Passive Forensics Based Digital Image Forgery Detection Techniques

A thesis submitted

in fulfillment of the requirement for the award of degree

of

Doctor of Philosophy

Submitted by

Navneet Kaur

Registration Number: 901606035

Under the Supervision of

Dr. Kulbir Singh
Professor, ECED

Dr. Neeru Jindal
Assistant Professor, ECED

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY, PATIALA-147004

July 2022

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "**Passive Forensics Based Digital Image Forgery Detection Techniques**", for the award of degree of **Doctor of Philosophy** in Electronics and Communication Engineering Department (ECED), Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision and guidance of Dr. Kulbir Singh, Professor, ECED, and Dr. Neeru Jindal, Assistant Professor, ECED, Thapar Institute of Engineering and Technology, Patiala.

The results presented in this thesis have not been submitted in part or in full to any other University or Institute for the award of any degree or diploma.

Navneet Kaur.

Navneet Kaur

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Dr. Kulbir Singh
Professor, ECED
TIET, Patiala, India

Dr. Neeru Jindal
Assistant Professor, ECED
TIET, Patiala, India

ACKNOWLEDGEMENTS

First and foremost, I bow down to the Lord Almighty, whose grace guided me from the very inception to the completion of this research work. Each moment during the course of this work, I experienced the Grace of God, who continuously inspired me to move forward, opened before me unexpected avenues, and enlightened my thoughts with his wisdom.

I would like to express my deepest gratitude to my supervisors **Dr. Kulbir Singh**, Professor, and **Dr. Neeru Jindal**, Assistant Professor, Electronics and Communication Engineering Department (ECED), Thapar Institute of Engineering and Technology (TIET), Patiala, for their noble guidance and encouragement throughout the course of this research work. Their invaluable feedback, constructive criticism, and approachable attitude have always provided the required impetus for this research work. This work could not have been accomplished without their consistent efforts.

I am highly grateful to Dr. Alpana Agarwal, Professor, and Head of Department, ECED, TIET, Patiala, for her continuous support and encouragement in my research work. I am deeply indebted to my doctoral committee members Dr. Amit Kumar Kohli, Associate Professor, Dr. Vinay Kumar, Associate Professor in ECED, TIET, Patiala and Dr. M. D. Singh, Associate Professor in Electrical and Instrumentation Engineering Department (EIED), TIET, Patiala for their valuable suggestions throughout my research work.

I owe a debt of gratitude to all academic, administrative, and technical staff from TIET, Patiala, for their motivation, timely assistance, and acquaintance throughout my candidature. I am also immensely thankful to my fellow researchers, Kanwarpreet Kaur, Amit Kumar, Hari Shankar, Gurinder Singh, Gittaly Dhingra, Mukti Yadav, and Aakanksha Lakhanpal for assisting and cheering me in all situations.

This acknowledgment would be incomplete without mentioning the painstaking efforts and patience of my family. No words are adequate to express my indebtedness to my parents, brother, and husband for their constant love and support, which kept me motivated and guided me during hard times. Their belief in my talent enabled me to accomplish this research work. I am also obliged to my in-laws for their understanding and blessings. Finally, I wish to convey my regards to everyone who directly or indirectly aided me in this endeavor. Your prayers and good wishes gave me the strength to persevere.

Navneet Kaur

ABSTRACT

Digital images are an essential part of our daily lives. They are beneficial in fields such as education, social media, newspapers, magazines, and others. Nowadays, images are the primary mode of communication. Images can be easily forged due to the rapid advancement of digital image technology and the availability of a wide range of digital image forgery devices. As a result, the main issue is the authenticity of digital images. The requirement to authenticate digital images created opportunities for an exciting field of research that focused on the development of algorithms for forgery detection. Thus, digital image authentication is critical, and the current field of study seeks to validate digital image authenticity. The proposed research aims to create a passive algorithm that detects two types of image forgery: Copy-move Forgery (CMF) and Image Splicing Forgery (ISF). In the present research, a passive strategy for forgery detection is used, which does not require any previous image data and is thus also known as the "blind strategy".

This research proposes an improved method for detecting CMF that is both efficient and sophisticated. Even though CMF is a prevalent type of forgery, detecting it is difficult because the copied part belongs to the same image and can thus exhibit similar features to that of the image. Thus, in the proposed work, block-based technique i.e. Adaptive Over-segmentation (AS), and keypoint-based techniques i.e. Accelerated KAZE (AKAZE), and Scale-Invariant Feature Transform (SIFT) are combined to detect CMF, which makes the proposed algorithm more computationally efficient and accurate for the detection and localization of single as well as multiple forgeries. The experimental results show that the proposed technique is robust against various attacks i.e. rotation, scaling, JPEG compression, and noise addition, and proved better when compared with the other existing techniques with an improved F_2 score of 99.80%, 99.81%, 99.35%, and 99.82% on benchmark datasets i.e. Image Manipulation Dataset (IMD), MICC-F220, COVERAGE, and GRIP, respectively.

Further research in this thesis is being conducted to detect ISF, which is also one of the most commonly used image manipulation techniques. In the proposed method, Markov features from both the Local Binary Pattern (LBP) and Discrete Wavelet Transform (DWT) domains are extracted and combined to efficiently detect ISF. For detecting ISF, it is crucial to capture the forgery-introduced artifacts, as image splicing produces sharp edges in a forged image. Moreover, the edges introduced by forgery differ from their neighbors, and thus the relationships between the spliced and original parts can be used to expose image forgery. To

describe these relationships, the proposed method makes use of the Markov Transition Probability Matrix (TPM). The experimental results performed on six datasets indicate that the proposed approach offers better results than the existing techniques in terms of various performance metrics.

A hybrid approach based on Discrete Fractional Cosine Transform (DFrCT) and LBP is proposed in this thesis to detect copy-move and splicing forgeries simultaneously. The additional parameter i.e. fractional parameter of DFrCT is utilized to improve the accuracy and LBP is used to highlight the tampering artifacts effectually. Additionally, localization is performed on both the copy-move and spliced images to localize the image's duplicated areas. The efficacy of the proposed scheme is confirmed by extensive simulations on six benchmark datasets, namely CASIA v1.0, GRIP, CASIA v2.0, IMD, COVERAGE, and Columbia, which achieved accuracy rates of 99.67%, 99.23%, 99.76%, 98.81%, 95%, and 98.17%, respectively, that surpasses existing techniques.

A Contrast Limited Adaptive Histogram Equalization (CLAHE) based Convolutional Neural Network (CNN) model is presented for effectively solving the issue of CMF detection. The CLAHE algorithm makes the hidden features of the image visible, as some of them are hard to detect in CMF. The proposed work primarily focuses on improving the performance parameters of the forgery detection, which are superior to the existing techniques with reduced False Negative Rate (FNR) of 0.0132, 0.0179, 0.0000, and 0.0010 on MICC-F220, GRIP, IMD and MICC-F2000 datasets, respectively. Also, the robustness of the proposed technique is demonstrated against several attacks like scaling, noise addition, JPEG compression, and rotation. Furthermore, statistical analysis tests such as Analysis of Variance (ANOVA) and cross-dataset performance are used to validate the effectiveness of all the proposed strategies. Future work could be devoted to expanding the current work to other object detection and localization applications, such as medical image analysis, face detection, and so on.

LIST OF PUBLICATIONS

Published Journal Publications:

[P.1] N. Kaur, N. Jindal, and K. Singh, "A passive approach for the detection of splicing forgery in digital images," *Multimed. Tools Appl.*, vol. 79, no. 43, pp. 32037-32063, 2020, doi: 10.1007/s11042-020-09275-w. **(SCIE-Indexed, Impact factor: 2.577)**

[P.2] N. Kaur, N. Jindal, and K. Singh, "Efficient hybrid passive method for the detection and localization of copy-move and spliced images," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 29, no. 2, pp. 561–582, 2021, doi: 10.3906/elk-2001-138. **(SCIE-Indexed, Impact factor: 0.853)**

[P.3] N. Kaur, N. Jindal, and K. Singh, "An improved approach for single and multiple copy-move forgery detection and localization in digital images," *Multimed. Tools Appl.*, 81, pp. 38817–38847, 2022, doi: 10.1007/s11042-022-13105-6. **(SCIE-Indexed, Impact factor: 2.577)**

[P.4] N. Kaur, N. Jindal, and K. Singh, "A deep learning framework for copy-move forgery detection in digital images," *Multimed. Tools Appl.*, pp. 1-28, 2022, doi: 10.1007/s11042-022-14016-2. **(SCIE-Indexed, Impact factor: 2.577)**

Communicated Journal Publications:

[C.1] N. Kaur, N. Jindal, and K. Singh, "Passive Image Forgery Detection Techniques: A Review, Challenges, and Future Directions," *Wirel. Pers. Commun.* **(SCIE-Indexed, Impact factor: 2.017)**

LIST OF ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
AKAZE	Accelerated KAZE
AMFE	Approximated Machado Fractional Entropy
ANOVA	Analysis of Variance
AP	Average Precision
AS	Adaptive Over-Segmentation
ASIFT	Affine Scale-Invariant Feature Transform
AUC	Area Under the ROC Curve
AWGN	Additive White Gaussian Noise
BDCT	Block Discrete Cosine Transform
BFM	Bessel Fourier Moments
BI	Bilinear Interpolation
BOF	Bag-of-Features
BRISK	Binary Robust Invariant Scalable Keypoints
C2RNet	Coarse-to-Refined Convolutional Neural Network
CA	Cellular Automata
CGH	Computer-Generated Holographic
CLAHE	Contrast Limited Adaptive Histogram Equalization
CMF	Copy-Move Forgery
CMFs	Copy-Move Forgeries
CMFD	Copy-Move Forgery Detection
CNN	Convolutional Neural Network
DAFMT	Discrete Analytical Fourier-Mellin Transform
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DFrCT	Discrete Fractional Cosine Transform
DOCT	Discrete Octonion Cosine Transform
DoG	Difference of Gaussian
DPCET	Discrete Polar Complex Exponential Transform
DRHFM	Discrete Radial Harmonic Fourier Moments
DWT	Discrete Wavelet Transform
DyWT	Dyadic Wavelet Transform

EPO	Emperor Penguin Optimization
FC	Fully Connected
FCM	Fuzzy C-Means
FED	Fast Explicit Diffusion
FGBO	Football Game-Based Optimization
FNR	False Negative Rate
FPR	False Positive Rate
FQZM	Fractional Quaternion Zernike Moments
FRFT	Fractional Fourier Transform
G-mean	Geometric Mean
GIMP	GNU Image Manipulation Program
GLCM	Grey Level Co-Occurrence Matrix
GLRLM	Gray Level Run Length Matrix
HE	Hamming Embedding
HH	Hue Histogram
HHT	Hilbert–Huang Transform
HWHT	Hybrid Wavelet Hadamard Transform
IDCT	Inverse Discrete Cosine Transform
IFDTs	Image Forgery Detection Techniques
IMD	Image Manipulation Dataset
ISF	Image Splicing Forgery
ISFD	Image Splicing Forgery Detection
kNN	K-Nearest Neighbor
KS	Kolmogorov Smirnov
LBP	Local Binary Pattern
LIOP	Local Intensity Order Pattern
LoG	Laplacian of Gaussian
LSH	Locality-Sensitive Hashing
MAP	Mean Average Precision
MBDCT	Multi-Size Block DCT
MCC	Mathews Correlation Coefficient
MIFT	Mirror Reflection Invariant Feature Transform
MLDB	Modified Local Difference Binary

MROGH	Multi-Support Region Order-Based Gradient Histogram
ORB	Oriented FAST and Rotated BRIEF
PCA	Principal Component Analysis
PCT	Polar Cosine Transform
PCET	Polar Complex Exponential Transform
QDCT	Quaternion DCT
QPCET	Quaternion Polar Complex Exponential Transform
RANSAC	Random Sample Consensus
RBF	Radial Basis Function
RGB-D	RGB and Depth
ReLU	Rectified Linear Unit
RMT	Random Matrix Theory
ROC	Receiver Operating Characteristics
SGDM	Stochastic Gradient Descent with Momentum
SIFT	Scale-Invariant Feature Transform
SLICO	Simple Linear Iterative Clustering
SPT	Steerable Pyramid Transform
SSIM	Structural-Similarity
STD	Standard Deviation
SURF	Speeded-Up Robust Features
SVD	Singular Value Decomposition
SVM	Support Vector Machine
SWT	Stationary Wavelet Transform
TNR	True Negative Rate
TPM	Transition Probability Matrix
TPR	True Positive Rate
TSF	Two-Stage Filtering
UWT	Undecimated Wavelet Transform
WLD	Weber Law Descriptor

GLOSSARY OF SYMBOLS

$\gamma(t)$	Scaling function
$\xi(t)$	Basic wavelet function
$eAl(l)$	Approximation coefficient of DWT
$eDl(l)$	Detailed coefficient of DWT
$L(W,V,\sigma)$	Scale-space of the image
$Gf(W,V,\sigma)$	Gaussian function
$Y(W,V)$	Input image
W,V	Coordinates of the input image
σ	Standard deviation of Gaussian function
$D(W,V,\sigma)$	Difference of Gaussian scale-space function
$D(w)$	Second-order Taylor expansion
$mg(W,V)$	Magnitude gradient of image
$\theta(W,V)$	Orientation of image
$Y*(W*,V*)$	Image after rotation
θ'	Rotation angle
$TT^{-1}\{\cdot\}$	Inverse transformation function
TT	Affine matrix
$\phi(m, p)$	Kernel function
γ	Free parameter in SVM
$\{m_i, n_i\}$	Training dataset
Ω	Number of correctly detected forged pixels
Ω_1	Total detected forged pixels
Ω_2	Number of forged pixels in the ground-truth forged image
L_{FE}	Low-frequency energy
H_{FE}	High-frequency energy
AC_3	Approximation coefficient of DWT
$DC_{kk}, HC_{kk}, VC_{kk}$	Detailed coefficients of DWT

L_{FC}	Percentage of low-frequency coefficients
S_{PP}	Size of superpixels
div	Divergence
LI	Luminance of the image
$C(x, y, t)$	Conductivity function
LI_{σ}	Smoothed Gaussian form of image
∇LI_{σ}	Gradient of LI_{σ}
Φ	Contrast factor
$Z_z(xx, yy)$	Difference arrays for LBP and DWT
D_{Pi}^{a}	Hermite-Gauss eigenvectors of the Fourier matrix
α	Fractional parameter of DFrCT
$(w+v)^{th}$	Order of keypoints
$I(w, v)$	Intensity
C_e	Centroid
ψ	Orientation of keypoints
$a\tan(\cdot)$	Arctangent function
E_j	Response of kernel
(q, R)	LBP parameters
$T(\cdot)$	Remapping algorithm
$p(i)$	Value of random pixel
D_w	Previous layer descriptor
W_w, H_w	Width and Height of Kernel
mi	Iteration number
η	Contribution of the preceding gradient step to current iteration
ϖ	Parameter factor
$L(\varpi)$	Loss function
$\nabla L(\varpi)$	Gradient of loss function

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This chapter aims to provide a brief overview of the fundamental concepts that serve as the foundation for arousing interest in delving into the depths of the research performed in this thesis. It discusses the historical viewpoint as well as terminologies associated with image forgery detection techniques.

1.1 Preamble

In the contemporary technological era, enormous advancements are created daily. According to history, images are the most dependable medium for information exchange. There is a saying that says, "A picture is worth a thousand words," which indicates that an image may portray a thought faster and more efficiently than several written words [1]. However, if the image's information is altered, it may transmit false information. Digital photographs are widely used as information carriers in our daily lives [2], [3]. Images, for example, are employed in research papers to represent the proposed idea and the experimental results. Other fields in which digital images play important roles include animation creation, computer-aided design, crime scene restoration, biomedicine, website designing, radar imaging [4], education, online insurance claims, digital magazine articles, facial expression reorganization [5], the newspaper industry, and medical [6]–[8]. In all of these domains, the use of digital photographs has resulted in significant advances. In most cases, the legitimacy of an image is a must. When the information of a digital image is manipulated, or the entire picture is produced utilizing graphic design software for malicious purposes, the image is referred to as a forged image, and the process of making such a forged image is termed digital image forgery [3], [9].

The availability of a huge number of low-cost software tools has made image forgery a simple operation. Among the multitudes of photo editing software applications available, Photoshop, CorelDraw, GNU Image Manipulation Program (GIMP), and others are some of the most popular. These tools can be used to create graphics, alter photos, and apply unique effects to images [10]. Nowadays, the amount of images or messages posted on various messaging apps and social networking sites like WhatsApp, Facebook, LinkedIn, and many others, has risen tremendously due to the proliferation of cloud-computing-based image sharing and various

image editing platforms. Cloud computing is an information technology (IT) prototype that offers fast access to shared, customizable structure resource pools and much more advanced features that are usually accessible via the Internet with little administrative effort [11]–[15]. There are around 3.725 billion active social media users [16]. In contrast, political parties often use social networking sites for election campaigns. In recent times, numerous counterfeit posts on the COVID-19 epidemic have been published on social media throughout the world [17]. According to these figures, when counterfeit images that are purposely generated with harmful intentions are posted on various social media platforms, they may reach millions of individuals in a short period. The repercussions of images containing misinformation can be serious since such posts can alter people's perceptions of the truth, which can occasionally result in riots and other social upheavals. As a result, image forgery has become an ever-increasing challenge for both the image forensic department and the general public.

1.2 Historical Viewpoint of Image Forgery

Image forgery has a long history, dating back to the 1800s. Photographs were altered within a few decades after Niepce made the first image in 1814. A brief history of image manipulation dating back to the 1800s is provided in this section. Hippolyte Bayard, a French photographer, who produced the first forged image in 1840, is famous for a photograph of himself committing suicide, as shown in Figure 1.1 [18]. The following paragraphs illustrate some image forgery instances that have impacted ordinary people.

Figure 1.1: The first forged image [18]

In an image taken by renowned photographer Mathew Brady in 1865, General Sherman is seen standing with his generals, as shown in Figure 1.2. In this figure, General Francis P. Blair (shown on the far right) was added to the picture because he was not present at the time. His photo was derived from the second picture shown, which was captured at the same sitting [19].

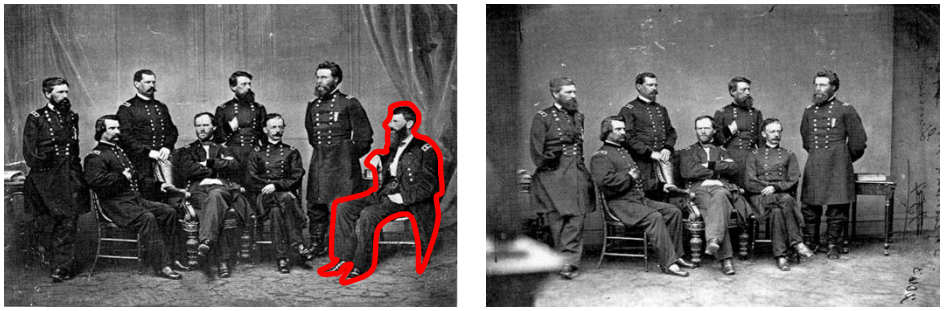


Figure 1.2: General Francis P. Blair was added to a photograph of General Sherman posing with his generals taken in 1865 [19]

As shown in Figure 1.3, Mao Tse-tung, depicted on the far right in a doctored picture, had Po Ku eradicated from the original photo in 1936, when Po Ku fell out of favor with Mao [19].

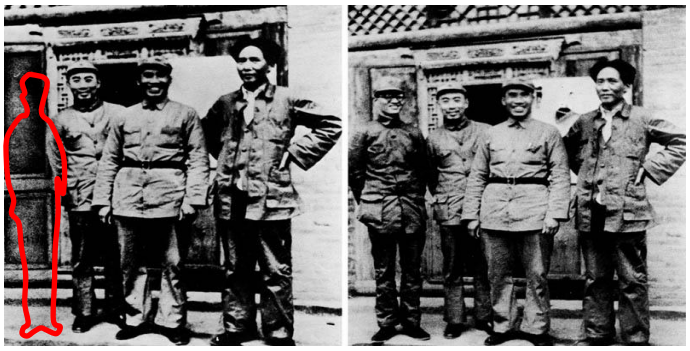


Figure 1.3: Po Ku has been removed by Mao Tse-tung in 1936 [19]

The doctored portrait of Canadian Prime Minister William Lyon Mackenzie King and Queen Elizabeth at Banff, Alberta, is displayed in Figure 1.4. King George VI was deleted from the original picture. This image appeared on a Prime Ministerial election poster. It is believed that Prime Minister has altered the picture since a photograph of only him and the Queen cast him in a more dominant light [19].

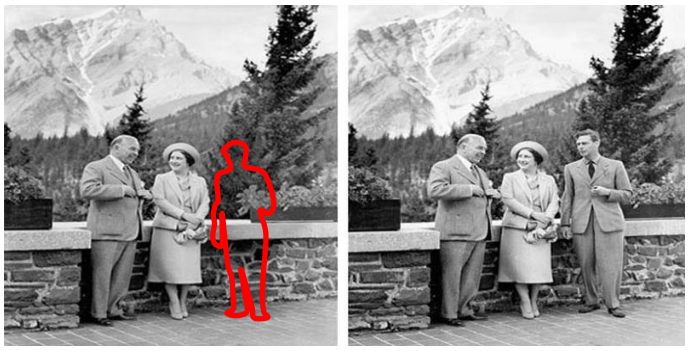
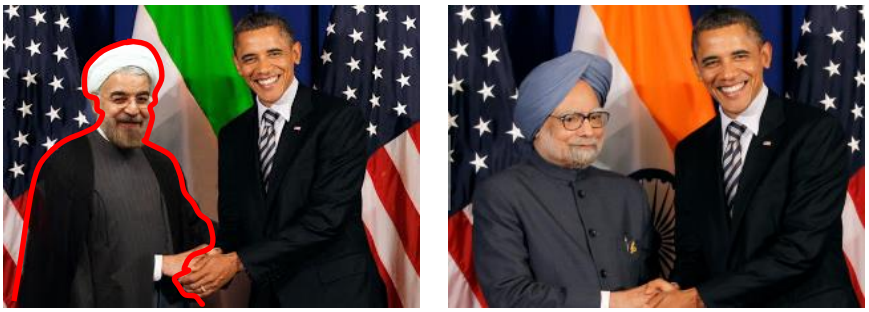


Figure 1.4: In 1939, the Canadian Prime Minister removed King George VI [19]

As indicated in Figure 1.5, Mary Ann Vecchio is shouting as she bends down over the corpse of student Jeffrey Miller at Kent State University in this Pulitzer Prize-winning image by John Filo. The Kent State Attacks happened when National Guardsmen opened fire on a group of protestors, killing four and injuring nine. The snap included a disrupting fencepost beyond Mary Ann Vecchio's head; however, it was deleted by an anonymous picture editor in the 1970s. The altered photograph was further featured in Life magazine and other media [19].

Figure 1.5: Fencepost Removal from the Kent State Massacre in 1970 [19]

BuzzFeed News Reporter [20], uploaded a photoshopped image on Twitter of Iranian President Hassan Rouhani shaking hands with US President Barack Obama, yet the two had never met. The original photograph, according to the report, was of Obama's meeting with former Indian Prime Minister Manmohan Singh in 2011, as depicted in Figure 1.6. In 2014, a California lawyer generated multiple false photographs [21] by combining her photo with photos of other celebrities, including President Obama. She then uploaded these photographs on her professional website to achieve popularity by revealing that she had met significant people. She was later sentenced for this crime.



(a) (b)

Figure 1.6: In 2011 (a) forged picture of Obama meeting the Iranian President, (b) original image of Obama meeting former Indian Prime Minister [20]

In 2017, one of the Indian politicians from Bihar state tweeted a faked photograph [22] of a political rally on social media to indicate that the event was full overcrowded. The photograph (Figure 1.7) instantly drew the attention of Twitter users and the politician who published it was criticized.

Figure 1.7: Lalu Prasad, an Indian politician from Bihar, reportedly uploaded a faked photograph of his rally in 2017 [22]

In brief, the number of manipulated images is rapidly increasing due to a variety of factors, including (a) the accessibility of low-cost and user-friendly photo editing tools, some of which are even free; (b) the cheaper cost of devices such as computers, laptops, smartphones, and so on, which enabled the user to tamper with digital pictures with little knowledge; and (c) the accessibility of online courses on multimedia and image processing, which made people more resourceful.

1.3 Digital Image Forgery

The process of creating a false picture by modifying the data of the main picture and passing it as original for malicious purposes is known as digital image forgery. It is an essential issue as digital image validation has recently attracted much attention since digital multimedia is widely employed in many safety applications or enterprises. As a result, effective movements for identifying transformed images are being studied. This research is significant since such detection algorithms may be employed in high-security businesses, like immigration, where digital pictures and e-passports interrelate, and then photographs saved on passport chips must be verified [3], [10], [19]. Consequently, verifying digital photographs is important, making forensic science essential.

Forensics is an application of knowledge and technology in courtrooms to investigate and verify legitimacy in criminal cases. It is classified into two parts: analog forensics and digital forensics. Figure 1.8 depicts the forensics taxonomy.

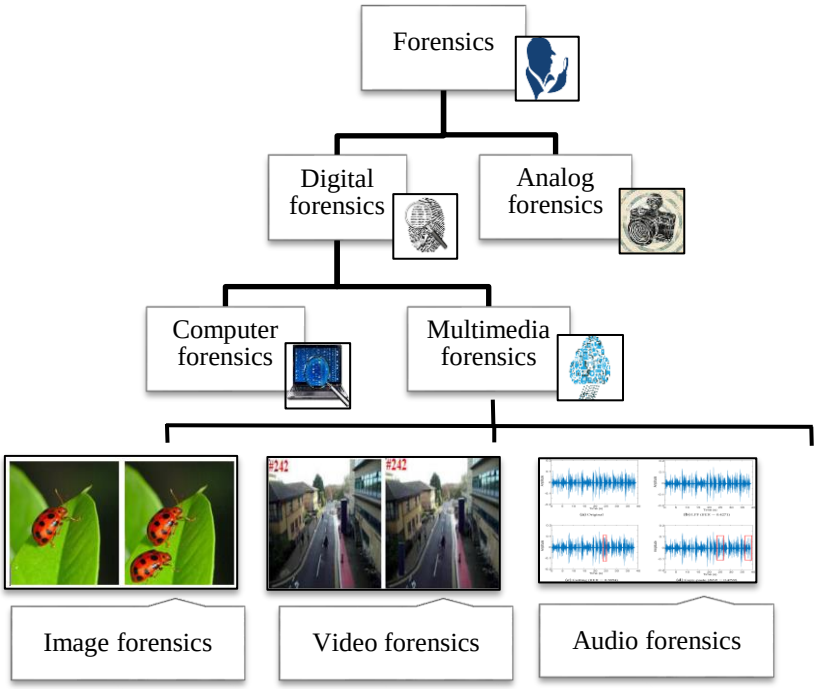


Figure 1.8: Taxonomy of forensics

Analog forensics emphasizes identifying tangible evidence in reality, which cannot be incorrect or recuse itself. Today, digital forensics, categorized into multimedia and computer forensics, is the primary focus of forensic research. Computer forensics is applied when computers are employed in real-world illegal activities, and experts seek to retrieve proof from computers. Therefore, this proof should be reliable enough to stand up in court. In terms of digital evidence, multimedia forensics is just like computer forensics; however, in this forensics, digital evidence is linked to the external world and cannot be manipulated using machines. Its goal is to recover some of the misplaced consistency of digital proof by developing tools that provide apparent signs of any modification that happened [3], [23], [24]. Image forensics is concerned with identifying proof of forgeries and primarily with increasing the trustworthiness of digital pictures. It tackles the topic of image authentication or image ancestries and gives reliable responses regarding the foundation and validity of digital photographs. Video forensics detects video tampering, which is subsequently exploited to generate false videos for unlawful reasons. Counterfeiting digital video is harmful since the video has long been an essential tool in the fight against crime. The field of forensics that deals with the audio collection, scrutiny, and assessment are known as audio forensics. They can finally be recognized as evidence in criminal law courts or as a source of government inquiries into corruption, accidents, or other civil actions [3], [25]–[28].

1.4 Digital Image Forgery Detection Techniques

The process of authenticating pictures and determining which parts of an image have been altered is termed image forgery detection. The picture's statistical properties are modified when a digital photo is forged. As a result, the statistical properties of the manipulated area are likely to differ from those of the original. The statistical features of each part of the picture are calculated and compared to spot tampered portions [29]. Active approaches and passive approaches are two basic kinds of digital image forgery detection techniques (IFDTs), as shown in Figure 1.9.

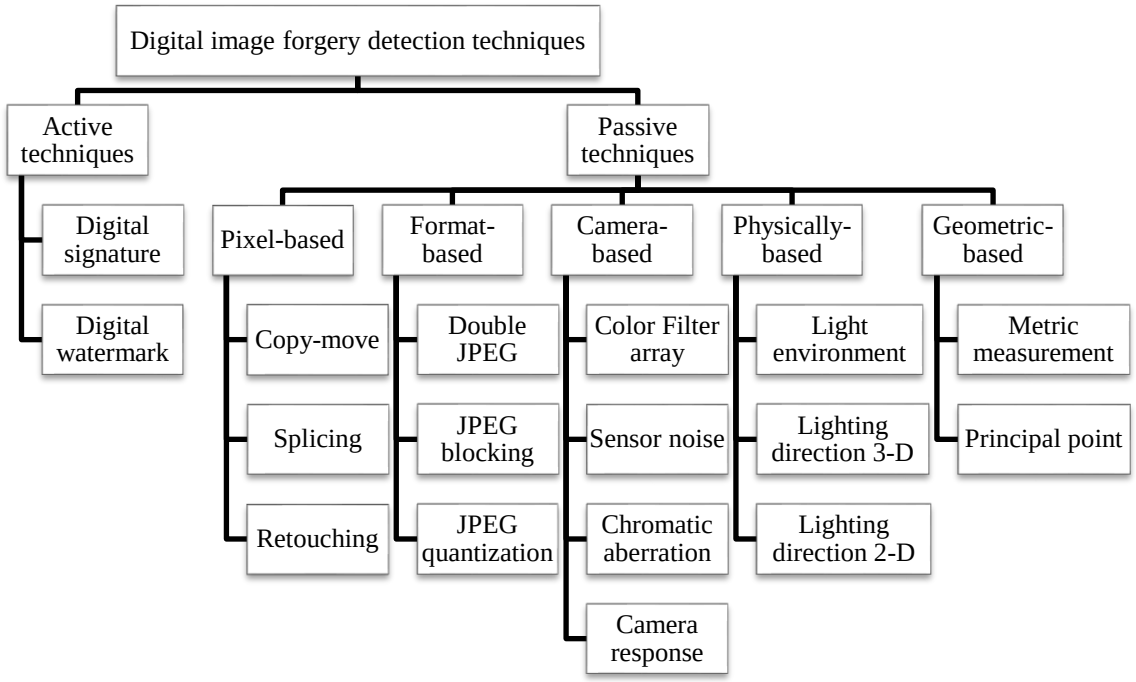


Figure 1.9: Taxonomy of digital image forgery detection techniques

1.4.1 Active techniques

In active techniques, pre-processing procedures such as creating digital signatures or inserting watermarks into images are done during the image creation process. The secret message embedded in a fake image cannot be retrieved. So, the validity of the images may be confirmed in this manner. Active methodologies are subdivided into digital watermarking and digital signature procedures [9], [10], [30]–[32].

- *Digital Signature:* This procedure extracts the image's characteristics while capturing the image. The signature is produced again when the image is authenticated using a similar process. The uniqueness of the picture can then be confirmed by comparison.

To find the essential feature components, adaptive Harris corner and wavelet transform is used. The forensic signature is created using arithmetic feature modules based on neighborhood data. Fisher norm and Forensic signature are then utilized to determine the image's validity [9], [10], [30].

- *Digital Watermark:* In this, the source creates a watermark picture that is subsequently embedded into the picture to form a watermarked picture. The watermarked picture is further used to recognize the watermark contained to confirm the image's validity. Development and testing of a Computer-Generated Holographic (CGH) coding scheme for watermark inserting are described. The hologram is embedded using a blind additive embedding process [9], [10], [30], [31], [33]–[35].

The main limitation of active techniques is that a watermark or signature must be placed during the recording process, limiting it to specially equipped digital cameras. The need for specialized hardware limits the domains in which these techniques may be used. Furthermore, hidden data may degrade image quality in some circumstances. So, the necessity of prior knowledge of the original image is a shortcoming of active schemes.

1.4.2 Passive Techniques

Although researchers used to prefer active techniques, passive techniques have gained more importance in recent years since they do not insert any secondary information like watermark or signature into the image. These techniques authenticate image manipulation without any previous knowledge of the original image or its attributes. In passive procedures, the statistics and content of the given picture are used to validate the image's authenticity. The verification procedure is carried out only based on the image's information, without any other information. Since passive strategies validate the forged picture based on accessible knowledge, they are also referred to as blind procedures. The methodologies for detecting passive image forgery are roughly grouped as follows: geometric-based, physically-based, format-based, pixel-based, and camera-based approaches [30], [36].

- *Pixel-based Techniques:* Image pixels are the basic building blocks of a digital image. The pixel-based approach concentrates on the pixels of images. These approaches rely on various statistical abnormalities generated at the pixel level. Changes in image statistics determine the functioning of such algorithms. Thus, these methods identify counterfeit pictures by identifying statistical anomalies at a pixel level. The most common types of forgeries in this approach are splicing and copy-move. The pixel-

based strategies are the most practical of the mentioned techniques since they do not require prior knowledge about the transformations/tampering applied to the picture, nor do they require any expertise or information-gathering procedure for the images.

- *Format-based Techniques:* These approaches are mostly dependent on picture formats, with JPEG being the preferred format. The statistical correlation generated by the lossy compression method is employed in format-based methods to identify image counterfeiting.
- *Camera-based Techniques:* Various image forensic techniques successfully exploit the abnormalities produced by the camera's sensor or lens in camera-based procedures. Based on various modeling ideas, these approaches are used to estimate the camera artifacts.
- *Physically-based Techniques:* These approaches unambiguously identify and model irregularities in the three-dimensional interface among camera, physical objects, and light.
- *Geometric-based Techniques:* The variation in the image's primary point works as proof to assess the image's validity in the geometry-based approach.

1.5 Classification of Passive Techniques

Many researchers have been focusing on the field of image forensics since the introduction of synthetic images to identify different methods of image forgeries. Notably, among passive IFDTs, copy-move and splicing forgery are still in their early stages of development, which has grabbed the interest of many academics in previous years. As a result, this research work focuses on passive IFDTs such as copy-move and splicing forgery.

1.5.1 Copy-Move Forgery (CMF)

CMF is a simple, successful, and widely used form of image forgery. Secure areas of a picture are copied, moved, and inserted into a different section in a similar image using this approach. This type of forgery is used to obscure a specific part or replicate it in a picture. CMF can be accomplished in a variety of ways. The plain CMF defines the act of duplicating a portion of a photograph and placing it into another portion of the same picture without modifying it. In a multiple CMF, multiple areas or objects are duplicated and placed into different parts of a picture. Sometimes, an item is replicated from an image and put in two separate locations. In a different scenario, several things are duplicated and inserted into various positions inside the

same picture [9], [30], [37]. Several methodologies of Copy-Move Forgery Detection (CMFD) have been proposed, and they may be categorized into three types: keypoint-based, block-based, and hybrid-based techniques. Figure 1.10 shows an illustration of CMF.

(a) (b)

Figure 1.10: Illustration of CMF (a) authentic image with one pillar; (b) forged image where the left pillar is copied (yellow-circled area) and pasted on the right side of a building (red-circled area)

1.5.1.1 Block-Based Techniques

In block-based CMFD methods, the input picture is partitioned into small blocks. This split might be non-overlapping or overlapping. The relevant attributes are then retrieved from each block. Finally, the features are compared using several feature matching algorithms to find the copy-moved areas.

1.5.1.2 Keypoint-Based Techniques

Keypoints are distinguishing local characteristics in a picture, such as blobs, corners, and edges. Such key points are extracted from a photo using the keypoint-based approach. Then, these keypoint attributes are compared for similarity to identify duplicate areas.

1.5.1.3 Hybrid-Based Techniques

A hybrid-based CMFD methodology combines block-based and keypoint-based methodologies to achieve the advantages of both methods. Even though block-based approaches outperform in flat regions and are resistant to post-processing procedures such as JPEG compression in duplicated areas. However, they consume more time. Keypoint-based approaches, on the other hand, do well against rotation and scaling attacks. They also have minimal computational time but perform poorly in flat regions.

1.5.2 Image Splicing Forgery (ISF)

ISF is a methodology of merging two or more images. To some extent, splicing is comparable to CMF. The region copied in copy-move comes from the same picture; however, the area to be inserted in the spliced picture comes from a different picture. The splicing method involves replicating a specific section from one picture and placing it onto a different picture to create a fake picture. Image splicing involves compressing, resampling, and blurring areas to create a forged image. Consequently, spliced pictures are being exploited for nefarious reasons because image splicing is simple, and detecting forged images with human eyes is challenging [9], [30]. Figure 1.11 depicts the illustration of ISF, in which two original pictures are joined to make a counterfeit picture.

(i) Authentic images (ii) Spliced image

Figure 1.11: An illustration of image splicing forgery

1.6 Generalized Framework of Image Forgery Detection

The structure of image forgery detection is exemplified in Figure 1.12.

- *Choosing an image dataset:* In digital forensics, choosing a dataset is crucial. A comprehensive dataset must include all sorts of forged images/videos that the classifier may face in reality. Many datasets are open to the public. Primarily, numerous studies were carried out on their datasets; nevertheless, maximum researchers eventually switched to using public datasets or synthetic versions. This has made comparing the performance of various approaches at ease.
- *Pre-processing:* To increase classification performance, a pre-processing step is performed. Many pre-processing approaches may be used before each image/video is subjected to feature extraction procedures to enhance the classification accuracy rate. Color conversion, equalization of JPEG quality factor, image blurring, noise filtering, and other methods are often employed.

- *Feature selection and extraction:* It is the act of converting a huge quantity of data into useful info. The classifier uses these characteristics to distinguish between original and tampered image classes. When choosing characteristics from an input image, factors such as the goal and approach of feature selection, feature dimensionality, and the computational cost of retrieving features are taken into account.
- *Classifier selection:* A classifier is a model-training method that takes a feature vector as input. After learning from the training dataset, a classifier generates a model. Support Vector Machine (SVM), Decision Tree, Deep Learning, Random Forest, and Naive Bayes are examples of classifiers. Accuracy, effectiveness, resilience, simplicity, and training time are essential variables when selecting a classifier.
- *Performance evaluation:* The performance of the classifier is assessed using a variety of parameters, including recall, precision, accuracy, and so on. To examine the final classification tool, all of these characteristics are combined [9], [10], [38].

Figure 1.12: A generalized framework of image forgery detection

1.7 Description of Various Attacks in Image Forgery

To test the effectiveness of the suggested methodology in this research work, various geometrical (i.e., rotation and scaling), as well as post-processing attacks (i.e., JPEG compression and noise addition), are added to the image. Image scaling and rotation are geometric operations that are used to reduce, zoom, and rotate images, respectively. These operations are used to geometrically alter images using geometric coordinate transformations. Let $Y(W,V)$ represents the input image which is geometrically transformed to the output image $Y^*(W^*,V^*)$, which is specified as:

$$Y^*(W^*,V^*) = Y(TT^{-1}\{(W,V)\}) \tag{1.7.1}$$

where $TT^{-1}\{\cdot\}$ denotes the inverse transformation function. Therefore, these operations are considered affine transformation, which is defined as the mapping of image statistics from one vector space to another based on matrix multiplication/addition and translation [39], [40].

Moreover, post-processing operations like JPEG compression and noise addition are added to the picture to determine the efficacy of the presented technique.

- *Rotation:* The rotation attack conceals tampering evidence, which makes the detection process more challenging. In this situation, tampered portions are rotated with different rotation angles. Assume a rotation angle θ' is applied to the input picture $Y(W,V)$ and the image after rotation is $Y*(W*,V*)$. The affine matrix for image rotation operation is expressed as:

$$TT = \begin{bmatrix} \cos\theta' & \sin\theta' & 0 \\ -\sin\theta' & \cos\theta' & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.7.2}$$

$$W* = W\cos\theta' - V\sin\theta', V* = W\sin\theta' + V\cos\theta' \tag{1.7.3}$$

- *Scaling:* The scaling attack also adds in hiding tampering evidence. Because scaling generally results in some pixel loss, it makes detection more difficult. Consequently, the performance of the suggested technique is evaluated under scaling assault. In this, the copied portions are rescaled among different scaling factors. The image scaling operation can be represented in the form of affine transformation with an affine matrix (TT) is defined as:

$$TT = \begin{bmatrix} ss_{W*} & 0 & 0 \\ 0 & ss_{V*} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.7.4}$$

With equivalent coordinate equations as below:

$$W* = ss_{W*}W, V* = ss_{V*}V \tag{1.7.5}$$

- *JPEG Compression:* Image compression is quite frequent in everyday life. Most images on the Internet are compressed, and it is also a handy way to hide tampering traces. Thus, a comparison experiment using a JPEG compression assault is conducted in this research. The tampered images are compressed via various quality factors.
- *Noise Addition:* Adding noise to a tampered image is a common approach for concealing the evidence of image manipulations. Consequently, a suitable procedure must be resistant to noise assaults. Herein, the cloned regions are added with Gaussian noise by standard deviations.

1.8 Research Motivation and Gaps

With the evolution of technology, modern lives got intertwined with the culture of data consumption. Metadata has grown more accessible; consequently, software allowing users to tamper with photographs has become highly prevalent. This creates a slew of issues with the legitimacy of pictures, particularly in official processes. Examples include utilizing photos as court proof, accident verifications for insurance claims, or in a healthcare context where an illness or other health abnormality may be modified and used as an authorized diagnosis for personal advantages, such as suing an employer. In contrast, there has been an upsurge in modifying personal images for unethical benefits, such as duplicating photo IDs or several types of licenses, legal papers, and so on. The contemporary state of employed approaches suffers from low proficiency and/or a lack of specialization in the detection of forgery types. These ongoing challenges encourage us to go deeper into this field to propose a possible solution based on cutting-edge achievements, but with an emphasis on ISF and CMF.

In the prior work, current block-based procedures [41]–[46] do better in flat regions and are resilient to post-processing procedures alike JPEG compression in duplicated regions. Nonetheless, these consume more time and are ineffective under scaling and rotation attacks. In comparison, keypoint-based strategies [47]–[50] do well against scaling and rotation. Furthermore, they require less time but perform poorly in flat areas. Thus, it motivated me to carry out this research work to detect CMF by fusing both keypoint-based and block-based techniques. Moreover, many of the existing methods [42], [47], [48], [51]–[55] are inefficient in localizing and identifying multiple forgeries in digital pictures. As a result, this research is motivated to develop a computationally proficient and accurate algorithm for localizing and detecting single and multiple Copy-Move Forgeries (CMFs).

Earlier techniques [56]–[58] used Local Binary Pattern (LBP) with Discrete Wavelet Transform (DWT) in scenarios like object recognition, image retrieval, and facial expression recognition. Moreover, an earlier study detects ISF using either the combination of LBP and DWT [59], [60] or the fusion of DWT and Markov [61], [62], but the fusion of Markov, DWT, and LBP is not applied in any application. Consequently, in this research, Markov attributes from LBP and DWT areas are retrieved and merged to identify ISF efficiently.

Several procedures [42], [47], [48], [51]–[55], [63]–[68] have been developed to identify copy-move and ISF independently, but very few [69]–[73] are available to identify them together. Nevertheless, a more effective solution is essential to handle the daily problems of detecting

numerous forgeries at the same time. Thus, in the research work, a passive hybrid technique is proposed to detect CMF and ISF concurrently and proficiently. Furthermore, localization of tampered pictures is done to identify duplicated regions in both forgery types.

In recent years, there has been a growth in the study of deep learning architectures for diverse image processing applications. Many such CNN-based models that learn complicated contextual features are proposed, however, their accuracy is low. Even though numerous similar approaches are presented, they do not outperform the CMF detection scheme. They either provide lower parametric values or need a significant amount of computing time to attain better performance. Subsequently, this research work presents a robust technique using deep learning to effectively solve the issue of CMF detection.

Based on the previous studies, the following research gaps are explored:

- Many existing approaches are ineffective in identifying and localizing multiple CMFs in images. So, there is scope to identify and locate multiple CMFs in digital images.
- Several existing algorithms have been developed to detect CMF and ISF separately, but only a few can identify both simultaneously. Thus, there is a scope of hybrid techniques for simultaneously detecting copy-move and splicing forgeries.
- Existing block-based approaches are often robust to JPEG compression in duplicated portions. However, these consume more time and are ineffective in detecting geometrical processes such as scaling and rotation. Further, keypoint-based approaches work well when rotated and scaled. They consume less time but perform poorly in flat regions. As a result, there is scope for combining both strategies to reap the profits of block-based and keypoint-based procedures.
- In earlier research, CMF is recognized by merging AS with SIFT or fusing SIFT and AKAZE. However, the combination of AS, AKAZE, and SIFT has not yet been applied in CMFD. So, there is the scope of this combination to detect CMF.
- Multiple combinations of LBP and other techniques are employed in the previous studies; however, a more precise methodology is still necessary. As a result, there is scope to combine DFrCT with LBP. It takes advantage of the flexibility of a fractional parameter in DFrCT. Further, LBP is used to detect picture forgeries by effectively emphasizing tampering artifacts.

- Motivated by the strong proficiency of the Markov TPM in describing pixel relationships, there is the scope of extracting and merging Markov features from DWT and LBP domains.
- Since some current approaches are ineffective against various attacks, there is scope to improve the robustness of the forgery detection structure under several attacks like JPEG compression, rotation, noise addition, and scaling.

1.9 Research Objectives

Based on initial studies, a comprehensive literature survey, and research gaps, the following objectives are proposed:

1. To study and analyse existing passive image forgery detection techniques.
2. To propose an improved copy-move passive image forgery detection approach.
3. To propose a hybrid passive image forgery detection technique and compare it with existing techniques.
4. To evaluate the robustness of proposed technique against geometrical attacks such as rotation, scaling, JPEG compression.

1.10 Contributions of Research Work

The significant contributions of the presented research work are as follows:

- Many existing approaches are ineffective in identifying and localizing multiple CMFs in digital pictures. The proposed technique in this research study can identify and locate single and multiple CMFs in digital images by integrating Accelerated KAZE (AKAZE), Adaptive Over-Segmentation (AS), and Scale-Invariant Feature Transform (SIFT).
- The experimental results demonstrate that the presented methodology attained an improved F_2 score of 99.81%, 99.80%, 99.35%, and 99.82% on respective benchmark datasets, i.e., MICC-F220, IMD, COVERAGE, and GRIP, as compared to existing methodologies.
- Motivated by the high capacity of Markov TPM in measuring correlation among pixels, Markov attributes from both LBP and DWT areas are retrieved and merged for identifying ISF in the research work. Also, there are abrupt changes during the development of ISF, which are emphasized by employing a Standard Deviation (STD) filter in the suggested methodology.

- The experimental findings using benchmark datasets, namely DSO-1, CASIA v1.0, Columbia, CASIA v2.0, DVMM, and IFS-TC, demonstrate that the proposed strategy outperforms current approaches on respective datasets with increased accuracy of 92.50%, 99.69%, 98.61%, 99.76%, 97.80%, and 96.90%.
- The research work is aimed at detecting CMF and ISF concurrently and proficiently by combining Discrete Fractional Cosine Transform (DFrCT) and LBP. LBP is employed to recognize tampering in photos by proficiently emphasizing tampering artifacts and DFrCT utilizes the flexibility of fractional parameter.
- Furthermore, in both forgery types i.e. CMF and ISF, forged image localization is done to locate tampered areas.
- This research work presents a robust approach for detecting CMF that makes use of the Contrast Limited Adaptive Histogram Equalization (CLAHE) and Convolutional Neural Network (CNN) model. This combination generates an effectual framework by outperforming existing techniques with improved performance metrics.
- Since some current approaches are ineffective against various attacks, the robustness of the suggested schemes are confirmed by outperforming existing strategies against different attacks such as scaling, JPEG compression, rotation, and noise addition.
- Furthermore, statistical analysis tests such as Analysis of Variance (ANOVA) and cross-dataset performance are used to validate the efficiency of the presented strategies.

1.10 Organization of Thesis

The following summarizes the flow of the work done in this thesis:

Chapter 1 provides an overview and description of the thesis. It contains thorough information about the history of image forgery, digital image forgery, digital IFDTs, and their classification, motivation, contributions of research work, and thesis organization.

Chapter 2 covers the existing literature on CMF and ISF. Various performance measures for assessment are also addressed, depending on which image forgery techniques' performance is evaluated. Furthermore, detailed information about image datasets required for evaluation is presented. The objectives, research gaps, and research methodology are also stated in this chapter.

Chapter 3 focuses on efficiently detecting single and multiple CMFs by merging block-based procedures such as AS and keypoint-based procedures such as AKAZE and SIFT. The

combination of both techniques makes the suggested methodology more resilient under several attacks and less computationally expensive. The experimental findings reveal that the suggested procedure is resistant to several attacks such as rotation, JPEG compression, scaling, and noise addition and outperforms other existing procedures. Furthermore, to prove the efficacy of the suggested methodology, the statistical analysis test and performance across datasets are tested.

Chapter 4 involves the detection of ISF. In this, Markov attributes from LBP and DWT areas are retrieved and merged for finding ISF. The experimental findings suggest that combining Markov attributes from the LBP and DWT domains improves different performance measures in contrast to conventional techniques. The experiment results indicate that the suggested approach is effective against various attacks, and it surpasses other current strategies. Furthermore, statistical analysis tests and cross-dataset performance are assessed.

Chapter 5 presents a passive hybrid approach based on DFrCT and LBP to identify CMF and ISF at the same time. The fractional parameter of DFrCT is employed as an additional parameter to improve accuracy, while LBP is used to efficiently reveal tampering artifacts. A comparison study with existing approaches in terms of several performance metrics has been conducted to validate the efficacy of the presented method. Furthermore, the efficacy of the suggested work against various attacks is verified, and enhanced results are obtained when compared to previous methodologies. Also, statistical analysis tests and cross-dataset performance are evaluated.

Chapter 6 introduced a deep learning CMFD framework, which classifies images as authentic or forged using CLAHE and CNN. In terms of various performance metrics, the experimental study demonstrates the effectiveness of the suggested approach among other approaches. Also, the resilience of the suggested approach against various attacks is proved. Further, the proposed system is evaluated on cross-datasets, and statistical analysis tests are assessed.

Chapter 7 provides the conclusion as well as the future scope of the research work conducted in this thesis.

CHAPTER 2

LITERATURE SURVEY

This chapter reviews the existing literature on CMF and ISF. Various performance metrics for assessment are also addressed, depending on which the performance of image forgery techniques is assessed. Furthermore, specific information on the image datasets necessary for evaluation purposes is discussed. This chapter also discusses the objectives, research gaps, and research methodology.

2.1 Review of CMF

The most common and well-known form of image forgery detection methodology is CMF. To improve/eliminate characteristics, a specific portion of the picture is used. This scheme includes copying a segment of a picture and placing it into another area within the same picture to obscure particular characteristics in an original picture or replicate data that is absent in the original. Since CMF is performed in the same picture, the transformed region's features will be identical to those of other portions. The CMFD procedures are distributed into keypoint-based, block-based, and hybrid-based procedures. Block-based procedures split images into overlying circular or rectangular blocks, which differ from procedure to procedure. In contrast, key point-based procedures extract characteristics from the entire image. Hybrid-based procedures are a combination of both keypoint-based and block-based procedures.

2.1.1 Block-Based Techniques

These procedures extract attributes from a single block and then compare them to attributes from various blocks. If the attributes of two blocks are the same, a block is inspected to confirm CMF. Traditional block-based procedures include Fractional Quaternion Zernike Moments (FQZM), Cellular Automata (CA), Zernike moments, DWT, Discrete Radial Harmonic Fourier Moments (DRHFM), Discrete Cosine Transform (DCT), Fractional Fourier Transform (FRFT) and many more.

The DCT-based methodology for CMFD was initially suggested by Fridrich *et al.* [74]. The picture is split into fixed-size overlaying picture sections at bitmap graphics, and DCT is employed for each section. The quantization feature vector is produced by examining the quantized DCT coefficient matrix in a zigzag pattern. The feature matrix is lexicographically

structured, and the match is determined via Euclidean distance. This procedure, however, has a high computational cost. Popescu *et al.* [75] reduced the dimensionality of block characteristics by Principal Component Analysis (PCA) and presented a method for detecting tampered regions in forged images. This format is unaffected by slight image variations caused by additive noise or lossy compression. However, it fails to execute image rotation, resize, and flipping operations. To identify unique artifacts, Li *et al.* [76] developed a technique employing DWT and Singular Value Decomposition (SVD). The DWT reconstructs and decomposes signals utilizing functions i.e., fundamental wavelet $\xi(t)$ and scaling $\gamma(t)$. The scaling function approximates the original signals, while the fundamental function, which is described below, detects detailed variations [62], [77].

$$\gamma(t) = \sum_n a(n)\sqrt{2}\xi(2t-n) \tag{2.1.1}$$

$$\xi(t) = \sum_n b(n)\sqrt{2}\xi(2t-n) \tag{2.1.2}$$

The $\gamma(t)$ is also employed to measure the fundamental wavelet function. $a(n)$ and $b(n)$ are filter's coefficients, their relationship is specified in eq. (2.1.3).

$$a(n) = (-1)^n b(m-n-1) \tag{2.1.3}$$

The length of the filter is denoted by m, while the number of levels is specified by n. The wavelet transform's decomposition is seen below:

$$eAl(l) = \sum_n b(n-2l)R(n) \tag{2.1.4}$$

$$eDl(l) = \sum_n a(n-2l)R(n) \tag{2.1.5}$$

The original signal is indicated by $R(n)$. The low-frequency information and high-frequency information of $R(n)$ are retained by approximation coefficient $(eAl(l))$ and detailed coefficient $(eDl(l))$, respectively.

To enhance the localization of the copied area, Bravo-Solorio *et al.* [41] generated a 1D descriptor that is uniform to reflection and rotation by aggregating a logging map across the log radii axis. However, it does not perform well when subjected to geometrical changes like JPEG compression, additive noise, and Gaussian blurring. Muhammad *et al.* [78] propose a blind CMFD procedure depending on undecimated Dyadic Wavelet Transform (DyWT). This technique is tested in three scenarios: constant size pictures and counterfeit without rotation, varied size pictures and counterfeit with or without rotation, and varying quality factors of

JPEG pictures. The block attributes are extracted by Li *et al.* [79] using Polar Cosine Transform (PCT). Bravo-Solorio *et al.* [42] developed a method for detecting duplicated areas impacted by scaling, rotation, and reflection using a one-dimensional descriptor with a low False Positive Rate (FPR). However, these techniques have a higher computational complexity.

Zhao *et al.* [43] offered a procedure based on DCT and SVD to identify CMF. Experiment findings show that this approach can identify multiple CMFs and correctly spot duplicated portions even when a picture has been affected by Gaussian blurring, JPEG compression, and Additive White Gaussian Noise (AWGN) or their combined processes. However, it fails against rotation and scaling attacks. Further, PatchMatch, a fast approximate nearest-neighbor search methodology designed specifically for the calculation of dense areas over images, is used by Cozzolino *et al.* [80] and, Zhong *et al.* [81] introduced a unique approach for supporting CMFD using Discrete Analytical Fourier-Mellin Transform (DAFMT). Discrete Polar Complex Exponential Transform (DPCET) is employed by Emam *et al.* [82] to retrieve block attributes and LSH to detect equivalent blocks. This method, however, is ineffective and slow.

Mahmood *et al.* [83] use both the Stationary Wavelet Transform (SWT) and DCT to expose CMF in digital images. The suggested approach initially employs SWT on the picture to obtain an approximation subband. Following that, the approximation subband is split into fixed-length overlying blocks. The experimental findings indicate that it is resistant to different geometrical attacks. Ouyang *et al.* [46] used the pyramid algorithm and Zernike moment to identify CMF, which delivers improved outcomes for assaults like scaling and rotation while demanding more processing time. Meena *et al.* [84] discovered duplicated regions using the tetrolet transform under several attacks; however, performance factors still need to be improved. Priyanka *et al.* [51] identified CMF employing DCT and SVD. This technique yields high recall results, yet, it offers a poorer accuracy owing to incorrect clustering produced by K-means clustering.

