRM features from image forensics [131]. 2) High-pass filtering residuals obtained from the DCT of the Y channel of the YCbCr color space $I_2 \in R^{H\times W}$, which considers high-frequency information caused by tampering and post-processing operations. Since research [132] has demonstrated that the YCbCr color system is known to be more susceptible to manipulation artifacts, first transformed the input RGB image into a YCbCr color space. The DCT coefficients of the Y -component are then used to learn forgery traces since they represent luminance information and comprise most of the image information. However, there isn't much difference between genuine and fake images in RGB space. Furthermore, this distinction between real and fake can still be seen in the frequency domain, particularly in the high-frequency region, even though it is difficult for the human eye to detect. A neural network can detect minute variations in the frequency domain even with low-quality images. Image forensics relies on capturing the evidence of tampering actions to detect and locate manipulation in an image. It is not easy to extract discriminative characteristics from the pixel domain of an image and directly record the inpainting traces because deep inpainting results in visually indistinguishable image contents. High-pass filtering of an image to suppress its contents and extract residuals is a standard procedure in many forensic techniques for gathering tampering traces [133][134]. Motivated by these efforts, apply a High-pass filter on the DCT of the Y channel of the YCbCr color space that can improve the quality of tampering traces. 3) LFR maps $I_3 \in R^{H\times W}$, as obtained in [82], Laplacian filter residual highlights the areas of rapid intensity change, which helps identify the discontinuity caused by tampering. LFR is obtained by first converting the input image to a grayscale image and then applying a Laplacian filter mask of size 3×3 to identify inconsistencies that may indicate manipulation after [82].

Further, these three input features are concatenated as $\acute{I} = [\, I_1;\, I_2;\, I_3] \in \mathcal{R}^{H\times W\times 3}$, as shown by the Eqn. below

$$\acute{I} = Concat(\, I_1, I_2, I_3) \tag{4.1}$$

4.1.2.2 Encoder

The main aim of the encoder is to learn the manipulation traces left behind by different manipulations. The encoder consists of a proposed Global Context Swin Transformer, namely GCST, to downsample and encode the pre-processed multi-modal input image into multi-scale high-dimensional feature maps, which are required inputs for the decoder.

Global Context Swin Transformer (GCST): Swin Transformer, a shifted window transformer is a hierarchical ViT architecture designed to increase the performance and efficiency of using transformers in computer vision tasks like object identification, image segmentation, and image classification [89]. Compared to ViT [13], Swin Transformer [89] is a hierarchical architecture that handles dense prediction issues and lowers computational complexity. In particular, it calculates self-attention in non-overlapping windows with small-scale sizes. Furthermore, the window partitions in succeeding layers differ to encode contextual information. As a result, local self-attention modules transform the long-range information throughout the network, making it a suitable choice for image segmentation tasks. Swin Transformer has four hierarchical stages, each generating tokens at different scales. Given an input of size H×W, the image is divided into non-overlapping patches and these are mapped into a vector of dimension C via a linear embedding. Tokens for $\frac{H}{4} \times \frac{W}{4}$; $\frac{H}{8} \times \frac{W}{8}$; $\frac{H}{16} \times \frac{W}{16}$; and $\frac{H}{32} \times \frac{W}{32}$ are produced, correspondingly, by stages 1, 2, 3, and 4. Each stage contains Patch Embedding followed by a few Swin Transformer Blocks. A Swin Transformer computes local self-attention using the Shifted Window Multi-Head Self-Attention (SW-MSA) instead of the MSA used in ViT.

Despite using a shifted-widow approach for the sequential layers of a hierarchical architecture and self-attention mechanism, Swin Transformers still have poor encoding for large-scale spatial contextual information, local window constraints and slow global information integration. Solution to this issue is to increase the corresponding field for spatial images using a Global Context Swin Transformer, or "GCST" for short. GCST makes it possible to encode long-range contextual information on different scales efficiently. More specifically, the Swin Transformer's

many stages are designed to accommodate the GCB. Adding GCB to the Swin Transformer helps improve its capacity to capture long-range dependencies and contextual information.

Global Context Block (GCB) directly enhances the model's ability to aggregate global information in a single step without relying on hierarchical progression. The GCB enhances the model's ability to detect and localize subtle image manipulations by explicitly integrating the global context information early in the network. This integration of GCB provides better consistency, coherence and sensitivity to the nuances of image manipulation, making it an essential enhancement for tasks like IMDL. The GCB is added within the Swin Transformer blocks. The GCB uses global average pooling to capture global context and transforms it using convolutions. This transformed context is added to the original input to enhance the feature representation. The stages of the Swin Transformer are processed as usual. The feature map is processed and enhanced with global context by the GCB. The feature map is reshaped to its original dimensions for the next stage.

Fig. 4.2 illustrates the detailed architecture of the GCB and is formulated as

$$z_i = x_i + W_v^2\, ReLU(LN(W_v^1 \sum_{j=1}^{N} \frac{\exp(W_k x_j)}{\sum_{m=1}^{N} \exp(W_k x_m)} x_j)) \tag{4.2}$$

Where, x_i represents the input feature at the spatial location i in the feature map. x_i is a vector of length C, where C is the number of channels in the input feature map. Denote $x = \{x_i\}_{i=1}^{N}$ as the input feature map of an image, where $N = H \cdot W$ is the number of positions in the feature map and z is the output feature after applying the GCB. z_i has the same dimension as the input feature map at location i. W_v and W_k denote bottleneck transform learnable weight matrices (e.g., 1×1 convolution). W_v^1 reduces the dimensionality of the global context vector in the first step of the bottleneck transform from C to C'. W_v^1 helps to reduce the computational cost and capture important features more efficiently. W_k represents the attention score weight. It is used to project the feature vector x_j into a scalar value (attention score = $W_k\, x_j$),

which is later used to compute the SoftMax weights. Subsequently, global attention pooling operation $Y = \sum_{j=1}^{N} \frac{\exp(W_k x_j)}{\sum_{m=1}^{N} \exp(W_k x_m)} x_j$ is performed to have a global context

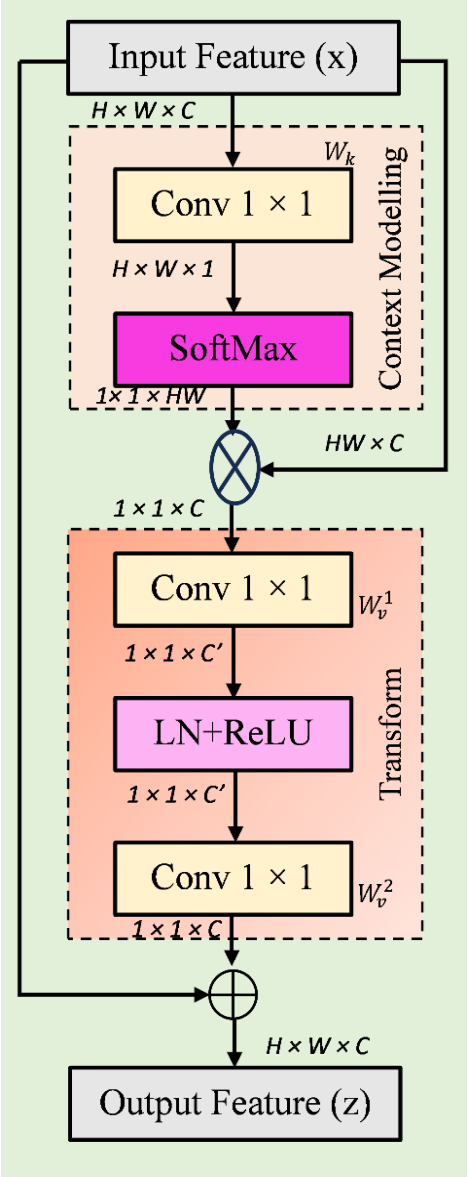


Fig. 4.2: Architecture of the Global Context Block (GCB).

vector. It computes a weighted sum of all input features x_j, with weights determined by the attention scores. A non-linear activation function ReLU is applied to the output of W_v^1 and Layer Normalization (LN) is used to normalize the intermediate feature vector to improve the network's stability during training. W_v^2 projects the transformed global context back to the original feature map's dimension from C' to C. The weights

for global attention pooling are given by $a_j = \frac{\exp(W_k x_j)}{\sum_{m=1}^{N} \exp(W_k x_m)}$ and $\mathcal{B}(\cdot) = W_v^2\, ReLU(LN(W_v^1(\cdot)))$ represents the bottleneck transform. GCB consists of three steps: (1) global attention pooling for context modeling, (2) bottleneck transform to encapsulate channel-wise dependencies and (3) broadcast element-wise addition for feature fusion.

4.1.2.3 Decoder

This study also uses FPN [89] to fuse different-scale features. Two key goals of FPN are to acquire multi-scale contextual information and to expand the receptive field. In a standard deep convolutional network, a pooling layer is added along with the activation function and convolutional layer. Following the pooling layer, the size and computation quantity of the feature map will typically decrease. Field extension is essential. However, the smaller feature map's lower spatial resolution will significantly lose spatial semantic information for the IMDL challenge. Not only does FPN enhance the detection and localization of large target regions, but it also expands the receptive field without compromising the spatial resolution. FPN is used to take advantage of multi-scale context through multi-level feature map fusion to produce a fine-grained localization result. More specifically, the four-head FPN is developed for the four stages of GCST. The decoder utilizes the coarse feature map as input and conducts sampling and convolution operations to generate dense feature maps that can be used for pixel-wise classification into manipulated and authentic images. Furthermore, as depicted in the pink box in *Fig. 4.1*, the {F1, F2, F3, F4} is sent to the FPN decoder and the classifier to generate the final prediction, *P*. To be more precise, start by applying a 1×1 convolutional layer to every feature map. Subsequently, the smaller feature map is sampled twice, and element-wise summation is performed to fuse them. Again, using a 1×1 convolutional layer, these features are fused by the element-wise summation and given to the classifier, where the classifier is made up of a convolutional layer with 3×3 kernel, batch normalization, SoftMax activation function and up-sampling to transform the feature map to match the *GT*, to produce prediction binary mask *B*.

4.1.2.4 Loss Function

The MDLFormer network parameters are learned during training by minimizing a loss function computed between the ground truth ($GT_{i,j}$) and the predicted binary masks ($B_{i,j}$). Consider two types of loss, each with its own target, i.e., a pixel-level loss to improve the model's sensitivity for pixel-level manipulation localization task and an image-level loss to improve the model's specificity for image-level manipulation detection task. To train the MDLFormer network, employed a linear combination of two loss functions: the conventional binary cross-entropy loss (L_{BCE}) and the Dice Loss (L_{DICE}), given by the following equation,

$$L = \lambda \cdot L_{BCE} + (1 - \lambda) \cdot L_{DICE} \tag{4.3}$$

Where λ is a hyperparameter that balances the two losses, it is set to 0.5 by default.

Pixel-Level Loss: A manipulated image typically contains more authentic pixels than manipulated ones. The traditional cross-entropy loss, calculated as the average of all pixels, will be more biased toward the authentic classes. This results in low performance in classifying manipulated pixels while doing well in classifying authentic pixels. As manipulated pixels are often in the minority in a given image, employ the Dice loss, which was found to be effective for learning stability from unbalanced data:

$$L_{DICE} = 1 - \frac{2 \cdot \sum_{i=1}^{H}\sum_{i=1}^{W} GT_{i,j} \cdot B_{i,j}}{\sum_{i=1}^{H}\sum_{i=1}^{W} (GT_{i,j})^2 + \sum_{i=1}^{H}\sum_{i=1}^{W} (B_{i,j})^2} \tag{4.4}$$

Where $GT_{i,j} \in \{0,1\}$ represents the pixel label value at position (i, j), and $B_{i,j}$ indicates the probability that the pixel at position *(i, j)* is manipulated.

Image-Level Loss: As the two classes at the image-level are more balanced than their counterpart at the pixel-level, utilize the BCE loss, extensively used for image classification, to compute the image-level loss:

$$L_{BCE} = -\sum[GT \cdot \log(B) + (1 - GT)\log(1 - B)] \tag{4.5}$$

Where $GT = \max(\{GT_{i,j}\})$ and $B = GMP(B_{i,j})$. Global Max Pooling (GMP) takes the maximum of $B_{i,j}$ as B, i.e., $B = B_{i^*,j^*}$, with $(i^*, j^*) = argmax_{i,j}(B_{i,j})$.

4.1.3. Experiments

This section consists of a set up of a few experiments in this study to evaluate the proposed MDLFormer framework. This section consists of the dataset and implementation details.

4.1.3.1 Dataset

Pre-trained Dataset: Current standard manipulation datasets do not have enough manipulated images to support deep neural network training. As a result, first, pre-train the MDLFormer network using a synthetic dataset, i.e., DEFACTO [135]. A synthetic dataset is employed to pre-train the model, enabling it to acquire fundamental features and patterns. Following this, the model's performance is assessed using standard benchmark datasets, ensuring a robust assessment of its effectiveness and generalization capabilities. DEFACTO [135] is a recent large-scale dataset with 149k forged images automatically manipulated by copy-move, splicing, and removal. The forged images were created from MSCOCO [136]. Several manipulation techniques, such as copy-move, splicing, and removal, were used to manipulate the images. In accordance with [137], pre-train the model on the DEFACTO [135] in order to enable a head-to-head comparison with SOTA methods. In this work randomly selected 60k manipulated images are used from the DEFACTO dataset. The manipulated images in this dataset resemble genuine forgeries, which helps the model learn various traces corresponding to manipulation. It is important to note that compared to some other research, including PSCC-Net [137] (100k samples) and ObjectFormer [68] (62k samples), the base dataset used in this work had fewer images. Using this synthetic dataset, the proposed network is trained with 90% of the data used for training and 10% for validation. Save the model when the network converges on this dataset to be tested and fine-tuned further on several standard manipulation datasets. *Table 4.1* lists all datasets and their key characteristics.

Standard Datasets To demonstrate the effectiveness of the proposed approach in localizing different types of manipulations, experiments are carried out on the following standard forgery datasets: Columbia [138], COVERAGE [139], CASIA

[140] and NIST16 [141], and IMD20 [142]. To fine-tune MDLFormer, the same training/testing split for Coverage, CASIA, and NIST16 as in [137] for fair comparisons is used. *Table 4.1* summarizes the manipulation types for each standard dataset and the number of images used to train and test the pre-trained and fine-tuned models.

Table 4.1: Dataset training-testing split for the Pre-trained and Fine-tuned models

Dataset	**Pre-trained**		**Fine-Tuned**		**Manipulation Type**
	Train	Test	Train	Test	
DEFACTO [135]	54000	6000	-	-	S, C, Re
Columbia [138]	-	180	-	180	S
Coverage [139]	-	100	75	25	C
CASIA [140]	-	6044	5123	921	S, C
NIST16 [141]	-	564	404	160	S, C, Re
IMD20 [142]	-	2010	-	2010	S, C, Re

"S": Splicing; "C": Copy-move; "Re": Removal; "-": Not applicable

The datasets are described as follows:

- **Columbia** [138] dataset consists of 180 spliced uncompressed images and ground-truth masks are also provided. It is used to evaluate the pre-trained model.
- **Coverage** [139] dataset includes 100 images based on the copy-move technique and ground-truth masks. To fine-tune the model, the dataset is split into 75/25 for training and testing.
- **CASIA** [140] dataset comprises both splicing and copy-move manipulated images of different objects. The tampered locations are carefully picked, and some post-processing techniques, such as filtering and blurring, are used. Ground-truth masks are created by thresholding the difference between modified and original images.

For fine-tuning, utilize 5123 images from CASIA v2.0 for training and 921 images from CASIA v1.0 for testing.

- **NIST16** [141] dataset has 564 manipulated images, including three types of manipulation: copy-move, splicing and content-removal. It is a challenging dataset as manipulated images are post-processed to remove visible traces and the ground-truth masks are provided for evaluation. To fine-tune the model, it is split into 404/160 for training and testing.
- **IMD20** [142] comprises 2010 real-life manipulated mages taken from the internet and includes three types of manipulation: copy-move, splicing and content-removal. It is used to evaluate the pre-trained model. It is used to test the MDLFormer.

4.1.3.2 Implementation Details

PyTorch framework is used to build the proposed approach and all experiments were conducted on Nvidia Quadro K4200 and Nvidia Tesla K40C GPUs. During the training phase, the model is optimized utilizing the Adam optimizer with a batch size of 8. The initial learning rate was set at 10^{-4}. Validations were performed after each epoch, and the model with the highest validation F1-score across all 100 epochs was chosen as the final model and used in the testing step. MDLFormer has 58M parameters and 23G FLOPs, strikes a balance between computational efficiency and detection accuracy.

