IMAGE FORENSIC USING MACHINE LEARNING

A Thesis

Submitted in fulfilment of the requirement for the award of the degree

of

DOCTOR OF PHILOSOPHY

in

Electronics and Communication Engineering

Submitted by

ABHISHEK

Reg. No. 951606007

Under the Supervision of

Dr. Neeru Jindal
Assistant Professor
TIET, Patiala (Punjab), India

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Electronics and Communication Engineering Department
Thapar Institute of Engineering and Technology, Patiala-147004

September-2020

CERTIFICATE

I, **Abhishek,** at this moment declares that the work contained in the thesis entitled **"IMAGE FORENSIC USING MACHINE LEARNING"** being submitted by me to the **Department of Electronics and Communication Engineering, Thapar Institute of Engineering and Technology, Patiala** in fulfillment of the award of the degree of **"Doctor of Philosophy"** is a record of original research work carried out under the supervision of **Dr. Neeru Jindal**. The matter presented in this thesis does not incorporate any material previously published or written by any other person except where due references are made in the text. The results obtained in this thesis have not been submitted in part or full to any other institute or university for the award of degree or diploma.

Date : 11/9/2020

Abhishek 11/09/2020

(Reg. No. 951606007)

This is to certify that the above statement made by the candidate is correct and true to the best of our knowledge and belief. He has worked under our supervision and fulfilled the requirements for the submission of this thesis, which has reached the requisite standard.

Dr. Neeru Jindal 11/9/2020

Assistant Professor, ECED

TIET, Patiala-147001, India

Date : 11/09/2020

ACKNOWLEDGEMENT

I sincerely thank my creator, the almighty God, for the gift of life, knowledge, and strength to accomplish this work. This research work is, by far, the most significant accomplishment in my life, and it would be impossible without peoples who supported me and believed in me. I want to extend my gratitude and my sincere gratitude to my honorable research advisor, Dr. Neeru Jindal, for her support and guidance during my study. She is not only a great professor with deep vision but also and, most importantly, a kind person. She has always been willing to give me insights and pointers to tackle the problems along the way, keeping a very keen interest in listening to my problems. She is always there when I needed to see her and attentive when I spoke with her. My respect for her grew and will always grow throughout my entire life. She always pointed me in the right direction when I lost the path and supported me when I was on the right path. I sincerely thank you for her right guidance and encouragement. Her trust and support inspired me in the most important moments of making the right decisions, and I am glad to work with her. Her moral support when I faced hurdles is unforgettable. I would also like to thank the facilities provided by the Thapar Institute of Engineering and Technology, Patiala, for the successful work.

I want to thank my committee members. Also, my completion of acknowledgment remains void without thanking my doctoral committee members Dr. Alpana Agarwal, Dr. Vinay Kumar, Dr. Sanjay Sharma, and Dr. MD Singh. I am also thankful to various researchers of the world, who are engaged in sharing their professional knowledge in the web communities of Research Gate, Wikipedia, and others.

I want to thank my wife, Hema Thakur, for inspiring me and raising the bar. She motivated me for a higher qualification. I would also like to offer my sincerest thanks and praise to Dr. Rajesh Kumar and Dr. Hardeep Singh Saini. They motivated me for research and conducted the highest quality research possible.

Last but not least, I would like to thank my parents, who taught me the value of hard work by their example, and all those persons who directly or indirectly helped me during my work and contributed towards this work.

Abhishek 11/09/2020

TIET, Patiala (Punjab), India

ABSTRACT

Nowadays, it is challenging to trust any digital image due to the convenient availability of manipulation software like Photoshop, GIMP, and Coral Draw etc. Therefore, it becomes tough to differentiate between an authentic image and tampered image. Traditional methods for image forgery detection generally use handcrafted features. The challenge with the traditional image tampering detection approaches is that most of the methods need improvement as only certain features are identified. These days, Machine learning (ML) and deep learning (DL) are widely used in image forgery. These techniques prove their efficacy with better accuracy and other performance parameters than traditional methods. There are many types of image forgery, like copy-move, splicing, and retouching. In this thesis, copy-move and splicing forgery are detected using ML and DL techniques.

The first algorithm provides a copy-move image forgery detection using machine learning and deep learning. In this work, machine and deep learning algorithms are proposed to find out different image forgeries. First, the proposed algorithm applies color illumination in pre-processing, then Scale Invarient Feature Transform (SIFT) is used to extract features, and Support Vector Machine (SVM) classifies correct forged pixels. The proposed methodology gives better results for CMF detection as Precision=97.25%, Recall=100%, and F1=98.53%.

The second algorithm provides a deep convolution neural network (DCNN) that uses automatic feature extraction and localizes copy-move forgery and splicing forgery. In the feature extraction and localize forgery, the performance can be enhanced using the ML and DL. Finally, the applications of proposed color illumination, convolution neural network, and semantic segmentation are demonstrated for forgery detection. The proposed algorithm performance accuracy is calculated on the CASIA v1.0 validation set, and the test set is 98% and 99%, respectively. The performance accuracy is calculated on the CASIA v2.0 validation set, and the test set is 98% and 98%, respectively. The DVMM dataset forgery detection accuracy is 97%. The BSDS300 dataset forgery detection accuracy is 98%. The proposed algorithm is tested on image-level on CMFD dataset and achieved performance accuracy, i.e. Precision (P) = 98%, Recall (R) = 100% and F1 = 99%.

The third algorithm presented robustness of algorithms against geometrical attacks using color illumination, a deep convolution neural network, SIFT, and SVM. Geometrical attacks, such as scaling, rotation, and JPEG, were identified. The plain CMF attack detection results are: P=97.25%; R=100% and F1=98.53%. The JPEG CMF attack detection results are: P=71.44%; R=58.44% and F1=63.77%. The scale CMF attack detection results are: P=85.2%; R=74.8% and F1=79.1%. The rotation CMF attack results are: P=87.83%; R=76.33% and F1=86.16%. Comparison with state-of-the-art techniques proves the efficacy of the presented algorithms. In the future, suggested algorithms can be implemented on real-time applications with some improvements.

LIST OF PUBLICATIONS

P1. Abhishek and Jindal N.: 'Image forensics using color illumination, block and key point-based approach,' Multimedia Tools and Applications, vol. 77, no. 19, pp. 26033–26053, 2020, 2018, doi: 10.1007/s11042-018-5836-5 2018, (SCI Indexed) (Impact Factor 2.31)

P2. Abhishek and Jindal N.: 'Hybrid Deep Learning and Machine Learning Approach for Passive Image Forensic,' IET Image Processing, vol. 14, no. 10, pp. 1952–1959, 2020, doi: 10.1049/iet-ipr.2019.1291, (SCI Indexed) (Impact Factor 2.07)

P3. Abhishek and Jindal N.: 'Copy Move and Splicing Forgery Detection using Deep Convolution Neural Network, and Semantic Segmentation,' Multimedia Tools and Applications, 2020. (SCI Indexed) (Impact Factor 2.31)

P4. Abhishek and Jindal N.: 'Machine Learning-Based Saliency Algorithm for Image Forgery Classification and Localization.' In 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), pp. 451-456. IEEE, 2018.

P5. Abhishek and Jindal N.: 'Geometrical Attack Classification using DCNN and Forgery Localization using Machine Learning,' International Journal of Recent Technology and Engineering (IJRTE), 2019, 7(5S2), pp. 2277-3878.

COMMUNICATED

P1. Abhishek and Jindal N.: 'Image Forensic from Machine Learning to Deep Learning: Review' (Sadhana Published by the Indian Academy of Sciences – SCI - Communicated) (Impact Factor 0.94)

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
AI	Artificial Intelligence
CMF	Copy Move Forgery
CNN	Convolution Neural Network
CPU	Central Processing Unit
DCNN	Deep Convolution Neural Network
DCT	Discrete Cosine Transform
DL	Deep Learning
DCNN	Deep Convolution Neural Network
DSNN	Deep Siamese Neural Network
ELF	Energy of the Low Frequency
FT	Fourier Transform
GPU	General Processing Unit
IF	Image Forgery
JPEG	Joint Photographic Experts Group
KNN	K Nearest Neighbour
ML	Machine Learning
MRR	Multi Region Relation
MSE	Mean Square Error
MTPM	Markov Transition Probability Matrix
PCA	Principle Component Analysis
PLF	Percentage of the Low Frequency
RELU	Rectified Linear Unit
RHS	Right Hand Side
SF	Splicing Forgery
SNR	Signal Noise Ratio
SVM	Support Vector Machine

GLOSSARY OF SYMBOLS

k	Scaling Factor
$\|.\|$	Absolute Value
∂	Differential Operator
$f^{\sigma}(\text{pixel})$	Observed Intensities at Position 'Pixel,
f	Frequency
$s(\lambda, x, y)$	Surface Reflection
$e(\lambda, x, y)$	Light Source
$c(\lambda)$	Camera Sensitivity
$f(x, y)$, (x, y)	Image Value, Spatial Coordinate in the Image
δ	Observed Intensities at pixel position
σ	Gaussian kernel
x, p	Minkowski Norm
$*$	Complex Conjugate
$\otimes$	Convolution
η	Nth Derivative Order
(x, y)	Pixel Coordinate

LIST OF FIGURES

Fig. No.	Name of Figure	Pg. No.
Figure 1.1	Digital Image Forgery Detection Methods	2
Figure 1.2	Pixel Based Digital Image Forgery Detection Methods	3
Figure 1.3	Image forgery examples from CMFD and CASIA Dataset. Original images (I1), (I3), and (I4). Copy Move forged image (I2), Spliced image (I5)	4
Figure 1.4	Relationship betwnnen AI, ML, DL techniques	4
Figure 1.5	Machine learning general block diagram	5
Figure 1.6	Machine learning-based methods	6
Figure 1.7	Digital image forgery detection using a supervised learning method	6
Figure 1.8	General framework of the Deep Neural Convolution Network (DCNN)	8
Figure 2.1	Percentage error rate of different algorithms	13
Figure 2.2	Growths of Machine Learning	16
Figure 3.1	Types of Image Forgery	24
Figure 3.2	Generalized Framework for CMFD	25
Figure 3.3	Categorization of Forgery Techniques	26
Figure 3.4	Simple framework for the machine learning algorithm	28
Figure 3.5	Mathematical model of the simple machine learning model	28
Figure 3.6	Flow diagram for image forgery detection using Machine Learning	31
Figure 3.7	Results of Image forgery detection using Machine Learning	32
Figure 3.8	Block diagram for image forgery classification and detection using hybrid machine learning and deep convolution neural network	34
Figure 3.9	Block diagram of Image Processing in Unsupervised Learning	36

Figure 3.10	Block diagram of the autoencoder in unsupervised learning	36
Figure 3.11	Block diagram of the auto decoder in unsupervised learning	37
Figure 3.12	The output of the Siamese neural network for the training of positive and negative triplets in columns (a), (b), (c), and (d)	38
Figure 3.13	The output shows the results of forgery detection for the CMFD dataset. Column (a) shows the original image, column (b) represent forgery detected image, and column (c) shows the ground truth image	39
Figure 3.14	Basic block diagram of the machine and deep learning methods	39
Figure 4.1	Flow chart of the proposed forgery detection algorithm.	42
Figure 4.2	Example maps for forged images (top row) in red, green, and blue, as well as Color illuminations, chart generated for the forged image (bottom row).	44
Figure 4.3	Convert image into an irregular block size (original images top row) and (adaptive block size bottom row).	46
Figure 4.4	Blocking/segmentation images with adaptive block size	46
Figure 4.5	Fabricated pictures: (top row); found forgeries (middle row); increase intensity for forged pixels (bottom row)	47
Figure 4.6	SIFT feature match identification with the ground truth image (bottom row). Forged pictures (top row).	48
Figure 4.7	SLICO operation (top row); SIFT, Morphological operation (middle row); Ground truth image (bottom row)operation (middle row); four ground truth image (bottom row)	49
Figure 4.8	SVM classifies the forged match pixel values with colors (last column).	50
Figure 4.9	Image I1, I2, and I3 forged images with corresponding ground truth image; Images in columns 2, 3, and 4 illustrate 150, 250, and 159/224 fixed block size with corresponding detected forgery; In fifth column adaptive images and the corresponding forgery detection results.	51

Figure 4.10	Graphical representation of Precision and Recall analysis of the Proposed Scheme with existing image-level approaches.	53
Figure 4.11	SLICO operation (top row); SIFT, Morphological operation (middle row); Ground truth image (bottom row)	54
Figure 5.1	The block diagram of a machine learning-based algorithm	57
Figure 5.2	The Deep Convolutional Neural Network (DCNN) operation	58
Figure 5.3	The block diagram of hybrid deep learning and machine learning approach for passive image forensic.	60
Figure 5.4	Convolution based deep neural network	62
Figure 5.5	Convolution with the kernel	62
Figure 5.6	Convolution between color image and kernel with a stride of one.	63
Figure 5.7	Zero paddings to perform convolution between 7×7 image matrix and 3×3 kernel.	64
Figure 5.8	The Max pooling and average pooling operation	64
Figure 5.9	Classification of not spliced (shown as Sp in red color) and spliced image (shown as Sp in green color)	65
Figure 5.10	The CASIA-1 (row1), CASIA-2 (row2), DVMM (row3), BSDS300 (row4), CoMFD (row5) dataset of CMF and SF detection results.	67
Figure 5.11	Bar graph for CMF and SF accuracy comparison	69
Figure 5.12	The block diagram of the proposed DCNN model	70
Figure 5.13	The graph between training accuracy and iteration of a 27 layer DCNN model	71
Figure 5.14	The graph between loss and iteration of 27 layer DCNN model	72
Figure 5.15	The graph between training accuracy and iteration of 54 layer DCNN model.	74
Figure 5.16	The graph between loss and iteration of 54 layer DCNN model	74

Figure 5.17	Confusion matrix between CMF, video frame forged, and spliced forgery for training and validation of DCNN model	75
Figure 5.18	Classification of three classes CMF, video frame forgery, and splicing forgery using DCNN model	76
Figure 5.19	Flowchart of image forgery detection and localization using the proposed deep learning method.	77
Figure 5.20	The block diagram of proposed 91-layer deep convolution neural network (DCNN)	78
Figure 5.21	Forgery detection results. a) Input forged image, b) After color illumination, c) Overlay image, and d) Detected Forgery.	83
Figure 5.22	Bar graph of a) Forged and not forged pixel frequency, b) Confusion matrix, and c) Image mean IoU	83
Figure 5.23	Intersection over the union between test image, truth, and prediction is IOU=1.	84
Figure 5.24	Intersection over the union between test image, truth, a prediction is IoU=0.28165	84
Figure 5.25	Training plot between iteration and accuracy	85
Figure 5.26	Test accuracy using proposed deep learning methods	85
Figure 5.27	Bar graph between forged, not forged pixels and test accuracy, IoU, average boundary F1 score	86
Figure 5.28	Training plot between iteration and accuracy	87
Figure 5.29	Test accuracy using proposed deep learning methods.	87
Figure 5.30	Bar graph between forged, not forged pixels and test accuracy, IoU, average boundary F1 score.	88
Figure 5.31	Training plot between iteration and accuracy	89
Figure 5.32	Test accuracy using proposed deep learning methods	89
Figure 5.33	Bar graph between forged, not forged pixels and test accuracy, IoU, average boundary F1 score	90

Figure 5.34	Forgery detection results on color illuminated and rotated images using the proposed deep convolution neural network method	90
Figure 5.35	Forgery detection results on without color illuminated and rotated images using the proposed deep convolution neural network method.	91
Figure 5.36	Rotation attack results on CMFD Dataset comparison with existing methods.	92
Figure 5.37	CMF and SF detection result in DVMM dataset comparison.	93
Figure 6.1	Copy move forged image	96
Figure 6.2	Original images (I1), (I3), and (I4). CMF Forged Image (I2), Spliced image (I5)	97
Figure 6.3	Step by step execution results of the input image	104
Figure 6.4	Row-1 depicts the JPEG compression factor from J-30 to 90; Row-2 depicts color illumination for the corresponding image; Row-3 depicts the results of detecting forgery of a similar picture. Row-4 describes the ground truth image	105
Figure 6.5	Line graph between JPEG Compression attack and Accuracy: (a) Precision, (b) Recall, and (c) F1	107
Figure 6.6	Row-1 depicts the forged images with scale factor varies from S-91 to S-99; Row-2 depicts the color illumination; Row-3 depicts the forgery detection results; Ground truth images are shown in Row-4	108
Figure 6.7	Line graph between Scale attack and Accuracy: (a) Precision, (b) Recall, and (c) F1	109
Figure 6.8	The first row depicts forged images with a rotation factor from 2° to 10°; the second row represents the color illumination of the corresponding image; the third row illustrates the reproduced object localization result of the rotation attack; the fourth row depicts the GT images.	110

Figure 6.9 Line graph between rotation attack and accuracy: (a) Precision, (b) Recall, and (c) F1 112

LIST OF TABLES

Table No.	Name of Table	Pg. No.
Table 4.1	Image superpixel calculation for I1, I2, I3, and I4 forged images	45
Table 4.2	Superpixel picture estimation of variable image size I1, I2, I3, and I4	48
Table 4.3	Precision and recall performance compared with or without the adjustable scale of a superpixel with existing methods at the image level	52
Table 4.4	Precision, recall, and F1 image-level analysis of the proposed hybrid method	54
Table 5.1	DCNN training on CASIA v1.0 using transfer learning	65
Table 5.2	DCNN training on CASIA v2.0 using transfer learning	66
Table 5.3	Comparison between CMF and SF dataset accuracy	68
Table 5.4	Twenty-seven layers of the DCNN model	70
Table 5.5	Training accuracy of proposed 27 layers DCNN model	71
Table 5.6	The 54 layers DCNN model for training of a copy-move, splicing, and forged video frames	72
Table 5.7	Training accuracy of proposed 54 layers DCNN deep learning model	73
Table 5.8	Training accuracy for twenty epochs using proposed deep learning methods	84
Table 5.9	Test accuracy for twenty epochs using proposed deep learning methods	85
Table 5.10	Test accuracy, IoU, and average boundary F1 score using proposed deep learning methods.	86
Table 5.11	Training accuracy for seven epochs using proposed deep learning methods	86
Table 5.12	Test accuracy for seven epochs using proposed deep learning methods	87

Table 5.13	Test accuracy, IoU, and average boundary F1 score using proposed deep learning methods.	88
Table 5.14	Training accuracy using proposed deep learning methods	88
Table 5.15	Test accuracy using proposed deep learning methods	89
Table 5.16	Test accuracy, IoU, and average boundary F1 score using proposed deep learning methods	90
Table 5.17	The CMFD Dataset rotation attack results compared with other methods [2], [47].	92
Table 5.18	Comparison of the proposed method with other methods [33], [48]	92
Table 5.19	Comparison of the proposed method with other methods	95
Table 6.1	Deep CNN layer's specification	98
Table 6.2	Comparison of validation accuracy for DVMM, CASIA V1.0, and CASIA V2.0	101
Table 6.3	Copy-move forgery localization results in comparison with existing methods	105
Table 6.4	Geometric attack localization results in comparison with existing methods	112

TABLE OF CONTENTS

	Page No.
Certificate	i
Acknowledgments	ii
Abstract	iii
List of Publications	v
List of Abbreviations	vi
Glossary of Symbols	vii
List of Figures	viii
List of Tables	xiv
Table of Contents	xvi
1. Introduction	**1-11**
1.1.Image Forgery	1
1.2.Machine Learning	4
1.2.1. Types of Machine Learning	5
1.2.1.1. Supervised Learning	6
1.2.1.2. Unsupervised Learning	7
1.2.1.3.Reinforcement learning	7
1.3.Deep Learning	7
1.4.Image forgery using machine learning	8
1.5.Challenges	9
1.6.Contribution of research work	9
1.7.Thesis Organization	10
2. Literature Survey	**12-21**
2.1. Image forging's and machine learning historical background	12
2.2. CMF detection using traditional methods	13
2.3. CMF and SF detection using machine learning	15
2.4. CMF and SF detection using deep learning	17
2.5.Motivation	19
2.6.Gaps in study	20
2.7.Objectives	21

3. Image forgery and machine learning **22-40**
3.1. Types of Image Forgery 22
3.2.Flow diagram for CMF Detection 25
3.3.Classification of CMFD Techniques 26
3.3.1. Methods based on blocks 27
3.3.2. Methods based on key points 27
3.3.3. Techniques for Hybrid CMFD 27
3.4.Basics of Machine Learning 28
3.4.1. Image forgery detection using Machine Learning 31
3.5.Basics of Deep Learning 32
3.5.1. Image forgery classification and detection using ML and DL 33
3.5.2. Image forgery detection using Siamese Neural Network 35
3.5.2.1. Pre Processing 36
3.5.2.2. Auto Encoder 36
3.5.2.3. Auto Decoder 37
3.5.2.4. Localization of Forgery 38
3.5.3. ML and DL computation requirements 40
3.6. Summary 40
4. CMF detection using Machine learning **41-55**
4.1.Background of Copy move forgery 41
4.2.Proposed CMF detection algorithm 42
4.2.1. Dataset description 43
4.2.2. Color Illumination 43
4.2.2.1. Illuminants for detecting forgeries 44
4.2.2.2. Adaptive Over Segmentation 45
4.2.2.3. Simple Linear Iterative Clustering (SLICO) 46
4.2.3. SIFT Feature Extraction 47
4.2.4. Feature Matching 48
4.2.5. Classification 49
4.2.6. Post Processing 50
4.3 Simulation Results 51

4.4 Summary 55

5. CMF and SF forgery detection using Deep learning 56-95

5.1. Introduction to CMF and SF detection 56

5.2. Building blocks of Machine Learning technique 57

5.3. Basics of Deep Learning Technique 58

5.4. CMF and SF detection using DCNN Algorithm 58

5.4.1. DCNN block diagram 59

5.4.2. Convolution 61

5.4.3. Stride 63

5.4.4. Padding 63

5.4.5. Pooling 64

5.4.6. ReLU Transfer Function 64

5.4.7. Image forgery classification 65

5.4.8. Proposed Algorithm Results 66

5.5 CMF and SF detection using DCNN and Semantic Segmentation 69

5.5.1 Choosing the Dataset 77

5.5.2 Proposed hybrid technique block diagram 78

5.5.2.1. Forgery detection and localization algorithm 82

5.5.3 Results of the proposed algorithm 82

5.5.4 Proposed model advantages, limitations, and open problems 93

5.5.5 Proposed model comparative evaluation 94

5.6 Summary 95

6. Geometrical Attacks 96-113

6.1 CMF, SF and Geometrical Attacks detection 97

6.2 Forgery Classification and Localization Algorithm 98

6.2.1 Classification of various geometric attacks 99

6.2.1.1. Pseudocode for the proposed algorithm 99

6.2.1.2. Prepare Data to set for Transfer Learning 100

6.2.1.3. Classification of CM and SP images 101

6.3 Localization of various geometric attacks 102

6.3.1 Setup for Feature Extraction 102

6.3.2 Pseudocode for forgery localization 102
6.3.3 Classifier selection and modelling 103
6.4 Experimental Results 103
6.4.1 JPEG attack localization 105
6.4.2 Scale attack localization 107
6.4.3 Rotation attack localization 110
6.5 Summary 112
7. Conclusions **114-115**
7.1 Conclusion 114
7.2 Future Scope of work 115
References **116-125**

CHAPTER 1

INTRODUCTION

The electronic computer uses software tools for editing digital images. These software tools process images to improve image quality and facilitate efficient storage. Visual artifacts, encoding, picture, and video frame recognition are examples of unique image processing operations. With the simple integration of image and video editing tools, anyone with simple computer knowledge can easily modify, change, and override accessible image properties. Therefore, as in the biography, the creation of digital forgeries has become relatively straightforward. People use photographs on social media platforms as the primary source of information. The image and video of evidence against someone are expected to be shown on TV news, including verifying truthfulness, conviction, and credibility [1]-[6]. Instant criminality recorded and caught in a CCTV camera is treated and used as evidence in law [7]-[8]. If used in the newspaper or the courts of law, these forgeries may negatively impact our culture. In addition to those drawbacks, the accessibility of digital visual media presents numerous disadvantages. In image forensics, there are broadly two categories: active and passive forgery. The main contribution of this work is to find passive image forgery with improved performance parameters using machine learning. In passive image forgery, copy-move forgery and splicing forgery are detected with the help of various datasets and then compared with the state-of-the-art. It has been observed that the proposed work has better efficacy. The major challenge in finding passive image forgery is that the source image is not present. Therefore, pixel-level features are extracted to separate the original and forged patches to solve this problem. Thus, the proposed work will improve image forgery detection techniques using machine learning in the research community. The following section elaborates on image forgery, its types, and methods.

1.1 IMAGE FORGERY

Image forensic is a multimedia security area that includes the identification of forged regions from the image. It is easy to maneuver and manipulate digital photos. Digital image fraud detection is further defined as detecting active forgery and passive falsification [8]-[9], as shown in Fig. 1.1. Determination of integrity and authenticity of images is a challenge or main research question for humans and machinery. So there is a great need for reliable algorithms to check

image forgery. Only in the presence of some previous picture information do active techniques work. Therefore, if images from inaccurate or untruthful sites are investigated, these techniques are not acceptable. However, the watermarking process significantly degrades and deteriorates as image quality. Passive methods are often referred to as blind approaches because the source picture is not visible. For any prior knowledge on the image, there are no preconditions, such as active techniques. Such methods are developed based on the assumption that the tempered representations approach instills objects by changing fundamental statistical features and characteristics. Such incoherence is widely used to investigate falsification. The picture is subjected to various forms of attack and ramification during tempering. CMF Pun [2], which involves replicating some region (or regions) in the image, is the simplest of all forgery and ramifications. Replication of image areas, such as in image splicing, is done sometimes from other digital images.