Al-Qershi *et al.* [44] suggested a better matching methodology based on the k-means clustering procedure. Furthermore, the matching approach is used in this algorithm combined Locality-Sensitive Hashing (LSH) with Zernike moments. Although the suggested technique improves detection accuracy and decreases processing time, it has a lower precision and recall value. Kasban *et al.* [85] extracted Hilbert–Huang Transform (HHT) characteristics from the C_r component after first converting the RGB image to YCbCr space. To determine forgery detection accuracy, the conclusions are validated using Structural-Similarity (SSIM). In

addition, the suggested method has been evaluated against a variety of post-processing threats. The objective of this study is to improve detection speed even in the existence of assaults by adjusting the value of SSIM. Ahmed *et al.* [86] presented a CMFD approach utilizing SVD and Kolmogorov Smirnov (KS) test. An image is first partitioned into blocks. Then, using steerable pyramid and SVD transformations, image characteristics are retrieved from each block. Finally, retrieved characteristics are lexicographically sorted, and the KS test is used to match them. It is resistant to color alteration, brightness change, picture blurring, and contrast modification.

Babu *et al.* [87] offer a two-step method for detecting forgeries. Primarily, the mistrusted picture is split into multiple directions using Steerable Pyramid Transform (SPT), and Grey Level Co-Occurrence Matrix (GLCM) characteristics are retrieved from each direction. Then, an Optimized SVM is employed to classify the images. If a suspected picture is classified as fake, then it is transformed into overlying blocks, and from each block, GLCM characteristics are retrieved. The suggested method needs to be modified in the future to evaluate if an image is legitimate or produced by a machine. Gani *et al.* [45] presented a block-based CMFD method that performs well with post-processing modifications. It extracts features from image blocks using the DCT and CA that are then matched using the patch match technique. In addition, a modest and quick CA-based technique is provided to reliably recover the cloned areas matching the matched characteristics. However, it has a long feature extraction time. The main disadvantage of block-based approaches has been identified as their computational cost, and more powerful feature extraction algorithms are required.

2.1.2 Keypoint-Based Techniques

This technique is dependent on Polar Complex Exponential Transform (PCET), Speeded-Up Robust Features (SURF), SIFT, Affine Scale-Invariant Feature Transform (ASIFT), Oriented FAST and Rotated BRIEF (ORB), KAZE, Binary Robust Invariant Scalable Keypoints (BRISK), AKAZE, etc.

Pan *et al.* [47] pioneered the use of key point matching for CMFD. Although this technique performs well against geometric alterations when the metrics are calculated by Random Sample Consensus (RANSAC) procedure, a more effective scheme is necessary. Based on SIFT characteristics, a novel methodology for image forensics analysis has been presented by Amerini *et al.* [48]. Further, Prakash *et al.* [88] used a combination of SIFT and AKAZE to

identify copied areas against geometrical assaults; however, a more precise technique is needed. The SIFT descriptor consists mostly of four phases, which are detailed below:

a) Detection of scale-space extrema: Primarily, scale-space extrema is identified using Difference of Gaussian (DoG). The convolution of Gaussian function $Gf(W,V,\sigma)$ with the input picture $Y(W,V)$ is image's scale-space $L(W,V,\sigma)$, as indicated below [89]:

$$L(W,V,\sigma) = Gf(W,V,\sigma) * Y(W,V) \tag{2.1.6}$$

$$\text{with } Gf(W,V,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(W^2+V^2)/2\sigma^2} \tag{2.1.7}$$

where W and V are input image's coordinates, σ is Gaussian function's standard deviation. DoG approach finds the keypoints in the picture at scale $k\sigma$, as indicated in the following equation:

$$D(W,V,\sigma) = (Gf(W,V,k\sigma) - Gf(W,V,\sigma)) * Y(W,V) = L(W,V,k\sigma) - L(W,V,\sigma) \tag{2.1.8}$$

The local minima and maxima of $D(W,V,\sigma)$ are determined by comparing each pixel to its eight neighbors at an equivalent scale or its nine neighbors up and down the scale. A pixel is defined as a candidate keypoint if its value is the lowest or highest of all the pixels compared.

b) Keypoint localization: The scale-space extrema detection creates a huge number of keypoint candidates, some of which are unstable. So, after identifying the keypoints by pixel and neighbor comparison, these are adjusted to provide more exact outcomes. Interpolation is conducted using the expansion of second-order Taylor series $D(w)$ to get more precise keypoint locations. Using the candidate keypoint as the basis, the Taylor expansion of DoG scale-space function $D(W,V,\sigma)$ is:

$$D(w) = D + \frac{\partial D^T}{\partial w} w + \frac{1}{2} w^T \frac{\partial^2 D^T}{\partial w^2} w \tag{2.1.9}$$

At a candidate keypoint, D and its derivatives are computed. $w = (W,V,\sigma)$ is the distance from this place. The extremum's location $\hat{w}$, is found by calculating the derivative of $D(W,V,\sigma)$ function for w and keeping it to zero, as illustrated:

$$\hat{w} = -\frac{\partial^2 D^{-1}}{\partial w^2} \frac{\partial D}{\partial w} \tag{2.1.10}$$

If $\hat{w}$ in either dimension is more than 0.5, the extremum is nearer to the other candidate's key point. In this instance, the candidate keypoint is modified and interpolation is conducted in its vicinity. To acquire an interpolated estimation of the extremum's position, the final offset $\hat{w}$ is applied to the candidate keypoint's position. The value of $D(w)$ is assessed at the offset $\hat{w}$ to eliminate the key points with poor contrast. If the intensity at any extremum falls below a particular level, the key points are removed. Thus, keypoint candidates are localized and selected in keypoint localization by removing low contrast keypoints [48], [88], [89].

c) Orientation assignment: It is performed using local picture gradient. This stage assigns an orientation to each key point depending on the local image attributes. In the first stage, the Gaussian picture L is computed. Then, orientation θ and magnitude gradient mg are computed as:

$$mg(W,V) = \sqrt{(L(W+1,V) - L(W-1,V))^2 + (L(W,V+1) - L(W,V-1))^2} \tag{2.1.11}$$

$$\theta(W,V) = \tan^{-1}\left(\frac{L(W,V+1) - L(W,V-1)}{L(W+1,V) - L(W-1,V)}\right) \tag{2.1.12}$$

Using the gradient locations of the data points, an orientation histogram is constructed, and then the orientation is generated using the orientation histogram's peaks.

d) Keypoint descriptor: Finally, using the previously mentioned local gradient data, keypoint descriptors are generated. A 16×16 neighborhood is formed around the keypoint and divided into 16 sub-blocks of 4×4 size. An 8-bin orientation histogram is created for each sub-block.

Zhao *et al.* [90] propose a passive forensic approach for identifying region duplicate picture counterfeiting depending on LBP and Harris feature points. Experimental results indicate that the procedure can identify CMF even under geometrical assaults and that it is extremely resistant to forging with the smooth area of small structures. Despite its promising detection performance, the suggested approach fails to identify CMF with scaling since LBP and Harris corners are susceptible to picture scaling, and thus, they are not being calculated on multiscale images.

For identifying identical areas in an image, Jaberi *et al.* [91] employed Mirror Reflection Invariant Feature Transform (MIFT). The outcomes show that the technique can identify CMF with greater accuracy, specifically when the size of the identical portion is small. Yu *et al.* [92] present a feature-based system that combines Hue Histogram (HH) and Multi-Support Region

Order-Based Gradient Histogram (MROGH) descriptors to improve feature coverage and matching efficiency. This approach can withstand modest degrees of scaling, additive noise, and their combination; nevertheless, performance suffers fast when these assaults are significant due to the Harris Corner Detector's instability under these situations.

Yang *et al.* [52] suggested a CMFD system based on hybrid characteristics. To retrieve additional feature points, KAZE, a strong interest point detector, is employed and coupled with SIFT. Experiments demonstrate that this approach can efficiently detect duplications, even when the duplicated parts have experienced multiple geometrical modifications. An improved keypoint-based CMFD for tiny smooth areas is introduced by Wang *et al.* [93]. The suggested CMFD technique has a greater computational cost, which means it cannot be employed successfully in real-time applications.

Alberry *et al.* [94] offered a unique strategy to utilize Fuzzy C-Means (FCM) clustering throughout the matching procedure. Before the actual matching phase, the method employs FCM to gather the observed SIFT key points. Consequently, the computational cost is reduced, resulting in a faster matching process. Wang *et al.* [95] discovered as well as localized CMFs depending on color resemblance framework and Quaternion Polar Complex Exponential Transform (QPCET). SURF detector and color invariance model are used to retrieve stable color image interest spots. Further, pairs of linked Delaunay triangles are constructed, and appropriate local graphic characteristics and QPCET coefficients are calculated. The SURF and BRISK descriptors are merged by Bilal *et al.* [96] for CMFD.

Dhivya *et al.* [97] employ SURF and SVM. The SURF feature extraction algorithm picks the most prominent spots in the input images. The SVM classifies the incoming images using object recognition to retrieve the matching feature points. Using a lower contrast threshold and scaling picture, Li *et al.* [49] presented CMFD that creates an acceptable number of key points in flat areas, but it has a long computing time. Yang *et al.* [98] offered a CMFD approach depending on adaptive keypoint removal and matching. Initially, adaptable even distribution criteria are used to extract image key points. Next, to express the local picture characteristic of picture keypoints, the BRISK descriptor is designed. Then, using the embedded random ferns method, local BRISK attributes are used to correlate picture key points. Lastly, the RANSAC is used to remove erroneously matched keypoint pairs, and tampering regions are localized.

Liu *et al.* [99] integrated Local Intensity Order Pattern (LIOP) and SIFT keypoints to exploit the supremacy of various key points, and the PatchMatch method is used to localize the

tampered regions rather than clustering or linear transformations. Wang *et al.* [100] suggested an efficient and high image CMFD framework that incorporates obtaining adaptive image keypoints, presenting a fast robust invariant feature, and sorting out incorrect pairs. In begin, smoothed key points are recovered dynamically from the counterfeit image using the rapid approximated Laplacian of Gaussian (LoG) filter. The image keypoints are then characterized using a quick robust invariant attribute and matched utilizing the Rg2NN method. Lastly, erroneously matched pairings are eliminated using segmentation-based candidate clustering, and replicated areas are located by optimal mean-residual normalized production correlation.

Uma *et al.* [101] propose the CMFD method that takes into account a tiny proportion of strongest key points selected from both DoG-based and FAST-corner key points. Also, it estimates SIFT descriptors, employs DWT for dimension reduction, and implements Football Game-Based Optimization (FGBO) for cluster space classification. Niu *et al.* [102] presented a quick and efficient CMFD. Primarily, a Bessel Fourier Moments (BFM) based descriptor is intended to provide a quick, robust, and exclusionary presentation of dense key points. The total matching complexity is then lowered by constructing a hierarchical matching method based on BFM invariant magnitude and phase data. Yang *et al.* [50] present Two-stage Filtering (TSF), an effective and economical CMFD method. This technique is used to filter out key point pairs that do not match. Following that, the Delaunay triangulation method is used for image matting to identify forgeries areas. Although it is superior at identifying forgeries when compared to post-processing and large-scale scaling procedures, it has a long average detection time. CMF has been recognized by Chen *et al.* [103] by clustering SIFT keypoints and monitoring comparable neighbors to find counterfeit regions. This approach can identify counterfeit areas in single and geometric altered forgery; however, it cannot detect multiple forgeries.

The primary benefits of keypoint-based CMFD techniques are their remarkable performance in computing cost, memory requirements, and uniform scaling and rotation procedures, although they indeed struggle to decrease false matches in plane areas. Another issue with key-point-based techniques is that fewer key points are extracted in smooth/flat and homogenous regions.

2.1.3 Hybrid-Based Techniques

Each keypoint-based technique has its own set of advantages and disadvantages. Duplication detection can sometimes be improved by using a fusion of various keypoint-based techniques.

Some researchers have also suggested combining keypoint-based approaches with block-based methodologies to address the limitations of keypoint-based methodologies in flat regions. As a result, several features of block-based methods are merged with classic keypoint-based approaches to generate fusion-based techniques.

Pun *et al.* [104] used AS and feature point matching to conduct studies on the Image Manipulation Dataset (IMD). For successful CMFD a block-based method utilizing Zernike moments and a keypoint-based methodology employing SIFT, together with filtering and post-processing, are implemented in smooth and non-smooth areas by Zheng *et al.* [105]. Sun *et al.* [106] propose a CMFD approach capable of addressing the problem of computation cost. Using SIFT and Zernike moments, features are retrieved and matched in the textural and smooth regions, respectively. Since it is resistant to various attacks, however, the impact of the process is not very excellent while dealing with highly smooth blocks, like walls and sky. So, there is a need to enhance location accuracy and detecting speed.

By hybridizing DCT and SURF, Ojeniyi *et al.* [107] discovered the CMF. This technique can address some of the challenges concerned with CMF; however, there is still a need for enhancement in detecting other types of image formats in jpeg format. Depending on Simple Linear Iterative Clustering (SLICO) segmentation and Helmert transformation, Huang *et al.* [108] suggested a keypoint-based methodology for CMF images. The Helmert transformation is employed to compute geometric features among matched pairs and deal with merging clusters. A SLICO algorithm, on the other hand, is utilized to locate the tampering locations more accurately.

Elhaminia *et al.* [109] employ over-segmentation as a preprocessing step to generate superpixels, which are subsequently treated as neurons in the Markov network. The SLICO technique is used to split pictures into multiple superpixels, and then PCT and SURF characteristics are utilized for matching features and identifying comparable portions of the picture. Further, SLICO superpixel segmentation and K-means clustering are used by Liu *et al.* [110] to split the image into complicated and smooth regions. To identify tampering in complicated regions, SIFT characteristics are utilized. To identify alterations in even zones, segment mask characteristics and RGB color characteristics are offered. Nonetheless, a more precise approach is required.

Niyishaka *et al.* [111] offer an efficient approach for detecting CMF using image blobs and the BRISK feature. It consists of various stages: detecting image blobs and BRISK characteristics;

recognizing BRISK key points positioned in the same blob; and finally, matching BRISK key points positioned in remote blobs to identify equal key points for copy-move areas. This approach has a high computational cost. Meena *et al.* [112] suggested a hybrid approach that combined FMT with SIFT. The key points are retrieved from the picture's texture component with SIFT descriptor, and FMT is employed for the image's flat portion. The retrieved characteristics are compared to find the image's replicated areas. The simulation findings show that this approach outperforms in the case of geometric alterations and post-processing procedures in an acceptable amount of time.

Agarwal *et al.* [113] offer a robust CMFD system that combines block-based and keypoint-based CMFD approaches. Using a modified FCM clustering system relying on superpixels, a picture is separated into non-overlaying blocks. FCM incorporates surrounding and comparable superpixel influences, and Emperor Penguin Optimization (EPO) is utilized to optimize the influential degree, which improves segmentation performance. To extract the characteristics from each block, the Gabor filter is used. Although it achieves outstanding forgery detection results even in the most challenging conditions, there is still a need for an enhanced method using deep learning.

Tinnathi *et al.* [114] developed a unique CMFD procedure that uses adaptive segmentation and a hybrid feature extraction system. The tampered picture is adaptively segmented into non-overlapping parts, which improves the proficiency of the CMFD method and diminishes computing complexity. When the image suffered specific geometrical changes, using Hybrid Wavelet Hadamard Transform (HWHT) to retrieve the segment characteristics enhanced the efficacy and gave a robust result. Tahaoglu *et al.* [54] used CLAHE on three independent channels from two color spaces (RGB and L*a*b*) to extract SIFT keypoints. This approach detected counterfeit areas during several attacks, yet, a more accurate methodology is necessary.

In current history, there has been a growth in the research of deep learning architectures for various applications such as remote sensing, object recognition, medical imaging, face aging, face editing, person re-identification, facial expression, image encryption, and tracking active fire locations [115]–[120]. Also, deep learning has grown ambitiously for image forensics with the advent of methods in artificial intelligence. Training samples are utilized to retrieve the features in different deep learning setups, such as CNN and deep residual architecture. So, several recent deep learning procedures have been proposed to address image forgeries.

Al_Azrak *et al.* [121] developed a scheme for identifying CMF using block processing and feature extraction from block transformations. In addition, CNN is used to detect forgeries. Goel *et al.* [122] employed a dual branch CNN to differentiate between authentic and manipulated images. This approach retrieved multi-scale attributes to enhance outcomes; however, it is vulnerable to many geometrical assaults.

Jaiswal *et al.* [123] created a CNN model that uses multi-scale input and various levels of convolutional layers. These layers are split into two parts: encoder and decoder. Likewise, extracted feature maps are merged and upsampled in decoder blocks. In the encoder block, retrieved feature maps from several phases of convolutional layers are merged and downsampled. The resulting feature map is utilized to categorize the images using a sigmoid activation function. Zhong *et al.* [124] employed Dense-InceptionNet architecture for CMFD. Agarwal *et al.* [125] introduced a technique that utilized SLICO for image segmentation, VGGNet extract features, and adaptive patch matching to match key points.

Elaskily *et al.* [126] suggested a unique CMFD methodology based on segmenting target images into various items and examining similarities between these items. These strategies are successful against various assaults, but there is still room for improvement in their performance characteristics. Rodriguez-Ortega *et al.* [127] suggested two deep learning approaches: a framework based on a custom design and a framework based on transfer learning. In this case, the influence of network complexity is assessed regarding various performance parameters. The VGG-16 framework by transfer learning achieves parameters about 10% greater than the framework by custom architecture, yet, it involves almost double inference time.

2.2 Review of ISF

Another common method for creating the forgery is image splicing. It comprises exchanging segments of one or more areas of an image with segments of a different image. Splicing is a system of picture modification in which parts of the equivalent or distinct pictures are combined to form a counterfeit image without any additional processing, such as smoothing and blurring of borders and edges between fragments. Splicing can cause differences in a variety of ways, such as an abnormal sharpness at the boundaries, which can be used to detect manipulation. When two separate pictures captured at different times and locations are combined, the original and counterfeit pictures become unbalanced; this discrepancy is used as a criterion for identifying ISF.

Ng *et al.* [128] presented an ISF detection technique that uses bicoherence magnitude and phase characteristics. Zhang *et al.* [129] developed a forgery detection method based on LBP using Multi-Size Block DCT (MBDCT) coefficients. There is an increased desire for a more precise technique to address the daily forgery issues. Shi *et al.* [130] regarded an image's adjacent changes in BDCT parameters as a 1-D signal. The relationships among nearby nodes in a specific orientation (vertical or horizontal) are represented by a Markov process and Transition Probability Matrix (TPM) that is employed as an exclusionary feature vector for the SVM classifier. SVM classifier is applied by Li *et al.* [131] on HHT-based and moment features. Dong *et al.* [132] identified ISF by discovering statistical characteristics from picture run-length and edge data. Although these approaches identified ISF, their accuracy rate is low. As a result, there is a need for a more effective technique that can identify ISF with greater accuracy.

He *et al.* [67] combined the Markov characteristics in DWT and DCT domains. Even though the article established the legality of the Markov feature, the overall accuracy still has to be improved. Carvalho *et al.* [133] utilized an SVM meta-fusion classifier based on color differences in image illumination. The SPT and LBP are employed to identify image splicing by Muhammad *et al.* [68]. The colored picture is first converted to YCbCr, then SPT transform is applied to C_b and C_r channels. LBP histograms are then used to characterize the texture in each SPT subband, and SVM classifies images as fake or original using the feature vector. Agarwal *et al.* [134] used rotation invariant co-occurrence among LBP operators to retrieve internal statistical characteristics.

Zhang *et al.* [63] suggest a Markov model-based methodology to identify ISF. This study uses the Markov model in the block DCT and Contourlet transform domain. Initially, by considering the various frequency bands of each block DCT coefficient, the Markov characteristics of inter-block among block DCT coefficients are enhanced. Furthermore, new attributes are retrieved in a Contourlet transform domain to define the dependence of locations among Contourlet subband parameters. For grayscale images, SVM is used to identify genuine and spliced pictures, whereas an ensemble classifier is used for the color pictures. Agarwal *et al.* [135] highlighted the image's details with the Undecimated Wavelet Transform (UWT) before extracting the features with the Markov process. Zhao *et al.* [64] propose a 2-D Markov scheme for ISF detection. The model is used in block DCT and DMWT domains, and cross-domain attributes are employed as features for classification. The approach is complicated but more resistant to JPEG compression and median filtering.

Chen *et al.* [136] consider both RGB and Depth (RGB-D) information while improving quaternion representation for effectively expressing RGB-D images. Experiment findings on datasets illustrate that the proposed ISF detection system attains good accuracy with the proper feature dimensionality. El-Alfy *et al.* [65] provide a reliable technique for substantiating grayscale and color pictures. The global and local objects caused by image modification are captured by integrating Markov features in DCT and LBP areas. The combination of dimension reduction utilizing PCA and an optimized SVM classifier has proven effective. For ISD, a Markov method in the Quaternion DCT (QDCT) domain is presented by Li *et al.* [66]. Initially, color information is recovered from blocked pictures to create a quaternion in its entirety, and QDCT coefficients of quaternion blocked pictures are acquired. Second, the extended Markov characteristics derived from transition possibility arrays in the QDCT domain preserve the relationship between block QDCT coefficients. Subsequently, SVM is used to categorize the Markov feature vector.

An effective SVD-based feature extraction approach for ISF detection is presented by Moghaddasi *et al.* [137]. The method employs several methodologies, including a roughness measure procedure to estimate the set of acquired singular values proficiently, a kernel PCA system operated as a feature reduction phase to enhance classification, and an SVM classifier to differentiate between original and spliced pictures. Alahmadi *et al.* [138] used LBP and DCT to identify ISF. The procedure translates LBP blocks in the DCT area, after which the STD is calculated and given to the SVM classifier. Jalab *et al.* [139] identified ISF using Approximated Machado Fractional Entropy (AMFE). Although it produces superior outcomes, it is affected by post-processing procedures like JPEG compression. Zhang *et al.* [61] and Sheng *et al*. [140] discovered Markov attributes in the block DWT region and Discrete Octonion Cosine Transform (DOCT) region, correspondingly employing an SVM classifier.

Niyishaka *et al.* [141] present a simple and highly efficient ISF detection method that considers a trade-off between requirements and user cost. The phases in the technique are as follows: Initially, chrominance and luminance are extracted from the source images; furthermore, illumination is approximated from luminance by using the Illumination-Reflectance model; and lastly, LBP normalized histogram is calculated for illumination and chrominance as well as used as the feature vector for classifying the images using various machine learning algorithms. Although the suggested approach is computationally cheap and reliable for ISF detection, it cannot locate spliced images. Hussain *et al.* [142] investigated the effectiveness of multi-scale Weber Law Descriptor (WLD) and multi-scale LBP in detecting ISF. Alahmadi *et*

al. [143] employed 2D-DCT to identify constrained discriminant information in the LBP region.

Korus *et al.* [144] investigated a non-uniformity analysis of a photo response to detect small forgeries. Wei *et al.* [145] used a splicing detection approach that feeds the CNN a sequence of non-overlapping image patches. However, when all pixels in an image patch come from tampered locations, the image patch is anticipated to appear un-tampered. Lyu *et al.* [146] presented a technique for estimating blind image noise and its use in identifying region splicing. Xiao *et al.* [147] used a Coarse-to-Refined Convolutional Neural Network (C2RNet) with diluted adaptive clustering to identify picture splicing forgery. Huh *et al.* [148] offered a detection approach that used an image's EXIF meta-data as a supervisory signal to assess if a picture is self-consistent; however, the detection findings are unsuccessful if the information in the image's EXIF meta-data is insufficient.

2.3 Review of Hybrid Techniques for Copy-move and Splicing Forgery

In hybrid procedures, various forensic techniques are integrated to increase the reliability and efficacy of detection procedures. These approaches in most recent research target CMF and ISF. Liu *et al.* [69] focused on discovering the two most common kinds of image alteration, copy-move and splicing forgeries. An integrated methodology uses a single authentication procedure to locate forged areas. JPEG block synthetic networks and local noise inconsistencies are utilized to produce attributes in this approach that are then merged with the picture quality score as a factor. Prakash *et al.* [70] used Block DCT and an improved threshold technique to extract features. Jaiprakash *et al.* [71] presented a low-dimensional feature-based approach for detecting CMF and ISF. Image characteristics and pixel correlations are used to extract features from the DCT and DWT domains.

Dua *et al.* [72] offer a complete technique for evaluating JPEG compressed tampered pictures. With remarkable accuracy, forged pictures are detected from a standard collection of pictures using a double stochastic model with quantized DCT coefficients. Further, using Bag-of-Features (BOF) and Hamming Embedding (HE)-based picture retrieval, Pham *et al.* [73] present a robust technique for clustering relevant pictures. The suggested model finds a cluster center that is the sole legitimate picture in a cluster from picture clusters by leveraging structural correlation among images.

2.4 Performance Metrics Used for Evaluation

The performance evaluation metrics such as Accuracy (DA), precision (P), Mathews Correlation Coefficient (MCC), recall (R), F_1 score (F_1), markedness, Receiver Operating Characteristics (ROC), F_2 score (F_2), and informedness are used in this research work to determine the effectiveness of the detection approaches. Also, True Positive (T_{PO}), False Positive (F_{PO}), True Negative (T_{NE}), and False Negative (F_{NE}) are utilized for the valuation which is included in the confusion matrix as shown in Figure 2.1. In general, T_{PO} instances are ones in which both the predicted and actual results are accurate. The F_{PO} instances are ones in which the predicted outcome is true but the actual outcome is false. The T_{NE} instances are those in which both the predicted and actual outcomes are incorrect. Predicted outcomes are false in F_{NE} instances, whereas actual results are real. In image forgery detection, T_{PO} stands for the sum of images perfectly recognized as forged, F_{PO} represents the sum of images recognized wrongly as tampered, F_{NE} depicts total tampered images that were missed, and at last T_{NE} denotes the sum of authentic images perfectly recognized as authentic [10], [65].

		Actual	
		Positive	**Negative**
Predicted	**Positive**	True Positive (T_{PO})	False Positive (F_{PO})
	Negative	False Negative (F_{NE})	True Negative (T_{NE})

Figure 2.1: Confusion matrix

The fraction of the summation of T_{PO} and T_{NE} to the entire images is known as Accuracy. The probability that the detected image is genuinely forged is referred to as precision. The ability to recognize a tampered image as tampered is termed TPR (True Positive Rate) or sensitivity or recall. F_1 score is the mean of precision and recall. The ability to identify an original image as original is defined as TNR (True Negative Rate) or specificity. F_2 score is mediocre in recall and precision. The proportion of unidentified forged images is termed FPR. Mathews Correlation Coefficient is the classifier's correlation coefficient amongst actual and predicted classes. The False Negative Rate (FNR) signifies the forged region's missing detection rate. Lower values for this parameter are far better for enhancing performance. The likelihood that

the classifier is aware of the condition is stated as informedness. Markedness is the likelihood that the circumstance is marked by the classifier [10], [65]. Below are the formulae for some widely used performance measures.

$$DA = \frac{T_{PO} + T_{NE}}{F_{PO} + T_{PO} + F_{NE} + T_{NE}} \tag{2.4.1}$$

$$P = \frac{T_{PO}}{T_{PO} + F_{PO}} \tag{2.4.2}$$

$$TPR = R = Sensitivity = \frac{T_{PO}}{T_{PO} + F_{NE}} \tag{2.4.3}$$

$$TNR = Specificity = \frac{T_{NE}}{T_{NE} + F_{PO}} \tag{2.4.4}$$

$$F_1 = 2\frac{P \cdot R}{P + R} \tag{2.4.5}$$

$$F_2 = 5\frac{P \cdot R}{4 \cdot P + R} \tag{2.4.6}$$

$$FNR = \frac{F_{NE}}{T_{PO} + F_{NE}} \tag{2.4.7}$$

$$FPR = \frac{F_{PO}}{F_{PO} + T_{NE}} \tag{2.4.8}$$

$$Informedness = TPR + TNR - 1 \tag{2.4.9}$$

$$Markedness = \frac{T_{PO}}{T_{PO} + F_{PO}} + \frac{T_{NE}}{T_{NE} + F_{NE}} - 1 \tag{2.4.10}$$

$$MCC = \frac{T_{PO} \times T_{NE} - F_{PO} \times F_{NE}}{\sqrt{\left((T_{PO} + F_{PO})(T_{PO} + F_{NE})(T_{NE} + F_{PO})(T_{NE} + F_{NE})\right)}} \tag{2.4.11}$$

To quantitatively assess the efficacy of image localization, three pixel-level metrics, precision (P_p), recall (R_p), and F_1 score (F_p), are generated for forged portions of a detected forged picture. These measures are advantageous for measuring the algorithm's overall localization efficacy [149]. Precision is defined at the pixel level as the ratio of correctly identified tampered pixels to entirely identified tampered pixels. The recall is the fraction of appropriately identified tampered pixels to tampered pixels in the ground-truth tampered picture.

$$P_p = \Omega/\Omega_1 \tag{2.4.12}$$

$$R_p = \Omega / \Omega_2 \tag{2.4.13}$$

where, Ω is total successfully identified tampered pixels, Ω_2 is total tampered pixels in the ground-truth tampered picture, and Ω_1 is total identified tampered pixels. As stated beneath, F_p fuses precision and recall into a singular value.

$$F_p = 2\frac{P_p \cdot R_p}{P_p + R_p} \tag{2.4.14}$$

Furthermore, the ROC curve is a plot of Sensitivity (TPR) as the y coordinate v/s its 1-specificity (FPR) as the x coordinate, which is an efficient means of assessing the effectiveness of the suggested methodology as displayed in Figure 2.2. The ROC curves closest to the upper left corner have the best accuracy.

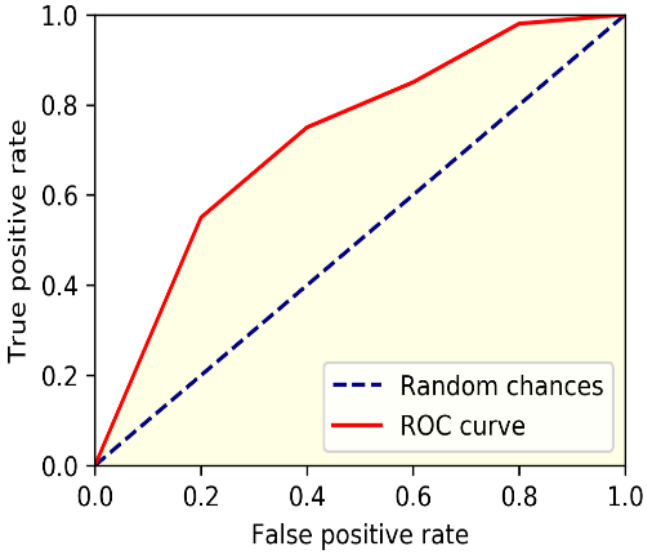


Figure 2.2: ROC Curve

2.5 Image Forgery Datasets

The image dataset is a necessary element of any image processing study since it allows researchers to analyze and compare the performance of any two competing methodologies. The efficiency of any approach can be verified by testing it on a well-designed image dataset. This section covers the image datasets that were used to test the performance of the IFDTs in this thesis. The benchmark datasets are used in the research work, which is illustrated in Table 2.1. Moreover, the examples of images from various datasets are illustrated in Figure 2.3.

Table 2.1: Dataset Descriptions for Image Forgery Detection

Datasets	Total Images			File format	Image Resolution
	Original	Tampered	Total		
IMD[1] [55]	48	48	96	PNG	3000×2300
MICC-F220[2] [48]	110	110	220	JPEG	722×480 to 800×600
MICC-F2000[2] [150]	1300	700	2000	JPEG	2048×1536
GRIP[3] [80]	80	80	160	PNG	768×1024
COVERAGE[4] [151]	100	100	200	TIF	400×486
CASIA v1.0[5] [152]	800	921	1721	JPEG	384×256
CASIA v2.0[5] [152]	7491	5123	12,614	JPEG, TIFF, BMP	240×160 to 900×600
Columbia[6] [153]	183	180	363	TIFF, BMP	757×568 to 1152×768
DVMM[7] [154]	933	912	1845	BMP	128×128
IFS-TC[8] [155]	1050	1150	2200	PNG	1024×768 to 2848×2144
DSO-1[9] [133]	100	100	200	PNG	2048×1536

[1] https://lme.tf.fau.de/dataset/image-manipulation-dataset/
[2] http://lci.micc.unifi.it/labd/2015/01/copy-move-forgery-detection-and-localization/
[3] http://www.grip.unina.it/research/83-multimedia_forensics/90-copy-move-forgery.html
[4] https://github.com/wenbihan/coverage
[5] http://forensics.idealtest.org/
[6] https://www.ee.columbia.edu/ln/dvmm/downloads/authsplcuncmp/
[7] https://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSplicedDataSet.htm
[8] http://ifc.recod.ic.unicamp.br/fc.website/index.py?sec=5
[9] https://recodbr.wordpress.com/code-n-data/#dso1_dsi1

In this thesis, two types of image forgeries, copy-move and image splicing, are detected, hence datasets including both copy-move forged images and image spliced images have been employed. The various datasets comprise various types of photographs such as animals, nature, character, indoor, and so on. Furthermore, all of the images relating to distinct datasets are of varying sizes. Thus, several datasets have been employed in the research work to demonstrate

the efficacy of the proposed methodologies and their accessibility for photos of varied sizes and origins. The following are explanations of various datasets:

- *CASIA v1.0:* The CASIA image tampering detection evaluation dataset (CITDE) offers a more puzzling as well as faithful image for the detection of tampering. The dataset comprises 800 authentic and 921 forged images [152].
- *CASIA v2.0:* This dataset comprises of 7491 authentic as well as 5123 forged images. It involves 9 categories, classified as animal, scene, architecture, plant, nature, indoor, character, article, and texture [152].
- *Columbia:* Columbia uncompressed image splicing detection evaluation dataset encloses 363 total images, out of which 183 are authentic images, else is forged. The images are in uncompressed formats i.e. BMP and TIFF. The images involve indoor sights, for instance, bookshelf, computer, or desks [153].
- *DVMM:* It is contributed by Columbia University to appraise the detection approaches. It has 933 authentic and 912 forged images. The tampering operation in this dataset has been created by cutting and pasting procedure across boundaries of the object or perpendicular/parallel strips, from a similar image or a dissimilar image [154].
- *IFS-TC:* This dataset was initially used in international competition planned by IFS-TC. It encompasses 1150 forged and 1050 authentic images [155].
- *DSO-1:* It comprises 100 authentic and 100 forged images. The dataset comprises both interiors as well as outside images. The fake images are generated by implanting one or more than one person in the original image. Various procedures such as alteration in color and illumination are executed with the motive of making realistic forged images [133].
- *IMD:* This dataset is made up of 48 authentic and 48 forged images with realistic copy-move manipulations, where an average resolution of 3000×2300 [55].
- *MICC-F220:* It comprises of equal distribution of tampered and authentic images accounting to 110 in each category, having resolutions varying 722×480 to 800×600 [48].
- *MICC-F2000:* This dataset involves 2000 images: 700 tampered and 1300 authentic of the same size, 2048×1536 [150].
- *GRIP:* This dataset contains 80 authentic and 80 forged images of the same size, 768×1024 [80].

- *COVERAGE:* This dataset consists of 100 authentic and forged images with an average resolution of 400×486. Each image contains identical but realistic objects. [151].

The efficacy of the proposed approach in Chapter 3, Chapter 5, and Chapter 6 is verified by conducting extensive simulations on datasets, i.e., IMD and GRIP. Further, CASIA v1.0, Columbia, and CASIA v2.0 datasets are utilized to demonstrate the effectiveness of the suggested approach in Chapter 4 and Chapter 5. Datasets like DVMM, DSO-1 and IFS-TC are employed in Chapter 4. Also, MICC-F220 and COVERAGE dataset is used in Chapter 5, and another dataset known as the MICC-F2000 dataset is considered in Chapter 6 to appraise the performance of the proposed work.

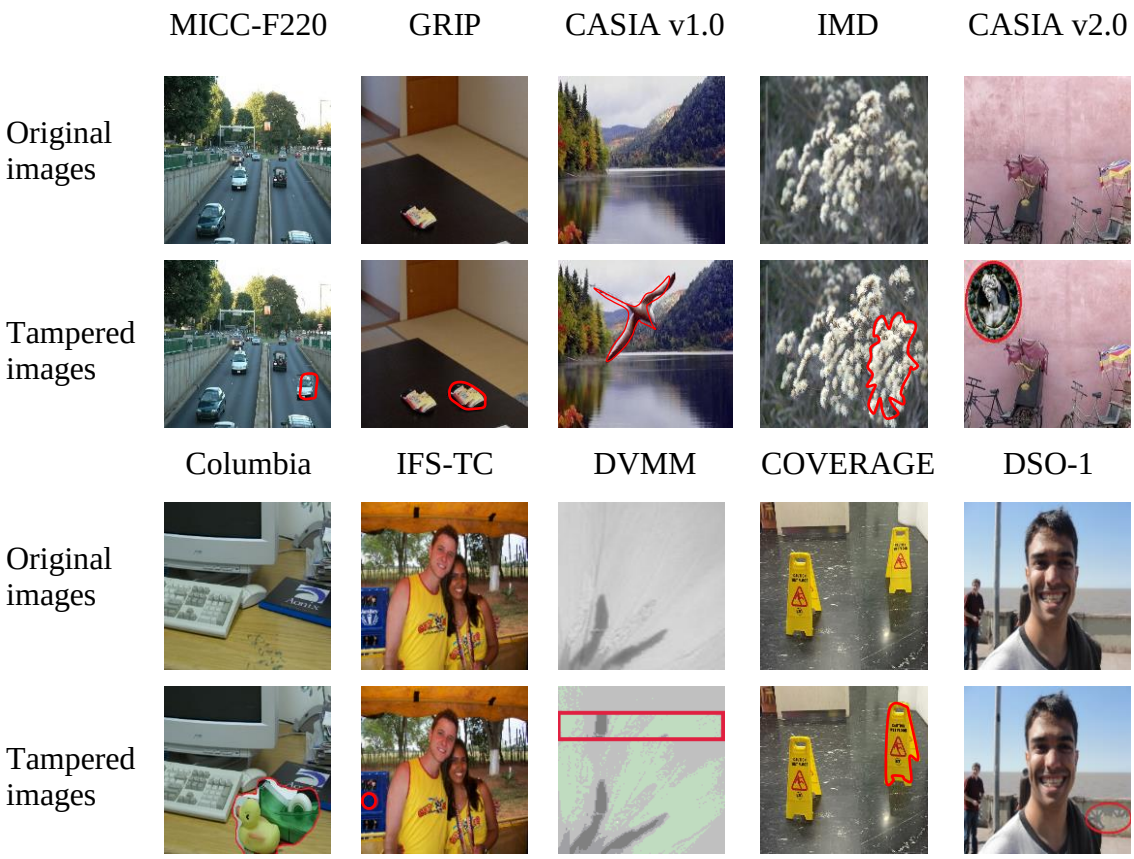


Figure 2.3: Examples of original and tampered images from different datasets, with the forged portion highlighted in red

2.6 Research Methodology

The main aspect of achieving the aforementioned research objectives is to perform a comprehensive examination of the existing literature to gather knowledge that aids in knowing

the fundamental concepts of the area of study. Figure 2.4 depicts the flow of the research work performed in this thesis.

Initially, a thorough examination of passive digital image forgery detection methodologies is conducted. This in-depth examination of the current literature on IFDTs contributes to the identification of possible solutions to open research areas. The proposed study focuses on forgery detection approaches based on CMF and ISF.

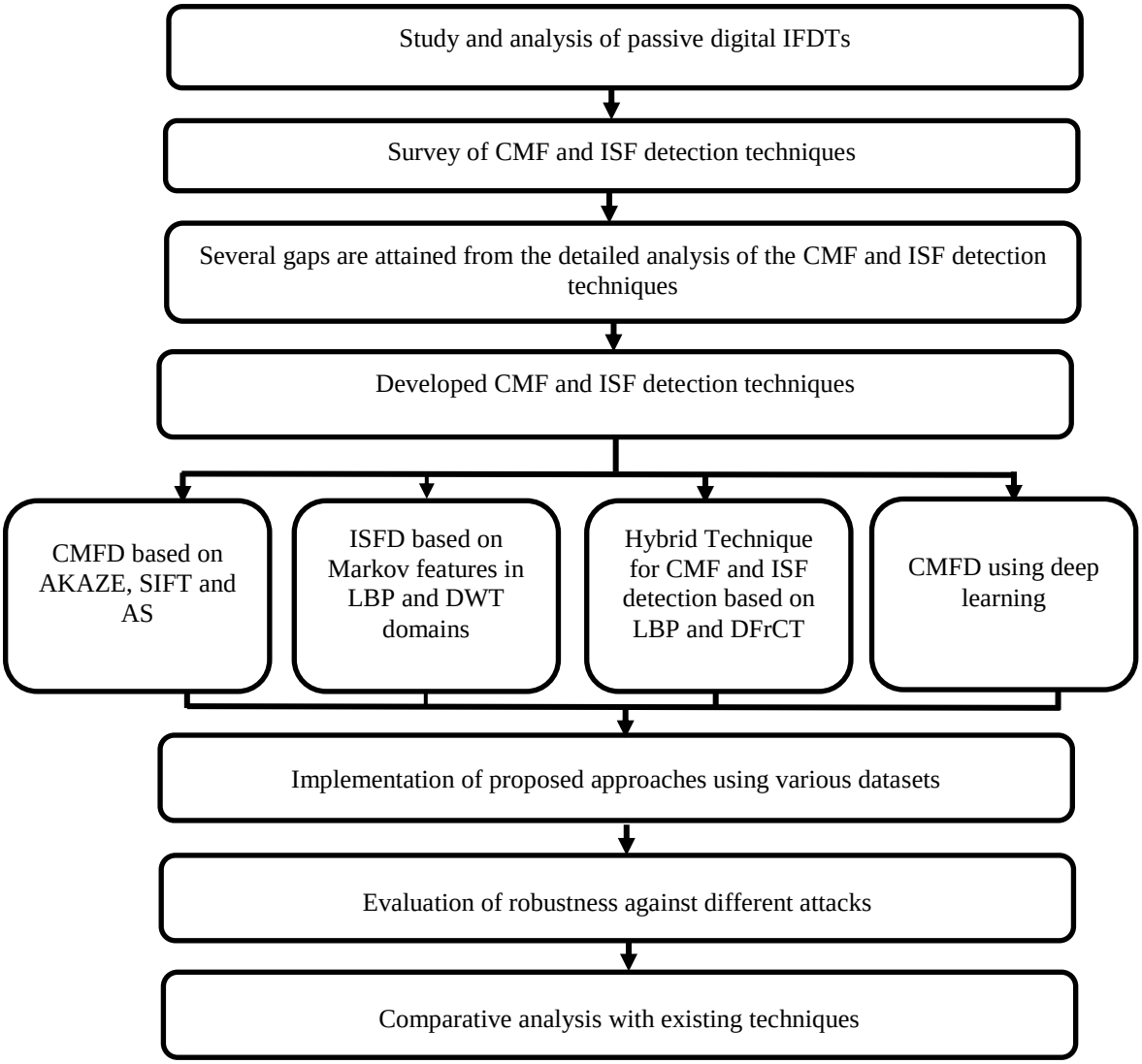


Figure 2.4: Flowchart of the work done

The suggested scheme detects CMF by combining block-based detection techniques, such as AS, and keypoint-based detection techniques, like AKAZE and SIFT. In addition, to identify ISF, Markov attributes are derived and merged from LBP and DWT areas. A passive hybrid technique based on DFrCT and LBP is also presented to identify CMF and ISF simultaneously. A deep learning CMFD framework using CLAHE and CNN is also proposed to classify

pictures as authentic or forged. Furthermore, the various image datasets like IMD, IFS-TC, GRIP, Columbia, MICC-F2000, CASIA v2.0, MICC-F220, CASIA v1.0, COVERAGE, DVMM, and DSO-1 are employed to test the proposed IFDTs. The robustness of the suggested procedure is tested against several attacks like rotation, JPEG compression, noise addition, and scaling. In addition, a comparison with existing approaches has been performed employing several metrics like F_1 score, recall, F_2 score, accuracy, precision, MCC, ROC curve, informedness, markedness, etc. In addition, cross-dataset performance and statistical analysis tests such as ANOVA are conducted to exhibit the efficacy of the suggested technique. The suggested scheme is run using MATLAB R2020a (9.8.0.1417392). The simulations are implemented on a 64-bit operating system with an 8.00 GB RAM and Microsoft Windows 8.1.

CHAPTER 3

DETECTION OF COPY-MOVE FORGERY

The comprehensive study of the literature related to digital image forensics creates the necessity to explore image forgery further to design an efficient image forgery detection technique. This chapter detects CMF proficiently by merging keypoint-based and block-based methodologies. The combination of block-based approaches, such as AS with keypoint-based methods, such as AKAZE and SIFT, makes the proposed system more resistant to various attacks and computationally inexpensive. Firstly, the test picture is transformed into various color channels. Then the C_r channel is employed since it identifies tampering artifacts remaining in the picture that human vision cannot observe. Furthermore, due to the extraction of an adequate number of key points, the CMF has identified more accurately, even in smooth areas. The experimental findings demonstrate that the suggested approach is robust to various attacks, such as JPEG compression, rotation, noise addition, and scaling, and outperforms other present techniques. Furthermore, cross-dataset performance and statistical analysis test is examined to test the efficacy of the suggested scheme.

3.1 Introduction

Digital images are one of the mediums of interaction in current society. The originality of photographs is important since they are widely used as proof in a variety of applications. As a result, IFDTs are essential to ensure the trustworthiness of digital images. CMF is generally done to conceal data or emulate specific attributes of a picture. In preceding work, CMF has been identified using either the combination of AS and SIFT [104] or the fusion of AKAZE and SIFT [88], but the combination of AS, AKAZE, and SIFT has not yet been used in the application of CMFD. Furthermore, various present methodologies [42], [47], [48], [51]–[55] are ineffective at recognising and localising multiple counterfeits in pictures. As a result, the work suggested in this chapter combines these techniques to make the suggested system computationally more effective and successful at detecting and localizing single and multiple forgeries. In addition, AKAZE is employed as its computational time (detection and description) is less than other procedures such as SURF, KAZE, SIFT, and so on. Furthermore, since AKAZE extracts a significant number of key points, the CMF is identified in flat areas. On the other hand, the SIFT procedure is effective against transformations such as rotation,

noise addition, and scaling. The AS reduces the algorithm's computational expenses by dividing the picture into blocks and is resistant to transformation like JPEG compression. Thus, in this chapter, the suggested technique based on the fusion of these procedures is also efficacious for various attacks.

3.2 Proposed Technique for CMFD

The suggested procedure's main goal is to determine if the input picture has been tampered with or not. If the tampered/forged picture is detected, then the analysis is carried out to identify the tampered sections in tampered pictures. Figure 3.1 depicts the framework of the proposed technique.

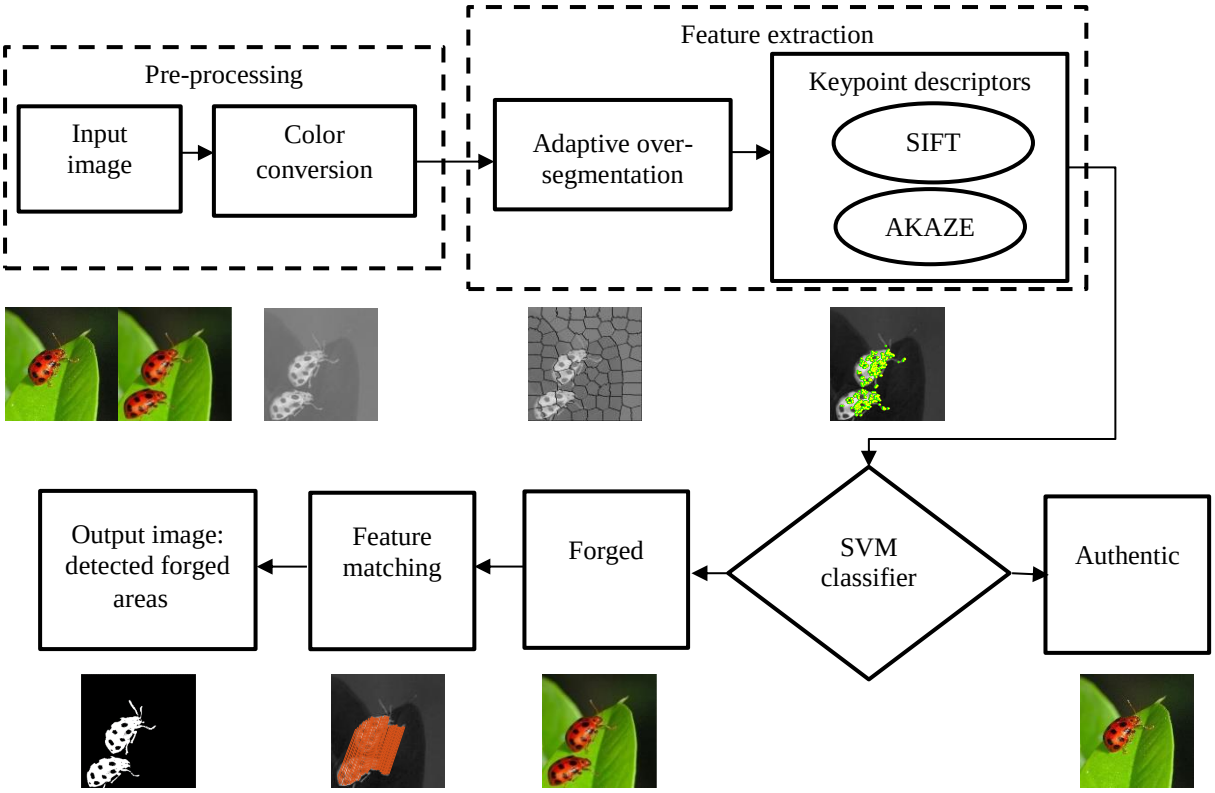


Figure 3.1: Comprehensive block diagram of the proposed methodology

The input pictures utilized in the research are collected from various datasets. It is critical to take an appropriate color space before feature extraction. As illustrated in Figure 3.2, the input picture is transformed in various color channels of YC_bC_r, RGB, and HSV color spaces. The figure shows that all components identify the image's content in detail, except for the Chroma components, which emphasize the image's poor signal content (low image detail). In most cases, the content of a picture is too powerful to conceal tampering artifacts. Because tampering might produce edge abnormalities in Chroma components, the proposed approach uses the

YC_bC_r color scheme, where Y is the luminance channel, and C_b and C_r are chrominance channels. Furthermore, a close pictorial examination of the insect's outline in Y, C_r, and C_b channels (as revealed in Figure 3.2) reveals that the Y channel preserves the most image content in contrast to chrominance channels that have significantly fewer picture data. Furthermore, human eyesight is more susceptible to luminance than to chrominance. Even though the tampered picture seems legitimate to human eyes, a few manipulation artifacts remain in the C_r channel. As a result, the proposed approach uses the C_r channel to detect tampering artifacts. The C_b and C_r components are acquired by deducting the luminance component from red and blue, respectively.

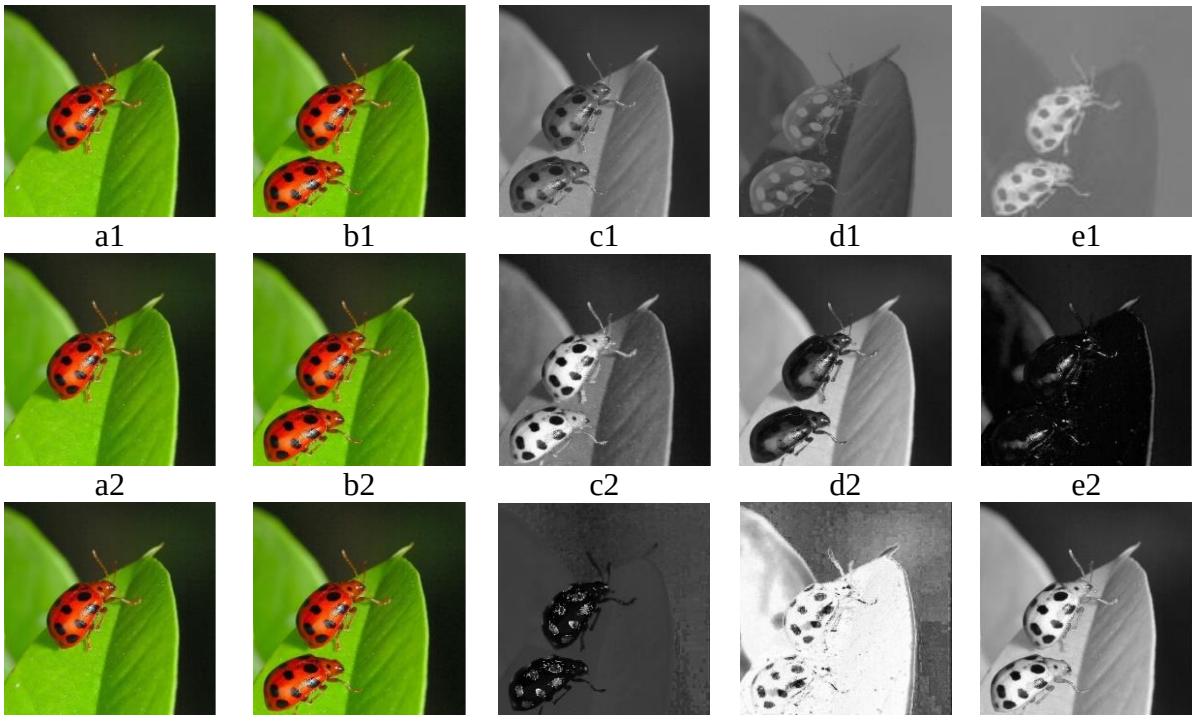


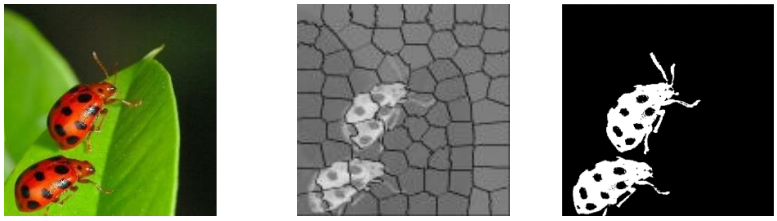
Figure 3.2: An illustration of a tampered picture with the relevant color channels (a1~a3) Original image, (b1~b3) Tampered image, (c1) Y, (d1) C_b, (e1) C_r, (c2) R, (d2) G, (e2) B, (c3) H, (d3) S, (e3) V of YC_bC_r, RGB, and HSV color spaces

The YC_bC_r picture is derived from RGB picture as follows:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \lambda \begin{bmatrix} 65.481 & 128.55 & 24.96 \\ -37.79 & -74.20 & 112 \\ 112 & -93.78 & -18.21 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (3.2.1)$$

here, the scaling factor is indicated by λ, and blue, red, and green channels of RGB picture are signified by $B, R,$ and $G,$ correspondingly. Hence, the C_r channel is utilized as it maintains

the majority of the counterfeit artifacts because the duplicated and pasted segment in Figure 3.2 is more noticeable in the C_r channel than in other color spaces [65]. Following that, keypoint-based and block-based methodologies are used to extract features. The input picture is split into overlapping even segments/blocks of a specified size in current block-based CMF detection algorithms. As a result, the detected regions are made up of even blocks incapable of representing accurate tampered areas. Consequently, the recall value of block-based approaches is quite poor. To solve these concerns, the suggested approach segments the picture into non-overlapping uneven parts using AS, which utilizes the SLICO technique. SLICO, a segmentation method, uses the k-means clustering approach to construct superpixels while effectively adhering to the bounds. The fundamental reason for using this segmentation approach is that non-overlapping segments have a lower computational cost than overlapping segments. Furthermore, irregular blocks may better show counterfeit zones than standard blocks. SLICO is done adaptively in the suggested technique based on the size of the superpixels. The input picture is first subjected to a three-level DWT utilizing the Haar wavelet; subsequently, low-frequency (L_{FE}), and high-frequency (H_{FE}) energy are obtained, as shown in Algorithm 1 [104], [156], [157]. As a consequence, the suggested method uses AS to segment the picture into segments adaptively, as illustrated in Figure 3.3.