4.1.4. Results and Discussion

This section presents the results of the proposed method, MDLFormer, which performs both detection and localization for manipulation detection on an image. For fair comparisons, SOTA methods whose source codes are either publicly available or whose pre-trained models are released by the authors are considered and if the codes are not available, then the results are obtained from their papers. Various evaluation metrics assess the model's performance in detecting and localizing tampered regions in manipulated images. A comparison of the proposed method with current SOTA methods on standard datasets like Columbia [138], Coverage [139], CASIA [140],

NIST16 [141] and IMD20 [142] is made. The localization performance of the proposed method under two different settings is shown in Section 4.1.4.1. Subsequently, manipulation detection performance analysis is done in Section 4.1.4.2. Furthermore, the ablation study is done to evaluate the antiablation capability of the proposed method in Section 4.1.4.3. Lastly, robust analysis is done in Section 4.1.4.4.

4.1.4.1 Manipulation Localization Results

Compared with binary image-level manipulation detection tasks, pixel-level manipulation localization is a bit more difficult as it requires the model to capture more refined manipulated artifacts. Following PSCCNet [137], evaluated the model under two settings: 1) Pre-training the model using the synthetic DEFACTO dataset and 2) Fine-tuning the pre-trained model based on the train/test split on the standard datasets. The pre-trained model demonstrates each method's generalization capability, while the fine-tuned model improves localization and reduces domain discrepancies. The stated results for all comparison approaches are based on their original papers or public codes.

4.1.4.1.1 Pre-Trained Model

A comparison of MDLFormer with several SOTA manipulation localization methods, including ManTra-Net [84], SPAN [85], PSCCNet [137], ObjectFormer [68] and TANet [143] is made. *Table 4.2* reports the pixel-level AUC score of various pre-trained models on five distinct standard datasets for image manipulation localization tasks. *Table 4.2* demonstrates the superiority of the MDLFormer in capturing the manipulated features and generalization capability of a variety of standard manipulated datasets. The pre-trained MDLFormer has the best pixel-level AUC performance on Coverage, CASIA v1, NIST16 and IMD20 dataset and second best on the Columbia dataset. On the Coverage, CASIA v1, NIST16 and IMD20 dataset, MDLFormer obtains a performance improvement of about 0.2%, 7.2%, 1.4% and 3.1%, respectively, when compared with TANet [143]. MDLFormer on the Columbia dataset outperforms the ManTraNet [84], SPAN [85], PSCCNet [137] and ObjectFormer [68] but trails TANet [143] by 2.8%. One of the possible reasons might be the significant difference in the data distribution between the DEFACTO and Columbia datasets. The

manipulated regions in the Columbia dataset are quite large compared to those in the synthetic DEFACTO dataset.

Table 4.2: Pixel-level AUC localization performance comparison of pre-trained MDLFormer.

Methods	Data	Columbia	Coverage	CASIA v1	NIST16	IMD20
ManTraNet [84]	64,000	0.824	0.819	0.817	0.79.5	0.748
SPAN [85]	96,000	0.936	0.922	0.797	0.840	0.750
PSCCNet [137]	100,000	0.982	0.847	0.829	0.855	0.806
ObjectFormer [68]	62,000	0.955	0.928	0.843	0.872	0.821
TANet [143]	60,000	**0.987**	0.914	0.853	0.898	0.849
MDLFormer	60,000	0.959	**0.916**	**0.925**	**0.912**	**0.88**

The bold values indicate the best results.

4.1.4.1.2 Fine-Tuned Model:

To account for the difference in visual quality between the synthetic and standard datasets, further fine-tune the pre-trained model on the specific datasets and compare it with other approaches in *Table 4.3*. The pre-trained model's network weights are utilized to initiate the fine-tuned models, which will be trained on the training splits of the Coverage, CASIA, and NIST16 datasets, respectively, using the same strategy as [137]. The best result values are reported from the literature to ensure a fair comparison with other methods. *Table 4.3* shows that MDLFormer performs best on average on all datasets, whether measured by pixel-level AUC or F1 score. MDLFormer achieves a performance gain of 0.3%, 0.6% in AUC and 3.8%, 2.1% in F1 score with respect to the second-best method TANet on Coverage and CASIA v1 dataset, respectively. However, on the NIST16 dataset, MDLFormer trails by 1.2% and 1.5% in AUC and F1 score, respectively, to the second-best method TANet. One of the possible reasons for this might be the wide range of image resolution varying from 500×500 to 5616×3744 in the NIST16 dataset. The significant performance gains can be seen, demonstrating that MDLFormer can capture subtle manipulating artifacts using multi-

modal input, local and global context hierarchical feature representation by the GCST encoder, and the FPN decoder to distinguish between authentic and manipulated pixels and produce a binary predicted mask. The fine-tuned MDLFormer has average pixel-level AUC performance and is either the best or second-best on all datasets, exhibiting outstanding generalization across manipulations.

Table 4.3: Performance comparison of the fine-tuned MDLFormer in pixel-level AUC and F1 score for image manipulation localization task.

Method	**Coverage**		**CASIA v1**		**NIST16**	
	AUC	F1	AUC	F1	AUC	F1
SPAN [85]	0.937	0.558	0.838	0.382	0.961	0.582
MVSS-Net [77]	-	0.824	-	0.753	-	0.737
PSCC-Net [137]	0.941	0.723	0.875	0.554	0.996	0.819
ImageForensicsOSN [144]	-	-	0.873	0.509	0.783	0.332
ObjectFormer [68]	0.957	0.758	0.882	0.579	0.996	0.824
TruFor [72]	-	0.735	-	0.822	-	0.470
TANet [143]	<u>0.978</u>	<u>0.782</u>	0.893	0.614	**0.997**	**0.865**
UnionFormer [69]	0.945	0.720	<u>0.972</u>	<u>0.863</u>	0.881	0.489
MDLFormer	**0.981**	**0.820**	**0.978**	**0.884**	<u>0.985</u>	<u>0.850</u>

The bold values indicate the best results, underlined values indicate the second-best values and "-" indicates that they are unavailable.

Table 4.4: IoU-based localization performance comparison

Methods	**Coverage**	**CASIA v1**	**NIST16**
DFCN [145]	-	-	0.23
ImageForensicsOSN [144]	-	0.358	0.214
Fals-Unet [146]	**0.886**	**0.927**	0.625
ViT-VAE [71]	0.108	0.106	0.171
MSCL-Net [147]	0.625	0.774	<u>0.718</u>
MDLFormer	<u>0.707</u>	<u>0.821</u>	**0.790**

The bold values indicate the best results, underlined values indicate the second-best values and "-" indicates that they are unavailable.

4.1.4.1.3 Qualitative Results:

Qualitative results of the MDLFormer method on image manipulation localization using the COVER, CASIA v1, Columbia and NIST16 are demonstrated in *Fig. 4.3*. The proposed MDLFormer outputs a probability map, which is subsequently thresholded to produce a binary map to localize the forged regions. An experiment with different thresholds within the range of $0.2 - 0.8$ was done and found no discernible difference in the results obtained. This is due to the fact that the probability values associated with forged regions are very close to 1, while the probability values associated with authentic pixels are very close to 0, or typically below 0.01. Therefore, the mid-value, 0.5, has been employed as the threshold for all the experiments in this paper. In *Fig. 4.3*, column 1 corresponds to the pristine image for each image, column 2 corresponds to the manipulated image, column 3 corresponds to the ground truth mask, and column 4 shows the predicted mask. As illustrated in *Fig. 4.3*, the method localizes the manipulated regions accurately. Furthermore, the MDLFormer exhibits less sensitivity to variation in scale. Effective localization is possible for both large (e.g., the fourth row in *Fig. 4.3*) and small (e.g., the fifth row in *Fig. 4.3*) manipulations.

4.1.4.2 Manipulation Detection Results

To analyze the image-level detection performance, a comparison is made between MDLFormer and the SOTA methods: MVSS-Net [77], GP-Net [74], UnionFormer [69] and MSCL-Net [147], using two commonly used metrics (image-level AUC and F1 score). *Table 4.5* shows the AUC and F1 scores for quantitative manipulation detection results. The results illustrate that MDLFormer performs best on the image-level AUC and F1 scores on most datasets, except for Columbia datasets.

Fig. 4.3: Qualitative localization performance of the proposed MDLFormer model against different manipulation techniques such as copy-move (first and fourth row), splicing (second and third row) and removal (fifth and sixth row) on CASIA v1, Columbia, Coverage, IMD20 and NIST16 standard dataset. From left to right, the first column is the authentic image, the second is the manipulated image, the third is the ground truth, and the fourth is the predicted binary mask.

MDLFormer's AUC score on the Columbia dataset is 1.2 % lower than UnionFormer's. It may be because the Columbia dataset is only used for testing, not for training. It is worth noting that the F1 score of MDLFormer is superior to all other models except for the Columbia dataset. These results demonstrate that MDLFormer performs well in image-level image manipulation detection.

Table 4.5: Image manipulation detection performance using image-level AUC and F1score

Method	**Columbia**		**Coverage**		**CASIA v1**		**NIST16**	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1
MVSS-Net [77]	0.980	0.802	0.731	0.244	0.937	0.758	-	-
GP-Net [74]	-	-	-	-	0.887	0.781	0.922	0.848
UnionFormer [69]	**0.998**	-	0.783	-	0.951	-	0.793	-
MSCL-Net [147]	-	0.818	-	0.724	-	0.901	-	0.875
MDLFormer	<u>0.986</u>	**0.827**	**0.989**	**0.853**	**0.983**	**0.910**	**0.988**	**0.937**

The bold values indicate the best results, underlined values indicate the second-best values, and "-" indicates that they are unavailable.

4.1.4.3 Ablation Studies

In this subsection, experiments are performed to investigate each component's effect in MDLFormer. For instance, MDLFormer with and without GCB and multi-modal input can be replaced by single-modality input or various input combinations. *Table 4.6* shows the results for the NIST16 dataset. Firstly, by comparing Variant 1 to Variant 6, which consists of different input combinations with the MDLFormer, the results demonstrate that the MDLFormer is better than all the first six variants. Secondly, comparing Variant 7 to MDLFormer, respectively, the results show that the MDLFormer with GCB-based Swin Transformer encoder remarkably enhances the performance in terms of AUC, F1 and IoU compared to MDLFormer without GCB-based Swin Transformer encoder (Variant 7). It has been discovered that even if Swin

Transformer (ST) uses a shifted window to increase the corresponding fields at various levels, ST + FPN still performs poorly. This operation has minimal impact on context information encoding. Thus GCB is included in the multiple stages of Swin Transformer. The features in the Transformer are rearranged like CNN's feature map after a certain level. Ultimately, the feature map is loaded into the Transformer's subsequent stage while maintaining its original size. GCST can learn contextual information more efficiently and has a bigger receptive field than the only Swin Transformer as an encoder. Lastly, by comparing Variant 1, 2, 3 and MDLFormer, the results demonstrate that while input as in the noise inconsistency (Variant 1) plays a major role, input as in the high-pass filter of DCT residual inconsistency (Variant 2) and input as in the Laplacian edge discontinuity (Variant 3), can assist the network to explore complementary tampering traces but combination of all these inputs significantly improves the performance of the network in all the three metrics AUC, F1 and IoU. The method's GCB module is designed to extract global information-based forgery features for manipulation localization and detection.

Table 4.6: Ablation study results on DEFACTO datasets. Pixel-level AUC and Image-level AUC values are reported.

Variants	**Pixel-level**		**Image -level**	
	AUC	F1	AUC	F1
MDLFormer with input I_1	0.836	0.642	0.880	0.728
MDLFormer with input I_2	0.735	0.539	0.836	0.751
MDLFormer with input I_3	0.825	0.628	0.857	0.766
MDLFormer with input $I_1 + I_2$	0.881	0.782	0.905	0.873
MDLFormer with input $I_1 + I_3$	0.928	0.814	0.944	0.836
MDLFormer with input $I_2 + I_3$	0.863	0.761	0.927	0.850
MDLFormer with I' and w/o GCB	0.934	0.880	0.956	0.874
MDLFormer	**0.989**	**0.930**	**0.993**	**0.915**

The bold values indicate the best results.

Moreover, to illustrate the effectiveness of GCB, it is removed from the MDLFormer and evaluate the tampering localization performance on the NIST16 dataset. The quantitative results are listed in *Table 4.6*. It can be observed that without GCB, the AUC scores decrease by 11.4%, the F1 score decreases by 7.3% and IoU decreases by 15.8% on the NIST16 dataset. The performance degradation validates that the use of GCB effectively improves the performance of the MDLFormer.

4.1.4.4 Robustness Analysis

In real-world scenarios, manipulated images often suffer from non-malicious manipulations or post-processing distortion operations such as noise, resizing, blurring and JPEG compression, which impacts the manipulation detection and localization. Consequently, in this subsection, the robustness of the proposed method against various commonly used distortion settings is evaluated. To further demonstrate the robustness of the MDLFormer, it is subjected to the images from the NIST16 dataset to various post-processing distortion methods. These methods include image resizing with different resizing factors $s = \{0.78, 0.50, 0.25\}$, Gaussian noise with a standard deviation $\sigma = \{3, 5, 11\}$, JPEG compression with a quality factor $q = \{100, 50, 25\}$ and Gaussian blur with a kernel size $k = \{3, 7, 15\}$. *Table 4.7* shows the robust performance of the MDLFormer measured by the F1-score and AUC against the various distortion parameters used. The model performs well under a variety of distortions.

Table 4.7 shows that MDLFormer is less affected by noise, resize and JPEG compression, while it is more sensitive to Gaussian blurring distortion operations. Especially on compressed images, the F1-score is only 0.35% lower than without the distortion when the quality factor is 100 and 0.82% lower than without the distortion when the quality factor is 50. The MDLFormer demonstrates robustness against multiple distortion operations.

Table 4.7: Robustness analysis of MDLFormer for image manipulation localization using AUC and F1 as the evaluation metric under various distortion scenarios on the NIST16 dataset

Distortion	AUC	F1
w/o distortion	0.985	0.850
Resize (s =0.78)	0.967	0.843
Resize (s =0.50)	0.944	0.822
Resize (s =0.25)	0.938	0.846
Gaussian noise ($\sigma = 3$)	0.968	0.838
Gaussian noise ($\sigma = 5$)	0.921	0.788
Gaussian noise ($\sigma = 11$)	0.870	0.756
Gaussian Blur ($k = 3$)	0.977	0.849
Gaussian Blur ($k = 7$)	0.963	0.832
Gaussian Blur ($k = 15$)	0.864	0.801
JPEG Compression ($q = 100$)	0.981	0.844
JPEG Compression ($q = 50$)	0.967	0.832
JPEG Compression ($q = 25$)	0.948	0.801

4.2 LFRViT Method for Multiple Image Forgery Detection

4.2.1 Introduction

The field of research and technology known as "multiple image forgery detection" is devoted to detecting situations in which multiple images are combined or altered to produce an inaccurate or misleading representation. Multiple-image forgery detection exposes instances of manipulation or tampering by examining relationships and inconsistencies between multiple images, in contrast to traditional single-image

forgery detection, which focuses on finding a single tampering operation within an image [1].

Effective techniques to identify manipulated or tampered images are more important than ever due to the spread of social media sites, online news sources, and digital archives. These kinds of images can be used to disseminate false information, sway public opinion, or trick people or institutions. Thus, creating efficient methods and algorithms for multiple image forgery detection has become essential research in the larger digital forensics and image analysis field. By applying advances in machine learning, computer vision, and signal processing, scholars and practitioners aim to improve the veracity and authenticity of visual content on digital platforms [148].

This study considers four different types of image tampering operations. These tampering operations are applied to each original image to have a manipulated image. The four different types of tampering operations are: AWGN, resampling, median filtering, and gaussian blurring. *Fig. 4.4*, shows the different tampering operations performed over the original image, taken from the RAISE dataset.