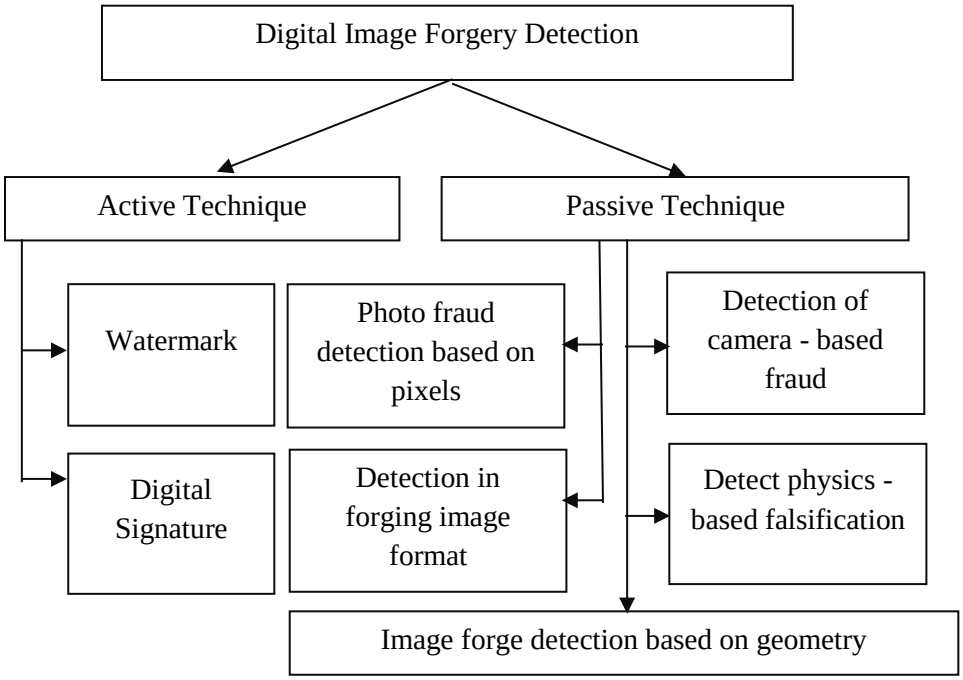


Fig. 1.1 Digital Image Forgery Detection Methods

The various algorithms for passive image falsification are mainly classified into five classes. We use a pixel-based approach to collect image information and concentrate on pixel image information, as shown in Fig. 1.2. Joint Photographic Experts Group (JPEG), Double JPEG, JPEG, and JPEG-blocks use the format-based technique. The majority of work on forged image detection on various image formats has been done in JPEG format. The camera-based approach

Singh *et al.* [3] will perform quantification, color analysis, and filtering from the camera sensor to memory. These steps can be changed from camera to camera. The various stage of the image creation process introduces several different artifacts.

The camera forgery can be detected by taking into account all inconsistencies. Lighting irregularities in light sources may be used to detect tampering with forgery detection systems that depend on physics under different lighting conditions. When several images are combined, the resulting luminous conditions are changed. Technology-based on geometry employs, explains, and integrates geometric limitations from future perspectives. These are categorized, assuming that the camera's basic parameters (such as the primary point, widest aperture, tilt, and image resolution) are metric and multiple view geometries [9]-[20]. The visual axis is associated with the target plane, i.e., the key point is located in the center of the original image. When a particular section is interpreted or moved close to a picture or two or more pictures are combined, the classifier considers it is challenging to keep the key point in its correct position. The use and development of efficient and effective detection systems are thus associated with projective geometry principles.

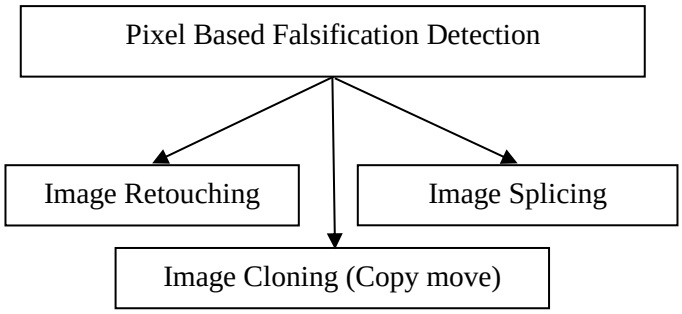


Fig. 1.2 Pixel Based Digital Image Forgery Detection Methods

Image forgery implies digital image editing to hide important image information, as shown in Fig. 1.3. The splicing combines two pictures into a picture for the random collection, as shown in Fig. 1.3 (I3, I4, I5). Few patches are copied and pasted into the same image in CMF, as shown in Fig. 1.3. (I1, I2) Li and Zhou [21]. In SF, only a few areas from one image are copied and inserted on another. Two commonly employed falsifications are the CMF and SF [22]-[58]. We typically use key-point [59]-[61] and block-based approaches to find and identify fraud of this kind. Block-based methods Zheng *et al.* [26] divide the image into many overlapping blocks, which causes

time complexity. Key point-based methods fail to find forgeries, and only a few key points can be extracted in the smooth image. Both approaches have some advantages and disadvantages.

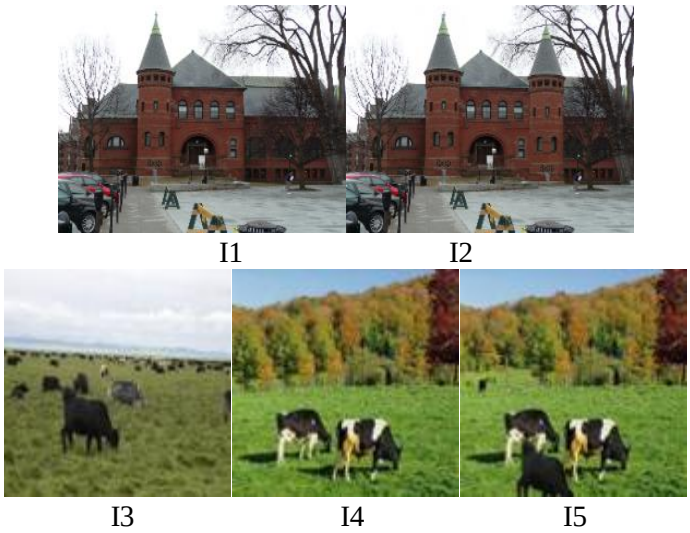


Fig. 1.3 Image forgery examples from CMFD and CASIA Dataset. Original images (I1), (I3), and (I4). Copy Move forged image (I2), Spliced image (I5)

1.2 MACHINE LEARNING

Machine learning's evolution and exponential growth are making it the most popular field of the twenty-first century. Artificial Intelligence (AI) is the more eminent domain in which Machine Learning (ML) is the subcategory of AI, and Deep Learning (DL) is the subclass of ML, as shown in Fig. 1.4. Machine learning algorithms are used to extract forged pixels and localize them.

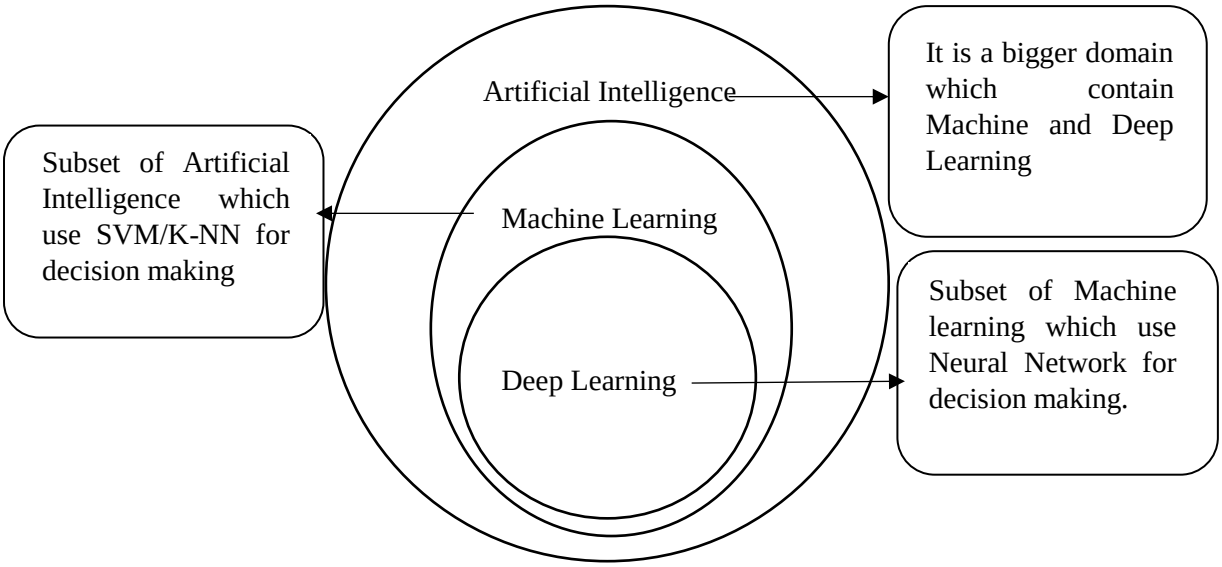


Fig. 1.4 Relationship between AI, ML, DL techniques

ML algorithm learns from examples and historical events. The ML definition allows an algorithm to understand whether an algorithm improves its performance in a given task with some training. While these techniques have been used since the 1940s, the General Processing Unit (GPU) data are making a difference now. GPUs make groundbreaking improvements for machine learning. ML and DL use two relatively recent trends: the availability of huge volumes of training data and parallel GPU computing. There is a need for automatic feature extraction in forgery detection. Therefore, the following areas of forgery work include algorithms that use machine learning and deep learning. ML recognizes an image based on characteristics.

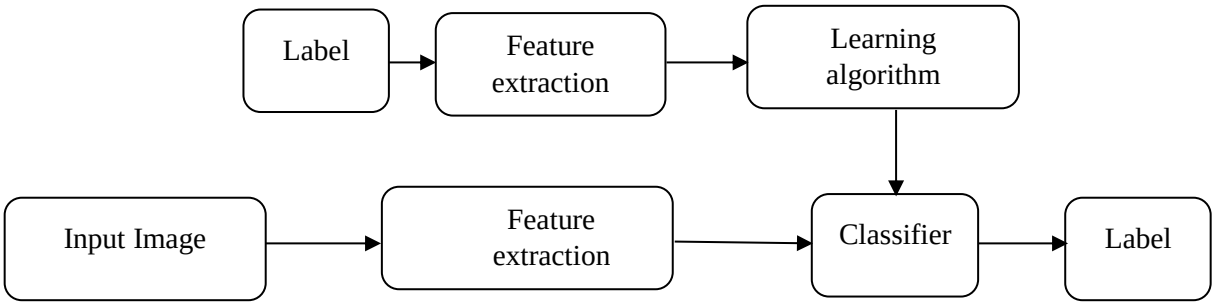


Fig. 1.5 Machine learning general block diagram [54]

The labeled dataset is given to the input block, as shown in Fig. 1.5. Features are first extracted from the label's dataset and then applied to the learning algorithm. During an evaluation, the classifier predicts results by evaluating test image features and learning image features. Several classifiers can classify the forged and original images 13, 14.

1.2.1 TYPES OF MACHINE LEARNING

ML technique is divided into supervised, unsupervised, and reinforcement learning, as shown in Fig. 1.6. Supervised learning problems are of two types, regression and classification. In the regression method, the dependent variable contains continuous probability values between zero and one. In classification problems, we had a dependent variable of certain types. The dependent variable here is zero and one. Deep learning makes extensive use of the sigmoid activation mechanism, which switches the discrete value extensively. Logistic regression comes from the activation function itself, having an output between 0 and 1. The sigmoid activation function is also known as the logistic activation function. Multiple-class classification problems, logistic regression, KNN, SVM, Naive Bayes, Decision Trees, and Random Forests techniques are used. Unsupervised learning is classified into three types: clustering, dimension reduction, and

association. Clustering algorithms such as k-means clustering and hierarchical clustering are used. Dimension reduction algorithms include Principal Component Analysis (PCA), Linear Discriminative Analysis (LDA), and Kernel PCA. Thompson sampling with upper confidence bounds is used in reinforcement learning's action reward scheme.

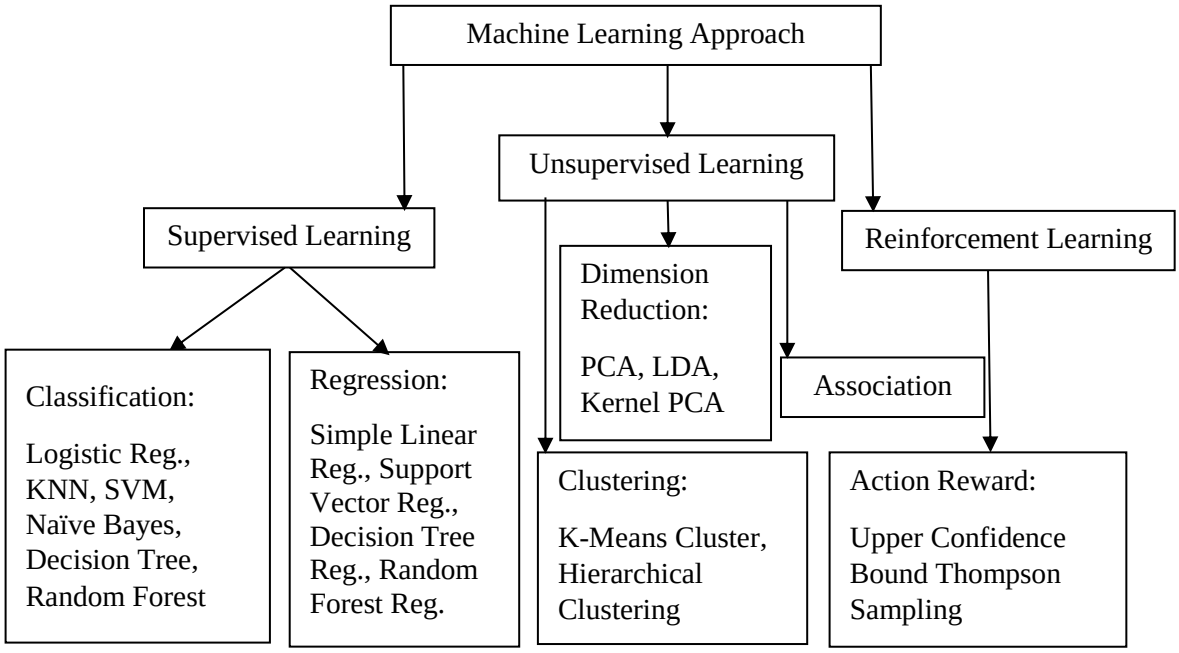


Fig. 1.6 Machine learning-based methods

1.2.1.1 SUPERVISED LEARNING

If labeled data is applied for the network's training, it is called supervised learning (SL). We can use supervised learning when a ground truth image is available for training and testing, as shown in Fig. 1.7. Thus finding forgery turns out to be a classification problem in SL. If the dataset of images does not contain any ground truth image, then Generative Adversarial Networks (GAN) is used to create it.

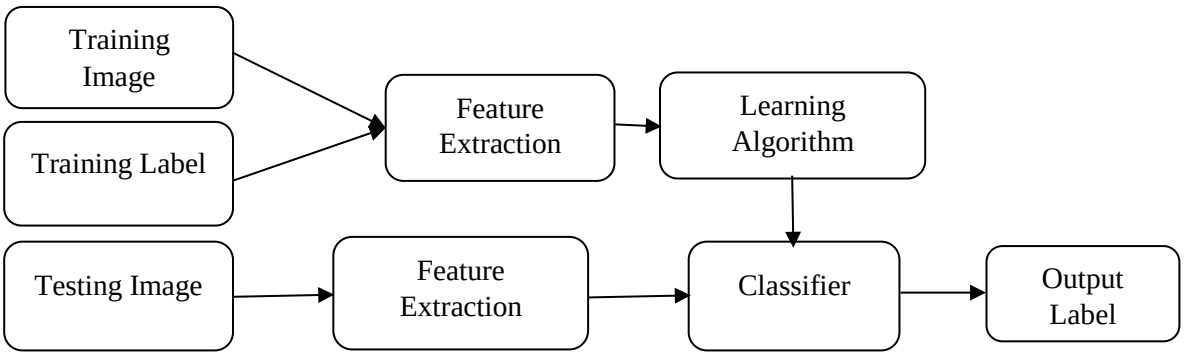


Fig. 1.7 Digital image forgery detection using a supervised learning method

1.2.1.2 UNSUPERVISED LEARNING

In unsupervised learning (UL), tagged information is not available for network training. When we have only the input image data for training, we use unsupervised learning to determine the given patterns. We cannot apply labeling to the group in this learning technique, such as a couple of fabricated or not-forged photos. Correlations are automatically identified during the training process in the dataset, and then clusters are formed in UL. With the help of patterns in the image dataset, it segregates reproduced images. Netflix uses this learning technique to divide film categories. Autoencoder and RBM use unsupervised learning. When we see a picture, we know what that image contains, but computers read photos in the form of numbers. It stores the copy into various dimensions, the color image stored with red, green, and blue pixel intensity. However, the machine does not recognize what the object is in the picture.

1.2.1.3 REINFORCEMENT LEARNING

In 1956, Arthur Samuel defined ML as that allows us to know trends without being deliberately programmed. Reinforcement learning (RL) works on the reward and penalty methodology. An agent automatically enumerates the solution and computes the bonus and penalty. An agent's role is to learn from indirect, delayed rewards and select acts, which generate the greatest cumulative reward. The agent may not have any prior knowledge of the environment. The agent interacts with the environment, and it takes action. Agent receives a penalty if the move turns out to be wrong. Agent gets a reward if the move turns out to be correct. This cycle goes on and on until it ultimately learns the environment properly. The Self-driving car is an example of reinforcement learning.

1.3 DEEP LEARNING

The input layer is the first layer in the Deep Convolution Neural Network (DCNN), as shown in Fig. 1.8. The dimension of the input image determines the size of the input layer. This input layer contains the raw image pixel values. This layer can also file normalization by deleting the second image from the training set's input image. Filters are used to configure the weights in a convolution layer. Neurons are connected with each neuron in the next layer while implementing a convolution network. Then use a kernel to move across the picture to compute the dot product. This product reduces the image size by 1/3rd between the weights matrices and the whole image matrices. The convolution layer is split into two groups after cognitive methods are applied to each layer.

Repeatedly such procedures are called chains of convolution. After convolution, RELU and pooling operations are performed. To achieve accurate performance, DCNN employs several embedded layers of Convolution, RELU, and Pooling. These several layers are responsible for producing a deep neural network.

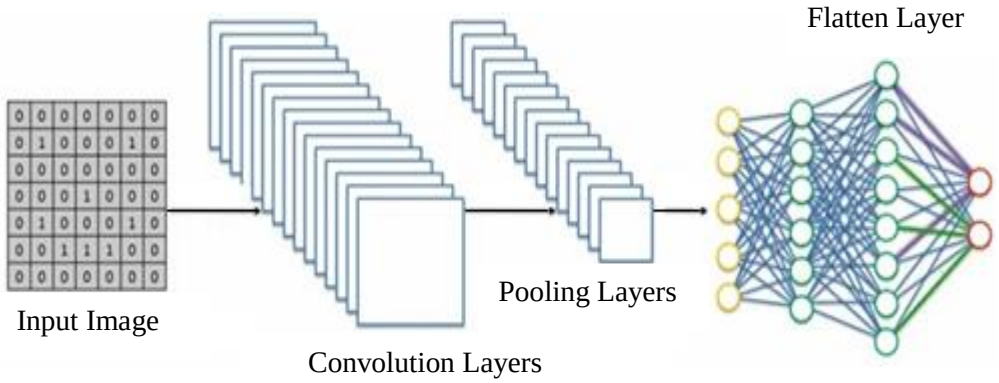


Fig. 1.8 General framework of the Deep Neural Convolution Network (DCNN) [30]

Today, most users use social media to share personal pictures and valuable information. They post billions of digital images on social media. Anyone can download these images and use them for illegal purposes to create false propaganda. Digital image tempering is now an effortless task with editing programs, the most common attacks in image rotation, scaling, JPEG compression, etc. The truth and authenticity of digital images are critical because they pose a significant threat to society.

1.4 IMAGE FORGERY USING MACHINE LEARNING

Machine learning [15] is used in various image processing fields such as image recognition, image forgery detection [16], speech recognition, traffic prediction, product recommendations, self-driving cars, email spam, and malware filtering, virtual personal assistant, online fraud detection, stock market trading, medical diagnosis, and automatic language translation. Image forgery detection [17] is a big challenge in this digital world. We observed that number of applications are using machine learning for image forgery detection. Mostly Scale Invariant Feature Transform (SIFT) [18] was used to detect features, and Support Vector Machine (SVM) [19] to classify forged pixels. The features were initially fed into the input layers. Then, these features were weighted and multiplied before being fed into the invisible layer. The bias weight was combined with these characteristics, and the output layer's weight was applied. Finally, the bias of the output layer was applied at the output layer, and the classified values were achieved. Later on, an

automatic feature extraction algorithm detects features using Convolution Neural Network (CNN) [20] and classifies these forgeries. Finally, a deep Convolution Neural Network (DCNN) [21] with semantic segmentation [22] detects and localizes forged pixels. Despite several existing algorithms, there are still challenges that are discussed in the next section.

1.5 CHALLENGES

Image forgery detection using machine learning is used for the past few years and getting promising results, but still, some of the challenges are:-

- Most of the work reported in the image forgery field is in active forgery detection. There is limited research done on passive forgery detection methods using machine learning.
- There is a scope of research in passive image forgery detection to find an improved technique with more accuracy and performance parameters.
- There are limited image forgery detection techniques verified on geometrical attacks like rotation, scaling, etc.
- There is scope to enhance the robustness of the Passive forgery detection technique against various attacks.
- The forgery extraction process takes a lot of computational time due to a large dataset in machine learning. Therefore, improving computational time is another major challenge in this research domain.

1.6 CONTRIBUTIONS OF RESEARCH WORK

The significant contributions of the presented work are as follows.

- In this research work, an improved passive image forgery algorithm for copy-move and splicing forgery is proposed. For the forgery detection, the discrete wavelet transform (DWT), color illumination Algorithm, Simple linear iterative clustering (SLIC) Algorithm, Scale Invariant Feature Transform (SIFT) Algorithm, Correlation Coefficient Map generation Algorithm, Block Matching Threshold Algorithm, and Feature Extraction Algorithm. We checked 48 images from the CMFD database and discovered image forgery detection at the image level with Precision = 97.25%, Recall = 100%, and F1 = 98.53%.
- The hybrid algorithm is proposed that uses DL and ML approaches for passive image forgery detection. The performance accuracy on the CASIA v1.0 validation set and test set is 98 and 99 percent, respectively. The performance accuracy on the CASIA v2.0 validation set and test

set is 98 and 98 percent, respectively. The accuracy of the DVMM dataset forgery detection is 97 percent. The accuracy of the BSDS300 dataset forgery detection is 98 percent. The proposed algorithm was tested on an image level on the CMFD dataset and achieved 98 percent precision (P), 100 percent recall (R), and 99 percent F1.

- Further, Color illumination, deep convolution neural networks, and semantic segmentation are used to identify and localize passive image forgeries. This algorithm determines whether or not the pixels in an image are forged. To localize forged pixels, these labeled images with color pixel labels are trained using semantic segmentation. These algorithms have been tested on datasets such as GRIP, DVMM, CMFD, and BSDS300. The experiment results show that the detection accuracy for forged and not forged pixels is greater than 98 percent.
- The robustness of the proposed work is checked on various geometrical attacks like JPEG compression, scaling, rotation, etc. The classification accuracy obtained during validation for CASIAv1.0 is 97.35, CASIAv2.0 is 97.93, and DVMM is 97.86. It has been observed that various other performance parameters like the F1 score, confusion matrix also improved with the proposed work.

Proposed work is research relevant for the machine learning community as passive forgery detection using machine learning techniques. Furthermore, a comparison with state-of-the-art proves the efficacy of work.

Limitations:

There are still some limitations in this work, like feature dimensionality and computational complexity. There is another limitation against typical geometrical transformations; however, the proposed work has proved robustness for rotation, compression, scaling, etc. The lack of a large dataset covering all possible forgery attacks is a significant challenge for the research community working in image forgery.

1.7 THESIS ORGANIZATION

The thesis consists of seven chapters, and as outlined in the chapters below:

Chapter-1: INTRODUCTION

This chapter demonstrates the historical development of forgery detection. Following that, some basic concepts about identifying forgeries using a machine and deep learning were discussed.

Chapter-2: LITERATURE REVIEW

In this chapter, a comprehensive review of related literature with their background is presented. It includes the preamble of forgery detection along with the motivation and objectives of the thesis. Furthermore, it comprises the mathematical definitions and information of machines and deep learning. Finally, based on the literature gaps, goals and methodology for the current work have been decided.

Chapter-3: IMAGE FORGERY AND MACHINE LEARNING

This chapter includes the background of image forgery and its types. Then different techniques are explained to detect image forgery using machine learning and deep learning. After that, a comparative analysis between proposed and existing methods is also presented.

Chapter-4: CMF DETECTION USING MACHINE LEARNING

In this chapter, the necessity of machine learning for forgery detection is presented. After that, the step-by-step algorithm explanation has been elaborated.

Chapter-5: CMF AND SF DETECTION USING DEEP LEARNING

In this chapter, we are using the concept of machine learning, and as an extension of deep learning, a hybrid method for image forgery detection is proposed.

Chapter-6: GEOMETRICAL ATTACKS

This chapter presents geometrical attack detection for tempered images and the accurate evaluation of different datasets. Furthermore, machine and deep learning algorithms are included here by using the proposed hybrid algorithms in the previous chapter.

Chapter-7: CONCLUSIONS AND FUTURE SCOPE

Finally, the summary of the proposed work and its possible future scope is documented in this last chapter of the thesis.

CHAPTER 2

LITERATURE SURVEY

Relevant literature is a crucial feature for every thesis. An actual survey sums up and integrates what is understood. Simultaneously, the knowledge base identifies a gap, promotes the hypothesis, closes places where adequate study exists, and uncovers areas where further studies are required. Many Block matching algorithms, Keypoint matching, ML, and DL have been derived and established in applications of different areas like CMF, SF detection, and geometrical attack detection.

2.1 MACHINE LEARNING AND IMAGE FORGERY HISTORICAL BACKGROUND

In 1943, McCulloch Pitts developed a simplified neuron model, which gives binary output by taking multiple inputs. Then Frank Rosenblatt came up with a new perceptron model. This Perceptron Model takes data and weights as input and offers decisions based on learned features. The first-generation multilayer perceptron gave by Ivakhnenko in 1965-1968. In 1969, Minsky outlined the limitations to the perceptron model that it cannot handle the ex-or problem. Around 1986 backpropagation algorithm was proposed, which solves the ex-or problem using a multilayer neural network. Finally, the theorem of universal approximation was proposed in 1989. This theorem states that you can model any continuous function if you have a multilayer neural network.

From 1969 to 1996, there was no significant progress had been noticed in artificial intelligence. From 1996 to 2006, there was some progress reported. In 2006, Salakhutdinov suggested a non - supervised deep autoencoder-based learning to achieve low dimensional data representation. From 2007-2009 further investigations of the effectiveness of unsupervised pre-training were carried out. From 2009 onward, there is a series of successes such as handwritten word recognition, speech recognition, handwritten digit recognition MNIST. General Processing Units (GPU) are used for digit recognition and in the Image Net challenge Yuan *et al.*, [5]. Artificial intelligence machines execute tasks efficiently and quickly. AI is a superset that contains ML and DL. Machine learning algorithms take input from historical data and learn by experiences. Fig.

2.1 shows the percentage error rate dropped from 16% (AlexNet) to 3.6% (MS ResNet). It detects objects better as compared to the human visual system.