Input: tampered image | Image obtained after AS | Output: detected forged regions

Figure 3.3: CMFD result after Adaptive Over-segmentation

In the case of even tampered areas, the starting superpixel's size is maintained quite enormous in AS to ensure that the superpixels may aggregate to the edges and have great feature points for tampering detection. Furthermore, bigger superpixels include fewer sections, which minimizes the computing cost associated with matching blocks. Alternatively, a more detailed texture of tampered area has a smaller superpixel size to assure excellent detection outcomes. After segmenting the picture, several keypoint descriptors such as SIFT, SURF, AKAZE, and KAZE are utilized [48], [52], [53], [88], [97], [158]–[161], and the optimal combination is

employed. Based on the findings described in Section 3.3, it is determined that the fusion of AS, SIFT, and AKAZE surpasses other combinations.

Algorithm 1: Pseudocode of Adaptive Over-segmentation

procedure

Input picture Y of size $W \times V$

Evaluate energy of high-frequency H_{FE} and low-frequency L_{FE}

$$L_{FE} = \sum |AC_3|$$

$$H_{FE} = \sum \left(\sum |DC_{kk}| + \sum |HC_{kk}| + \sum |VC_{kk}| \right), \; kk = 1,2,3$$

where, DWT's approximation coefficient is signified by AC_3 and DWT's detailed coefficients are indicated by VC_{kk}, DC_{kk}, and HC_{kk}

Calculate the proportion of low-frequency coefficients using:

$$L_{FC} = (L_{FE} / (L_{FE} + H_{FE})) \times 100\%$$

Calculate superpixels' size S_{PP} after the assessment of L_{FC} as shown:

if $L_{FC} > 50\%$, then S_{PP} is evaluated as:

$$S_{PP} = (0.02 \times V \times W)^{1/2}$$

end if

if $L_{FC} \leq 50\%$, S_{PP} is evaluated as:

$$S_{PP} = (0.01 \times V \times W)^{1/2}$$

end if

end procedure

The AKAZE method is useful for recognizing key points across uniform regions in CMFD. This technique uses a non-linear scale-space to blur the picture data, which minimizes noise without interfering with the picture data. The non-linear diffusion system models the picture's brightness by raising the scaling levels. The diffusion process is controlled by the divergence of a flow function, which is done using a partial differential equation [88], [160]. The following equation depicts non-linear diffusion:

$$\frac{\partial LI}{\partial t} = div(C(x,y,t) \nabla LI) \tag{3.2.2}$$

where t is scale factor, luminance of the picture is represented by LI , ∇ is gradient operator, and conductivity function $C(x,y,t)$ is specified as:

$$C(x,y,t) = G(|\nabla LI_{\sigma}(x,y,t)|) \tag{3.2.3}$$

LI_{σ} is smoothed Gaussian form of picture and ∇LI_{σ} is a gradient of LI_{σ}. In the present study, one of the two conductivity functions developed by Perona and Malik is explored [161]. The

conductivity function in AKAZE that supports larger regions over smaller ones is illustrated by G_2 as given below:

$$G_2 = \frac{1}{1+\left|\nabla LI_\sigma\right|^2 \Big/ \Phi^2} \tag{3.2.4}$$

The contrast element signified by Φ influences the amount of diffusion and is associated with edge data. The lower value of Φ stores more edge data. As an estimated value for Φ, this method uses 70% of the gradient histogram ∇LI_σ. AKAZE generates scale pictures at each scale using Fast Explicit Diffusion (FED). FED is a basic explicit approach that employs inconsistent time step sizes and is substantially faster than other strategies. AKAZE detector is based on the Hessian matrix determinant, and Scharr filters are employed to improve rotation invariance quality. AKAZE's descriptor also relies on the very effective Modified Local Difference Binary (MLDB) technique. Due to the nonlinear scale-spaces, AKAZE is used since it is resilient to rotation and scale and has more remarkable uniqueness at changing scales [88], [160]. The SIFT descriptor is used to retrieve the SIFT attributes of a 128-D feature vector as described in Chapter 2 [48], [88].

Numerous tests are carried out to examine the suggested system in various methods to prove its effectiveness. The suggested approach (AS+SIFT+AKAZE) is compared to the block-based technique (AS coupled with SIFT, AKAZE, KAZE, and SURF) in Table 3.1. Based on the findings described in Section 3.3, it is determined that the suggested method, i.e., AS+SIFT+AKAZE, outperforms other combinations. As a consequence, the SIFT-extracted characteristics are combined with AKAZE-extracted features. AKAZE with points $AK = \{AK_1,...,AK_d\}$ consists of 64-D feature vectors $hk = \{hk_1,\ldots,hk_d\}$ and SIFT with points $SF = \{SF_1,...,SF_c\}$ consists of 128-D feature vectors $fd = \{fd_1,..., fd_c\}$. AKAZE attributes are horizontally merged with themselves, and further, both attributes are vertically fused to generate a single feature vector with equivalent dimensions [88]. Significant characteristics such as Gray Level Run Length Matrix (GLRLM), GLCM, and histogram points are derived from the merged feature vector to minimize the size of the feature vector. The fundamental goal of feature extraction is to transform the input into a feature set for reducing features since the procedure's input is too large to be handled and is predicted to be redundant. Histogram attributes such as skewness, kurtosis, mean, variance, standard deviation, and entropy are derived from first-order statistics. In contrast, GLCM and GLRLM are derived from second-order and higher-order statistics, respectively [162].

Furthermore, the SVM classifier with Radial Basis Function (RBF), kernel has been used to differentiate between authentic and faked photos. The SVM is selected for the suggested work since it is among the most robust and exact classification systems available. Some binary classification problems lack a basic hyperplane as a useful separation criterion. For these concerns, there is a mathematical technique that retains all of the easiness of SVM separating hyperplane. The SVM classifier's basic concept is to provide a computationally proficient technique for learning 'excellent' by splitting hyperplanes across distinct classes into a higher dimensional space. The SVM classifier recognizes a collection of linearly separable hyperplanes that are linear functions of the higher dimensional space. The hyperplanes are positioned to ensure the most significant possible distance from both classes. The input feature vectors are translated to higher dimensional space by employing a kernel function. The three most frequent kernels are RBF, sigmoid and polynomial. The suggested technique employs the RBF kernel, which has high efficiency and versatility. The RBF kernel is denoted as:

$$\phi(m, p) = \exp\left(-\gamma \left\| m_i - p_i \right\|^2\right) \quad (\gamma > 0) \tag{3.2.5}$$

where the kernel function that translates the input into high dimensionality feature space is signified by $\phi(m, p)$. When overfitting occurs, γ (a free factor) is utilized to equalize the kernel function. Often, LIBSVM with an RBF kernel is used to train SVM classifiers. To select the optimum value γ, the grid search technique with five-fold cross-validation is used. The SVM classifier employing the RBF kernel is shown in Figure 3.4. It is divided into two phases: training and testing, which are parted using a hyperplane. $\{m_i, n_i\}$ signifies the training dataset for $i = 1,2,...,d$, where m_i is training data, n_i is target output and test input vector is indicated by p_i.

The suggested approach performs SVM classifier training using LIBSVM with RBF kernel. Thus, feature vectors are fed into an SVM classifier, which is utilized to differentiate between two classes, original and tampered [97], [163]. Once the SVM classifier has declared the pictures as tampered, feature matching is used to properly identify the tampered areas. The K-Nearest Neighbor (kNN) search is used to find the equivalent key points in the area. The suggested method finds the closest neighbor by using Euclidean distance. The kNN search is employed because it is a quick, convenient, and effective method for matching features.

In addition, many tampered areas in digital photographs are discovered. Finally, the detected copy-move forged sections are generated using the morphological closing operation. In the

detected forged picture, the morphological process removes tiny areas, minimizes isolated pixels, and fills small gaps.

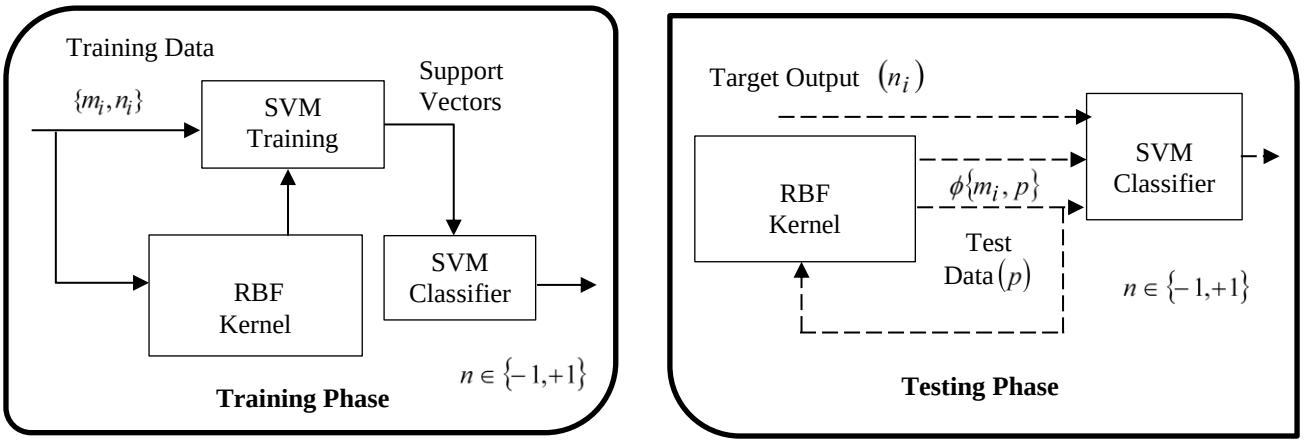


Figure 3.4: RBF kernel-based two-class SVM classifier schematic diagram

Initially, a collection of key points is retrieved from the tampered picture, and the Euclidean distance between the key points is calculated. The k=2 parameter in kNN indicates that the nominated keypoint is compared to its second closest neighbor. The compared key points are similar if the proportion of the nominated keypoint's Euclidean distance to the second closest neighbor's Euclidean distance is less than the threshold value (thr). The choice of thr effects the performance of computing equivalent keypoint pairs. When thr reaches 1, numerous keypoint pairs are retrieved, which helps to define the tampered areas more effectively, but it can result in more deceptively identified areas. If thr approaches zero, the trade-off is assessed in the other direction. A good matching technique must preserve as many true-matching keypoint sets as feasible though eliminating the majority of false-matching pairs. The suggested study uses $thr = 0.6$, as it is the most often used value in prior works [48], [49]. The feature matching algorithm is shown below.

Algorithm 2: Feature matching using KNN

Input- Tampered picture
Output- Mark the tampered areas

procedure

Input tampered picture
Evaluate the keypoints $\left(kpp = \{kpp_1, kpp_2, \cdots, kpp_{npp}\}\right)$ from the tampered picture
Compute the Euclidean distance $\left(e = \{e_1, e_2, \cdots, e_{npp-1}\}\right)$ between the nominated keypoint and neighboring keypoint
Sort the key points in ascending order i.e. $e_1 \le e_2 \le \cdots \le e_{npp-1}$

if $e_1 / e_2 < thr,$ where $thr \in (0,1)$

The nominated keypoint is compared with its neighbor

Highlight the identified areas in the tampered picture

else

The key points do not correspond to each other

end if

Fill the minor gaps in a detected tampered picture using morphological close operation

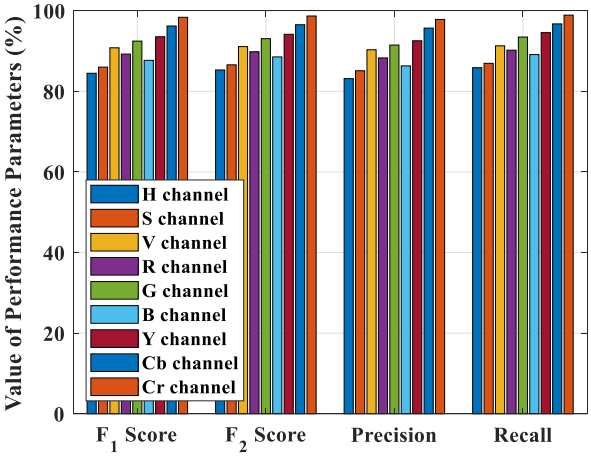
end procedure

3.3 Experimental Results and Discussions

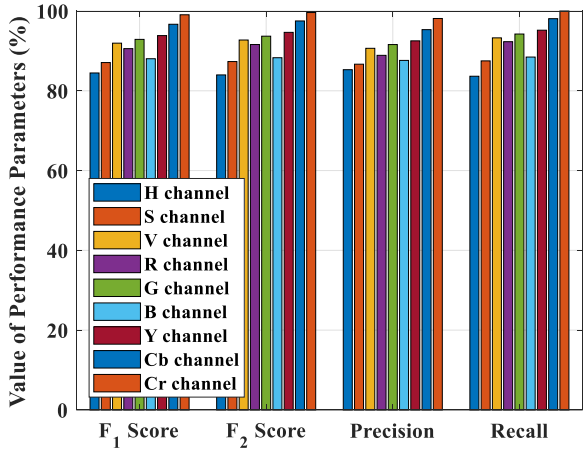
Various experimentations are carried out to evaluate the robustness and efficacy of the suggested procedure. Four datasets, namely MICC-F220, GRIP, COVERAGE, and IMD are employed in the suggested approach, and a comparative analysis has been performed. Performance measures like recall, F_1 score, precision, and F_2 score are assessed based on the SVM classifier's predicted results.

3.3.1 Simulation Results of CMF

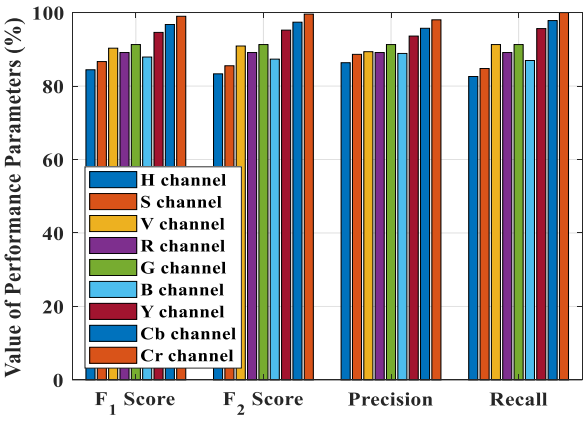
In the beginning, numerous performance metrics for various color channels, as mentioned in Section 3.2, are examined to monitor the performance of these various channels. Figure 3.5 compares performance metrics for color channels on various datasets. Figure 3.5 shows that the C_r channel has improved precision, F_1 score, recall, and F_2 score than other channels, indicating that the C_r channel is the finest for the suggested approach for detecting CMF. Experimentations are also carried out to evaluate the comprehensive performance of the suggested technique. On the relevant datasets, the results of the combination of AS, AKAZE, and SIFT are compared to the results of AS with KAZE, AKAZE, SURF, and SIFT separately as given in Table 3.1.



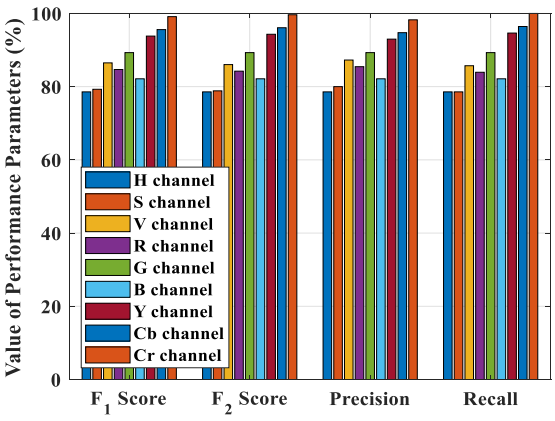
(a) COVERAGE



(b) MICC-F220



(c) IMD



(d) GRIP

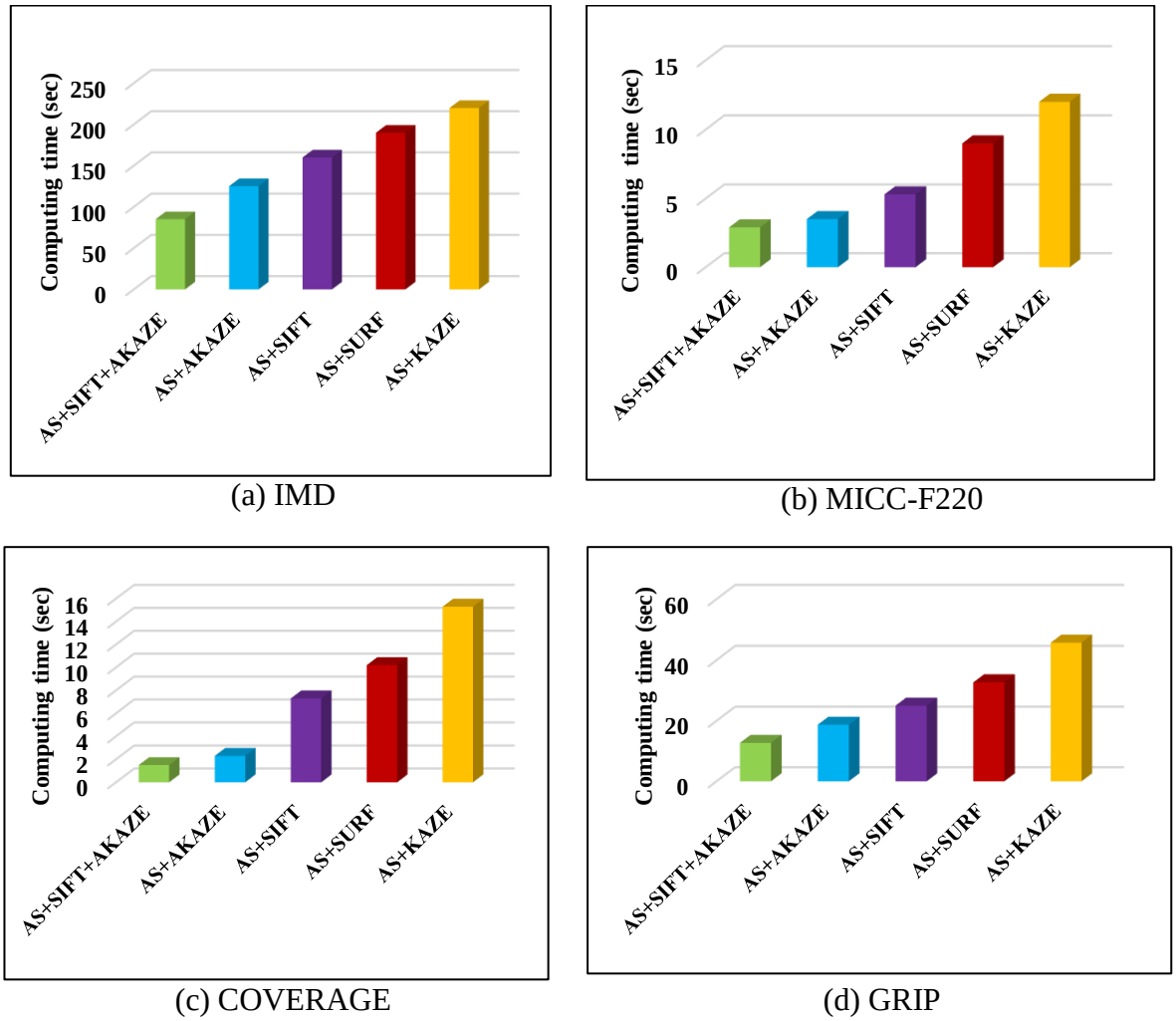
Figure 3.5: Metrics for distinct color channels across various datasets

Table 3.1: Comparative analysis of performance metrics (%) for several feature descriptors

Dataset	Feature descriptors	Recall	F_1 score	Precision	F_2 score
IMD	AS + SIFT	100	98.52	97.09	99.40
	AS + AKAZE	100	99.01	98.04	99.60
	AS + SURF	98.00	97.03	96.08	97.61
	AS + KAZE	97.00	96.04	95.10	96.61
	AS + SIFT+ AKAZE	**100**	**99.50**	**99.01**	**99.80**
MICC-F220	AS + SIFT	100	98.88	97.79	99.55
	AS + AKAZE	100	99.06	98.13	99.62
	AS + SURF	98.50	97.40	96.32	98.05
	AS + KAZE	97.74	96.65	95.59	97.31
	AS + SIFT+ AKAZE	**100**	**99.53**	**99.06**	**99.81**
COVERAGE	AS + SIFT	98.00	97.03	96.08	97.61
	AS + AKAZE	98.91	98.38	97.85	98.70
	AS + SURF	96.74	96.22	95.70	96.53
	AS + KAZE	94.57	94.05	93.55	94.36
	AS + SIFT+ AKAZE	**99.46**	**99.19**	**98.92**	**99.35**
GRIP	AS + SIFT	100	99.03	98.08	99.61
	AS + AKAZE	100	99.12	98.25	99.64
	AS + SURF	97.83	97.56	97.30	97.72
	AS + KAZE	96.20	95.93	95.68	96.09
	AS + SIFT+ AKAZE	**100**	**99.56**	**99.12**	**99.82**

Bold indicates the maximum value

Table 3.1 shows that combining AS, SIFT, and AKAZE yields noteworthy results when compared to other combinations for several performance characteristics like recall, F_1 score, precision, and F_2 score. Furthermore, as shown by Alcantarilla [156], AKAZE descriptors are quicker than KAZE, SIFT, and SURF descriptors, and AS minimizes the computing cost. Thus, combining these strategies reduces the total computing time of the suggested strategy. Figure 3.6 illustrates a comparison of calculation time for various features. The figure indicates that the suggested technique is quicker to evaluate than others and performs well in identifying CMF.



(a) IMD (b) MICC-F220

(c) COVERAGE (d) GRIP

Figure 3.6: Comparison of computing time on various datasets using different feature descriptors

Several experiments are also carried out for feature reduction. At first, the outcomes are achieved using distinct features, such as histogram, GLCM, or GLRLM, the combination of these features is assessed on the IMD dataset, as seen in Figure 3.7. The figure displays that combining all of the feature vectors mentioned above results in a considerable boost in

performance metrics. Consequently, the following simulations are run on various datasets utilizing the fusion of all features, namely GLCM, GLRLM, and histogram.

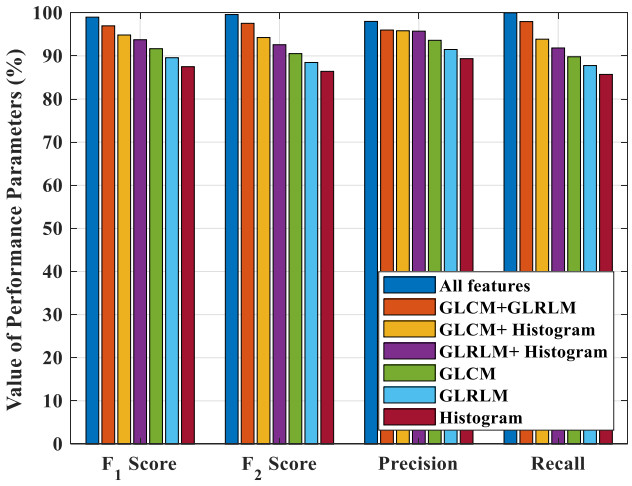


Figure 3.7: Impact of different features on proposed scheme's performance

Furthermore, the performance of the SVM classifier is compared to other classifiers, such as naive Bayes, decision tree, random forest, and discriminant analysis, to assess its effectiveness. All classifiers are applied under a similar setting to provide a fair assessment, i.e., modeling with an equivalent dataset (IMD), as displayed in Figure 3.8. The figure demonstrates that, when compared to other classifiers, the SVM classifier achieves the greatest efficacy for precision, F_1 score, recall, and F_2 score. It indicates that the SVM has higher classification power and proficiency than other classifiers for categorizing authentic and tampered photos.

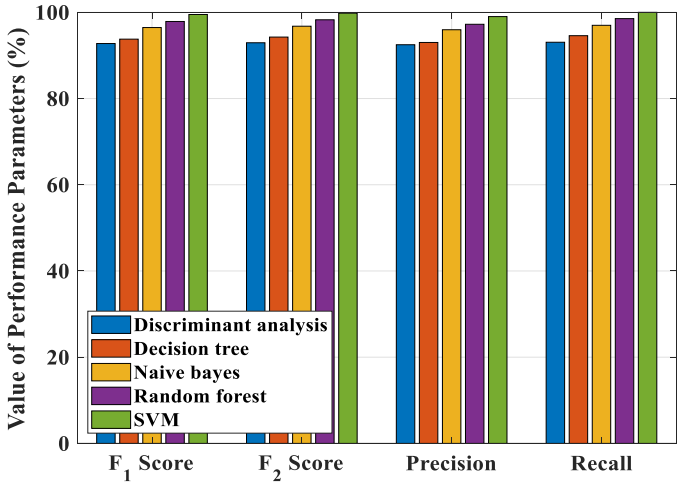


Figure 3.8: Comparison of various performance metrics for different classifiers

3.3.2 Localization Results of CMF

The suggested method's performance is examined in the existence of single and multiple copied areas in the experimentation by localizing the tampered area. Pixel-level measures, viz.

precision (P_p), F_1 score (F_p), and recall (R_p) are analyzed to quantify the efficacy of image localization for tampered areas of an identified tampered picture. Figures 3.9 and 3.10 show the value of these measures and localization outcomes for single and multiple CMFs, correspondingly.

(a) Single CMF: A tampered portion is duplicated and pasted over another portion in the equivalent picture in a single CMF to change the authentic picture. For visual aid, Figure 3.9 compares the detection results of a single CMFD to the ground truth. The authentic and tampered photos are shown in the first and second rows of Figure 3.9, correspondingly. Figure 3.9 shows that the suggested CMFD approach identifies the single tampering successfully.

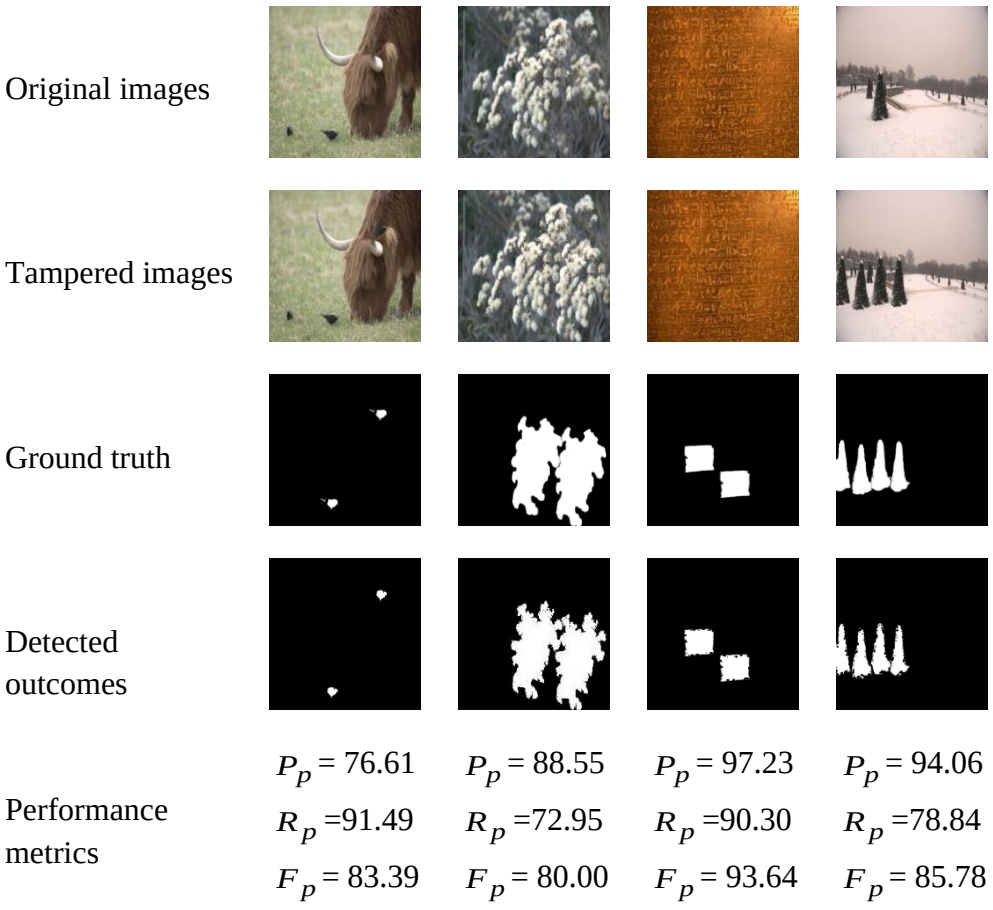


Figure 3.9: CMFD outcomes for single forgery

(b) Multiple CMF: In this, numerous copies of similar tampered areas are produced, i.e., one tampered portion is duplicated and pasted in the same picture in separate portions. Figure 3.10 shows that the suggested approach can accurately identify multiple forgeries in the picture.

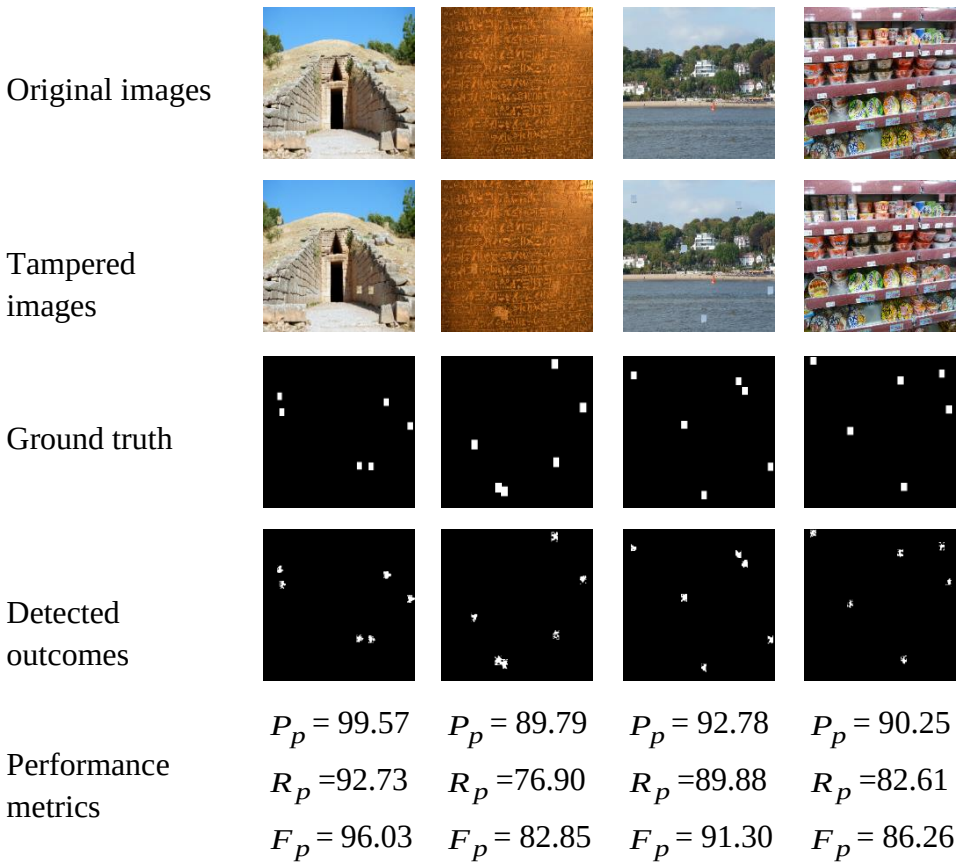


Figure 3.10: CMFD outcomes of the presented scheme for multiple forgeries

3.3.3 Comparative Analysis

A comparison study is performed on the datasets mentioned above to verify the efficacy of the suggested technique. Table 3.2 compares the proposed approach's different performance metrics to those of current techniques.

According to Table 3.2, our suggested technique provides better detection results in terms of several performance characteristics. In IMD, the suggested strategy obtains a recall of 100%, 99.50% F_1 score, and 99.80% F_2 score, which are much higher than the current strategies. The approach suggested by Chen [103] has a higher precision value than the suggested procedure but has a worse recall, F_1 score, and F_2 score. On the other hand, the suggested approach surpasses the other current strategies for the MICC-F220 dataset with improved precision, F_2 score, F_1 score, and recall of 99.06%, 99.81%, 99.53%, and 100%, correspondingly. Precision, F_1 score, recall, and F_2 score for the COVERAGE dataset are 98.92%, 99.19%, 99.46%, and 99.35%, respectively, which surpasses other cutting-edge algorithms.

Table 3.2: Comparative analysis of performance metrics (%) of presented methodology with existing methodologies

Dataset	Methodologies	Precision	Recall	F_1 score	F_2 score
IMD	Bravo [42]	94.00	97.92	95.92	97.11
	Pan [47]	88.37	79.17	83.52	80.92
	Pun [104]	95.92	97.92	96.91	97.51
	Christlein [55]	90.56	100	95.05	97.95
	Amerini [48]	86.40	66.67	75.29	69.86
	Yang [52]	90.27	78.61	84.04	80.69
	Wang [53]	88.24	93.75	90.91	92.59
	Cozzolino [80]	92.15	97.92	94.95	96.71
	Sun [106]	90.91	83.33	86.96	84.74
	Prakash [88]	92.30	87.80	89.98	88.65
	Priyanka [51]	90.03	97.12	93.44	95.61
	Ouyang [46]	94.50	100	97.17	98.84
	Meena [112]	NE	NE	96.97	NE
	Tahaoglu [54]	NE	NE	94.00	NE
	Li [49]	NE	100	98.97	NE
	Chen [103]	**99.96**	98.59	99.24	98.86
	Zhong [124]	70.85	58.85	64.29	60.91
	Tinnathi [114]	94.52	95.32	93.56	NE
	Proposed	99.01	**100**	**99.50**	**99.80**
MICC-F220	Christlein [55]	76.68	20.91	32.86	24.46
	Amerini [48]	91.53	98.18	94.74	96.77
	Bravo [42]	74.95	19.09	30.43	22.43
	Cozzolino [80]	83.42	84.55	83.98	84.32
	Ojeniyi [107]	93.86	97.27	95.53	96.56
	Bilal [96]	95.64	96.30	95.96	96.16
	Tahaoglu [54]	NE	100	NE	NE
	Li [49]	NE	100	99.10	NE
	Niyishaka [111]	94.49	93.63	94.05	93.80
	Proposed	**99.06**	**100**	**99.53**	**99.81**

Dataset	Methodologies	Precision	Recall	F_1 score	F_2 score
	Amerini [48]	40.43	85.71	54.95	70.02
	Bravo [42]	**99.98**	50.55	67.15	56.09
	Christlein [55]	75.00	46.15	57.14	49.99
COVERAGE	Cozzolino [80]	61.97	59.34	65.45	67.72
	Li [49]	NE	80.22	72.28	NE
	Proposed	98.92	**99.46**	**99.19**	**99.35**
	Chen [103]	93.5	90.00	91.72	90.67
	Christlein [55]	74.76	100	73.68	71.39
	Meena [84]	NE	NE	98.76	NE
	Cozzolino [80]	91.85	98.75	95.18	97.28
GRIP	Amerini [48]	77.56	70.00	73.68	71.39
	Wang [53]	91.76	97.50	94.54	96.29
	Bravo [42]	95.59	97.74	96.65	97.31
	Li [49]	NE	100	**100**	NE
	Sun [106]	90.54	83.75	87.01	85.02
	Proposed	**99.12**	**100**	99.56	**99.82**

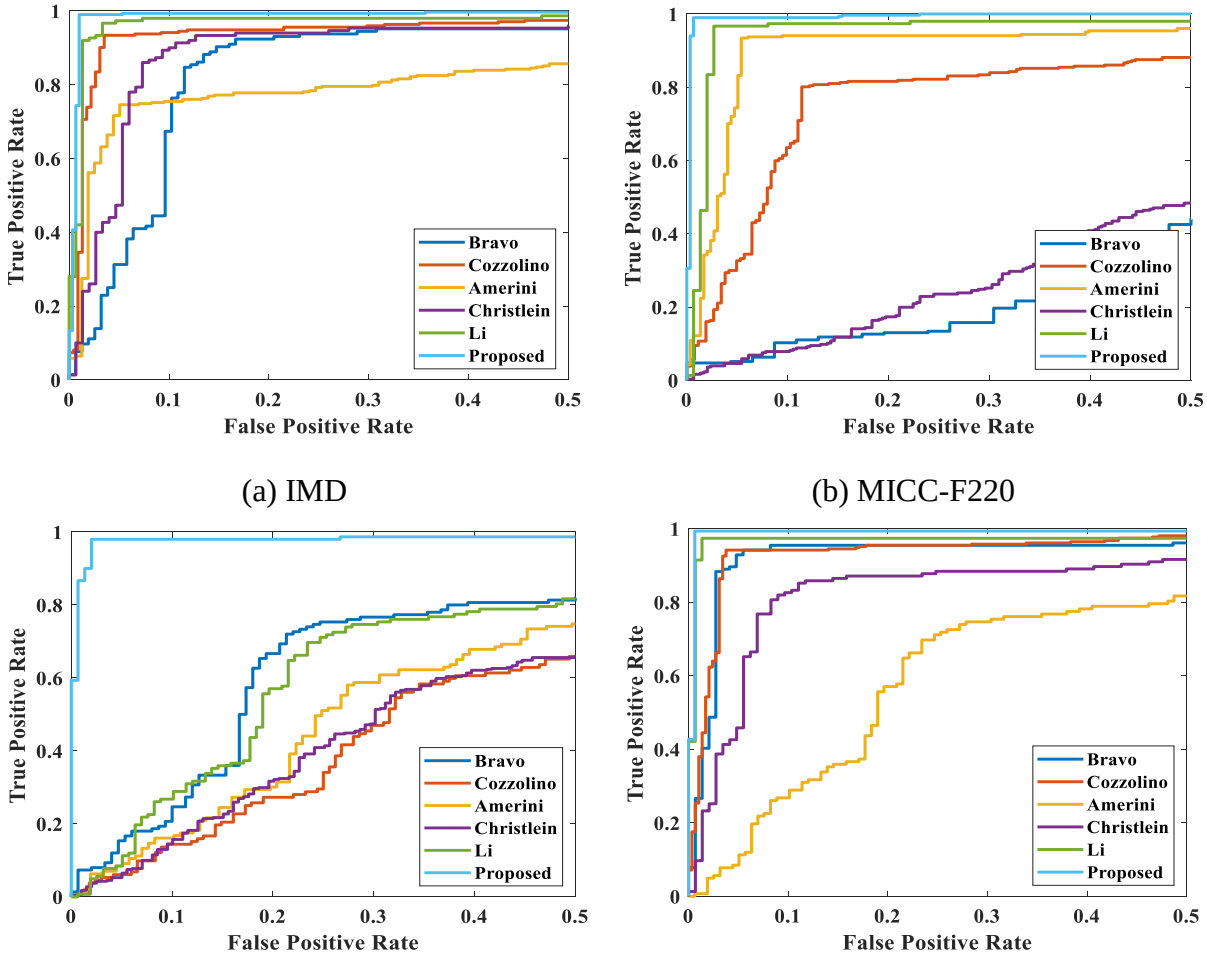
NE: Not evaluated by the respective author and Bold indicates the maximum value

Although Bravo's [42] approach has a higher precision value than the suggested technique, it has a worse recall, F_1 score, and F_2 score. Likewise, the simulations are conducted on dataset GRIP with a 99.82% F_2 score, 99.12% precision, and 100% recall. Nonetheless, on the GRIP dataset, the approach described by Li [49] has a marginally better F_1 score than the suggested system but at a high CPU time. Figure 3.11 shows the zoomed ROC curves for clear visibility for comparison. As a result, the suggested technique's ROC curves are nearer to the top-left edge signifying maximum accuracy. Furthermore, the Area Under the ROC Curve (AUC) metric has been analyzed to provide a quantitative comparison in Table 3.3, which represents the area under curve obtained by the ROC curve. The maximum value of AUC shows that the proposed method is effective for classifying images as authentic or forged.

3.3.4 Run-Time Analysis

The run time analysis calculates the average CPU time needed by an algorithm. Consequently, the average CPU time of various strategies is compared with the presented methodology on all

four datasets, as shown in Figure 3.12. Additionally, computational complexity is a crucial performance metric to assess the suggested technique's performance. The suggested method's computational complexity is separated into three steps.



(a) IMD (b) MICC-F220

(c) COVERAGE (d) GRIP

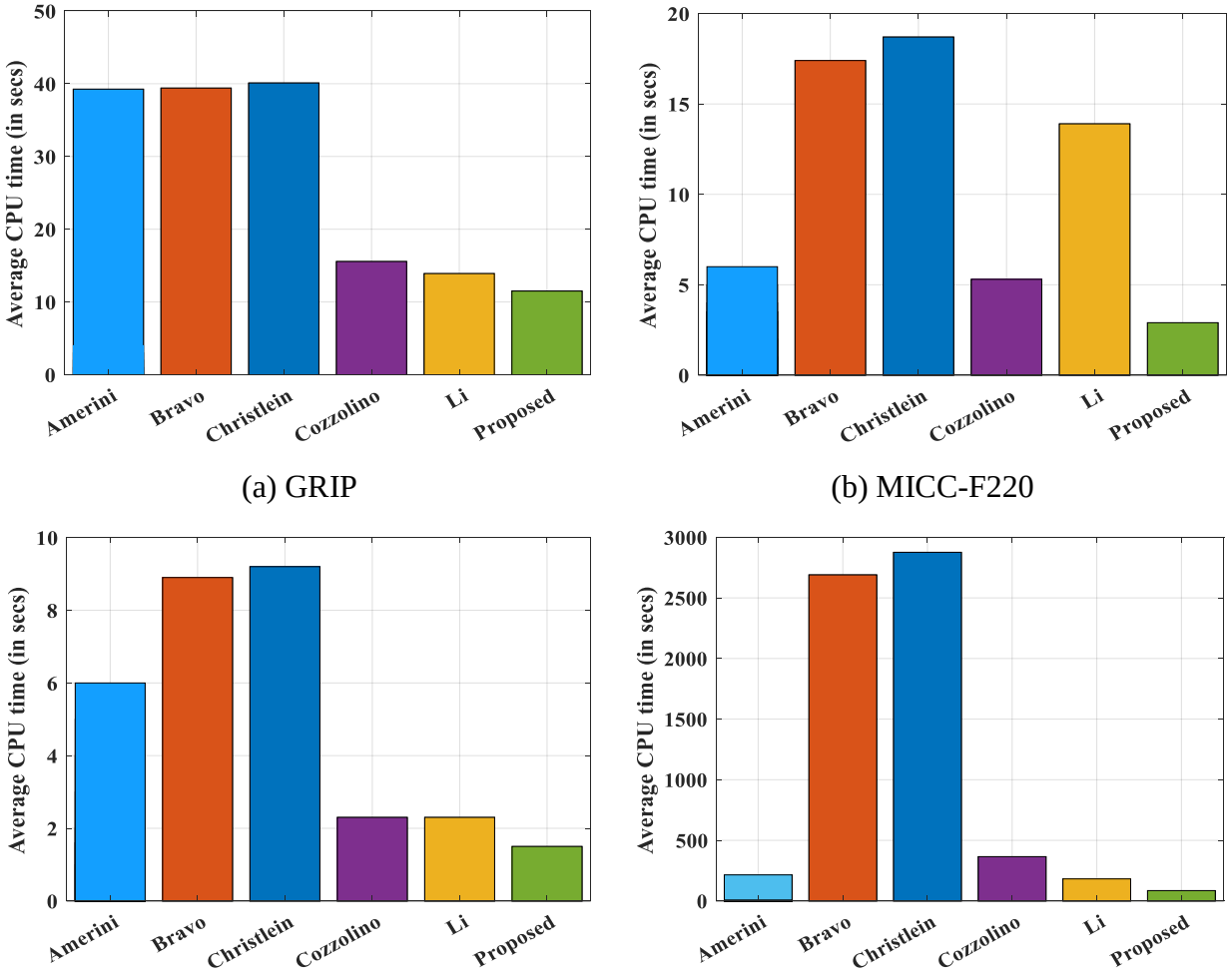
Figure 3.11: ROC curve's comparison of presented approach with existing approaches on various datasets

Table 3.3: Comparison of presented approach's AUC with existing approaches

	IMD	**MICC-F220**	**COVERAGE**	**GRIP**
Bravo [42]	94.02	73.29	80.45	96.24
Cozzolino [80]	92.51	83.12	65.23	74.25
Amerini [48]	85.70	90.53	72.56	78.57
Christlein [55]	91.02	78.61	67.26	76.29
Li [49]	97.54	98.02	79.22	90.75
Proposed	**99.02**	**99.01**	**98.90**	**99.13**

Bold indicates the maximum value

The first step entails feature extraction, which is dependent on both methodologies, namely block-based (AS) and keypoint-based (SIFT, AKAZE). AS employs DWT and SLICO in its block-based methodology. The computational complexity of both DWT and SLICO is $O(n)$ [160], [164]. As a result, the overall computational cost of AS is $2O(n)$. SIFT and AKAZE have $O(n^2)$ computational costs [106], [165]. So, keypoint-based methodologies have $2O(n^2)$ computational complexity. Then, the second step comprises of SVM classifier with a computing cost of $O(n^3)$ [166]. The next step involves feature matching employing kNN with the computing cost of $O(nm)$ [167]. Consequently, after addition, the total computational cost of the suggested methodology is $2(O(n)+O(n^2))+O(n^3)+O(nm)$. It is clear from the graphs in Figure 3.12 that the suggested methodology consumes less time for execution in comparison to the current methodologies for all the datasets.



(a) GRIP (b) MICC-F220

(c) COVERAGE (d) IMD

Figure 3.12: Comparative study of average CPU time on various datasets

3.4 Robustness Under Various Attacks

This section examines the robustness of the presented methodology by conducting several tests under different attacks. In this scenario, tampered pictures are created by utilizing the IMD dataset's 48 images, and the duplicated parts are attacked by transforms such as noise addition, rotation, JPEG compression, and scaling [54], [104].

(a) Rotation: The cloned parts of the picture are rotated in stages of 2^o with a rotation angle ranging from 2^o to 10^o. This scenario runs a test on a total of $48 \times 5 = 240$ photos. The top row of Figure 3.13 shows original photos with no forging, whereas the second-row shows tampered pictures that are rotationally invariant at various angles of 2^o, 4^o, 6^o, and 8^o, respectively. The suggested technique's detected forged area is shown in the fourth row of Figure 3.13 and compared to ground truth for visual aid.

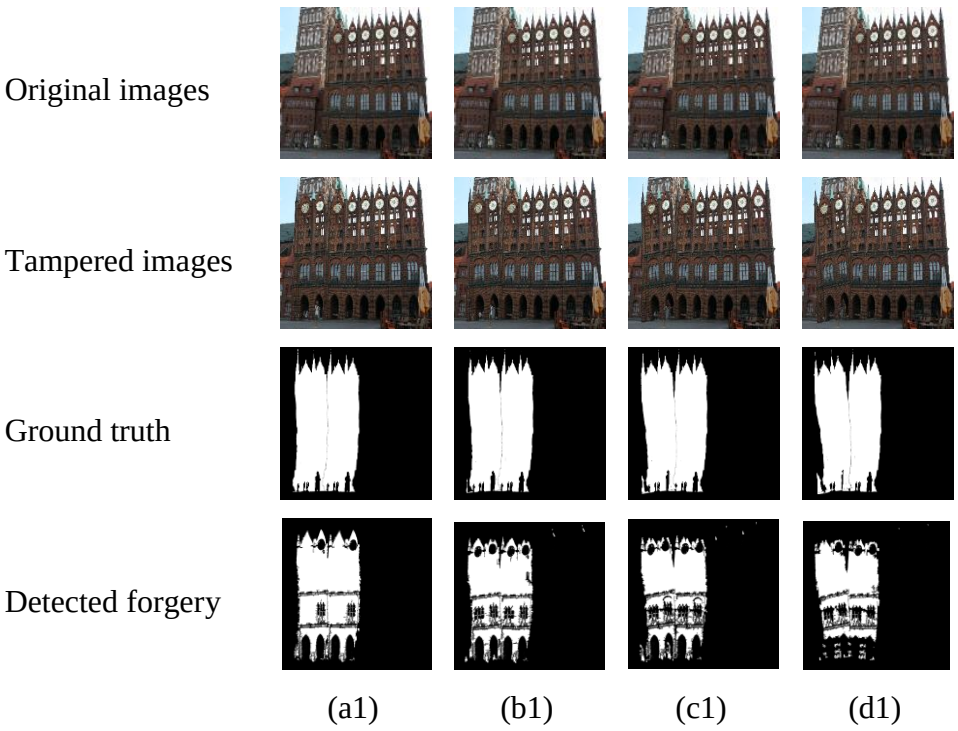


Figure 3.13: CMFD outcomes of presented methodology for (a1) 2^o, (b1) 4^o, (c1) 6^o, (d1) 8^o rotation degree

(b) Scaling: The duplicated portions are rescaled with a step size of 2% between 91% and 109%. The simulations are run on $48 \times 10 = 480$ photos in this example. Detection outcomes are provided in Figure 3.14, where the tampered areas are rescaled at various scaling factors, 97, 99, 101, and 103, as indicated in the final row of Figure 3.14. When the detection outcomes

are matched to the ground truth, the suggested CMF approach effectively detects the forgery under scaling attack.

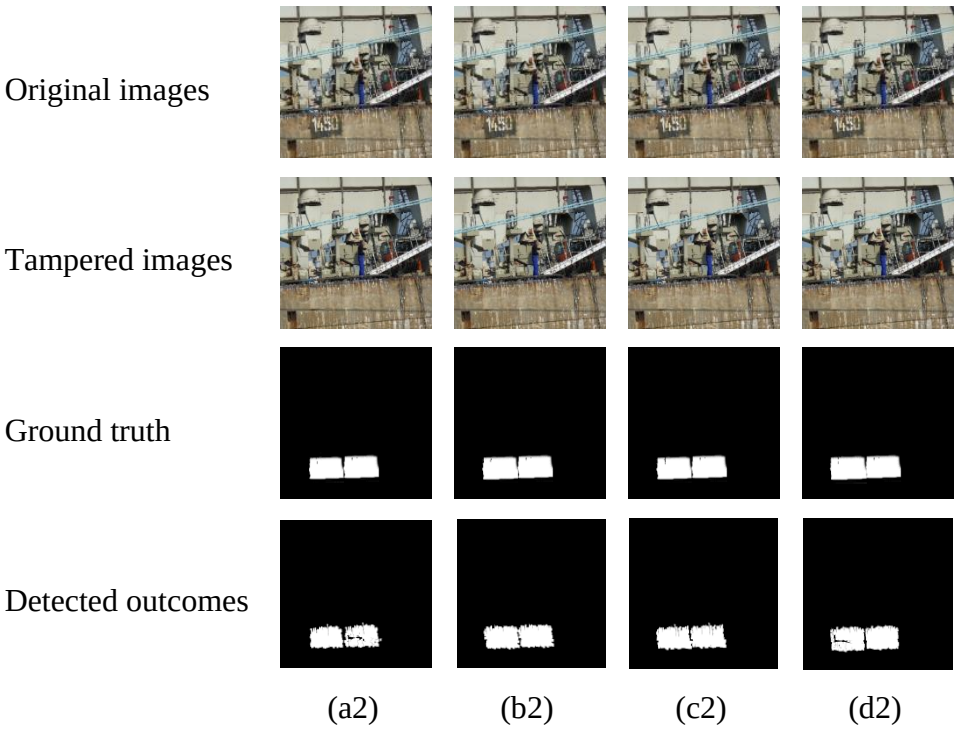


Figure 3.14: CMFD results of presented methodology for (a2) 97, (b2) 101, (c2) 99, and (d2) 103 scaling factors

(c) JPEG compression: The tampered photos are JPEG compressed using a quality factor (QF) between 20 and 100 in 10-step increments. The test is run on a total of $48 \times 9 = 432$ photos in this scenario. Figure 3.15 depicts the detection outcomes with varied QF of 40, 60, 80, and 90. The discovered tampered parts using the presented technique are shown in Figure 3.15's fourth row. Figure 3.15 shows that the suggested approach accurately identifies forgeries even under this attack.

(d) Noise addition: The cloned regions are augmented with zero mean Gaussian noise with STD ranging from 0.02 to 0.1 in 0.02 increments. A test is run on a total of $48 \times 5 = 240$ photos in this example. The noise addition detection results shown in Figure 3.16's fourth row demonstrate that the suggested technique identifies forgeries accurately even under noise addition attack.