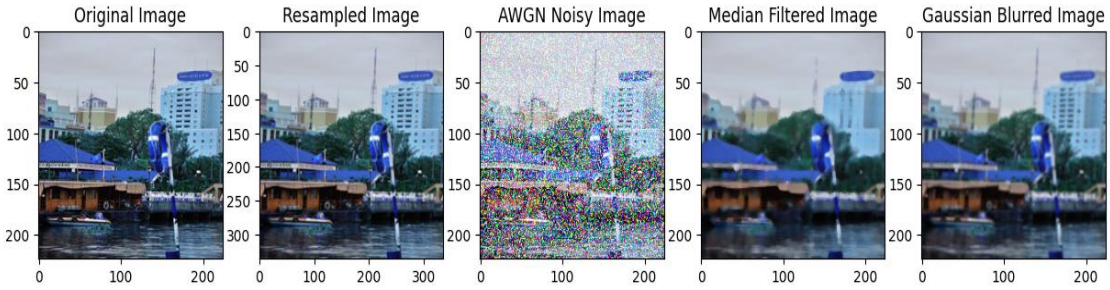


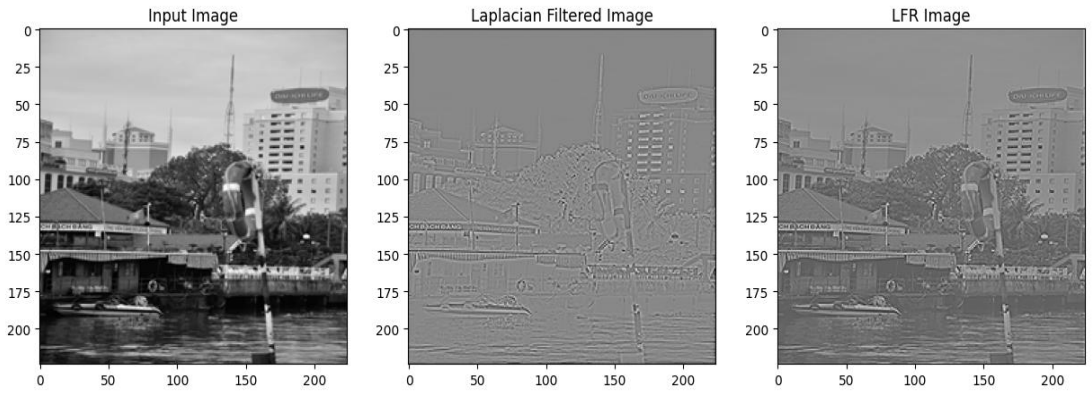
Fig. 4.4: From RAISE database, ***(a)*** *original image and different operations are performed on this original image,* ***(b)*** *resampled image with a scaling factor of 1.5,* ***(c)*** *AWGN noisy image with standard deviation of 2,* ***(d)*** *median filtered image with a 5 × 5 kernel size and* ***(e)*** *Gaussian blurred image with 5 × 5 kernel and σ = 1.1.*

This study introduces a novel universal method for detecting image editing, which has the ability to autonomously acquire knowledge about the traces left by the editing operation. In order to achieve this, ViT [13] is used. ViTs have recently led to

significant progress in image recognition by enabling dynamic learning of classification features instead of relying on hand-crafted features. A deep-learning approach is presented that turns local image regions into masked features using patch-level learnable masking. The ViT model receives these masked features to identify any global discrepancies between the local masked features produced. These generated features are highly valuable in the manipulation detection area because they can efficiently learn the residual artifacts in manipulated images, whether local or global. From now on, refer to this model as LFRViT. *Fig. 4.6* shows the entire architecture of the proposed model LFRViT.

4.2.2 LFRViT Model

This study proposes an LFRViT, a deep learning-based classification model that distinguishes between real and altered images produced by various tampering operations. In the proposed approach, the tampering activities are investigated to discern between real and altered images, suppressing the image content using the LFR [82]. The residual serves as an input for the ViT architecture, which detects and classifies altered images based on their traces. *Fig. 4.5*, shows the LFR image corresponding to the input image. The image undergoes some pre-processing before being fed directly as input to the ViT. In pre-processing, first, resize the images to 224×224. A Laplacian filter mask of 3×3 is utilized, as indicated by (1).



(a) *(b)* *(c)*

Fig. 4.5: Illustration of the Laplacian filter-based CNN layer output, (a) input image, (b) Laplacian filtered image obtained via (2) and (c) LFR image obtained via (3).

$$Laplacian\ filter\ mask = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{4.6}$$

The Laplacian of the image is further obtained via (4.7).

$$L_{i,j} = Laplacian(X_{i,j}) \tag{4.7}$$

Where $X_{i,j}$ represents the intensity of the pixel at the i[th] and j[th] pixels. Finally, subtracting the image from the Laplacian of the image as provided by (4.7), yields the LFR. As stated, the LFR is given by:

$$LFR_{i,j} = L_{i,j} - X_{i,j} \tag{4.8}$$

The solution to this problem is addressed by leveraging the inconsistent occurrence of nearly undetectable residual artifacts in altered images. A model is developed that can detect the presence of these artifacts by identifying the inconsistencies within the manipulated image. To do this, initially apply the Laplacian filter to the input image, resulting in the LFR image. Subsequently, this LFR image will be utilized as input for the ViT model. The ViT model initially divides the image into patches of a predetermined size. A patch size of 16×16 is taken. Every patch is considered as "token," similar to how words are processed in tasks involving natural language processing. Each patch is encoded into a vector representation using an embedding layer known as token embeddings. The embeddings preserve spatial information of the patches. Positional encodings are incorporated into the token embeddings of the Vision Transformer to compensate for its lack of innate understanding of spatial relationships between patches, an ability that CNNs possess through convolutions. These encodings convey information about the position of each patch inside the image. The token embeddings, in addition to positional encodings, are subsequently fed into a Transformer encoder. This encoder comprises several layers of self-attention mechanisms, which are then followed by feedforward neural networks. The self-attention mechanism enables the model to capture interdependencies among various patches in the image. Ultimately, the result of the Transformer encoder is sent

into a classification head, which usually comprises one or more fully connected layers. This classification head generates the final output probabilities for various classes [13]. *Fig. 4.6*, provides a visual representation of the operational concept of the proposed model LFRViT.

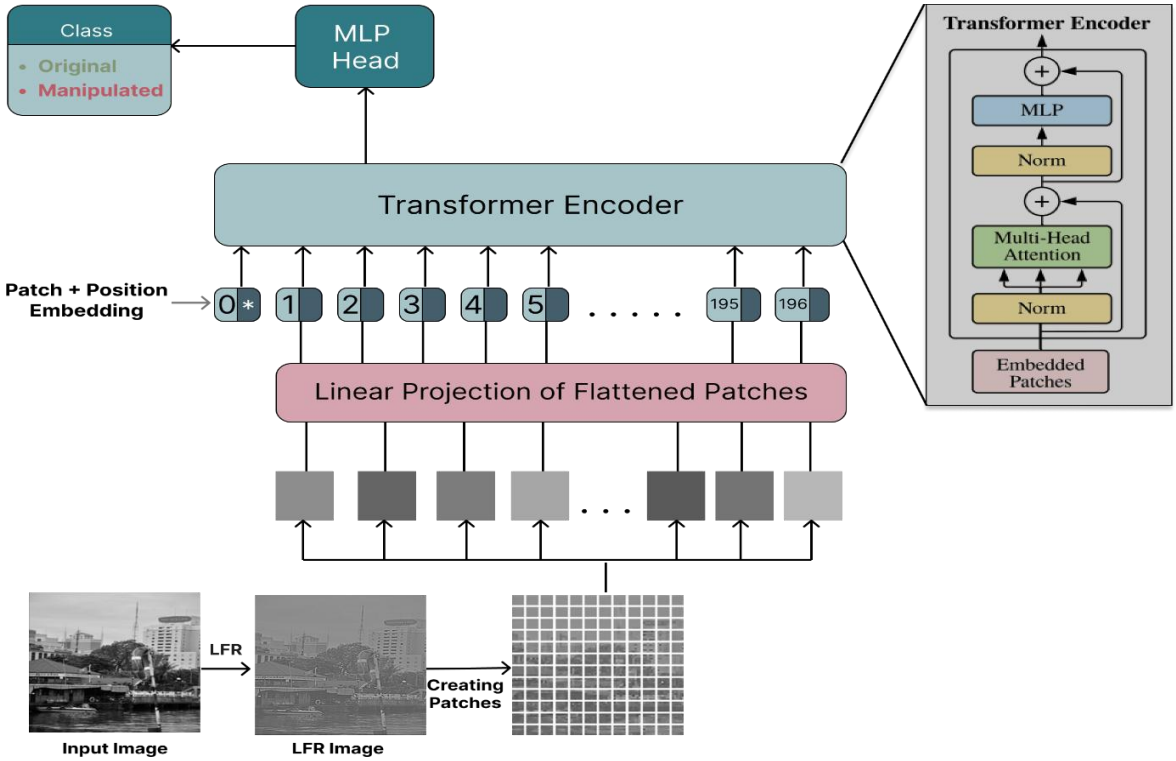


Fig. 4.6: Architecture of the proposed LFRViT model.

A comprehensive set of experiments is conducted to evaluate the effectiveness of the proposed method in identifying different image-manipulating activities. Standard image datasets, such as Uncompressed Color Image Datasets (UCID) [149], Break Our Stenographic System (BOSSBase) [130], raw images dataset for digital image forensics (RAISE) [150], and the Dresden image dataset (DID) [151], were utilized to create different training and testing sets for different experiments. A total of 33593 images were assembled using 1388 images from UCID with dimensions of 512×386, 9074 images from BOSSBase with dimensions of 512 ×512, 15025 images from DID with varying dimensions, and 8156 images from the RAISE dataset. Several training and testing datasets were then created using the original image set as a foundation. Four distinct types of manipulation operations are applied to each image:

AWGN, resampling, median filtering, and gaussian blurring. Then produced a set of altered images. To achieve this, a collection of unaltered images is subjected to the individual manipulation operations such as AWGN with standard deviation of 2, resampling using a scaling factor of 1.5, median filtering with a 5×5 kernel and gaussian blurring with a 5×5 kernel and a standard deviation of $\sigma = 1.1$.

4.2.3 Result

In this section, manipulation detection results of the proposed method is presented for binary and multi-class classification. Different evaluation metrics assess the model's accuracy in identifying manipulated images. The study utilizes commonly used criteria, including accuracy, precision, recall, and F1 Score, to quantify the effectiveness of classifying manipulated images.

4.2.3.1 Binary Classification

In the initial set of experiments, the proposed model is trained separately for each of the four manipulations to detect them individually. With the identical architecture described in Section 3, each model corresponds to a binary classifier that can identify a single kind of potential image processing. The original and manipulated images correspond to the two neurons that comprise the output layer. Decisions are made by selecting the class corresponding to the highest activated neuron. Training set for each type of forgery is constructed using 26875 unaltered photos and the corresponding manipulated images. Similarly, to create the testing data for each type of forgery, 6718 original images were selected, along with the corresponding manipulated ones. In total 53750 training images and 13436 testing images are used for each binary classification.

The performance of the proposed model for binary classification to identify the underlying manipulating operations is compiled in *Table 4.8*. This table shows that the proposed approach can differentiate between original and manipulated images with a minimum of 99.32% accuracy, a minimum value of 0.96 for precision and recall and a minimum value of 0.95 for F1-Score.

4.2.3.2 Multi-class Classification

In the second set of the experiment, the model was trained for multiclass classification to identify several forms of image manipulation, such as median filtering, Gaussian blurring, AWGN, and resampling, compared to real images. As in the first set of experiments, a choice is made by selecting the class corresponding to the neuron with the highest activation level. A 26875 unaltered images and their four corresponding tampered images are selected to create the training set. Similarly, 6718 original images and the four corresponding tampered images are used for the testing data. For testing 33590 images and for training, 134375 images have been used. A summary of the simulation results is shown in *Table 4.9*. The proposed model detects the four main types of forgeries with a minimum accuracy of 99.28%. This confusion matrix shows us how well the model can identify each alteration.

There are multiple reasons why these results are significant. First, they demonstrate how the model, which can be trained to identify multiple manipulations without changing its architecture, represents a universal approach to manipulation detection. Perhaps most surprisingly, the model can be taught to learn detection features for every manipulation without human assistance automatically. This implies that the model may learn to detect new manipulations as they are considered or created, eliminating the requirement for a human expert to define detection features.

Table 4.8: LFRViT performance as a binary classifier

Evaluation Parameters	**Tampering operations**			
	Resampling	**AWGN**	**Median Filtering**	**Gaussian Blurring**
Accuracy (A)	99.78%	99.62%	99.32%	99.47%
Precision	0.99	0.98	0.96	0.97
Recall	0.98	0.99	0.94	0.96
F1-Score	0.98	0.98	0.95	0.95

Table 4.9: Confusion Matrix of LFRViT as a Multi-Class Classifier

	Original	***Resampling***	***AWGN***	***Median Filtering***	***Gaussian Blurring***
Original	**99.61**	1.37	0.51	1.70	1.52
Resampling	1.20	**99.78**	0.18	0.57	0.68
AWGN	0.07	0.09	**99.62**	1.29	1.01
Median Filtering	0.26	0.31	0.51	**99.28**	1.59
Gaussian Blurring	0.72	0.83	0.16	1.30	**99.47**

4.3 Summary

This chapter explores methods for detecting and localizing multiple forgeries. This chapter presents MDLFormer, a Multi-modal Global Context-based Swin Transformer tailored for image manipulation detection and localization tasks. MDLFormer consists of three regions: multi-modal input, GCST encoder and FPN decoder. The multi-modal input from the SRM filter layer, the high-pass filter of DCT coefficients of the Y channel of the YCbCr color space and the Laplacian residual enables the model to capture noise inconsistencies-based features between manipulated and authentic regions. The GCST encoder is a global context-based Swin Transformer that aims to provide features of spatial characteristics of manipulated regions. Finally, the FPN decoder learns spatial mapping to produce the binary predicted mask. Extensive experiments are performed to test the performance of the MDLFormer on various standard datasets such as CASIA, Columbia, Coverage, IMD20 and NIST16. The results have demonstrated the superiority of the MDLFormer model against SOTA methods regions in terms of F1 score, AUC and IoU for detecting manipulated images and localizing the manipulated. Despite exhibiting outstanding performance, the proposed approach still has certain drawbacks. For instance, certain images are still

not well localized, particularly on the NIST'16 dataset. Moreover, MDLFormer should be strengthened to fight against online social networks based shared manipulations.

Additionally, the chapter introduces LFRViT, a Laplacian Filter Residual-based ViT specifically developed for Multiple Image Forgery Detection. The method introduces a novel convolutional layer that utilizes a Laplace filter mask to recognize multiple image manipulations. This mask generates Laplacian filter residuals specifically designed to suppress the image's content and enables the ViT to better capture subtle forgery patterns and irregularities. Results conclusively demonstrated that the LFRViT model can autonomously acquire the ability to identify a variety of image manipulations.

MDLFormer and LFRViT are highly effective for detecting multiple traditional image manipulation types, including copy-move, splicing, and inpainting, but they are not suitable for deepfake-based manipulation detection. In the future, methods for deepfake detection will be investigated.

Chapter 5

Deepfake Face Manipulation Detection

This chapter presents a framework based on hybrid learning and kernel principal component analysis for deepfake face manipulation detection. The effectiveness of the proposed approach is explained and validated through experiments on standard datasets and state-of-the-art comparisons of obtained results.

5.1 Introduction

Face manipulation is altering a face's features in images or videos to produce artistic, cosmetic, or misleading effects. It can entail a variety of adjustments, ranging from minor improvements to significant changes. Face manipulation can be divided into four primary categories: exchanging identities, swapping expressions, manipulating attributes, and generating synthetic faces. Facial identity manipulation is the process of replacing one person's face with another. The most widely used methods for manipulating facial identities are FaceSwap[1] and DeepFakes[2]. Facial expression manipulation replaces one person's facial expressions with another while preserving the facial identity. Face2Face [152] and NeuralTextures [153] are the two most popular methods for manipulating facial expressions. While the DeepFakes and NeuralTextures approaches are based on deep learning techniques, the FaceSwap and Face2Face approaches are based on computer graphics techniques. Face attribute-manipulated images identify alterations to specific facial features or characteristics such as gender, age, hair, beard, and glasses. The two most popular methods used to generate attribute-manipulated images are FaceAPP[3] and StarGAN [154]. Computer graphics, deep learning, or other digital methods artificially produce synthetic facial images. These are not photographs of actual people; rather, they are the result of

[1] Faceswap:https://github.com/MarekKowalski/FaceSwap.
[2] Deepfakes:https://github.com/deepfakes/faceswap.
[3] FaceApp:https://faceapp.com/app.

models or algorithms. The popular methods used to generate all synthesized faces are PGGAN [155] and Style-GAN [156]. Face manipulation techniques can be used for more controversial applications, like producing misleading content known as "DeepFakes," or for legitimate purposes, such as retouching photographs for aesthetic reasons. "DeepFakes" encompasses digitally fabricated content created using deep learning techniques. It gained notable prominence in late 2017 when a Reddit user known as "DeepFake" unveiled the development of a machine-learning algorithm capable of replacing celebrities' faces in explicit videos [157]. The harmful consequences of deepfake are rooted in its potential for malicious applications, including generating deceptive pornography, spreading false information, perpetuating hoaxes, and facilitating financial fraud [158]. Nevertheless, like any technological advancement, deepfake can also be exploited, compromising personal integrity and media production to disrupt elections and fuel political instability. As a result, digital media, including news broadcasts, online video clips, and live streams, are experiencing trust issues [12]. Therefore, ensuring the authenticity of these videos or images is critical.