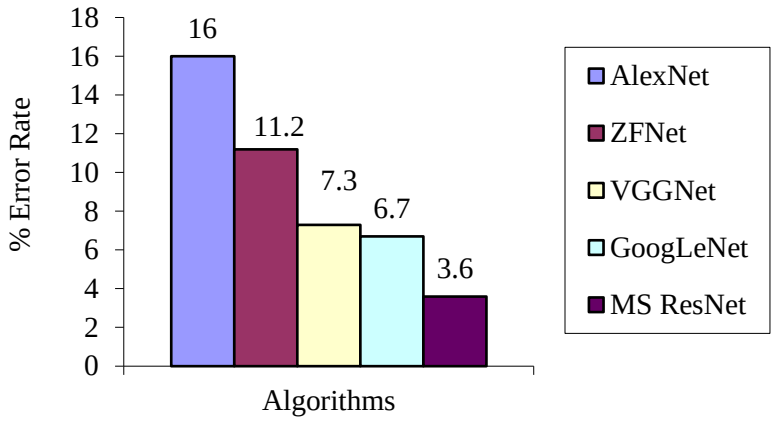


Fig. 2.1 Percentage error rate of different algorithms

In image forgery detection, the input image is analyzed using Wang *et al.*, [78] block-based techniques, key point-based techniques, Rajawat *et al.*, [54] machine learning-based hybrid methods, and Y. Zhang *et al.*, [17], F. Yang *et al.*, [18] deep learning-based approaches.

2.2 CMF DETECTION USING TRADITIONAL METHODS

Block-based CMFD methods have time complexity because the image is divided into many overlapping blocks. Chang, Yu *et al.*, [1] use the vector similarity field to identify false positives and multi-region relation (MRR) to investigate the manipulated spaces. They used the suspicious region to find similarity blocks and multi-region links to detect forged areas. Algorithms identify dubious and forged areas. The method detects inpainting images from the copy-move forgery dataset quickly and accurately. Pun, Yuan *et al.* [2] proposed a technique that divides the picture into several sections, which are not overlapping or inconsistent. The copy-move forgery was detected using block-based and keypoint-based methods. Under challenging conditions such as geometric transformations, JPEG compression, and down-sampling, the algorithm reduces computational costs and improves detection results. Singh *et al.* [3] introduced the Markov Transition Probability Matrix (MTPM). The MTPM algorithm was tested on a variety of images gathered from different social media sites. The forgery detection model has a 90.58 percent average accuracy and a lower computational complexity. Su *et al.* [4] proposed Exponential Fourier's moments (EFMs) to investigate the video frame's manipulation. The EFMs algorithm detects

region replication in videos with higher detection accuracy and computational performance and tracks the tampered regions in subsequent photos. Yuan *et al.* [5] recommend a Gaussian filter multi-scale difference to construct multi-visual charts. The algorithm excels at visual perception and is well-suited to real-time processing applications. D'Amiano *et al.* [6] suggested a dense-field technique in which the features are not varied that assure strength to various postprocessing operations. The dense-field method locates video copy-move forgeries with invariant characteristics, ensuring robustness to multiple postprocessing operations. Even in challenging environments, identify and localize video copy–moves with high accuracy. Tariang *et al.* [7] recommended the anti-forensic technique that detects the stable local connection and improves image classification. The robust residual dense network investigates the antiforensic median filtered images, which achieves superior JPEG compressed images. Elaskily *et al.*, [8] present a method that segment image into different items. Key point-based methods extract a few features, and it fails when images are smooth. Ardizzone *et al.* [24] proposed a triangle-based approach to detect features with an inner triangle angle. The algorithm finds matches between triangle features but cannot cover the entire copy-moved region. Nirmala *et al.* [9] proposed a hierarchical BIRCH algorithm that detects forgery and clusters the forged images from the COMOFOD, GRIP, and MICC-220 datasets. SIFT and SURF methods find copied areas, scaling, and translation artifacts and achieve outstanding results. Kakar *et al.* [10] offer a technique that detects motion blur using image gradients, differentiating copied areas in an image. Christlein *et al.* [11] compared different SIFT and SURF, block-based DCT, DWT, KPCA, and PCA methods. After comparison, Zernike features achieve excellent performance. Carvalho *et al.* [12] analyze machine-learning-based techniques. The machine learning-based HOG10 method extracts features. Then SVM classifier predicts a correct classification rate of 86 percent on 200 images and 83 percent on 50 images. Lyu *et al.* [13] elaborate on the image noise patterns from different camera sources. On the Columbia uncompressed image dataset, inconsistencies in local noise levels are detected to identify splicing forgery. Costanzo *et al.* [14] proposed a method to detect copy-move forgery using the detectors. In CLBA forged image datasets, SIFT-based methods detect copy-move forgery with detection accuracies of 92.5 percent, 93.5 percent, and 100 percent, respectively. Liu *et al.* [15] offer an ensemble learning approach for feature dimension reduction and prevents overfitting. This technique was used in film preservation, image and video processing, and restoration. The algorithm outperforms and effectively exposes the forgery in inpainting. Wo *et al.* [16] introduced

the multi-radius PCET to extract the rotational invariant and multi-scale features. This algorithm's applications include texture analysis and image registration. The proposed method reduces computing time by using GPU acceleration technology. Zhang *et al.* [17] offer an autoencoder to detect tampering in the social media platform. The accuracy of the proposed algorithm over 15% tampering rate is 0.885, 15% -25% tampering rate is 0.919, 25% -50% tampering rate is 0.796, and >=50% tampering rate is 0.784. Yang *et al.* [18] proposed a KAZE interest point detector combined with SIFT, which extracts additional feature points. The algorithm is suitable for use on the internet as well as in wireless application areas. This method has a low accuracy due to the image's self-similarity property. Hosny *et al.* [19] Present an exponential transform that is used for searching for similar items. The proposed method can detect both authentic and forged copy-move images with an F1 Score of 99.5 percent. Vidyadharan *et al.* [20] investigate illumination inconsistency to detect the spliced faces from DSO-1 dataset. The sensitivity of the Local Phase Quantization descriptors was 70.91 percent in the Generalized Grey World map and 65.45 percent in the Inverse Intensity Chromaticity maps. Li *et al.* [21] propose a technique to solve the critical point matching problems. The proposed technique achieves high detection accuracy on FAU, MICC-F220, MICC-F600, CMH, GRIP, and COVERAGE copy-move datasets using the SIFT algorithm. Muhammad *et al.* [22] proposed a method that describes the texture in each SPT sub-band using LBP histograms. Using the SIFT algorithm, SVM identifies and predicts binary classes with high accuracy on the CASIA v1.0, CASIA v2.0, and Columbia datasets. Neenu *et al.* [23] proposed a technique that contains illumination and reshaping properties for detecting forged images. This method provides reliable results on splicing forgery in medical images. Ardizzone *et al.* [24] presented a method, which uses triangle blocks to see features. On the self-created copy-move forgery dataset, key point-based methods perform more accurately than block-based techniques. Le *et al.* [25] propose HoG and HOGG methods to detect a forgery in an image. Faceial forgery is detected using a segmenting facial hair algorithm on the FERET and NIST Multiple Biometric Grand Challenge - 2008 (MBGC) databases. Zheng *et al.* [26] define regions' acceptable initial size to divide the picture into un overlapped sections. When detecting forgery in smooth regions, the choice of threshold has a significant impact on the performance.

2.3 CMF AND SF DETECTION USING MACHINE LEARNING

Around 2300 years ago, Plato (427-347 BC) concluded that the ability to think is found in a priori knowledge of the concept. Plato student Aristotle (384-322 BC) criticized his teacher's theory. He

said that the previous method was not considering the critical aspect of learning from the changing world. In the learning terminology, we use artificial intelligence. From Fig. 2.2, we can see that the growth of machine learning from 2016 to 2021 increase the market share from 1378.19 to 16241.52 million U.S. dollars. Software and technology usually take human intelligence functions, such as visual perception, voice recognition, decision making, and language translation. Artificial Intelligence includes the following areas of specifications: game playing, expert systems, natural language processing, neural networks, robotics, and image forensics. For example, in image forgery detection, algorithms predict whether a given image is forged or not based on the image's characteristics. A human can detect these images, but one person cannot identify all if pictures are more in numbers. Human has limited working capacity and calculative power. Computers can work 24/7 with high calculative power and efficiency. Do you want a human to learn from this data or task a machine to do so? We have a million copies of data with its labels to learn. We will feed every record to a computer that determines all the characteristics of the image.

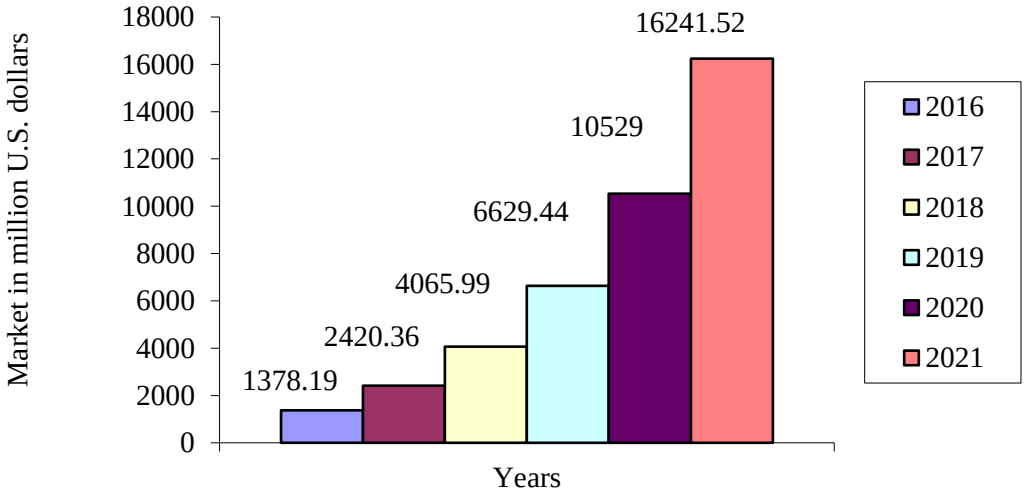


Fig. 2.2 Growths of machine learning

Carvalho *et al.* [27] propose a methodology to select visual properties to detect image forgeries. On DSO-1 and DSI-1 datasets, the proposed method detects splicing forgery and achieves classification accuracy of 94% and 84%, respectively. Ferreira *et al.* [28] proposed the ML-based technique and behavior knowledge space to detect image forgeries. This algorithm achieved an f-measure of 84.14 percent by using pyramidal decomposition on CPHNOISE 1, CPHNOISE 2, and CPHNOISE 4 copy-move uncompressed image datasets. The supervised learning system and vector support have been provided by Cristin *et al.* [29]. This method detects copy forgery on

DSO-1 and DSI-1 datasets with an accuracy of 95.23 percent, specificity of 95.83 percent, and sensitivity of 94 percent. Muzaffer *et al.* [30] propose a DL-based identification of the falsification of features utilizing Alex Net. The AlexNet convolutional neural network was pre-trained to extract image subblock features from simple copy-move forgery images. This method's average F-measure pixel-level detection rate is 93%. Liu *et al.* [31] offer a community approach to learning that recognizes multiple essential elements. To solve the crucial matching problems, they used the keypoint extraction algorithm (SIFT), a hierarchical feature point matching, and an iterative localization scheme without using any clustering or segmentation procedures.

2.4 CMF AND SF DETECTION USING DEEP LEARNING

Long *et al.* [32] proposed the DL approach to detect multiple types of forgeries in semantic categories (bridge, mountain, and sun) as well as geometric classes (horizontal, vertical, and sky) using CNN. The author used CNN on the PASCAL VOC, NYUDv2, and SIFT Flow datasets and drew links to prior models to apply spatially dense prediction tasks. The pixel accuracy of the FCN-16s was 85.2 percent, the mean accuracy was 51.7 percent, the average IU was 39.5 percent, the f.w. IU was 76.1 percent, and the geometric accuracy was 94.3 percent. Rao *et al.* [33] present a convolutional neural network (CNN) to see images' manipulation. The author used CNN for image tampering detection applications on the CASIA v1.0, CASIA v2.0, and DVMM datasets. CASIA v1.0 had a 98.04 percent accuracy, CASIA v2.0 had a 97.83 percent accuracy, and DVMM had a 96.38 percent accuracy. Bayar *et al.* [34] proposed CNN architecture to see features directly from training data. The author used CNN to distinguish between authentic and forged images on an IMDB format dataset with at least 99.31 percent accuracy. Hafemann *et al.* [35] introduced a DCNN algorithm to detect manipulation and improve forgery detection performance. On the GPDS-160 dataset, the author used DCNN to verify signatures and achieved an error rate of 2.74 percent and a Directional Probabilistic Density Function of 6.97 percent. Choi *et al.* [36] propose CNN. It studies the training images of different artifacts and mines other properties than the actual photograph. To detect composite forgery, the author used CNN for BOSS RAW and Dresden database. They used 64×64 sub-image blocks and achieved 81.93 percent on the original image and 96.50 percent on Comb. 1. Bunk *et al.* [37] propose the Radon transforms, Gaussian conditional random field model to detect features and create a heat map. The author used CNNs and LSTM on the NIST Nimble 2016 dataset for the media forensics challenge and obtained a classification accuracy of 94.86 percent and an AUC of 91.38 percent.

Alotaibi *et al.* [38] offer a DCNN technique, which generates high-level features used to distinguish between forged and original ones. On the NUAA dataset, the author used CNN to detect spoofing attacks with a 99 percent accuracy. Kurban *et al.* [39] created a virtual dataset with the score level, and the VGG network is used to extract a feature. Huang *et al.* [40] offered a CNN network that distinguishes forgery types using automatic feature extraction. On the Eurocom Kinect Face dataset and the BodyLogin Gesture Silhouettes dataset, the author used the VGG face deep learning model to achieve a higher genuine match rate (GMR) and a lower false acceptance rate (FAR). Zhang *et al.* [41] proposed a technique to generate different feature maps using Gaussian filter multi-scale difference. They used Stack Autoencoder to get the tampered image block features. The method is validated on a benchmark dataset and achieves 92.84 percent localization accuracy and a 0.9375 Area Under Curve (AUC) ranking. Liu *et al.* [42] offer CNN Kernel method to detect CMFD. On the MICC-F220, CoMoFoD dataset, the author used Convolutional Kernel Network to detect copy-move forgery. Bappy *et al.* [43] provide a technique to segment forged and not forged pieces using reshaping, LSTM, and an encoder-decoder network. Chen *et al.* [44] propose CNNs for the detection of image manipulations. The author used a CNN model to detect copy move forgery on 40,000 raw images obtained from various cameras and achieved a detection accuracy of 95.87 percent. Yang *et al.* [45] introduced an MDL-CNN to increase the learning process using in-depth learning features. The author used a multi-domain learning convolutional neural network to detect original and updated images on BOSSbase 1.01 and a laboratory database, achieving an average accuracy of more than 95%.

The other applications are optimal image restoration Kutay *et al.*, [46], sparse representation of images Koç *et al.*, [47], image fusion Meher *et al.*, [48] for dictionary learning, region-based image fusion Meher *et al.*, [49], image steganography Mahana *et al.*, [50], fog removal Neha *et al.*, [51] for foggy images, preserving privacy Kukkala *et al.*, [52], propagation patterns Cherifi *et al.*, [53], predictive analysis of medical data Rajawat *et al.*, [54], license plate recognition Kumari *et al.*, [55], secure authentication Varshney *et al.*, [56], web phishing detection Varshney *et al.*, [57], and Warif *et al.*, [58] for copy-move forgery with reflection attack.

We pointed out from the survey that standard machine learning methods have been used initially to detect falsification, and DL techniques to detect falsifiers are now used. The main reason for using the DL protocol is the automated falsification of extensive data on the Internet with less computational complexity. Everyone now has a mobile Internet phone, and therefore, they usually

upload images and information on the Internet frequently. The majority of the images on the Internet have been altered, and a new algorithm for determining their authenticity is needed.

2.5 MOTIVATION

Traditional techniques are less accurate and have time complexity. Moreover, standard machine learning algorithms have been inefficient while working with many (for example, 100, 1000, or more) data, like object recognition and localization. People call this the curse of dimensionality as they observe specific ML algorithms behave poorly when dimensions in the dataset are high. Nowadays, deep learning-based methods are widely used to find our image forgeries. The deep learning-based approach finds out real-time counterfeits on the Internet. Social media networks such as Whatsapp, Snapchat, Youtube, Instagram, etc., need such techniques to find out the image and video forgeries. In this chapter, a deep Siamese neural network (DSNN) using unsupervised learning is proposed.

From the review conducted between these four techniques, we conclude that deep learning is best among all. The block-based method has a time complexity problem. Key point-based ways are not able to extract features if images are smoothened. Machine learning-based hybrid techniques have time complexity because they use traditional algorithms to remove parts. Deep learning-based methods are widely used nowadays because feature extraction is automatically done through a neural network and produces better results than other techniques. It takes time to train the model, but the testing time is very least. For real-time forgery detection, this approach is the best of all. In this segment, we'll go over the fundamentals of falsification and how to spot them using ML and DL techniques.

Following a thorough review of the literature, it was determined that some research is conducted to detect forgery for copying and splicing. However, there is still scope for an improvement in CMF detection because of more computation complexity. Furthermore, various other methods for hybrid forgery detection are unreported. Therefore, this work led us to develop hybrid CMF and splicing methods for forgery detection with greater detection precision.

Subsequently, an interest arises to detect geometrical attacks in CMF and SF detection. Finally, the analysis describes the issues and recommends the planned work using applied research and literature knowledge.

2.6 GAPS IN THE STUDY

The previously reported literature survey aimed to develop digital forensic methods to identify digital falsification by detecting digital multimedia content editing signs. Since photographs are instant and considered factual, they have always played a key role in communicating. Digital imaging has further encouraged photographs as the most effective means to share clear messages during the last 20 years. Digital imagery, however, has also provided a great motivation to picture manipulation, and images now face a trust crisis. To provide clarity in terms of the research objectives the research questions are as follows:

- What are the different types of digital image forensics?
- What are the various approaches and modeling methods that form the basis of forgery detection and localization techniques available in literature?
- What are the strengths and shortcomings of the existing methods?
- What are the significant challenges found by the researchers in the literature?
- Why are deep neural networks preferred over traditional algorithms now?
- How will the proposed approaches be better than the existing literature techniques on benchmark datasets?
- How can performance parameters be achieved better using improved techniques?

The main statements of the problem based on identifying research gaps are:

- In the literature review, due to the widespread availability of digital image editing software (Photoshop, Corel DRAW, GIMP, etc.), it is easy to manipulate images to create a forgery, which is not detected using the naked eyes. Therefore, there is a growing need for image forgery detection algorithms to face the challenges of authenticity and integrity.
- Despite the significant advancements in image forgery detection algorithms, still, there is a demand for image forgery detection techniques that can detect one or multiple forgeries (CMF, Splicing, etc.).
- Hackers generally rotate, scale, or compress photos without permission to show connectedness between forged images and the universe. Thus, there is a great need to identify such geometrical attacks and check the algorithms' robustness.
- It has been observed that the performance of forensic algorithms is evaluated using Recall, Precision, etc. But still, there is always needed to improve these parameters to get a more efficient method.

Based on these gaps, the objectives of this thesis are formed.

2.7 OBJECTIVE

After the literature study and motivation gathered by depicting gaps, the following objectives are considered for research work:

- To propose an algorithm for copy-move image forgery detection using machine learning.
- To propose an improved hybrid copy move and splicing image forgery detection algorithm using either machine learning or deep learning.
- To evaluate the performance of the proposed algorithms based on parameters and various geometrical attacks.

CHAPTER 3

IMAGE FORGERY AND MACHINE LEARNING

Image forgery is widespread nowadays to create false propaganda. Machine learning is the emerging field for automatic forgery detection. In this chapter, the basics of image forgery and machine learning are discussed. Then, the process of forgery detection is discussed from traditional approaches to a machine and deep learning.

3.1 TYPES OF IMAGE FORGERY

With the exponential growth of digital imaging technology and user-friendly editing software, modifying their contents even for an amateur forger becomes more straightforward. Unfortunately, documented photos have increased frequency and complexity in the last decades, and various digital counterfeit instruments emerge in endless streams. The most common ones are splicing and duplication, which manipulates pictures to be hard to understand from a human-perceivable method. Therefore, finding authentic images is an important research topic since advanced image processing techniques may allow people to produce forged images quickly.

Digital falsification identification can be separated into active and passive approaches. Active methods collect primary data from a digital image to establish authenticity (i.e., digital watermarks or signatures). The source image is not accessible in passive approaches, often referred to as blind strategies. There is no need for any previous image information when detecting active forgery. The methodology is developed on the assumption that the manipulations in the still object of fossilized pictures are due to changing fundamental statistical properties. Such anomalies are used in the analysis of falsification. The most direct attack is the CMF, where a specific region (or regions) is duplicated in the picture. Image areas can sometimes be reproduced with other digital images, such as image splicing. The different algorithms to identify passive images can be clustered similarly into five classes: frame, picture, compression, computational, and physical schemes. As illustrated in Fig. 3.1, passive imaging detection techniques can be divided into five categories Chen *et al.*, [44].

First of all, pixel-based technologies detect pixel-level statistical abnormalities. For example, the presence of copy-move processing forgeries of images causes block alteration. These solutions are based on detecting statistical anomalies in image pixels produced by tampering. Furthermore, these algorithms look for connections between pixels created by a specific type of tampering in a geographic domain or a modified area. In practice, these tactics are regularly employed.

The second format-based technique utilizes statistical correlations that are introduced by a specific system of loss compression. The alteration of a fabricated image for reduction and other uses might make forgery detection extremely difficult. JPEG picture compression, for example, has been found to make fraud detection extremely difficult. However, in forensics investigation, some features of JPEG compression are used to identify tampering evidence. These techniques can be classified as JPEG quantization-based, double JPEG compression-based, multiple JPEG compression-based, and JPEG blocking-based.

Third, camera-based technology uses the artifacts introduced in the postprocessing camera lens and sensor or chip. The picture acquisition process in a digital camera system includes several phases of processing. The light enters the camera lens first, then goes via the Color Filter Array to the sensors (CFA). The sensor has an array of photodetectors that catch incident light and convert it into voltages, converted to digital by the Analog-to-Digital (A/D) converter. Today, most digital cameras use Complementary Metal-Oxide Semiconductor (CMOS) technology, with only a few manufacturers still employing classic Charged Coupled Device (CCD) technology. Next, CFA is used to obtain color images from these sensors. The sensors capture only one color, and the remaining colors are inferred via interpolations (demosaicing). The correlations introduced during the interpolation process can be used to detect tampering. Before final storage, the image quality is enhanced with procedures such as Gamma correction and white balance. Finally, the artifacts created during the various steps of the image production process are used to detect manipulation. Different camera artifacts can be estimated using chromatic aberration, source camera identification, color filter array, demosaicing artifacts, and sensor noise defects. The irregularities in these various forms of artifacts can be used to detect tampering.

Techniques based on the fourth physical environment explicitly model and detect anomalies of the relationship between physical objects, light, and the camera in three - dimensions. Natural images are typically taken in a variety of lighting settings. As a result, in splicing processes (where

two or more photos are utilized to create a forged image), the lighting of a forged region may differ from the original. In physics-based approaches, anomalies in the light source between distinct objects in the scene disclose tampering evidence.

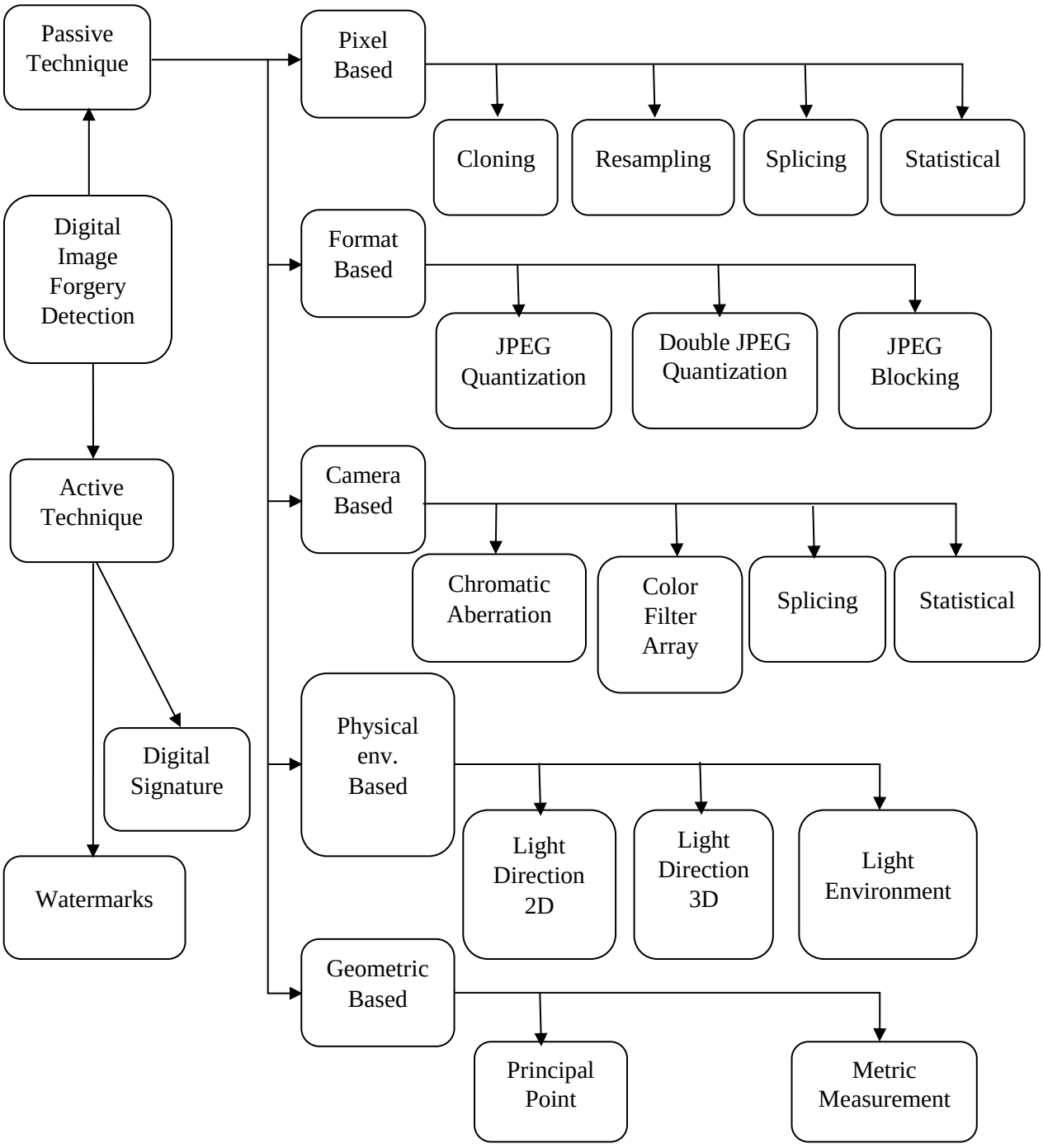


Fig. 3.1 Types of Image Forgery

Geometry-based methods calculate artifacts and their locations in comparison to the camera in the universe. Geometric constraints are used under perspective views in geometric-based forgery detection systems. These techniques are further classified as intrinsic parameters-based (such as focal length, principal point, aspect ratio, and skew) depending on the camera, metric measurement-based, and multiple view geometry-based. For example, in real photographs, the primary point (the intersection of the optical axis and the image plane) lies at the image's center. When a small part of an image is moved or translated (as in copy-move), or two or more embodiments are joined (as in splicing), it becomes challenging to preserve the main image point in its proper perspective. Robust forgery detection algorithms can thus be created using projective geometry principles.