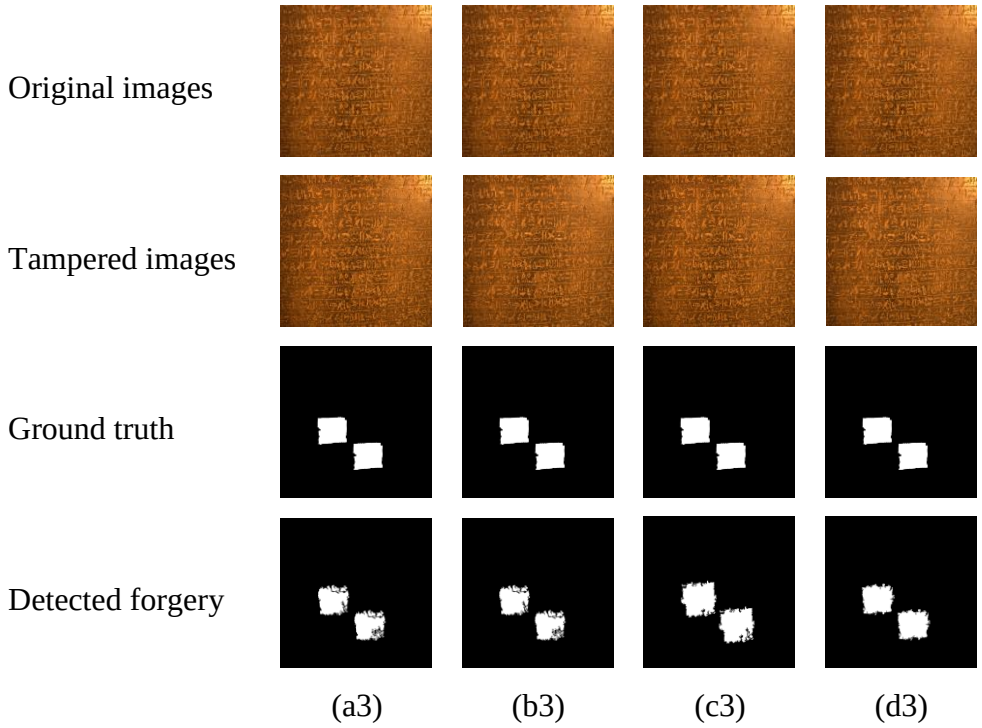


Figure 3.15: CMFD outcomes for (a3) 40, (b3) 80, (c3) 60, and (d3) 90 QF of JPEG compression for presented methodology

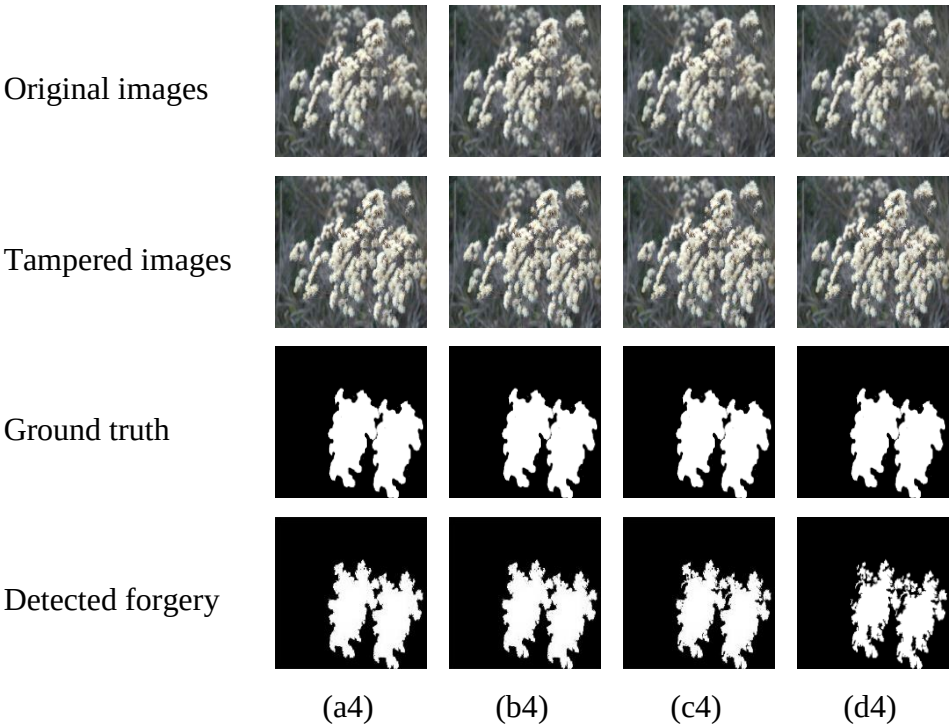
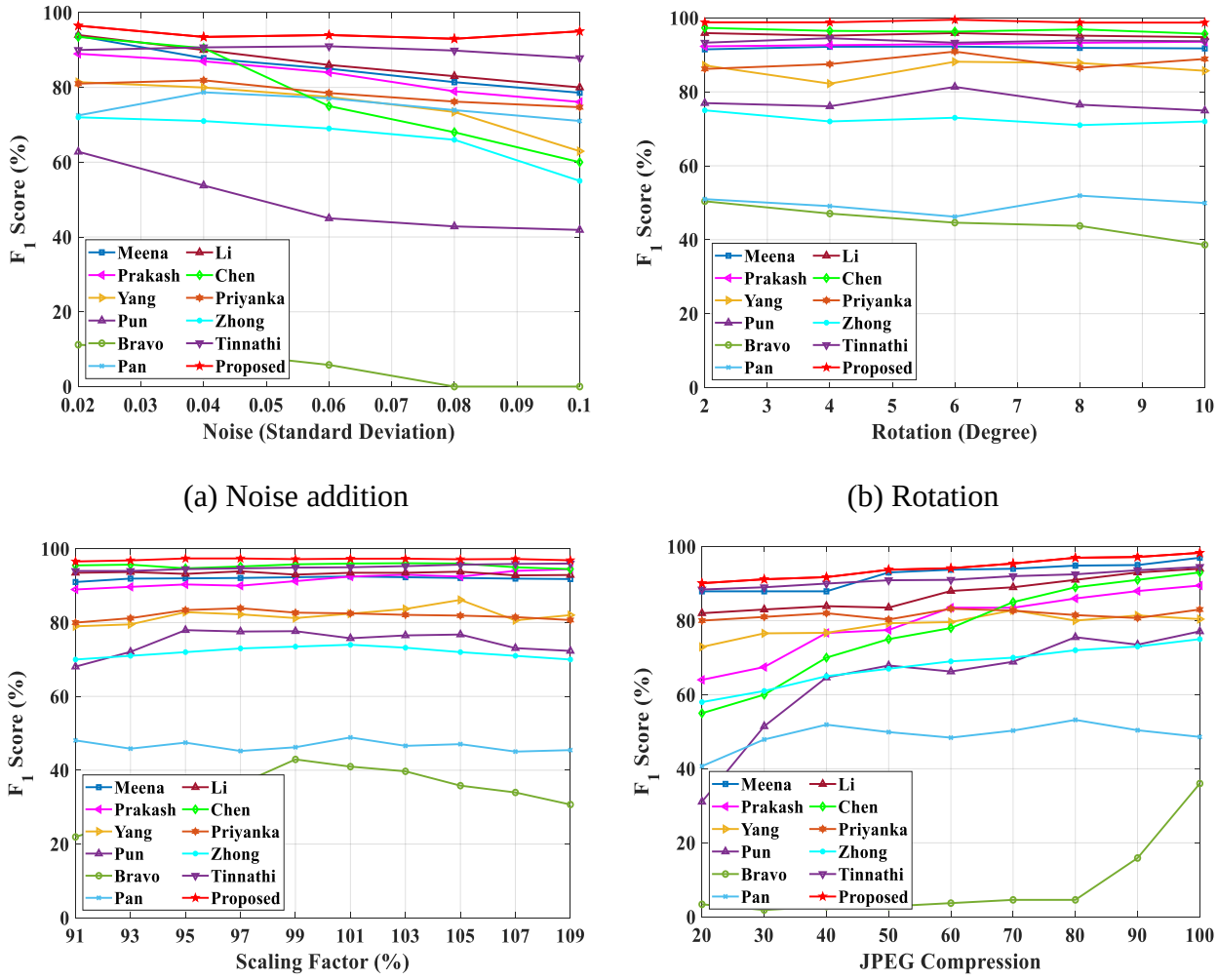


Figure 3.16: CMFD under noise addition attack with STD (a4) 0.02, (b4) 0.04, (c4) 0.08, and (d4) 0.1

From these detection outcomes, it is perceived that the suggested system can accurately recognize the presence of tampering against various attacks. In addition, the comparative

analysis of the F_1 score under various attacks is depicted in Figure 3.17. In the graphs below, the x-axis represents the rotation angle, JPEG compression's QF, scaling factor, and various noise's standard deviations. The F_1 score is shown on the y-axis. It has been discovered that the F_1 score surpasses other current approaches for various attacks. Similarly, the presented scheme may plot graphs for additional performance characteristics.



(a) Noise addition

(b) Rotation

(c) Scaling

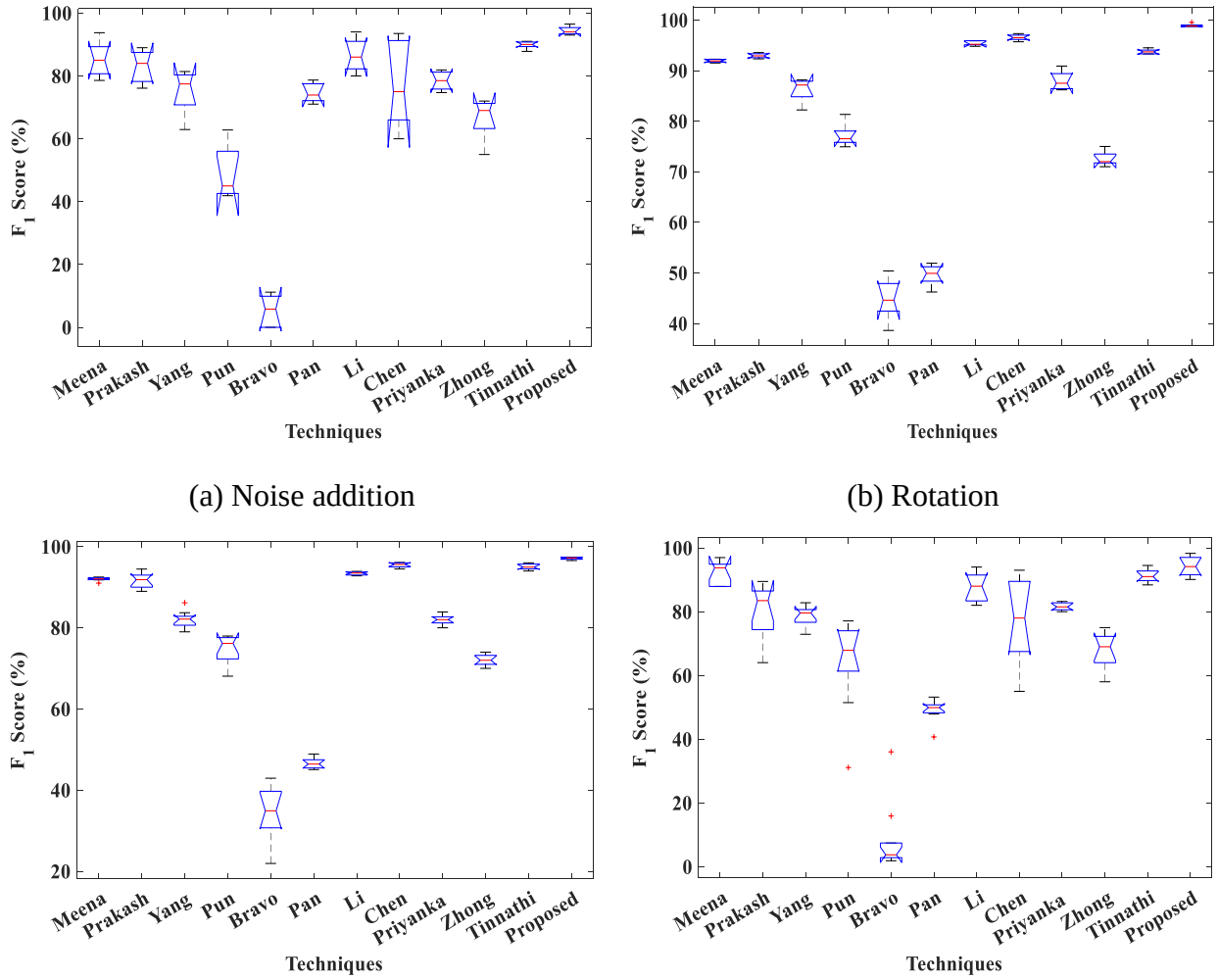
(d) JPEG compression

Figure 3.17: Comparison of F_1 score for various methodologies under various attacks

3.5 Statistical Analysis

In the previous subsection, it is discovered that the suggested approach outperforms the current strategies. However, the statistical significance of the detected variance in the performance of different methodologies must be confirmed, since it provides critical information in establishing the reliability of the data. To compare the methodologies, the ANOVA test is employed to determine if a statistically significant variance exists among the medians of

compared approaches. The ANOVA test is run on the outcomes of the presented and current methodologies under different attacks. Figure 3.18 represents the statistical study of the F_1 score for various strategies under various attacks.



(a) Noise addition

(b) Rotation

(c) Scaling

(d) JPEG compression

Figure 3.18: Statistical analysis of different procedures under various attacks

As described in Figure 3.18, ANOVA produces a box plot of F_1 score of suggested and current methodologies [2], [4], [5], [7], [16], [20], [21], [24], [26], [114], [124]. A box plot is used in the descriptive analysis to depict numerical data groups visually. Thus, the box plots in the present scheme give a visual depiction of the F_1 score for various methodologies under different attacks. The center point on each box represents the median, while the top and bottom margins of the box represent the 75^{th} and 25^{th} percentiles, correspondingly. The whiskers lengthen to the most extreme points, i.e., the highest and lowest non-outlier box plot values. The points lying outside the box plot's whiskers are called Outliers and are denoted by the '+'

sign. The whiskers (signifying box plot's lowest and highest values) of the suggested procedure reach about 100, which is greater than other current procedures, according to box plots in Figure 3.18. Box plots provide notches for comparing values of the median. The median values of two distinct procedures differ statistically substantial if the notches on two distinct box plots don't overlap. As a result, the two medians of compared procedures are considerably different at a 95% confidence level. Consequently, the data representation in Figure 3.18 shows that the current approaches are much weaker than the suggested methodology.

3.6 Cross-Dataset Performance

In this chapter, the final evaluation to depict flaws and ensure the efficiency of the presented method is cross-dataset performance. For successful practical uses, it is necessary to train a model with one dataset and then test it with another dataset obtained from separate sources, a process called cross-dataset performance. All datasets (involved in this study) are used to test cross-dataset performance. For example, the IMD dataset is utilized for training, while the MICC-F220 dataset is employed for testing. Further, the experiment is performed vice versa to assess the algorithm's performance, with MICC-F220 and IMD operating as the training set and testing set congruently [168]. Table 3.4 shows the performance of the cross-dataset for several performance characteristics such as F_1 score, recall, precision, and F_2 score. Table 3.4 shows that the presented method works well across datasets and has a wide range of performance metrics, indicating that the suggested approach has some generality to photos from various sources and sizes.

Table 3.4: Cross-dataset performance of the presented methodology

Training dataset	Testing dataset	Precision	F_1 score	Recall	F_2 score
IMD	GRIP	93.75	94.59	95.45	95.11
	MICC-F220	93.81	95.07	96.36	95.84
	COVERAGE	92.38	94.63	97.00	96.04
MICC-F220	GRIP	90.72	86.70	83.02	84.45
	COVERAGE	89.13	85.42	82.00	83.33
	IMD	83.93	90.38	97.92	94.76
COVERAGE	GRIP	95.48	95.69	95.91	95.82
	MICC-F220	92.98	94.64	92.98	95.67
	IMD	88.68	93.07	97.92	95.92

GRIP	MICC-F220	94.98	94.76	94.55	94.63
	COVERAGE	91.86	92.06	92.27	92.19
	IMD	93.61	93.39	93.18	93.27

3.7 Summary

In this chapter, an effective hybrid strategy is developed for the successful recognition of single and multiple CMFs employing block-based and keypoint-based methodologies. According to the finding of this study, the suggested method outperforms previous approaches via recall, F_1 score, precision, and F_2 score when using plain CMF. Furthermore, the suggested approach can accurately detect cloned parts from a tampered picture regardless of whether they are subjected to various attacks such as rotation, noise addition, scaling, and JPEG compression. Also, the statistical analysis utilizing ANOVA proves the efficiency of the suggested approach in contrast to the existing CMFD strategies. Performance across datasets is also assessed to verify the effectiveness of the suggested CMFD technique. The experimental findings demonstrate that the suggested approach yields outstanding detection results for CMF. Thus, it inspired to identify another type of forgery called splicing forgery in the next chapter.

CHAPTER 4

DETECTION OF IMAGE SPLICING FORGERY

This chapter focuses on detecting image splicing forgeries as it is one of the most commonly employed image-altering strategies. Markov attributes from DWT and LBP domains are retrieved and merged in this chapter to identify ISF. Discrete Haar Wavelet is used to employ three-level DWT to an input image. After employing one level of DWT, a picture is partitioned into low and high-frequency sub-bands. Furthermore, the low-frequency sub-band is decomposed twice to generate three-level DWT, yielding larger data and less noise. Six benchmark datasets are used to assess the efficacy of the suggested method. Furthermore, the SVM classifier is trained to distinguish between tampered and authentic images. The presented scheme's effectiveness is assessed using multiple performance measures which outperform other present methodologies.

4.1 Introduction

Image splicing is another common forgery type in which one segment of a picture is removed and injected into another picture to develop a new image. So, spliced pictures are being utilized for nefarious reasons because image splicing is simple, and detecting forged images with human eyes is difficult. Consequently, developing consistent splicing identification methods to regulate the authenticity of photographs has become a severe concern, encouraging scholars to create new strategies for detecting splicing forgery. Conventional techniques [56]–[58] utilized the fusion of DWT and LBP in domains such as image retrieval, recognition of objects, and facial expression. Furthermore, preceding work has detected ISF either by merging of DWT and LBP [59], [60] or a mixture of DWT and Markov [61], [62]; however, no application has yet employed the fusion of Markov, LBP, and DWT.

Consequently, Markov attributes from DWT and LBP domains are retrieved and merged in this chapter to identify ISF proficiently. As ISF generates sharp corners in a counterfeit photo, ISF detection (ISFD) relies on identifying the forgery-introduced artifacts. As the edges added by forgery vary from their surroundings, the correlations between the spliced area and the original area can be exploited to expose picture counterfeit. The suggested scheme describes these relationships using Markov TPM. Likewise, DWT is employed as wavelet analysis is efficient at catching the localized changes in pictures induced by splicing processes. Other

transforms, such as DCT and DFT, have lower time and frequency resolution than DWT. LBP, on either hand, is employed as it is a satisfactory completion operator that retains the slight differences in texture of counterfeit pictures since the actual texture of the image is deformed during manipulations. As a result, the suggested Markov-based technique for ISFD is effective.

4.2 Proposed Technique for Detecting ISF

After the LBP and DWT regions, the Markov procedure is used in this chapter. Following that, characteristics in both regions are integrated and normalized. The derived attributes are based on falsification influencing the interaction network amongst pixels. Consequently, characteristics from the DWT domain are retrieved and integrated with attributes from the LBP region. The Markov technique is used to illustrate statistical variations in both domains. The proposed algorithm's layout is depicted in Figure 4.1.

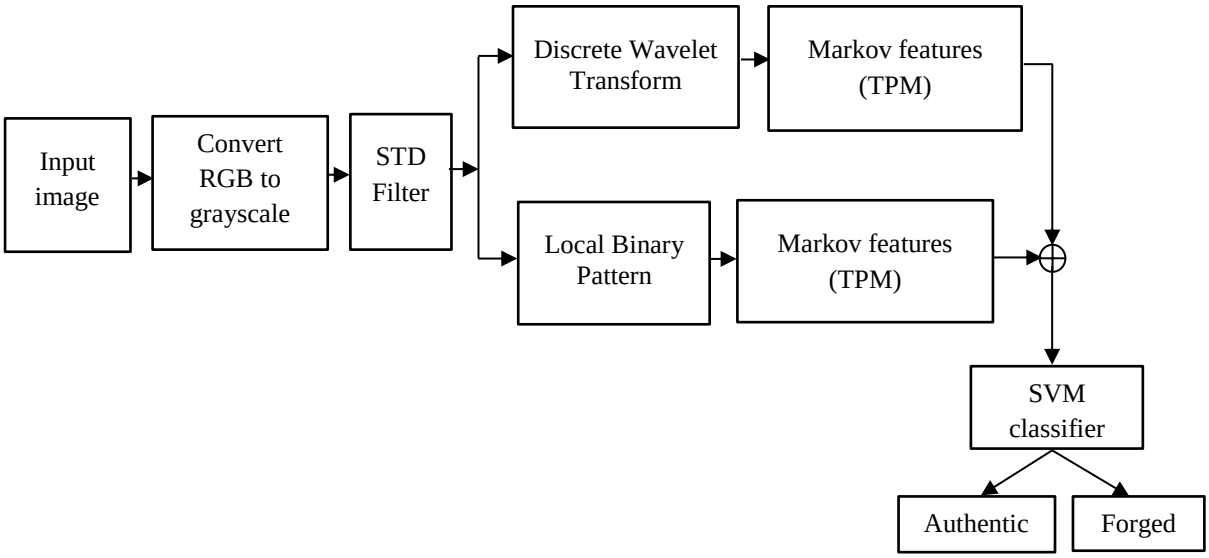


Figure 4.1: The architecture of proposed methodology

Before proceeding to the next phase, pre-processing activities are performed on photographs. The RGB picture Y is converted to a grayscale picture in this stage, as seen beneath [82]:

$$Y = 0.299R + 0.587G + 0.114B \tag{4.2.1}$$

where, G, R, and B are the image's green, red, and blue components, respectively. Then, an STD filter is applied to emphasize anomalies in the fabricated photos. The STD filter was chosen for its capacity to assess discrepancies in spliced pictures since its level of intensity fluctuates by a significant amount near the edge of a spliced picture [134], [169], [170]. Furthermore, the boundaries of the spliced component diverge from the boundaries of the rest

of the picture; therefore, their relationship can be utilized to disclose ISF. As the best indicator of diversity, the STD filter is chosen as it's proficient in spotting such relationships in the pictures. This filter replaces each pixel's value with the STD of its neighbors and the value of the filter itself. The STD filter emphasizes the area of the picture that is cut and pasted to create forgeries in spliced photos. As the STD filter is employed to remove solitary noisy spots in the picture, the features of the boundaries in the spliced picture are restored. In the scenario of edge detection, however, some filters are more prone to noise. In Figure 4.2, for illustration, a bird is extracted from one picture and put into another to create a counterfeit. Figure 4.2 shows that the STD filter detects the boundaries of the spliced component more effectively than the variance, skewness, and kurtosis filters. So, the suggested approach employs the STD filter to emphasize the sudden shifts in the spliced pictures.

(a) Original picture | (b) Tampered picture | (c) Grayscale picture | (d) STD filtered picture | (e) Variance filtered picture [169] | (f) Skewness filtered picture [169] | (g) Kurtosis filtered picture [169]

Figure 4.2: Illustration of various filters

Wavelet analysis may effectively identify localized signal changes. The Daubechies, Meyer, Haar, Symlets, and Coiflets are families of wavelets. The discrete Haar wavelet is used in this research because it is a memory-efficient, quick, and theoretically simple wavelet form. As per the findings, boundary details that comprise a large portion of the picture info are stored in these DWT sub-bands. An input picture is divided into four sub-bands in DWT's first level. Each sub-band has both high and low-frequency bands. Congruently, LH, HH, and HL signify high-frequency sub-bands (detailed coefficients) in vertical, horizontal, and diagonal directions, and LL exemplifies low-frequency sub-band (approximation coefficient).

The LL is further fragmented twice in third-level decomposition to reduce picture size and retrieve attributes. Furthermore, the breakdown of low-frequency sub-bands results in more data and less interference. Because of its ability to expand the picture in multiple resolutions at various points, DWT is employed to determine ISF. Other full-frame transformations include DCT and Discrete Fourier Transform (DFT). Any change in the parameters of both transformations will influence the entire picture. However, DWT has spatial frequency

localization, i.e., the picture will be altered locally in case of an embedded signal. As a result, a wavelet transform describes a picture in spatial and frequency terms [62], [77].

LBP is a powerful texture generator that detects local abnormalities in the regularity of transformed pictures. In this approach, each pixel is designated by the relative grey levels of its neighbors. If the grey level of the adjacent pixel is more or equivalent to the grey level of the central pixel, the pixel's value is one; otherwise, the pixel's value is zero. Finally, for each central pixel, the binary pattern is obtained. LBP code is referred to as the summation of pattern bits [129], [171]. The formula for assessing the LBP operator is:

$$L(xx, yy) = \sum_{q=0}^{q-1} (h_q - h(xx, yy))2^q \tag{4.2.2}$$

where $'q'$ is total pixels in the neighborhood of radius R, $h(xx, yy)$ is number of central pixel at (xx, yy), h_q is q^{th} pixel in zone and $Z(h_q - h(xx, yy))$ is threshold function.

$$Z(h_q - h(xx, yy)) = \begin{cases} 1 & (h_q - h(xx, yy)) \geq 0 \\ 0 & (h_q - h(xx, yy)) > 0 \end{cases} \tag{4.2.3}$$

When a picture is counterfeit, the original quality of the picture is misrepresented. Because the LBP is skilled at detecting texture variations, it is employed in the suggested method to detect counterfeit and legitimate photos. The artifacts formed at the image's boundaries by the altering method are the traits that distinguish fabrication. The relationship between nearby pixels is recorded for this purpose by measuring the transformations in minor diagonal (M), vertical (V), main diagonal (D), and horizontal (H) directions for DWT and LBP factors [172]. The difference arrays $L_z(xx, yy), z \in \{M, V, D, H\}$ for LBP are computed by:

$$L_H(xx, yy) = L(xx, yy) - L(xx+1, yy) \tag{4.2.4}$$

$$L_v(xx, yy) = L(xx, yy) - L(xx, yy+1) \tag{4.2.5}$$

$$L_D(xx, yy) = L(xx, yy) - L(xx+1, yy+1) \tag{4.2.6}$$

$$L_M(xx, yy) = L(xx+1, yy) - L(xx, yy+1) \tag{4.2.7}$$

where, $L(xx, yy)$ is considered LBP code, $1 \leq xx \leq R_{xx}, 1 \leq yy \leq R_{yy}$, $R_{xx} \times R_{yy}$ is image's dimensions. The contrasts between DWT-based Markov attributes are assessed in all four dimensions in the same manner that LBP is. In the above-mentioned equations, $L(xx, yy)$ is substituted by $W(xx, yy)$ to obtain $W_z(xx, yy), z \in \{V, H, D, M\}$ for DWT.

The Markov system is an essential technique for extracting features since it identifies the link between the characteristics. As mentioned in the theories of random processes, the Markov TPM is a framework used to characterize the links between the spliced part and the original part of the forged picture. As a result, a Markov-based attribute is a measure that can reflect statistical changes caused by splicing. Since picture splicing results in sharp edges in a counterfeit picture, preserving the forgery-caused artifacts is critical for ISFD. So the edges generated by counterfeit vary from their "neighbors," and the links between the spliced and original parts can be utilized to detect image counterfeit. As a result, Markov is useful in picture splicing challenges.

The STD filter is first applied to the input picture, revealing tampering artifacts' abnormalities. As a result, Markov TPM is used to detect counterfeit regions in photos by thoroughly investigating the abnormalities of tampering artifacts. Since employing the Markov system to a difference array reduces the dimensionality of the Markov TPM, the Markov system is applied to difference arrays rather than directly on the picture or coefficients 2-D array. TPMs derived from difference arrays in both LBP and DWT areas monitor pixels or coefficient relations to identify splicing artifacts. Figure 4.3 depicts the general block structure of Markov feature extraction, whereas Figure 4.4 depicts the difference arrays for the major diagonal, horizontal, vertical, and minor diagonal.

The difference arrays of both DWT and LBP regions are restricted to $[-T,+T]$. If $L(xx,yy)$ or $W(xx,yy)$ is lesser than $-T$ or larger than T, it is indicated by $-T$ or T, congruently as given in the following equation [65].

$$Z_z(xx,yy)=\begin{cases} T & F_z(xx,yy)>+T \\ -T & F_z(xx,yy)<-T \\ F_z(xx,yy) & otherwise \end{cases} \tag{4.2.8}$$

where $F_z(xx,yy)$ is either $L_z(xx,yy)$ or $W_z(xx,yy)$, for $z\in\{M,V,D,H\}$. The T constrains the total states that the statistics required. The number T is chosen at 4 in the suggested approach to keep a balance between computing efficacy and classifier ability. The Markov system is classified by TPM [172]. For one-step TPM, the total number of components in every direction is $(2T+1)\times(2T+1)$. The following equations indicate the attained TPM for M,H,V and D directions:

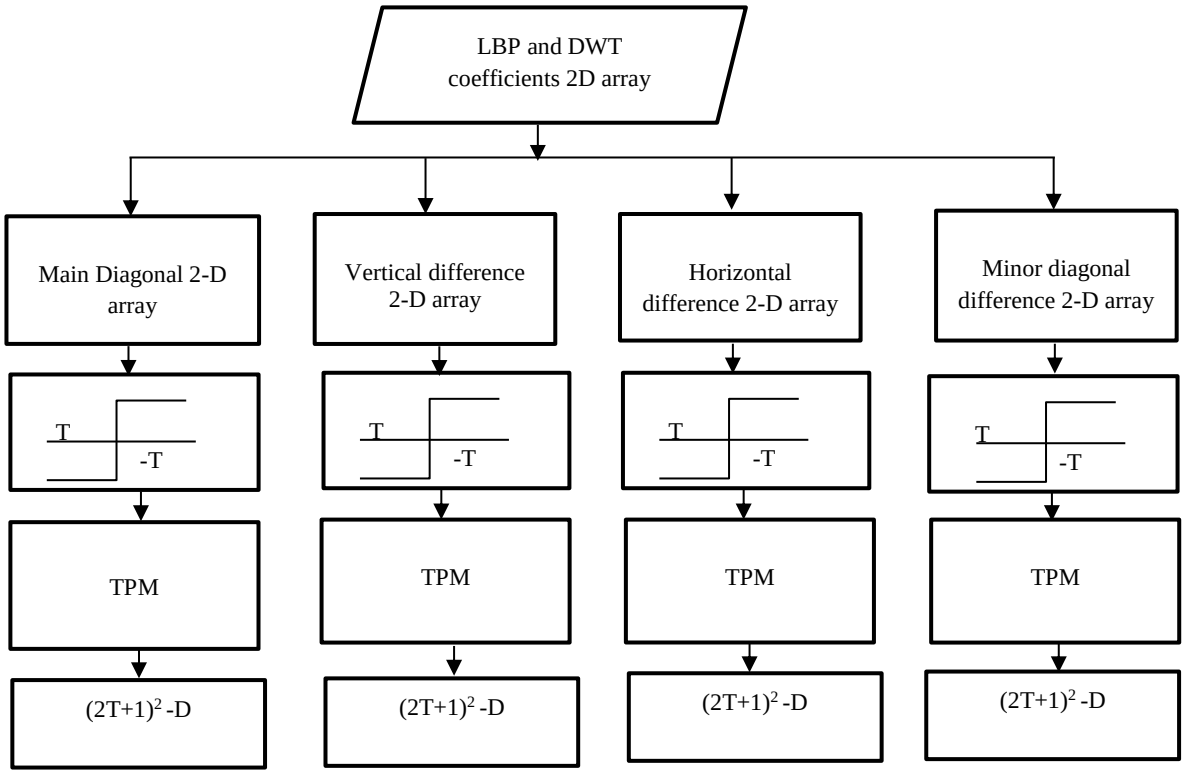


Figure 4.3: Block diagram of Markov feature extraction procedure

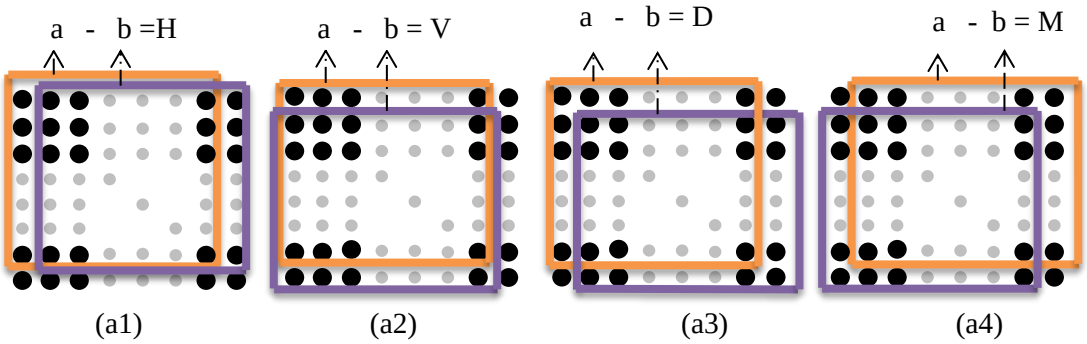


Figure 4.4: Difference 2-D array: (a1) Horizontal (H), (a2) Vertical (V), (a3) Main diagonal (D), and (a4) Minor diagonal (M)

$$P[Z_h(xx+1,yy)=qq \mid Z_h(xx,yy=pp)]=\frac{\sum_{xx=1}^{R_{xx}-1}\sum_{yy=1}^{R_{yy}}\delta(Z_h(xx,yy)=pp, Z_h(xx+1,yy)=qq)}{\sum_{xx=1}^{R_{xx}-1}\sum_{yy=1}^{R_{yy}}\delta(Z_h(xx,yy)=pp)} \quad (4.2.9)$$

$$P[Z_v(xx,yy+1)=qq \mid Z_v(xx,yy=pp)]=\frac{\sum_{xx=1}^{R_{xx}}\sum_{yy=1}^{R_{yy}-1}\delta(Z_v(xx,yy)=pp, Z_h(xx,yy+1)=qq)}{\sum_{xx=1}^{R_{xx}}\sum_{yy=1}^{R_{yy}-1}\delta(Z_h(xx,yy)=pp)} \quad (4.2.10)$$

$$P[Z_d(xx+1, yy+1) = qq \mid Z_d(xx, yy = pp)] = \frac{\sum_{xx=1}^{R_{xx}-1} \sum_{yy=1}^{R_{yy}-1} \delta(Z_d(xx, yy) = pp, Z_d(xx+1, yy+1) = qq)}{\sum_{xx=1}^{R_{xx}-1} \sum_{yy=1}^{R_{yy}-1} \delta(Z_d(xx, yy) = pp)} \quad (4.2.11)$$

$$P[Z_m(xx, yy+1) = qq \mid Z_m(xx+1, yy = pp)] = \frac{\sum_{xx=1}^{R_{xx}-1} \sum_{yy=1}^{R_{yy}-1} \delta(Z_m(xx+1, yy) = pp, Z_m(xx, yy+1) = qq)}{\sum_{xx=1}^{R_{xx}-1} \sum_{yy=1}^{R_{yy}-1} \delta(Z_m(xx+1, yy) = pp)} \quad (4.2.12)$$

where, $pp, qq \in \{-T, -T+1, ..., 0, ..., T-1, T\}$, $R_{xx} \times R_{yy}$ is the image's dimensionality [172]. If the criteria is satisfied $\delta(\cdot) = 1$, otherwise $\delta(\cdot) = 0$ as stated in the equation below:

$$\delta(A = pp, B = qq) = \begin{cases} 1 & A = pp, B = qq \\ 0 & otherwise \end{cases} \quad (4.2.13)$$

The feature vector is obtained after assessing the Markov in both areas, namely DWT and LBP, at $T = 4$, which is applied to the SVM classifier to classify images. Furthermore, the computational time is minimized, and detection performance is enhanced. The following is the suggested technique's pseudocode:

Pseudocode for the proposed scheme

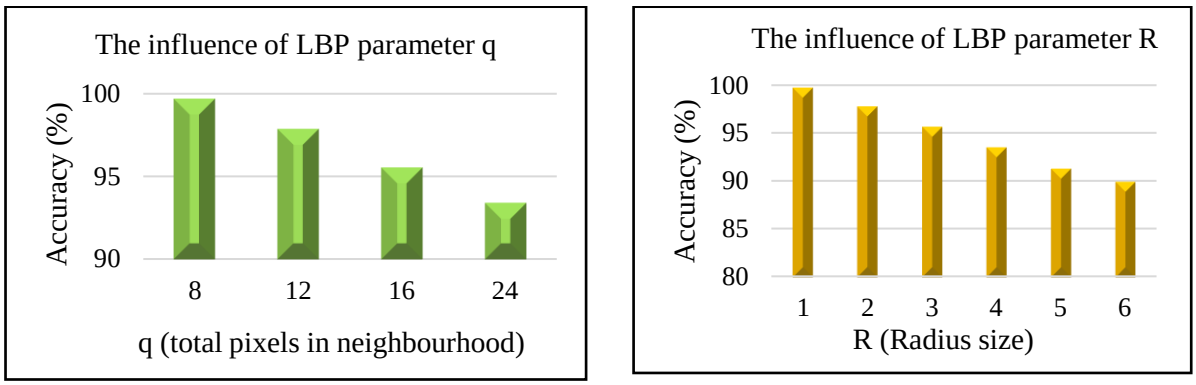
Input- Image (Authentic/Forged)
Output- To recognize a picture as authentic or forged

procedure
Input image
if input image is colored
Convert into a grayscale
else
Go to the next step
end if
Apply STD filter to the input image
/* Compute LBP Markov-based features in all four directions: minor diagonal (M), diagonal (D), horizontal (H), vertical (V) */
Compute LBP of the obtained image
for $z \in \{V, H, D, M\}$
Calculate the difference matrix L_Z
Apply thresholding
Compute TPM_Z^{LBP}
end for
/* Compute DWT Markov-based features in all four directions: H, V, D, and M*/
Compute DWT of the obtained image
for $z \in \{V, H, D, M\}$

Calculate difference matrix w_Z
Apply thresholding
Compute TPM_Z^{DWT}
end for
Combine all TPM_Z^{LBP} and TPM_Z^{DWT} into a feature vector
Apply SVM to classify (Authentic/Forged)
end procedure

4.3 Experimental Results and Discussions

All experimentation is conducted in this section to assess the efficacy of the suggested algorithm. Standard datasets such as DVMM, IFS-TC, CASIA v2.0, Columbia, CASIA v1.0, and DSO-1 are employed in the experimental assessment. Recall, F_1 score, TPR, F_2 score, precision, TNR, informedness, accuracy, markedness, and MCC are used to measure the procedure's efficacy. This study also includes a short explanation of how factors are selected in the trials of the suggested method. The suggested technique includes several factors like LBP factors and Markov's threshold value, i.e., T. Various trials are carried out on CASIA v1.0 with various LBP factors (q, R) to determine the set that resulted in the finest performance; here, q is total pixels in the neighborhood of radius R [138], [143]. Based on the experimentations, Figure 4.5 shows that the LBP factors $q = 8, R = 1$ provide the optimum value with higher accuracy. As a result, the succeeding tests are carried out using these ideal LBP parameter settings.



(a1) (a2)

Figure 4.5: The influence of LBP factors (a1) q (a2) R on efficiency

Furthermore, a few considerations should be considered when choosing optimal values for the threshold. If a lower threshold value is set, it is challenging to catch tampering artifacts. On the other side, if the threshold value is too high, the dimension of feature vectors will be quite high, and the computational cost may become unmanageable. As a result, selecting T resulted in a

trade-off between the accuracy rate and the algorithm's processing cost. Several works [63], [66], [67], [140] that use the Markov attributes set the threshold value to 4; thus, $T = 4$ is chosen in our experimentation.

4.3.1 Simulation Results of ISF

As previously stated, the suggested approach is evaluated using six datasets. In virtually all datasets, the total tampered photographs exceed the number of real images; thus, a proportion between original and forged pictures is retained. Thus, photos are picked randomly so that an equal proportion of both pictures are used for detection. The photos are labeled with the corresponding label, which is then employed to train the classifier. On the other hand, the photos employed for testing have no label and are employed to validate the system's efficiency. 80% of photographs are utilized for training and 20% for testing to assess the efficacy of the suggested strategy. A confusion matrix summaries the classifier's effectiveness on test data. In CASIA v1.0, there are 800 original and 921 tampered pictures. To preserve balance, 800 original and 800 fabricated photographs are used for testing, yielding a total of 1600 images for CASIA v1.0. The classifier is trained using 1280 photos and tested with 320 images in an 80:20 ratio. As a result, the confusion matrix is constructed using 320 pictures to show the classifier's accuracy by comparing real and predicted classes. Table 4.1, from left to right, shows the confusion matrix of testing pictures of all datasets employed in this study. Table 4.2 displays the values of several performance metrics for all datasets, whereas Figure 4.6 illustrates its graphical depiction.

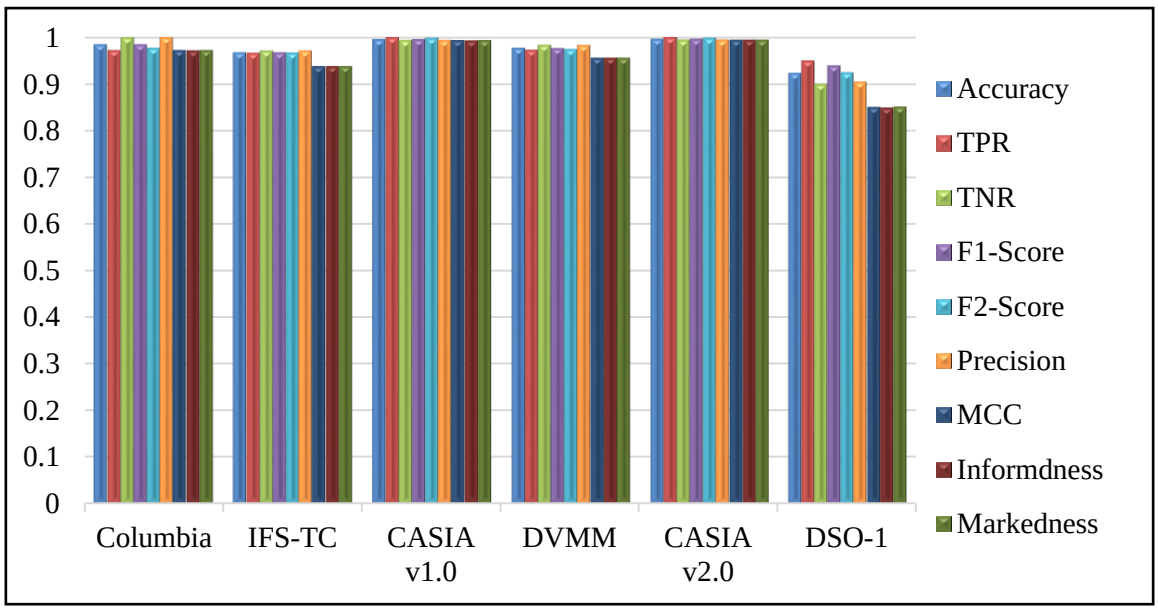


Figure 4.6: Graphical illustration of performance metrics on various datasets

Table 4.1: Various dataset's confusion matrices

CASIA v1.0	Predicted Negative	Predicted Positive
Actual Negative	159	1
Actual Positive	0	160

DSO-1	Predicted Negative	Predicted Positive
Actual Negative	18	2
Actual Positive	1	19

Columbia	Predicted Negative	Predicted Positive
Actual Negative	36	0
Actual Positive	1	35

CASIA v2.0	Predicted Negative	Predicted Positive
Actual Negative	1022	3
Actual Positive	2	1023

IFS-TC	Predicted Negative	Predicted Positive
Actual Negative	204	6
Actual Positive	7	203

DVMM	Predicted Negative	Predicted Positive
Actual Negative	179	3
Actual Positive	5	177

Several tests are conducted on the six datasets listed. Furthermore, the fused Markov characteristics of DWT and LBP are compared to the DWT and LBP Markov properties independently on the specific datasets, as shown in Table 4.3.

Table 4.2: Performance metrics (%) of presented method on several datasets

Metrics	Columbia	DSO-1	CASIA v1.0	DVMM	IFS-TC	CASIA v2.0
Accuracy	98.61	92.50	99.69	97.80	96.90	99.76
Precision	100	90.48	99.38	98.33	97.13	99.71
MCC	97.26	85.11	99.38	95.61	93.81	99.51
TNR	100	90.00	99.38	98.35	97.14	99.71
F_1 score	98.59	94.06	99.69	97.79	96.90	99.76
Markedness	97.30	85.21	99.38	95.62	93.81	99.51
TPR	97.22	95.00	100	97.25	96.67	99.80
Informedness	97.22	85.00	99.37	95.60	93.81	99.51
F_2 score	97.77	92.47	99.88	97.49	96.76	99.79

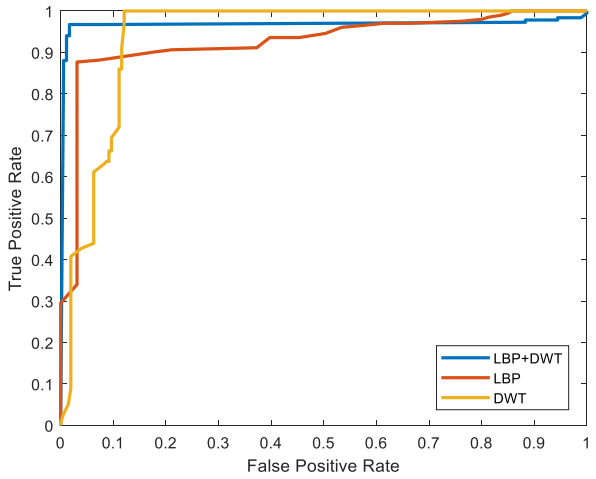
Table 4.3: Results for datasets with different features

Dataset	Features	Accuracy	F_1 score	TNR	Precision	Informedness	F_2 score	TPR	Markedness	MCC
CASIA v1.0	LBP	86.25	86.08	87.50	87.18	72.50	85.43	85.00	72.55	72.52
	DWT	77.81	81.75	56.25	69.43	55.62	91.48	99.38	68.33	61.65
	LBP+DWT	**99.69**	**99.69**	**99.38**	**99.38**	**99.37**	**99.88**	**100**	**99.38**	**99.38**
CASIA v2.0	LBP	78.49	79.65	72.78	75.57	56.98	82.32	84.20	57.73	57.35
	DWT	86.83	88.34	73.76	**99.90**	73.66	94.94	**99.90**	79.06	76.31
	LBP+DWT	**99.76**	**99.76**	**99.71**	99.71	**99.51**	**99.79**	99.80	**99.51**	**99.51**
Columbia	LBP	88.89	88.57	91.67	91.18	77.78	87.08	86.11	78.02	77.90
	DWT	81.94	82.67	77.78	79.49	63.89	84.70	86.11	64.34	64.11
	LBP+DWT	**98.61**	**98.59**	**100**	**100**	**97.22**	**97.77**	**97.22**	**97.30**	**97.26**
DVMM	LBP	93.65	93.26	**98.90**	**98.76**	87.23	90.24	88.33	88.31	87.77
	DWT	93.13	93.33	90.11	90.67	86.26	95.01	96.15	86.58	86.42
	LBP+DWT	**97.80**	**97.79**	98.35	98.33	**95.60**	**97.47**	**97.25**	**95.62**	**95.61**
IFS-TC	LBP	73.57	70.71	83.33	79.29	47.14	66.40	63.81	49.01	48.07
	DWT	72.62	78.26	46.67	64.89	45.24	89.30	**98.57**	61.92	52.93
	LBP+DWT	**96.90**	**96.90**	**97.14**	**97.13**	**93.81**	**96.76**	96.67	**93.81**	**93.81**
DSO-1	LBP	85.00	85.00	85.00	85.00	70.00	85.00	85.00	70.00	70.00
	DWT	82.50	84.44	70.00	76.00	65.00	90.48	95.00	69.33	67.13
	LBP+DWT	**92.50**	**94.06**	**90.00**	**90.48**	**85.00**	**92.47**	**95.00**	**85.21**	**85.11**

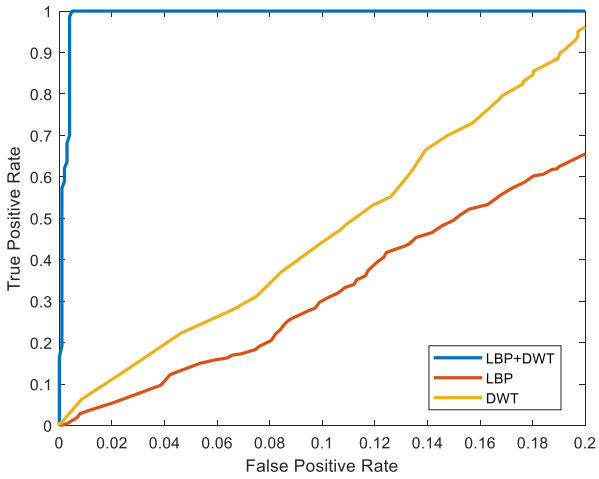
Bold indicates the maximum value

Table 4.3 shows that when Markov attributes from both areas, DWT and LBP, are retrieved and integrated, noteworthy results are obtained. According to the findings in Table 4.3, for the DVMM dataset, Markov characteristics in the LBP region outperform Markov characteristics in the DWT region. Further, combining both regions yields better outcomes than each region independently. In terms of accuracy and specificity, LBP-based Markov attributes surpass DWT-based Markov attributes for the Columbia and CASIA v1.0 datasets. When DWT and LBP are combined, detection performance improves. Although DVMM, CASIA v1.0, and Columbia datasets are widely utilized, their sizes are not enormous. So, the same strategy is used for CASIA v2.0 to validate the efficacy of the suggested methodology on a bigger dataset. Markov features in DWT outperform Markov features in LBP in this evaluation. However, integrating characteristics from both domains results in the best performance. In addition, two more datasets, IFS-TC and DSO-1, have been employed in the experiment. These datasets are manipulated by copying and pasting varying degrees of photorealism. DSO-1 is employed to test the results on a relatively small dataset of 100 real and 100 counterfeit photos. For this dataset, LBP-based features have superior accuracy to DWT features. Furthermore, when compared to individual characteristics, combining both characteristics outperforms the accuracy rate.

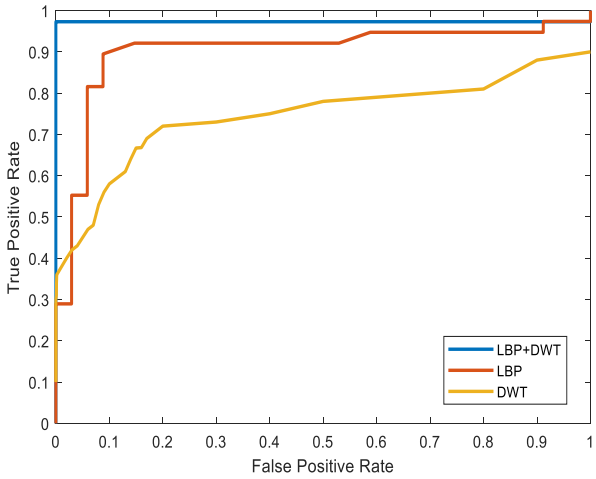
Figure 4.7 depicts the ROC curves for LBP, DWT, and integrated Markov attributes on several datasets. The ROC curves for CASIA v1.0 and CASIA v2.0 datasets have been zoomed in for easier viewing. The ROC curves for merged characteristics are nearer the top left corner in Figure 4.7, indicating the maximum accuracy. The classifier's efficacy is determined by employing both DWT and LBP-based Markov attributes independently and in combination. Furthermore, as shown in Table 4.4, the AUC of the proposed model is compared to current approaches on all datasets. The greatest AUC value suggests that the proposed model is appropriate for classifying images as authentic or forged by combining LBP and DWT regions. Table 4.3, Table 4.4 and the ROC curves for the corresponding datasets indicate that combining Markov characteristics from the LBP and DWT domains leads to better outcomes than each domain individually. The accuracy attained by merging LBP and DWT regions for datasets such as DVMM, CASIA v2.0, IFS-TC, Columbia, DSO-1, and CASIA v1.0 is 97.80%, 99.76%, 96.90%, 98.61%, 92.50%, and 99.69%, respectively.



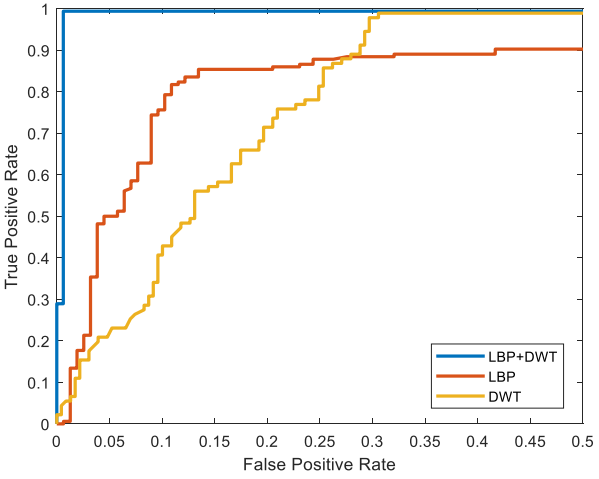
(a) DVMM



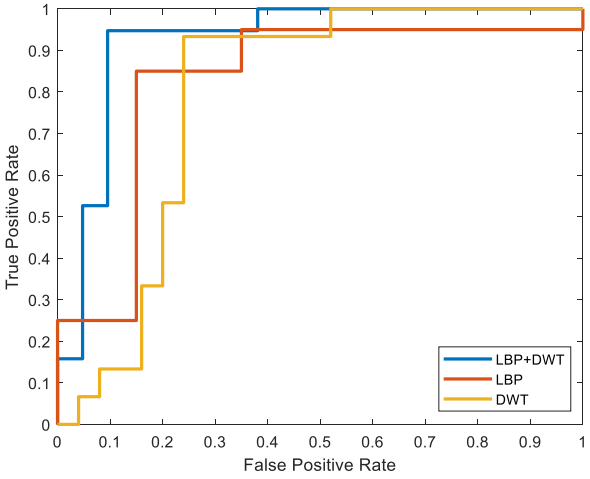
(b) CASIA v2.0



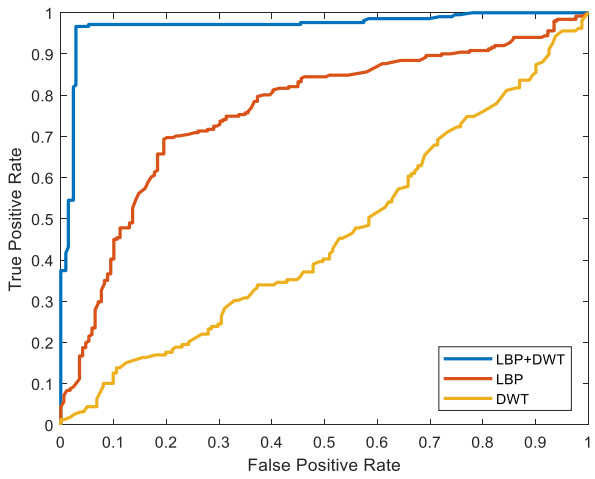
(c) Columbia



(d) CASIA v1.0



(e) DSO-1



(f) IFS-TC

Figure 4.7: ROC curves for different features tested on different datasets

Table 4.4: Comparison of AUC for different features tested on different datasets

	CASIA v1.0	CASIA v2.0	Columbia	DVMM	IFS-TC	DSO-1
LBP	87.26	78.65	88.46	93.67	73.67	85.39
DWT	77.37	86.23	81.67	93.13	72.84	82.95
LBP+DWT	**99.08**	**99.78**	**98.61**	**97.56**	**96.45**	**92.87**

Bold indicates the maximum value

4.3.2 Comparative Analysis

Table 4.5 provides a comparative analysis of performance metrics of the proposed system with several existing ISFD approaches to demonstrate the effectiveness of the suggested scheme. The ROC curve is shown to demonstrate the classifier's efficacy. The ROC curve towards the upper left corner represents the suggested scheme's best performance. Figure 4.8 provides the ROC curve's comparison of the suggested model and current techniques for all datasets. For better visualization, the ROC curves for the CASIA v2.0 and CASIA v1.0 datasets have been magnified.