Face manipulation detection involves various methods, ranging from conventional image analysis to advanced deep learning methods. The increasing advances in deep learning have made it difficult to detect face manipulation. Certain artifacts are used by some facial manipulation detection methods as an indication of manipulation [159], whereas some have employed deep neural networks that use general artifacts to indicate manipulation for facial manipulation detection [92]. Most of the work is not robust enough to withstand simple attacks like resizing, compression, or additive noise [160]. On the other hand, real-world situations frequently involve these kinds of manipulations. Also, the existing methods based on deep neural networks used for face manipulation detection are very complex and require large computational resources, and most of the features among them are redundant and do not significantly contribute to classification. As a result, the important feature test instances are wrongly interpreted more often than the majority ones.

Consequently, the model exhibits low specificity (when Indigenous images are in the minority) and high sensitivity (if the image belongs to the majority class) when handling binary classification cases (such as deepfake data). The general method used to remove the issue of high computational power is to use feature ranking. Using a deep neural network to extract the features and take into account only the most significant features, the features of each class in the training data are ranked. A hybrid learning model is used for classification to have a high accuracy rate. The most significant advantage of hybrid learning is that it enhances average prediction performance. To address these issues, this study introduces a novel and robust method for identifying fake facial images by employing hybrid learning and KPCA to differentiate between authentic and manipulated facial images. *Fig. 5.1* illustrates the steps involved in the proposed system.

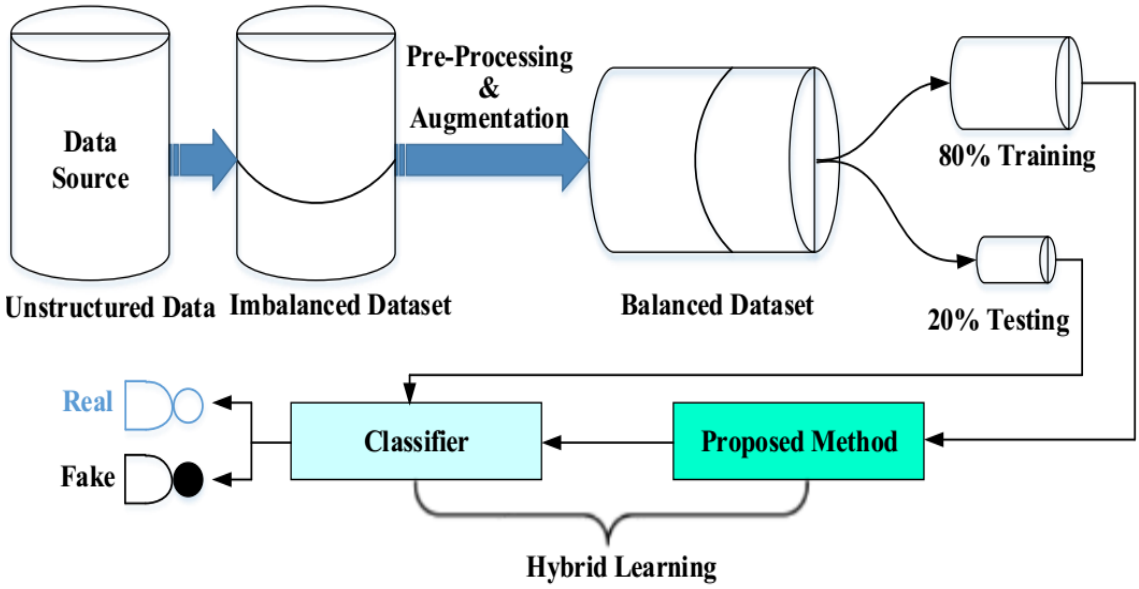


Fig. 5.1: Steps in the proposed facial manipulation detection method.

Hybrid learning combines both traditional machine learning methods and deep learning techniques. Hybrid machine learning models offer the advantages of both traditional and deep learning approaches, enabling more accurate predictions, improved feature representation and enhanced scalability. The proposed approach involves utilizing the EfficientNetV2-L model to extract image features, followed by feature ranking using KPCA and SVM classifier for classification. The resulting

features are then used to classify the real and fake faces. KPCA uses a kernel function to implicitly map the data into a high-dimensional feature space where linear operations can be carried out, whereas PCA operates on the covariance matrix of the input data [161]. This study presents a new method that combines hybrid learning with KPCA to learn features for facial manipulation detection. Through a series of experiments, the effectiveness of the proposed method is evaluated as a face manipulation detection technique.

5.2 Framework based on hybrid learning and KPCA

This section thoroughly explains the proposed framework used for facial manipulation detection based on hybrid learning and KPCA. The main advantage of this proposed method is that using the hybrid learning concept with KPCA works efficiently and fast. The proposed framework consists of a deep learning network, EfficientNetV2-L, for feature extraction, followed by feature ranking using KPCA, and classification is done using a machine learning technique. EfficientNetV2 is the best feature extractor deep learning model [162]. The KPCA with feature dimensionality reduction would help the SVM classifier to make the classification between real and fake facial images efficient and fast. In this section, the proposed method is explained. A detailed framework of the proposed method is shown in *Fig. 5.2.*

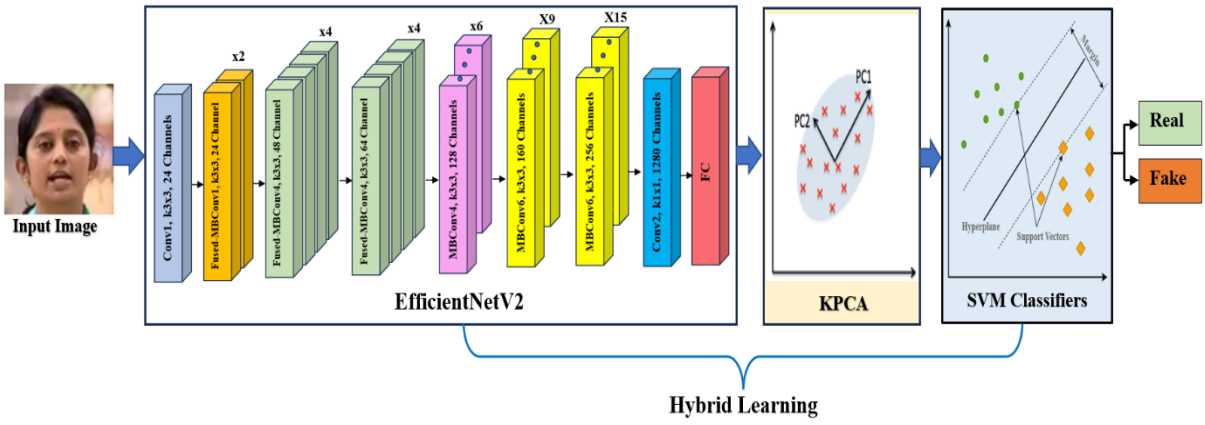


Fig. 5.2: The methodological architectural analysis of the proposed framework for DeepFake detection.

Hand-crafted feature engineering will not be a suitable solution for this problem. Using deep CNN as feature extractors could solve the problem. However, due to the limited availability of high computational resources and the time-consuming process, it would be not easy to get discriminative features from large deep neural networks. Therefore, a popular EfficientNetV2-L model is used as a feature extractor for an efficient facial manipulation detection method. This gives a dynamic, robust, and efficient feature extractor to solve the problem of facial manipulation detection.

Feature learning is performed on the preprocessed images. In feature learning, several features are extracted to distinguish between real and fake facial images. EfficientNetV2 is known for its efficiency in terms of accuracy, parameter, and faster training speed, making it a popular choice as a feature extractor for facial manipulation detection tasks. EfficientNetV2 networks are not just small but also less computational. EfficientNetV2 architecture is up to 6.8 times smaller and significantly faster than previous and more recent SOTA models. Additionally, the V2 version's parameter count is almost half that of the original EfficientNet. EfficientNetV2 achieves better accuracy than previous SOTA models using fewer parameters and less computation. Therefore, EfficientNetV2-L has been utilized as a feature extractor as it has significantly outperformed other ConvNets [162].

PCA is a commonly employed statistical technique for dimensionality reduction. It finds extensive application in tasks such as image compression, text classification, and face recognition [163]. PCA cannot deal only with linear data $x_i \, \mathrm{i} = 1, \ldots., \mathrm{N}, x_i \epsilon \, R, \, \frac{1}{N}\sum_{i=1}^{N} x_i = 0$. PCA is inefficient for working with deep networks and generating features for deepfake images. The KPCA technique is used to deal with nonlinearity in the data. The kernel version enables coping with more complex data patterns, which are not visible under linear transformations alone. KPCA was developed to assist with classifying data whose decision boundaries are described by a nonlinear function [164]. The idea is to go to a higher-dimensional space where the decision boundary becomes linear. Consider a nonlinear transformation $\varphi(x_i)$ that maps the original D-dimensional feature space to a higher-dimensional feature space with *M* dimensions, where *M* is typically much larger than *D*. After the transformation,

each data point *xi* is mapped to a point $\varphi(x_i)$ in the new feature space *F*. Although standard PCA can be applied in this transformed space, it can be computationally expensive and inefficient. However, kernel technique can be utilized to simplify computation.

First, assume that the projected new features, $\varphi(x_i), \ldots\ldots, \varphi(x_N)$, have zero mean, i.e., $\frac{1}{N}\sum_{i=1}^{N}\varphi(x_i) = 0$. The covariance matrix M × M of the projected features is calculated by

$$\mathrm{C} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\varphi(x_i)\varphi(x_i)^{\mathrm{T}} \tag{5.1}$$

Its eigenvalues δ and eigenvectors $V \epsilon F$ *are* satisfying

$$C_{vk} = \delta_{\mathrm{k}} V_{\mathrm{k}} \tag{5.2}$$

where k = 1, 2,…,M. Substituting Eq. (1) and multiplying both sides by $\varphi(x_i)$, will give

$$\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\{\varphi(x_i)\varphi(x_i)^{\mathrm{T}} V_{\mathrm{k}}\varphi(x_i)\} = \delta_{\mathrm{k}} V_{\mathrm{k}}\varphi(x_i) \tag{5.3}$$

and there exist coefficients $a_1, \ldots, a_M$ such that

$$V_k = \sum_{i=1}^{N} a_{ki}\,\varphi(x_i) \tag{5.4}$$

Define the kernel function, i.e.,

$$\mathrm{K}(\mathrm{x_i}, \mathrm{x_j}) = \varphi(x_j)\varphi(x_i)^{\mathrm{T}} \tag{5.5}$$

Substituting Eq. (4) into Eq. (3), will have

$$\frac{1}{N}\sum_{i=1}^{N} k(x_l x_i)\sum_{j=1}^{N} a_{kj} k(x_i, x_j) = \delta_k \sum_{i=1}^{N} a_{ki} k(x_j x_i) \tag{5.6}$$

The matrix notation used is

$$\mathrm{K}^2\mathrm{a_k} = \delta_k \mathrm{NKa_k} \tag{5.7}$$

The kernel principal components are extracted by computing the projections of the image of a test point $\varphi(x)$ onto the eigenvectors, V^k in *F* and calculated using

$$Y_k(x) = \varphi(x)^{\mathrm{T}}\mathrm{V_k} = \sum_{\mathrm{i}=1}^{\mathrm{N}} a_i^k(\varphi(x_i), \varphi(x)) \tag{5.8}$$

The power of kernel method is that you don't have to compute $\varphi(x_i)$ explicitly; they are needed in dot products only. The kernel matrix is directly constructed from the training dataset x_i without actually performing the map φ. The proposed method is effective and robust to various facial manipulation techniques such as identity swap, expression swap, attribute-based manipulation, and entirely synthesized faces. The proposed method has the advantages of being robust, less complicated, having a fast

feature learning process, and taking less execution time. The main advantage of this proposed system is that using the hybrid learning concept with KPCA works efficiently and fast.

5.3 Experiment

In this section, the experiment is carried out on the proposed method. First, the dataset used in the experiment is introduced. Next, the experimental setup is described. Then, the preprocessing and augmentation steps are described in detail. Further, various evaluation parameters used to analyze the performance of the proposed method are described.

5.3.1 Dataset

The DFFD [165] is a large collection that combines several prior datasets. It uses eight algorithms and three genuine image sources to create fake faces. The DFFD presents four primary categories of facial manipulation: exchanging identities, swapping expressions, manipulating attributes, and generating completely synthetic faces. DFFD gathers data from these four groups utilizing cutting-edge techniques to produce synthetic images. Almost half of the images and video frames (47.7%) feature male subjects, while 52.3% depict females. Most samples fall within the age range of 21 to 50 years. To ensure less bias in the distribution of gender, age, and face size, both real and fake samples encompass a range of image qualities, including both low and high-quality images. DFFD uses the FFHQ[4] and CelebA [166] datasets as authentic face samples. These datasets encompass a wide range of variations in terms of gender, race, expression, pose, age, camera quality, illumination, and resolution. In addition to these datasets, DFFD incorporates the source frames from FaceForensics++ [160] as supplementary real faces. The process involves swapping identities and expressions. To achieve facial identity and expression swapping, DFFD utilizes all the video clips available in FaceForensics++. This dataset consists of 1,000 genuine videos sourced from YouTube, along with 3,000 manipulated versions. These manipulated versions

[4] FFHQ:https://github.com/NVlabs/ffhq-dataset.

are divided into two categories: identity swap using FaceSwap and deepfake and expression swap using Face2Face [160]. Two methods were utilized to generate attribute-manipulated images: FaceAPP and StarGAN [154]. FaceAPP, a smartphone app designed for consumers, offers 28 filters that can be used to modify specific facial attributes such as gender, age, hair, beard, and glasses. PGGAN [154] and StyleGAN [156] are the methods used to generate all synthesized faces. In this process, 4,000 faces from the FFHQ dataset and 2,000 from the CelebA dataset are used as the input real images. For each face in the FFHQ dataset, three fake images were generated: two with a randomly chosen manipulation filter and one with multiple manipulation filters applied. On the other hand, for each face in the CelebA dataset, 40 fake images were created using StarGAN, a GAN-based method for translating images to different domains. A collection of 92,000 attribute-manipulated images was obtained through these processes. The DFFD dataset comprises 240336 fake images and 58703 genuine images [165].

5.3.2 Experimental setup

A dataset of real and fake human face images is utilized to develop and apply the proposed facial manipulation detection technique. These images and their corresponding target labels are organized into a dataset. The dataset is then divided into two parts: training and testing data. The method is trained using 80%of the dataset, resulting in a highly optimized and parametrized approach. The performance of the proposed method is evaluated on unseen test data, which accounts for 20% of the dataset. The proposed approach demonstrates exceptional accuracy in predicting outcomes for unseen data. The experiment was done on the DFFD dataset. The Keras library is used across the framework, with Tensor-Flow as the backend. Adam optimizer trains the EfficientNetV2-S network for ten epochs with a momentum rate of 0.9; the initial learning rate is set to 10−4 and a batch size of 64. Early stopping with a patience setting of 100 is utilized to reduce overfitting. In KPCA, the Radial Basis Function kernel is used. The number of components is determined by setting up the explained level to 0.95. The proposed method is trained using a single 12GB NVIDIA Tesla K80 GPU and runs in the Linux operating system.

5.3.3 Preprocessing and Augmentation

In this work, preprocessing is essentially used to ensure that images are in a suitable form for analysis. As the default fixed size of the image taken for training by the EfficientNetV2 is 224×224, all the images from the DFFD dataset have been resized to 224×224.

Augmentation is done to balance classes to improve the quality of the dataset. The DFFD dataset consists of 240336 fake images and 58703 genuine images, which is imbalanced. Therefore, data augmentation is performed on the real images in the dataset to avoid class imbalance. Data augmentation techniques used in this work are as follows:

• **Hue** represents the color's tone or position on the color wheel. Hue jitter introduces a change in the perceived shade of colors within an image. A tiny positive offset chosen randomly from the range [0.05, 0.15] was used to change the hue of the input image.

• **Scale**: To create a scale transformation that resized the input image, a scale factor was randomly selected from the specified range of [1.2, 1.5]. The input image was resized using the obtained scale factor. The scale transformation uniformly resized the image in horizontal and vertical directions, applying the same factor to each dimension.

• **Shear**: Applied a horizontal shear transformation with a randomly chosen shear angle within the range of [−30, 30].

• **Rotation**: When a specific amount changes an image's orientation, it is said to be rotated. It can align tilted photographs or attain a particular viewing angle. A rotation angle was chosen randomly from [−45, 45] degrees.