3.2 FLOW DIAGRAM FOR CMF DETECTION

1. Image data preparation: A few procedures, including rotates, cropping, RGB color scale conversion, etc., for enhancing classification performance are carried out and tested via a test image before any feature extraction process. The generalized framework for Copy Move Forgery Detection (CMFD) is shown in Fig. 3.2.

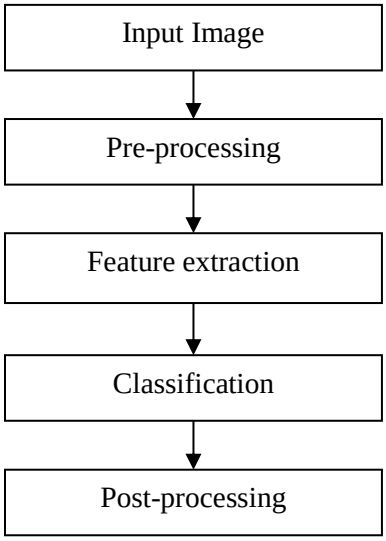


Fig. 3.2 Generalized Framework for CMFD

2. Pre-processing: An appropriate classifier has to be selected or built based on the extracted features. The classifier training includes various digital images and some of the relevant and essential classifier parameters.

3. Feature extraction: Features set are explicitly extracted for each class that helps distinguish them from all other groups. In contrast, an invariant of all attribute variations in a class from the host manipulated results.

4. Classification: A classifier split each image into two classes, accurate and distorted digital images. An appropriate performance measure is selected or designed depending on the extraction features set.

5. Postprocessing: This final step involves morphological activities that are carried out to reduce false positives. The patches with identical shift vectors are labeled with the same, usually white, to distinguish the duplicate fields and discriminate against different patches by shifting them.

3.3 CLASSIFICATION OF CMFD TECHNIQUES

The CMFD systems currently in operation are categorized into different types, as shown in Fig. 3.3. The CMFD techniques are based on blocks, key points Zhang *et al.*, [17], color illumination, hybrid methods, machine learning, and deep learning. Similar blocks are determined based on some criteria of resemblance in a block-based methodology.

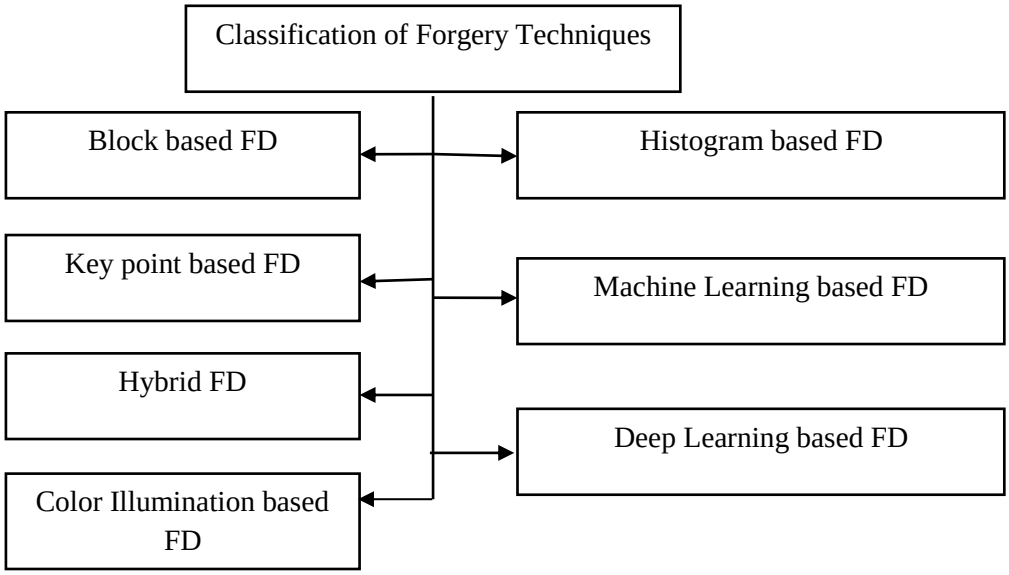


Fig. 3.3. Categorization of Forgery Techniques

The CMFD-based key point approach also uses local features of the points of interest to distinguish duplicated regions. Host images are divided into image blocks with hybrid CMFD algorithms before key point elimination. Besides the computational burden, the techniques are pretty resilient for different geometric assaults.

3.3.1 METHODS BASED ON BLOCKS

Pun et al. [2] suggested image forgery detection to identify disrupted regions using a block-based detection algorithm. The authors also subdivided the image into overlap pieces, accompanied by the DCT application on each picture to obtain the characteristics of Hosny *et al.*, [19]. Lexicographic representation was carried out to reduce computational problems. Different neighboring block pairs have therefore been considered as potential duplicate regions. A histogram was calculated for refining the results, which counted the corresponding points, which were equally distant. Eventually, a predetermined threshold value is used to eliminate false alarms and finally determine patches that have been duplicated. This system seemed to find the best compromise between sophistication and performance, but it could not detect tiny repeated regions simultaneously.

3.3.2 METHODS BASED ON KEY POINTS

The novel method combines Scale Invariant Feature Transformation (SIFT) with abstraction, which corresponds to the key arguments made by Ardizzone et al. [24]. In the instance of cloning, Bondi et al. [59] anticipated clustering around crucial locations, and Mayer et al. [60] predicted interference within the next step. This approach generated high precision, recall, consistency, and resistance to image attacks. If a suspect image exists, it could be found by the proposed algorithm, i.e., repeating particular patches. Ansari et al. [61] used a geometric change to carry out this trick, to find out image forgery. The investigation revealed efficiency in a variety of operational conditions, including multiple and hybrid clonings.

3.3.3 TECHNIQUES FOR HYBRID CMFD

Pun et al. [2] proposed a hybrid CMFD technique that includes both block and keypoint detection algorithms. Costanzo et al. [14] employed the SIFT technique to extract feature points from each patch, then compared to determine which feature points were essential. The authors also presented a method for recognizing, describing, and identifying counterfeit areas. This system outperformed previous detection methods in terms of precision and recall. To distinguish non-overlapping areas, Li employed image segmentation. SIFT was proposed by Birajdar et al. [62] for finding and extracting critical points for each patch. The Kd tree identifies a crucial point, and the nearest neighbor calculates the distance between critical points. Traditional machine learning techniques are accurate, but they need a significant amount of time to detect forgery. Deep learning methods

require a large amount of data as well as a lot of computational power. All of these conditions have now been met due to advancements in big data and processing capability.

3.4 BASICS OF MACHINE LEARNING

Machine learning (ML) learns from previous occurrences and improves the performance of intelligent systems. The ML enables robots to accomplish tasks without the need for explicit programming. Numerical and statistical approaches, such as artificial neural networks can encapsulate learning in models. Werbos provided a very thorough and generic backpropagation in his Ph.D. thesis at Harvard in 1974. In 1986, Rumelhart, Hinton, and Williams presented feedback neural network backpropagation. Hochreiter and Schmidhuber proposed the Long short-term memory (LSTM) feedback network in 1997. The LSTM algorithm was utilized in text processing, Word processing, and document processing. For example, your smartphone displays some suggestions based on the LSTM or RNN recurrent network when you type an email. Fig. 3.4 shows a simple framework for the machine learning algorithm.

Fig. 3.4 Simple framework for the machine learning algorithm

Data model and user input are the essential ingredients for data collection. Then feature extraction is used to find out the best features. Based on the features, the machine learning model learns to use some rules. Then the decision is given by defining some threshold level. Fig. 3.5 shows the mathematical model of the linear machine learning model. First, input is applied to the model, then some function operates, and finally, the estimated output is given.

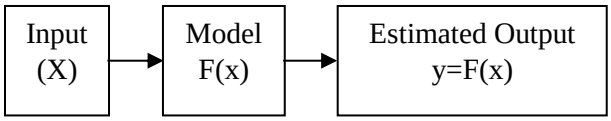


Fig. 3.5 Mathematical model of the linear machine learning model

Multiple Inputs: $y = w_0 + w_1x_1 + w_2x_2 + ... + w_Dx_D$

w_0 = bias; w_1, w_2, w_D = weights; x_1, x_2, x_D = inputs; $y = w^Tx$

$$y = \sum_{j=0}^{D} w_j x_j; x_0 = 1 \tag{3.1}$$

Multiple Outputs: $y = [y_1, y_2, ..., y_C]$

$$y_1 = w_{1,0} + w_{1,1}x_1 + w_{1,2}x_2 + ... + w_{1,D}x_D$$

$$y_2 = w_{2,0} + w_{2,1}x_1 + w_{2,2}x_2 + ... + w_{2,D}x_D$$

$w_{1,0}$=bias; w_1, w_2, w_D=weights; x_1, x_2, x_D=inputs; y_1, y_2, y_D=outputs

$y = wx; W \in \mathbb{R}$

$$y_i = \sum_{j=0}^{D} w_{i,j} x_j; i \in \{1, 2, ..., C\}; j \in \{1, 2, ..., D\} \tag{3.2}$$

A regression estimate is a real vector $y \in \mathbb{R}^C$. To find an error over multiple entries in regression, mean squared and mean absolute errors are used.

The mean square error (MSE) is:

$$MSE = \frac{1}{N} \sum_{s=1}^{N} \left(y[s] - y_d[s] \right)^2 \tag{3.3}$$

The mean absolute error (MAE) is:

$$MAE = \frac{1}{N} \sum_{s=1}^{N} \left(y[s] - y_d[s] \right) \tag{3.4}$$

Classification estimate a label $y \in \{1, ..., C\}$. Training set Inputs $\{x_1, x_2, ...\}$, Targets $\{y^d_1, y^d_2, ...\}$, test set inputs, $\{x^d_1, x^d_2, ...\}$ and unseen targets $\{y^{d,t}_1, y^{d,t}_2, ...\}$. The output (y) is given by:

$$y = \sigma\left(w_0 + w_1 x_1 + w_2 x_2 \right) \tag{3.5}$$

The loss (L) is calculated using:

$$L = \frac{1}{N} \left(y[s] - y_d[s] \right)^2 \tag{3.6}$$

In the case of classification, a non-linearity problem is not solved by a simple analytical solution. The Softmax non-linearity function is used in the multi-class category for a vector y. It is differentiable and represented by:

$$y_i = \text{soft max}(\sum_{j=0}^{D} w_{i,j} x_i); i \in \{1, 2, ..., C\} \tag{3.7}$$

$$y_i = \text{soft}\max(h_i) = \frac{e^{h_i}}{\sum_{k=1}^{C} e^{h_k}} \tag{3.8}$$

where $y_i \in (0,1)$, $\sum_i y_i = 1$. The gradient descent function overcomes this problem of non-linearity and optimizes the loss function. To reduce the loss function w.r.t. Weights (and biases) by an algorithm using gradient descent ($w \leftarrow w - \eta \frac{\partial L}{\partial w}$).

Weight values are initialized in the first step. The gradient analysis is then performed in the second phase. Weights are adjusted in the third phase. In step four, regression stops the algorithm from step two being checked otherwise. A categorical cross-entropy function is used to calculate the loss element. It is represented by (L):

$$L = -\frac{1}{N} \sum_{s=1}^{N} \left(\sum_{i=1}^{C} y_{d,i} \log(y_i) \right) \tag{3.9}$$

When L and w become constant, then we stop training and evaluate the model. Gradient descent is an iterative optimizer. It has a hyperparameter step size (η). Generally, we start with the large step size, and gradually we reduce the step size. The data normalization $x[s]$ is used to calculate.

$$x[s] \leftarrow \frac{x[s] - \mu}{\sigma} \tag{3.10}$$

where mean (μ) is

$$\mu = \frac{1}{N} \sum_{s=1}^{N} x[s] \tag{3.11}$$

variance (σ^2) is

$$\sigma^2 = \frac{1}{N} \sum_{s=1}^{N} (x[s] - \mu)^2 \tag{3.12}$$

The evaluation metric used to evaluate models is confusion matrix, accuracy, precision, recall, and F score.

$$\text{Accuracy} = \frac{\text{correct}}{\text{all}} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.13}$$

Where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative.

$$\text{Precision}(P) = \frac{TP}{TP + FP} \tag{3.14}$$

$$\text{Recall}(R) = \frac{TP}{TP + FN} \tag{3.15}$$

$$F-\text{score} = \text{harmonic mean}(P, R) = \left(\frac{P^{-1} + R^{-1}}{2} \right)^{-1} = \frac{2PR}{P + R} \tag{3.16}$$

3.4.1 IMAGE FORGERY DETECTION USING MACHINE LEARNING

The steps involving image forgery detection using traditional SIFT, SURF algorithms are shown in Fig. 3.6.

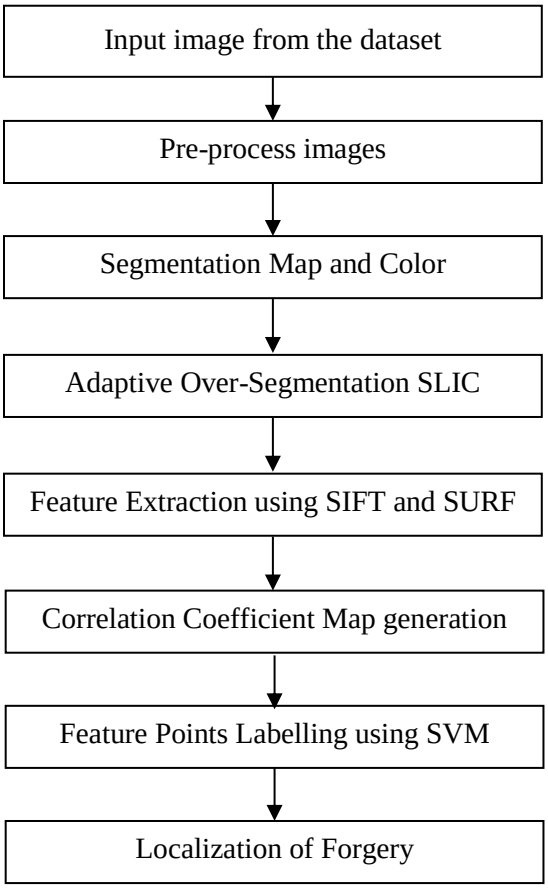


Fig. 3.6 Flow diagram for image forgery detection using Machine Learning

1 **Pre-processing:** In this, the pre-processing image is taken from the CMFD dataset, then apply normalization, segmentation map Li *et al.*, [63], and color illumination.

2 **Adaptive Over-Segmentation using SLIC algorithm:** In this step, divide the input image into small blocks. Then adaptive over-segmentation algorithm divides the input image into non-overlapping blocks.

3 **Feature Extraction using SIFT and SURF algorithm:** Two algorithms SIFT and SURFs, extract features from non-overlapping blocks. In addition, these algorithms extract standard features from the output.

4 **Correlation Coefficient Map generation:** The correlation coefficient map is generated after extracting standard features from forged images using SIFT and SURF algorithms.

5 **Feature Points Labelling:** With the help of a correlation map, feature points are labeled using a support vector machine.

6 **Localization of Forgery:** Morphological operation highlights the labeled forgery pixels, as shown in Fig. 3.7.

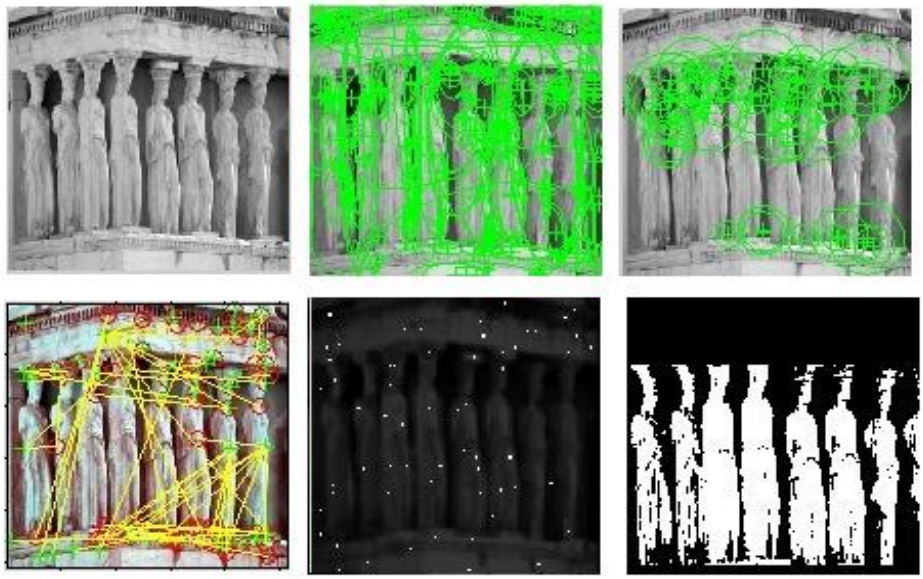


Fig. 3.7 Results of Image forgery detection using Machine Learning

3.5 BASIC OF DEEP LEARNING

The essential difference between machine learning and deep learning is that feature extraction is manual, but feature extraction is automatic for deep learning. Deep learning is a method used to carry out machine learning. Deep learning uses specialized techniques involving multilayers (2+) artificial neural networks Majumder *et al.*, [64]. Layering allows cascaded learning and abstraction levels (e.g., line, shape, object, and seen). Deep learning uses a neural network Kim *et al.*, [65] analogical to the brain. Deep learning is a different architecture that decides with a neural network Qian *et al.*, [66]. All those activities or problems that machine learning algorithms could not solve until deep learning Ouyang *et al.*, [67] can solve these problems. ML algorithms

like Regression and logistic Regression still they are very much relevant in DL. These algorithms are building blocks for DL Bayar *et al.*, [68], as shown in Fig. 3.8.

Pre-processing of data is performed in both ML and DL architecture. The first step is pre-processing in the algorithm to detect and localize image forgery. Pre-processing is used to improve the classification performance of image falsification. The next stage is creating two dataset groups and the grouping of data with the appropriate labels. The data are divided into photos and names. It consists of training and testing datasets. The third step is to extract features for every class from the training set. For feature extraction in training, appropriate classifier and hyperparameters need to be chosen. The training set uses 70% of the images. In the fourth step, images are tested on test data for forged and not forged categories. During the last point, pixel-based semantic segmentation is performed to distinguish the counterfeit from actual color pixels. Black (not fabricated) color is labeled for similar pixels, and white is segmented (forged).

3.5.1 IMAGE FORGERY CLASSIFICATION AND DETECTION USING ML AND DL

Copy move forgery is the type of forgery in which one part of the image is copied and pasted on the other part of the same image to duplicate or hide information. The main reason to create copy-move forgery is to hide helpful information and make false propaganda Redi *et al.*, [69]. Therefore, this section emphasizes copy-move, splicing image forgery detection, and localization in this section. The significant steps for forgery classification and localization are given below.

1. Pre-processing: All the images are resized and labeled into the proper size (116×116×3). Then, apply the segmentation map and color illumination to each image.

2. Prepare Training and Validation Dataset: This step extracts all the image color information for category labels. Next, generate training and validation set from the categories of forged and not forged images. Genuine and fabricated type picture data set loading (80 percent for training and 20 percent validation). The CASIA-1 and COMOFOD dataset of copy-move and splicing are taken for the experiment. In the second step, colored information and category labels are extracted to generate training and validation datasets.

3. Setup of CNN: Input layer of CNN with no data augmentation, and it has a bank of filters. In CNN Chen *et al.*, [70], the convolutional layer has a bank of filters, padding of four pixels, a ReLU layer, a max-pooling layer with a 5x5 spatial pooling area, and a Stride of two pixels.

4. The Deep CNN: Deep Convolution Neural Network (DCNN) Sreela *et al.*, [71] is nothing, but multiple layers are added with existing CNN. In this step, multiple convolution layers, ReLU layers, and max-pooling layers were added with a 5x5 spatial pooling area and a stride of two pixels.

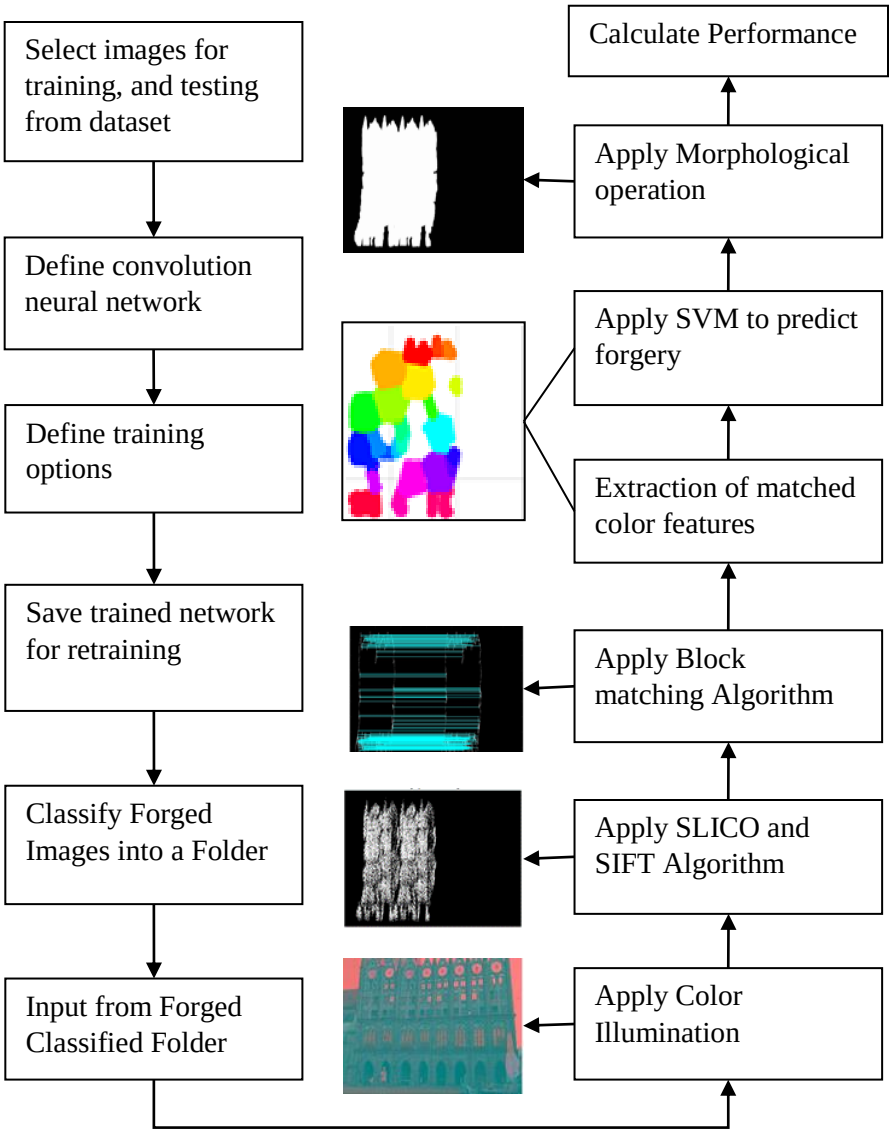


Fig. 3.8 Block diagram for image forgery classification and detection using hybrid machine learning and deep convolution neural network

5. The Fully Connected Layers: Add a fully connected layer Zhou *et al.*, [72] with two output neurons. In this experiment, the SoftMax loss layer generates the maximum value of the forged categories, and the classification layer classifies the image category.

6. Train the Model: Training Options are stochastic gradient descent with momentum is 0.9, Initial Learn Rate is 0.001, Learn Rate Schedule is piecewise, Learn Rate Drop Factor is 0.25, Learn Rate Drop Period is 20, L2 Regularization is 0.004, Max Epochs are 60, Min Batch Size is 20, Verbose is set to true.

7. Validate the Model: This step is essential because if the model is trained with 100% accuracy, but in the validation step, it shows 50% accuracy, then the model is underfitting. If the model is trained with 50% accuracy, but it shows 100% accuracy in the validation step, then the model is overfitting. The model should be conditioned to the equivalent proportions of each picture of the group. To validate the model on 20% of unseen images are used.

8. DCNN classifier: The classification output is shown in Fig. 3.8. In this Fig.3.8, the trained model prediction shows forged and accurate classification. The steps are the same as machine learning, in which we extract the color information for authentic and forged category labels. Then generate a training and validation set. Images are loaded from the dataset as 80% for training and 20% for validation. In these steps, all the process remains the same as shown in Flow chart. After these steps, our system correctly classifies the type of forgery.

9. Setup of the Feature Extraction: In this step, all forged images are passed through a machine learning algorithm for localization. The forged image is passed through SLICO, Block Feature Extraction using SIFT Algorithm, correlation coefficient map generation, block matching threshold calculation, matched blocks location, and feature points labeling algorithms. The feature selection procedure is then utilized to reduce system and time complexity by eliminating the insignificant feature before classification and localization.

10. Forgery Localization: Local color feature extraction use SVM to classify correct feature boundaries. The purpose of the classifier is to discriminate against the given images and form original and forged images. According to the steps described above, the commonly used structure of a blind copy-move and splicing forgery detection system for digital images is given in Fig. 3.8.

3.5.2 IMAGE FORGERY DETECTION USING SIAMESE NEURAL NETWORK

In this section, real-time image forgery detection using a deep Siamese neural network is proposed. The main reason to use the Siamese neural network is that we have fewer data and no output labels. To find image forgery CASIA 2.0, and CMFD datasets are used. Then, we apply a series of operations, as shown in Fig. 3.9, 3.10, and 3.11, to extract forged patches from the image.

3.5.2.1 Pre Processing:

This approach performs copy move and splicing image forgery detection on CASIA 2.0, DVMM, BDSS, and Colombia datasets. All these datasets have different sizes of images Bashar *et al.*, [73]. All forged images are placed in one folder. We need a pre-processing image technique to match all images into the same size. Then, color illumination is applied to all images. The color illuminated image edges and corners are detected quickly. As shown in Fig. 3.9, the Harris detector detects strong corners and converts these into a .npy array of anchors, positive and negative. These arrays are of size (30×30×3) are then passed to the autoencoder.