Table 4.5: Performance metrics (%) of the proposed methodology compared to current methodologies

Dataset	Methodologies	Accuracy	TNR	Informedness	TPR
CASIA v1.0	Alahmadi [138]	97.50	98.24	94.99	96.75
	Muhammad [68]	94.89	93.91	89.06	95.15
	Hussain [142]	94.29	NC	NC	NC
	Sheng [140]	98.77	NC	NC	NC
	Hakimi [60]	97.21	NC	NC	NC
	Kanwal [27]	98.25	NC	NC	NC
	Proposed	**99.69**	**99.38**	**99.37**	**100**
CASIA v2.0	Alahmadi [138]	97.50	96.88	95.19	98.31
	Muhammad [68]	97.33	96.53	95.03	98.50
	He [67]	89.76	NC	NC	NC
	Prakash [70]	96.68	97.52	93.29	95.77
	Sheng [140]	97.59	NC	NC	NC

Dataset	Methodologies	Accuracy	TNR	Informedness	TPR
CASIA v2.0	Jalab [139]	99.50	99.00	94.00	95.00
	Kanwal [27]	97.59	NC	NC	NC
	Proposed	**99.76**	**99.71**	**99.51**	**99.80**
DVMM	He [67]	93.55	93.83	87.11	93.28
	Zhang [61]	89.88	87.31	79.81	92.50
	El-Alfy [65]	96.83	96.84	93.67	96.83
	Dong [132]	84.36	85.53	68.76	83.23
	Zhao [64]	93.36	93.75	86.74	92.99
	Shi [130]	90.15	90.31	80.32	90.01
	Kumar [173]	88.43	NC	NC	NC
	Proposed	**97.80**	**98.35**	**95.60**	**97.25**
Columbia	Agarwal [134]	93.81	NC	NC	NC
	Wei [145]	NC	NC	NC	09.90
	Korus [144]	NC	NC	NC	75.4
	Alahmadi [143]	96.60	NC	NC	NC
	Lyu [146]	NC	NC	NC	00.70
	Xiao [147]	NC	NC	NC	61.20
	Kanwal [27]	96.66	NC	NC	NC
	Huh [148]	NC	NC	NC	84.30
	Proposed	**98.61**	**100**	**97.22**	**97.22**
IFS-TC	He [67]	91.87	89.02	84.32	95.30
	Zhang [63]	84.53	84.57	69.06	84.49
	Zhang [61]	92.10	89.14	84.80	95.66
	Li [66]	89.61	88.14	79.52	91.38
	Proposed	**96.90**	**97.14**	**93.81**	**96.67**
DSO-1	Agarwal [134]	85.31	NC	NC	NC
	Agarwal [135]	88.33	**91.44**	78.23	86.79
	Proposed	**92.50**	90.00	**85.00**	**95.00**

NC: Not calculated by the respective author and Bold indicates the maximum value

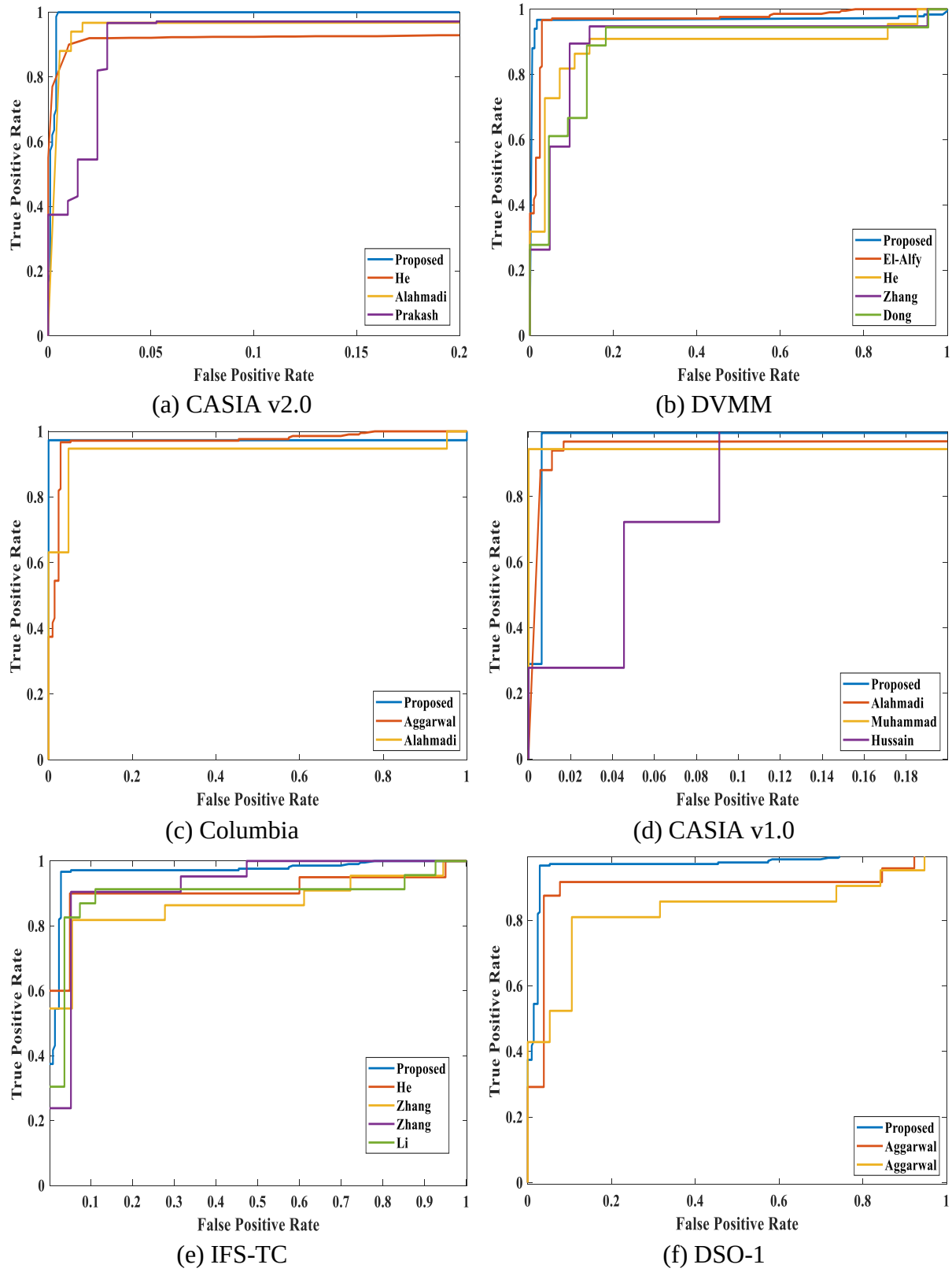


Figure 4.8: ROC curve comparison for the proposed technique assessed on several datasets

According to Table 4.5, the suggested system surpasses existing strategies in terms of performance measures such as specificity, accuracy, informedness, and sensitivity. As seen in

Figure 4.8, the suggested scheme's ROC curve for various datasets is nearer to the top-left corner, indicating that it achieves a higher accuracy rate than previous approaches. Moreover, the AUC of proposed approach and existing approaches is compared in Table 4.6.

Table 4.6: Comparison of presented approach's AUC with existing approaches

Dataset	Approaches	AUC
CASIA v1.0	Alahmadi [138]	97.20
	Muhammad [68]	96.89
	Hussain [142]	94.29
	Proposed approach	**99.46**
CASIA v2.0	Alahmadi [138]	97.56
	He [67]	89.66
	Prakash [70]	96.78
	Proposed approach	**99.62**
DVMM	He [67]	92.55
	Zhang [61]	91.88
	El-Alfy [65]	96.89
	Dong [132]	89.36
	Proposed approach	**97.29**
Columbia	Agarwal [134]	94.81
	Alahmadi [143]	95.60
	Proposed approach	**98.45**
IFS-TC	He [67]	91.98
	Zhang [63]	85.43
	Zhang [61]	92.87
	Li [66]	89.54
	Proposed approach	**96.85**
DSO-1	Agarwal [134]	85.42
	Agarwal [135]	88.98
	Proposed approach	**92.26**

Bold indicates the maximum value

The experimental findings indicate that combining Markov TPM attributes in DWT and LBP regions surpasses in terms of specificity, sensitivity, informedness, and accuracy. Furthermore,

compared to existing algorithms, the suggested scheme achieves good detection performance for all datasets.

4.3.3 Run-Time Analysis

The run time analysis of the suggested technique for detecting splicing forgeries on all six datasets is reviewed in this section. The suggested approach's average running time is shown in Table 4.7. The running time varies depending on the size of the photos and the total photos in the dataset. The suggested technique achieved 0.372, 0.110, 2.748, 2.478, 0.508, and 4.482 seconds per picture for IFS-TC, DSO-1, CASIA v1.0, Columbia, CASIA v2.0, and DVMM, respectively. The DVMM dataset has the shortest average run time of the five datasets since each image is just $128_{\times}128$ pixels in size. However, because the IFS-TC dataset contains large-sized pictures, the suggested approach requires the most processing time. The suggested technique outperforms the other four datasets on the smaller dataset, i.e., DSO-1. Meanwhile, the CASIA v2.0 and CASIA v1.0 datasets run marginally faster than the others. Table 4.7 shows that the suggested technique's average run time grows dramatically with the total number of photos or the dimension of photos.

Table 4.7: Analysis of running time on several datasets

Dataset	Total pictures	Picture Size	Running Time (sec/picture)
DVMM	1845	128×128	0.110
Columbia	363	757×568 to 1152×768	2.478
CASIA v2.0	12,614	240×160 to 900×600	0.508
DSO-1	200	2048×1536	2.748
CASIA v1.0	1721	384×256	0.372
IFS-TC	2200	1024×768 to 2848×2144	4.482

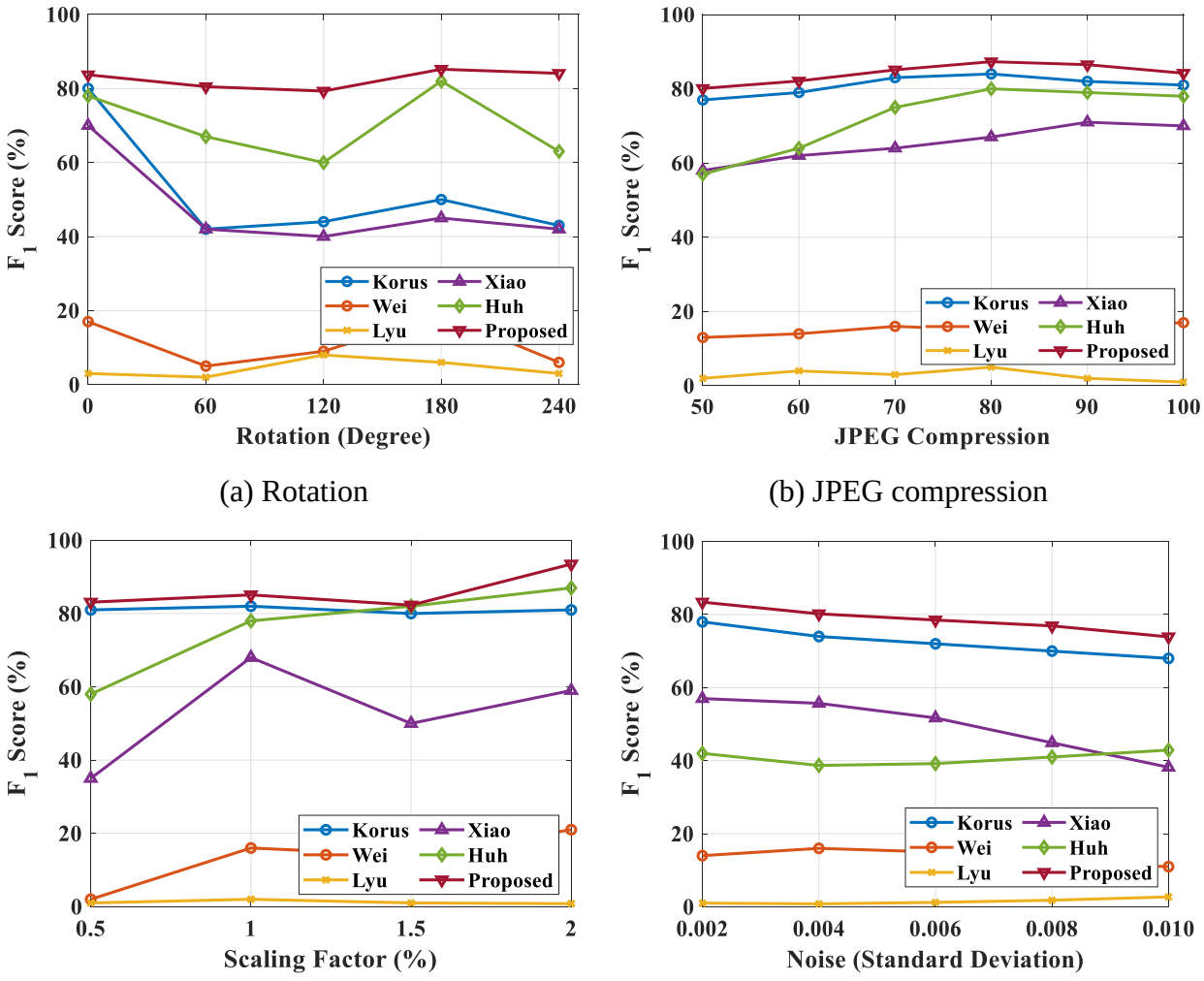
4.4 Robustness Under Various Attacks

The following attacks are employed on the Columbia dataset in this section to assess the reliability of the suggested methodology:

- **Noise Addition:** In splicing counterfeit images, white Gaussian noise with a mean value of zero and different variances is introduced.

- **Scaling:** The counterfeit photos that have been spliced are scaled by a factor ranging from 0.5 to 2.
- **JPEG Compression:** The splicing forgery pictures are stored in JPEG format at various compression levels. Consequently, the photos are compressed using a QF ranging from 50 to 100.
- **Rotation:** The spliced forged pictures are rotated at an angle of 0 to 240 degrees.

The outcomes of the F_1 score with scaling, noise addition, rotation, and JPEG compression attack are shown in Figure 4.9 and compared to other approaches such as Korus [144], Wei [145], Lyu [146], Xiao [147], Huh [148]. The outcome shows that the suggested strategy is robust to various attacks since the proposed approach's F_1 score outperforms other current approaches.



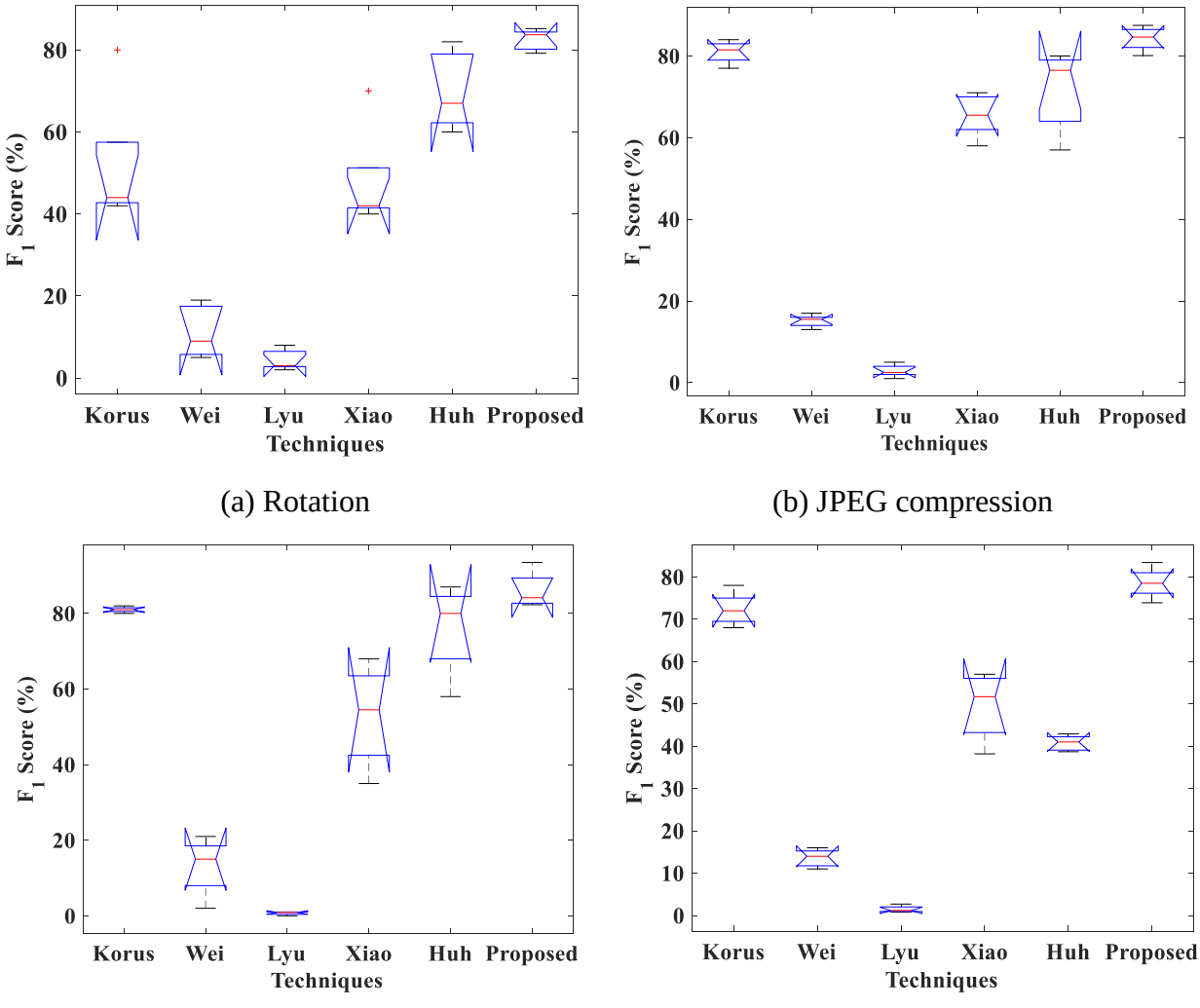
(a) Rotation (b) JPEG compression

(c) Scaling (d) Noise addition

Figure 4.9: Comparison analysis under several attacks

4.5 Statistical Analysis

In this part, the ANOVA is applied to see if there is a difference in the medians of compared techniques. The ANOVA method is used to analyze the outcomes of the suggested approach versus current approaches under several attacks, such as noise addition, scaling, JPEG compression, and rotation. Figure 4.10 displays the statistical analysis of the F_1 score for several procedures under different attacks. According to the analysis of Figure 4.10, the suggested methodology's whiskers reach approximately 100, which is greater than the other current methodologies. The median values are superior to the current methods at a confidence level of 95%. Based on the depiction of the data in Figure 4.10, it is revealed that the current detection approaches are much inferior to the suggested methodology.



(a) Rotation

(b) JPEG compression

(c) Scaling

(d) Noise addition

Figure 4.10: Statistical analysis of F_1 score for several procedures under various attack

4.6 Cross-Dataset Performance

This section assesses cross-dataset performance, a prominent subject of research and an integral part of practical systems where various pictures should be classified. Training a model on one dataset and evaluating it on another dataset obtained from a separate source is referred to as cross-dataset performance. Table 4.8 displays the cross-dataset performance for several performance indicators for all six mentioned datasets employed in this work. As per the table, the recommended approach worked better, suggesting that it can be used for photographs of various sizes and origins.

Table 4.8: Proposed methodology's cross-dataset performance

Training dataset	Testing dataset	Precision	F_1 score	Recall	F_2 score
CASIA v1.0	DVMM	86.44	87.07	87.71	87.45
	IFS-TC	92.00	92.68	93.36	93.09
	Columbia	80.89	80.49	80.09	80.25
	DSO-1	84.50	85.12	85.74	85.49
	CASIA v2.0	95.84	96.07	96.31	96.21
CASIA v2.0	DVMM	85.19	85.71	86.24	86.02
	CASIA v1.0	97.53	97.41	97.29	97.34
	Columbia	87.19	87.94	88.69	88.39
	DSO-1	80.59	80.09	79.60	79.80
	IFS-TC	90.22	90.44	90.66	90.57
Columbia	DVMM	96.81	96.93	97.05	97.00
	CASIA v1.0	84.95	85.47	85.99	85.78
	DSO-1	86.16	86.69	87.22	87.00
	CASIA v2.0	81.48	81.28	81.08	81.16
	IFS-TC	88.88	89.64	90.41	90.10
DVMM	CASIA v1.0	85.15	85.57	85.99	85.82
	Columbia	94.17	94.74	95.33	95.09
	DSO-1	88.16	88.91	89.68	89.37
	CASIA v2.0	83.90	84.21	84.52	84.39
	IFS-TC	79.55	78.96	78.37	78.61

Training dataset	Testing dataset	Precision	F_1 score	Recall	F_2 score
IFS-TC	DVMM	84.22	84.73	85.25	85.04
	CASIA v1.0	80.09	79.60	79.11	79.31
	Columbia	81.18	80.88	80.58	80.70
	CASIA v2.0	88.86	89.51	90.17	89.90
	DSO-1	96.82	97.05	97.29	97.20
DSO-1	DVMM	86.89	87.42	87.96	87.74
	CASIA v1.0	85.71	86.34	86.97	86.72
	Columbia	96.07	96.19	96.31	96.26
	CASIA v2.0	83.25	83.76	84.27	84.06
	IFS-TC	82.01	81.91	81.81	81.85

4.7 Summary

This chapter presents a forgery detection mechanism for validating the detection of ISF. In the beginning, the standard deviation filter is employed to emphasize the variations in the faked photos. To discover the ISF, Markov characteristics are recovered independently from the DWT and LBP regions and integrated. The SVM classifier is then utilized to assess the algorithm's efficacy. The suggested approach achieves an accuracy of 99.76% and 99.69% on CASIA v2.0 and CASIA v1.0, 98.61% and 97.80% on Columbia and DVMM, and 92.50% and 96.90% on DSO-1 and IFS-TC, correspondingly. The simulation findings reveal that combining Markov attributes from the DWT and LBP domains surpasses in terms of various metrics when compared to other current approaches. Furthermore, the suggested method's robustness is proven for JPEG compression, noise addition, scaling, and rotation attacks, and its efficacy is demonstrated by conducting a statistical analysis test employing ANOVA. Based on the analysis of identifying splicing forgery in this chapter, the next chapter will introduce a hybrid approach that can detect both CMF and ISF simultaneously.

CHAPTER 5

HYBRID TECHNIQUE TO DETECT COPY-MOVE AND SPLICING FORGERY

To authenticate the originality and integrity of photographs, digital passive IFDTs are extensively employed nowadays. The most popular kinds of passive image forgery are splicing and copy-move. Numerous methods for detecting these forgeries one at a time are displayed, but very few methods for detecting them together are available. Therefore, a more efficient strategy to identify these forgeries is still required to address the day-to-day demands. A passive hybrid technique based on DFrCT and LBP is presented in this chapter to detect CMF and ISF simultaneously. The DFrCT's fractional parameter is used to improve accuracy, and LBP is utilized to identify tampering artifacts efficiently. The images are then classified as authentic, copy-move, or spliced using an SVM classifier. The duplicated parts of the image are then localized in both the copy-move and spliced images.

5.1 Introduction

The two primary types of passive techniques are ISF and CMF detection. In CMF, matching areas in pictures are identified, whereas feature inconsistencies in ISF are identified. Thus, developing a process for detecting tampering that applies to everyday issues is difficult. Several strategies [129], [138], [143] used a fusion of LBP and other procedures to achieve higher improved accuracy rates. However, a more accurate procedure is necessary to meet the day-to-day forgery dilemmas. On the other hand, the suggested methodology in this chapter has employed LBP and DFrCT because it takes advantage of the versatility of an additional parameter "α" in DFrCT. Moreover, various techniques [129], [138], [143] only intents to classify the picture as forged or authentic. On the other hand, the proposed scheme is employed to classify and localize tampered areas for the two forgeries, CMF and ISF.

5.2 Proposed Technique for Detecting Copy-move and Splicing Forgery

The presented scheme's key goal is to determine if the input picture is tampered with or not. If tampering is identified, the SVM classifier is employed to test for the existence of CMF and ISF. Further tampered portions are identified in both copy-move and spliced pictures.

5.2.1 Framework for Classifying Copy-move and Splicing Forgery

The research work in this chapter offered a scheme that can discover both CMF and ISF at the same time. Figure 5.1 illustrates a thorough structure of the suggested method.

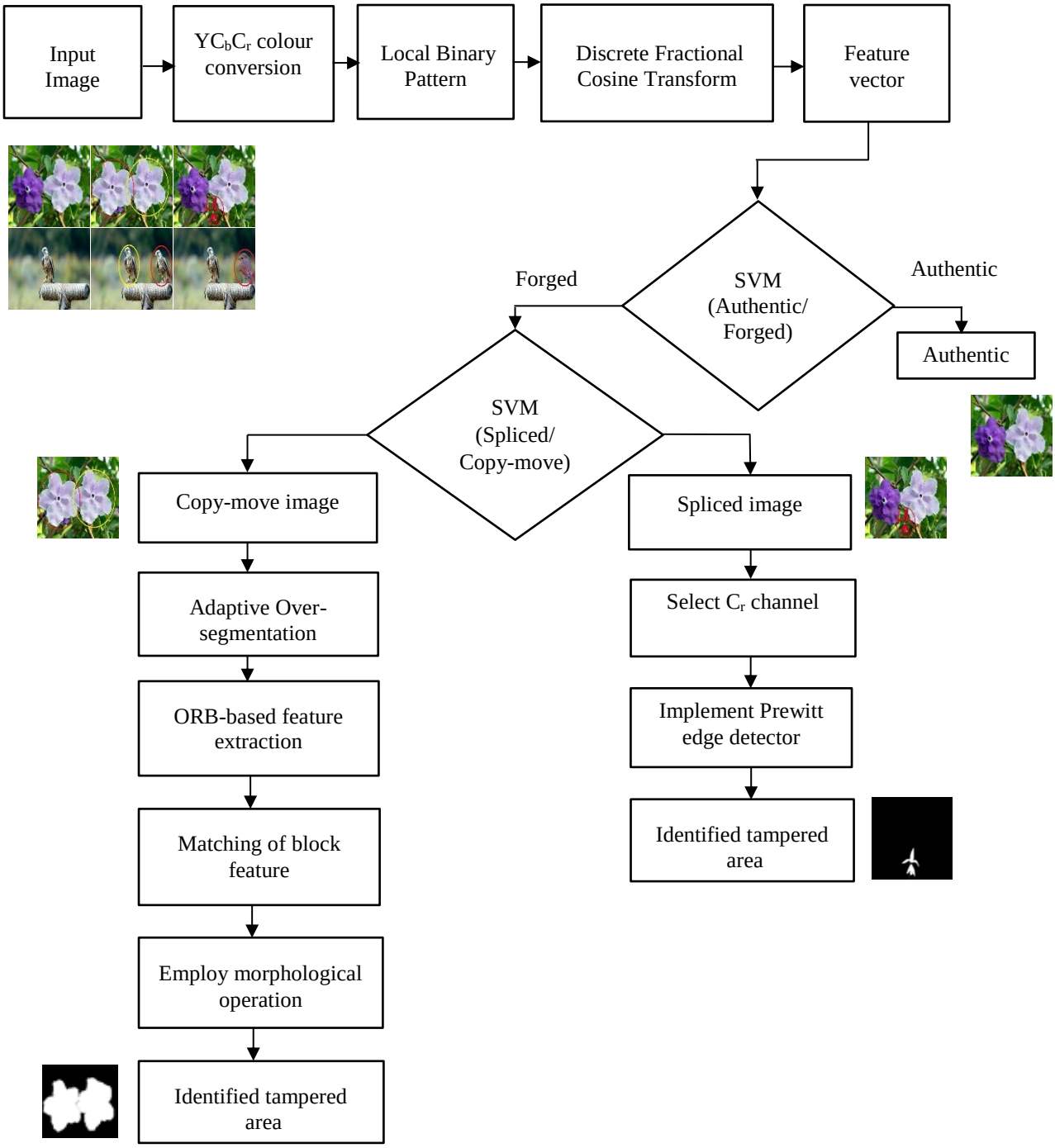


Figure 5.1: Comprehensive framework of the proposed algorithm

The YC_bC_r color channel, a subset of the RGB color channel, is employed in this chapter, with Y representing the luminance part and C_b and C_r representing the chrominance parts. The Y

channel preserves more visual content than the C_b and C_r channels. Human vision is more sensitive to luminance than chrominance. Although the modified picture cannot be clearly seen by human vision, a tiny proportion of counterfeit artifacts remains in the C_r channel. As a result, the C_r channel is employed to detect counterfeit [65]. The YC_bC_r picture is calculated from the RGB picture in Equation (3.1.1) of Chapter 3.

Figure 5.2 shows an example of all channels of the YC_bC_r channel. This figure illustrates that the chrominance component's C_r channel maintains the majority of the tampering artifacts. The outline of the flower in the sixth column of Figure 5.2 stands out in comparison to the other objects in the image. Thus, edges that are tampered with may be easily spotted in the C_r channel. LBP is used to record statistical changes that occur throughout the copy-paste procedure, such as edges. The margins of the copied region alter, resulting in irregularity along the pasted region's boundaries. As a consequence, the local frequency distribution varies. Furthermore, there is no connection between the picture pixels in the area. Identifying statistical irregularities is thus a key step in identifying image alteration. As a result, LBP is suitable for emphasizing tampering artefacts [65], [138], [143]. The LBP operator, described in Chapter 4 as Equation (4.2.2), is applied to each picture.

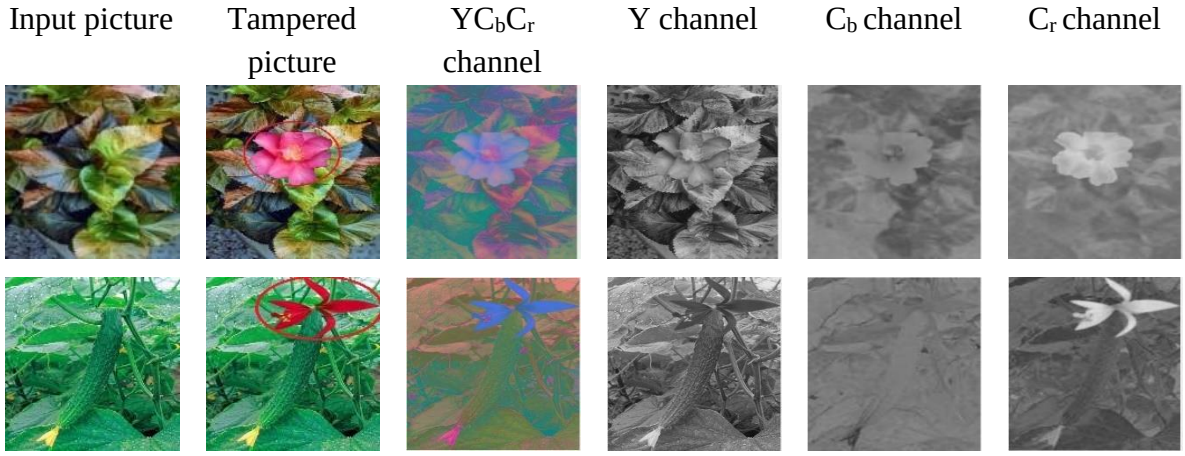


Figure 5.2: Column-wise representation of picture components of YC_bC_r color channel

Furthermore, the LBP image is translated into the frequency domain by DFrCT to acquire changes in a local frequency distribution. Over the last few decades, fractional transforms have been extensively employed in signal and image processing applications [37], [139], [174]–[176]. DFrCT is a general variant of DCT that includes an additional free parameter that is utilized in any task where DCT is advantageous. [174]. DCT for $y[pi]$, $0 \leq pi \leq Pi-1$ is given as below:

$$Y(qi) = \alpha(qi) \sum_{pi=0}^{Pi-1} y[pi] \cos\left[\frac{(2pi+1)\pi qi}{2Pi}\right], \text{ for } \quad 0 \le q \le P-1 \tag{5.2.1}$$

$$\alpha(qi) = \begin{cases} \frac{1}{\sqrt{Pi}} & \text{for} \quad qi = 0 \\ \sqrt{\frac{2}{Pi}} & \text{for} \quad 0 \le qi \le Pi-1 \end{cases} \tag{5.2.2}$$

The inverse DCT (IDCT) may be reconstructed due to the orthogonal sequence as:

$$y(pi) = \sum_{qi=0}^{Pi-1} \alpha(qi) Y[qi] \cos\left[\frac{(2pi+1)\pi qi}{2Pi}\right], \text{ for } \quad 0 \le pi \le Pi-1 \tag{5.2.3}$$

In DFrCT, the DCT kernel eigen decomposition is applied, and the even Hermite-Gauss eigenvectors of the Fourier matrix D_{Pi}^{a} are employed to produce the eigenvectors in the cosine case [174]. The kernel matrix of N-point DFrCT to compute DFrCT coefficients is defined as:

$$K_{Pi,\alpha} = V_{pi} D_{Pi}^{a} V_{Pi}^{T} = V_{pi} D_{Pi}^{2\alpha/\pi} V_{Pi}^{T} = V_{Pi} \begin{bmatrix} 1 & & 0 \\ e^{-2j\alpha} & \ddots & \\ 0 & & e^{-j2(Pi-1)\alpha} \end{bmatrix} V_{Pi}^{T} \tag{5.2.4}$$

where, the rotation angle is signified by $\alpha = a\pi/2$, $V_{Pi=}[v_o \,|\, v_1 \,|\cdots v_{2Pi-2}]$, and D_{Pi} is the diagonal matrix, with diagonal entries that have the equivalent eigenvalues as the column eigenvectors of a matrix V_{Pi}. Two-dimensional DFrCT is commonly utilized in image processing tasks. As a result, 1-D DFrCT (row and column) is utilized twice to generate 2-D DFrCT. So, here, two rotation angles, β and α, are measured individually in two dimensions. Five characteristics, namely mean, standard deviation, skewness, variance, and kurtosis, are derived from coefficients of DFrCT for each picture to reduce dimensionality and generate a feature vector. Thus, five is the feature vector's size in each image. Furthermore, the SVM classifier with RBF kernel is applied to classify the pictures as authentic, copy-move, or spliced. Furthermore, the computational cost of the suggested technique is lowered by employing two binary SVM classifiers. The LIBSVM with an RBF (Gaussian) kernel is used for SVM classifier training since it delivers higher accuracy. As illustrated in Figure 5.1, the first SVM classifier is utilized to determine if photos are original or tampered with. If the input picture is original, it is not analyzed further. The second SVM classifier's objective is to identify

fake photos and categorize them into copy-move and splicing forgery pictures. Algorithm 1 contains the forgery detection algorithm.

Algorithm 1 Hybrid approach for forgery detection

Input- Input image (copy-move, authentic, and spliced)
Output- Whether the picture is copy-move or authentic or spliced

procedure
Transform input picture (I) into $YC_bC_r(Ig)$
for Every component YC_bC_r, Y, C_b, C_r **do**
$Ig_{LBP} \leftarrow$ Apply LBP
$Ig_{DFrCT} \leftarrow$ Apply DFrCT (Ig_{LBP})
end for
Integrate the retrieved attributes to create a feature vector
Employ SVM to classify (Forged/Authentic)
if Forged **then**
Employ SVM to categorize (Spliced/Copy-move)
if Copy-move
$Ig_x \leftarrow$ Implement DWT (Ig)
$Ig_y \leftarrow$ Implement SLICO (Ig_x)
$f_1 \leftarrow$ Implement ORB (Ig_y)
$f_2 \leftarrow$ Feature matching (f_1)
$f_3 \leftarrow$ Matched blocks (f_2)
$f_4 \leftarrow$ Implement morphological operations (f_3)
Identified counterfeit region
end if
if Spliced
$C_r \leftarrow$ Select C_r channel (Ig)
$Ig_{pe} \leftarrow$ Employ Prewitt edge detector (C_r)
Identified counterfeit area
end if
else
Authentic
end if
end procedure

5.2.2 Localization of CMF

If CMF is spotted, a mechanism is carried out to identify the tampered areas. The suggested approach combines keypoint-based and block-based techniques for CMFD. To begin, AS is applied to the input picture using DWT and SLICO to divide it into non-overlapping and

unequal blocks. ORB is then employed in each block to generate the feature points. Next, among the block characteristics, matching is performed, and ultimately, the forged region is located using morphological operations [104], [157], [177]. Initially, a three-level DWT is used in the AS approach for assessing the frequency distribution of the test picture. The proportion of low-frequency distribution is calculated using high and low-frequency energies and is then employed to compute the super pixel's size, as shown in Chapter 3 as Algorithm 1.

The SLICO strategy also divides the input picture into non-overlapping sections of uneven form. This methodology is an adaptation of the k-means clustering approach for successful superpixel creation. The SLICO technique divides the input picture into image blocks with computed superpixel sizes [104], [157]. The suggested method then used ORB to retrieve block attributes from picture blocks. It extracts the characteristics from each picture block because it is more efficient and quicker than previous approaches such as SIFT and SURF. ORB is the result of the combination of feature recognition and extraction approaches like BRIEF descriptor and FAST detector, which have benefits like cheap cost, high efficacy, and resistance to light and blur. The key points are initially determined using the FAST detector. To appropriately measure corner orientation, the intensity centroid (IC) methodology is engaged to add an orientation element to FAST [178]. The order moment $(w+v)^{th}$ of key points with variable intensities $I(w,v)$ is stated as follows:

$$mm_{ab} = \sum_{w,v} w^a v^b I(w,v) \tag{5.2.5}$$

The centroid (C_e) is calculated from keypoint moments using:

$$C_e = \left[\frac{mm_{10}}{mm_{00}}, \frac{mm_{01}}{mm_{00}} \right] \tag{5.2.6}$$

The orientation ψ of key points is then given by a route from the center O to the centroid $O\vec{C}_e$.

$$\psi = a\tan\left[\frac{mm_{10}}{mm_{00}} \Big/ \frac{mm_{01}}{mm_{00}} \right] = a\tan\left(mm_{01}, mm_{10}\right) \tag{5.2.7}$$

where $a\tan(\cdot)$ represents arctangent function. Following that, ORB employs the r-BRIEF, a better version of the steered BRIEF descriptor, in conjunction with a suitable learning phase.

The suggested approach pairs block characteristics with other blocks to compute the proper connections among all blocks. After calculating the total connected feature points, a correlation coefficient map is constructed. As a result, two patches are formed, and the two key points associated with the patches are computed. The matched block pairs are then located by determining the keypoint threshold. Lastly, relevant spots in the linked blocks are derived to pinpoint the suspected forgery area's position. The forged regions may be located using the designated feature points. The superpixels can detect picture counterfeiting and split the input picture effectively. The morphological method is used to recognize image forgery locations. At last, a binary picture containing the identified forged area is attained [104], [178].

5.2.3 Localization of ISF

If the classifier detects a spliced image, further processing is conducted to locate the tampered region. Initially, the C_r channel is chosen from YC_bC_r since it retains several counterfeit artifacts. Furthermore, as previously stated, the borders of the altered area differ from the rest of the picture. Thus, edge detection is critical in finding tampered parts. The magnitude and orientation of the picture are computed using a Prewitt edge detector. Prewitt is often employed to identify horizontal and vertical edges of a picture to find areas where the intensity varies rapidly [179]. Figure 5.3 shows a pair of 3×3 convolution kernels in the Prewitt edge detector.

-1	0	+1
-1	0	+1
-1	0	+1

-1	-1	-1
0	0	0
+1	+1	+1

Figure 5.3: Prewitt operator's 3×3 mask

The magnitude of the local edge gradient is calculated as follows, using the maximum response of all kernels for the pixel position:

$$|E| = \max\left(\left|E_j\right|,\ j = 1:s\right) \tag{5.2.8}$$

where E_j indicates j kernel's response at an appropriate location of pixel, and S signifies total convolution kernels. Vertical and horizontal gradients are computed and combined. The

threshold is then imposed by specifying the threshold value. As a result, an altered part is highlighted in the picture.

5.3 Experimental Results and Discussions

The effectiveness of the suggested technique is proved in the experimentation by employing the following datasets: IMD, CASIA v1.0, COVERAGE, Columbia, CASIA v2.0, and GRIP. The suggested scheme's efficacy is calculated using several image-level performance measures such as TPR, F_1 score, informedness, F_2 score, precision, accuracy, TNR, MCC, and markedness.

5.3.1 Simulation Results

The LBP operator, as previously stated, highlights the tampering artifacts present in the counterfeit image. The LBP picture is then frequency domain translated using DFrCT to retrieve the local frequency fluctuations generated by these abnormalities. The picture is exposed to the LBP operator. LBP has two parameters: q is the total pixels in a spherical neighborhood, and R is its radius [143]. In LBP, each central pixel in a $3_{\times}3$ picture block is compared to its eight neighbors. If the neighbor's value is less than that of the center pixel, it will contain the binary digit '0,' and if the neighbor's value is equivalent to or greater than the center pixel, it will store '1'. Binary code is generated for each specified center pixel by integrating all of these binary digits clockwise, beginning with one of its top-left neighbors. The resultant binary code replaces the center pixel value, and the LBP code is the binary code's decimal value. Figure 5.4 depicts the LBP code calculation.

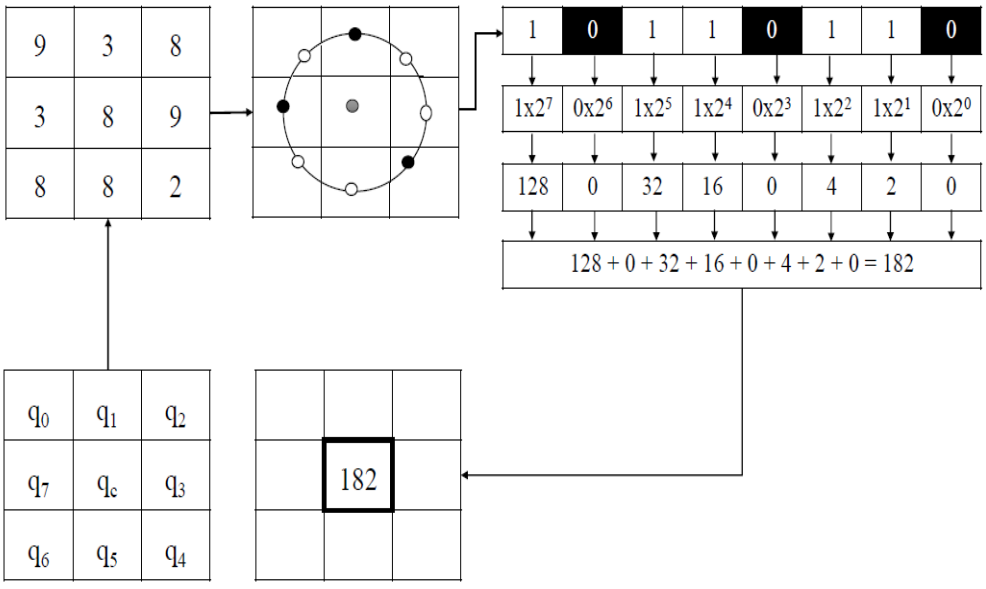


Figure 5.4: Calculation of LBP code

Extensive trials have been carried out on the CASIA v1.0 dataset with various LBP factors (q, R). The testing takes into account the different mixtures of q and R, i.e. (8, 1), (16, 2), and (24, 3), as illustrated in Figure 5.5, and it is discovered that utilizing $q = 8$ and $R = 1$ yields the best results. It is noticed that the accuracy drops with an increase in LBP factors (q, R). Larger values of LBP parameters do not result in greater performance because they neglect small-scale features that are extremely discriminative, reducing performance accuracy.

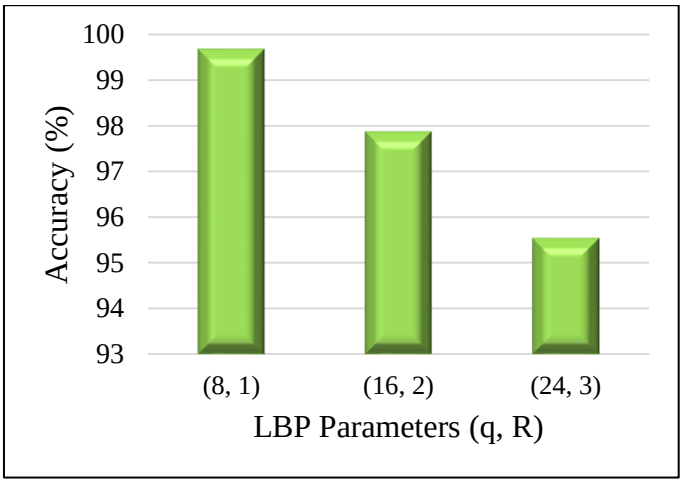


Figure 5.5: The outcome of LBP parameters (q, R) on the accuracy

Furthermore, the LBP code's size becomes unsustainable as it grows exponentially with q. If (q, R) is utilized, the size of LBP codes is 2^q. E.g., when $q = 8$ is used, the value of the LBP code becomes 256. Also, work in [138], [143] demonstrates that the highest efficiency is obtained with (8, 1) LBP factors. Hence, the following trials are run applying the same optimal values of LBP factors, i.e. (8, 1), for additional datasets as well. Furthermore, the influence of fractional order values on DFrCT characteristics is estimated. Figures 5.6 and 5.7 depict the performance of several metrics versus fractional order "α", like sensitivity, accuracy, F_1 score, precision, and F_2 score. As illustrated in Figure 5.6, the fractional order is first diverse from 0 to 1 in 0.1 increments. It has been noted that performance metrics have a higher value for fractional orders between 0.9 and 1.

As shown in Figure 5.7, the fractional order is changed from 0.9 to 1 in 0.01 increments. For CASIA v2.0 and CASIA v1.0 datasets, the suggested approach yields the finest outcomes at fractional order $\alpha = 0.98$. The best results are achieved with fractional order $\alpha = 0.97$, 0.99, 0.94, and 0.96, for the Columbia, GRIP, COVERAGE, and IMD datasets, respectively.

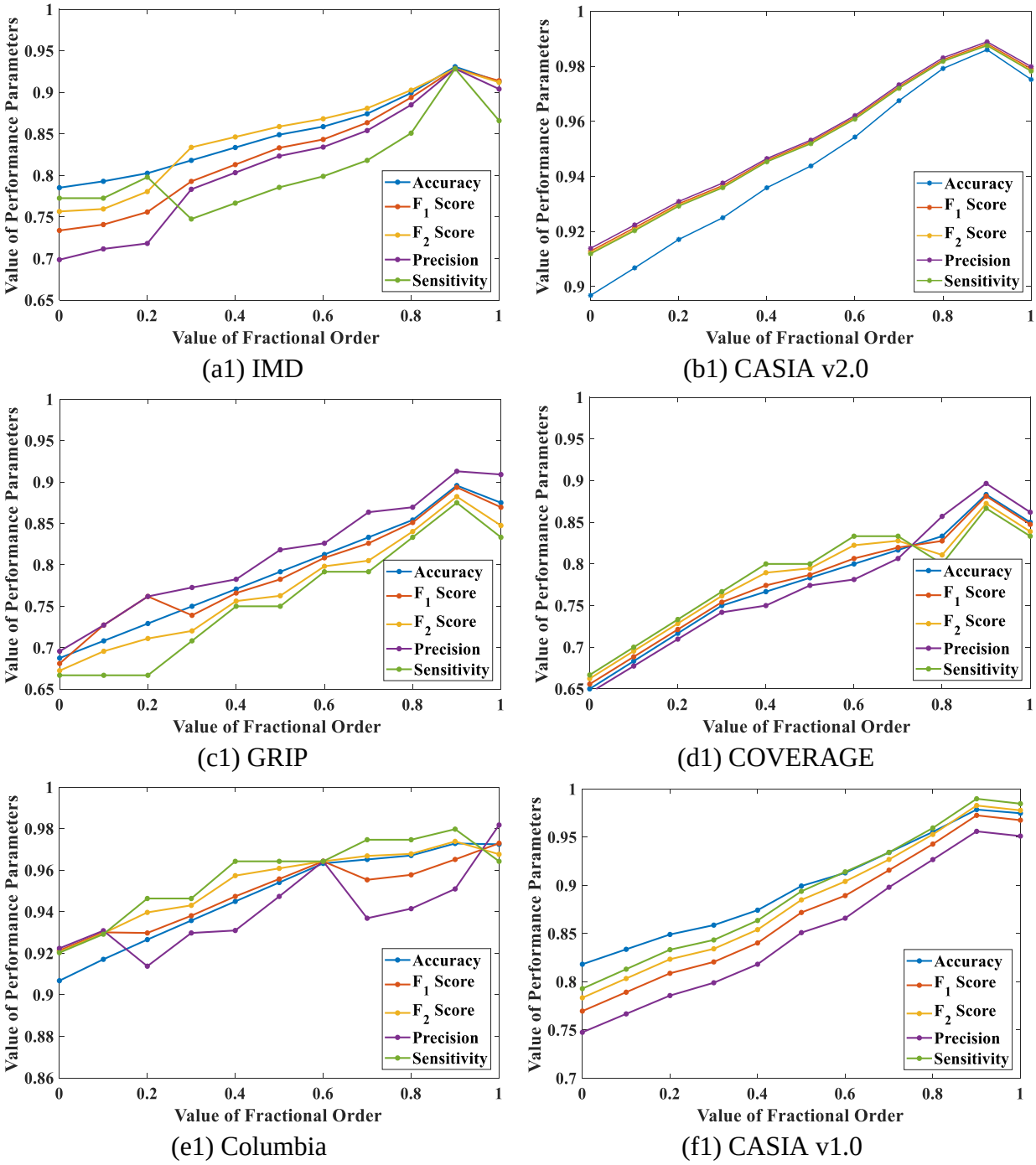
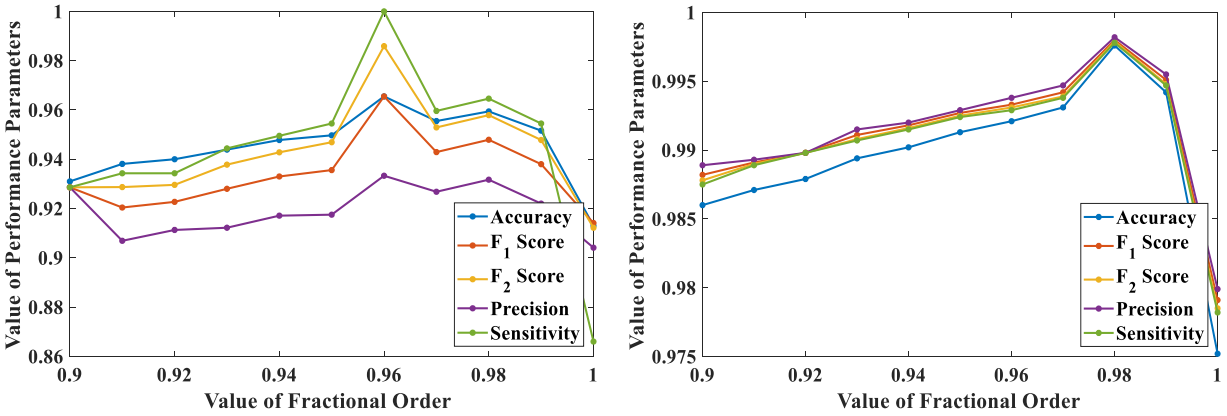


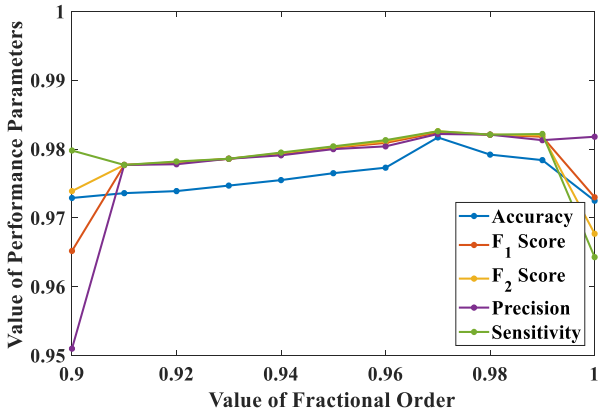
Figure 5.6: Performance metrics for various fractional orders range from 0 to 1 for various datasets

Table 5.1 displays various performance measures for several color channels on specified datasets. It is noticed that the C_r channel's performance characteristics are larger than other color channels, signifying that the C_r channel works efficiently for the suggested scheme. Also, the Y channel outperforms all other color channels on a variety of performance measures.

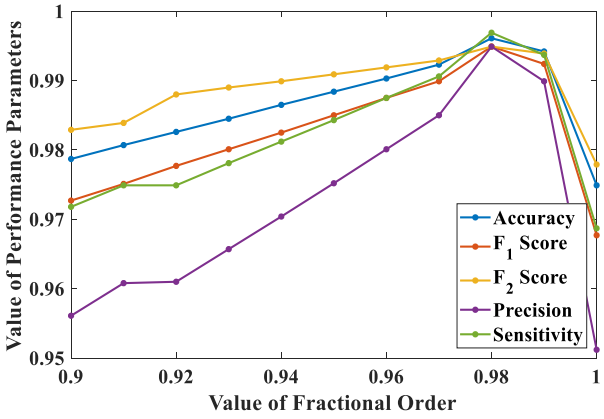


(a2) IMD

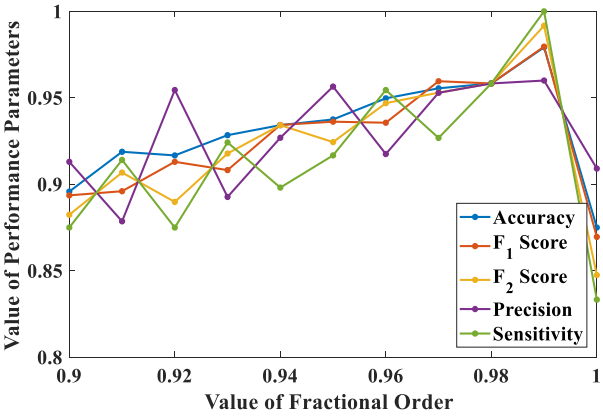
(b2) CASIA v2.0



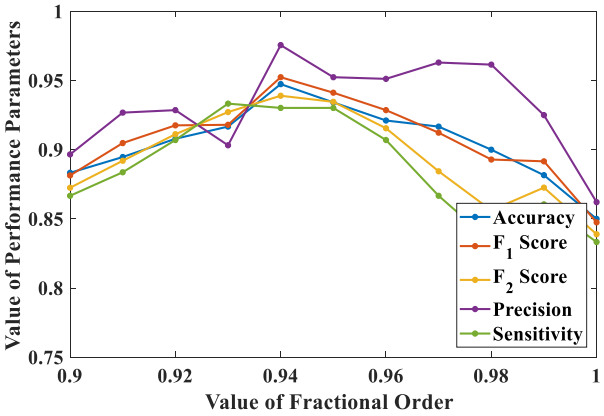
(c2) Columbia



(d2) CASIA v1.0



(e2) GRIP



(f2) COVERAGE

Figure 5.7: Performance metrics for various fractional orders range from 0.9 to 1 for various datasets

5.3.2 Localization Results

The proposed approach is unique in that it performs localization of altered portions on both ISF and CMF after detecting the existence of tampering in the picture. To quantify the efficacy of picture localization, pixel-level metrics like precision (P_p), recall (R_p), and F1 score (F_p),

are calculated for tampered portions of a detected forged picture. These measurements are convenient for assessing the algorithm's overall localization performance [149]. Figures 5.8 and 5.9 show the value of these measures and the detection results for copy-move and spliced images, respectively on different datasets.