5.4 Results and Discussion

This section shows the performance of the proposed method on the publicly available DeepFake face dataset. Further, an ablation study investigates the KPCA component's contribution to the overall system's performance. In the end, a comparative analysis of the proposed method is done with the existing SOTA DeepFake face detection methods.

5.4.1 Performance Analysis

The performance of the proposed method is demonstrated on the Diverse Fake Face Dataset (DFFD). Performance analysis is based on accuracy, F1 Score, and execution time. The execution time is computed as the total preprocessing, training, and classification time. Various version of the EfficientNetV2 model, i.e., EfficientNetV2-S, EfficientNetV2-M, and EfficientNetV2-L, is used as feature extractor along with different classifiers, i.e., KNN, Naive Bayes, Decision Tree, Random Forest, and SVM to classify the problem of DeepFake face detection.

Table 5.1: The performance results of the proposed method are compared with those of different EfficientNetV2 models used as feature extractors along with other classifiers

Method	**Classifier**	**Accuracy (%)**	**F1 Score (%)**	**Execution Time (minutes)**
Efficient Net V2 – S	KNN	78.2	75.9	512
	Naive Bayes	79.6	76.2	498
	Decision Tree	82.7	81.5	397
	Random Forest	86.5	85	438
	SVM	89.4	87.63	447
	KNN	79.5	78	628
Efficient Net V2 - M	Naive Bayes	89.8	86.2	609
	Decision Tree	92.2	89.4	562
	Random Forest	96.5	95.4	595
	SVM	97.25	96.1	578
	KNN	92.8	90.4	908
Efficient Net V2 - L	Naive Bayes	93.4	93	897
	Decision Tree	95.6	93.8	842
	Random Forest	96.7	95.2	859
	SVM	97.8	96.92	872
	KNN	94.5	93.7	784
Efficient Net V2 - L + KPCA	Naive Bayes	95.1	94.8	758
	Decision Tree	97.5	96.8	703
	Random Forest	98.2	97.5	714
	SVM	**99.3**	**98.09**	736

Table 5.1 illustrates that the proposed method consisting of EfficientNetV2-L as a feature extractor followed by the KPCA for dimensionality reduction of the feature vector and then classified using SVM classifier achieves the highest classification accuracy of 99.3% and F1 Score of 98.09%. Also, it can be observed that the SVM classifier performed exceptionally well in every case and outperformed all the other classifiers, as shown in *Table 5.1*. The EfficientNetV2- S model achieves the lowest classification accuracy of 78.2% and F1 Score of 75.9% when KNN is used as the classifier. EfficientNetV2-L model used as a feature extractor achieves a classification accuracy of 97.8% and an F1 Score of 96.92%, which is significantly less than the proposed method consisting of a combination of both EfficientNetV2-L and KPCA, with SVM as a classifier. This signifies that using EfficientNetV2-L as a feature extractor along with KPCA is better than using EfficientNetV2-L only. Also, it can be observed from *Table 5.1* that if the EfficientNetV2-S model is used as a feature extractor and the decision tree is used as a classifier, then the execution time is less (approximately 397 min), but the accuracy is 82.7% and F1 Score is 81.5% which is relatively less than the proposed method.

The performance of the proposed approach is evaluated across a range of threshold values using the AUC-ROC curve shown in *Fig. 5.3*. A probability curve called the ROC curve demonstrates how well the classes are separated. It shows the extent to which the method can distinguish between different classes. A higher AUC value indicates better method performance. *Fig. 5.3* shows that the proposed approach has the highest AUC value of 0.980.

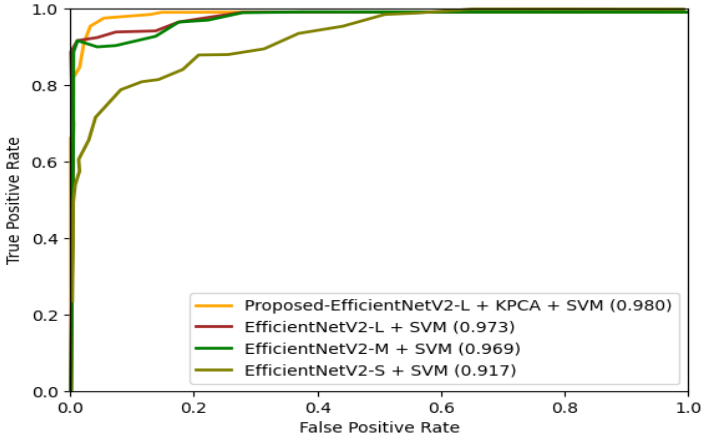


Fig. 5.3: AUC-ROC curve of the proposed method.

5.4.2 Comparitive Analysis

This section details the comparative analysis of the performance of the proposed method compared to the existing SOTA methods. As shown in *Table 5.2*, the experimental results indicate that the proposed method outperforms all the other SOTA methods. The models are trained and tested on the DFFD dataset. The proposed method has the highest accuracy of 99.3%, precision of 0.99, Recall of 0.972, and the highest F1 Score of 0.980. It has been observed that the existing models are very complex and computationally expensive. The proposed method takes less execution time and is less complicated and robust to various facial manipulation techniques such as identity swap, expression swap, attribute-based manipulation, and entirely synthesized faces.

Table 5.2: Comparative analysis of the proposed method.

Method	Accuracy	Precision	Recall	F1 Score
YOLO+LBPH [91]	-	0.889	0.937	-
FF-LBPH DBN [92]	97.82%	-	-	-
DenseNet-121 [93]	80.40%	-	-	-
CNN+PCA [95]	90.76%	0.914	0.901	0.908
FaceMD [96]	90.80%	-	-	-
Proposed	**99.60%**	**0.99**	**0.972**	**0.98**

5.5 Summary

This chapter focuses on advanced techniques for identifying deepfake manipulations, particularly those involving face manipulation. It introduces an effective framework based on hybrid learning and KPCA for deepfake face manipulation detection. The proposed methodology includes EfficientNetV2-L and a KPCA-based hybrid learning approach for facial manipulation detection. EfficientNetV2-L was used to extract the complex discriminative features between real and fake face images from the DFFD dataset. Further, KPCA is used to reduce the dimension of the features extracted from

the EfficientNetV2-L so that the classification between real and fake images can be done in less execution time and there should be less dependency on the computational resources. The experimental results demonstrate the superiority of this method over other facial manipulation detection techniques. The proposed model's accuracy is 99.3%, precision is 0.99, recall is 0.972, and F1 Score is 0.98. Future work proposes an extension of the proposed systems to integrate the KPCA component of the framework into the feature extractor model itself while also investigating and innovating it further. In the future, the goal is to optimize resource utilization, reduce execution time, and enhance overall detection efficiency. Additionally, will prioritize improving the detection model's generalization ability as much as possible as part of the future work.

Chapter 6

Conclusion, Future Scope and Social Impact

This chapter concludes the research conducted in the thesis and summarizes the previous chapters, key findings, contributions and limitations. This chapter also discuss potential future directions for future research in this rapidly evolving field and social impact of this work beyond academic circles.

6.1 Conclusion

This work started with background knowledge about image manipulation detection. The basic terminologies and existing methods concerning manipulation detection with different tampering operations have been discussed in detail. Based on the findings from theoretical and experimental work in this study, it has been observed that deep learning-based models are some of the prominently used methods and perform very well in image manipulation detection. From the analysis, it has been found that initially, a single tampering operation was performed and later on, multiple tampering operations were adopted. The findings reveal that localization of image manipulation is a bit more difficult than image manipulation detection. The deep learning model learns the image's content; however, for manipulation detection, residuals left behind after the tampering operation are used to discriminate between the authentic and manipulated image. It has also been found that using the residuals as the input to the deep learning-based model is much more effective. The experiment result reveals that having multi-modal input is much more effective. The proposed work provides practical implications for OfSFD, multiple manipulation detection, and deepfake face detection, which could be useful in protecting the world from misleading information.

The thesis presented a systematic approach to image manipulation detection spanning foundational aspects and a review of existing detection techniques to develop methods for specific, multiple and synthetic manipulations. A robust and efficient method, namely eSNN, has been introduced for WIOfSV. The technique uses the pre-

trained model (EfficientNet) to direct the feature learning process in the twin network of the SNN to distinguish between genuine and forged signatures. The method has the advantages of being less complicated and taking less time to train and infer. Also, a residual-based CNN model has been developed for CMFD. Architectures such as MDLFormer and LFRViT illustrate the efficacy of integrating the residuals-based inputs and ViT to detect multiple forgery detection. These models enhance detection accuracy, robustness, and generalizability, addressing the limitations identified in existing methods. A hybrid learning-based approach, including KPCA, has been introduced for deepfake face manipulation detection. The technique uses the EfficientNetV2-L model for feature extraction, which is topped up with KPCA for feature dimensionality reduction to have an effective and fast feature learning process. The method is robust to various facial manipulation techniques such as identity swap, expression swap, attribute-based manipulation, and entirely synthesized faces. The proposed methods demonstrate significant improvements over existing SOTA methods, validated through extensive experimentation on standard datasets.

6.2 Future Scope

The researchers have adopted many different approaches to understand better and characterize image manipulation detection; this diversification helps to focus on the future enhancement of image manipulation detection techniques. Despite substantial advancement in the research field, open research issues still require further study. Our findings suggest a need for additional research, which consists of the following aspects:

- The goal in the future is to optimize resource utilization, have efficient architecture and improve overall detection accuracy. Additionally, will prioritize enhancing the detection model's generalization ability as part of future work.
- Deepfake content detection is one of the emerging topics. Future work should focus on developing unified deepfake detection systems that can identify image- and video-based manipulations.

- Given the increasing prevalence of manipulated media on online social networks (OSNs), future work should focus on developing models resilient to manipulations caused by shared and spread across OSNs. Moreover, the manipulation detection frameworks should be robust enough to withstand the manipulations applied by OSNs.
- As manipulation techniques continue to evolve and also given the possibility that forgers would adapt to detection approaches over time, robust models that may evolve in response to new manipulations are required. Future work may include implementing continuous learning or domain adaptation approaches, enabling models to remain effective even when new manipulating styles and techniques emerge.
- Generative adversarial methods generate synthetic fake images and add noise to the image, making it difficult for the detector to detect the manipulation. Image manipulation detection methods are subject to adversarial attacks, where minimal alterations lead to incorrect classifications. Future work should focus on developing detection frameworks immune to adversarial perturbations.

6.3 Social Impact

The development of a robust framework for image manipulation detection has profound social implications in today's digital age, where visual content plays a pivotal role in communication, decision-making and the dissemination of information. The proliferation of advanced image editing tools and generative technologies, such as GANs, has significantly increased the potential for creating manipulated or falsified images. This trend seriously challenges societal trust, media credibility, and individual rights. Techniques like digital OfSFD, CMFD, splicing detection, inpainting detection and deepfake detection are essential for safeguarding societal trust and ensuring the authenticity of digital content.

One of the key social impacts of this research is its ability to combat misinformation and disinformation campaigns. Manipulated images often spread false narratives, incite violence, or mislead public opinion on social and political issues. By providing reliable tools for detecting such manipulations, the proposed framework

can help maintain the integrity of digital media, ensuring that the public receives accurate and authentic information.

Digital signatures are pivotal in financial transactions, legal documents, and authentication processes. Forged signatures can lead to identity theft, financial fraud, and legal disputes. A reliable OfSFD framework can help organizations and individuals identify forged signatures, preventing financial losses and protecting reputations. By ensuring the authenticity of such signatures, the proposed framework contributes to building trust in digital documentation systems and reducing vulnerabilities in critical infrastructures.

Copy-move forgery, where parts of an image are duplicated and pasted within the same image, is often used to conceal or manipulate visual evidence. This type of forgery is commonly found in fake news, manipulated evidence in legal cases, and fraudulent claims in insurance or real estate. CMFD can help uncover hidden manipulations, ensuring the integrity of visual evidence and mitigating the spread of false information, which can have far-reaching societal consequences.

The ability to detect multiple forgeries, such as combinations of copy-move, splicing, inpainting and many more, enhances the framework's utility in complex scenarios. This capability is particularly valuable in forensic investigations and digital media verification, where multiple manipulations can obscure the truth. By localizing and identifying all manipulated regions, the framework supports law enforcement agencies, judicial systems, and media outlets in maintaining the integrity of evidence and reporting. This contributes to societal accountability and preventing malicious intent in critical domains.

Furthermore, deepfake technology, powered by advanced AI algorithms, poses a significant threat to social trust by generating hyper-realistic yet fabricated videos or images. Deepfakes have been weaponized for political propaganda, defamation, and even financial fraud. Detecting deepfakes is essential to prevent the erosion of public trust in visual media and ensure that the dissemination of falsified content does not destabilize societies or harm individuals. The proposed framework

contributes to developing a safer digital ecosystem by addressing this emerging threat.

On a broader level, this work supports the ethical use of technology and encourages accountability in digital media creation and distribution. By fostering a culture of authenticity, it addresses the societal need for trust in digital interactions and mitigates the negative impacts of technological misuse. This work for image manipulation detection can potentially address critical challenges in misinformation, legal justice, personal security, and media ethics. Its societal impact extends beyond technical advancements, contributing to a safer, more trustworthy, and equitable digital environment.

References

[1] R. Thakur and R. Rohilla, “Recent advances in digital image manipulation detection techniques: A brief review,” *Forensic Sci. Int.*, vol. 312, p. 110311, 2020, doi: 10.1016/j.forsciint.2020.110311.

[2] V. Schetinger, M. M. Oliveira, R. da Silva, and T. J. Carvalho, “Humans are easily fooled by digital images,” *Comput. Graph.*, vol. 68, pp. 142–151, 2017, doi: 10.1016/j.cag.2017.08.010.

[3] H. Farid, “Photo Tampering Throughout History,” *[Online] Http//Www. Cs. Dartmouth. Edu/Farid/Research/Digitaltampering*, no. 2, 2011.

[4] P. Korshunov and S. Marcel, “DeepFakes: a New Threat to Face Recognition? Assessment and Detection,” pp. 1–5, 2018, [Online]. Available: http://arxiv.org/abs/1812.08685.

[5] Farid Hany, “A Survey of Image Forgery Detection,” *IEEE Signal Process. Mag.*, vol. 26, no. March, pp. 16–25, 2009.

[6] M. Dalal and M. Juneja, “Steganography and Steganalysis (in digital forensics): a Cybersecurity guide,” *Multimed. Tools Appl.*, vol. 80, no. 4, pp. 5723–5771, 2021.

[7] G. Zhou and D. Lv, “An overview of digital watermarking in image forensics,” *Proc. - 4th Int. Jt. Conf. Comput. Sci. Optim. CSO 2011*, pp. 332–335, 2011, doi: 10.1109/CSO.2011.85.

[8] X. Lin, J. H. Li, S. L. Wang, A. W. C. Liew, F. Cheng, and X. S. Huang, “Recent Advances in Passive Digital Image Security Forensics: A Brief Review,” *Engineering*, vol. 4, no. 1, pp. 29–39, 2018, doi: 10.1016/j.eng.2018.02.008.

[9] S. Mushtaq and A. H. Mir, “Signature verification: A study,” *Proc. - 4th IEEE Int. Conf. Comput. Commun. Technol. ICCCT 2013*, pp. 258–263, 2013, doi: 10.1109/ICCCT.2013.6749637.

[10] N. B. A. Warif *et al.*, “Copy-move forgery detection: Survey, challenges and future directions,” *J. Netw. Comput. Appl.*, vol. 75, pp. 259–278, 2016, doi: 10.1016/j.jnca.2016.09.008.

[11] L. Zheng, Y. Zhang, and V. L. L. Thing, “A survey on image tampering and its detection in real-world photos,” *J. Vis. Commun. Image Represent.*, vol. 58, pp. 380–399, 2019, doi: 10.1016/j.jvcir.2018.12.022.

[12] M. Masood, M. Nawaz, K. M. Malik, A. Javed, A. Irtaza, and H. Malik, “Deepfakes generation and detection: state-of-the-art, open challenges, countermeasures, and way forward,” *Appl. Intell.*, vol. 53, no. 4, pp. 3974–4026, 2023, doi: 10.1007/s10489-022-03766-z.

[13] A. Dosovitskiy *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv Prepr. arXiv2010.11929*, 2020.

[14] A. Agarwal and V. Khandelwal, “Multiple Manipulation Detection in Images Using Frequency Domain Features in 3D-CNN,” *Arab. J. Sci. Eng.*, vol. 48, no. 11, pp. 14573–14587, 2023, doi: 10.1007/s13369-023-07727-7.