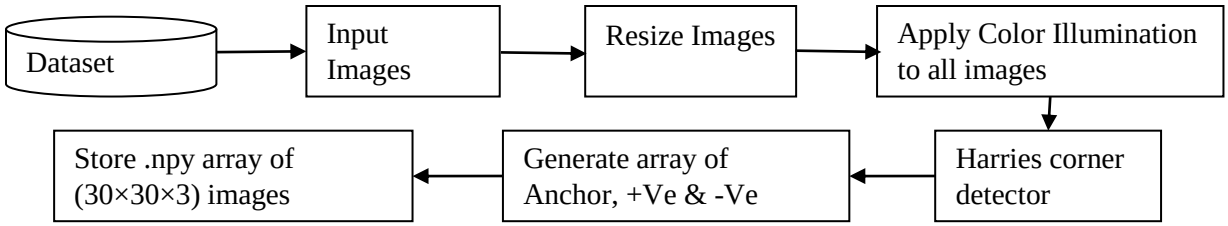


Fig. 3.9 Block diagram of Image Processing in Unsupervised Learning

3.5.2.2 Auto Encoder:

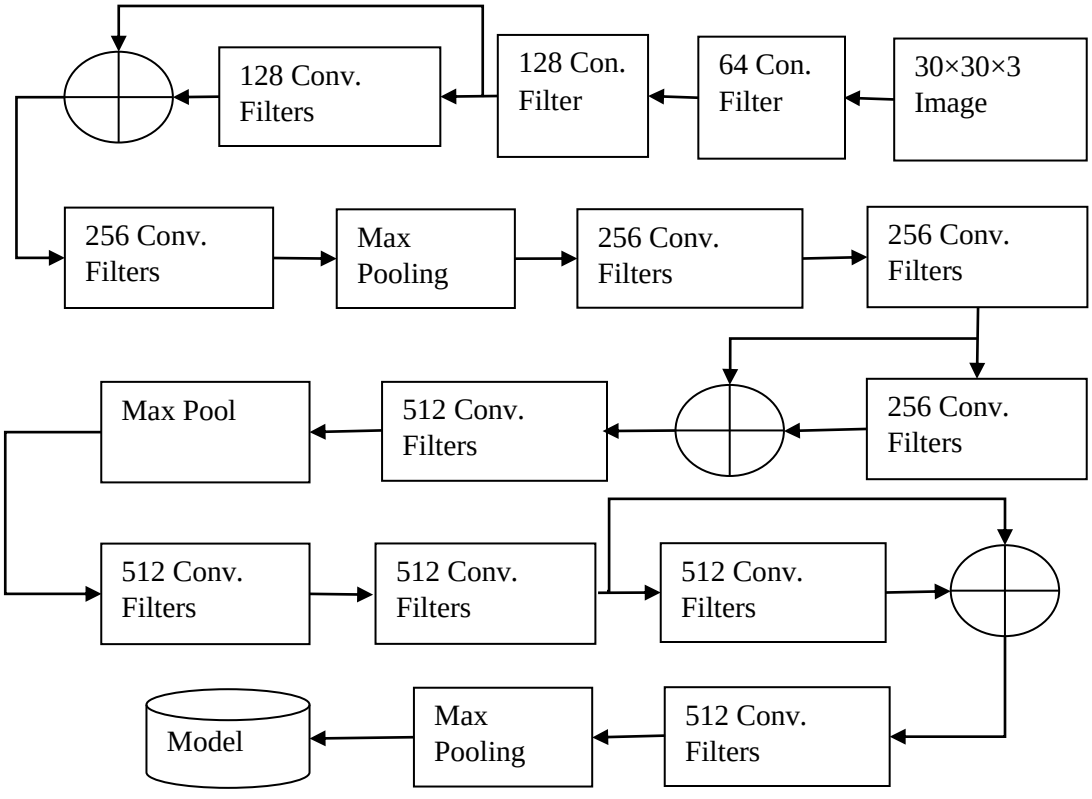


Fig. 3.10 Block diagram of the autoencoder in unsupervised learning

Autoencoder uses a nonlinear transformation to reduce the number of dimensions. The types of autoencoders are convolutional, deep, variation, and denoising autoencoders. It comprises of encoder, code, and decoder. The convolution neural network is used in this approach because it performs better for images than others. First, it encodes the input into simple signals. Next, it comprises multiple convolution layers followed by an adder from output three and output 4 with max Pooling. Then, it downsamples the input image up to the maximum point of compression. Finally, the code determines which part of the image is essential. The hyperparameters of the autoencoder are code size, number of layers, number of nodes per layer, and loss function. We used 128 batches, in which each batch contains ten images for 2000 epochs with a 1e-4 learning rate. First, the input image is reshaped into 30×30×3 patches and normalized by dividing 255. The convolution layer uses a stride of one, activation as ReLU, and padding as same. The complete flow of the program is shown in Fig. 3.10.

3.5.2.3 Auto Decoder:

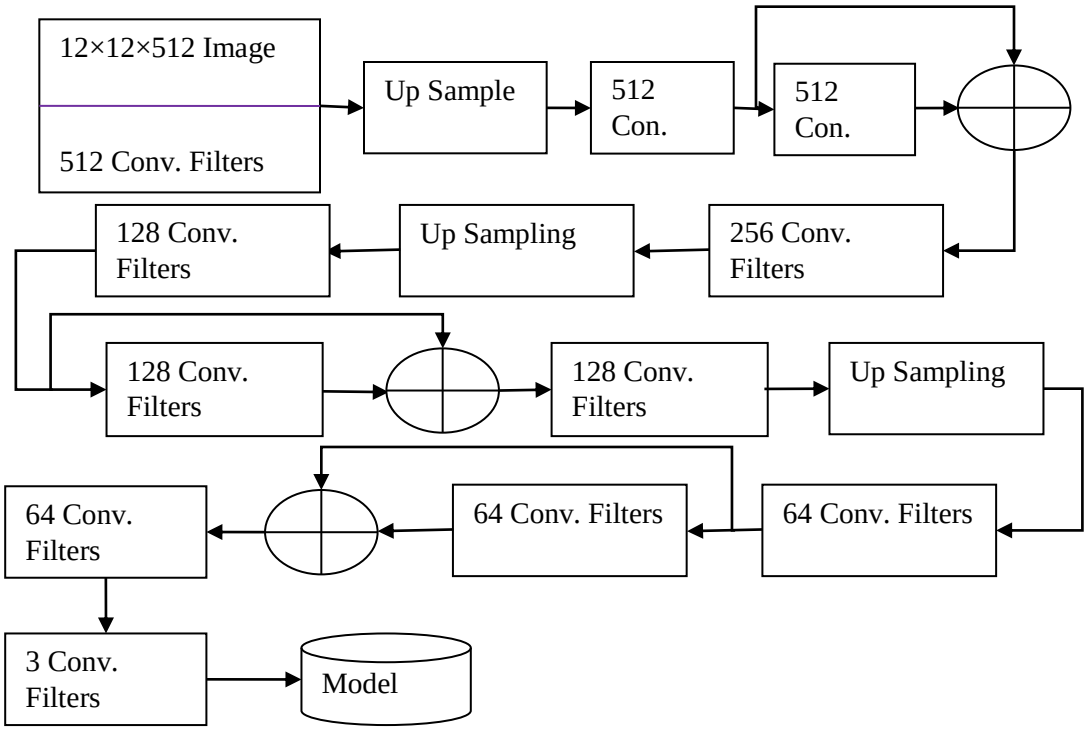


Fig. 3.11 Block diagram of the auto decoder in unsupervised learning

An auto decoder is used to reconstruct images from the minimum most essential pixels. It can replicate the output image into an input image with some degraded quality. It comprises multiple convolution layers followed by an adder from output three and output four with upsampling. The

convolution layer uses stride of one, activation as ReLU, and padding as valid. In the final convolution, layer activation is used as a sigmoid with padding as same. The image reconstruction process is shown in Fig. 3.12. The training process used mean squared error to find a loss between positive, negative, and anchor images. Adom optimizer is used to optimize the model. We used CoMoFoD and CMFD image datasets, 80% data for training and 20% data for testing. The training weights are stored for encoder and decoder with .h5 extension. The trained patches of positive, negative, and anchor are stored in .jpg format, as shown in Fig. 3.12.

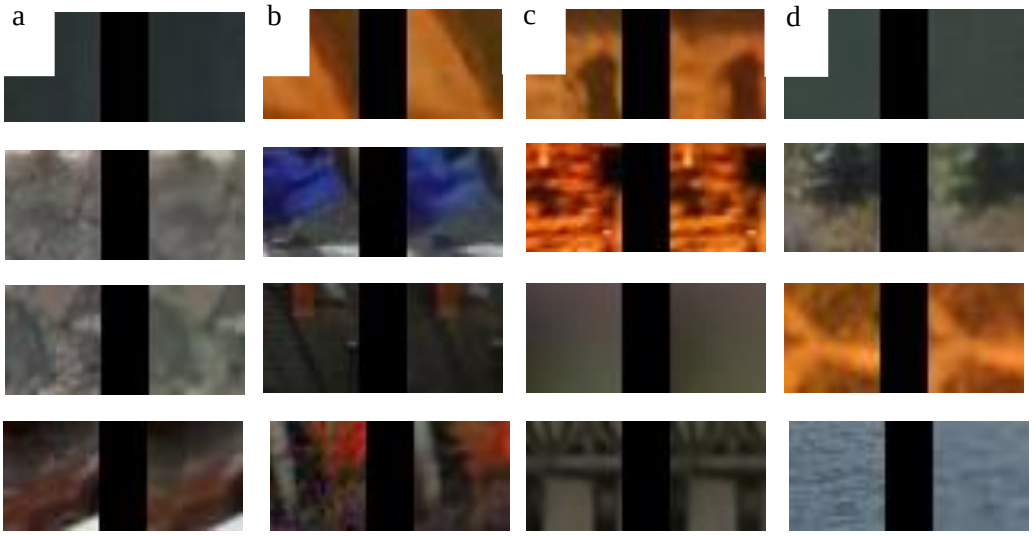


Fig. 3.12 The output of the Siamese neural network for the training of positive and negative triplets in columns (a), (b), (c), and (d)

3.5.2.4 Localization of Forgery:

The output of the Siamese autoencoder/decoder is trained using triplet loss. These triplets are trained using a positive and negative triplet distance margin of 0.6. There are 100 epochs between significant decreases in the learning rate. The learning rate decay factor is 1.0. L2 regularization was used with a random seed of 666. The number of images per batch is 20. The number of batches per epoch is 1000. The number of epoch to be run is 500. The number of images per group is 6. The number of images to be processed per batch are 99. We use the ADAM optimizer for the optimization of the results. The exponential decay for the tracking of training parameters is 0.99. The initial learning rate is 1e-5. These triplets are passed to localize the forged region. The forgery validation process finds the score of forged pixels and the distance between embedding. The final output of the fabricated pixels is given when the performance is greater than the threshold. This algorithm shows the output, as shown in Fig. 3.13. Machine learning depends on the massive stores

of training data. If you have more amounts of data, only then will it be able to give correct accuracy. It will create only one model for a particular task. It has time constraints as it has to learn much historical data.

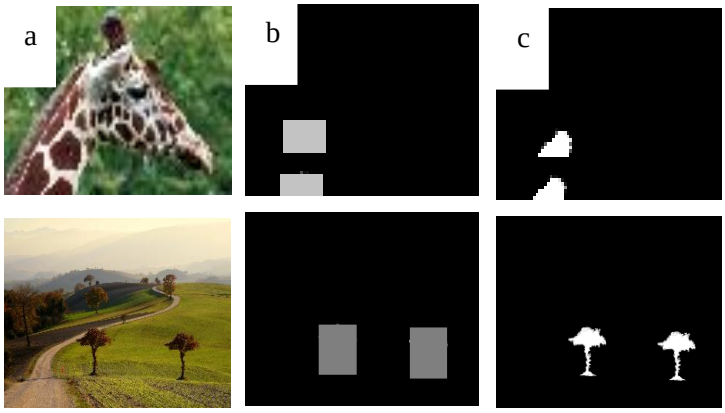


Fig. 3.13 The output shows the results of forgery detection for the CMFD dataset. Column (a) shows the original image, column (b) represent forgery detected image, and column (c) shows the ground truth image

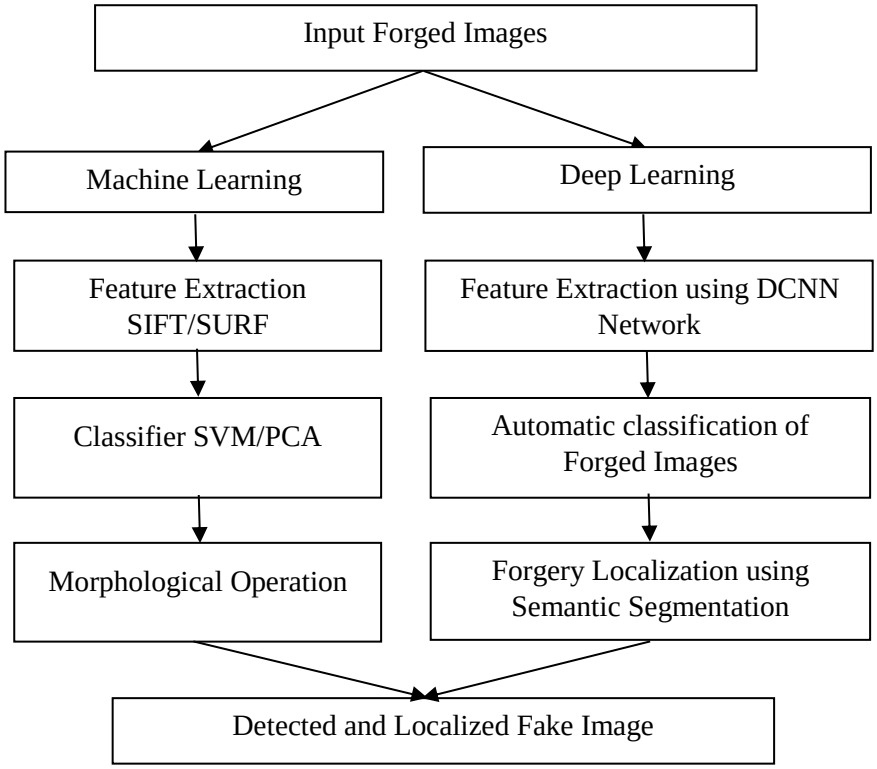


Fig. 3.14 Basic block diagram of the machine and deep learning methods

First of all, a large amount of information is needed for machine learning. In machine learning, we extract features using SIFT, SURF, HOG, etc., algorithms. These feature vectors are fed to

classifiers such as SVM, SVD, K-mean to classify, as shown in Fig. 3.14.

3.5.3 ML AND DL COMPUTATION REQUIREMENTS

Deep learning, by itself a subset of machine learning, is part of artificial intelligence. Scientists had started to study Deep Learning in the 1950s and devoted significant resources for the next seventy years. However, the foundation for the current era was late in the 1980 and 1990s. Research Yann LeCun [91] developed a convolutional neural network, the most significant achievement in this era. In 1980, the backpropagation algorithm [91] was proposed for training image processing. The backpropagation algorithm learns from its mistakes by leveraging the chain rule of derivatives. Neural winter starts from 1998 to 2007 because of a lack of computation power and data. In 2006 Hinton [91] define deep learning. In 2007, GPU training was possible, and due to the easy availability of the Internet, digital cameras and mobile phones create big data. In 2009 Fei-Fei Li [92] proposed ImageNet, the first convolution neural network for image processing applications. In 2013 Alex Krizhevsky et al. [92] proposed ALEXNET, and in 2015, Kaiming He et al. [92] proposed 154 layers, the RESNET model. Later in 2018, we have a deep reinforcement learning [92] offered by the Deep Mind project of Google. They designed neural networks to play the Atari game. To play games, we first have to understand the game, and we start learning the game. The computer and the human brain are mapped in this game.

Nowadays, Tensor Processing Units (TPUs) are becoming faster than General Processing Units (GPUs). Due to the reduction in computation time using GPUs and TPUs and data availability, a deep neural network performs exceptionally well compared to traditional methods. Artificial neural networks are different from traditional algorithms as they work like the human brain. They are more flexible, have better handling capabilities, and anticipate abnormalities and nobilities in data. Deep learning uses non - linear layers of processing units to derive and convert functions. It successively uses the output from the previous layer as an input to the next layer.

3.6 SUMMARY

The main features of this chapter are:

This chapter gives a basic understanding of image forgery, machine, and deep learning. First, the step-by-step approach is shown to find out image forgery from traditional algorithms. Then advance machine and deep learning algorithms are explained. After that, some more methods are discussed, like supervised, unsupervised learning for image forgery detection.

CHAPTER 4

CMF DETECTION USING MACHINE LEARNING

Everybody now posts vital communication records and images via social media. Viewers can easily import and change these pictures to create false propaganda. Such pictures are being used as evidence by visual forensic experts in the courtrooms. These types of forgery are challenging to detect. For the identification of this form of imitation, continuous research efforts are required. So it is the need of the hour to have an effective and efficient method to verify the photograph's authenticity. For picture forgery detection, color illumination techniques have been proposed. This method is used at the stage of pre-processing. A technique for distinguishing faked locations, integrating color lighting, blocks, and required points-based methods, has been presented in this chapter. According to experimental results, the proposed methodology gives better results under simple CMF as Precision=97.25%, Recall=100%, and F1=98.53% of the original high-resolution images.

4.1 BACKGROUND OF COPY-MOVE FORGERY

The picture detection algorithm reads an input picture first and transforms it into grayscale. This grayscale is used to handle segmentation with DWT. DWT transforms the host photo to LL, HL, LH, HH. In the original RGB image, color improvement takes place, and this image is applied for superpixel computing using SLICO in combination with a DWT output picture. A block feature extraction algorithm is used to detect image features. The MSER and SURF components are identified in this algorithm. Block matching technology is used to suit the observed characteristics by choosing the correlation coefficient, block matching threshold and then comparing these characteristics Akbarpour *et al.*, [74]. A morphological operation identifies the suitable component of forging. Human intervention for minor superpixel substitution is required in this phase. To measure the performance of the proposed algorithm, precision, recall, and F1 values are calculated.

$$\text{Precision} = \left(\frac{\text{correctly detected forged pixels}}{\text{total detected forged pixels}} \right) \tag{4.1}$$

$$\text{Recall}=\left(\frac{\text{correctly detected forged pixels}}{\text{forged pixels}}\right) \tag{4.2}$$

$$\text{F1}=2\times\left(\frac{\text{Recall}\times\text{Precision}}{\text{Recall}+\text{Precision}}\right) \tag{4.3}$$

4.2 PROPOSED CMF DETECTION ALGORITHM

This chapter analyzes the hybrid techniques that identify image forgery within data sets and integrate them according to pixel-level norms.

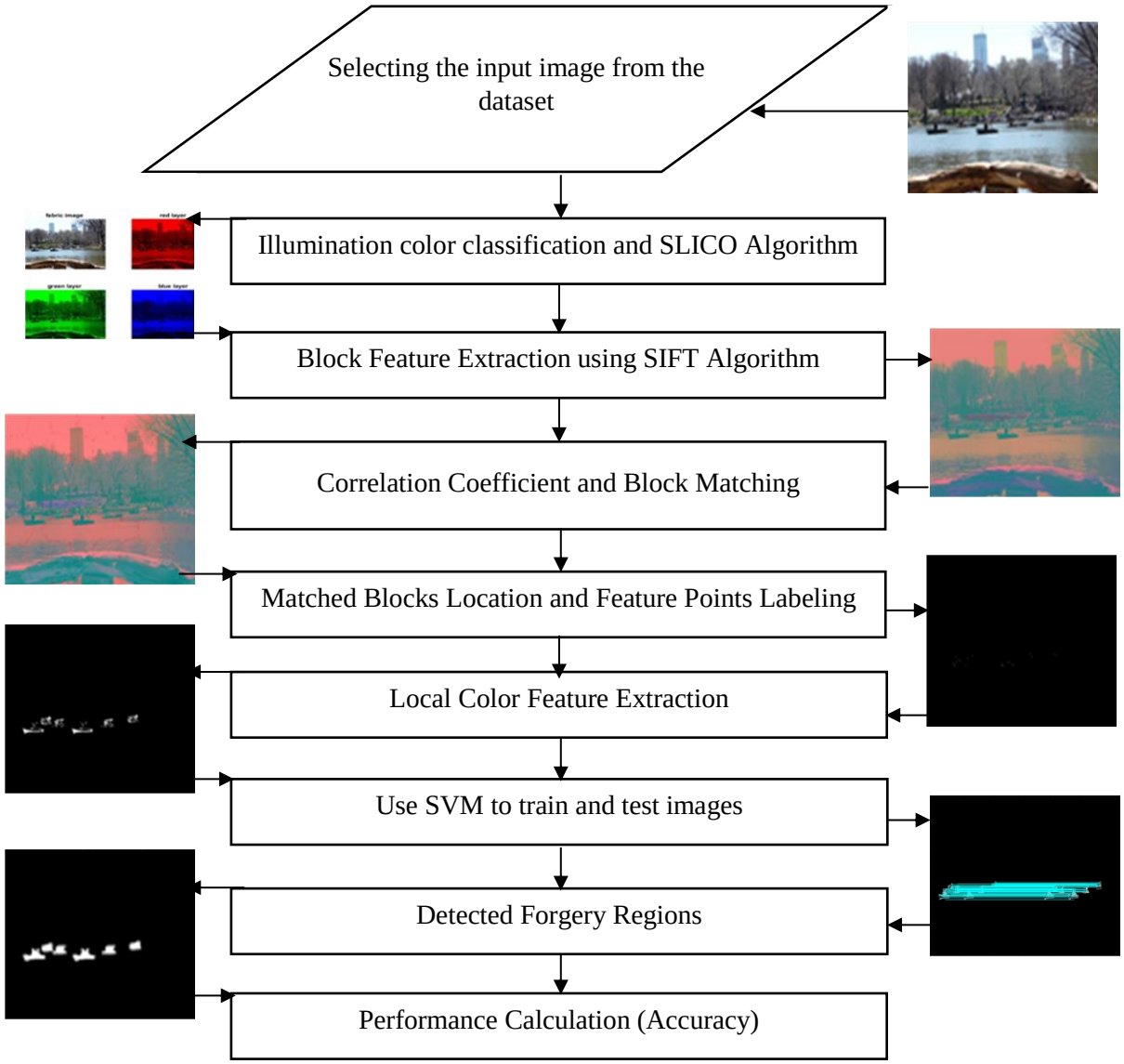


Fig. 4.1 Flow chart of the proposed forgery detection algorithm

Fig. 4.1 shows the flow charts for the proposed falsification detection method. We use MATLAB and a collection of forged images Christlein et al., [11] to figure out forgery using the proposed algorithm. The total number of images in this database is 48. We analyzed 48 JPEG compressed and original images with different compression factors, for example, 20, 40, 60, 80, and 100. We also tested and compared these findings with Pun et al., [2] in precision, recall, and F1 values.

4.2.1 DATASET DESCRIPTION

We select an image from the dataset Christlein *et al.*, [11]. There are 48 images of different sizes in this database, and manipulated images are provided to compare results. This dataset contains 48 images of different sizes and modified photos to evaluate the results. In this step, the train and test folder have been created in the MATLAB working directory. For the training and testing of our program, we have used SVM. Once accessing the database, we need to select the image to make the correct choice manually.

4.2.2 COLOR ILLUMINATION

At this stage, the color lighting maps were implemented to suggest a new hybrid technique to detect falsification by selecting visual properties. The purpose here is to detect irregularities in the image's original color, combine color, shape, and texture. The identification of splicing is very time-consuming and erroneous as compared to copy-move forgery [17]. We address the identification of the picture by several copy-paste parts. We generated an SVM train and test database, as shown in Fig.4.1, in which images are divided in similar colors and stored into interactive vectors, with algorithms calculating the color region of the input image. The features that compose the representation to train the SVM are forged and non forged pixels. SVM integrates image feature vectors to learn image patterns among classes to define each new vector. The SLIC Algorithm for block extraction must be used to choose such superpixels to identify the falsifying component and approximate lighting color. Also, we should use the block matching algorithm to calculate the distance. After that, the decision to identify falsification is taken using SVM classifiers automatically. We have discarded isolated pixels through morphological operation, leaving boundary pixels and creating a segmented region for falsification. Since we have read and checked various articles, color space is not used to examine image falsification. To detect the slightest variance, we seek to increase the number of testing color spaces. This work uses the Lab,

HSV, and RGB uniform color spaces Bashar *et al.*, [73], Nguyen *et al.*, [75]. We adopted the extension of this concept by van De Weijer et al. [76], Carvalho *et al.* [27] using these color spaces.

4.2.2.1 ILLUMINANTS FOR DETECTING FORGERIES

$$\text{Let } f(\text{pixel}) = \left(f_{red}(\text{pixel}),\ f_{green}(\text{pixel}), f_{blue}(\text{pixel})\right)^{T};\ \text{RGB color;} \tag{4.4}$$

F (pixel) is the obtained red, green, and blue color of a pixel at location' pixel'. Van de Weijer *et al.* [76] assumed no reflection, and the camera has a linear response. Then, f(pixel) is formed by

$$F(\text{pixel}) = \int_{\Omega} e(\lambda,\ \text{pixel}) s(\lambda,\ \text{pixel}) c(\lambda) d\lambda \tag{4.5}$$

Where '**Ω**' is the gamut of visible illumination. 'λ' is the wavelength of the illumination. 'e(λ, pixel)' is the spectrum of the illuminant. 's (λ, pixel)' is the surface reflectance of an object. 'c(λ)' is the color sensitivities of the camera (i.e., one function per color channel). Put these values in the van de Weijer *et al.*, [76] model to estimate the illuminant's color, as shown in the equation below.

$$ke^{n,p,\sigma} = \left(\int \left| \frac{\partial^n f^{\sigma}(\text{pixel})}{\partial x^n} \right|^p d\text{pixel} \right)^{\frac{1}{p}} \tag{4.6}$$

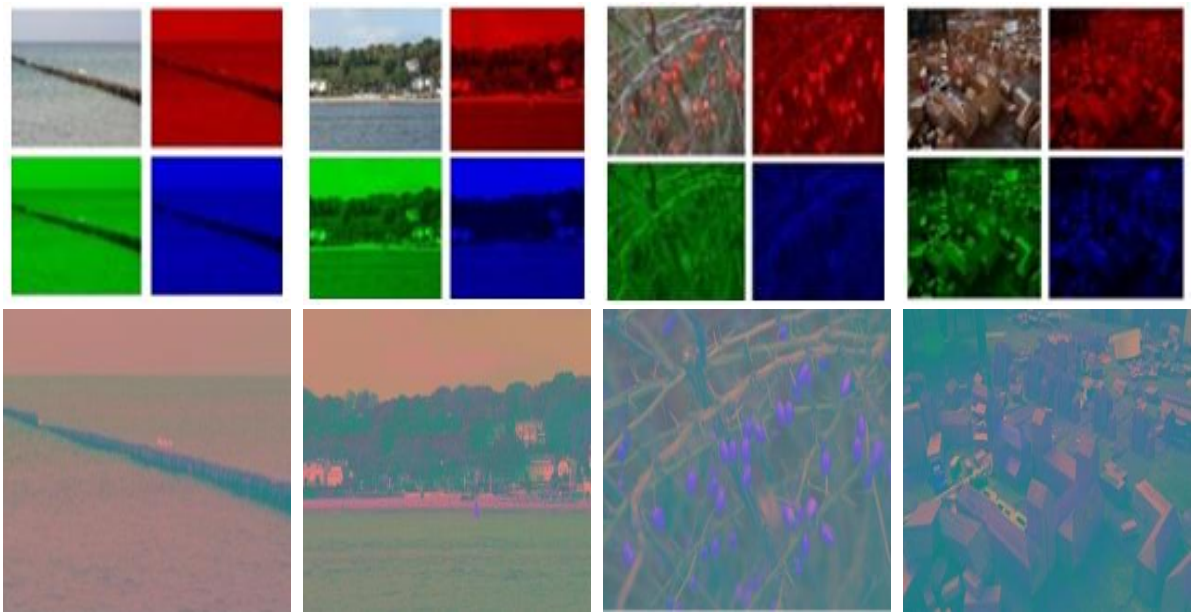


Fig. 4.2 Example maps for forged images (top row) in red, green, and blue, as well as Color illuminations, chart generated for the forged image (bottom row)

From this equation, the integral is calculated from the image overall pixels, where x is pixel coordinate, k is scaling factor, |. | is the absolute value, ∂ is the differential operator, and f^{σ}(pixel) observed intensities at position' pixel,' smooth with a Gaussian kernel ∂ . These colored spaces are an input to the system we suggest. DWT is then used to produce high-frequency and low-frequency image components. On the input image, DWT is decomposed with the Haar transform. With the help of steps 1, 2, 3, 4, and 5, we calculated the low-frequency energy (ELF), high-frequency energy (EHF), percentage of the low-frequency distribution (PLF), initial size S of the superpixels are calculated respectively. Step-4 determines the scale of superpixels S if the PLF is greater than 50 percent. The scale of the superpixel S is determined using step-5 if the PLF is less than 50%.