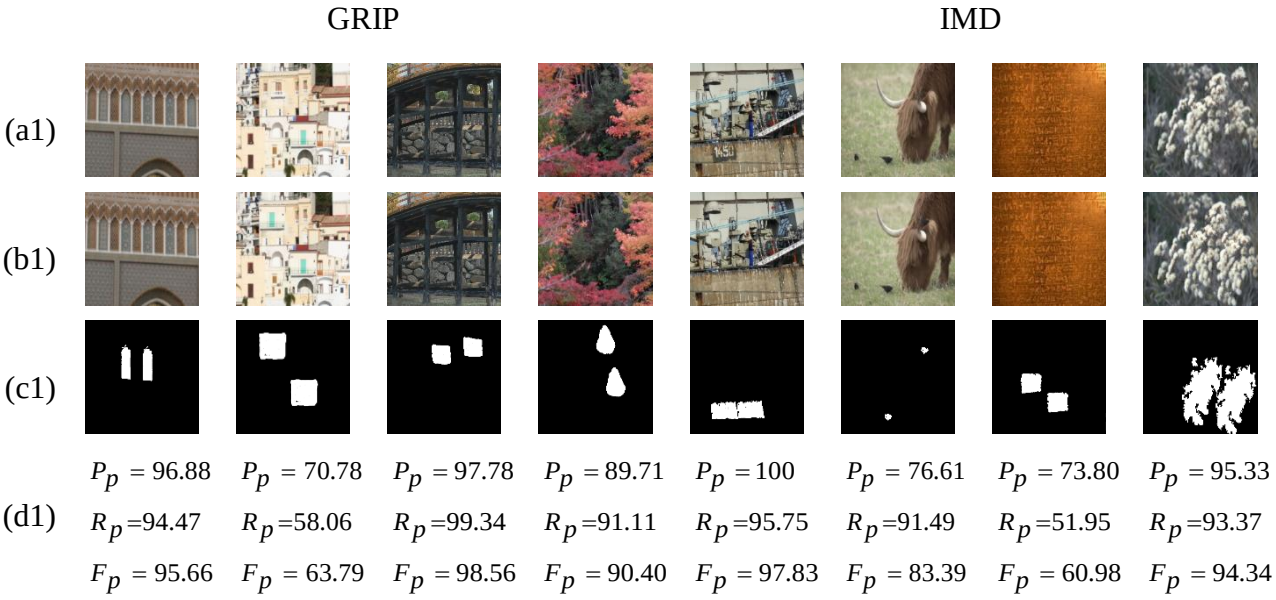


Figure 5.8: CMFD outcomes (a1) original pictures, (b1) tampered pictures, and (c1) detection outcomes, and (d1) qualitative performance metrics

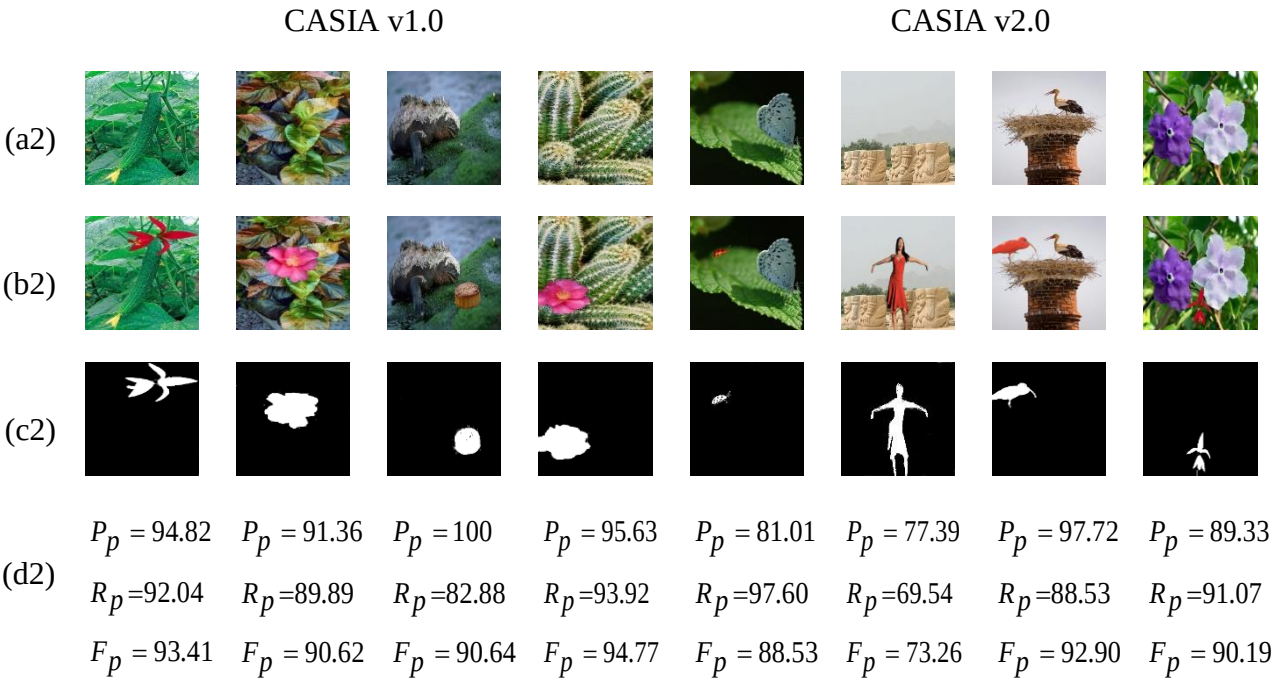


Figure 5.9: ISFD outcomes (a2) original pictures, (b2) tampered pictures, (c2) detection outcomes, and (d2) quantitative performance metrics

Table 5.1: Performance metrics (%) for several color channels

Dataset	Color Channel	Accuracy	TPR	F_1 Score	TNR	Precision	Informedness	MCC	F_2 Score	Markedness
CASIA v1.0	YCbCr	99.42	99.49	99.24	99.37	98.99	98.87	98.77	99.39	98.68
	Y	98.84	98.99	98.49	98.74	98.00	97.73	97.55	98.79	97.37
	Cb	99.22	98.99	98.99	99.37	98.99	98.36	98.36	98.99	98.36
	Cr	**99.67**	**99.56**	**99.56**	**99.74**	**99.56**	**99.30**	**99.30**	**99.56**	**99.30**
CASIA v2.0	YCbCr	99.71	99.73	99.76	99.68	99.78	99.41	99.40	99.74	99.39
	Y	99.55	99.60	99.62	99.48	99.64	99.08	99.07	99.61	99.06
	Cb	99.60	99.64	99.67	99.55	99.69	99.19	99.18	99.65	99.17
	Cr	**99.76**	**99.78**	**99.80**	**99.74**	**99.82**	**99.54**	**99.51**	**99.77**	**99.58**
Columbia	YCbCr	97.25	96.43	97.30	98.11	98.18	94.54	94.51	96.77	94.58
	Y	95.41	96.43	95.58	94.34	94.74	90.77	90.83	96.09	90.89
	Cb	96.33	96.43	96.43	96.23	96.43	92.65	92.65	96.43	92.65
	Cr	**98.17**	**98.21**	**98.21**	**98.11**	**98.21**	**96.33**	**96.33**	**98.21**	**96.33**
IMD	YCbCr	96.55	100	96.55	93.33	93.33	93.33	93.33	98.59	93.33
	Y	89.66	92.86	89.66	86.67	86.67	79.52	79.52	91.55	79.52
	Cb	93.10	100	93.33	86.67	87.50	86.67	87.08	97.22	87.50
	Cr	**98.81**	**100**	**98.99**	**97.14**	**98.00**	**97.14**	**97.57**	**99.59**	**98.00**
GRIP	YCbCr	97.72	100	97.76	95.83	96.00	95.83	95.92	99.17	96.00
	Y	93.75	95.83	93.88	91.67	92.00	87.50	87.58	95.04	87.65
	Cb	95.83	100	96.00	91.67	92.31	91.67	91.99	98.36	98.51
	Cr	**99.23**	**100**	**99.25**	**98.44**	**98.51**	**98.44**	**98.47**	**99.70**	**98.51**
COVERAGE	YCbCr	93.33	90.00	93.10	96.67	96.43	86.67	86.86	91.22	87.05
	Y	90.00	83.33	89.25	96.67	96.15	80.00	80.72	85.86	81.45
	Cb	91.67	91.67	91.23	96.67	96.30	83.33	80.72	88.44	84.18
	Cr	**95.00**	**95.00**	**95.08**	**93.33**	**93.55**	**90.00**	**83.75**	**96.03**	**90.10**

Bold indicates the highest value

5.3.3 Comparative Analysis

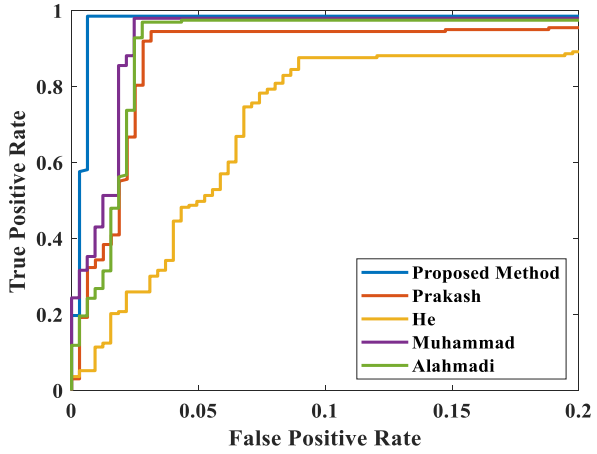
A comparison study with the existing algorithms is performed to measure the efficiency of the suggested work, as shown in Table 5.2. The presented approach obtains 99.67% accuracy, 99.56% TPR, 99.74% TNR, and 99.30% Informedness for CASIA v1.0, demonstrating that the technique is reliable for forgery detection. Similarly, with 99.76% accuracy, 99.78% TPR, and 99.52% informedness, the suggested scheme surpasses the existing CASIA v2.0 approaches. However, the suggested approach has a little lower TNR value than El-Alfy [65]. Also, the experimental findings for the Columbia dataset obtained 98.17% accuracy, 98.21% TPR, 98.11% TNR, and 96.33% Informedness.

Table 5.2: Comparative analysis with current approaches on various datasets

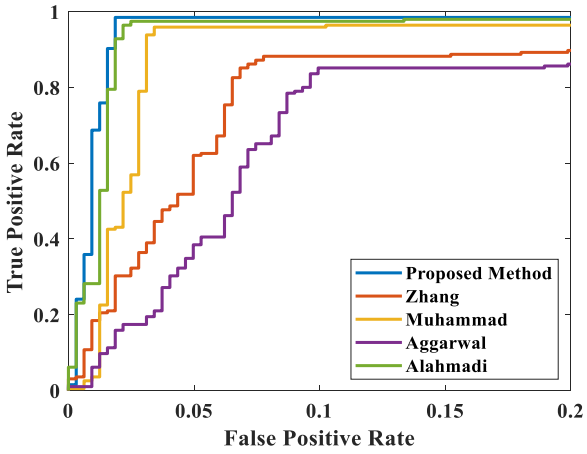
Dataset	Procedures	TNR	Accuracy	Informedness	TPR
CASIA v1.0	He [67]	NC	94.29	NC	NC
	Muhammad [68]	93.91	94.89	89.06	95.15
	Zhang [63]	95.31	96.69	93.36	98.05
	Alahmadi [138]	98.24	97.50	94.99	96.75
	Sheng [140]	NC	98.77	NC	NC
	Lamba [37]	NC	99.65	NC	NC
	Prakash [70]	99.50	99.45	98.87	99.37
	Proposed	**99.74**	**99.67**	**99.30**	**99.56**
CASIA v2.0	He [67]	NC	89.76	NC	NC
	Muhammad [68]	96.53	97.33	95.03	98.50
	Alahmadi [138]	96.88	97.50	96.88	98.31
	El-Alfy [65]	**99.76**	99.73	99.46	99.70
	Lamba [37]	NC	99.01	NC	NC
	Prakash [70]	97.52	96.68	93.29	95.77
	Proposed	99.74	**99.76**	**99.52**	**99.78**
Columbia	Zhang [129]	NC	91.38	NC	NC
	Alahmadi [143]	NC	96.60	NC	NC
	Muhammad [68]	NC	96.39	NC	NC
	Agarwal [135]	NC	93.81	NC	NC
	Proposed	**98.11**	**98.17**	**96.33**	**98.21**

NC: Not calculated by the respective author and Bold indicates the maximum value

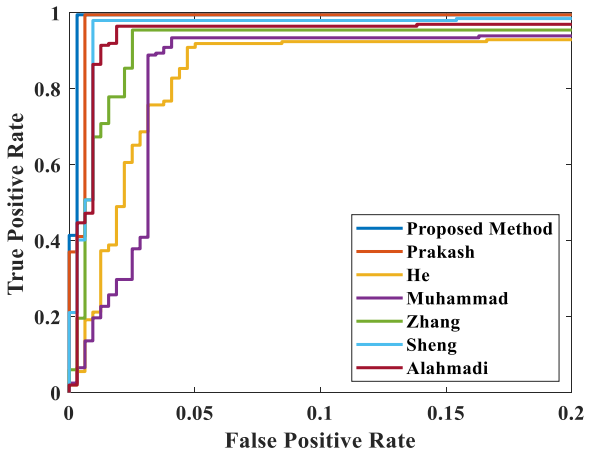
Figure 5.10 compares the ROC curves for several datasets, which are magnified in for easier viewing. It is noted that the presented algorithm's ROC curve is nearer to top left corner, indicating that it achieves a higher accuracy than the current systems. Table 5.3 compares the AUC of various techniques for different datasets.



(a) CASIA v2.0



(b) Columbia



(c) CASIA v1.0

Figure 5.10: Comparative analysis of ROC curves

Table 5.3: Comparison of presented approach's AUC with existing approaches

Dataset	Techniques	AUC
CASIA v1.0	He [67]	94.31
	Muhammad [68]	94.91
	Zhang [63]	96.69
	Alahmadi [138]	97.24
	Sheng [140]	98.76
	Prakash [70]	99.50
	Proposed	**99.75**

CASIA v2.0	He [67]	89.75
	Muhammad [68]	97.53
	Alahmadi [138]	97.24
	Prakash [70]	96.62
	Proposed	**99.76**
Columbia	Zhang [129]	93.39
	Alahmadi [143]	96.62
	Muhammad [68]	96.39
	Agarwal [135]	91.81
	Proposed	**98.17**

Bold indicates the maximum value

Furthermore, Table 5.4 compares the proposed approach to existing algorithms on the COVERAGE, IMD, and GRIP datasets. The sensitivity of 100%, the precision of 98%, the F_1 score of 98.99%, and the F_2 score of 99.59% are achieved on the IMD dataset. The suggested approach achieves a sensitivity of 96.67%, precision of 93.55%, a 95.08% F_1 score, and a 96.03% F_2 score on the COVERAGE dataset.

Table 5.4: Comparison with existing schemes on various datasets

Dataset	Procedures	Sensitivity	Precision	F_1 Score	F_2 Score
IMD	Pan [47]	79.17	88.37	83.52	80.92
	Amerini [48]	79.20	88.40	83.54	80.88
	Emam [82]	87.50	92.70	90.02	88.49
	Yang [52]	78.61	90.27	84.04	80.69
	Bravo [42]	97.92	94.00	95.92	97.11
	Pun [104]	97.92	95.92	96.91	97.51
	Li [49]	100	NC	98.97	NC
	Prakash [88]	87.80	92.30	89.98	88.65
	Proposed	**100**	**98.00**	**98.99**	**99.59**
GRIP	Li [79]	83.75	70.52	76.57	80.72
	Cozzolino [80]	98.75	91.85	95.18	97.28
	Christlein [55]	100	74.76	85.56	93.67
	Amerini [48]	70.00	77.56	73.68	71.39
	Li [49]	100	NC	**100**	NC
	Proposed	**100**	**98.51**	99.25	**99.70**

	Li [49]	80.22	NC	72.28	NC
	Amerini [48]	85.71	40.43	54.95	70.02
COVERAGE	Cozzolino [80]	59.34	61.97	65.45	67.72
	Christlein [55]	46.15	75.00	57.14	49.99
	Proposed	**96.67**	**93.55**	**95.08**	**96.03**

NC: Not calculated by the respective author and Bold indicates the maximum value

Furthermore, the GRIP dataset has a sensitivity of 100%, a precision of 98.51%, a 99.25% F_1 score, and a 99.70% F_2 score. Thus, the findings show that the suggested strategy surpasses other current methodologies. On the GRIP dataset, the suggested approach gets a significantly lower F_1 score than Li [49]. The approach in [49] addressed the matching issue over huge key points, but at the expense of considerable computing cost. Furthermore, Li [49] only identifies CMF, but the suggested approach identifies and localizes two forgeries, namely CMF and ISF.

5.3.4 Run-Time Analysis

Run-time analysis has been estimated and compared with existing approaches in this section. The run-time is the time taken by the system to execute pictures. Similarly, an enormous feature vector length has a significant possibility of increasing the computing load or run-time. On the CASIA v1.0 dataset, Lamba [37] examined block sizes 4×4, 8×8, 16×16 and found that block size 16×16 with feature vector length 14 produced better accuracy. As shown in Table 5.5, our suggested approach uses five features (almost $1/3^{th}$ of [37]) and reduces running time by around 6-7 times. However, the suggested approach has a drawback in that block size is not taken into account since DFrCT produces blocking artifacts when compared to DFrWT employed by [37].

Table 5.5: Run-time comparison on CASIA v1.0 dataset

Methods	Feature-length	Block size	Run time (seconds)
Lamba [37]	14	16×16	140-150 seconds depending on picture size
Proposed	5	--	7-20 seconds depending on picture size

The average run-time of the suggested methodology is compared with Li [49] on the various datasets in Table 5.6. It is revealed that the suggested strategy is more computationally intensive.

Table 5.6: Average run-time (sec) comparison on various datasets

Methods	IMD	COVERAGE	GRIP
Li [49]	86.6	2.3	13.9
Proposed	33.2	1.9	9.4

5.4 Robustness Under Various Attacks

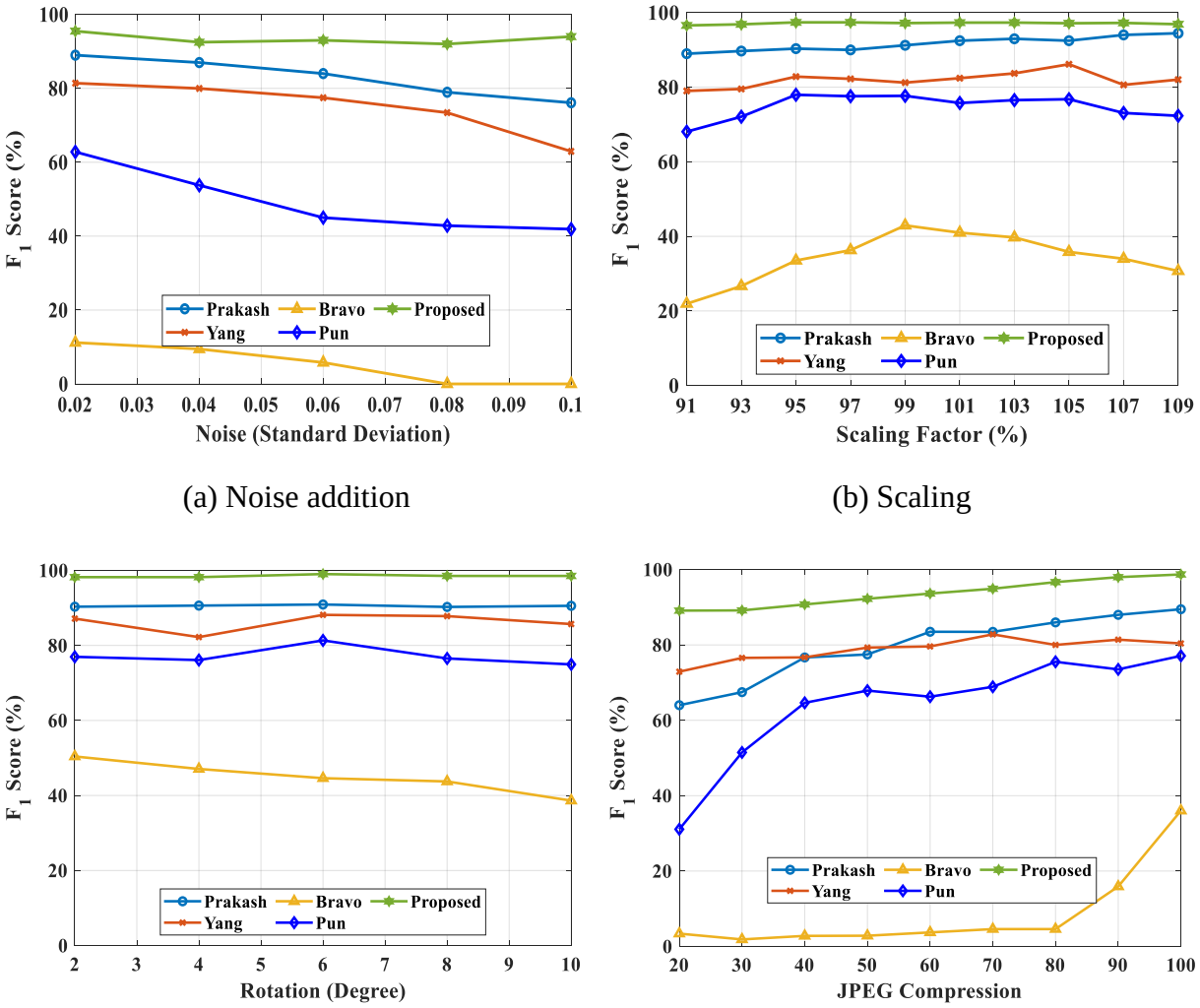
Furthermore, the suggested scheme's robustness has been assessed against numerous attacks i.e. rotation, scaling, noise addition, and JPEG compression attack. In this scenario, false pictures are constructed using 48 photos from the IMD dataset, and the duplicated portions are transformed using the following:

1. **Rotation:** The duplicated areas are rotated with the rotation angle ranging from 2^{o} to 10^{o}, in steps of 2^{o}. In this scenario, an experiment is implemented on $48 \times 5 = 240$ pictures.
2. **Scaling:** The tampered portions are rescaled with a step size of 2% between 91% and 109%. The test is run on a total of $48 \times 10 = 480$ photos in this scenario.
3. **JPEG compression:** The altered photos are JPEG compressed with a QF of 20 to 100. The experiment is run on $48 \times 9 = 432$ photos in this scenario.
4. **Noise addition:** Herein, Gaussian noise is added to cloned regions with STD in a range between 0.02 and 0.1 in 0.02 increments. An experiment is run on $48 \times 5 = 240$ photos in this case.

In Figure 5.11, the outcomes of the F_1 score with various attacks are shown and compared to other conventional approaches. In the graphs here, the x-axis represents the various STDs, scaling factor, rotation angle, and JPEG compression's QF, correspondingly. The y-axis represents the F_1 score value. It is discovered that the value of the F_1 score for various assaults surpasses other current methodologies. Graphs for additional performance metrics can be prepared in the same way.

5.5 Statistical Analysis

In this part, the ANOVA method is used to analyze the outcomes of the suggested approach versus current approaches under different attacks, such as noise addition, scaling, JPEG compression, and rotation.



(a) Noise addition

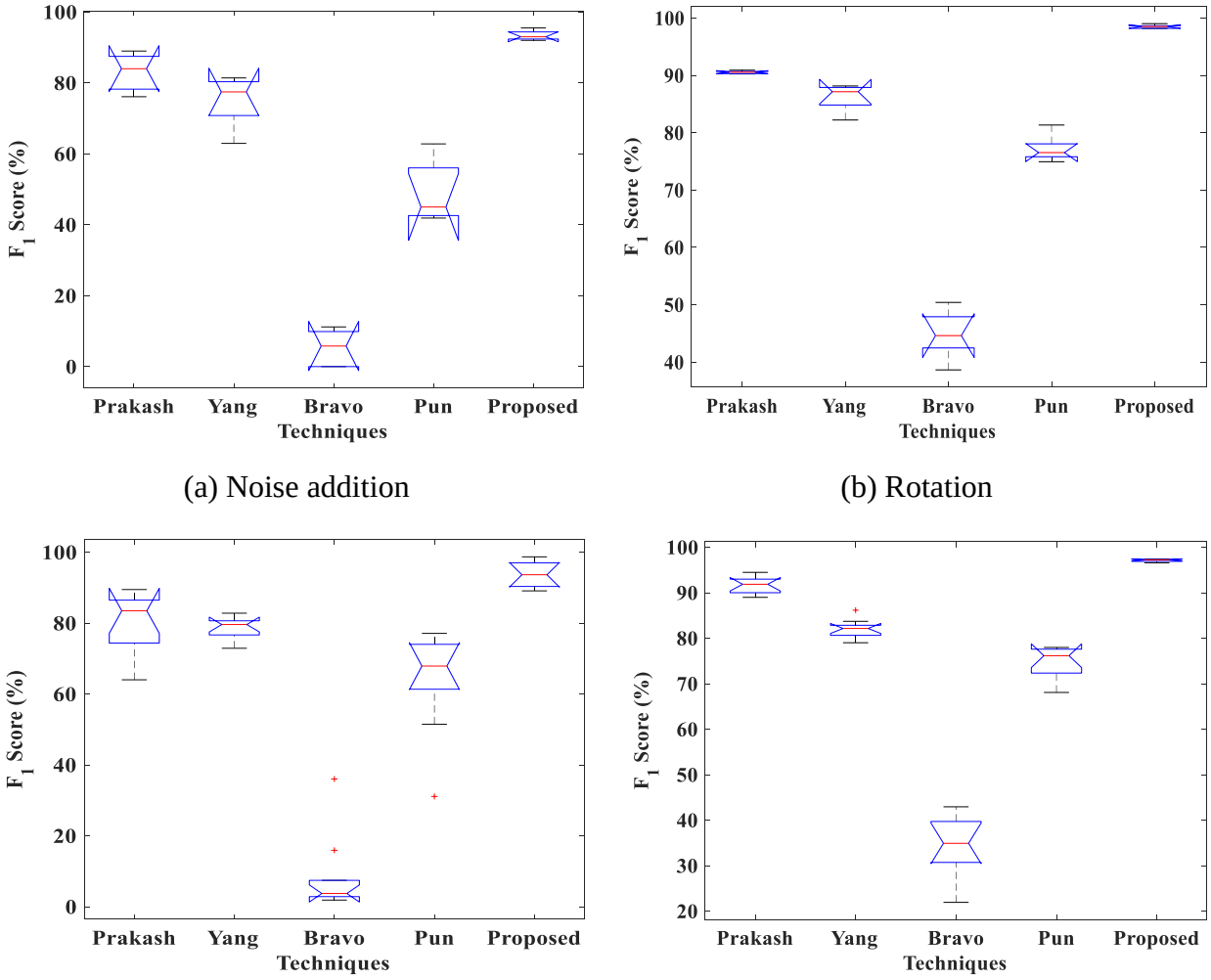
(b) Scaling

(c) Rotation

(d) JPEG compression

Figure 5.11: Comparative analysis under various attacks

Figure 5.12 displays a statistical examination of the F_1 score for various approaches when compared to different attacks. According to box plots in Figure 5.12, the suggested approach's whiskers (that constitute the largest and lowest values of the box plot) reach about 100, which is higher than other current methods. Further, as the notches of two distinct box plots are non-overlapping, the values of a median of the compared processes differ considerably at a confidence level of 95%. Consequently, it suggests that a statistically significant difference exists between various procedures. As a result, the data representation in Figure 5.12 demonstrates that the conventional methodologies are considerably weaker than the suggested methodology.



(a) Noise addition

(b) Rotation

(c) JPEG compression

(d) Scaling

Figure 5.12: Statistical analysis for various techniques under several attacks

5.6 Cross-Dataset Performance

Extensive cross-dataset evaluations are carried out to accurately determine the proposed model's generalization capacity. So, the cross-dataset performance is evaluated in this section, which is a key field of study and an important component in real-world applications where diverse pictures must be classified. Cross-dataset performance is the process of training a model on one dataset and tested on a different dataset obtained from separate sources. Table 5.7 displays the cross-dataset performance for numerous performance measures for all employed datasets.

According to the Table 5.7, the proposed approach performed better, which exhibits that the suggested scheme is applicable to images of diverse sources and sizes.

Table 5.7: Cross-dataset performance of the presented methodology

Training dataset	Testing dataset	Precision	F_1 score	Recall	F_2 score
CASIA v1.0	Columbia	94.07	93.97	93.86	93.90
	CASIA v2.0	96.81	96.70	96.59	96.63
	GRIP	87.69	88.38	89.09	88.80
	IMD	81.85	81.95	82.04	82.00
	COVERAGE	84.12	84.22	84.31	84.27
CASIA v2.0	Columbia	93.21	93.42	93.63	93.55
	GRIP	89.95	90.76	91.59	91.25
	CASIA v1.0	97.94	97.83	97.72	97.77
	IMD	82.61	82.89	83.18	83.06
	COVERAGE	80.72	80.81	80.90	80.87
Columbia	CASIA v2.0	95.01	95.11	95.22	95.18
	GRIP	88.11	88.71	89.31	89.07
	CASIA v1.0	92.98	93.19	93.40	93.32
	IMD	79.63	79.81	80.00	79.92
	COVERAGE	82.08	82.17	82.27	82.23
IMD	Columbia	78.65	79.09	79.54	79.36
	CASIA v2.0	86.00	86.29	86.59	86.47
	GRIP	95.46	95.57	95.68	95.63
	CASIA v1.0	80.27	80.36	80.45	80.41
	COVERAGE	90.82	91.54	92.27	91.98
GRIP	Columbia	81.17	81.27	81.36	81.32
	CASIA v2.0	86.26	86.65	87.04	86.88
	CASIA v1.0	79.00	79.27	79.54	79.43
	IMD	94.33	94.43	94.54	94.50
	COVERAGE	93.65	93.75	93.86	93.82
COVERAGE	CASIA v2.0	78.95	79.13	79.31	79.24
	GRIP	94.98	94.88	94.77	94.81
	CASIA v1.0	80.04	80.13	80.22	80.19
	Columbia	84.73	84.64	84.54	84.58
	IMD	91.44	91.85	92.27	92.10

5.7 Summary

In this chapter, a hybrid passive method that detects both copy-move and splicing forgeries is proposed. Primarily, the input picture is transformed into the YC_bC_r color channel. Then for each picture, LBP is employed and translated into a frequency region employing DFrCT to acquire the local frequency variations. A feature vector is created by extracting five characteristics from DFrCT coefficients: skewness, mean, standard deviation, variance, and kurtosis. After that, the structure is trained with tampered and original photos. The photos are then classified using SVM. Localization is performed after identifying spliced and copy-move pictures to discover the tampered portions. The suggested technique is extensively tested on six datasets, i.e. COVERAGE, CASIA v2.0, IMD, GRIP, CASIA v1.0, and Columbia, yielding accuracy values of 95%, 99.76%, 98.81%, 99.23%, 99.67%, and 98.17%, correspondingly. Also, in terms of various performance characteristics, the suggested approach outperforms existing techniques. The simulation findings further demonstrate that the suggested methodology can identify tampering areas in the existence of rotation, noise addition, JPEG compression, and scaling attacks. In the next chapter, a deep learning model is employed to detect CMF in digital images.

CHAPTER 6

COPY-MOVE FORGERY DETECTION USING DEEP LEARNING

A deep learning CMFD framework is proposed in this chapter, which classifies images as authentic or forged using CLAHE and CNN. The CLAHE algorithm makes the hidden features of the image visible, as some of them are hard to detect in CMF. The effectiveness of the suggested scheme is assessed using standard datasets. In terms of various performance metrics, the experimental study demonstrates the effectiveness of the presented methodology among other methodologies. Also, the efficacy of the presented procedure is demonstrated under several attacks like scaling, noise addition, JPEG compression, and rotation.

6.1 Introduction

Deep learning architecture research has exploded in recent years for a variety of image processing applications such as remote sensing, object recognition, medical imaging, face aging, face editing, and person re-identification as well as image forgery detection issues. Many CNN-based models that learn complex contextual features have been proposed, but their accuracy is low. Despite the fact that numerous similar approaches are presented, none outperform the CMFD scheme. They provide lower values of performance metrics. Following that, this chapter presents a robust deep learning technique for effectively resolving the CMFD problem.

6.2 Proposed Technique for Deep Learning-Based CMFD

In this chapter, the CMFD's primary goal is to categorize the input picture into original and tampered. To accomplish this, the presented structure is distributed into two portions: the foremost conducts pre-processing, and the subsequent is the CNN model, which extracts attributes from the pre-processed images and performs dual classification of pictures as authentic or forged. The proposed system is illustrated in Figure 6.1.

Image pre-processing is essential since training a CNN on raw images is likely to result in poor classification performance and low accuracy. Furthermore, image pre-processing is essential for accelerating training. So, input images, initially, during the image pre-processing section are resized to a defined size without any cropping, since large-sized images increase computational complexity, and the system takes longer to converge. It is demonstrated that an image's size is not the most important element affecting prediction quality using various

transforms, dimensionality reduction, and feature extraction methods for CMFD. The collective features of a pixel group are thought to be more significant than individual pixel characteristics [122], [180]. As a result, the images are resized to a predetermined size i.e. 50 × 50 to allow the computation to proceed without changing the image's attributes or qualities. Image resizing, or down-sampling, is performed on images acquired from several datasets, including MICC-F220, GRIP, MICC-F2000, and IMD, in the proposed technique.

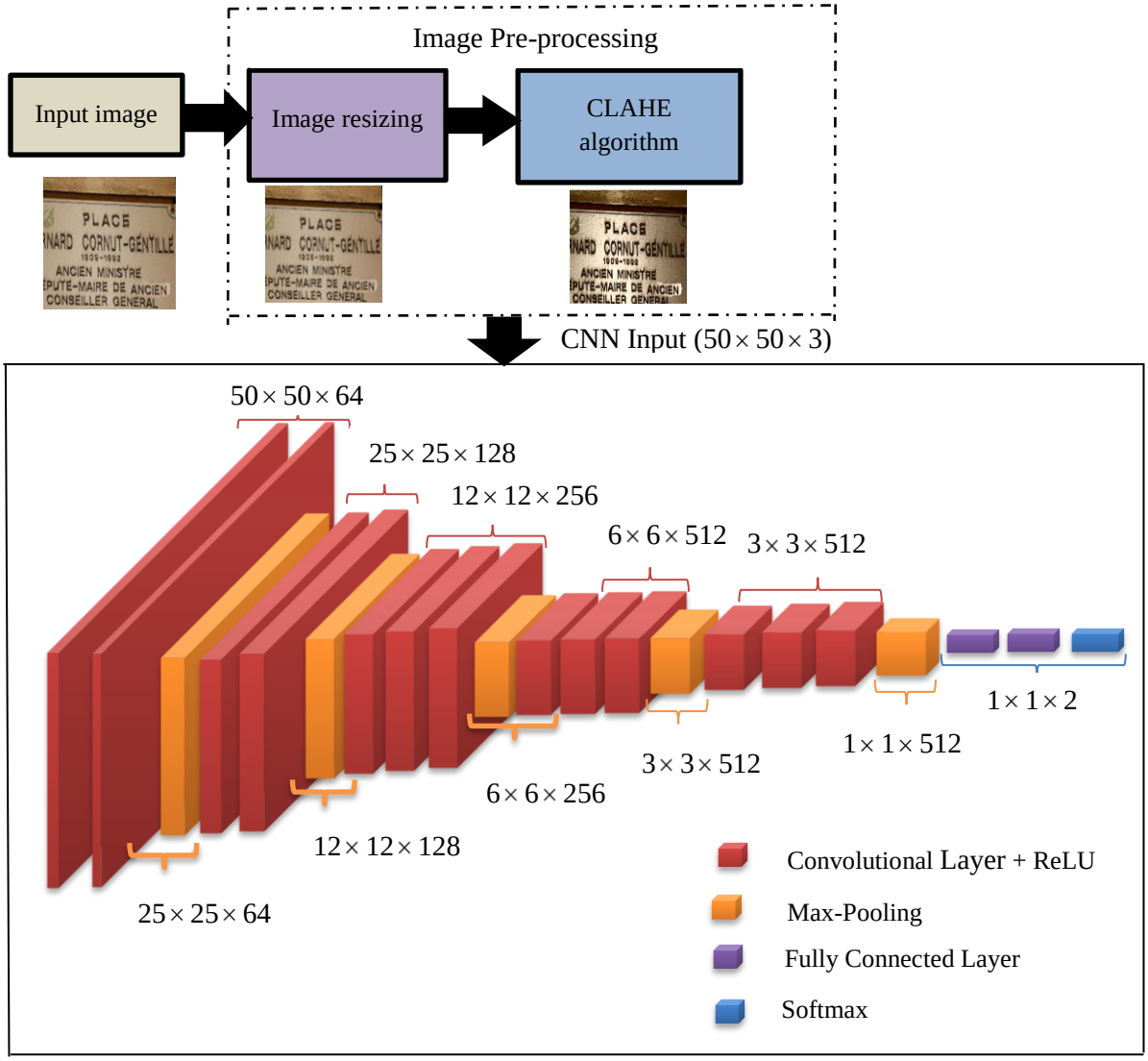


Figure 6.1: Outline of the proposed scheme

The IMD includes images with the same dimensions of 3000 × 2300. That indicates the image to be resized has an input size of 3000 × 2300, where 3000 is the number of rows and 2300 is the number of columns. Due to an uneven number of rows and columns, the input photo is down-sampled by two distinct sample sizes to generate an output image size of 50 × 50. The rows are down-sampled by 3000/50=60 pixels and the columns by 2300/50=46 pixels. As a

result, the sample size for the aforementioned image is (60, 46). Similarly, image resizing is applied to different datasets. Figure 6.2 illustrates the image for greater clarity of image resizing.

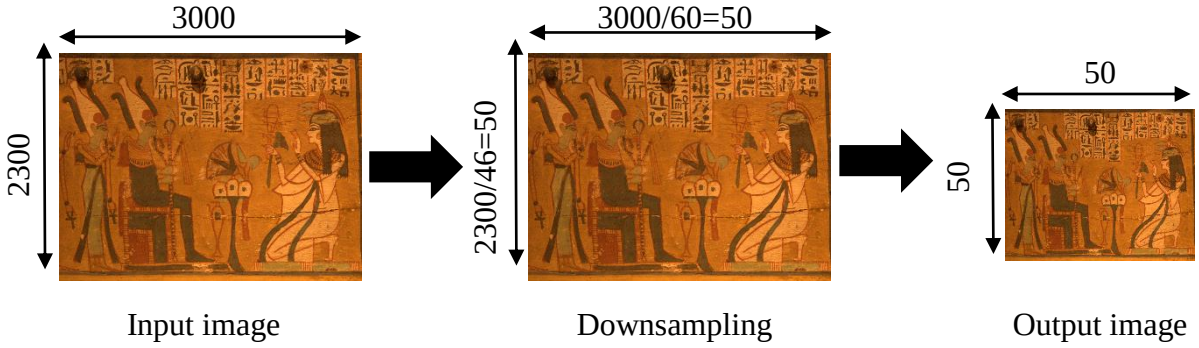


Figure 6.2: Illustration of image resizing

Furthermore, CMF hides certain image characteristics and duplicates them with noise, color, contrast, or other characteristics. As a result, the CLAHE algorithm is utilized in the pre-processing stage to make the image's hidden characteristics visible. Thus, the CLAHE algorithm is applied to the resized images. The image quality is improved by altering image intensities to highlight the target areas. CLAHE method is used for image enhancement; it operates on specific portions of the image rather than the entire image. As the name implies, the CLAHE technique employs histogram equalization after partitioning the image into contextual sections called tiles or blocks. Bilinear Interpolation (BI) is used to combine adjacent tiles and eliminate artificially induced borders. To prevent possible block artifacts, BI is used among blocks. Thus, every pixel value is estimated in adjacent blocks, as illustrated in Figure 6.3. Points a, b, c, and d are the four block's central pixels, whereas p is a random pixel bordered by four blocks. Pixel p using BI is obtained as:

$$T(p(i)) = m \cdot (n \cdot T_a \cdot p(i) + (1-n) \cdot T_b \cdot p(i)) + (1-m) \cdot (n \cdot T_c \cdot p(i) + (1-n) \cdot T_d \cdot p(i)) \qquad (6.2.1)$$

where $T(\cdot)$ indicates remapping algorithm; $p(i)$ is the value of a random pixel i with coordinates (x, y). Blocking artifacts are eradicated during the interpolation phase. Because of the individual dispensation of blocks, CLAHE provides a small computing complexity. The contrast inhomogeneous regions might be restricted to avoid increasing noise that may exist in the image [122], [181]. Also, CLAHE makes the hidden characteristics of the image clearly visible, as some hidden features of the forged image are hard to detect in CMFD. Figure 6.4 shows an illustration of CLAHE performance.

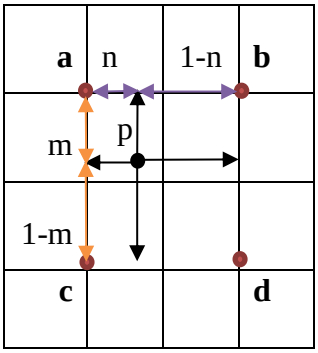


Figure 6.3: Bilinear interpolation used in CLAHE

In Figure 6.4 (b), a forged portion is not clearly apparent to human eyes. As a result, CLAHE enhances the contrast of the picture to show hidden information, i.e. the forged portion, for clear visibility. Further, the pre-processed image is employed as an input to the CNN model.

(a) (b) (c)

Figure 6.4: Illustration of an image after applying CLAHE (a) authentic image, (b) forged image where the yellow circle is copied part and the red circle is forged part, and (c) resultant image after applying CLAHE

Artificial Intelligence (AI) is now widely used in image processing. Artificial neural networks, a subdomain of AI, have become the most often used procedures in image processing. A CNN, in contrast, is a specific kind of neural network that processes and analyses data using perceptrons, artificially designed neurons for supervised learning. To build CNN, various perceptrons are coupled in a multi-layer way to accomplish the least amount of total processing. An input layer (containing pre-processed images) is followed by numerous convolutional layers in the CNN. Each convolution layer employs Rectified Linear Unit (ReLU) activation, after which comes a pooling layer that produces results, which are then catered to the Fully Connected (FC) layer, which generates output.

Convolution filters, which are 2D digital filters, are used to create the convolutional layer. To build the feature maps, digital filters are convolved with the input image. An activation function is required during the training phase, thus it is applied after each convolution step. The ReLU function is a proficient activation function utilized in deep learning. Typically, the activation

function is a procedure that is assigned to each part of the output produced after convolution. It empowers more effective and faster training by transforming negative input to zero while holding positive values. The activation process does not affect the input's size. In the proposed model, there are thirteen convolutional layers and each layer has a 3×3 filter size, one stride, and one padding. Each convolutional layer has a different number of filters and outputs. The filters in the final convolutional layer identify more complicated patterns than the filters in the preceding convolutional layers. Table 6.1 represents the different properties of each convolutional layer. The first 36 visual features learned by the convolutional layers are represented in the table below.

The pooling layer reduces the image's size. It merges adjacent pixels in a distinct area of the image to produce a particular representative value. It depicts the pixel's highest or average value. The proposed approach employs the max-pooling layer. By merging the outcomes of the convolution layer with various weights, the FC layer decreases the amount of input to match the number of classes for which CNN is taught. CNN's last layer i.e. classification layer, classifies the FC layer's output. In a multi-class classification procedure, the soft-max method is typically assigned to this layer to allocate every entity of the FC layer to a specific class. The probability for each feasible class is calculated by the Soft-max algorithm [122], [125], [182]. Some weights and biases are applied to every neuron in the hidden layer, which is shared at every location in the convolutional layer, and therefore every convolutional layer has its weight, which is provided by the following equation:

$$Weight = D_w \times W_w \times H_w \tag{6.2.2}$$

where D_w is the previous layer descriptor, W_w is kernel width, and H_w is kernel height. Table 6.2 shows the various weights allocated to the layers. Table 6.2 shows that in the first convolutional layer, 64 high-pass filters of size $3\times3\times3$ are used to initialize the weights. As a result, the total weight of the first convolutional layer is $3\times3\times3\times64$=1728 weights. As a result, the weight of each layer may be classified based on the filter size, total filters, and the previous layer description. For instance, layer 7 (conv2) has $3\times3\times64\times128$ weights, which implies that 3×3 is the filter size, 128 is the total filters employed in the current layer, and 64 is the total filters used in the preceding layer. In this situation, the total weight can be calculated as $3\times3\times64\times128$=73,728 weights. Similarly, the weights of each layer can be classified using the weight descriptions provided in Table 6.2.

Table 6.1: Characteristics of each convolutional layer

S. no.	Layer No.	Layer Name	No. of Filters	Output	Visual Features
1	2	conv1	64	$50 \times 50 \times 64$	
2	4	conv2	64	$50 \times 50 \times 64$	
3	7	conv3	128	$25 \times 25 \times 128$	
4	9	conv4	128	$25 \times 25 \times 128$	
5	12	conv5	256	$12 \times 12 \times 256$	
6	14	conv6_1	256	$12 \times 12 \times 256$	
7	16	conv6_2	256	$12 \times 12 \times 256$	
8	19	conv7	256	$6 \times 6 \times 256$	
9	21	conv8_1	512	$6 \times 6 \times 512$	
10	23	conv8_2	512	$6 \times 6 \times 512$	
11	26	conv8_3	512	$3 \times 3 \times 512$	
12	28	conv8_4	512	$3 \times 3 \times 512$	
13	30	conv8_5	512	$3 \times 3 \times 512$	

In the Softmax layer, the Softmax function converts K true values vector into a vector of K true values which add to 1. If one of the inputs is small or negative, Softmax transforms it to lesser probability, and if the input is high, it converts it to huge probability, but it always stays between 0 and 1. The softmax function $\sigma : \Re^K \to [0,1]^K$ is given by the following formula when K is greater than one:

$$\sigma(\vec{v}) = \frac{e^{v_b}}{\sum_{c=1}^{K} e^{v_c}} \text{ for } b = 1,...,KL \text{ and } z = (z_1, z_2, ..z_K) \in \Re^K \tag{6.2.3}$$

where all v_b values are input vector elements and may accept any real value i.e. negative, zero, or positive. e^{vb} is exponential function applied to every element of input vector. The

normalization factor $\left(\sum_{c=1}^{K} e^{v_c}\right)$ at the bottom of the calculation guarantees that all of the function's output values add to 1, resulting in a legal probability distribution.

Table 6.2: Description of various weights

Layer No.	Layer Name	Description of Weights	Layer No.	Layer Name	Description of Weights
2	conv1	$3\times3\times3\times64$	19	conv7	$3\times3\times256\times512$
4	conv2	$3\times3\times64\times64$	21	conv8_1	$3\times3\times512\times512$
7	conv3	$3\times3\times64\times128$	23	conv8_2	$3\times3\times512\times512$
9	conv4	$3\times3\times128\times128$	26	conv8_3	$3\times3\times512\times512$
12	conv5	$3\times3\times128\times256$	28	conv8_4	$3\times3\times512\times512$
14	conv6_1	$3\times3\times256\times256$	30	conv8_5	$3\times3\times512\times512$
16	conv6_2	$3\times3\times256\times256$	33	fc	2×512

*conv represents convolution layer

*fc represents the fully connected layer

Consider a CNN that determines if a picture is authentic or forged. Because the image must be either authentic or counterfeit and cannot be both, the two groups are mutually incompatible. Normally, the network's FC layer produces numbers like [-7.876, 2.13], which are not regulated and cannot be read as probabilities. By adding the Softmax layer to the network, numbers may be converted into a probability distribution. This means that output may be exposed to the user, for example, a model is 95% confident that this is authentic.

So, while training a CNN, the Softmax layer is crucial. In the proposed approach, CNN is used to learn to differentiate between authentic and forged images. The general classification problems do not have natural orderings of the classes. Fortunately, researchers found a straightforward approach to represent categorical data many years ago: the one-hot encoding. A one-hot encoding is a vector with as many components as many classes are present. The component corresponding to the class of a certain instance is set to 1, and all other components are set to 0. For example, two class labels will be encoded as 0 and 1. Then encoded to vectors as Class 0: [1, 0] and Class 1: [0, 1]. This is called one-hot encoding. So in the presented work, the authentic image is set to class 1 and the forged image to class 2. Preferably, when an authentic image is fed into the proposed network, it will output the vector [1, 0]. When a forged

picture is supplied, the vector [0, 1] is output. The neural network image processing is completed at the last FC. This layer generates two non-probabilistic scores for authentic and forged images. It is a common exercise to include a Softmax layer after a neural network to turn the output into a probability distribution. The neural network weights are unsystematically set at the start of training. As a result, the authentic image is processed and transformed to scores [1.9, 0.1] by the image processing techniques. When this is run through the Softmax layer, the output probabilities are attained by the following equation:

$$\begin{bmatrix} P(authentic) \\ P(forged) \end{bmatrix} = \sigma\left(\begin{bmatrix} 1.9 \\ 0.1 \end{bmatrix}\right) = \begin{bmatrix} \frac{e^{1.9}}{e^{1.9}+e^{0.1}} \\ \frac{e^{0.1}}{e^{1.9}+e^{0.1}} \end{bmatrix} = \begin{bmatrix} 0.86 \\ 0.14 \end{bmatrix} \tag{6.2.4}$$

The probabilities, [0.86, 0.14], are obtained by giving [1.9, 0.1] into Softmax function. It is closer to the desired outcome of [1, 0], indicating that the input image is authentic. This illustration shows that Softmax gives non-integer data between 0 and 1 that may be read as probabilities. As a result, the images can be categorized into both classes i.e. authentic and forged.

Table 6.3 exemplifies the details of the suggested CNN model layers. There are eight convolutional layers comprising features maps of 64 for the 1st and 2nd layer, 128 for the 3rd and 4th layer, 256 for the 5th and 6th layer, and 512 for the 7th and 8th layers. All of the convolutional layers use ReLU activation, and further, a 2×2 max-pooling layer consisting of two strides is employed after each convolutional layer. The experimental outcomes are presented in the subsequent segment.

6.3 Experimental Results and Discussions

The below-given section provides a comprehensive assessment of results delivered using the suggested algorithm. Furthermore, the incurred outcomes are compared to the prior outcomes mentioned in the literature section. Moreover, the robustness of the suggested technique against several geometrical attacks is evaluated and compared with the current techniques. The suggested system is executed using MATLAB R2020a (9.8.0.1417392). The simulations are carried out using a 64-bit operating system with 8.00 GB memory, operated by Microsoft Windows 8.1.

Table 6.3: Details of proposed CNN model layers

Layer No.	Layer Name	Description of Layer
1	Input layer	Image with zerocenter normalization
2, 4	conv1, conv2	64 $3\times3\times3$ convolutions containing [1 1] stride and [1 1 1 1] padding
3, 5, 8, 10, 13, 15, 17, 20, 22, 24, 27, 29, 31	ReLU_1, ReLU_2, ReLU_3, ReLU_4, ReLU_5, ReLU_6, ReLU_7, ReLU_8, ReLU_9, ReLU_10, ReLU_11, ReLU_12, ReLU_13	ReLU
6, 11, 18, 25, 32	pool_1, pool_2, pool_3, pool_4, pool_5	2×2 max pooling containing[2 2] stride and [0 0 0 0] padding
7	conv3	128 $3\times3\times64$ convolutions containing [1 1] stride and [1 1 1 1] padding
9	conv4	128 $3\times3\times128$ convolutions containing [1 1] stride and [1 1 1 1] padding
12	conv5	256 $3\times3\times128$ convolutions containing [1 1] stride and[1 1 1 1] padding
14, 16	conv6_1, conv6_2	256 $3\times3\times256$ convolutions containing [1 1] stride and [1 1 1 1] padding
19	conv7	512 $3\times3\times256$ convolutions containing [1 1] stride and [1 1 1 1] padding
21, 23, 26, 28, 30	conv8_1, conv8_2, conv8_3, conv8_4, conv8_5	512 $3\times3\times512$ convolutions containing [1 1] stride and [1 1 1 1] padding
33	fc	2 fully connected layer
34	softmax	softmax
35	classoutput	output with two classes authentic and forged

6.3.1 Simulation Results

In the experimentation, datasets such as MICC-F220, MICC-F2000, GRIP, and IMD are engaged to validate the efficacy of the suggested approach. Several tests have been conducted using varying epochs to identify the total epochs that provide the highest performance. Before model training, the dataset is randomly partitioned into the test, train, and validation sets. After each epoch of training, the model is assessed on a validation set. If the model's performance on the validation set begins to deteriorate (e.g., loss begins to grow or accuracy begins to drop) or reaches a constant value, the training process is terminated. This is known as early stopping which is used for solving the issue of overfitting the data.

The goal of the training period is to reduce loss function to its lowest possible value. The optimization algorithm (or optimizer) is the most common strategy used nowadays for training a machine learning model to reduce its loss. In the proposed approach, to minimize the loss function, the Stochastic Gradient Descent with Momentum (SGDM) optimizer is used [183]. The standard gradient descent approach changes the network constraints (weights and biases) to minimize loss function by taking minor steps in the path of negative gradient of loss function at every iteration. The stochastic gradient descent algorithm can fluctuate across the gradient descent path to the optimum. The addition of momentum term to constraint apprise is one technique to decrease the fluctuation. The SGDM update is given in the equation below.

$$\varpi_{mi+1} = \varpi_{mi} - \beta \nabla L(\varpi_{mi}) + \eta(\varpi_{mi} - \varpi_{mi-1}) \tag{6.3.1}$$

where, mi is iteration number, $\beta > 0$ is learning rate, ϖ is parameter vector, $L(\varpi)$ is the loss function, $\nabla L(\varpi)$ is a gradient of the loss function, and η determines the preceding gradient step's contribution to the current iteration. The learning rate is a hyper-parameter used in neural network training that regulates how much to modify the model in response to predicted error each time the model weights are updated. It has a modest positive number, usually ranging between 0.0 and 1.0. In the proposed scheme, the model is trained by an SGDM optimizer and has 32 size of mini-batch and a learning rate of 0.0001. To make the study more visible, Figure 6.5 displays the implementation environment of the proposed model.

Table 6.4 displays the results obtained on the MICC-F220 dataset for various epochs. It is perceived that the accuracy of the validation set increases from 50% to 99.34% and training accuracy rises from 62.50% to 100% with 50 epochs. Moreover, the training loss and validation loss keep on decreasing. The highest performance in this dataset is obtained with 50 epochs

because, after the 50th epoch, the value of validation accuracy becomes constant, as shown in Table 6.4. So, the training procedure is terminated after the 50th epoch, with a 99.34% accuracy.

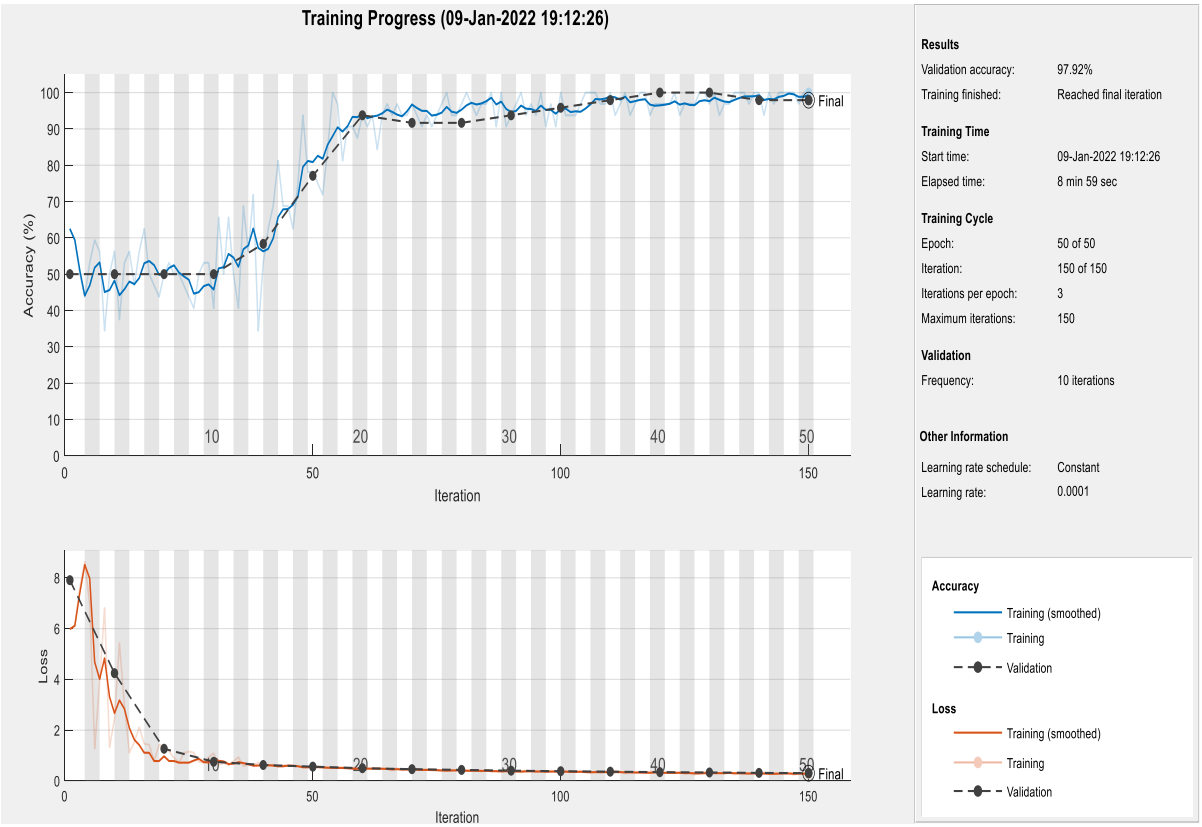


Figure 6.5: Implementation environment of proposed model

Table 6.4: Training results achieved by proposed scheme on MICC-F220 dataset

Epochs	Iteration	Training accuracy	Training loss	Validation accuracy	Validation loss	Base Learning rate
1	1	62.50%	5.6809	50.00%	7.9712	0.0001
10	40	87.50%	0.5178	59.21%	0.9239	0.0001
20	80	96.88%	0.1218	94.08%	0.1297	0.0001
30	120	96.88%	0.0648	97.37%	0.0644	0.0001
40	160	98.36%	0.0599	98.68%	0.0448	0.0001
50	200	99.02%	0.0430	99.34%	0.0352	0.0001
60	220	100.00%	0.0229	99.34%	0.0346	0.0001

It is perceived from Table 6.5 that the value of validation accuracy for the GRIP dataset improves from 50% to 98.21% and then decreases after the 47th epoch to 97.32%. As a result, the finest performance is obtained with 47 epochs on the GRIP dataset.

Table 6.5: Training results attained by proposed scheme on GRIP dataset

Epochs	Iteration	Training accuracy	Training loss	Validation accuracy	Validation loss	Base Learning rate
1	1	56.25%	6.2494	50.00%	7.9435	0.0001
7	20	37.50%	0.9234	51.79%	0.7318	0.0001
14	40	65.63%	0.6158	56.25%	0.7217	0.0001
20	60	53.13%	0.6534	83.04%	0.4755	0.0001
27	80	81.25%	0.4402	75.89%	0.4601	0.0001
34	100	87.50%	0.3320	95.54%	0.3292	0.0001
40	120	95.99%	0.2627	96.43%	0.2898	0.0001
47	140	96.88%	0.2590	98.21%	0.2524	0.0001
50	150	93.75%	0.2368	97.92%	0.2421	0.0001

The highest performance in the IMD dataset is obtained with 40 epochs because, after the 40^{th} epoch, the value of validation accuracy becomes constant, as shown in Table 6.6. Initially, the validation accuracy rises from 50% to 100% with epochs ranging from 1 to 30 and then attains a constant value. Thus, the training process is stopped after achieving the best performance. Furthermore, the validation and training loss continue to decrease.