[15] A. Mazumdar, J. Singh, Y. S. Tomar, and P. K. Bora, “Universal Image Manipulation Detection using Deep Siamese Convolutional Neural Network,” pp. 1–6, 2018, [Online]. Available: http://arxiv.org/abs/1808.06323.

[16] B. Bayar and M. C. Stamm, “A deep learning approach to universal image manipulation detection using a new convolutional layer,” *IH MMSec 2016 - Proc. 2016 ACM Inf. Hiding Multimed. Secur. Work.*, pp. 5–10, 2016, doi: 10.1145/2909827.2930786.

[17] O. M. Al-Qershi and B. E. Khoo, “Evaluation of copy-move forgery detection: datasets and evaluation metrics,” *Multimed. Tools Appl.*, vol. 77, no. 24, pp. 31807–31833, 2018, doi: 10.1007/s11042-018-6201-4.

[18] W. Wang, J. Dong, and T. Tan, “A survey of passive image tampering detection,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5703 LNCS, pp. 308–322, 2009, doi: 10.1007/978-3-642-03688-0_27.

[19] J. He, Z. Lin, L. Wang, and X. Tang, “Detecting doctored JPEG images Via DCT coefficient analysis,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3953 LNCS, pp. 423–435, 2006, doi: 10.1007/11744078_33.

[20] A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, G. Bebis, and H. Mathkour, “Passive detection of image forgery using DCT and local binary pattern,” *Signal, Image Video Process.*, vol. 11, no. 1, 2017, doi: 10.1007/s11760-016-0899-0.

[21] T. Mahmood, Z. Mehmood, M. Shah, and T. Saba, “A robust technique for copy-move forgery detection and localization in digital images via stationary wavelet and discrete cosine transform,” *J. Vis. Commun. Image Represent.*, vol. 53, 2018, doi: 10.1016/j.jvcir.2018.03.015.

[22] K. Hayat and T. Qazi, “Forgery detection in digital images via discrete wavelet and discrete cosine transforms,” *Comput. Electr. Eng.*, vol. 62, pp. 448–458, Aug. 2017, doi: 10.1016/J.COMPELECENG.2017.03.013.

[23] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, “A SIFT-based forensic method for copy-move attack detection and transformation recovery,” *IEEE Trans. Inf. Forensics Secur.*, vol. 6, no. 3 PART 2, pp. 1099–1110, 2011, doi: 10.1109/TIFS.2011.2129512.

[24] B. Xu, J. Wang, G. Liu, and Y. Dai, “Image copy-move forgery detection based on SURF,” *Proc. - 2010 2nd Int. Conf. Multimed. Inf. Netw. Secur. MINES 2010*, pp. 889–892, 2010, doi: 10.1109/MINES.2010.189.

[25] Y. Zhang, J. Goh, L. L. Win, and V. Thing, “Image region forgery detection: A deep learning approach,” *Cryptol. Inf. Secur. Ser.*, vol. 14, pp. 1–11, 2016, doi: 10.3233/978-1-61499-617-0-1.

[26] Z. Bao and R. Xue, “Survey on deep learning applications in digital image security,” *Opt. Eng.*, vol. 60, no. 12, pp. 1–32, 2021, doi: 10.1117/1.oe.60.12.120901.

[27] X. Jin, P. Jing, and Y. Su, “AMFNet: An adversarial network for median

filtering detection," *IEEE Access*, vol. 6, pp. 50459–50467, 2018, doi: 10.1109/ACCESS.2018.2867370.

[28] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, "Detection of GAN-Generated Fake Images over Social Networks," *Proc. - IEEE 1st Conf. Multimed. Inf. Process. Retrieval, MIPR 2018*, pp. 384–389, 2018, doi: 10.1109/MIPR.2018.00084.

[29] M. K. Kalera, S. Srihari, and A. Xu, "Offline signature verification and identification using distance statistics," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 18, no. 07, pp. 1339–1360, 2004.

[30] M. E. Munich and P. Perona, "Visual identification by signature tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 200–217, 2003.

[31] G. Dimauro, S. Impedovo, G. Pirlo, and A. Salzo, "A multi-expert signature verification system for bankcheck processing," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 11, no. 05, pp. 827–844, 1997.

[32] M. M. Hameed, R. Ahmad, M. L. M. Kiah, and G. Murtaza, "Machine learning-based offline signature verification systems: A systematic review," *Signal Process. Image Commun.*, vol. 93, no. January, p. 116139, 2021, doi: 10.1016/j.image.2021.116139.

[33] A. Jain, S. K. Singh, and K. P. Singh, "Handwritten signature verification using shallow convolutional neural network," *Multimed. Tools Appl.*, vol. 79, no. 27–28, pp. 19993–20018, 2020, doi: 10.1007/s11042-020-08728-6.

[34] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[35] N. Sharma *et al.*, "Offline signature verification using deep neural network with application to computer vision," *J. Electron. Imaging*, vol. 31, no. 4, p. 41210, 2022.

[36] P. Mehta, M. K. Singh, and N. Singha, "Near-duplicate image detection based on wavelet decomposition with modified deep learning model," *J. Electron.*

Imaging, vol. 31, no. 2, p. 23017, 2022.

[37] E. Parcham, M. Ilbeygi, and M. Amini, “CBCapsNet : A novel writer-independent offline signature verification model using a CNN-based architecture and capsule neural networks,” *Expert Syst. Appl.*, vol. 185, no. April, p. 115649, 2021, doi: 10.1016/j.eswa.2021.115649.

[38] Y. M. Al-Omari, S. N. H. S. Abdullah, and K. Omar, “State-of-the-art in offline signature verification system,” in *2011 International Conference on Pattern Analysis and Intelligence Robotics*, 2011, vol. 1, pp. 59–64.

[39] F. E. Batool *et al.*, “Offline signature verification system: a novel technique of fusion of GLCM and geometric features using SVM,” *Multimed. Tools Appl.*, vol. 83, no. 5, pp. 14959–14978, 2024, doi: 10.1007/s11042-020-08851-4.

[40] B. Fang, C. H. Leung, Y. Y. Tang, K. W. Tse, P. C. K. Kwok, and Y. K. Wong, “Off-line signature verification by the tracking of feature and stroke positions,” *Pattern Recognit.*, vol. 36, no. 1, pp. 91–101, 2003.

[41] A. Alaei, S. Pal, U. Pal, and M. Blumenstein, “An efficient signature verification method based on an interval symbolic representation and a fuzzy similarity measure,” *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 10, pp. 2360–2372, 2017.

[42] M. A. Ferrer, J. B. Alonso, and C. M. Travieso, “Offline geometric parameters for automatic signature verification using fixed-point arithmetic,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 993–997, 2005.

[43] S. Dey *et al.*, “Signet: Convolutional siamese network for writer independent offline signature verification,” *arXiv Prepr. arXiv1707.02131*, no. 1, pp. 1–7, 2017.

[44] Y. Guerbai, Y. Chibani, and B. Hadjadji, “The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters,” *Pattern Recognit.*, vol. 48, no. 1, pp. 103–113, 2015.

[45] A. Hamadene and Y. Chibani, “One-class writer-independent offline signature

verification using feature dissimilarity thresholding," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 6, pp. 1226–1238, 2016.

[46] T. Longjam, D. R. Kisku, and P. Gupta, "Multi-scripted Writer Independent Off-line Signature Verification using Convolutional Neural Network," *Multimed. Tools Appl.*, vol. 82, no. 4, pp. 5839–5856, 2023, doi: 10.1007/s11042-022-13392-z.

[47] A. B. Jagtap, D. D. Sawat, R. S. Hegadi, and R. S. Hegadi, "Verification of genuine and forged offline signatures using Siamese Neural Network (SNN)," *Multimed. Tools Appl.*, vol. 79, no. 47–48, pp. 35109–35123, 2020, doi: 10.1007/s11042-020-08857-y.

[48] B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants," *Forensic Sci. Int.*, vol. 171, no. 2–3, pp. 180–189, 2007, doi: 10.1016/j.forsciint.2006.11.002.

[49] Y. Cao, T. Gao, L. Fan, and Q. Yang, "A robust detection algorithm for copy-move forgery in digital images," *Forensic Sci. Int.*, vol. 214, no. 1–3, pp. 33–43, 2012, doi: 10.1016/j.forsciint.2011.07.015.

[50] Z. Junhong, "Detection of copy-move forgery based on one improved LLE method," in *2010 2nd International Conference on Advanced Computer Control*, 2010, vol. 4, pp. 547–550.

[51] G. Lynch, F. Y. Shih, and H.-Y. M. Liao, "An efficient expanding block algorithm for image copy-move forgery detection," *Inf. Sci. (Ny).*, vol. 239, pp. 253–265, 2013, doi: https://doi.org/10.1016/j.ins.2013.03.028.

[52] Y. Liu, Q. Guan, and X. Zhao, "Copy-move forgery detection based on convolutional kernel network," *Multimed. Tools Appl.*, vol. 77, no. 14, pp. 18269–18293, 2018, doi: 10.1007/s11042-017-5374-6.

[53] C.-M. Pun, X.-C. Yuan, and X.-L. Bi, "Image forgery detection using adaptive oversegmentation and feature point matching," *ieee Trans. Inf. forensics Secur.*, vol. 10, no. 8, pp. 1705–1716, 2015.

[54] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, “Median Filtering Forensics Based on Convolutional Neural Networks,” *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1849–1853, 2015, doi: 10.1109/LSP.2015.2438008.

[55] P. Peng, T. Sun, X. Jiang, K. Xu, B. Li, and Y. Shi, “Detection of Double JPEG Compression with the Same Quantization Matrix Based on Convolutional Neural Networks,” *2018 Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. APSIPA ASC 2018 - Proc.*, no. November, pp. 717–721, 2019, doi: 10.23919/APSIPA.2018.8659763.

[56] W. Shan, Y. Yi, R. Huang, and Y. Xie, “Robust contrast enhancement forensics based on convolutional neural networks,” *Signal Process. Image Commun.*, vol. 71, no. December 2018, pp. 138–146, 2019, doi: 10.1016/j.image.2018.11.011.

[57] P. Yang, R. Ni, and Y. Zhao, “Recapture image forensics based on Laplacian convolutional neural networks,” in *Digital Forensics and Watermarking: 15th International Workshop, IWDW 2016, Beijing, China, September 17-19, 2016, Revised Selected Papers 15*, 2017, pp. 119–128.

[58] Y. Rao and J. Ni, “A deep learning approach to detection of splicing and copy-move forgeries in images,” *8th IEEE Int. Work. Inf. Forensics Secur. WIFS 2016*, pp. 1–6, 2017, doi: 10.1109/WIFS.2016.7823911.

[59] R. Salloum, Y. Ren, C. C. Jay Kuo, and C.-C. J. Kuo, “Image splicing localization using a multi-task fully convolutional network (MFCN),” *J. Vis. Commun. Image Represent.*, vol. 51, pp. 201–209, 2018, doi: 10.1016/j.jvcir.2018.01.010.

[60] D. Cozzolino, G. Poggi, and L. Verdoliva, “Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection,” *IH MMSec 2017 - Proc. 2017 ACM Work. Inf. Hiding Multimed. Secur.*, no. Section 5, pp. 159–164, 2017, doi: 10.1145/3082031.3083247.

[61] J. Ouyang, Y. Liu, and M. Liao, “Copy-move forgery detection based on deep learning,” *Proc. - 2017 10th Int. Congr. Image Signal Process. Biomed. Eng. Informatics, CISP-BMEI 2017*, vol. 2018-Janua, pp. 1–5, 2018, doi:

10.1109/CISP-BMEI.2017.8301940.

[62] Y. Wu, W. Abd-Almageed, and P. Natarajan, “Image copy-move forgery detection via an end-to-end deep neural network,” *Proc. - 2018 IEEE Winter Conf. Appl. Comput. Vision, WACV 2018*, vol. 2018-Janua, no. d, pp. 1907–1915, 2018, doi: 10.1109/WACV.2018.00211.

[63] B. Liu and C.-M. M. Pun, “Locating splicing forgery by adaptive-SVD noise estimation and vicinity noise descriptor,” *Neurocomputing*, vol. 387, pp. 172–187, Apr. 2020, doi: 10.1016/j.neucom.2019.12.105.

[64] Y. Wu, W. Abd-Almageed, and P. Natarajan, “BusterNet: Detecting copy-move image forgery with source/target localization,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11210 LNCS, doi: 10.1007/978-3-030-01231-1_11.

[65] B. Bayar and M. C. Stamm, “A generic approach towards image manipulation parameter estimation using convolutional neural networks,” *IH MMSec 2017 - Proc. 2017 ACM Work. Inf. Hiding Multimed. Secur.*, pp. 147–157, 2017, doi: 10.1145/3082031.3083249.

[66] J. Bunk *et al.*, “Detection and Localization of Image Forgeries Using Resampling Features and Deep Learning,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2017-July, pp. 1881–1889, 2017, doi: 10.1109/CVPRW.2017.235.

[67] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury, “Hybrid LSTM and Encoder-Decoder Architecture for Detection of Image Forgeries,” *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3286–3300, 2019, doi: 10.1109/TIP.2019.2895466.

[68] J. Wang *et al.*, “Objectformer for image manipulation detection and localization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2364–2373.

[69] S. Li, W. Ma, J. Guo, S. Xu, B. Li, and X. Zhang, “UnionFormer: Unified-

Learning Transformer with Multi-View Representation for Image Manipulation Detection and Localization," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern RecognitionProceedings IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024.

[70] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1053–1061, doi: 10.1109/CVPR.2018.00116.

[71] T. Chen, B. Li, and J. Zeng, "Learning Traces by Yourself: Blind Image Forgery Localization via Anomaly Detection With ViT-VAE," *IEEE Signal Process. Lett.*, vol. 30, pp. 150–154, 2023, doi: 10.1109/LSP.2023.3245947.

[72] F. Guillaro, D. Cozzolino, A. Sud, N. Dufour, and L. Verdoliva, "TruFor: Leveraging all-round clues for trustworthy image forgery detection and localization," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 20606–20615, 2023, doi: 10.1109/CVPR52729.2023.01974.

[73] F. Li, H. Zhai, X. Zhang, and C. Qin, "Image Manipulation Localization Using Spatial–Channel Fusion Excitation and Fine-Grained Feature Enhancement," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–14, 2024, doi: 10.1109/TIM.2023.3338703.

[74] J. Peng, C. Liu, H. Pang, X. Gao, G. Cheng, and B. Hao, "GP-Net: Image Manipulation Detection and Localization via Long-Range Modeling and Transformers," *Appl. Sci.*, vol. 13, no. 21, 2023, doi: 10.3390/app132112053.

[75] H. Li and J. Huang, "Localization of deep inpainting using high-pass fully convolutional network," in *proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 8301–8310.

[76] X. Chen, C. Dong, J. Ji, J. Cao, and X. Li, "Image manipulation detection by multi-view multi-scale supervision," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 14185–14193.

[77] C. Dong, X. Chen, R. Hu, J. Cao, and X. Li, "Mvss-net: Multi-view multi-scale

supervised networks for image manipulation detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3539–3553, 2022.

[78] P. Zhou *et al.*, "Generate, segment, and refine: Towards generic manipulation segmentation," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, vol. 34, no. 07, pp. 13058–13065.

[79] M. Huh, A. Liu, A. Owens, and A. A. Efros, "Fighting fake news: Image splice detection via learned self-consistency," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 101–117.

[80] R. Mehta, K. Aggarwal, D. Koundal, A. Alhudhaif, and K. Polat, "Markov features based DTCWS algorithm for online image forgery detection using ensemble classifier in the pandemic," *Expert Syst. Appl.*, vol. 185, p. 115630, 2021.

[81] Y. Liu, X. Zhu, X. Zhao, and Y. Cao, "Adversarial learning for constrained image splicing detection and localization based on atrous convolution," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 10, pp. 2551–2566, 2019.

[82] R. Thakur and R. Rohilla, "Copy-Move Forgery Detection using Residuals and Convolutional Neural Network Framework: A Novel Approach," in *2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*, 2019, pp. 561–564, doi: 10.1109/PEEIC47157.2019.8976868.

[83] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury, "Hybrid lstm and encoder–decoder architecture for detection of image forgeries," *IEEE Trans. image Process.*, vol. 28, no. 7, pp. 3286–3300, 2019.

[84] Y. Wu, W. Abdalmageed, and P. Natarajan, "Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, vol. 2019-June, doi: 10.1109/CVPR.2019.00977.

[85] X. Hu, Z. Zhang, Z. Jiang, S. Chaudhuri, Z. Yang, and R. Nevatia, “SPAN: Spatial pyramid attention network for image manipulation localization,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, 2020, pp. 312–328.