4.2.2.2 ADAPTIVE OVER-SEGMENTATION

Input: Take Color of the illuminant Image.

$ELF = \sum|CA4|$

$EHF = \sum (\sum|\text{CDi}| + \sum|\text{CHi}| + \sum|\text{CVi}|)$; i=1, 2,…, 4

$PLF = \frac{\text{ELF}}{(\text{ELF} + \text{EHF})} * 100\ \%$

$S = \sqrt{(0.02 \times M \times N)}$; $PLF > 50\%$

$S = \sqrt{(0.01 \times M \times N)}$; $PLF \leq 50\%$

Calculate initial size $S = M \times N$ of the host image.

Output: Adaptive over Segmented Image.

The explanation for transforming fabricated images to an image block of unequal scale is shown in Table 4.1. These fake pictures are shown in Fig. 4.3 (top row), and a varying picture block size (bottom row) is indicated with the accompanying superpixel.

Table 4.1 Image superpixel calculation for I1, I2, I3, and I4 forged images

Sr. No.	After DWT Image Size (M*N)	ELF	EHF	PLF	S
I1	267 × 400 = 106800	*ELF* = 134	*EHF1*= 2.000	*PLF1*=98.5294	S1=46.2169
I2	1007 × 1520 = 1530640	*ELF2* = 106.00	*EHF2*=-2	*PLF2*=101.9231	S2=174.9651
I3	1296 × 1944 = 2519424	*ELF3* = 140	*EHF3*=2.000	*PLF3*=98.5915	S3=224.4738
I4	791 ×1181 = 934171	*ELF*4 = 296.00	*EHF4*=-4.000	*PLF4*=101.3699	S4=136.6873

Fig. 4.3 Convert image into an irregular block size (original images top row) and (adaptive block size bottom row)

4.2.2.3 SIMPLE LINEAR ITERATIVE CLUSTERING (SLICO)

The entire image is branched into smaller blocks with the aid of the SLICO algorithm. SLICO is used for generating pixel values starting through a signal of length N. In this process, the image data is separated into several non-convergence areas of unknown shape. Fig. 4.4 demonstrates that the input image is broken into several uncertain shapes that do not overlap. A forgery region algorithm is implemented to match these non-convergence and unstable areas. The SLICO is used as a base to create and list superpixels of irregular shapes in the image that are not overlapping. K-means clustering approach generates and displays superpixels that quickly and accurately identify edges and boundaries.

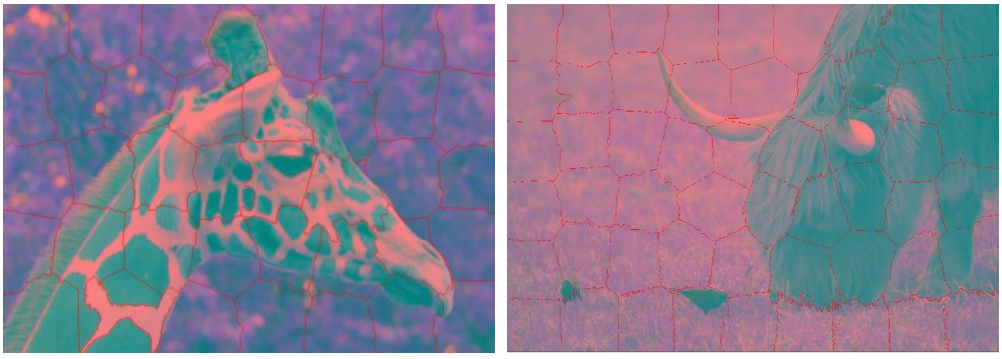


Fig. 4.4 Blocking/segmentation images with adaptive block size

Figure 4.4 suggests that the SLICO segmentation method illustrates a non-convergence and unpredictable block. SLICO integrates an unusual and non-convergence block for reducing execution costs that yields better results than the standard block size. The initial size of the superpixel is complicated to calculate. SLICO chooses the maximum color variance value.

4.2.3 SIFT FEATURE EXTRACTION

Using the SIFT algorithm, the images and their attributes are evaluated. Table 4.2 indicates the input image size, PLF, and S values. Input images are divided into sections by an algorithm of the initial adaptive dimension. The adaptive block size increases the precision of the results of the falsification detection. Good detection results and low processing costs are obtained from the proposed algorithm. Block features from image blocks are created. Most of the technologies used are standard block sizes and applications used in the past, but these features cannot provide location information.

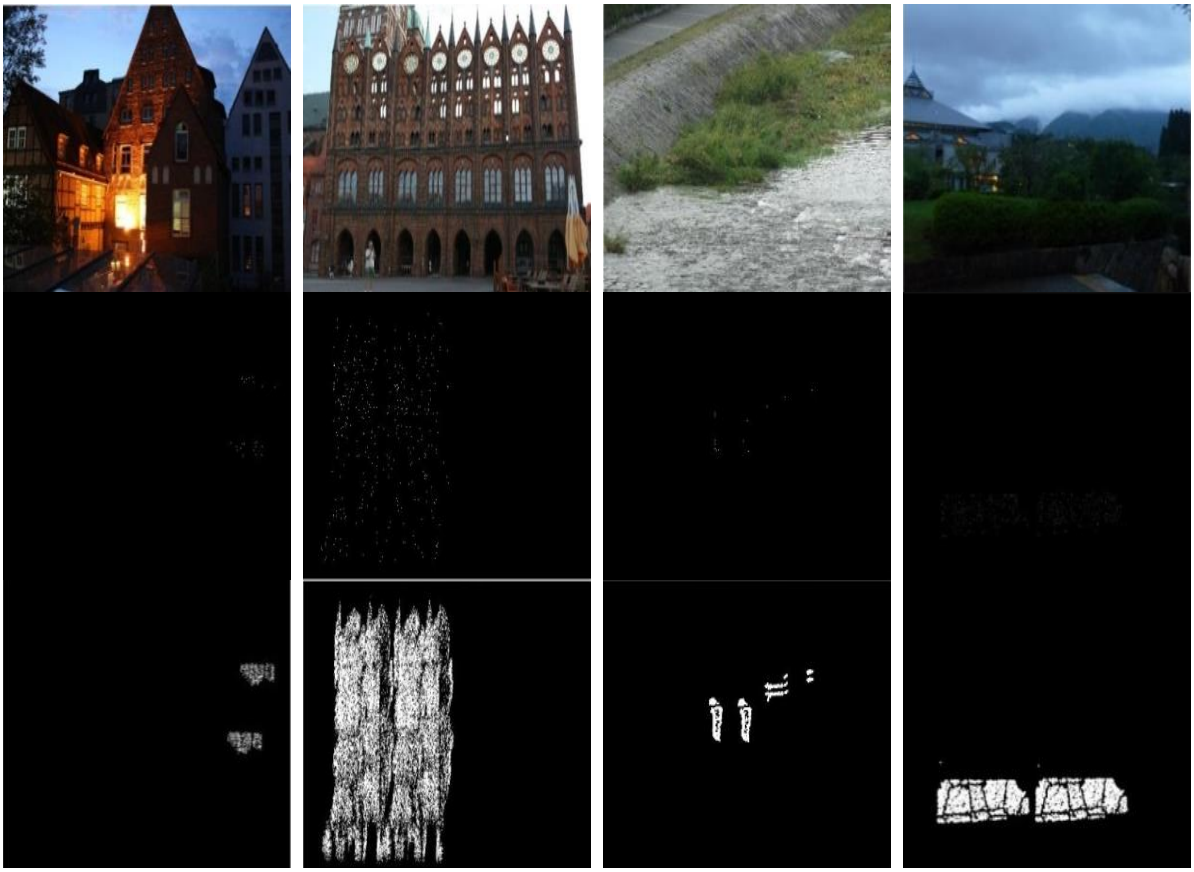


Fig. 4.5 Fabricated pictures: (top row); found forgeries (middle row); increase intensity for forged pixels (bottom row)

In this chapter, we have used a hybrids technique to distinguish feature points from all images sections, and transformation attacks do not impact them. Furthermore, a key point-based SIFT method is used for the determination of feature points in copy-move forgery.

Table 4.2 Superpixel picture estimation of variable image size I1, I2, I3, and I4

Sr. No.	After DWT Image Size(M*N)	*PLF*	*S*
I1	I1=267 × 400 = 106800	PLF1=98.5294	S1=46
I2	I2=1296× 1944=2519424	PLF2 = 85.7143	S2 = 224
I3	I3=1224× 1632= 1997568	PLF3=100	S3 = 199
I4	I4=1224× 1632= 1997568	PLF4=100	S4 = 199

4.2.4 FEATURE MATCHING

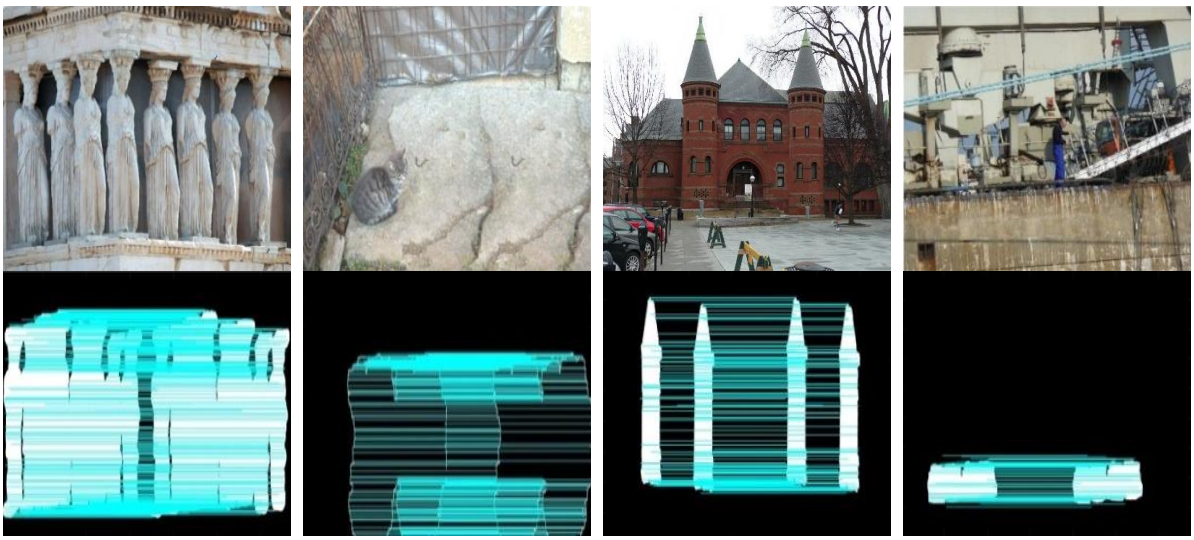


Fig. 4.6 SIFT feature match identification with the ground truth image (bottom row). Forged pictures (top row)

The block features here in this algorithm are compared to other blocks to display the correct matches between every block. Existing methods can detect specific block pairs if other blocks are fitted with the same vectors. If the user sets the threshold and crosses the threshold, the shift vector is defined as a fabricated field. We equate patches with a vector threshold in our algorithm. If the pixel value is less than 0.15, the value is considered, and the pixel size approaches 0.15, the pixel value is negligible. Two maps, A, B, and key points, respectively, are determined as x and y.

Decide the right key point with the help of a threshold value. To highlight the key points, we use a color illumination algorithm. Morphological operation is taken into account to promote the right fabricated block. The automatic forged feature matching is performed using SVM, which eliminates human intervention. SVM is used to train and diagnose the proposed system correctly.

4.2.5 CLASSIFICATION

Detected characteristics give insights into where the fabricated component is found. Superpixels find the picture forgery places and powerfully segment the host image. Using SIFT, we obtain labeled function points loaded with tiny superpixels to accept the forged regions. We detect the right faked parts of the picture by adding the morphological operation. Without human intervention, SVM trains the device and creates the right forgery areas. Fig. 4.7 shows input four forged images after SLICO operation (top row) and corresponding SIFT, Morphological operation (middle row); Ground truth image (bottom row).

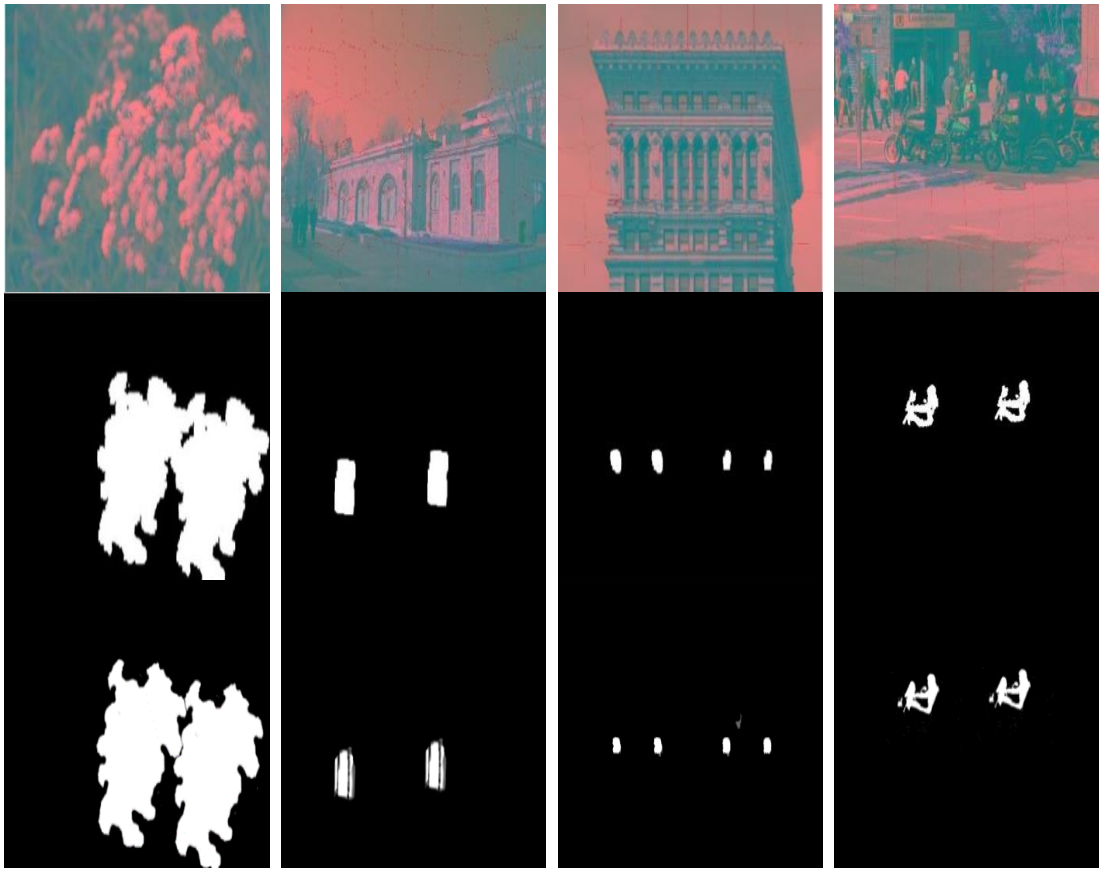


Fig. 4.7 SLICO operation (top row); SIFT, Morphological operation (middle row); Ground truth image (bottom row)

The SVM classifier is designed to provide a computationally fast way for learning. It separates hyperplanes in a high-dimensional feature space across several classes. The SVM classifier is used to identify a set of linearly separable hyperplanes that are linear functions of high-dimensional feature space. The hyperplanes are set up to be as far away from both classes as possible. A kernel function is used to translate the input feature vectors to a higher-dimensional space. For classification, linear SVM was used in this study. Take a look at the image in Fig. 4.8, column 4 and row 1. The dimensions of the matched features in this image are 2128×6. The SVM classifier is fed feature vectors and used to distinguish between two classes: authentic and forged. The precision-recall and F1 results are compared with [2] and published online.

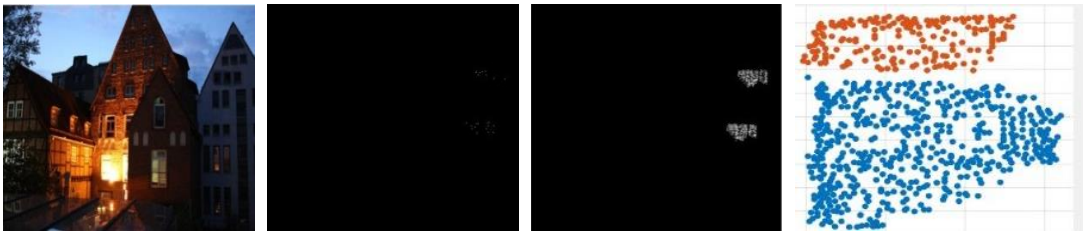


Fig. 4.8 SVM classifies the forged match pixel values with colors (last column).

4.2.6 POST-PROCESSING

We conducted experiments to evaluate the precision, recall, and F1. The dataset contains 48 copy-move-forged PNG high-resolution planes. The results are calculated with Precision-97.20%, recall-100%, and F1-98.3% to identify picture falsification.

Forgery Region Extraction

Input: Take Forgery Detected Image and Ground Truth Image; **Output:** Precision, Recall, and F1.

Load the Forgery Detected Image and Ground Truth Image.

Find the size of the forgery detected image and ground truth image. If equal, then calculate otherwise exit.

for i=1: row; executes for a row; for j=1: column; executes for the column; if the original image region is white, then increment variable k; if original image equal to forge image then increment l; precision=l/k;

for i=1: row; executes for a row; for j=1: column; executes for the column; if the original image region is white, then increment variable p; if original image equal to forge image then increment q; recall=q/p;

f=2* ((precision * recall) / (precision + recall));

4.3 SIMULATION RESULTS

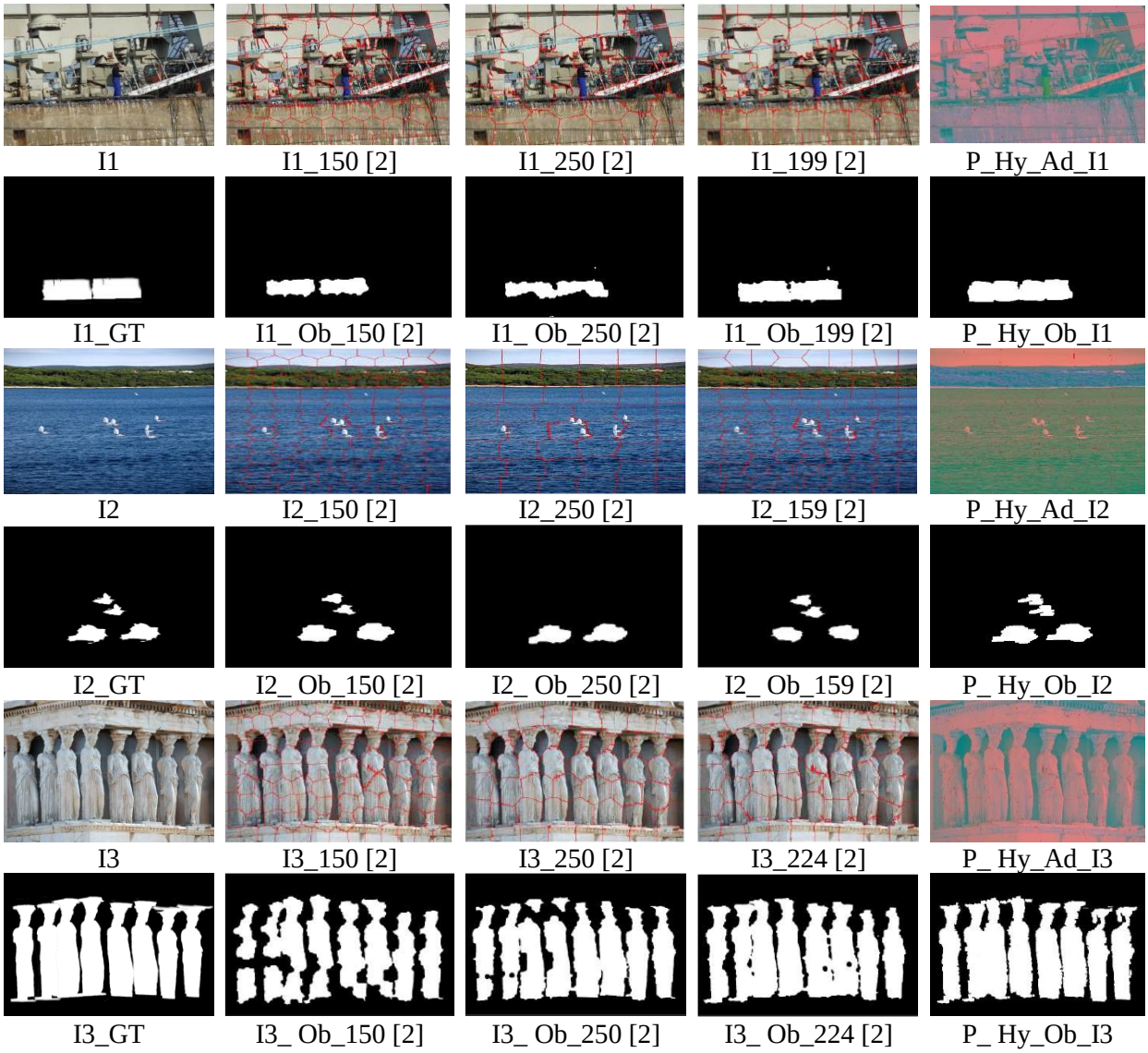


Fig. 4.9 Image I1, I2, and I3 forged images with corresponding ground truth images; Images in columns 2, 3, and 4 illustrate 150, 250, and 159/224 fixed block size with corresponding detected forgery; In fifth column adaptive images and the corresponding forgery detection results.

Many experiments have been carried out in this segment with a hybrid technique to predict improved accuracy, recall, and F1 values. Christian *et al.*, [11], comprising 48 high-resolution PNG images, are tested using the proposed method. Images vary in size from 754 × 1024 to 2592 × 3888. These pictures are a combination of blended, smooth, and textured natural, human-animal living things.

The dataset contains 48 copy-move forgery images. These images are scaled, rotated, compressed with different factors. Then the total images become 1826 for final evaluation. CMF, scale, and rotation have to be identified for various factors. Fig. 4.9 shows I1, I2, I3 relative to Pun *et al.*, [2] with adjustable block size. In contrast to Pun et al., the results of the proposed approach are better. Fig. 4.9 shows the I1, I2, and I3 forged images with corresponding ground truth images. Columns 2, 3, and 4 illustrate the fixed block size 150, 250, and 159/224 with corresponding detected forgery. In the fifth column, an adaptive image block size using adaptive block size and the corresponding forgery detection results. The proposed approach (fifth column) results are compared with Pun *et al.*, [2] (second, third, and fourth column) forgery detection results. As can be seen from Fig. 4.9, better results than Pun *et al.*, [2] are reported using the color illumination, block, and keypoint-based approach.

Table 4.3 Precision and recall performance compared with or without the adjustable scale of a superpixel with existing methods at the image level

Results with or without adaptive size				
I1	S_I1=150 [2]	S_I1=250 [2]	S_I1=199 [2]	P_Ob_I1
Precision	91.44	91.91	93.85	95.52
Recall	69.99	69.74	99.12	99.25
I2	S_I2=150 [2]	S_I2=250 [2]	S_I2=159 [2]	P_Ob_I2
Precision	93.07	93.26	96.6	97.82
Recall	90.75	77.43	78.9	85.12
I3	S_I3=150 [2]	S_I3=250 [2]	S_I3=224 [2]	P_Ob_I3
Precision	96.9	95.59	95.28	96.14
Recall	81.49	89.46	95.19	97.82

Table 4.3 shows comparative results with existing approaches for precision and recall with or without adaptive size at the image level. The adaptive block sizes in the proposed algorithm differ depending on the shape of the image objects. Fig. 4.10 shows a graphical representation of an

image level comparison between precision, recall, and F1 of the proposed scheme and existing methods. Compared to current practices, as shown in Table 4.3 and Fig. 4.10, the proposed results are better.

In this experiment, 48 high-resolution PNG images were examined and evaluated in normal conditions. The pixel-level and image-level forgery detection are shown in Fig. 4.7 (center row) and Fig. 4.11 (middle row). All the pictures are analyzed in these statistics, and the fake picture is discovered. Then we equate the observed falsification picture with the ground truth picture.

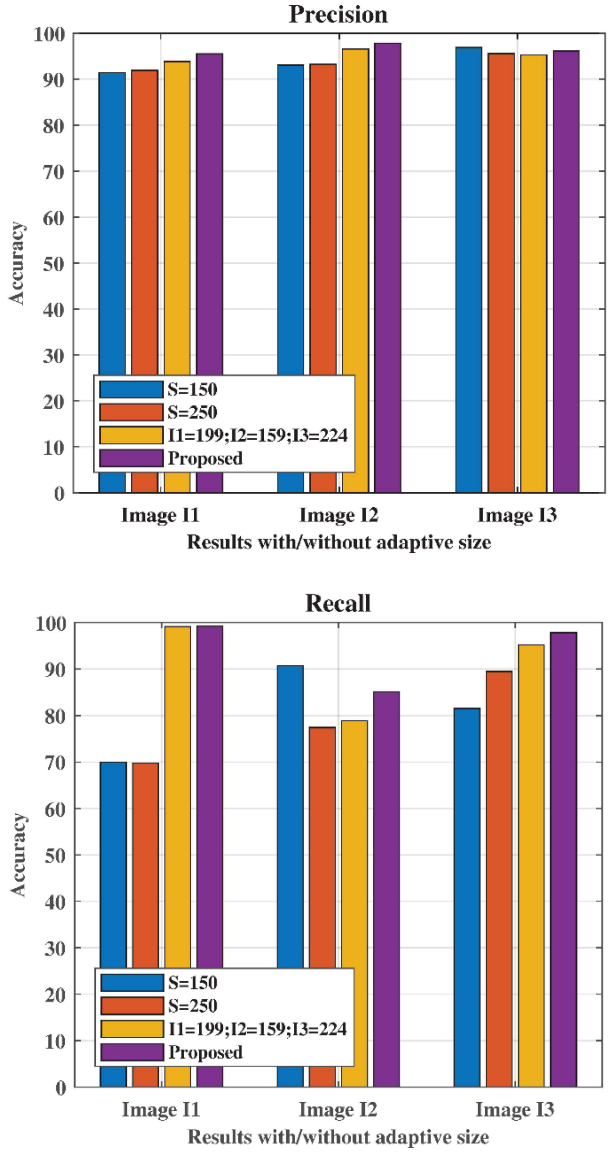


Fig. 4.10 Graphical representation of Precision and Recall analysis of the Proposed Scheme with existing image-level approaches

The identification results are shown in Table 4.4 as Precision=97.25%; Recall=100% and F1=98.53% at image level. In this comparison, other methods achieved precision up to 96%, recall 100%, and F1 value 97.96%. As shown in Table 4.3, Fig. 4.9, 4.10, and Fig. 4.11, our new hybrid approach provides better detection results over existing technology in precision, recall, and F1 value.