Table 6.6: Training results attained by the proposed scheme on the IMD dataset

Epochs	Iteration	Training accuracy	Training loss	Validation accuracy	Validation loss	Base Learning rate
1	1	46.88%	3.3398	50.00%	7.9712	0.0001
10	20	62.50%	0.6024	76.67%	0.7326	0.0001
20	40	93.75%	0.3330	83.33%	0.3281	0.0001
30	60	100.00%	0.2204	100.00%	0.2113	0.0001
40	80	100.00%	0.1809	100.00%	0.1706	0.0001
50	100	100.00%	0.1526	100.00%	0.1406	0.0001

Table 6.7 shows the outcomes obtained for several epochs on the MICC-F2000 dataset. With 10 epochs, it is noticed that validation set accuracy grows from 74% to 99.17%, and training accuracy increases from 78% to 98.89%. Furthermore, the training and validation losses are reducing. The best performance in this dataset is obtained with 10 epochs since the value of validation accuracy becomes constant after the tenth epoch, as shown in Table 6.7. As a

consequence, with a 99.17%, the training process is stopped by employing early stopping criteria after the 10^{th} epoch.

Table 6.7: Training results attained by the proposed scheme on MICC-F2000 dataset

Epochs	Iteration	Training accuracy	Training loss	Validation accuracy	Validation loss	Base Learning rate
1	1	78.13%	0.4711	74.00%	0.5550	0.0001
2	50	90.63%	0.2606	94.33%	0.2551	0.0001
4	150	93.75%	0.1569	98.50%	0.1209	0.0001
6	250	94.68%	0.1243	98.67%	0.1083	0.0001
7	300	95.97%	0.1138	98.83%	0.0949	0.0001
10	400	96.88%	0.0687	99.17%	0.0504	0.0001
12	500	98.89%	0.0537	99.17%	0.0512	0.0001

The performance of the suggested scheme is also thoroughly examined using accuracy and loss plots for training as well as validation. If our model achieves considerably better on the training set than on the validation set, it is likely overfitting. Figure 6.6 shows these plots, which depict the model's training quality. The closely following training and validation curves for a model infer that it is neither underfitting nor overfitting, since the validation set performs better than the training set. This indicates that the model's generalizability on previously unseen data is improving with each successive step. The plots are obtained over the training epochs for datasets i.e. MICC-F220, GRIP, MICC-F2000, and IMD. Each graph has two curves, one for training and one for validation. The performance measures namely accuracy and loss are obtained for assessing the model's learning and generalization abilities simultaneously in the case of both training and validation sets. It is observed that initially, the accuracy curves show an inclining trend before reaching a constant value at a point of maximum performance. Likewise, validation loss and training loss in loss plots keep on decreasing attaining minimum value. At different epochs, the best performance of various datasets is achieved.

The best performance is conquered by 50 epochs for the MICC-F220 dataset, 47 epochs for the GRIP dataset, 30 epochs for the IMD, and 10 epochs for the MICC-F2000 dataset. Furthermore, all datasets, including the MICC-F220, MICC-F2000, GRIP, and IMD, have been integrated to form a large combined database. The benefit of merging several datasets goes beyond merely expanding dataset size to generalizing the evaluation procedure of the proposed approach and verifying overfitting.

Dataset used		Graphs
MICC-F220	(a1)	
GRIP	(a2)	
IMD	(a3)	
MICC-F2000	(a4)	

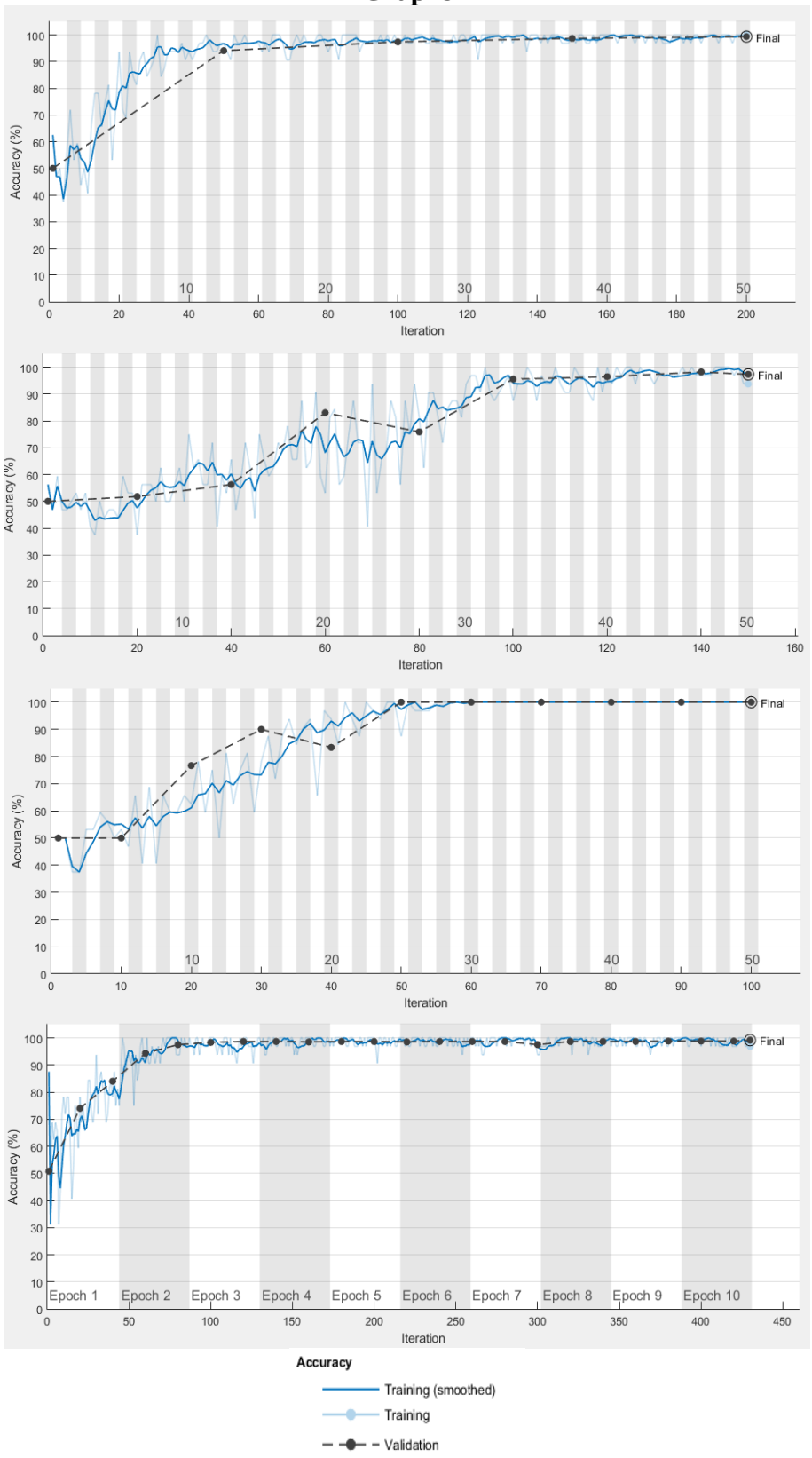


Figure 6.6: Graph of (a1~a4) accuracy v/s iterations and (b1~b4) loss v/s iterations obtained by proposed methodology on datasets: MICC-F220, GRIP, IMD, and MICC-F2000 (contd.)

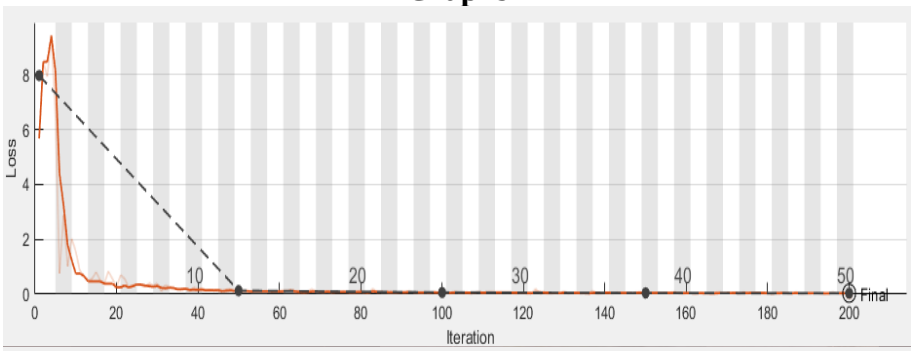
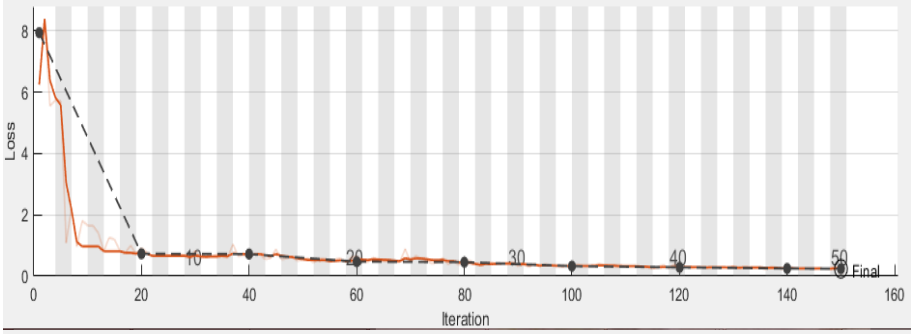
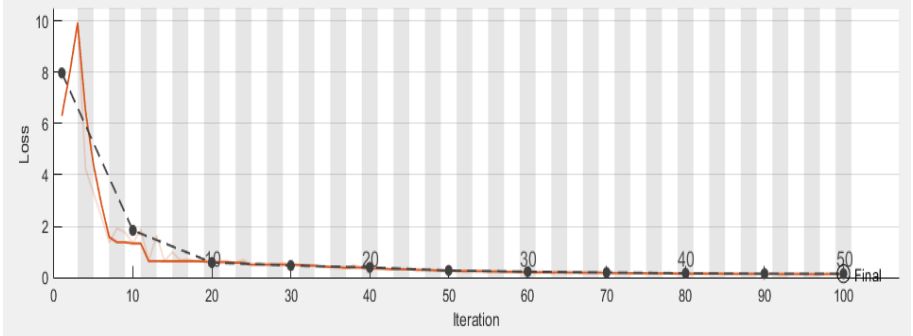
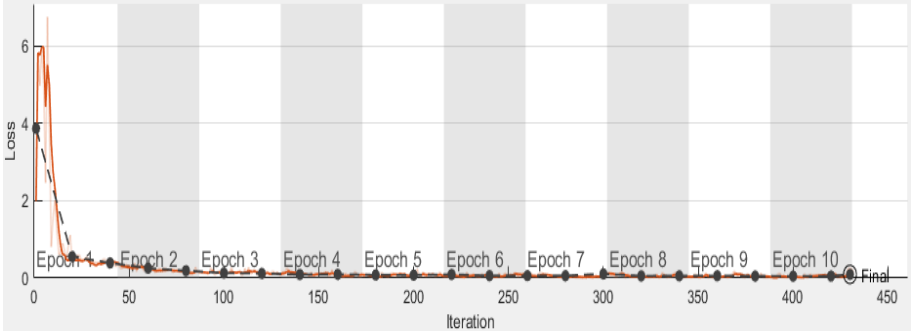
Dataset used		Graphs
MICC-F220	(b1)	
GRIP	(b2)	
IMD	(b3)	
MICC-F2000	(b4)	

Figure 6.6: Graph of (a1~a4) accuracy v/s iterations and (b1~b4) loss v/s iterations obtained by proposed methodology on datasets: MICC-F220, GRIP, IMD, and MICC-F2000

The combinational dataset contains 2476 images: 991 tampered and 1485 authentic, collected from datasets: MICC-F220, MICC-F2000, GRIP, and IMD. Table 6.8 depicts the training results on the combinational dataset. According to the Table, the combinational dataset

produced the best results at the 12th epoch because validation accuracy begins to decline after the 12th epoch. Thus, the training procedure is terminated based on the early stopping criteria. As a result, the suggested model is not overfitting, because overfitting happens when your network's training loss is considerably poorer than its validation loss, or when its training accuracy is considerably larger than its validation accuracy. However, in this scenario, neither the validation accuracy nor the validation loss is less than the training accuracy and training loss, respectively.

Table 6.8: Training results attained by the proposed scheme on the combinational dataset

Epochs	Iteration	Training accuracy	Training loss	Validation accuracy	Validation loss	Base Learning rate
1	1	25.00%	62.10%	11.956	6.0427	0.0001
2	50	65.63%	70.36%	0.6213	0.5609	0.0001
4	150	84.38%	86.10%	0.3758	0.3637	0.0001
6	250	90.63%	89.52%	0.2146	0.2394	0.0001
8	350	90.63%	93.63%	0.1735	0.1890	0.0001
10	450	93.88%	95.81%	0.1498	0.1357	0.0001
12	500	96.88%	98.76%	0.0958	0.0863	0.0001
14	600	96.88%	97.81%	0.0649	0.0549	0.0001

Further, some additional evaluation metrics like MCC, Geometric Mean (G-mean), precision-recall curve, Average Precision (AP), Mean Average Precision (MAP), ROC, and ROC Area under curve (AUC) have been evaluated. MCC denotes the classifier's correlation coefficient between predicted and actual classes. The minimum value of MCC indicates that a classifier labels all positives as negatives and all negatives as positives. In contrast, the maximum score of MCC indicates perfect classification and if the MCC value is close to zero, it indicates that the prediction is equivalent to random guessing. The geometric mean also referred to as the G-mean, is the geometric mean of TPR and TNR. It is a metric that compares classification performance in both classes. As a result, it is one of the evaluation measures for imbalanced classification. The precision-recall curve illustrates the balance between recall and precision at various thresholds. AP calculates an average value of precision $P(RR)$ over an interval from $RR = 0$ to $RR = 1$. In general, AP is the area under the precision-recall curve. MAP for a set of queries is the mean of average precision scores for each query (Q) i.e. it is the average of AP.

$$G-mean=\sqrt{TPR\times TNR} \tag{6.3.2}$$

$$AP=\int_{0}^{1}P(RR)dR \tag{6.3.3}$$

$$MAP=\frac{1}{Q}\sum_{a=1}^{Q}AP_a \tag{6.3.4}$$

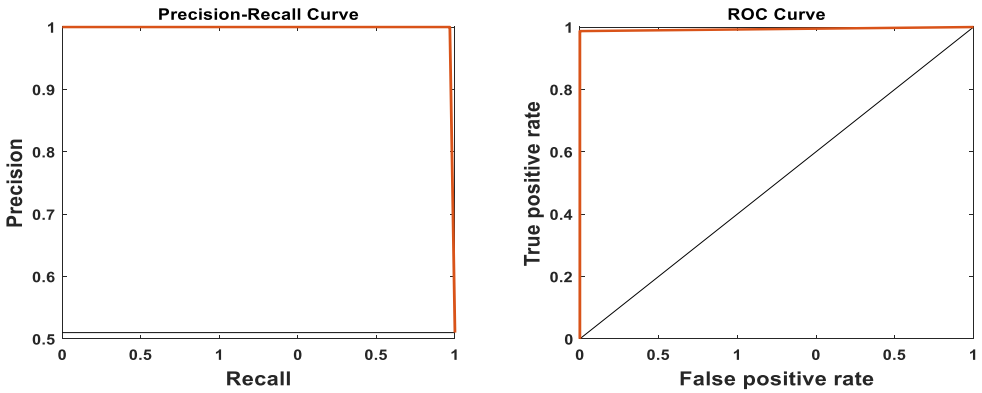
To visualize the classifier's performance, the ROC curve is plotted. ROC is calculated by plotting the fraction of TPR (on the y-axis) v/s FPR (on the x-axis). The AUC metric depicts the area under curve obtained by the ROC curve. The Precision-Recall curves of all datasets achieve a great AUC, indicating high recall and precision, with high precision corresponding to small FPR and high recall relating to a low FNR, as shown in Figure 6.7. High scores for both specify that the classifier is generating true results (high precision) and providing the majority of all positive outcomes (high recall). The Precision-Recall curve is a valuable metric for predicting the success of a model with imbalanced data. Furthermore, ROC curves for all datasets are close to the top left corner, indicating that the proposed model generates accurate results. Table 6.9 illustrates the value of various evaluation metrics for various datasets. The fact that almost all of the measures in the Table 6.9 approach one implies that the proposed scheme is a good classification model that can also be employed for imbalanced data.

Table 6.9: Various evaluation metrics for different datasets

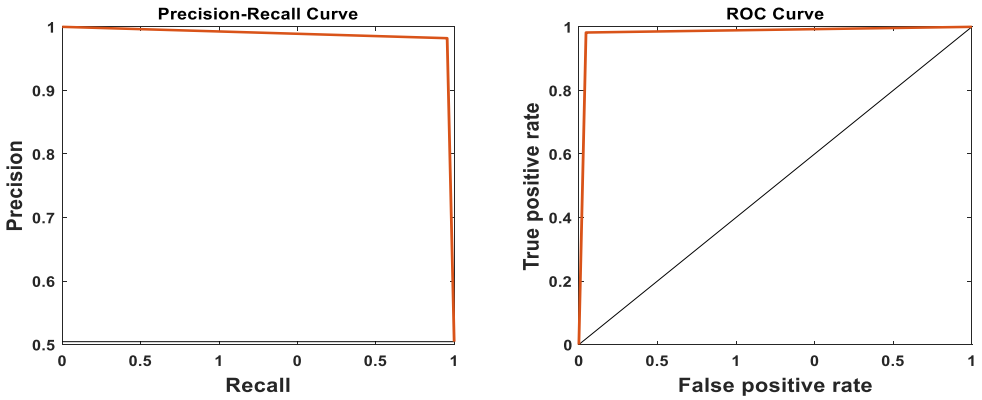
Dataset	MCC	G-mean	AP	AUC	MAP
MICC-F220	0.9868	0.9933	0.9968	0.9935	0.9968
GRIP	0.9640	0.9820	0.9866	0.9820	0.9866
IMD	1.0000	1.0000	1.0000	1.0000	1.0000
MICC-F2000	0.9692	0.9889	0.9913	0.9891	0.9913

A confusion matrix is a MJ×MJ matrix utilized to appraise the performance of the model, where MJ is the total target classes. The matrix matches the actual goal values to the machine learning model's predictions. This gives a clear picture of how well the classification model is fitting and what kinds of mistakes it is producing. So, the performance of the proposed approach is verified by confusion matrices for different datasets as revealed in Figure 6.8. To identify the features after 50 epochs, the graphs of accuracy v/s total iterations and loss v/s total iterations have been shown for 100 epochs.

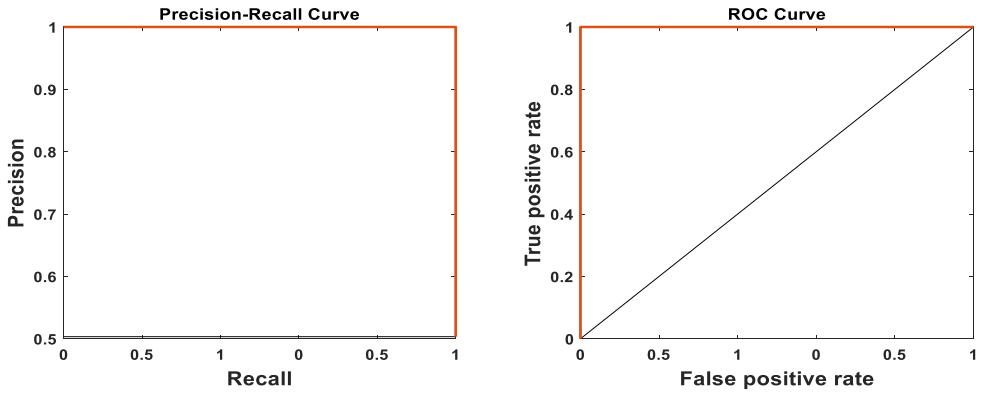
(A)



(B)



(C)



(D)

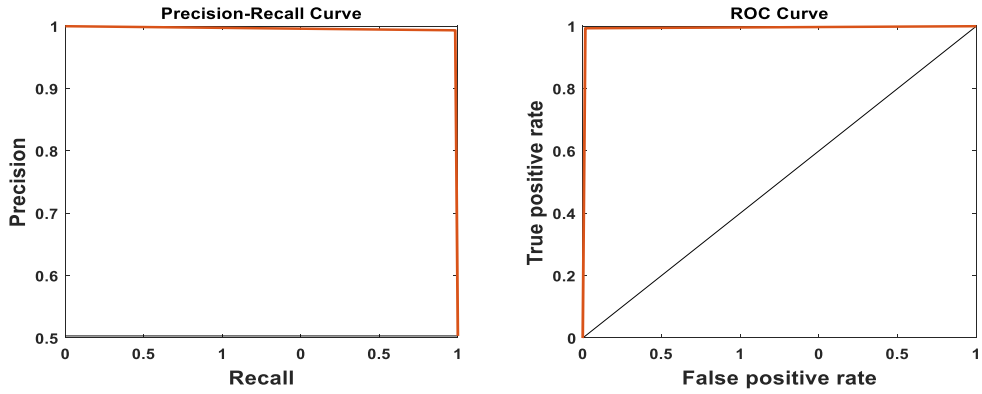


Figure 6.7: Precision-Recall curve and ROC curve for (A) MICC-F220 (B) GRIP (C) IMD and (D) MICC-F2000 datasets

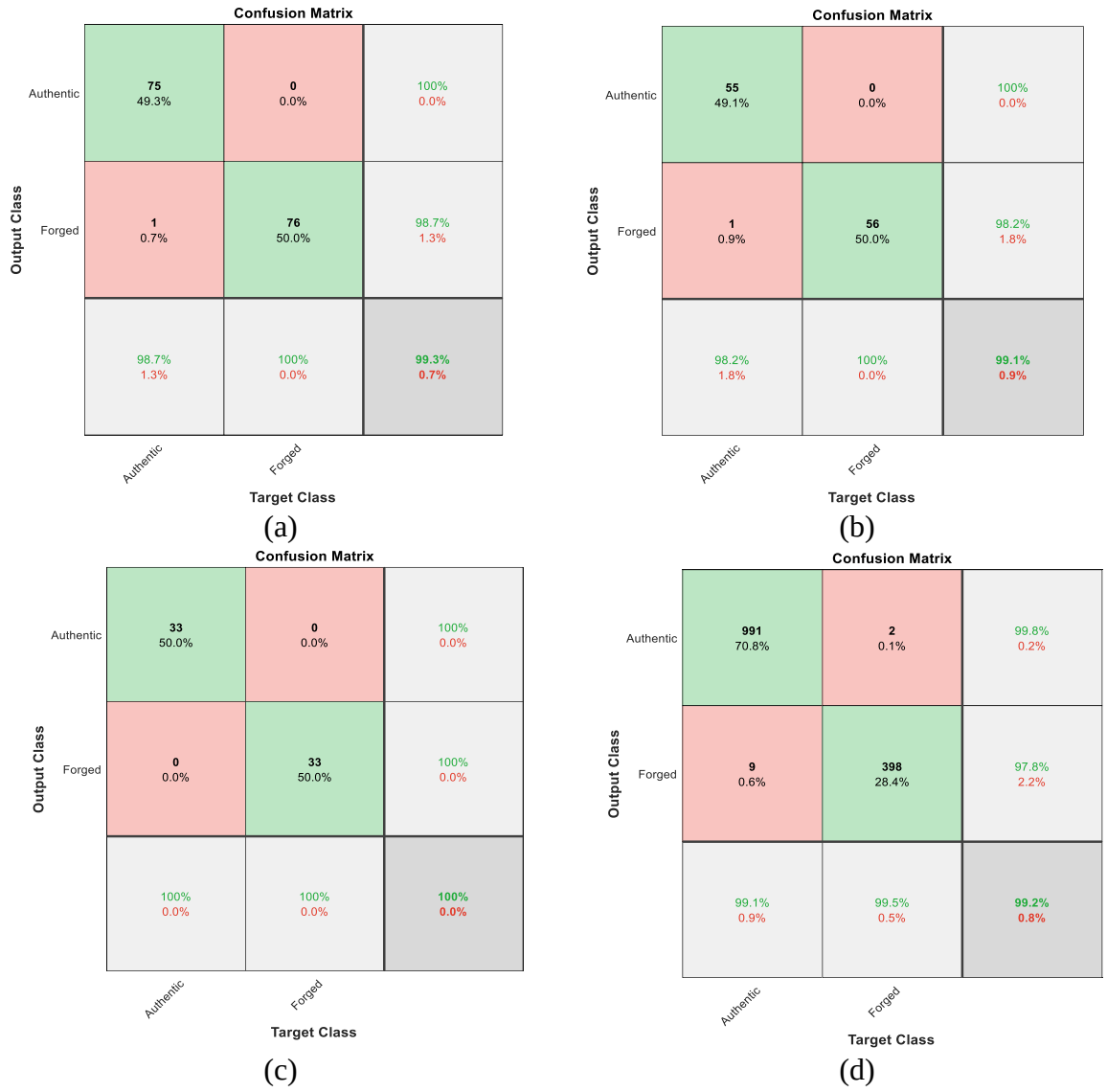


Figure 6.8: Confusion matrix for different datasets (a) MICC-F220 (b) GRIP (c) IMD and (d) MICC-F2000

Figure 6.9 depicts a graph of accuracy versus iterations across 100 epochs for four datasets: MICC-F220, GRIP, MICC-F2000, and IMD, with the x-axis representing iterations and the y-axis representing accuracy. The blue line designates training accuracy, whereas the black line signifies validation accuracy. In this, the training epoch is shown by a shaded backdrop. That is, the value of the tick mark right above the x-axis corresponds to the value of epochs. Similarly, Figure 6.10 shows a graph of loss and iterations, with the x-axis signifying iterations and the y-axis demonstrating loss. The red line in this figure represents the training loss, whereas the black line represents the validation loss. As shown in Figure 6.9 (a), the value of training and validation accuracy remains constant after 50 epochs, therefore proposed model employs an early stopping criterion to avoid overfitting. In Fig 6.9 (b), the best performance is reached at the 47th epoch. After the 47th epoch, validation accuracy drops and fluctuates.

(a)

(b)

(c)

(d)

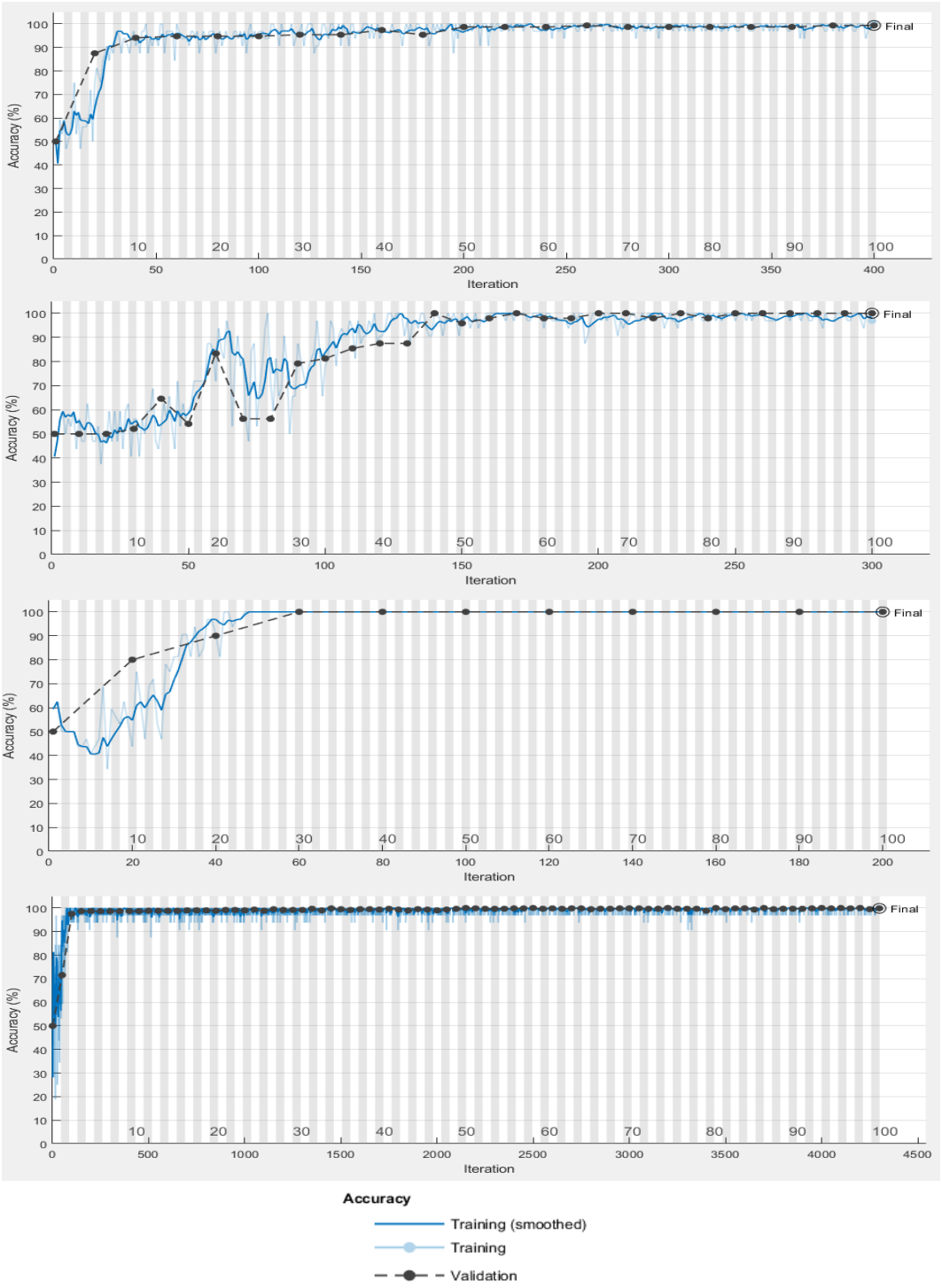


Figure 6.9: Graph between accuracy and iteration for 100 epochs on various datasets (a) MICC-F220, (b) GRIP, (c) IMD and (d) MICC-F2000

Similarly, in Figure 6.9 (c), the curve for accuracy v/s iterations shows that for the IMD dataset, both validation and training accuracy acquire a constant value after the 30^{th} epoch. The training plot for the MICC-F2000 dataset is shown in Figure 6.9 (d). For this dataset, the x-axis

represents iterations ranging from 0 to 4500, and it has been demonstrated that after the tenth epoch, the accuracy remains constant.

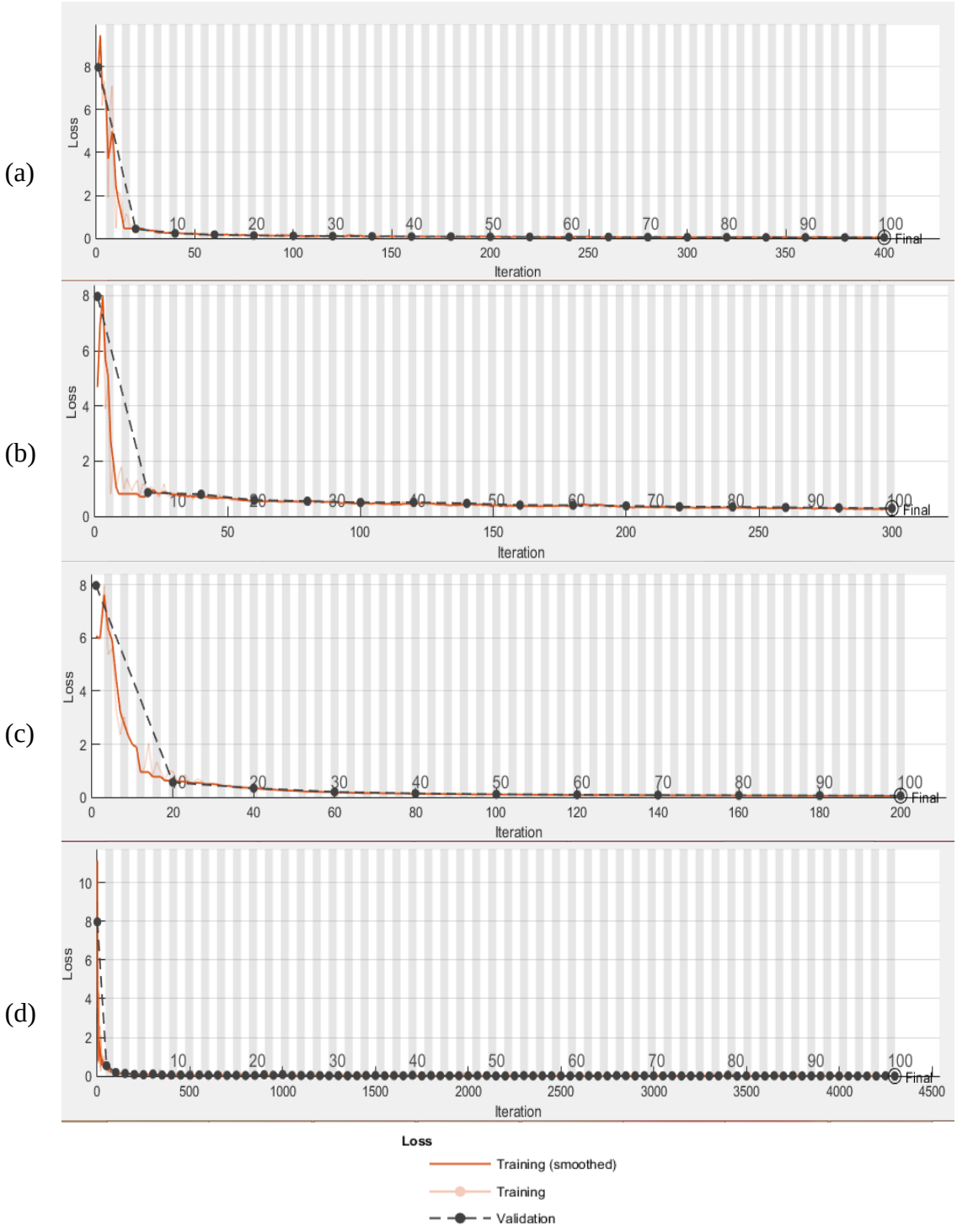


Figure 6.10: Graph between loss and iteration for 100 epochs on various datasets (a) MICC-F220, (b) GRIP, (c) IMD and (d) MICC-F2000

Furthermore, Figure 6.10 indicates that the validation and training losses remain constant or close to zero for the MICC-F220 and IMD datasets, but the loss decreases after the 50th epoch

for the GRIP dataset. Similarly, Figure 6.10 (d), which illustrates the plot of loss v/s accuracy for the MICC-F2000 dataset, shows that loss remains constant after the 10^{th} epoch for this dataset. As a result, for each dataset, the features have been identified after 50 epochs.

6.3.2 Comparative Analysis

This segment provides a comparative analysis of the suggested technique with several other techniques. The comparison of the suggested scheme is done with other techniques as shown in Table 6.10.

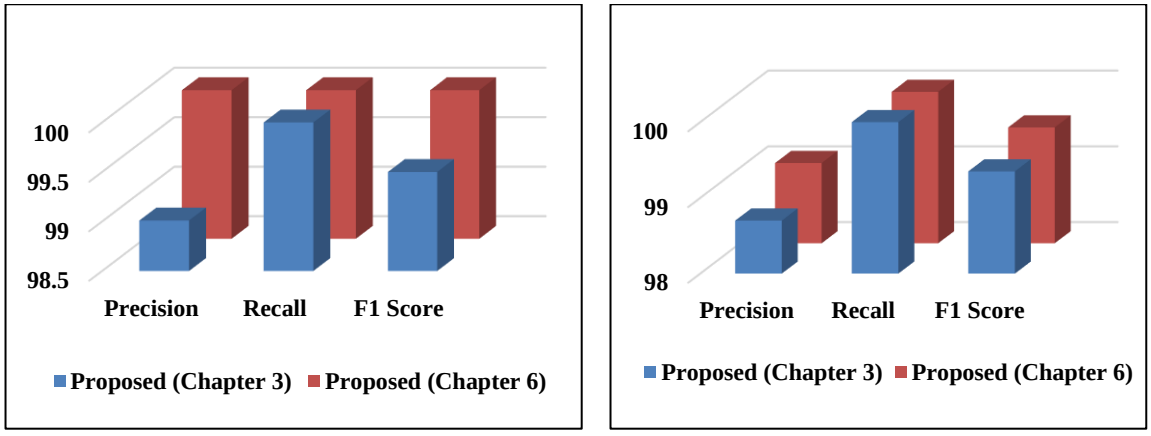
Table 6.10: Comparative analysis of proposed methodology with current methodologies

Dataset	Technique	Accuracy	Precision	TPR	TNR	F_1 Score	FNR	FPR
MICC-F220	Agarwal [125]	0.9500	0.9802	0.8958	0.9710	NE	0.0920	0.5500
	Amerini [150]	NE	NE	1.0000	0.9400	NE	0.0600	0.0000
	Elaskily [126]	NE	NE	1.0000	0.9820	NE	0.0000	1.8000
	Rodriguez [127]	0.9700	NE	NE	NE	0.9700	NE	NE
	Proposed	**0.9934**	**0.9870**	**1.0000**	**0.9868**	**0.9935**	**0.0132**	**0.0000**
GRIP	Christlein [55]	NE	0.7476	1.0000	NE	0.7368	NE	NE
	Cozzolino [80]	NE	0.9185	0.9875	NE	0.9518	NE	NE
	Wang [53]	NE	0.9176	0.9750	NE	0.9454	NE	NE
	Proposed	**0.9821**	**0.9821**	**0.9821**	**0.9818**	**0.9821**	**0.0179**	**0.0182**
IMD	Tinnathi [114]	NE	0.9452	0.9532	NE	0.9356	NE	NE
	Chen [103]	NE	0.9996	0.9859	NE	0.9924	NE	NE
	Meena [112]	NE	NE	NE	NE	0.9697	NE	NE
	Zhong [124]	NE	0.7085	0.5885	NE	0.6429	NE	NE
	Prakash [88]	NE	0.9230	0.8780	NE	0.8998	NE	NE
	Yang [52]	NE	0.9027	0.7861	NE	0.8404	NE	NE
	Bravo [42]	NE	0.9400	0.9792	NE	0.9592	NE	NE
	Pun [104]	NE	0.9592	0.9792	NE	0.9691	NE	NE
	Proposed	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.0000**	**0.0000**
MICC-F2000	Amerini [150]	NE	NE	0.9486	0.9085	NE	0.0514	0.0915
	Elaskily [126]	NE	NE	0.9840	0.9365	NE	0.0160	0.0635
	Amerini [48]	NE	NE	0.9342	0.8839	NE	0.0658	0.1161
	Proposed	**0.9871**	**0.9830**	**0.9990**	**0.9591**	**0.9909**	**0.0010**	**0.0409**

NE: Not evaluated by the respective author and Bold indicates the optimum values

For MICC-F220 dataset, 99.34% accuracy, 98.70% precision, 100% TPR, 98.68% TNR, and 99.35% F_1 score is achieved. The GRIP dataset attains an accuracy of 98.20%, a precision of 98.21%, TPR of 98.21%, 98.18% TNR, and 98.21% F_1 score. Also, accuracy, precision, TPR, and TNR of 100% are achieved for the IMD dataset. For MICC-F2000 dataset, 98.71% accuracy, 98.30% precision, 99.90% TPR, 95.91% TNR, and 99.09% F_1 score is achieved. The results in Table 6.10 illustrate that the suggested deep learning-based CMFD procedure surpasses traditional CMFD algorithms in terms of several performance parameters alike TPR, F_1 score, accuracy, precision, TNR. Furthermore, a lower FNR and FPR value demonstrates that the suggested procedure is effective for CMFD.

Furthermore, as shown in Figure 6.11, the results of Chapter 3 are compared to the proposed results on IMD, and MICC-F220 datasets. It has been observed that deep learning-based CMFD achieves superior results when compared to conventional CMFD algorithms used in Chapter 3, demonstrating the utility of deep learning in image forgery detection.



(a) IMD (b) MICC-F220

Figure 6.11: Comparison of various performance parameters (%) on different datasets

6.3.3 Run-Time Analysis

Run-time analysis or computational latency is a metric used in deep learning to assess the performance of various models in a given application. In general, run-time analysis is the time it takes to process one unit of data, assuming only one unit of data is processed at a time. The time it takes to categorize images as authentic or forged is specified as computational latency in the proposed system. Because the input images are resized, the image input size for all datasets is $50_{\times}50$. As a result, the computational latency of the proposed methodology is influenced by the total images in the dataset. Table 6.11 shows the computational latency for three datasets: MICC-F220, GRIP, and IMD.

Table 6.11: Run-time analysis for various datasets

Epochs	Time Elapsed (hh:mm:ss)		
	MICC-F220	**GRIP**	**IMD**
1	00:00:19	00:00:15	00:00:15
10	00:03:18	00:02:33	00:01:45
20	00:06:16	00:05:25	00:03:21
30	00:08:47	00:08:44	00:04:47
40	00:11:44	00:10:20	00:06:12
50	00:13:54	00:12:44	00:07:34

According to the Table, the MICC-F220 dataset takes longer to process than the other two datasets since it contains more images than the other two. IMD, on the other hand, has lower computational latency due to a lesser number of images. Furthermore, optimum performance for the MICC-F2000 dataset is achieved with 10 epochs in 29 minutes and 47 seconds, hence the process was terminated after the 10^{th} epoch. As a result, it is not included in the table.

6.4 Robustness Under Various Attacks

Also, to confirm the efficacy of the suggested procedure, it is evaluated under various attacks. Invariance is a characteristic of CNN that permits it to recognize objects even when positioned in different orientations. A CNN may be invariant to scaling, rotation, or compression, or flipping. This is the fundamental concept of data augmentation. A dataset of images acquired under specific conditions is present in the real-world scenario. On the other hand, the target application of the proposed model may exist under a variety of conditions, such as different orientations, scale, compression, noise addition, flipping, and so on. In this scenario, tampered photos are formed by exploiting pictures of the IMD dataset created by Christlein [55], and the duplicated parts are assaulted by transforms like scaling, noise addition, rotation, and JPEG compression obtaining a significant augmentation. Moreover, data augmentation can help in increasing the amount of relevant data in a dataset.

(a) Rotation: The rotation attack conceals the tampering evidence, which makes the detection process more challenging. In this scenario, copied portions are rotated in steps of 2^o with a rotation angle ranging from 2^o to 10^o. Further, large rotation angles like 20^o, $60^{o,}$ and 180^o have been included and the experiment is done on $48 \times 8 = 384$ pictures.

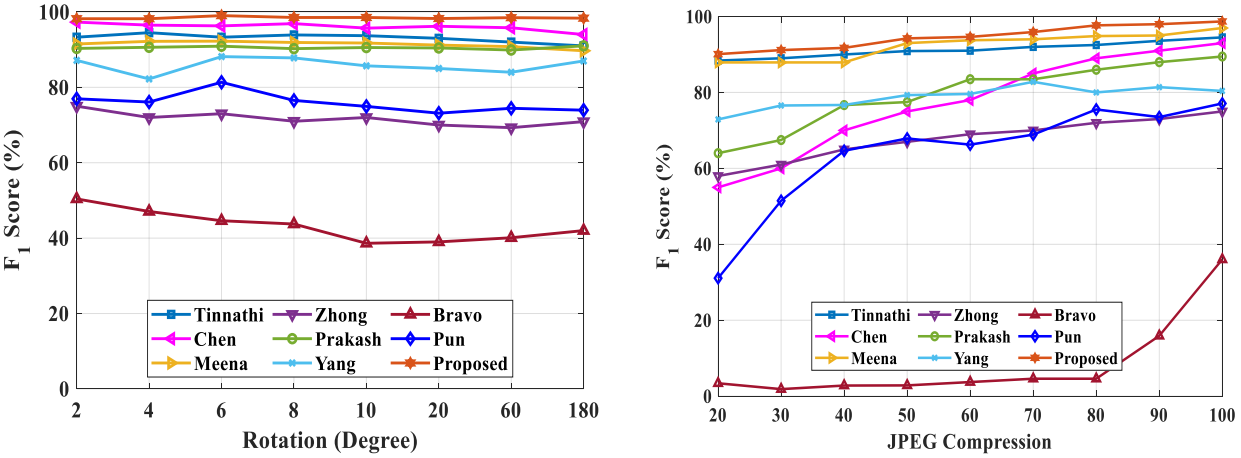
(b) Scaling: In image processing, an invariant is a property of the image (a function in this context) that does not change or only slightly changes when the image is transformed (rotated,

scaled, etc.). A basic CNN taught deprived of data augmentation can only recognize the characteristics on a scale it was taught on; such a CNN does not deal with scale-invariant characteristics at other scales. A simple way to overcome stated restrictions is to reveal CNN to numerous scales through training. It is often used as a data augmentation method to enhance training data and hence diminish overfitting. The scaling procedure also adds in hiding tampering evidence. Because scaling generally results in some pixel loss, it makes detection more difficult. Consequently, the performance of suggested and current detection techniques is evaluated under scaling assault. The copied portions are rescaled among 91% and 109%, using a 2% step size. The trial is employed on $48 \times 10 = 480$ photos in this scenario.

(c) JPEG compression: Image compression is quite frequent in everyday life. Most images on the Internet are compressed, and it is also a handy way to hide tampering traces. Thus, a comparison experiment using a JPEG compression assault is conducted in this research. The tampered pictures are compressed via QF ranging from 20 to 100 by 10 step size. The experiment is executed on $48 \times 9 = 432$ pictures in this scenario.

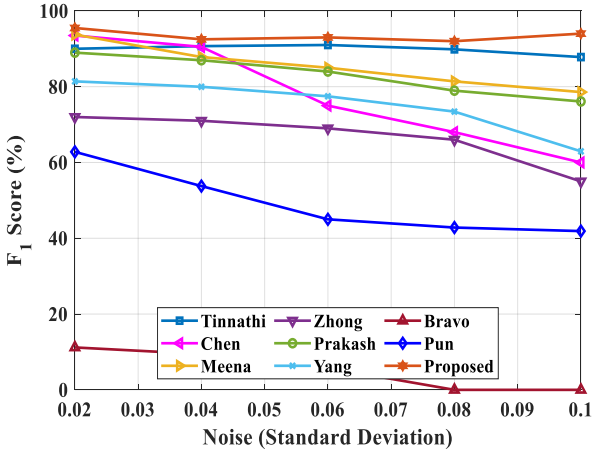
(d) Noise addition: Adding noise to a tampered image is a common approach for concealing the evidence of image manipulations. Consequently, a good procedure must be resistant to noise assaults. Herein, the cloned regions are added with Gaussian noise by standard deviation in a range between 0.02 and 0.1 in 0.02 increments. A trial is implemented on $48 \times 5 = 240$ pictures in this instance.

The outcomes of the F_1 score with scaling, noise addition, rotation, and JPEG compression attack are displayed and compared in Figure 6.12 with other techniques alike Tinnathi [114], Chen [103], Meena [112], Zhong [124], Prakash [88], Yang [52], Bravo [42], and Pun [104]. In the graphs below, the x-axis represents the rotation angle (varying from 2^o to 10^o), scaling factor (ranging from 91% and 109%), various quality levels of JPEG compression (varying from 20 to 100), and values of standard deviations of the noise (ranging from 0.02 to 0.1). The y-axis signifies the F_1 score's value in form of a percentage. The results labeled as 'Proposed' exhibit the outcomes of the suggested detection procedure. This experimentation illustrates the proposed scheme's robustness to scaling, noise addition, rotation, and JPEG compression attack since the value of the F_1 score of the proposed approach outperforms other existing approaches for these geometrical attacks. Also, graphs for other performance metrics can be plotted in the same way.

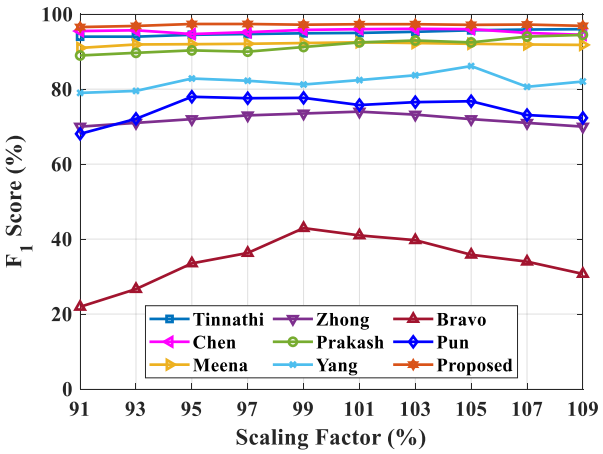


(i) Rotation

(ii) JPEG compression



(iii) Noise addition



(iv) Scaling

Figure 6.12: Comparison analysis under different attacks

Furthermore, data augmentation techniques such as horizontal and vertical flips are used to diminish bias, attain improved generality, and compensate for lesser training data. An augmentation approach is used to generate a transformed picture $I(e', f')$ for each picture $I(e, f)$ in the training set. The horizontal flip randomly flips the input picture across its vertical (left to right) axis with a predefined frequency. Its formula is as follows:

$$\begin{bmatrix} e' \\ f' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} e \\ f \\ 1 \end{bmatrix} \tag{6.4.1}$$

$$e' = -e, f' = f \tag{6.4.2}$$

The vertical flip randomly flips the input picture across its horizontal (top to bottom) axis with a predefined frequency. Its formula is as follow:

$$\begin{bmatrix} e' \\ f' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} e \\ f \\ 1 \end{bmatrix} \tag{6.4.3}$$

$$e' = f, e' = -f \tag{6.4.4}$$

Figure 6.13 depicts an example of image flipping. In the proposed model, images are flipped both horizontally and vertically, and then simulations are run on all of the images in the dataset.

Input image Horizontal Flip Vertical Flip

Figure 6.13: Illustration of image flipping (horizontal and vertical)

Figure 6.14 depicts the values of several performance criteria reached by horizontal and vertical flipped images. The figure shows that the proposed model achieves a significant amount of various performance metrics. The high value of metrics such as precision, recall, TPR, TNR, G-mean, accuracy, MCC, F_1 score, and F_2 score, and low value of metrics such as FPR, and FNR indicate that the proposed model is effective at detecting flipped images. As a result, it is concluded that the proposed CLAHE-based CNN model can detect fabricated objects even when they are positioned in different orientations.

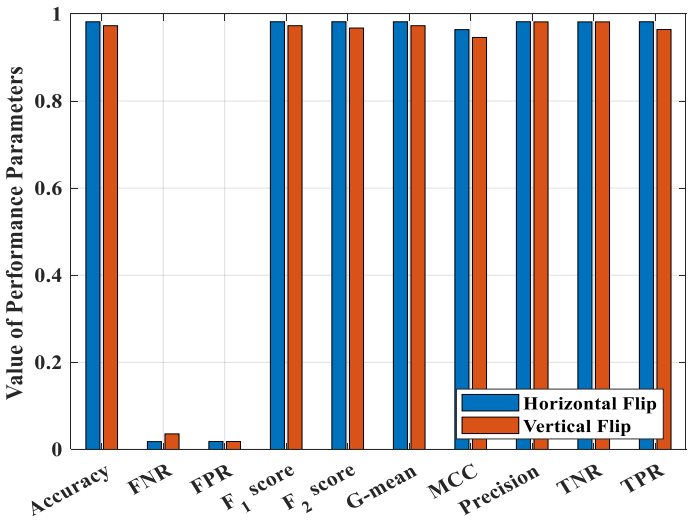
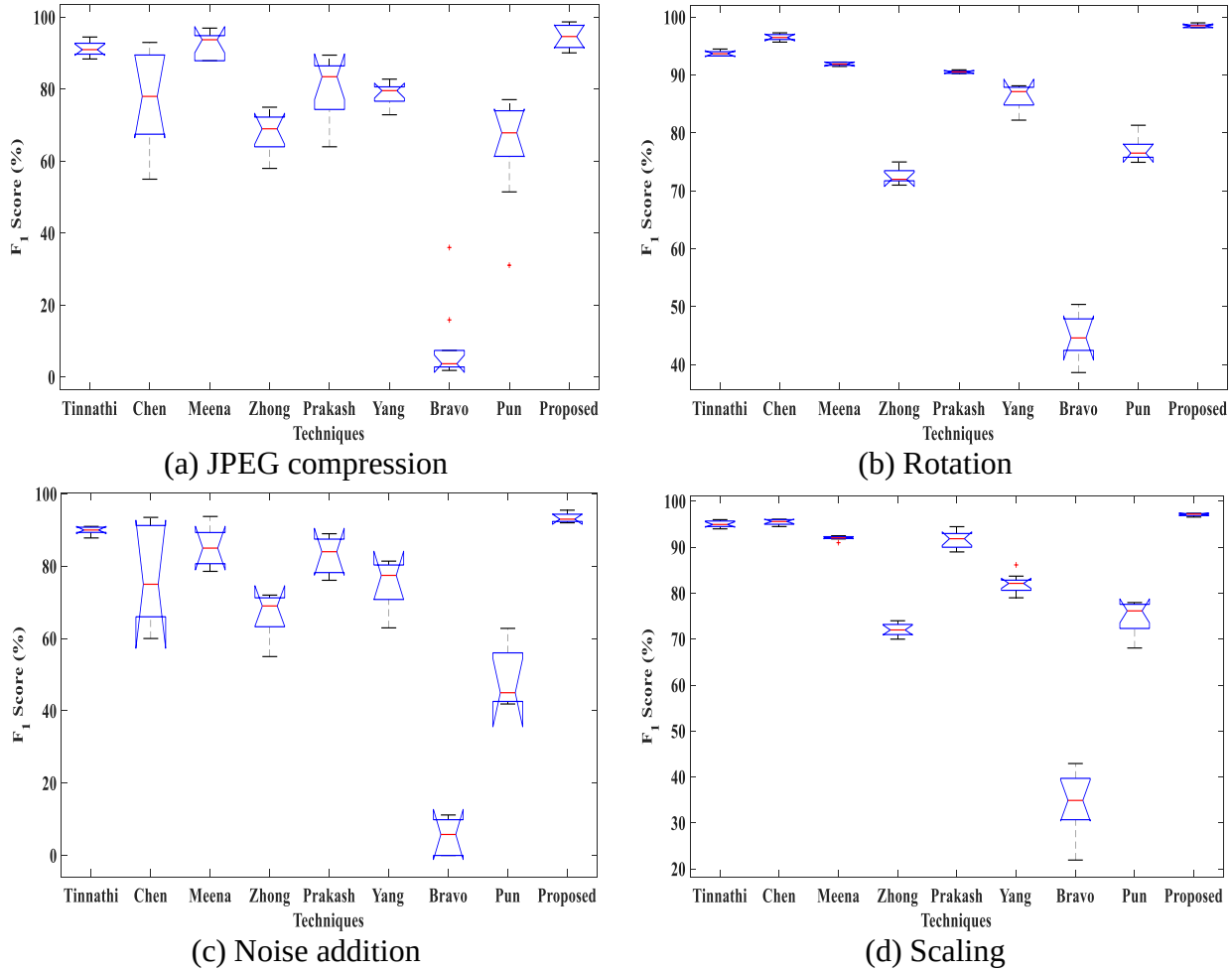


Figure 6.14: Value of different performance parameters attained by horizontal and vertical flipped images

6.5 Statistical Analysis

In the previous section, it has been discovered that the suggested approach surpasses the current techniques. However, the statistical importance of the detected modification in the performance of various methodologies must be confirmed, as it provides a critical vision in assessing the reliability of data. Therefore, to compare the methodologies, the ANOVA test is employed. The ANOVA is applied to the consequences of the suggested methodology and current methodologies under several attacks. Figure 6.15 depicts a statistical examination of the F_1 score for various methodologies against various attacks. According to box plots in Figure 6.15, the whiskers of the suggested method are superior to other current methods as it approaches 100. Furthermore, as the notches of two separate box plots don't overlap, the median values of the compared procedures are considerably diverse. As a consequence, it signifies that the differences between distinct techniques are statistically significant. Thus, the data representation in Figure 6.15 shows that the current approaches are substantially weaker than the suggested approach.