[86] A. Vaswani, “Attention is all you need,” *Adv. Neural Inf. Process. Syst.*, 2017.

[87] W. Wang *et al.*, “Pyramid vision transformer: A versatile backbone for dense prediction without convolutions,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[88] D. Zhang, H. Zhang, J. Tang, M. Wang, X. Hua, and Q. Sun, “Feature pyramid transformer,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, 2020, pp. 323–339.

[89] Z. Liu *et al.*, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10012–10022.

[90] Y. Zhang, H. Liu, and Q. Hu, “Transfuse: Fusing transformers and cnns for medical image segmentation,” in *Medical image computing and computer assisted intervention–MICCAI 2021: 24th international conference, Strasbourg, France, September 27–October 1, 2021, proceedings, Part I 24*, 2021, pp. 14–24.

[91] M. Tan *et al.*, “Evaluation of deepfake detection using YOLO with local binary pattern histogram,” *PeerJ Comput. Sci.*, vol. 8, no. 1, p. e1086, 2022, doi: https://doi.org/10.1016/j.procs.2018.10.171.

[92] S. T. Suganthi, M. U. A. Ayoobkhan, N. Bacanin, K. Venkatachalam, H. Štěpán, and T. Pavel, “Deep learning model for deep fake face recognition and detection,” *PeerJ Comput. Sci.*, vol. 8, p. e881, 2022.

[93] S. Solaiyappan and Y. Wen, “Machine learning based medical image deepfake detection: A comparative study,” *Mach. Learn. with Appl.*, vol. 8, p. 100298, 2022.

[94] L. Stroebel, M. Llewellyn, T. Hartley, T. S. Ip, and M. Ahmed, “A systematic literature review on the effectiveness of deepfake detection techniques,” *J. Cyber Secur. Technol.*, vol. 7, no. 2, pp. 83–113, 2023.

[95] H. Sabah, “A Detection of Deep Fake in Face Images Using Deep Learning,” *Wasit J. Comput. Math. Sci.*, vol. 1, no. 4, pp. 60–71, 2022, doi: 10.31185/wjcm.92.

[96] M. Aloraini, “FaceMD: convolutional neural network-based spatiotemporal fusion facial manipulation detection,” *Signal, Image Video Process.*, vol. 17, no. 1, pp. 247–255, 2023, doi: 10.1007/s11760-022-02227-x.

[97] P. V Hatkar and Z. J. Tamboli, “Image Processing for Signature Verification,” *Int. J. Innov. Res. Comput. Sci. Technol.*, vol. 3, no. 3, pp. 127–129, 2015.

[98] J. Long, C. Xie, and Z. Gao, “High discriminant features for writer-independent online signature verification,” *Multimed. Tools Appl.*, vol. 82, no. 25, pp. 38447–38465, 2023, doi: 10.1007/s11042-023-14638-0.

[99] M. Houtinezhad and H. R. Ghaffari, *Off-line signature verification system using features linear mapping in the candidate points*, vol. 81, no. 17. Multimedia Tools and Applications, 2022.

[100] P. Agrawal, D. Chaudhary, and V. Madaan, “Automated bank cheque verification using image,” *Multimed. Tools Appl.*, vol. 80, pp. 5319–5350, 2020.

[101] G. S. Eskander, R. Sabourin, and E. Granger, “Hybrid writer-independent–writer-dependent offline signature verification system,” *IET biometrics*, vol. 2, no. 4, pp. 169–181, 2013.

[102] E. N. Zois, A. Alexandridis, and G. Economou, “Writer independent offline signature verification based on asymmetric pixel relations and unrelated training-testing datasets,” *Expert Syst. Appl.*, vol. 125, pp. 14–32, 2019.

[103] S. Chopra, R. Hadsell, and Y. LeCun, “Learning a similarity metric discriminatively, with application to face verification,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*,

2005, vol. 1, pp. 539–546.

[104] M. N. A. Bhatti, I. Siddiqi, and M. Moetesum, “LSTM-based Siamese neural network for Urdu news story segmentation,” *Int. J. Doc. Anal. Recognit.*, 2023, doi: 10.1007/s10032-023-00441-y.

[105] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*, 2019, pp. 6105–6114.

[106] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Trans. Syst. Man. Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.

[107] M. A. Ferrer, M. Diaz-Cabrera, and A. Morales, “Synthetic off-line signature image generation,” in *2013 international conference on biometrics (ICB)*, 2013, pp. 1–7, doi: 10.1109/ICB.2013.6612969.

[108] J. Ortega-Garcia *et al.*, “MCYT baseline corpus: a bimodal biometric database,” *IEE Proceedings-Vision, Image Signal Process.*, vol. 150, no. 6, pp. 395–401, 2003.

[109] J. Fierrez-Aguilar, N. Alonso-Hermira, G. Moreno-Marquez, and J. Ortega-Garcia, “An off-line signature verification system based on fusion of local and global information,” in *Biometric Authentication: ECCV 2004 International Workshop, BioAW 2004, Prague, Czech Republic, May 15th, 2004. Proceedings*, 2004, pp. 295–306.

[110] S. Pal, A. Alaei, U. Pal, and M. Blumenstein, “Performance of an off-line signature verification method based on texture features on a large indic-script signature dataset,” in *2016 12th IAPR workshop on document analysis systems (DAS)*, 2016, pp. 72–77.

[111] A. Soleimani, K. Fouladi, and B. N. Araabi, “UTSig: A Persian offline signature dataset,” *IET Biometrics*, vol. 6, no. 1, pp. 1–8, 2017, doi: 10.1049/iet-bmt.2015.0058.

[112] A. Dutta, U. Pal, and J. Lladós, “Compact correlated features for writer

independent signature verification," in *2016 23rd international conference on pattern recognition (ICPR)*, 2016, pp. 3422–3427.

[113] N. Çalik *et al.*, "Large-scale offline signature recognition via deep neural networks and feature embedding," *Neurocomputing*, vol. 359, pp. 1–14, 2019, doi: 10.1016/j.neucom.2019.03.027.

[114] Jahandad, S. M. Sam, K. Kamardin, N. N. Amir Sjarif, and N. Mohamed, "Offline signature verification using deep learning convolutional Neural network (CNN) architectures GoogLeNet inception-v1 and inception-v3," *Procedia Comput. Sci.*, vol. 161, pp. 475–483, 2019, doi: 10.1016/j.procs.2019.11.147.

[115] M. M. Yapıcı, A. Tekerek, and N. Topaloğlu, "Deep learning-based data augmentation method and signature verification system for offline handwritten signature," *Pattern Anal. Appl.*, vol. 24, no. 1, pp. 165–179, 2021, doi: 10.1007/s10044-020-00912-6.

[116] S. Y. Ooi, A. B. J. Teoh, Y. H. Pang, and B. Y. Hiew, "Image-based handwritten signature verification using hybrid methods of discrete Radon transform, principal component analysis and probabilistic neural network," *Appl. Soft Comput. J.*, vol. 40, pp. 274–282, 2016, doi: 10.1016/j.asoc.2015.11.039.

[117] A. Soleimani, B. N. Araabi, and K. Fouladi, "Deep multitask metric learning for offline signature verification," *Pattern Recognit. Lett.*, vol. 80, pp. 84–90, 2016.

[118] F. Alonso-Fernandez, M. C. Fairhurst, J. Fierrez, and J. Ortega-Garcia, "Automatic measures for predicting performance in off-line signature," in *2007 IEEE international conference on image processing*, 2007, vol. 1, pp. I–369.

[119] H. Hezil, R. Djemili, and H. Bourouba, "Signature recognition using binary features and KNN," *Int. J. Biom.*, vol. 10, no. 1, pp. 1–15, 2018.

[120] A. K. Bhunia, A. Alaei, and P. P. Roy, "Signature verification approach using fusion of hybrid texture features," *Neural Comput. Appl.*, vol. 31, pp. 8737–8748, 2019.

[121] P. Maergner *et al.*, “Combining graph edit distance and triplet networks for offline signature verification,” *Pattern Recognit. Lett.*, vol. 125, pp. 527–533, 2019, doi: https://doi.org/10.1016/j.patrec.2019.06.024.

[122] S. Shariatmadari, S. Emadi, and Y. Akbari, “Patch-based offline signature verification using one-class hierarchical deep learning,” *Int. J. Doc. Anal. Recognit.*, vol. 22, no. 4, pp. 375–385, 2019.

[123] S. Masoudnia, O. Mersa, B. N. Araabi, A.-H. Vahabie, M. A. Sadeghi, and M. N. Ahmadabadi, “Multi-representational learning for Offline Signature Verification using Multi-Loss Snapshot Ensemble of CNNs,” *Expert Syst. Appl.*, vol. 133, pp. 317–330, 2019, doi: https://doi.org/10.1016/j.eswa.2019.03.040.

[124] S. Chen and S. Srihari, “Use of exterior contours and shape features in off-line signature verification,” in *Eighth International Conference on Document Analysis and Recognition (ICDAR’05)*, 2005, pp. 1280–1284.

[125] S. Chen and S. Srihari, “A new off-line signature verification method based on graph,” in *18th International Conference on Pattern Recognition (ICPR’06)*, 2006, vol. 2, pp. 869–872.

[126] R. Kumar, J. D. Sharma, and B. Chanda, “Writer-independent off-line signature verification using surroundedness feature,” *Pattern Recognit. Lett.*, vol. 33, no. 3, pp. 301–308, 2012.

[127] M. Liwicki *et al.*, “Signature verification competition for online and offline skilled forgeries (sigcomp2011),” in *2011 International conference on document analysis and recognition*, 2011, pp. 1480–1484, doi: 10.1109/ICDAR.2011.294.

[128] G. Alvarez, B. Sheffer, and M. Bryant, “Offline signature verification with convolutional neural networks,” *Tech. report, Stanford Univ.*, 2016.

[129] D. Tralic, I. Zupancic, S. Grgic, and M. Grgic, “CoMoFoD—New database for copy-move forgery detection,” in *Proceedings ELMAR-2013*, 2013, pp. 49–54.

[130] P. Bas, T. Filler, and T. Pevný, “” Break our steganographic system”: the ins and outs of organizing BOSS,” in *International workshop on information hiding*, 2011, pp. 59–70.

[131] J. Fridrich and J. Kodovsky, “Rich models for steganalysis of digital images,” *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 3, pp. 868–882, 2012.

[132] M.-J. Kwon, I.-J. Yu, S.-H. Nam, and H.-K. Lee, “CAT-Net: Compression artifact tracing network for detection and localization of image splicing,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 375–384.

[133] B. Bayar and M. C. Stamm, “On the robustness of constrained convolutional neural networks to jpeg post-compression for image resampling detection,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2152–2156.

[134] H. Li, W. Luo, X. Qiu, and J. Huang, “Identification of various image operations using residual-based features,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 1, pp. 31–45, 2016.

[135] G. Mahfoudi, B. Tajini, F. Retraint, F. Morain-Nicolier, J. L. Dugelay, and P. I. C. Marc, “Defacto: Image and face manipulation dataset,” in *2019 27Th european signal processing conference (EUSIPCO)*, 2019, pp. 1–5.

[136] T.-Y. Lin *et al.*, “Microsoft coco: Common objects in context,” in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 2014, pp. 740–755.

[137] X. Liu, Y. Liu, J. Chen, and X. Liu, “PSCC-Net: Progressive spatio-channel correlation network for image manipulation detection and localization,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7505–7517, 2022.

[138] T.-T. Ng, J. Hsu, and S.-F. Chang, “Columbia image splicing detection evaluation dataset,” *DVMM lab. Columbia Univ CalPhotos Digit Libr*, 2009.

[139] B. Wen, Y. Zhu, R. Subramanian, T.-T. T. Ng, X. Shen, and S. Winkler,

"COVERAGE—A novel database for copy-move forgery detection," in *2016 IEEE international conference on image processing (ICIP)*, 2016, vol. 2016-Augus, pp. 161–165, doi: 10.1109/ICIP.2016.7532339.

[140] J. Dong, W. Wang, and T. Tan, "Casia image tampering detection evaluation database," in *2013 IEEE China summit and international conference on signal and information processing*, 2013, pp. 422–426, doi: 10.1109/ChinaSIP.2013.6625374.

[141] H. Guan *et al.*, "MFC datasets: Large-scale benchmark datasets for media forensic challenge evaluation," *Proc. - 2019 IEEE Winter Conf. Appl. Comput. Vis. Work. WACVW 2019*, pp. 63–72, 2019, doi: 10.1109/WACVW.2019.00018.

[142] A. Novozamsky, B. Mahdian, and S. Saic, "IMD2020: A large-scale annotated dataset tailored for detecting manipulated images," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, 2020, pp. 71–80.

[143] Z. Shi, H. Chen, and D. Zhang, "Transformer-Auxiliary Neural Networks for Image Manipulation Localization by Operator Inductions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 9, pp. 4907–4920, 2023, doi: 10.1109/TCSVT.2023.3251444.

[144] H. Wu, J. Zhou, J. Tian, J. Liu, and Y. Qiao, "Robust Image Forgery Detection Against Transmission over Online Social Networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 443–456, 2022, doi: 10.1109/TIFS.2022.3144878.

[145] P. Zhuang, H. Li, S. Tan, B. Li, and J. Huang, "Image Tampering Localization Using a Dense Fully Convolutional Network," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 2986–2999, 2021, doi: 10.1109/TIFS.2021.3070444.

[146] F. Z. El Biach, I. Iala, H. Laanaya, and K. Minaoui, "Encoder-decoder based convolutional neural networks for image forgery detection," *Multimed. Tools Appl.*, vol. 81, no. 16, pp. 22611–22628, 2022, doi: 10.1007/s11042-020-10158-

3.

[147] R. Bai, “Image manipulation detection and localization using multi-scale contrastive learning,” *Appl. Soft Comput.*, vol. 163, no. October 2023, p. 111914, 2024, doi: 10.1016/j.asoc.2024.111914.

[148] R. Thakur and R. Rohilla, “An effective framework based on hybrid learning and kernel principal component analysis for face manipulation detection,” *Signal, Image Video Process.*, 2024, doi: 10.1007/s11760-024-03117-0.

[149] G. Schaefer and M. Stich, “UCID: An uncompressed color image database,” in *Storage and retrieval methods and applications for multimedia 2004*, 2003, vol. 5307, pp. 472–480.

[150] D.-T. T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, “Raise: A raw images dataset for digital image forensics,” in *Proceedings of the 6th ACM multimedia systems conference*, 2015, pp. 219–224, doi: 10.1145/2713168.2713194.

[151] T. Gloe and R. Böhme, “The’Dresden Image Database’for benchmarking digital image forensics,” in *Proceedings of the 2010 ACM symposium on applied computing*, 2010, pp. 1584–1590.

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

[153] J. Thies, M. Zollhöfer, and M. Nießner, “Deferred neural rendering: Image synthesis using neural textures,” *Acm Trans. Graph.*, vol. 38, no. 4, pp. 1–12, 2019.

[154] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.

[155] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation,” *arXiv Prepr. arXiv1710.10196*, 2017.

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

[157] L. Jiang, R. Li, W. Wu, C. Qian, and C. C. Loy, “Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2889–2898.

[158] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, “An introduction to digital face manipulation,” in *Handbook of Digital Face Manipulation and Detection: From DeepFakes to Morphing Attacks*, Springer International Publishing Cham, 2022, pp. 3–26.

[159] F. Matern, C. Riess, and M. Stamminger, “Exploiting visual artifacts to expose deepfakes and face manipulations,” in *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2019, pp. 83–92.

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

[161] V. Kalpana, M. Jayalakshmi, and V. V. Kishore, “Medical Image Forgery Detection By A Novel Segmentation Method With KPCA,” *Cardiometry*, no. 24, pp. 1079–1085, 2022.

[162] M. Tan and Q. V. Le, “EfficientNetV2: Smaller Models and Faster Training,” *Proc. Mach. Learn. Res.*, vol. 139, pp. 10096–10106, 2021.

[163] A. I. Taloba, D. A. Eisa, and S. S. I. Ismail, “A comparative study on using principle component analysis with different text classifiers,” *arXiv Prepr. arXiv1807.03283*, 2018.

[164] B. Schölkopf, A. Smola, and K. R. Müller, “Kernel principal component analysis,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1327, no. 3, pp. 583–588, 1997, doi: 10.1007/bfb0020217.

[165] H. Dang *et al.*, “On the detection of digital face manipulation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, 2020, pp. 5781–5790.