Table 4.4 Precision, recall, and F1 image-level analysis of the proposed hybrid method

	Bravo [77]	Wang [78]	SIFT [79]	SURF [80]	Pun [2] Fixed	Pun [2] Adaptive	Proposed
Precision	87.27	92.31	88.37	90.49	95.92	96	97.25
Recall	100	100	79.17	89.58	97.92	100	100
F1	93.2	96	83.52	90.53	96.91	97.96	98.53

When the two duplicated regions are related, and the attached component is smaller than the block size 'S,' it is challenging and entangles to identify falsification. If fabricated pieces are overlapping or intertwined, a single abnormal block is allocated to two blocks. The database Christlein *et al.* [11] was used to evaluate these images.

Fig. 4.11 SLICO operation (top row); SIFT, Morphological operation (middle row); Ground truth image (bottom row)

4.4 SUMMARY

This chapter achieves better results for the copy-move forgery detection using the pre-processing color illumination technique.

- A hybrid technique for distinguishing faked locations, integrating color lighting, blocks, and required points-based methods, is presented in this chapter. According to experimental results, the proposed methodology gives better results under simple CMF as Precision=97.25%, Recall=100%, and F1=98.53% of the original high-resolution images.
- The summary findings of these results demonstrate and conclude that the proposed technique's accuracy is better than other techniques.

CHAPTER 5

CMF AND SPLICING FORGERY DETECTION USING DEEP LEARNING

The analysis of large image databases does not involve traditional methods. The complexity of this algorithm is time. These algorithms have a significant limitation time to find relevant characteristics Gong *et al.*, [81]. This problem is resolved through in-depth learning algorithms. Such algorithms are used for the automatic detection of forgery. Automatically essential image features are found in the deep learning algorithm Chen *et al.*, [82]. It takes some time for the algorithm to prepare for this learning process. We can quickly identify outcomes in the validation procedure after training. On large datasets, the deep learning falsification process takes less time to simulate. We split our algorithm into two phases based on this concept. Based on this definition, we divide our algorithm into two phases. In the first step, the original and spliced image is classified. The second step uses a spliced image dataset to detect and locate the spliced image. Two algorithms have been proposed to identify and monitor splicing images. The DCNN algorithm is used in the first stage to categorize genuinely and split pictures, and the second stage, to identify and locate the splitting imagery. The algorithm Saliency extracts essential image features. Our method uses the salience algorithm to detect variations of illuminated maps. Then, the morphological operation extracts these inconsistencies. The boundary box and the entity borders will be drawn on the spliced area after the fabricated field/element extraction. The performance accuracy of the proposed algorithm will be calculated on the benchmark datasets CASIA v1.0, CASIA v2.0 on the validation set, and the test set. The performance accuracy will also be calculated on the DVMM dataset and BSDS300 dataset. Comparison of the proposed and existing state-of-the-art techniques will result in improved accuracy in the results.

5.1 INTRODUCTION TO CMF AND SF DETECTION

Currently, democracy and openness have been accomplished by their pictures being posted on social networks. Other people have also been able to access personal images in social networks quickly and modify these images. The definition of photo forensics is thus introduced to test the genuineness of these pictures. Many scholars have actively worked in forensic photography, contributing to the value of images in courts of law and media. In many lawsuits, photographs and other digital media have proved their significance. For legal cases, these photographs and

other digital media have to verify original ideas, honesty, and fact, as a testimony against others. Offense events captured by the camera must be tested in a courtroom for their authenticity and genuineness. Therefore the protection of digital media is a special requirement in this regard. Nowadays, software tools easily modified such photos. You don't need to be entirely familiar with the application of software to incorporate tempered regions to adjust the images. Retouching, copying, and splicing are the most common forging carried out in every corner of the globe. Image processing comprises object detection, differentiation, photo falsification detection, and image falsification detection. Passive and active approaches are the main category of falsification monitoring systems. The operational methodology is when input images information will be collected for more analysis of the image. The digital caption and digital signatures could be taken as an example.

The latter technique does not require any previous information named a passive approach for processing the image. However, it takes time to use traditional methods to visualize forensic science like a block or critical point. Therefore, we need to create a less computationally intensive concept and requires minimal human interaction. The automatic development of computers provided hope that these goals could be met. The conventional learning methods for machine learning forensics are, nevertheless, very sophisticated. For example, an algorithm based on machine learning learns to make copy-move and splicing falsifications Zuo *et al.*, [83].

5.2 BUILDING BLOCKS OF MACHINE LEARNING TECHNIQUE

Machine learning identifies a feature-based entity. Where placed data are given for the network training, it is controlled.

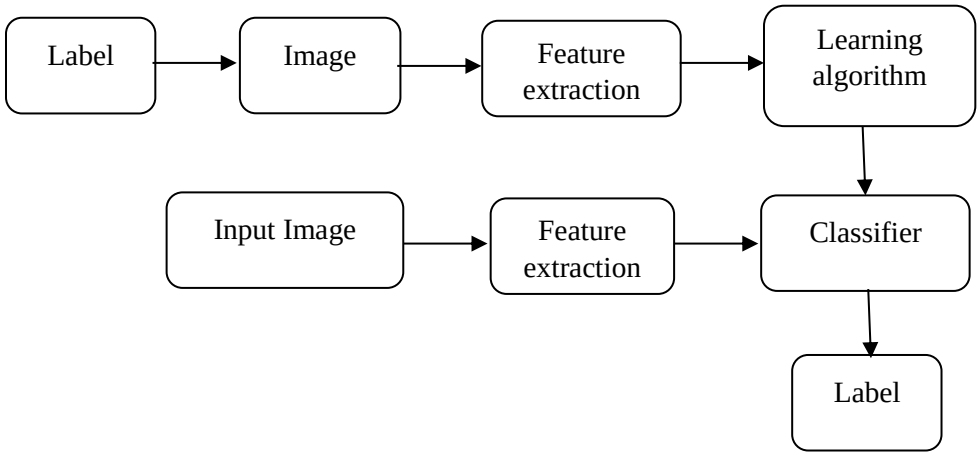


Fig. 5.1 The block diagram of a machine learning-based algorithm

When labeled data are not available for network training, unsupervised learning is delivered. First, the forged characteristics are removed from labeled datasets. Such derived functions are then used for the network training learning algorithm. During the test stage classification, the performance is estimated by comparing the characteristics of the test picture and the trained picture. The labeled data set for all photos is essential for supervised learning. Classifiers such as an SVM, linear component analysis know which view is fabricated or original with labeled images. As can be seen in Fig. 5.1, the input block contains training photos. The features are extracted from these images. Such features are used with the label to learn machine algorithms. The human has to remove parts in this learning algorithm. The same procedure will be conducted during the test phase, and the prediction model will provide the predicted output level.

5.3 BASICS OF DEEP LEARNING TECHNIQUE

The first layer of DCNN contains image pixel values. Normalization is achieved by removing the second image from the input image of the training set. Thirty-two filters are used in the convolution layer. The small image patch uses the dot product by sliding across the pixel. Each layer is subject to a sequence of cognitive operations. If these layers are many, the network will be called DNN, as illustrated in Fig. 5.2.

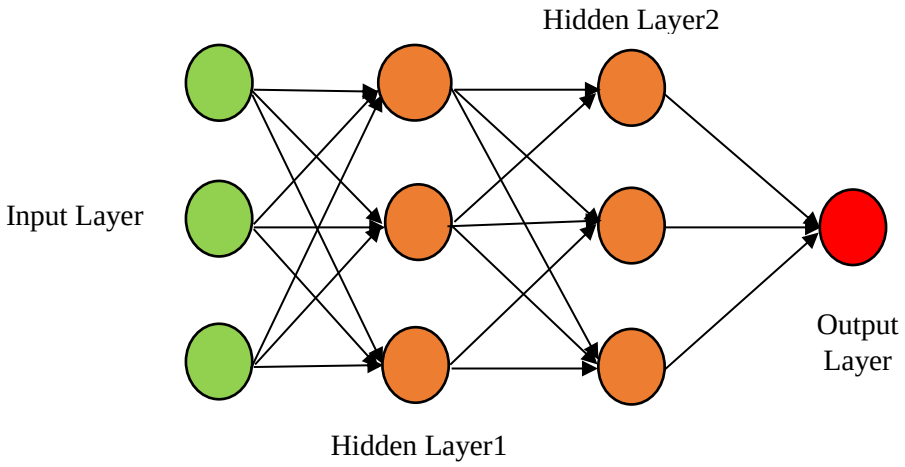


Fig. 5.2 The Deep Convolutional Neural Network (DCNN) operation

5.4 CMF AND SF DETECTION USING DCNN ALGORITHM

This section detects CMF and splicing fake detections using a hybrid color illumination approach based on machine learning methods. A deep neural network is used to identify fake and not

fabricated images. For training and testing of the DCNN model, extensive data collection is used. The training is conducted with guided labeled images from the group. Our following task is to define the mesh layer, size, and a number of the filters. The next step is to define training options like motivation, initial learning pace, the learning rate schedule, regularization (L2), maximum cycles, and mini-batch size. The network is now ready for training. The convolutional neural network with minimal picture batch size is validated and checked. The last step is to detect passive forgeries in public data sets using a color illumination process. It has two essential measures, i.e., image classification and forgery identification. Finally, calculate Precision (P), Recall (R), and F1.

5.4.1 DCNN BLOCK DIAGRAM

The block diagram includes hybrid DL and ML methods for the classification and detection of falsification. Neural networks are used to classify falsifications and color lighting algorithms for localizing forgeries. DCNN learns from the in-depth knowledge of a labeled picture of the forged pictures. DCNNs have recently achieved excellent results in a variety of computer vision (CV) sectors. CNN's require an extensive labeled data set of images or extracted features, where the entire network will learn the little details of the image in the information given. The technique of transfer learning is used to construct a deep convolution network with multiple sets of data. That dataset contains 80% of photos in the train folder and 20% in the test folder. Fig. 5.3 block diagram represents a possible flow chart to identify pictures of hybrid duplication using a color illumination approach based on a machine learning approach.

For different data sets, Transfer Learning is used for training a DCNN model. The model forecasts two categories of pictures, forged and not forged. The CASIAv1.0, CASIAv2.0, DVMM, and BSDS300 dataset is trained on a deep network (XONet) with a base batch size of 20. Both CMF and SF databases are included. CASIA-1 dataset contains images of splicing forgery. This data set includes 800 authentic color images and 921 spliced images with a range of 384×256 pixels.

Additionally, legitimate images are divided into different classes by scene or questions such as creature, character, surface, and plant. The data set Christlein *et al.* [11] is used to diagnose the CMF in the following investigations. The data set consists of 48 uncompressed high-resolution real-world PNG images of size 1500×1500. The photos are from the human, environment, human-made, and mixed groups in the data set. These pictures are subject to various attacks, such as

scaling, rotation, JPEG, and compression. This collection contains a total of 1826 CMF pictures. The deep neural network is the specialized iterations of machine learning that dynamically extracts features and lists the knowledge from which each network layer expands.

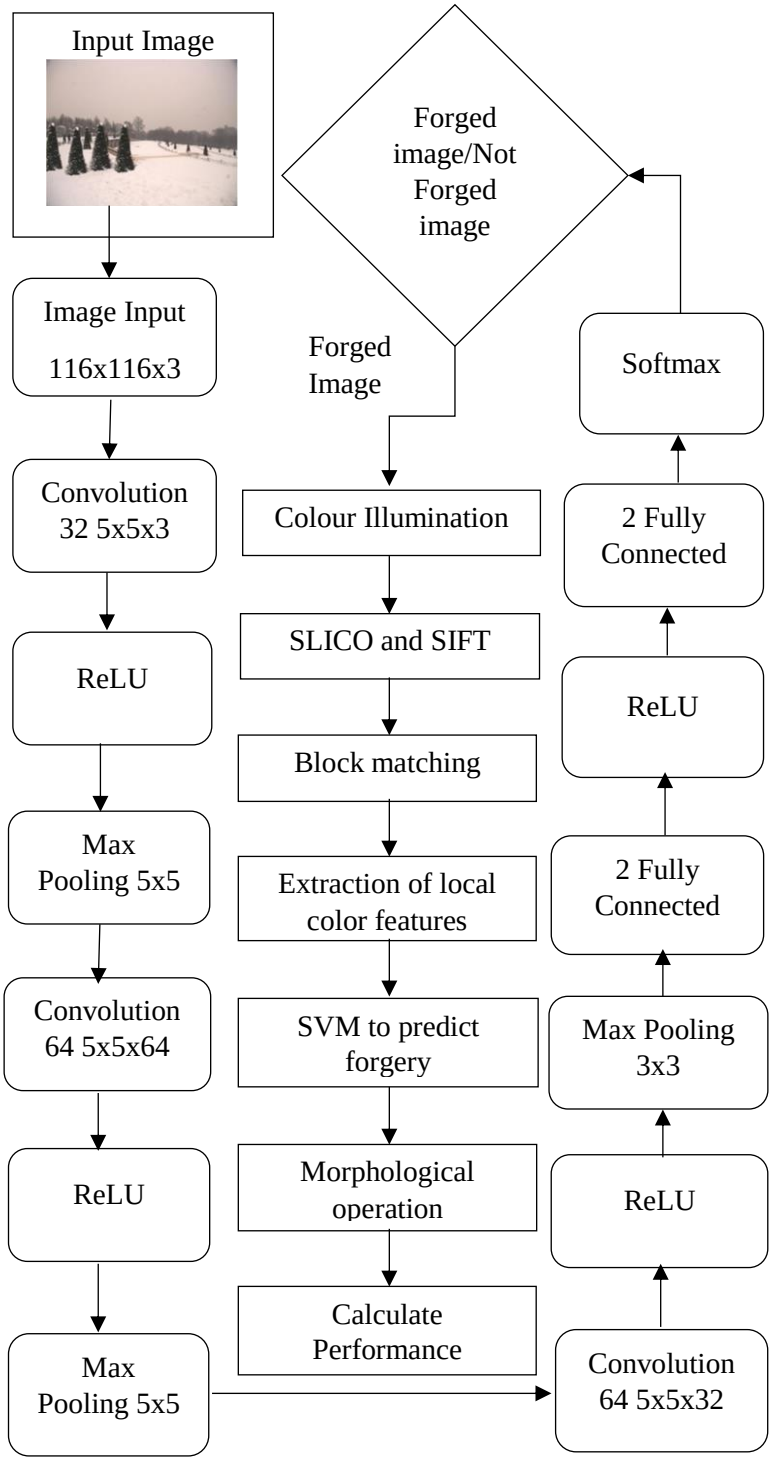


Fig. 5.3 The block diagram of hybrid DL and ML approach for passive image forensic

The deep learning process automatically detects forgery with less technical complexities for extensive Internet data—general schematic of the block displaying the configuration of CNN in Fig. 5.3. The data set of millions of images is used to build the network in-depth properly, but creating a deep CNN needs a heavy and oversized computing memory resource. Otherwise, training will take a long time. Concepts such as overfitting and convergence also hinder network training. The overfitting problem is solved using data augmentation. We transform images by rotations, scaling, changing height, width, and horizontal flip to generate extensive augmentation data. Therefore, we need a large amount of data on each category and high-end GPUs to address this problem. The data is taken in such a manner that it will not bias to a particular class. Deep learning from scratch often becomes time-consuming, so some researchers see an opportunity for other processes using the already trained models. The method to find out forgery demonstrates step by step procedure in Fig. 5.3.

The block diagram consists of two sections. The first portion is used to estimate picture falsification, and the second part is for identifying faked patches using the color illumination-based ML technique. The fourteen layers of the proposed model are the input layer, convolution layer, ReLU layer, Max pooling layer, fully connected layer, and Softmax layer. Weights are then distributed in the original convolution layer with 32 simple high pass filters. The first classification uses 32 filters, although 16 filters are used in the second class. The first convolution layer comprises three color channels, 32 output characteristics scaling, five by five weight matrix scale filters, and 32 fundamental filters must be initialized in 94 weight matrices. The following training choices are taken.

Any input layer is known as the first layer. This layer is used to set the input image size. A large dataset is usually used to train the network with millions of small photos. Therefore, the image of small size can be handled more efficiently. The sample size of the image is 116×116×3. For the initialization of weights, 32 high pass filters were used in the convolution layer. Each layer of the CNN is further linked in the next layer to another neuron.

5.4.2 CONVOLUTION

CNN has an input layer and hidden layers, as well as the output layer. In addition, there are several neurons linked to hidden layers within the input layer.

The gray image input consists of one channel only and has one channel depth. The representation of color input consists of three color channels and has three depths. Let the three color channel

images. M is the column number, and N is the row number. So, in this picture, the total number of pixels is 3×M×N. Each print has three color channels so that we have a 3D matrix. To get a three-dimensional matrix, the input image and the kernel must be the same color channel. If the color image with the 3×w×h kernel is specifically convolved, then we have one output. The deep neural network convolution operation is shown in Fig. 5.4.

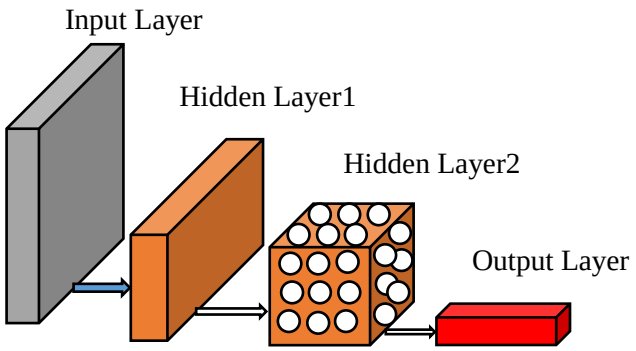


Fig. 5.4 Convolution based deep neural network

Likewise, with the kernel 3×w×h, we get another result when making unique convolutions for a color image. The size of the output channel is 2× (M-w+1) × (N-h+1), where 2 is the color channel; where M is the width of the image, w is the kernel; N is the height of the image, and 'h' is the kernel. The convolution process with the kernel is shown in Fig. 5.5.

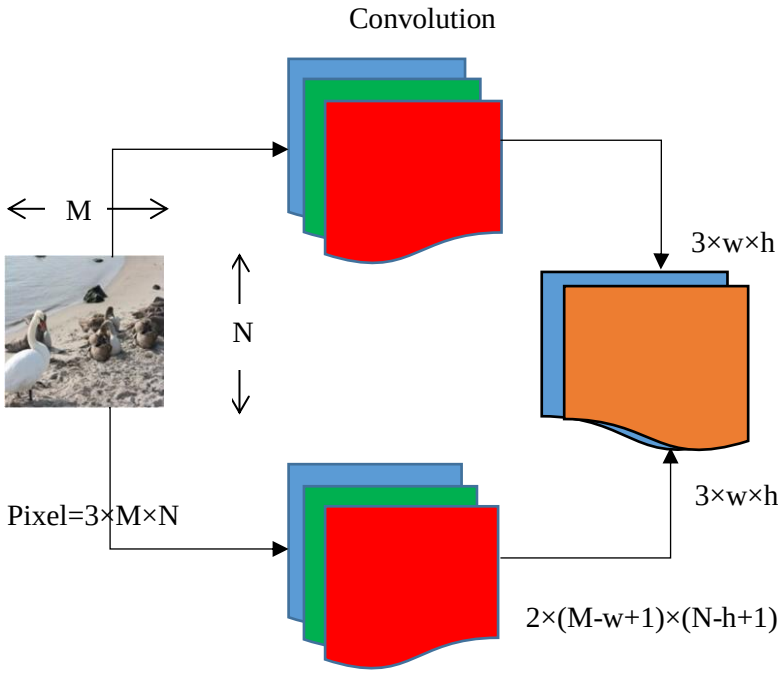


Fig. 5.5 Convolution with the kernel

5.4.3 STRIDE

With the stride operation, the object size is reduced. Fig. 5.6 shows the color image having a pixel value x_k at the location, k is convolved with weight w_k a2t location k of the kernel. The equation gives the output of the function:

$$\text{Output}=b+\sum_k w_k \times x_k \tag{5.1}$$

Where in Eqn. 4 represent 'b' is the bias of the neuron, 'x_k' is the color image pixel value, 'w_k' is the weight of the kernel. This operation gives output for one pixel.

Similarly, we slide the kernel throughout the image, and we achieve all pixel values. The convolution of the 8×8 pixel with 3×3 kernel and stride one gives an output of a 6×6 matrix. The calculation for the output width and height matrix is given by equations 2 and 3, respectively.

$$a_w=\frac{M-w}{S_w}+1 \tag{5.2}$$

where M, w is the width and kernel, respectively, S_w is the stride value and a_w is the output matrix.

$$a_h=\frac{N-h}{S_h}+1 \tag{5.3}$$

where N, h is the height and kernel, respectively, S_h is the stride value and a_h is the output matrix.

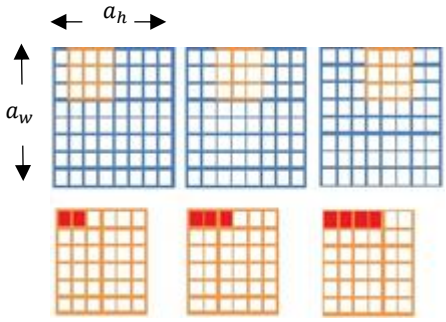


Fig. 5.6 Convolution between color image and kernel with a stride of one

5.4.4 PADDING

The image matrix size shifts during this process so that zero paddings match the image matrix proportions. Let us take an example to perform convolution between the 7×7 image matrix and 3×3 kernel. In this case, zero paddings are used to complete this operation, as shown in Fig. 7.

$$o_w=\frac{M-w+2p_w}{S_w}+1 \tag{5.4}$$

Where M, w is the width and kernel, respectively, S_w is the stride value, $2p_w$ is the padding width and o_w is the output matrix width.

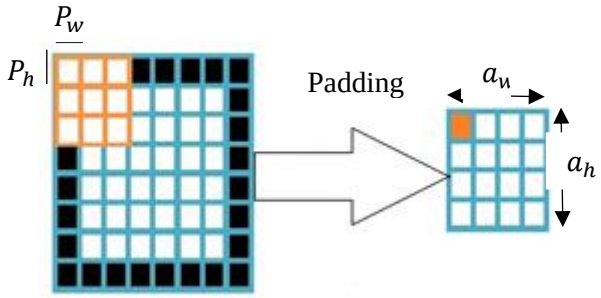


Fig. 5.7 Zero paddings to perform convolution between 7×7 image matrix and 3×3 kernel

$$o_h = \frac{N-h+2p_h}{S_h} + 1 \quad (5.5)$$

Where N, h is the width and kernel, respectively, S_h is the stride value, $2p_h$ is the padding height and o_h is the output matrix height.

5.4.5 POOLING

With a pooling technique, the picture size is reduced. When we pool, we look in a small area and pool it with one single value. The process adds all attributes into one number. The maximal and average pooling activity is indicated in Fig. 5.8.

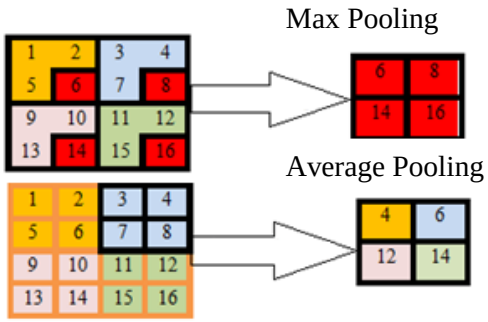


Fig. 5.8 The max pooling and average pooling operation

5.4.6 RELU TRANSFER FUNCTION

Rectified Linear Unit (ReLU) transforms positives in 1 and negative values in 0, respectively. Thus, the negative value derivative is zero, and the positive value derivative is unity. The most important thing is that the file is filled with adequately labeled images. As we learned, to recall a picture is to remember a matrix. This matrix has a range of values from 0 to 255 pixels. The

mapping of the input information to the output label is carried out in supervised learning.

We also execute an input-image and kernel convolution. The object type defines the size of the kernel in the image. The image-derived features are included in the matrix used as an interface to the next layer.

The next step involves pooling. The process of convolution will form a significant matrix. This matrix size is challenging for further analysis. The pooling method is used to reduce the matrix's size and thus reduce the complexity of the model's implementation. The next stage is the translation of likelihood. The large numerical values of the matrix images are used to render the probability of an image element appearing in the photo. Finally, the softmax function is used to assign a probability.

5.4.7 IMAGE FORGERY CLASSIFICATION

DCNN is used to replicate the action and SF labeling, classifying the picture fiction as fabricated and not forged. We get a score on the validation sets once the classification process has finished. The classification algorithm provides the results as fake and not fabricated photos and correctly classified, as shown in Fig. 5.9.

Fig. 5.9 Classification of not spliced and spliced image

Table 5.1 DCNN training on CASIA v1.0 using transfer learning

Epoch	Mini-batch Loss	Mini-batch Accuracy	Time Elapsed (seconds)	Iteration
1	0.6931	50.00%	2.27	1
50	0.4314	85.00%	291.18	550
100	0.1377	100.00%	582.36	1100
150	0.1184	100.00%	878.15	1650
200	0.1166	100.00%	1172.63	2200

Table 5.2 DCNN training on CASIA v2.0 using transfer learning

Epoch	Mini-batch Loss	Mini-batch Accuracy	Time Elapsed (seconds)	Iteration
1	0.6932	25.00%	1.86	1
50	0.1409	90.00%	11612.89	22200
100	0.0050	100.00%	23513.76	44850
150	0.0101	100.00%	35397.85	67500
200	0.0027	100.00%	47384.30	90150

Table 5.1 and Table 5.2 shows the training of DCNN using the transfer learning approach for multiple data sets. The performance accuracy is calculated on the CASIA v1.0 validation set, and the test set is 0.9807 and 0.9878, respectively. Performance accuracy is calculated on the CASIA v2.0 validation set, and the test set is 0.9767 and 0.9805, respectively. In the image forgery localization stage our algorithm gives performance, accuracy using equation (1-3) in term of Precision=97%; Recall=100% and F1=99% for CoMoFoD dataset.