(a) JPEG compression (b) Rotation

(c) Noise addition (d) Scaling

Figure 6.15: Statistical examination of F_1 score for various procedures under several attacks

6.6 Cross-Dataset Performance

Extensive cross-dataset evaluations are carried out to correctly determine the suggested model's generalization capacity. So, the cross-dataset performance is evaluated in this section, which is a key field of study and an important component in real-world applications where diverse pictures must be classified. Table 6.12 displays the cross-dataset performance for numerous performance measures for four datasets, namely IMD, MICC-F220, MICC-F2000, and GRIP. According to the table, the suggested approach performed better, which demonstrates that the suggested approach has some generality to images from various sources and sizes.

Table 6.12: Cross-dataset evaluation of the proposed approach

Training dataset	Testing dataset	Precision	Recall	F_1 score	F_2 score
IMD	GRIP	96.44	98.41	97.41	98.01
	MICC-F220	91.87	95.00	93.41	94.36
	MICC-F2000	94.68	97.05	95.85	96.56
GRIP	IMD	94.91	97.50	96.19	96.97
	MICC-F220	89.89	92.95	91.40	92.33
	MICC-F2000	93.61	96.59	95.08	95.98
MICC-F220	IMD	90.31	93.18	91.72	92.59
	GRIP	88.62	92.05	90.30	91.34
	MICC-F2000	96.66	98.64	97.64	98.23
MICC-F2000	IMD	92.73	95.68	94.18	95.08
	GRIP	91.89	95.23	93.53	94.54
	MICC-F220	95.80	98.41	97.09	97.88

6.7 Summary

In this chapter, CLAHE based CNN approach is employed to detect CMF. The fundamental role of this study is the expansion of the CNN model for classifying input images into two categories: authentic and forged. Initially, image pre-processing is achieved by using image resizing and the CLAHE algorithm. Also, features of the image are extracted by the CNN model, and feature maps are created from them. The average of output feature maps is taken by CNN and further, it explores feature dependencies automatically. Post-training, a CNN model, is prepared to test and categorize images for detecting CMF. The proposed approach is tested on a variety of datasets like MICC-F220, GRIP, MICC-F2000, and IMD. The simulation

results illustrate that the finest performance is attained with 50 epochs, 47 epochs, 40 epochs, and 10 epochs for MICC-F220, GRIP, IMD, and MICC-F2000 datasets, respectively. In regard of TPR, accuracy, TNR, precision, FNR, F_1 score, and FPR, the proposed deep learning-based method effectively outperforms the existing techniques.

CHAPTER 7

CONCLUSIONS AND FUTURE SCOPE

This chapter is dedicated to providing conclusions based on the research work done in this thesis in the fields of digital image forgery detection techniques. It also gives the main highlights and future prospects of the research work.

7.1 Conclusions

Digital images are a fundamental way of communication in the age of digital computing. The modification of images has become easier due to the availability of tools for the editing of images particularly Adobe Photoshop, GIMP, and others. The uniqueness of images is important since they are extensively used to provide evidence in a variety of applications mainly medical imaging, law enforcement, criminal investigation, forensic analysis, journalistic photography, and insurance claims. So, IFDTs are necessary for checking if digital pictures are reliable or not, they are divided into two categories: active and passive. Active techniques; digital watermarking and digital signatures, rely on the pictures' preceding information. Because preceding information about the images is not always accessible, passive techniques can be used to determine the authenticity of photographs. CMF and ISF are the two most popular forms of passive image forgeries.

In the presented work, keypoint-based techniques (AKAZE, SIFT) and block-based approaches (AS) are combined, resulting in an algorithm having more computational efficiency and accuracy to detect and localize single and multiple CMFs. The performance of the suggested scheme is implemented on benchmark datasets i.e. MICC-F220, IMD, COVERAGE, and GRIP with an improved F_2 score of 99.81%, 99.80%, 99.35%, and 99.82%, respectively, which are better than the conventional methods. Also, the performance of localized single and multiple CMF images is assessed quantitatively. Furthermore, the robustness of the suggested scheme is validated against several attacks, and in contrast to existing methodologies, the simulation results show an improved average F_1 score of 2.35% for rotation attack, 1.52% for scaling attack, 1.31% for JPEG compression attack, and 1.95% for noise addition attack. Moreover, the robustness of the proposed approach is validated by performing the cross-dataset performance and statistical analysis test like ANOVA.

In this research work, the extraction of Markov attributes is carried out from DWT and LBP domains and merged to efficiently detect ISF. The experimental findings using six benchmark datasets, namely DSO-1, CASIA v1.0, Columbia, CASIA v2.0, DVMM, and IFS-TC, demonstrate that the proposed strategy outperforms current approaches on respective datasets with increased accuracy of 92.50%, 99.69%, 98.61%, 99.76%, 97.80%, and 96.90%. Moreover, the suggested model performs well across datasets, demonstrating that it has some applicability to photos of diverse sources and sizes. The simulation results demonstrate that the presented ISFD method outperforms existing approaches under various attacks, with improvements in average F_1 score of 5%, 6.18%, 3.21%, and 12.56% for scaling, noise addition, JPEG compression, and rotation, respectively. ANOVA is also performed to validate the effectiveness of the suggested technique.

Further research in this thesis proposes a hybrid technique based on DFrCT and LBP to identify CMF and ISF at the same time. The DFrCT's fractional parameter magnifies accuracy, while LBP is employed to identify tampering artifacts effectively. Additionally, the duplicated parts of the picture are localized on both the spliced as well as copy-move pictures. The proposed scheme's efficacy is authenticated by executing comprehensive tests on six datasets, i.e., GRIP, CASIA v1.0, IMD, CASIA v2.0, Columbia, and COVERAGE which surpasses the current approaches by accuracy rates of 99.23%, 99.67%, 98.81%, 99.76%, 98.17%, and 95%, respectively. Furthermore, the robustness of the presented scheme is examined under different attacks. In comparison to current methodologies, the simulation results show an improved average F_1 score of 7.94 % for rotation attack, 4.96 % for scaling attack, 14.12 % for JPEG compression attack, and 5.77% for noise addition attack. Cross-dataset performance and statistical analysis tests such as AVOVA are also conducted to verify the effectiveness of the suggested work.

This research presents a CLAHE-based CNN model for effectively solving the issue of CMF detection. The CLAHE algorithm makes the hidden features of the image visible, as some of them are hard to detect in CMF. The effectiveness of the proposed structure is appraised using benchmark datasets: GRIP, MICC-F2000, IMD, and MICC-F220. In terms of various performance metrics, the experimental study demonstrates the efficacy of the presented technique, among other approaches. Further, the simulation results illustrate the proposed scheme's robustness to scaling, noise addition, rotation, and JPEG compression attack since

the value of the F_1 score of the proposed approach outperforms other existing approaches with the progress of 1.52%, 2.13%, 2.27%, and 1.24%, respectively.

7.2 Main Highlights of the Research Work

The key highlights of the presented research work in the field of digital image forgery detection methods are as follows:

- An improved methodology has been developed to effectively detect single and multiple CMFs by combining keypoint-based methodologies, i.e., SIFT and AKAZE, and block-based techniques, i.e., AS. This combination produces an effective framework by outperforming existing procedures with an improved F_1 score of 99.81%, 99.80%, 99.35%, and 99.82% on benchmark datasets, i.e., MICC-F220, IMD, COVERAGE, and GRIP.
- In addition, to detect ISF, a passive forgery detection mechanism based on extracting and combining Markov characteristics from the LBP and DWT domains is proposed.
- Also, to identify CMF and ISF simultaneously, a passive hybrid technique based on DFrCT and LBP is presented. Moreover, the tampered regions in both forgeries are spotted by localizing the forged images.
- Furthermore, a deep learning system based on CLAHE and CNN is presented to detect CMF.
- The suggested work's robustness is also confirmed against several attacks, such as JPEG compression, scaling, rotation, and noise addition, and better results are obtained than the previous approaches.
- The performance of suggested approaches is validated by evaluating several parameters like precision, recall, accuracy, F_1 score, F_2 score, MCC, ROC curve, informedness, markedness, and statistical analysis test like ANOVA.
- The presented schemes are further examined on cross-datasets (training and testing on different datasets) to see how well it generalizes to new information in practical scenarios.

The novelty of the proposed work is that it provides improved results for passive image forgery detection in digital images regardless of whether they have been exposed to different attacks like JPEG compression, scaling, rotation, and noise addition. Furthermore, the presented work will be beneficial to the research community working in the field of classification, object

detection, and localization applications such as medical image analysis, face detection, and so on.

7.3 Future Scope

The research work in this thesis can be expanded in the following directions for further explorations in this field:

- Based on the findings in the experimental results section, a forgery detection approach that is robust to various other attacks such as blurring, down sampling, cropping, brightness alteration, contrast adjustment, and so on can be developed in the future.
- The proposed techniques can also be applied to other object detection and localization applications, such as medical image analysis, face detection, and so on.
- Deep learning could be used in the future to locate the forged portion in tampered images.
- In the future, it may be expanded to detect video tampering.

REFERENCES

[1] Pearson, "A Picture is Worth a Thousand Words – Meaning, Origin and Usage," *English-Grammar-Lessons.com*. https://english-grammar-lessons.com/a-picture-is-worth-a-thousand-words-meaning/ (accessed Apr. 20, 2022).

[2] J. A. Redi, W. Taktak, and J. Dugelay, "Digital image forensics : a booklet for beginners," *Multimed. Tools Appl.*, vol. 51, no. 1, pp. 133–162, 2011, doi: 10.1007/s11042-010-0620-1.

[3] S. Sadeghi, S. Dadkhah, H. A. Jalab, G. Mazzola, and D. Uliyan, "State of the art in passive digital image forgery detection : copy-move image forgery," *Pattern Anal. Appl.*, vol. 21, no. 2, pp. 291–306, 2018, doi: 10.1007/s10044-017-0678-8.

[4] A. Singh and P. K. Jain, "A compartive study of SVD and ICA for target detection in through-the-wall radar images," in *Proc. Int. Conf. Ind. Inf. Syst.*, 2016, pp. 608–613, doi: 10.1109/ICIINFS.2016.8263011.

[5] A. M. M. Shabat and J. R. Tapamo, "Angled local directional pattern for texture analysis with an application to facial expression recognition," *IET Comput. Vis.*, vol. 12, no. 5, pp. 603–608, 2018, doi: 10.1049/iet-cvi.2017.0340.

[6] N. Rawat, M. Singh, and B. Singh, "Wavelet and Total Variation Based Method Using Adaptive Regularization for Speckle Noise Reduction in Ultrasound Images," *Wirel. Pers. Commun.*, vol. 106, no. 3, pp. 1547–1572, 2019, doi: 10.1007/s11277-019-06229-w.

[7] J. Liu and X. Yuan, "Obscure bleeding detection in endoscopy images using support vector machines," *Optim. Eng.*, vol. 10, no. 2, pp. 289–299, 2009, doi: 10.1007/s11081-008-9066-y.

[8] J. Qi and R. M. Leahy, "Iterative reconstruction techniques in emission computed tomography," *Phys. Med. Biol.*, vol. 51, no. 15, pp. 541–578, 2006, doi: 10.1088/0031-9155/51/15/R01.

[9] S. Walia and K. Kumar, "Digital image forgery detection: a systematic scrutiny," *Aust. J. Forensic Sci.*, vol. 51, no. 5, pp. 488–526, 2019, doi: 10.1080/00450618.2018.1424241.

[10] Z. Zhang, C. Wang, and X. Zhou, "A survey on passive image copy-move forgery detection," *J. Inf. Process. Syst.*, vol. 14, no. 1, pp. 6–31, 2018, doi: 10.3745/JIPS.02.0078.

[11] P. Singh, B. Raman, N. Agarwal, and P. K. Atrey, “Secure cloud-based image tampering detection and localization using POB number system,” *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 13, no. 3, 2017, doi: 10.1145/3077140.

[12] Neeraj, M. S. Goraya, and D. Singh, “A comparative analysis of prominently used MCDM methods in cloud environment,” *J. Supercomput.*, vol. 77, no. 4, pp. 3422–3449, 2021, doi: 10.1007/s11227-020-03393-w.

[13] M. S. Goraya and L. Kaur, “Fault tolerance task execution through cooperative computing in grid,” *Parallel Process. Lett.*, vol. 23, no. 1, pp. 1–20, 2013, doi: 10.1142/S0129626413500035.

[14] A. Verma and S. Kaushal, “Cost-Time Efficient Scheduling Plan for Executing Workflows in the Cloud,” *J. Grid Comput.*, vol. 13, no. 4, pp. 495–506, 2015, doi: 10.1007/s10723-015-9344-9.

[15] A. Keivani, F. Ghayoor, and J. R. Tapamo, “A review of recent methods of task scheduling in cloud computing,” in *Proc. IEEE Mediterr. Electrotech. Conf.*, 2018, pp. 104–109, doi: 10.1109/MELCON.2018.8379076.

[16] K. Smith, “126 Amazing Social Media Statistics and Factsitle,” *brandwatch.com*. https://www.brandwatch.com/blog/amazing-social-media-statistics-and-facts/ (accessed Dec. 30, 2019).

[17] J. McDonald, “Social Media Posts Spread Bogus Coronavirus Conspiracy Theory,” *factcheck.org*. https://www.factcheck.org/2020/01/social-media-posts-spread-bogus-coronavirus-conspiracy-theory/ (accessed Jan. 24, 2020).

[18] M. Sapir, “The Impossible Photograph: Hippolyte Bayard’s Self-Portrait as a Drowned Man,” *MFS Mod. Fict. Stud.*, vol. 40, no. 3, pp. 619–629, 1994, doi: 10.1353/mfs.1994.0007.

[19] H. Farid, “Digital image forensics,” *Sci. Am.*, vol. 298, no. 6, pp. 66–71, 2008, doi: 10.1038/scientificamerican0608-66.

[20] Ii. Ben-Meir, “Anti-Iran Deal TV Ad Uses Fake Image Of Obama Meeting Iranian President,” *buzzfeednews.com*. https://www.buzzfeednews.com/article/ilanbenmeir/anti-iran-deal-tv-ad-uses-fake-image-of-obama-meeting-irania (accessed Jul. 23, 2015).

[21] P. House, “California Attorney Photoshops Herself With Celebrities In Effort To Fool Public,” *fstoppers.com*. https://fstoppers.com/humor/california-attorney-photoshops-herself-celebrities-effort-fool-public-38081 (accessed Sep. 21, 2014).

[22] A. Kumar, “Lalu tweets image of packed RJD rally in Patna, rivals challenge his claim,” *Hindustan Times*. https://www.hindustantimes.com/india-news/did-lalu-prasad-post-a-

morphed-image-of-rjd-s-patna-rally-twitter-is-confused/story-jPJJZ4rwA3PlobuSfKghJK.html. (accessed Aug. 27, 2017).

[23] M. Reith, C. Carr, and G. Gunsch, "An Examination of Digital Forensic Models," *Int. J. Digit. Evid.*, vol. 1, no. 3, pp. 1–12, 2002.

[24] Monika, D. Bansal, and A. Passi, "Image Forensic Investigation Using Discrete Cosine Transform-Based Approach," *Wirel. Pers. Commun.*, vol. 119, no. 4, pp. 3241–3253, 2021, doi: 10.1007/s11277-021-08396-1.

[25] P. S. Raskar and S. K. Shah, "VFDHSOG: Copy-Move Video Forgery Detection Using Histogram of Second Order Gradients," *Wirel. Pers. Commun.*, vol. 122, no. 2, pp. 1617–1654, 2022, doi: 10.1007/s11277-021-08964-5.

[26] S. Jia, Z. Xu, H. A. O. Wang, C. Feng, and T. A. O. Wang, "Coarse-to-fine Copy-move Forgery Detection for Video Forensics," *IEEE Access*, vol. 6, pp. 25323–25335, 2018, doi: 10.1109/ACCESS.2017.

[27] N. Kanwal, A. Girdhar, L. Kaur, and J. S. Bhullar, "Digital image splicing detection technique using optimal threshold based local ternary pattern," *Multimed. Tools Appl.*, vol. 79, no. 19, pp. 12829–12846, 2020, doi: 10.1007/s11042-020-08621-2.

[28] M. Imran, S. Tah, Z. Ali, S. T. Bakhsh, and S. Akram, "Blind Detection of Copy-Move Forgery in Digital Audio Forensics," *IEEE Access*, vol. 5, pp. 12843–12855, 2017, doi: 10.1109/ACCESS.2017.2717842.

[29] Z. Zhang, Y. Ren, X. Ping, Z.-Y. He, and S.-Z. Zhang, "A survey on passive-blind image forgery by doctor method detection," in *Proc. Int.Conf. Mach. Learn. Cybern.*, 2008, pp. 12–15, doi: 10.1109/ICMLC.2008.4621003.

[30] V. Vinolin and M. Sucharitha, "Hierarchical Categorization and Review of Recent Techniques on Image Forgery Detection," *Comput. J.*, vol. 64, no. 11, pp. 1692–704, 2021, doi: 10.1093/comjnl/bxz148.

[31] R. Ahuja and S. S. Bedi, "Video watermarking scheme based on IDR frames using MPEG-2 structure," *Int. J. Inf. Comput. Secur.*, vol. 11, no. 6, pp. 585–603, 2019, doi: 10.1504/IJICS.2019.103065.

[32] G. Cao, Y. Zhao, R. Ni, and X. Li, "Contrast enhancement-based forensics in digital images," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 3, pp. 515–525, 2014, doi: 10.1109/TIFS.2014.2300937.

[33] R. Ahuja and S. S. Bedi, "Robust video watermarking scheme based on intra-coding process in MPEG-2 style," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 6, pp. 3332–3343, 2017, doi: 10.11591/ijece.v7i6.pp3332-3343.

[34] Y. Guo, B. Z. Li, and N. Goel, “Optimised blind image watermarking method based on firefly algorithm in DWT-QR transform domain,” *IET Image Process.*, vol. 11, no. 6, pp. 406–415, 2017, doi: 10.1049/iet-ipr.2016.0515.

[35] B. Z. Li and Y. P. Shi, “Image watermarking in the linear canonical transform domain,” *Math. Probl. Eng.*, vol. 2014, 2014, doi: 10.1155/2014/645059.

[36] H. Farid, “Image Forgery Detection,” *IEEE Signal Process. Mag.*, vol. 26, no. 2, pp. 16–25, 2009, doi: 10.1109/MSP.2008.931079.

[37] A. K. Lamba, N. Jindal, and S. Sharma, “Digital image copy-move forgery detection based on discrete fractional wavelet transform,” *Turkish J. Electr. Eng. Comput. Sci.*, vol. 26, no. 3, pp. 1261–1277, 2018, doi: 10.3906/elk-1701-275.

[38] S. Gupta, N. Mohan, and P. Kaushal, “Passive image forensics using universal techniques : a review,” *Artif. Intell. Rev.*, vol. 55, pp. 1629–1679, 2022, doi: 10.1007/s10462-021-10046-8.

[39] F. Y. Shih, *Digital Watermarking and Steganography Fundamentals and Techniques*. Boca Raton: CRC Press, 2017.

[40] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*. New Delhi: McGraw Hill Education, 2010.

[41] S. Bravo-Solorio and A. K. Nandi, “Passive forensic method for detecting duplicated regions affected by reflection, rotation and scaling,” in *Proc. Eur. Sig. Process. Conf.*, 2009, pp. 824–828.

[42] S. Bravo-Solorio and A. K. Nandi, “Exposing duplicated regions affected by reflection, rotation and scaling,” in *Proc. Int. Conf. Acoust. Speech Sig. Process.*, 2011, pp. 1880–1883, doi: 10.1109/ICASSP.2011.5946873.

[43] J. Zhao and J. Guo, “Passive forensics for copy-move image forgery using a method based on DCT and SVD,” *Forensic Sci. Int.*, vol. 233, no. 1–3, pp. 158–166, 2013, doi: 10.1016/j.forsciint.2013.09.013.

[44] O. M. Al-Qershi and B. E. Khoo, “Enhanced block-based copy-move forgery detection using k-means clustering,” *Multidimens. Syst. Signal Process.*, vol. 30, no. 4, pp. 1671–1695, 2019, doi: 10.1007/s11045-018-0624-y.

[45] G. Gani and F. Qadir, “Copy move forgery detection using DCT, PatchMatch and cellular automata,” *Multimed. Tools Appl.*, vol. 80, no. 21, pp. 32219–32243, 2021, doi: 10.1007/s11042-021-11174-7.

[46] J. Ouyang, Y. Liu, and M. Liao, “Robust copy-move forgery detection method using pyramid model and Zernike moments,” *Multimed. Tools Appl.*, vol. 78, no. 8, pp. 10207–

10225, 2019, doi: 10.1007/s11042-018-6605-1.

[47] X. Pan and S. Lyu, “Region duplication detection using image feature matching,” *IEEE Trans. Inf. Forensics Secur.*, vol. 5, no. 4, pp. 857–867, 2010, doi: 10.1109/TIFS.2010.2078506.

[48] I. Amerini, L. Ballan, S. Member, 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, pp. 1099–1110, 2011, doi: 10.1109/TIFS.2011.2129512.

[49] Y. Li and J. Zhou, “Fast and Effective Image Copy-Move Forgery Detection via Hierarchical Feature Point Matching,” *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 5, pp. 1307–1322, 2019, doi: 10.1109/TIFS.2018.2876837.

[50] J. Yang, Z. Liang, Y. Gan, and J. Zhong, “A novel copy-move forgery detection algorithm via two-stage filtering,” *Digit. Signal Process.*, vol. 113, pp. 103032–103048, 2021, doi: 10.1016/j.dsp.2021.103032.

[51] Priyanka, G. Singh, and K. Singh, “An improved block based copy-move forgery detection technique,” *Multimed. Tools Appl.*, vol. 79, no. 19, pp. 13011–13035, 2020, doi: 10.1007/s11042-019-08354-x.

[52] F. Yang, J. Li, W. Lu, and J. Weng, “Copy-move forgery detection based on hybrid features,” *Eng. Appl. Artif. Intell.*, vol. 59, no. 2017, pp. 73–83, 2017, doi: https://doi.org/10.1016/j.engappai.2016.12.022.

[53] C. Wang, Z. Zhang, and X. Zhou, “An Image Copy-Move Forgery Detection Scheme Based on A-KAZE and SURF Features,” *Symmetry (Basel).*, vol. 10, no. 12, pp. 706–726, 2018, doi: 10.3390/sym10120706.

[54] G. Tahaoglu, G. Ulutas, B. Ustubioglu, and V. V. Nabiyev, “Improved copy move forgery detection method via L*a*b* color space and enhanced localization technique,” *Multimed. Tools Appl.*, vol. 80, no. 15, pp. 23419–23456, 2021, doi: 10.1007/s11042-020-10241-9.

[55] V. Christlein *et al.*, “An Evaluation of Popular Copy-Move Forgery Detection Approaches,” *IEEE Trans. Inf. forensics Secur.*, vol. 7, no. 6, pp. 1841–1854, 2012, doi: 10.1109/TIFS.2012.2218597.

[56] J. A and P. K, “Object Recognition Based on Lbp and Discrete Wavelet Transform,” *Int. J. Adv. Signal Image Sci.*, vol. 2, no. 1, p. 24, 2016, doi: 10.29284/ijasis.2.1.2016.24-30.

[57] M. A. Muqeet and R. S. Holambe, “Local binary patterns based on directional wavelet

transform for expression and pose-invariant face recognition," *Appl. Comput. Informatics*, vol. 15, no. 2, pp. 163–171, 2019, doi: 10.1016/j.aci.2017.11.002.

[58] P. Srivastava and A. Khare, "Integration of wavelet transform, Local Binary Patterns and moments for content-based image retrieval," *J. Vis. Commun. Image Represent.*, vol. 42, pp. 78–103, 2017, doi: 10.1016/j.jvcir.2016.11.008.

[59] M. Kaur and S. Gupta, "A passive blind approach for image splicing detection based on DWT and LBP histograms," in *Proc. Int. Symp. Secur. Comput. Commun.*, 2016, vol. 625, pp. 318–327, doi: 10.1007/978-981-10-2738-3_27.

[60] F. Hakimi, M. Hariri, and F. GharehBaghi, "Image splicing forgery detection using local binary pattern and discrete wavelet transform," in *Proc. Int. Conf. Knowl. Based Eng. Innov.*, 2015, pp. 1–4, doi: 10.1109/KBEI.2015.7436195.

[61] Q. Zhang, W. Lu, R. Wang, and G. Li, "Digital image splicing detection based on Markov features in block DWT domain," *Multimed. Tools Appl.*, vol. 77, no. 23, pp. 31239–31260, 2018, doi: 10.1007/s11042-018-6230-z.

[62] B. Su, Q. Yuan, S. Wang, C. Zhao, and S. Li, "Enhanced state selection Markov model for image splicing detection," *Eurasip J. Wirel. Commun. Netw.*, vol. 2014, no. 7, pp. 1–10, 2014, doi: https://doi.org/10.1186/1687-1499-2014-7.

[63] Q. Zhang, W. Lu, and J. Weng, "Joint image splicing detection in DCT and Contourlet transform domain," *J. Vis. Commun. Image Represent.*, vol. 40, pp. 449–458, 2016, doi: 10.1016/j.jvcir.2016.07.013.

[64] X. Zhao, S. Wang, S. Li, and J. Li, "Passive image-splicing detection by a 2-D noncausal markov model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 2, pp. 185–199, 2015, doi: 10.1109/TCSVT.2014.2347513.

[65] E. M. El-alfy and M. A. Qureshi, "Robust content authentication of gray and color images using lbp-dct markov-based features," *Multimed. Tools Appl.*, vol. 76, no. 12, pp. 14535–14556, 2017, doi: 10.1007/s11042-016-3855-7.

[66] C. Li, Q. Ma, L. Xiao, M. Li, and A. Zhang, "Image splicing detection based on Markov features in QDCT domain," *Neurocomputing*, vol. 228, pp. 29–36, 2017, doi: 10.1016/j.neucom.2016.04.068.

[67] Z. He, W. Lu, W. Sun, and J. Huang, "Digital image splicing detection based on Markov features in DCT and DWT domain," *Pattern Recognit.*, vol. 45, no. 12, pp. 4292–4299, 2012, doi: 10.1016/j.patcog.2012.05.014.

[68] G. Muhammad, M. H. Al-Hammadi, M. Hussain, and G. Bebis, "Image forgery detection using steerable pyramid transform and local binary pattern," *Mach. Vis. Appl.*,

vol. 25, no. 4, pp. 985–995, 2014, doi: 10.1007/s00138-013-0547-4.

[69] B. Liu, C. M. Pun, and X. C. Yuan, "Digital image forgery detection using JPEG features and local noise discrepancies," *Sci. World J.*, vol. 2014, pp. 1–12, 2014, doi: 10.1155/2014/230425.

[70] C. S. Prakash, A. Kumar, S. Maheshkar, and V. Maheshkar, "An integrated method of copy-move and splicing for image forgery detection," *Multimed. Tools Appl.*, vol. 77, no. 20, pp. 26939–26963, 2018, doi: 10.1007/s11042-018-5899-3.

[71] S. P. Jaiprakash, M. B. Desai, C. S. Prakash, V. H. Mistry, and K. L. Radadiya, "Low dimensional DCT and DWT feature based model for detection of image splicing and copy-move forgery," *Multimed. Tools Appl.*, vol. 79, no. 39, pp. 29977–30005, 2020, doi: 10.1007/s11042-020-09415-2.

[72] S. Dua, J. Singh, and H. Parthasarathy, "Detection and localization of forgery using statistics of DCT and Fourier," *Signal Process. Image Commun.*, vol. 82, pp. 115778–115796, 2020, doi: 10.1016/j.image.2020.115778.

[73] N. T. Pham and J. Lee, "Structural Correlation Based Method for Image Forgery Classification and Localization," *Appl. Sci.*, vol. 10, no. 13, pp. 4458–4475, 2020, doi: 10.3390/app10134458.

[74] J. Fridrich, D. Soukal, and J. Lukáš, "Detection of copy-move forgery in digital images," in *Proc. Digit Forensic Res. Work.*, 2003, pp. 55–61.

[75] A. C. Popescu and H. Farid, "Exposing Digital Forgeries by Detecting Duplicated Image Regions," Dept. Comput. Sci., Dartmouth Coll. Tech. Rep., Hanover NH, 2004. doi: 10.1109/TSP.2004.839932.

[76] G. Li, W. Qiong, T. Dan, and S. Shaojie, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD," in *Proc. IEEE Int. Conf. Multimed. Expo.*, 2007, pp. 1750–1753, doi: 10.1109/ICME.2007.4285009.

[77] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 886–902, 1998, doi: 10.1109/78.668544.

[78] G. Muhammad, M. Hussain, and G. Bebis, "Passive copy move image forgery detection using undecimated dyadic wavelet transform," *Digit. Investig.*, vol. 9, no. 1, pp. 49–57, 2012, doi: 10.1016/j.diin.2012.04.004.

[79] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy-move forgery detection scheme," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 3, pp. 507–518, 2015, doi: 10.1109/TIFS.2014.2381872.

[80] D. Cozzolino, G. Poggi, and L. Verdoliva, “Efficient dense-field copy-move forgery detection,” *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 11, pp. 2284–2297, 2015, doi: 10.1109/TIFS.2015.2455334.

[81] J. Zhong and Y. Gan, “Detection of copy-move forgery using discrete analytical Fourier-Mellin transform,” *Nonlinear Dyn.*, vol. 84, no. 1, pp. 189–202, 2016, doi: 10.1007/s11071-015-2374-9.

[82] M. Emam, Q. Han, and X. Niu, “PCET based copy-move forgery detection in images under geometric transforms,” *Multimed. Tools Appl.*, vol. 75, no. 18, pp. 11513–11527, 2016, doi: 10.1007/s11042-015-2872-2.

[83] 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, pp. 202–214, 2018, doi: 10.1016/j.jvcir.2018.03.015.

[84] K. B. Meena and V. Tyagi, “A copy-move image forgery detection technique based on tetrolet transform,” *J. Inf. Secur. Appl.*, vol. 52, pp. 102481–102490, 2020, doi: 10.1016/j.jisa.2020.102481.

[85] H. Kasban and S. Nassar, “An efficient approach for forgery detection in digital images using Hilbert–Huang transform,” *Appl. Soft Comput. J.*, vol. 97, pp. 106728–106745, 2020, doi: 10.1016/j.asoc.2020.106728.

[86] B. Ahmed, T. A. Gulliver, and S. AlZahir, “Blind copy-move forgery detection using SVD and KS test,” *SN Appl. Sci.*, vol. 2, no. 8, pp. 1–12, 2020, doi: 10.1007/s42452-020-3181-6.

[87] S. B. G. T. Babu and C. S. Rao, “An optimized technique for copy-move forgery localization using statistical features,” *ICT Express*, vol. 8, no. 2, pp. 244–249, 2022, doi: 10.1016/j.icte.2021.08.016.

[88] C. S. Prakash, S. Maheshkar, P. Pralhad, and P. Hari, “Detection of copy-move forgery using AKAZE and SIFT keypoint extraction,” *Multimed. Tools Appl.*, vol. 78, no. 16, pp. 23535–23558, 2019, doi: 10.1007/s11042-019-7629-x.

[89] D.G.Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004, doi: 10.1023/b:visi.0000029664.99615.94.

[90] J. Zhao and W. Zhao, “Passive forensics for region duplication image forgery based on harris feature points and local binary patterns,” *Math. Probl. Eng.*, vol. 2013, pp. 1–12, 2013, doi: 10.1155/2013/619564.

[91] M. Jaberi, G. Bebis, M. Hussain, and G. Muhammad, “Accurate and robust localization

of duplicated region in copy-move image forgery," *Mach. Vis. Appl.*, vol. 25, no. 2, pp. 451–475, 2014, doi: 10.1007/s00138-013-0522-0.

[92] L. Yu, Q. Han, and X. Niu, "Feature point-based copy-move forgery detection: covering the non-textured areas," *Multimed. Tools Appl.*, vol. 75, no. 2, pp. 1159–1176, 2016, doi: 10.1007/s11042-014-2362-y Feature.

[93] X. Wang, S. Li, Y. Liu, Y. Niu, H.-Y. Yang, and Z. Zhou, "A new keypoint-based copy-move forgery detection for small smooth regions," *Multimed. Tools Appl.*, vol. 76, no. 22, pp. 23353–23382, 2016.

[94] H. A. Alberry, A. A. Hegazy, and G. I. Salama, "A fast SIFT based method for copy move forgery detection," *Futur. Comput. Informatics J.*, vol. 3, no. 2, pp. 159–165, 2018, doi: 10.1016/j.fcij.2018.03.001.

[95] X. Y. Wang, L. X. Jiao, X. B. Wang, H. Y. Yang, and P. P. Niu, "A new keypoint-based copy-move forgery detection for color image," *Appl. Intell.*, vol. 48, no. 10, pp. 3630–3652, 2018, doi: 10.1007/s10489-018-1168-4.

[96] M. Bilal, H. A. Habib, Z. Mehmood, T. Saba, and M. Rashid, "Single and Multiple Copy-Move Forgery Detection and Localization in Digital Images Based on the Sparsely Encoded Distinctive Features and DBSCAN Clustering," *Arab. J. Sci. Eng.*, vol. 45, no. 4, pp. 2975–2992, 2020, doi: 10.1007/s13369-019-04238-2.

[97] S. Dhivya, J. Sangeetha, and B. Sudhakar, "Copy-move forgery detection using SURF feature extraction and SVM supervised learning technique," *Soft Comput.*, vol. 24, pp. 14429–14440, 2020, doi: 10.1007/s00500-020-04795-x.

[98] H. Yang, S. Qi, Y. Niu, P. Niu, and X. Wang, "Copy-move forgery detection based on adaptive keypoints extraction and matching," *Multimed. Tools Appl.*, vol. 78, no. 24, pp. 34585–34612, 2019, doi: 10.1007/s11042-019-08169-w.

[99] K. Liu *et al.*, "Copy move forgery detection based on keypoint and patch match," *Multimed. Tools Appl.*, vol. 78, no. 22, pp. 31387–31413, 2019, doi: 10.1007/s11042-019-07930-5.

[100] X. Y. Wang, C. Wang, L. Wang, L. X. Jiao, H. Y. Yang, and P. P. Niu, "A fast and high accurate image copy-move forgery detection approach," *Multidimens. Syst. Signal Process.*, vol. 31, no. 3, pp. 857–883, 2020, doi: 10.1007/s11045-019-00688-x.

[101] S. Uma and P. D. Sathya, "Copy-move forgery detection of digital images using football game optimization," *Aust. J. Forensic Sci.*, vol. 54, no. 2, pp. 258–279, 2022, doi: 10.1080/00450618.2020.1811376.

[102] P. Niu, C. Wang, W. Chen, H. Yang, and X. Wang, "Fast and effective Keypoint-based

image copy-move forgery detection using complex-valued moment invariants," *J. Vis. Commun. Image Represent.*, vol. 77, pp. 103068–103085, 2021, doi: 10.1016/j.jvcir.2021.103068.

[103] H. Chen, X. Yang, and Y. Lyu, "Copy-Move Forgery Detection Based on Keypoint Clustering and Similar Neighborhood Search Algorithm," *IEEE Access*, vol. 8, pp. 36863–36875, 2020, doi: 10.1109/ACCESS.2020.2974804.

[104] 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, doi: 10.1109/TIFS.2015.2423261.

[105] J. Zheng, Y. Liu, J. Ren, T. Zhu, Y. Yan, and H. Yang, "Fusion of block and keypoints based approaches for effective copy-move image forgery detection," *Multidimens. Syst. Signal Process.*, vol. 27, no. 4, pp. 989–1005, 2016, doi: 10.1007/s11045-016-0416-1.

[106] Y. Sun, R. Ni, and Y. Zhao, "Nonoverlapping Blocks Based Copy-Move Forgery Detection," *Secur. Commun. Networks*, vol. 2018, pp. 1–12, 2018, doi: 10.1155/2018/1301290.

[107] J. A. Ojeniyi, "Hybridized Technique for Copy-Move Forgery Detection Using Discrete Cosine Transform and Speeded-Up Robust Feature Techniques," *I. J. Image, Graph. Signal Process.*, vol. 10, no. 4, pp. 22–30, 2018, doi: 10.5815/ijigsp.2018.04.03.

[108] H. Huang and A. Ciou, "Copy-move forgery detection for image forensics using the superpixel segmentation and the Helmert transformation," *EURASIP J. image video Process.*, vol. 2019, no. 1, pp. 1–16, 2019, doi: 10.1186/s13640-019-0469-9.

[109] B. Elhaminia, A. Harati, and A. Taherinia, "A probabilistic framework for copy-move forgery detection based on Markov Random Field," *Multimed. Tools Appl.*, vol. 78, no. 18, pp. 25591–25609, 2019, doi: 10.1007/s11042-019-7713-2.

[110] Y. Liu, H. Wang, Y. Chen, H. Wu, and H. Wang, "A passive forensic scheme for copy-move forgery based on superpixel segmentation and K-means clustering," *Multimed. Tools Appl.*, vol. 79, no. 1–2, pp. 477–500, 2020, doi: 10.1007/s11042-019-08044-8.

[111] P. Niyishaka and C. Bhagvati, "Copy-move forgery detection using image blobs and BRISK feature," *Multimed. Tools Appl.*, vol. 79, no. 35, pp. 26045–26059, 2020, doi: 10.1007/s11042-020-09225-6.

[112] K. B. Meena and V. Tyagi, "A hybrid copy-move image forgery detection technique based on Fourier-Mellin and scale invariant feature transforms," *Multimed. Tools Appl.*, vol. 79, no. 11, pp. 8197–8212, 2020, doi: 10.1007/s11042-019-08343-0.

[113] R. Agarwal and O. P. Verma, "Robust copy-move forgery detection using modified

superpixel based FCM clustering with emperor penguin optimization and block feature matching," *Evol. Syst.*, vol. 13, no. 1, pp. 27–41, 2022, doi: 10.1007/s12530-021-09367-4.

[114] S. Tinnathi and G. Sudhavani, "An efficient copy move forgery detection using adaptive watershed segmentation with AGSO and hybrid feature extraction," *J. Vis. Commun. Image Represent.*, vol. 74, pp. 102966–102979, 2021, doi: 10.1016/j.jvcir.2020.102966.

[115] A. Sharma *et al.*, "IoT and deep learning-inspired multi-model framework for monitoring Active Fire Locations in Agricultural Activities," *Comput. Electr. Eng.*, vol. 93, pp. 107216–107235, 2021, doi: 10.1016/j.compeleceng.2021.107216.

[116] L. Boussaad and A. Boucetta, "Deep-learning based descriptors in application to aging problem in face recognition," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 1–7, 2020, doi: 10.1016/j.jksuci.2020.10.002.

[117] X. Ning, K. Gong, W. Li, L. Zhang, X. Bai, and S. Tian, "Feature Refinement and Filter Network for Person Re-Identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3391–3402, 2021, doi: 10.1109/TCSVT.2020.3043026.

[118] R. D. Singh, A. Mittal, and R. K. Bhatia, "3D convolutional neural network for object recognition: a review," *Multimed. Tools Appl.*, vol. 78, no. 12, pp. 15951–15995, 2019, doi: 10.1007/s11042-018-6912-6.

[119] S. Wen, Z. Zeng, T. Huang, Q. Meng, and W. Yao, "Lag synchronization of switched neural networks via neural activation function and applications in image encryption," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 26, no. 7, pp. 1493–1502, 2015, doi: 10.1109/TNNLS.2014.2387355.

[120] Y. Liu, X. Yuan, X. Gong, Z. Xie, F. Fang, and Z. Luo, "Conditional convolution neural network enhanced random forest for facial expression recognition," *Pattern Recognit.*, vol. 84, pp. 251–261, 2018, doi: 10.1016/j.patcog.2018.07.016.

[121] F. M. Al_Azrak *et al.*, "An efficient method for image forgery detection based on trigonometric transforms and deep learning," *Multimed. Tools Appl.*, vol. 79, no. 25, pp. 18221–18243, 2020, doi: 10.1007/s11042-019-08162-3.

[122] N. Goel, S. Kaur, and R. Bala, "Dual branch convolutional neural network for copy move forgery detection," *IET Image Process.*, vol. 15, no. 3, pp. 656–665, 2021, doi: 10.1049/ipr2.12051.

[123] A. K. Jaiswal and R. Srivastava, "Detection of Copy-Move Forgery in Digital Image Using Multi-scale, Multi-stage Deep Learning Model," *Neural Process. Lett.*, vol. 54, no. 1, pp. 1–26, 2022, doi: 10.1007/s11063-021-10620-9.

[124] J. Zhong, C. Pun, and S. Member, "An End-to-End Dense-InceptionNet for Image Copy-Move Forgery Detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 2134–2146, 2020, doi: 10.1109/TIFS.2019.2957693.

[125] R. Agarwal and O. P. Verma, "An Efficient Method of Copy Move Forgery Detection Using a deep learning based Feature Extraction and Matching Algorithm," *Multimed. Tools Appl.*, vol. 79, no. 11, pp. 7355–7376, 2019, doi: https://doi.org/10.1007/s11042-019-08495-z.

[126] M. A. Elaskily, H. A. Elnemr, M. M. Dessouky, and O. S. Faragallah, "Two stages object recognition based copy-move forgery detection algorithm," *Multimed. Tools Appl.*, vol. 78, no. 11, pp. 15353–15373, 2019, doi: 10.1007/s11042-018-6891-7.

[127] Y. Rodriguez-Ortega, D. M. Ballesteros, and D. Renza, "Copy-move forgery detection (CMFD) using deep learning for image and video forensics," *J. Imaging*, vol. 7, no. 3, pp. 59–75, 2021, doi: 10.3390/ jimaging7030059.

[128] T. T. Ng and S. F. Chang, "A model for image splicing," in *Proc. Int. Conf. Image Process.*, 2004, vol. 5, pp. 1169–1172, doi: 10.1109/ICIP.2004.1419512.

[129] Y. Zhang, C. Zhao, Y. Pi, and S. Li, "Revealing Image Splicing Forgery Using Local Binary Patterns of DCT Coefficients," in *Communications, signal processing, and systems*, 2012, pp. 181–189, doi: 10.1007/978-1-4614-5803-6.

[130] Y. Q. Shi, C. Chen, and W. Chen, "A natural image model approach to splicing detection," in *Proc. 9th workshop Multimed. Secur.*, 2007, pp. 51–62, doi: 10.1145/1288869.1288878.

[131] X. Li, T. Jing, and X. Li, "Image splicing detection based on moment features and Hilbert-Huang Transform," in *Proc. IEEE Int. Conf. Inf.Theory Inf. Secur.*, 2010, pp. 1127–1130, doi: 10.1109/ICITIS.2010.5689754.

[132] J. Dong, W. Wang, T. Tan, and Y. Q. Shi, "Run-length and edge statistics based approach for image splicing detection," in *Int. workshop digit. waterm.*, 2008, pp. 76–87, doi: 10.1007/978-3-642-04438-0_7.

[133] T. Carvalho, C. Riess, H. Pedrini, and A. Rocha, "Exposing Digital Image Forgeries by Illumination Color Classification," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 7, pp. 1182–1194, 2013, doi: 10.1109/TIFS.2013.2265677.

[134] S. Agarwal, "Texture Operator based Image Splicing Detection Hybrid Technique," in *Second Int. Conf. Comput. Intell. Commun. Technol.*, 2016, pp. 9–13, doi: 10.1109/CICT.2016.31.

[135] S. Agarwal and S. Chand, "Image forgery detection using Markov features in

undecimated wavelet transform," in *Proc. 2016 9th Int. Conf. Contemp. Comp.*, 2017, pp. 1–6, doi: 10.1109/IC3.2016.7880221.

[136] B. Chen, X. Qi, X. Sun, and Y. Shi, "Quaternion pseudo-Zernike moments combining both of RGB information and depth information for color image splicing detection," *J. Vis. Commun. Image Represent.*, vol. 49, pp. 283–290, 2017, doi: /10.1016/j.jvcir.2017.08.011.

[137] Z. Moghaddasi, H. A. Jalab, and R. M. Noor, "Image splicing forgery detection based on low-dimensional singular value decomposition of discrete cosine transform coefficients," *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7867–7877, 2018, doi: 10.1007/s00521-018-3586-y.

[138] 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, pp. 81–88, 2017, doi: 10.1007/s11760-016-0899-0.

[139] H. A. Jalab, T. Subramaniam, R. W. Ibrahim, H. Kahtan, and N. F. M. Noor, "New texture descriptor based on modified fractional entropy for digital image splicing forgery detection," *Entropy*, vol. 21, no. 4, pp. 1–9, 2019, doi: 10.3390/e21040371.

[140] H. Sheng, X. Shen, Y. Lyu, Z. Shi, and S. Ma, "Image splicing detection based on Markov features in discrete octonion cosine transform domain," *IET Image Process.*, vol. 12, no. 10, pp. 1815–1823, 2018, doi: 10.1049/iet-ipr.2017.1131.

[141] P. Niyishaka and C. Bhagvati, "Image splicing detection technique based on Illumination-Reflectance model and LBP," *Multimed. Tools Appl.*, vol. 80, no. 2, pp. 2161–2175, 2021, doi: 10.1007/s11042-020-09707-7.

[142] M. Hussain, S. Q. Saleh, H. Aboalsamh, G. Muhammad, and G. Bebis, "Comparison between WLD and LBP descriptors for non-intrusive image forgery detection," in *Proc. 2014 IEEE Int. Symp. Innov. Intell. Syst. Appl.*, 2014, pp. 197–204, doi: 10.1109/INISTA.2014.6873618.

[143] A. A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, and G. Bebis, "Splicing image forgery detection based on DCT and Local Binary Pattern," in *Proc. 2013 IEEE Global Conf. Signal Inf. Process.*, 2013, pp. 253–256, doi: 10.1109/GlobalSIP.2013.6736863.

[144] P. Korus and J. Huang, "Multi-Scale Analysis Strategies in PRNU-Based Tampering Localization," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 4, pp. 809–824, 2017, doi: 10.1109/TIFS.2016.2636089.

[145] Y. Wei, X. Bi, and B. Xiao, "C2R Net: The Coarse to Refined Network for Image

Forgery Detection," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Priv. in Computi. Commun.*, 2018, pp. 1656–1659, doi: 10.1109/TrustCom/BigDataSE.2018.00245.

[146] S. Lyu, X. Pan, and X. Zhang, "Exposing Region Splicing Forgeries with Blind Local Noise Estimation," *Int. J. Comput. Vis.*, vol. 110, no. 2, pp. 202–221, 2014, doi: 10.1007/s11263-013-0688-y.

[147] B. Xiao, Y. Wei, X. Bi, W. Li, and J. Ma, "Image splicing forgery detection combining coarse to refined convolutional neural network and adaptive clustering," *Inf. Sci. (Ny).*, vol. 511, pp. 172–191, 2020, doi: 10.1016/j.ins.2019.09.038.

[148] M. Huh, A. Liu, A. Owens, and A. A. Efros, "Fighting Fake News: Image Splice Detection via Learned Self-Consistency," in *Proc. Eur. Conf. Comp. Vision*, 2018, pp. 101–117, doi: 10.1007/978-3-030-01252-6_7.

[149] N. T. Pham, J. Lee, G. Kwon, and C. Park, "Hybrid Image-Retrieval Method for Image-Splicing Validation," *Symmetry (Basel).*, vol. 11, no. 83, pp. 1–15, 2019, doi: 10.3390/sym11010083.

[150] I. Amerini, L. Ballan, R. Caldelli, A. Del, L. Del, and G. Serra, "Copy-move forgery detection and localization by means of robust clustering with J-Linkage," *Signal Process. Image Commun.*, vol. 28, no. 6, pp. 659–669, 2013, doi: 10.1016/j.image.2013.03.006.

[151] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler, "COVERAGE-A novel database for copy-move forgery detection," in *Proc. 2016 IEEE Int. Conf. Image Process.*, 2016, pp. 161–165.

[152] J. Dong, W. Wang, and T. Tan, "CASIA Image Tampering Detection Evaluation Database," in *Proc. 2013 IEEE China Summit Int. Conf. Signal Image Process.*, 2013, pp. 422–426, doi: 10.1109/ChinaSIP.2013.6625374.

[153] Y.-F. Hsu and S.-F. Chang, "Detecting image splicing using geometry invariants and camera characteristics consistency," in *Proc. 2006 IEEE Int. Conf. Multimed. Expo.*, 2006, pp. 549–552, doi: 10.1109/ICME.2006.262447.

[154] T.-T. Ng and S. Chang, "A Data Set of Authentic and Spliced Image Blocks," Columbia University, New York, Technical Report 203-2004-3, 2004.

[155] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques," in *Proc. 2014 IEEE Int. Conf. Image Process.*, 2014, pp. 5302–5306, doi: 10.1109/ICIP.2014.7026073.

[156] R. Achanta, A. Shaji, K. Smith, and A. Lucchi, "SLIC Superpixels Compared to State-

of-the-Art Superpixel Methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2281, 2012, doi: 10.1109/TPAMI.2012.120.

[157] C. Han, "Improved SLIC imagine segmentation algorithm based on K-means," *Cluster Comput.*, vol. 20, no. 2, pp. 1017–1023, 2017, doi: 10.1007/s10586-017-0792-9.

[158] M. Bilal, H. A. Habib, Z. Mehmood, R. M. Yousaf, T. Saba, and A. Rehman, "A robust technique for copy-move forgery detection from small and extremely smooth tampered regions based on the DHE-SURF features and mDBSCAN clustering," *Aust. J. Forensic Sci.*, vol. 53, no. 4, pp. 459–482, 2021, doi: 10.1080/00450618.2020.1715479.

[159] C. Wang, Z. Zhang, Q. Li, and X. Zhou, "An Image Copy-Move Forgery Detection Method Based on SURF and PCET," *IEEE Access*, vol. 7, pp. 170032–170047, 2019, doi: 10.1109/ACCESS.2019.2955308.

[160] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *Proc. 2013 Brit. Mach. Vision Conf.*, 2013, vol. 34, no. 7, pp. 1281–1298, doi: 10.5244/C.27.13.

[161] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, 1990, doi: 10.1109/34.56205.

[162] Ş. Öztürk and B. Akdemir, "Application of Feature Extraction and Classification Methods for Histopathological Image using GLCM, LBP, LBGLCM, GLRLM and SFTA," in *Proc. Int. Conf. Comp. Intell. Data Sci.*, 2018, vol. 132, pp. 40–46, doi: 10.1016/j.procs.2018.05.057.

[163] J. Manikandan and B. Venkataramani, "Design of a real time automatic speech recognition system using Modified One Against All SVM classifier," *Microprocess. Microsyst.*, vol. 35, no. 6, pp. 568–578, 2011, doi: 10.1016/j.micpro.2011.06.002.

[164] D. Sundararajan, "Sundararajan Fundamentals of the Discrete Haar Wavelet Transform," 2011.

[165] H. Seong, H. Choi, H. Son, and C. Kim, "Image-based 3D building reconstruction using A-KAZE feature extraction algorithm," in *Int. Symp. Automati. Robot. Const.*, 2018, pp. 1–5, doi: 10.22260/isarc2018/0127.

[166] A. Abdiansah and R. Wardoyo, "Time Complexity Analysis of Support Vector Machines (SVM) in LibSVM," *Int. J. Comput. Appl.*, vol. 128, no. 3, pp. 28–34, 2015, doi: 10.5120/ijca2015906480.

[167] S. Raschka, "STAT 479: Machine Learning Lecture Notes," 2019. [Online]. Available: https://pages.stat.wisc.edu/~sraschka/teaching/stat479-fs2019/.

[168] K. Asghar, X. Sun, P. L. Rosin, M. Saddique, M. Hussain, and Z. Habib, “Edge–texture feature-based image forgery detection with cross-dataset evaluation,” *Mach. Vis. Appl.*, vol. 30, no. 7, pp. 1243–1262, 2019, doi: 10.1007/s00138-019-01048-2.

[169] V. Kumar and P. Gupta, “Importance of Statistical Measures in Digital Image Processing,” *Int. J. Emerg. Technol. Adv. Eng.*, vol. 2, no. 8, pp. 56–62, 2017.

[170] Sun XW, Li YJ, and Chen Y, “Application of local standard deviation filtering in image processing,” *Electron. Opt. Control*, vol. 15, no. 9, pp. 32–34, 2008.

[171] D. Vaishnavi and T. S. Subashini, “Recognizing image splicing forgeries using histogram features,” in *Proc. 2016 MEC Int. Conf. Big Data Smart City.*, 2016, pp. 53–56, doi: 10.1109/ICBDSC.2016.7460342.

[172] C. Chen and Y. Q. Shi, “JPEG Image Steganalysis Utilizing both Intrablock and Interblock Correlations,” in *Proc. 2008 IEEE Int. Symp. Circuits Syst.*, 2008, pp. 3029–3032, doi: 10.1109/ISCAS.2008.4542096.

[173] A. Kumar, C. S. Prakash, S. Maheshkar, and V. Maheshkar, “Markov Feature Extraction Using Enhanced Threshold Method for Image Splicing Forgery Detection,” in *Smart Innovat. Comm. Comput. Sci.*, 2019, pp. 1–17, doi: 10.1007/978-981-10-8971-8_2.

[174] N. Jindal and K. Singh, “Image and video processing using discrete fractional transforms,” *Signal, image video Process.*, vol. 8, no. 8, pp. 1543–1553, 2012, doi: 10.1007/s11760-012-0391-4.

[175] S. N. Sharma, R. Saxena, and S. C. Saxena, “Tuning of FIR filter transition bandwidth using fractional Fourier transform,” *Signal Processing*, vol. 87, no. 12, pp. 3147–3154, 2007, doi: 10.1016/j.sigpro.2007.06.005.

[176] B. Chen, M. Yu, Q. Su, H. J. A. E. Shim, and Y. Shi, “Fractional Quaternion Zernike Moments for Robust Color Image Copy-Move Forgery Detection,” *IEEE Access*, vol. 6, pp. 56637–56646, 2018, doi: 10.1109/ACCESS.2018.2871952.

[177] N. Jindal and K. Singh, “Digital image forensics-gateway to authenticity: Crafted with observations, trends and forecasts,” in *Handbook of Multim. Inf. Security: Techn. Appl.*, 2019, pp. 681–701.

[178] E. Rublee, W. Garage, and M. Park, “ORB : an efficient alternative to SIFT or SURF,” in *Proc. 2011 IEEE Int. Conf. Comput. Vision*, 2011, pp. 2564–2571, doi: doi:10.1109/iccv.2011.6126544.

[179] P. Sujatha and K. K. Sudha, “Performance Analysis of Different Edge Detection Techniques for Image Segmentation,” *Indian J. Sci. Technol.*, vol. 8, no. 14, pp. 1–6, 2015, doi: 10.17485/ijst/2015/v8i14/72946.

[180] K. S. Sudeep and K. K. Pal, “Preprocessing for image classification by convolutional neural networks,” in *Proc. IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol.*, 2017, pp. 1778–1781, doi: 10.1109/RTEICT.2016.7808140.

[181] A. Hegazi, A. Taha, and M. M. Selim, “An improved copy-move forgery detection based on density-based clustering and guaranteed outlier removal,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 33, no. 9, pp. 1055–1063, 2021, doi: 10.1016/j.jksuci.2019.07.007.

[182] A. B. Z. Abidin, H. B. A. Majid, A. B. A. Samah, and H. B. Hashim, “Copy-Move Image Forgery Detection Using Deep Learning Methods: A Review,” in *Proc. 2019 6th Int. Conf. on Res. and Innov. in Inf. Syst.*, 2019, pp. 3–8, doi: 10.1109/ICRIIS48246.2019.9073569.

[183] S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv Prepr.*, pp. 1–14, 2016, [Online]. Available: http://arxiv.org/abs/1609.04747.

VITA

Navneet Kaur was born in Sri Muktsar Sahib, Punjab, India, in 1991. She received the B.Tech and M.Tech degree in Electronics and Communication Engineering from I. K. Gujral Punjab Technical University, Jalandhar, Punjab, India, in 2014 and 2016, respectively. She is presently working towards a Ph.D. degree in the Electronics and Communication Engineering Department, Thapar Institute of Engineering and Technology, Patiala, Punjab, India. Her research interests include image processing, digital image forensics and signal processing. Her email ID is navneetbrar5@gmail.com.