[166] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC)
G. L. Bajaj Inst. of Technology and Management Greater Noida, U. P., India, Oct 18-19, 2019

Copy-Move Forgery Detection using Residuals and Convolutional Neural Network Framework: A Novel Approach

Rahul Thakur[1]
ECE Department
Delhi Technological University
Delhi, India
[1]sirrahulthakur@gmail.com

Rajesh Rohilla[2]
ECE Department
Delhi Technological University
Delhi, India
[2]rajesh@dce.ac.in

***Abstract*—With the sudden advancement in digital image processing, there has been a huge upsurge in the creation of doctored or tampered images with the successful aid of softwares like GNU Gimp and Adobe Photoshop. These manipulated images have become a serious cause of concern, especially in the news, politics and the entertainment sector. Therefore, there is an alarming requirement for a robust image tampering detection system which can distinguish between authentic and tampered images. Common image tampering techniques include copy-move forgery, seam carving, splicing and re-compress. Amongst these techniques, copy-move forgery detection (CMFD) and splicing are dominating the research field due to their complexity stratum and difficulty in detection. In this work, we focus on proposing an efficient splicing detection and CMFD pipeline architecture that focuses on detecting the traces left by various post-processing operations of Splicing and copy-move forgery that are JPEG Compression, noise adding, blurring, contrast adjustment, etc. We use second difference of median filter (SDMFR) on the image as one of the residual and the Laplacian filter residual (LFR) together to suppress image content and focus only on the traces of the tampering operations. The proposed method achieves higher accuracy of 95.97% on the CoMoFoD dataset and 94.26% on the BOSSBase dataset.**

***Index Terms*—copy-move forgery detection, image tampering, median filtering detection, laplacian filter, deep learning, convolutional neural network (CNN)**

I. Introduction

There has been a significant rise in the number of images being manipulated since the advent of the image editing softwares. Consequently, a large variety of image manipulation tools and softwares have been developed which can be further used for malicious activities like mob agitation and fake news spreads through platforms of social media. Recently, even US president Donald Trump was fooled by a DeepFake video of house speaker Nancy Pelosi stammering in a news conference. Such manipulation of images and videos is done with the aim of making the tampering undetectable, or to leave the least amount of traces. Thus, there arises a need to develop even more novel detection methods to find traces of forgery in images and hence successfully classify them as authentic or tampered. Such methods could help in establishing the authenticity of an image and hence ensure their proper relevance. For this, a number of blind forensic techniques had been developed [14-16]. These techniques seek to develop robust systems to detect the traces or fingerprints of tampering in an image.

The most challenging forgery technique is copy-move forgery that encompasses copying a part of the image and pasting it within the same image. Many CMFD related works have been proposed which are primarily based on the two approaches: 1) Key- point based feature matching [9,10] for detecting duplicate regions and 2) Block based feature matching [11-13] which divides the image into overlapping regions. However, these methods have high computational complexity and other drawbacks. Therefore, several works incorporated the use of adaptive oversegmentation [17-19] to divide the image into non-overlapping patches to reduce the computational complexity and then perform feature matching to detect forgery.

However, rather than feature matching parts of images and detecting copy move forgery, we focus on detecting the traces of operations performed after copy-move and splicing to blend it with the original image. In the literature, a lot of work had been done to detect the traces left by image tampering post-processing operations like median filtering [1-6], re-compression [7], and contrast enhancement [8]. Such operations are employed to make the forgery look more convincing, median filtering being the most widely used among them.

Deep learning based approaches are now used in every field of research because of its automatically learning features capability and achieving high accuracy in classification. Various deep learning based approaches were also used for detecting tempering in a image and used to prove better results. Generally, in a deep learning model images are directly given as the input to the network layer and the network automatically learns the features based on the content of the image. But in case of image tempering detection, instead of learning the content based features, the traces left after the tempering operation performed on the image are learned and use to classify the image as authentic or tempered one. To

978-1-7281-1793-5/19/$31.00 ©2019 IEEE

Forensic Science International 312 (2020) 110311

Contents lists available at ScienceDirect

Forensic Science International

journal homepage: www.elsevier.com/locate/forsciint

Review Article

Recent advances in digital image manipulation detection techniques: A brief review

Rahul Thakur, Rajesh Rohilla*

ECE Department, Delhi Technological University, Delhi 110042, India

ARTICLE INFO

Article history:
Received 16 January 2020
Received in revised form 3 April 2020
Accepted 24 April 2020
Available online 7 May 2020

Keywords:
Image manipulation
Image tampering
Convolutional neural network
Deep learning

ABSTRACT

A large number of digital photos are being generated and with the help of advanced image editing software and image altering tools, it is very easy to manipulate a digital image nowadays. These manipulated or tampered images can be used to delude the public, defame a person's personality and business as well, change political views or affect the criminal investigation. The raw image can be mutilated in parts or as a whole image so there is a need for detection of what type of image tampering is performed and then localize the tampered region. Initially, single handcrafted manipulated images were used to detect the only image tampering present in the image but in a real-world scenario, a single image can be mutilated by numerous image manipulation techniques. Nowadays, multiple tampering operations are performed on the image and post-processing is done to erase the traces left behind by the tampering operation, making it more difficult for the detector to detect the tampering. It is seen that the recent techniques that are used to detect image manipulation are based on deep learning methods. In this paper, more focus is on the study of various recent image manipulation detection techniques. We have examined various image forgeries that can be performed on the image and various image manipulation detection and localization methods.

Contents

1. Introduction

Image manipulation has become very convenient nowadays with the help of image editing tools, for example, Adobe Photoshop [1], GNU image manipulation programs (GIMP) [2], Affinity Photo, Paintshop and many more. A large number of images are being produced and with the ease of availability of computer software or

* Corresponding author.
E-mail address: rajesh@dce.ac.in (R. Rohilla).

http://dx.doi.org/10.1016/j.forsciint.2020.110311
0379-0738/

Signal, Image and Video Processing (2024) 18:4811–4820
https://doi.org/10.1007/s11760-024-03117-0

ORIGINAL PAPER

An effective framework based on hybrid learning and kernel principal component analysis for face manipulation detection

Rahul Thakur[1] · **Rajesh Rohilla**[1]

Received: 6 February 2024 / Revised: 22 February 2024 / Accepted: 25 February 2024 / Published online: 2 April 2024

Abstract
Face manipulation is the process of modifying facial features in videos or images to produce a variety of artistic or deceptive effects. Face manipulation detection looks for altered or falsified visual media in order to differentiate between real and fake facial photographs or videos. The intricacy of the techniques used makes it difficult to detect face manipulation, particularly in the context of technologies like DeepFake. This paper presents an efficient framework based on Hybrid Learning and Kernel Principal Component Analysis (KPCA) to extract more extensive and refined face-manipulating attributes. The proposed method utilizes the EfficientNetV2-L model for feature extraction, topped up with KPCA for feature dimensionality reduction, to distinguish between real and fake facial images. The proposed method is robust to various facial manipulations techniques such as identity swap, expression swap, attribute-based manipulation, and entirely synthesized faces. In this work, data augmentation is used to solve the problem of class imbalance present in the dataset. The proposed method has less execution time while achieving an accuracy of 99.3% and an F1 Score of 0.98 on the Diverse Fake Face Dataset (DFFD).

Keywords DeepFake · Face manipulation detection · Deep learning · Hybrid learning · EfficientNetV2

1 Introduction

Face manipulation is the technique of altering a face's features in images or videos in order to produce artistic, cosmetic, or misleading effects. This can entail a variety of adjustments, ranging from minor improvements to significant changes. Face manipulation can be divided in to four primary categories: exchanging identities, swapping expressions, manipulating attributes, and generating completely synthetic faces. Facial identity manipulation is the process of replacing one person's face with another. The most widely used methods for manipulating facial identities are FaceSwap[1] and DeepFakes.[2] Facial expression manipulation is the process of replacing one person's facial expressions with another while preserving the facial identity. The two most popular methods for manipulating facial expressions are Face2Face [21] and NeuralTextures [22]. While the DeepFakes and NeuralTextures approaches are based on deep learning techniques, the FaceSwap and Face2Face approaches are based on computer graphics techniques. Face attribute-manipulated images involve identifying alterations made to specific facial features or characteristics such as gender, age, hair, beard, and glasses. To generate attribute-manipulated images, the two most popular methods used are FaceAPP[3] and StarGAN [3]. Synthetic face images refer to artificially produced facial images created via computer graphics, deep learning, or other digital methods. These are not photographs of actual people; rather, they are the result of models or algorithms. To generate entire synthesized faces, the popular methods used are PGGAN [9] and StyleGAN [10]. Face manipulation techniques can be used for more controversial applications, like producing misleading content known as "DeepFakes," or for legitimate purposes, such as retouching photographs for aesthetic reasons. The term "DeepFakes" encompasses digitally fabricated content created using deep learning techniques. It gained notable prominence in late 2017 when a Reddit user known as

[1] Faceswap:https://github.com/MarekKowalski/FaceSwap.
[2] Deepfakes:https://github.com/deepfakes/faceswap.

✉ Rajesh Rohilla
rajesh@dce.ac.in

Rahul Thakur
rahulthakur@dtu.ac.in

[1] Electronics and Communication Engineering, Delhi Technological University, Delhi, Delhi 110042, India

[3] FaceApp:https://faceapp.com/app.

LFRViT: Laplacian Filter Residual-based Vision Transformer for Multiple Image Forgery Detection

2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT) | 979-8-3503-7024-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCCNT61001.2024.10724100

1st Rahul Thakur
Electronics and Communication Engineering Department
Delhi Technological University
Delhi, India
rahulthakur@dtu.ac.in

2nd Rajesh Rohilla
Electronics and Communication Engineering Department
Delhi Technological University
Delhi, India
rajesh@dce.ac.in

***Abstract*— When producing a manipulative image, a forger can alter an image through a variety of image tampering techniques. Due to the need to test for various image editing operations and alterations, there has been a significant interest in developing a universal image forgery detection approach that can detect multiple tampering operations performed over an image. This paper presents a comprehensive forensic method for detecting manipulation using Vision Transformer. We present a novel network architecture called Laplacian Filter Residual-based Vision Transformer (LFRViT) that can automatically learn features for detecting manipulation from training data. Vision Transformers, as they are now designed, primarily to learn features that represent the content of an image rather than features that detect manipulation. To address this problem, we have created a novel type of model specifically designed to suppress the image's content and dynamically acquire features for detecting manipulations. By conducting a sequence of studies, we provide evidence that our proposed method can automatically learn the features to identify various image alterations. The experimental findings demonstrate that our proposed model, LFRViT, is capable of autonomously identifying various types of manipulations with an accuracy of over 99%.**

Keywords—Vision Transformer, Laplace, Image Forgery Detection, Deep Learning, Image Forensics

I. INTRODUCTION

In the digital age, as images are used extensively in documentation, entertainment, and communication, it is critical to assure their authenticity. However, image manipulation and forgery have become more sophisticated and accessible with the development of image editing software [1]. This has sparked questions about how dependable and credible the visual content that is being shared online and in other media is.

The field of research and technology known as "multiple image forgery detection" is devoted to detecting situations in which multiple images are combined or altered to produce an inaccurate or misleading representation. Multiple-image forgery detection exposes instances of manipulation or tampering by examining relationships and inconsistencies between multiple images, in contrast to traditional single-image forgery detection, which focuses on finding a single tampering operation within an image [1].

Effective techniques to identify manipulated or tampered images are more important than ever due to the spread of social media sites, online news sources, and digital archives. These kinds of images can be used to disseminate false information, sway public opinion, or trick people or institutions. Thus, in the larger field of digital forensics and image analysis, the creation of efficient methods and algorithms for multiple image forgery detection has become essential research. Through the application of advances in machine learning, computer vision, and signal processing, scholars and practitioners aim to improve the veracity and authenticity of visual content on digital platforms [2].

In this study, we have considered four different types of image tampering operations. These tampering operations are applied to each original image to have a manipulated image. The four different types of tampering operations are: Additive White Gaussian Noise (AWGN), resampling, median filtering, and gaussian blurring. Fig. 1, shows the different tampering operations performed over the original image, taken from the RAISE dataset.

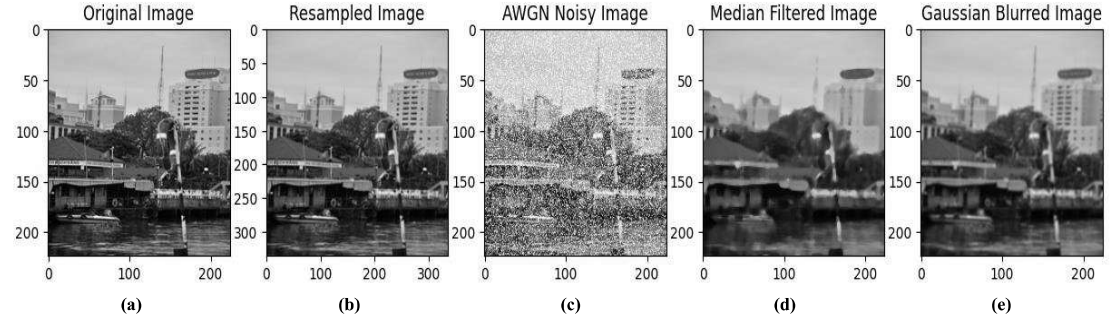


Fig.1. From RAISE database, **(a)** original image and different operations are performed on this original image, **(b)** resampled image with a scaling factor of 1.5, **(c)** AWGN noisy image with standard deviation of 2, **(d)** median filtered image with a 5 × 5 kernel size and **(e)** Gaussian blurred image with 5 × 5 kernel and σ = 1.1.

Rahul Thakur

Thesis Rahul Thakur.pdf

Delhi Technological University

Document Details

Submission ID
trn:oid:::27535:89000411

Submission Date
Apr 1, 2025, 12:45 PM GMT+5:30

Download Date
Apr 1, 2025, 12:50 PM GMT+5:30

File Name
Thesis Rahul Thakur.pdf

File Size
2.8 MB

123 Pages

33,937 Words

190,408 Characters

9% Overall Similarity

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

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 10 words)

Exclusions

- 4 Excluded Sources

Match Groups

183 Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

3% Internet sources
5% Publications
3% Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

Replaced Characters
33 suspect characters on 20 pages
Letters are swapped with similar characters from another alphabet.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

183 Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

3% Internet sources
5% Publications
3% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	
dspace.dtu.ac.in:8080		<1%
2	Submitted works	
University of Macau on 2023-05-20		<1%
3	Publication	
Zenan Shi, Haipeng Chen, Dong Zhang. "Transformer-Auxiliary Neural Networks f...		<1%
4	Internet	
technodocbox.com		<1%
5	Submitted works	
University of Warwick on 2017-09-05		<1%
6	Internet	
www.mdpi.com		<1%
7	Internet	
jusst.org		<1%
8	Publication	
Debashis Das Chakladar, Pradeep Kumar, Partha Pratim Roy, Debi Prosad Dogra, ...		<1%
9	Publication	
Belhassen Bayar, Matthew C. Stamm. "A Deep Learning Approach to Universal Im...		<1%
10	Publication	
Shefali Patil, Puja Padiya, Sangita Chaudhari. "A Detailed Analysis of Image Forge...		<1%

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis Development of Framework for Image Manipulation Detection

Total Pages 123

Name of the Scholar Rahul Thakur

Supervisor Prof. Rajesh Rohilla

Department of Electronics and Communication Engineering

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

Software used: Turnitin Similarity Index: 9% Total Word Count: 33937

Date:

Candidate's Signature **Signature of Supervisor**

Rahul Thakur

Assistant Professor
Department of Electronics and Communication Engineering
Delhi Technologocal University
Shahbad Daulatpur, Main Bawana Road,
Delhi-110042, India
Ph: +91-8447480977
E-mail: rahulthakur@dtu.ac.in,
sirrahulthakur@gmail.com

Qualification:

- Ph.D.: Delhi Technological University, Delhi.
- M.Tech.: Netaji Subhas University of Technology (East Campus), Delhi.
- B.Tech.: Guru Gobind Singh Indraprastha University, Delhi.
- 12^{th}, CBSE Board, Delhi.
- 10^{th}, CBSE Board, Delhi.

Honors/Awards:

- Research excellence award, at Delhi Technological University (DTU), Delhi.
- Qualified for JRF & Assistant Professor, UGC NET (Electronic Science).
- Qualified, GATE (Electronics and Communication).

Research Profile:

- h- index: 03; i-10 index: 02; Citations: 147
- Orcid ID: 0000-0002-2672-6067; Scopus ID: 57216845790
- Web of Science Researcher ID: KRP-8282-2024
- Google Scholar:
  https://scholar.google.com/citations?user=e51fOvMAAAAJ&hl=en&oi=sra
- Webpage: https://sites.google.com/view/rahulthakur/home

Research Interest:

- Multimedia Forensics
- Digital Image Processing
- Computer Vision
- Machine Learning and Deep Learning