5.4.8 PROPOSED ALGORITHM RESULTS

In this chapter number of experiments are performed on different data sets. These experiments are performed on the GPU-enabled machine. Image forgery classification and localization using the proposed approach on other datasets Tralic *et al.*, [84], Dong *et al.*, [85] are carried out in MATLAB software. Tralic *et al.*, [84], 48 images of plain CMF are available along with ground truth images. These images are evaluated, tested, and compared with different parameters to the existing technique. In the dataset Dong *et al.*, [85], spliced images are present. CASIA 1.0 and 2.0 versions have Authentic and Forged image folders. In the CASIA 1.0 version's authentic folder, 800 images are present, and in CASIA 2.0 version authentic folder, 7491 images are available. In the CASIA 1.0 version's spliced folder, 800 images are present, and in CASIA 2.0 version, 7491 images are available. CMF and splicing are two types of forgeries that are available in the dataset BSDS. In this data set, we have taken 100 images for test and 200 images to train the network

The result is shown in Fig. 5.10, row1, for the CASIA v.1.0 database. In the CASIA v1. 0 authentic folders contain 800 images, and in the spliced folder, 921 images are present. These images are divided into two folders, such as train and test.

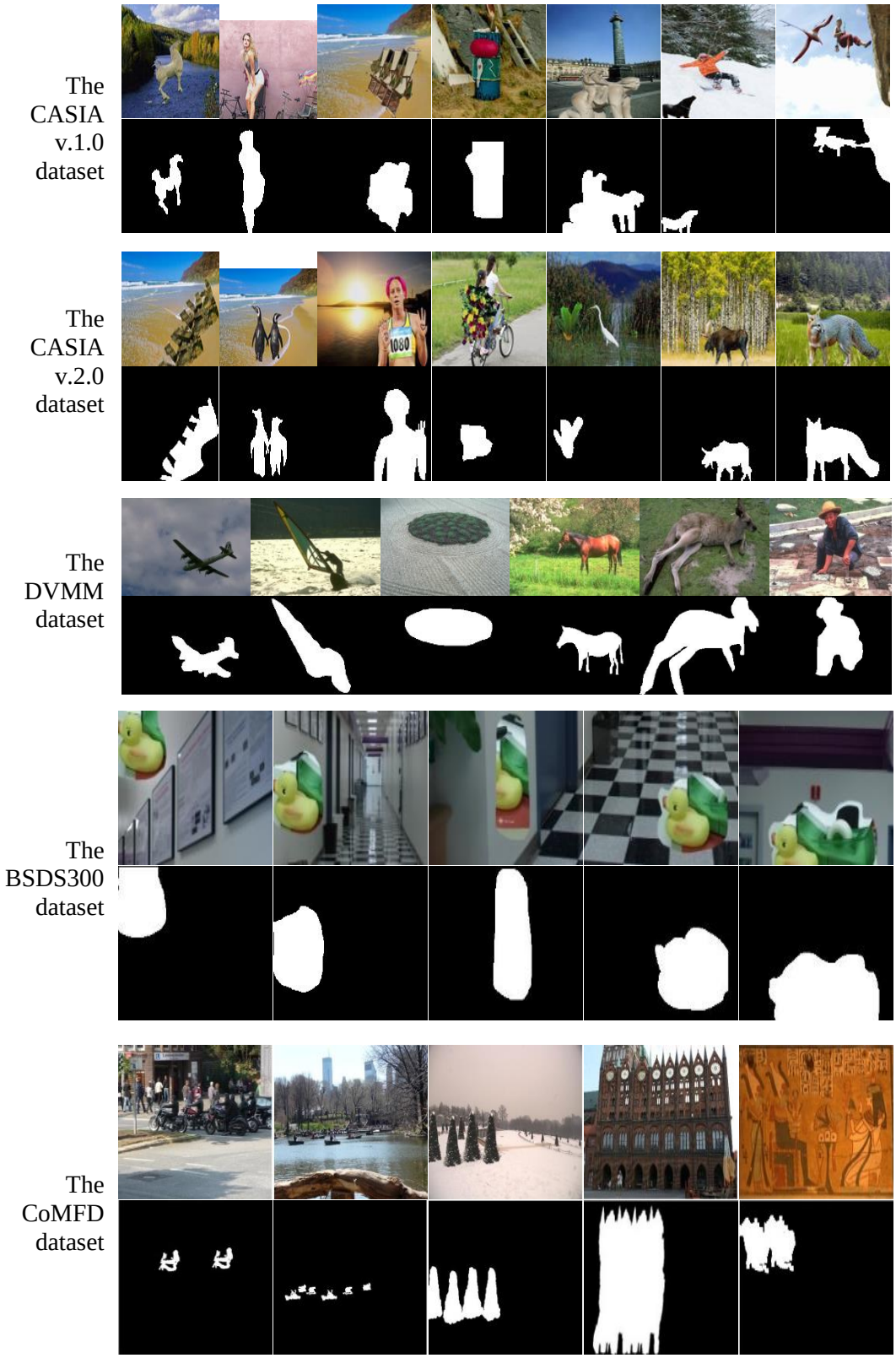


Fig. 5.10 The CASIA-1 (row1), CASIA-2 (row2), DVMM (row3), BSDS300 (row4), CoMFD (row5) dataset of CMF and SF detection results

Training images are 80%, and test images are 20%. All SF images are shown in the 1st row (top). The image region analyzer application creates forgery detected output shown in the 1st row (bottom). The performance accuracy is calculated on the CASIA v1.0 validation set, and the test set is 98% and 99%, respectively. The result is shown in Fig. 5.10, row2, for the CASIA v.2.0 database. The CASIA v2.0 authentic folder contains 7491 images in the dataset, and in the spliced folder, 5123 images are available. All SF images are shown in the 2nd row (top). The image region analyzer application creates forgery detected output shown in the 2nd row (bottom). The performance accuracy is calculated on the CASIA v2.0 validation set, and the test set is 97% and 98%, respectively.

The result is shown in Fig. 10, row3, for the DVMM database. Splicing and copy-move forgeries are two types of forgeries that are available in the DVMM dataset. The dataset contains 100 images for testing and 200 images for training the network. In the third row (top), all spliced images are shown. In the third row (bottom) image region analyzer application creates forgery detected output. The DVMM dataset forgery detection accuracy is 97%.

The result is shown in Fig. 5.10, row4, for the BSDS300 database. In this dataset, 200 images are available for training, and 100 images are available for testing. In the first row, all spliced images are shown. In the fourth row (top), machine learning-based color illumination forgery detected regions are shown. Finally, in the fourth row (bottom) image region analyzer application creates forgery detected output. The BSDS300 dataset forgery detection accuracy is 98%.

Table 5.3 Comparison between CMF and SF dataset accuracy

Authors	CASIA1 Acc. (%)	CASIA2 Acc. (%)	DVMM Acc. (%)	BSDS Acc. (%)	CMF Acc. (%)
Proposed	99	98	97	98	99
Zh. [86]	-	-	94	-	-
He [87]	-	90	94	-	-
Mu. [22]	95	97	-	-	-
Y. R. [33]	98	98	96	-	-

The result is shown in Fig. 5.10, row5 for CoMFD dataset. This dataset contains 48 images of plain copy-move forged images. This dataset contains natural, architects, animals, art, plant, and text images. In the fifth row (top), all copy-move forged images are shown. In the fifth row (bottom), morphological operation shows forgery detected output. The proposed algorithm is

tested on image-level using the achieved performance accuracy values, i.e. Precision (P) = 98%, Recall (R) = 100% and F1 = 99%.

Table 5.3 shows the accuracy result comparison on CASIA 1, CASIA 2, DVMM, BSDS300, and CMFD datasets. From the comparison, we can say that our proposed method performs better in all datasets. Fig. 5.11 shows the bar graph between the proposed and other methods. From fig. 5, .11, the blue color represents the proposed method, orange color represents Yuan Rao, Yellow color represents Muhammad, Violet color represents 'He,' and the green represents Zhao. Thus, the forgery detection accuracy is better as compared to all other states of the methods.

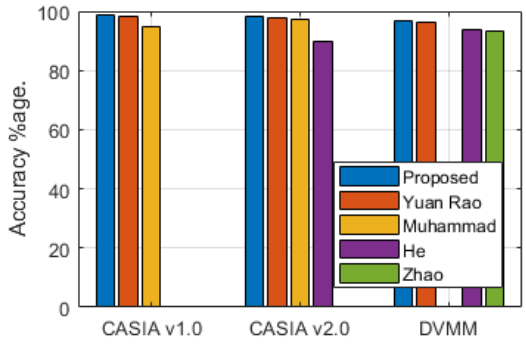


Fig. 5.11 Bar graph for CMF and SF accuracy comparison on different datasets.

5.5 CMF AND SF DETECTION USING DCNN AND SEMANTIC SEGMENTATION

The proposed algorithm uses DCNN and Semantic Segmentation to classify and localize forgery. This model contains three versions of DCNN with different hidden layers. The datasets are taken from the CMF image manipulation dataset as follows: Image Processing Research Groups dataset (GRIP), Columbia uncompressed image splicing dataset (DVMM), and Berkeley segmentation dataset (BSDS300). All these datasets consist of original, forged, and labels of each image. In the first step, authentic and forged images are applied to 27 layers DCNN network for training. The trained model is retrained in the second step using 54 DCNN transfer learning network layers with the copy-move, spliced, and fake video images. The trained model classifies three fake categories as a copy-move, splicing, and forged video frame. In the third step, all images and labels are applied to 91 layers VGG network for training. The trained model classifies forged and not fake pixels of all fake images. The forged segments are correctly detected, as shown in Fig. 5.12.

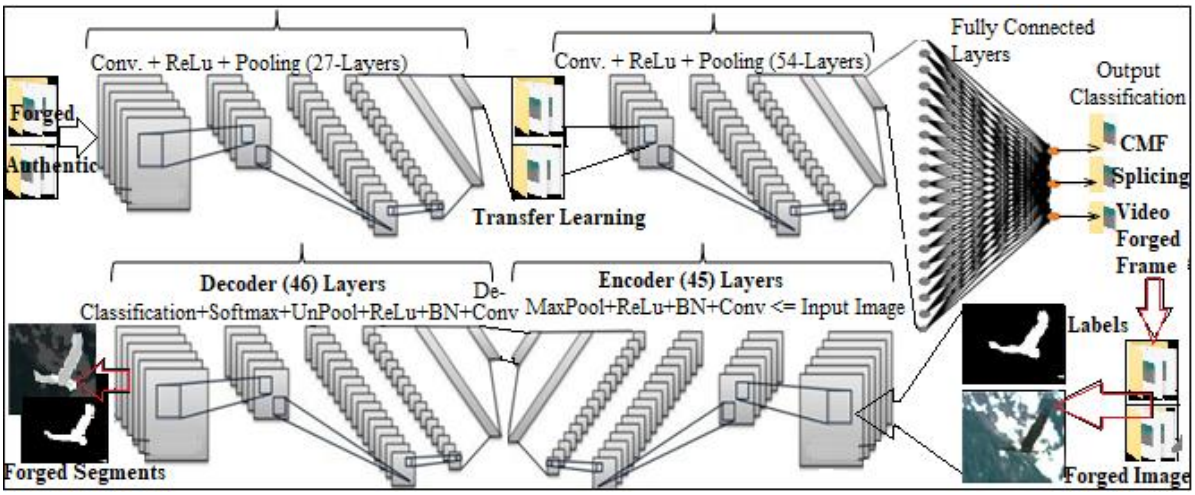


Fig. 5.12 The block diagram of the proposed DCNN model

In the first step, forged and authentic images are applied to the model for training. The model consists of an input layer, hidden layers, and classification layers. The twenty-seven layers DCNN model is shown in Table.5.4.

Table 5.4 Twenty-seven layers of the DCNN model

Layer No.	Layer Name	Layer Description
1	Image Layer	32 5x5 convolutions with stride [1 1] and padding [4 4 4 4]
2, 5, 8, 11, 13, 16	Convolution	32 5x5 convolutions with stride [1 1] and padding [4 4 4 4]
3, 6, 9, 12, 14, 17	ReLU	ReLU
4, 7, 10, 15, 18	Max Pooling	5x5 max pooling with stride [2 2] and padding [0 0 0 0]
19, 22	Convolution	64 5x5 convolutions with stride [1 1] and padding [4 4 4 4]
20, 23	ReLU	ReLU
21, 24	Max Pooling	5x5 max pooling with stride [2 2] and padding [0 0 0 0]
25	Fully Connected	Two fully connected layer
26	Softmax	Softmax
27	Classification	Cross-entropy

In the first layer, the input is applied with thirty-two 5×5 filters. These filters use one stride and four paddings in convolution operation. The layers in between the second and twenty-four are known as hidden layers. These hidden layers consist of multiple convolutions, ReLu, and Max Pooling layers. In this model, 5×5, max pooling with two strides, and four paddings are used. The model's 25th, 26th, and 27th layers are fully connected, softmax classification and output layer.

The trained model is saved for retraining a copy-move, splicing, and forged video frames using transfer learning.

Table 5.5 Training accuracy of proposed 27 layers DCNN model

Batch	Iteration	Time Elapsed	Mini-batch Accuracy	Mini-batch Loss
1	1	0:00:00	47.67%	8.3413
50	34550	2:02:57	88.00%	0.2715
100	69100	4:04:51	98.67%	0.1188
150	103650	6:06:51	98.67%	0.0444
200	138200	8:09:04	98.33%	0.0487
250	172750	10:15:55	98.33%	0.0473
300	207300	12:24:33	98.33%	0.047

Table 5.5 shows the training accuracy using the proposed 27 layers DCNN model. In this training process, 300 batches of images are trained for twelve hours, twenty-four minutes, and thirty-three seconds with 207300 iterations. As a result, the mini-batch accuracy improves from 47.67% to 98.33%, and the mini loss reduces from 8.3413 to 0.047.

Fig. 5.13 shows the graph between training accuracy and iteration of the 27 layers DCNN model. The x-axis represents iteration from $0 \text{ to } 2\times10^5$ whereas the y-axis represents accuracy from 0 to 100%. Thus, performance on the training set is 97.97%, and performance on the validation set is 97.91%.

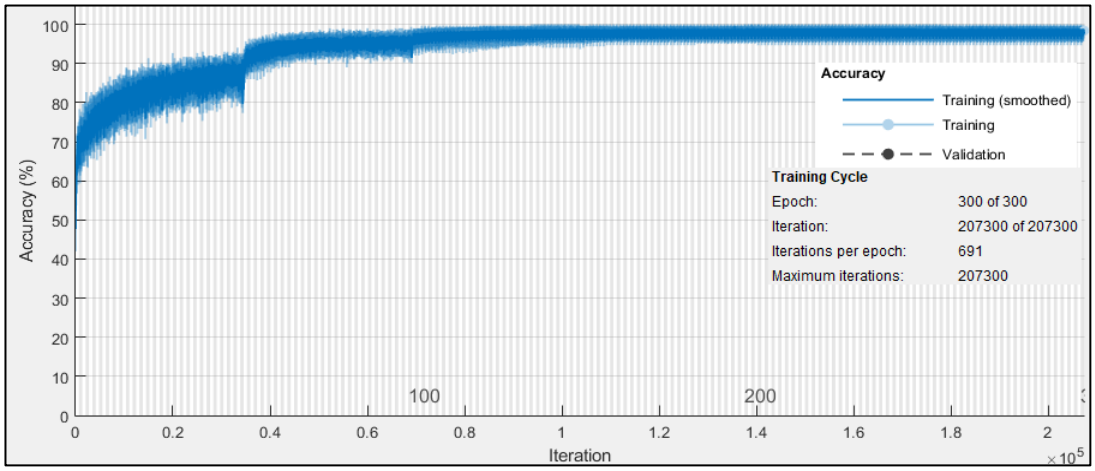


Fig. 5.13 The graph between training accuracy and iteration of a 27 layer DCNN model

Fig. 5.14 shows the graph between loss and iteration of the 27 layers DCNN model. The x-axis represents iteration from $0 \text{ to } 2\times10^5$ whereas the y-axis represents a loss that varies from 0 to 8. In

training, multi GPUs are used with a piecewise learning rate schedule $1e^{-8}$. The mini loss reduces from 8.3413 to 0.047.

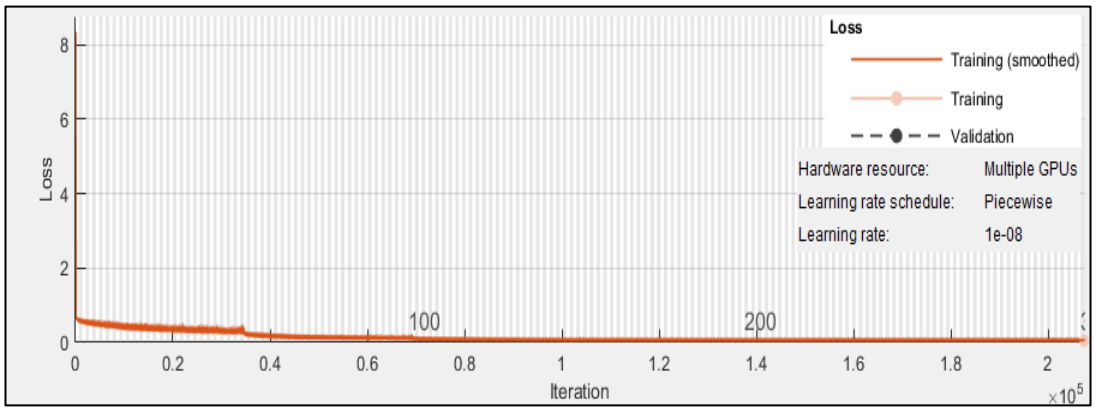


Fig. 5.14 The graph between loss and iteration of 27 layer DCNN model

In the second step, the trained model is retrained using the DCNN model. The authentic and forged images of copy-move, splicing, and forged video frames classes are retrained using 54 layers DCNN transfer learning model.

Table 5.6 The 54 layers DCNN model for training of a copy-move, splicing, and forged video frames

Layer No.	Layer Name	Layer Descriptions
1	Image Layer	images with zero center normalization
2, 8, 11, 17, 20, 26, 29, 41	Convolution	32 3x3 convolutions with stride [1 1] and
3, 9, 12, 18, 21, 27, 30, 42	ReLU	padding [4 4 4 4]; ReLU; 5x5 max
4, 10, 13, 19, 22, 28, 31, 43	Max Pooling	pooling with stride [2 2] and padding [0 0 0 0]
5, 14, 23, 32, 35, 38, 44, 47	Convolution	64 3x3 convolutions with stride [1 1] and
6, 15, 24, 33, 36, 39, 45, 48	ReLU	padding [4 4 4 4]; ReLU; 5x5 max
7, 16, 25, 34, 37, 40, 46, 49	Max Pooling	pooling with stride [2 2] and padding [0 0 0 0]
50, 52	Fully Connected	Fully-connected layer
51	ReLU	ReLU
53	Softmax	Softmax
54	Classification	Cross-entropy

Table.5.6 shows the 54 layers DCNN model. In the first layer, the input is applied with zero center normalization. The layers in between the second and forty-nine are known as hidden layers. These

hidden layers consist of multiple convolutions, ReLu, and max-pooling layers. The convolution layer uses thirty-two filters of 3×3 sizes with one stride and four paddings. Max pooling uses a 5×5 window size with two strides and zeroes paddings. Two fully connected layers are used. The 50 and 52 layers of the model are fully connected. The 53 and 54 layers are softmax and classification layers, respectively. These layers are called output layers, which classify copy-move, splicing, and forged video frames. The trained model is tested on the training and validation dataset. The performance on the training set is 100%, in which one out of 23462 wrong training classifications. The performance on the validation set is 99.99%.

Table 5.7 shows the training accuracy of the proposed 54 layers DCNN model. In this training process, 50 batches of images are trained for twenty-six hours, fifty-eight minutes with 14650 iterations. The mini-batch accuracy improved from 53% to 100%, and mini-batch loss reduced from 2.9193 to 0.0013.

Table 5.7 Training accuracy of proposed 54 layers DCNN deep learning model

Iteration	Time Elapsed	Mini-batch Accuracy	Mini-batch Loss
1	0:01	53.00%	2.9193
50	0:07	57.00%	1.0069
100	0:14	58.00%	0.8319
200	0:25	71.00%	0.6139
400	0:47	71.00%	0.6129
800	1:33	98.00%	0.1159
1600	3:05	98.00%	0.0666
3200	6:04	98.00%	0.0744
6400	12:01	100.00%	0.0116
12800	23:35	100.00%	0.0021
14650	26:58	100.00%	0.0013

Fig. 5.15 shows the graph between training accuracy and iteration of the 54 layers DCNN model. The x-axis represents iteration from $0 \text{ to } 14650$ whereas a y-axis represents accuracy from 0 to 100%. Thus, the performance on the training set is 100%, and performance on the validation set is 99.99%.

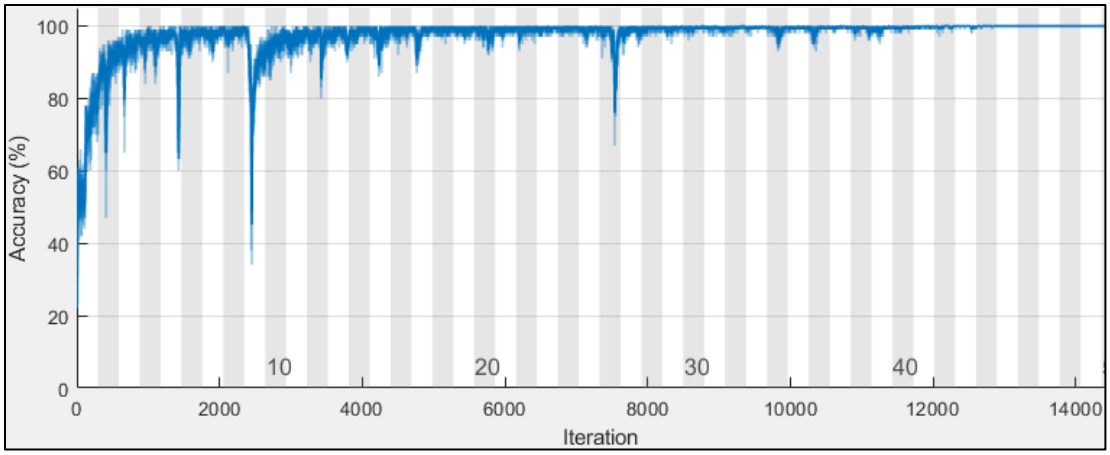


Fig. 5.15 The graph between training accuracy and iteration of 54 layer DCNN model

Fig. 5.16 shows the graph between loss and iteration of the 54 layers DCNN model. The x-axis represents iteration from $0 \text{to} 14650$ whereas a y-axis represents a loss that varies from 0 to 4. In training, multi GPUs are used with a piecewise learning rate schedule $1e^{-8}$. The mini loss was reduced from 2.9193 to 0.0013.

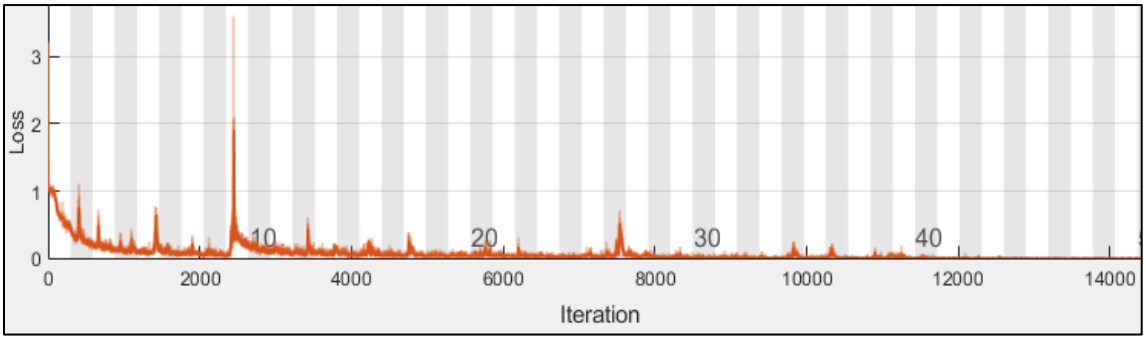
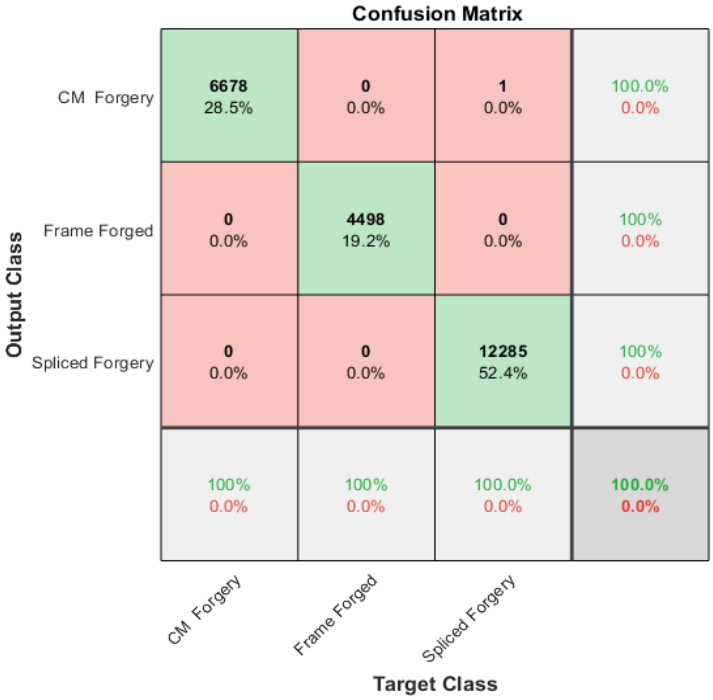


Fig. 5.16 The graph between loss and iteration of 54 layer DCNN model

Fig. 5.17 shows the confusion matrix between CMF, video frame forged, and spliced forgery for training and validation of the DCNN model. Fig. 5.17 (a) represents the training confusion matrix, whereas Fig. 5.17 (b) shows a validation confusion matrix. In the training and validation, the overall accuracy for classification is 100%. All the images between the three classes are correctly classified.



(a)

(b)

Fig. 5.17 Confusion matrix between CMF, video frame forged, and spliced forgery for training and validation of DCNN model

Fig.5.18 shows the three classification CMF, video frame forgery, and splicing forgery using the DCNN model. These three classes are classified into their categories in testing with percentages. The CM Forgery dataset contains 8348 images, the video frame forgery dataset contains 5622 images, and the Spliced forgery dataset contains 15358 images. We take 6678 CM forgery, 4498 video frame forgery, and 12286 splicing forgery images in training. In the testing phase, 1670 CM forge images, 1124 video frame forgery images, and 3072 splicing forgery images are used. In training, only one spliced image is wrongly classified as CM forgery. In the testing, all images are correctly classified with 100% accuracy in all three categories.

The third step is to load the image dataset, then divide it into training and testing categories. These training and test images are resized into equal size as the VGG network. All photos are categories into two-pixel colors using labels of the corresponding image. Black color represents authentic and white color represent forged objects. The model is trained using a stochastic gradient descent backpropagation algorithm. To prepare the model, we define the piecewise learning rate with a learning rate drop factor of 0.5. The learn rate drop period is 5, Mini Batch Size is 11, and Max epochs are 20. To plots the training progress, we use a multi GPU execution environment with Verbose set to true. Verbose is used to display the training process runtime.

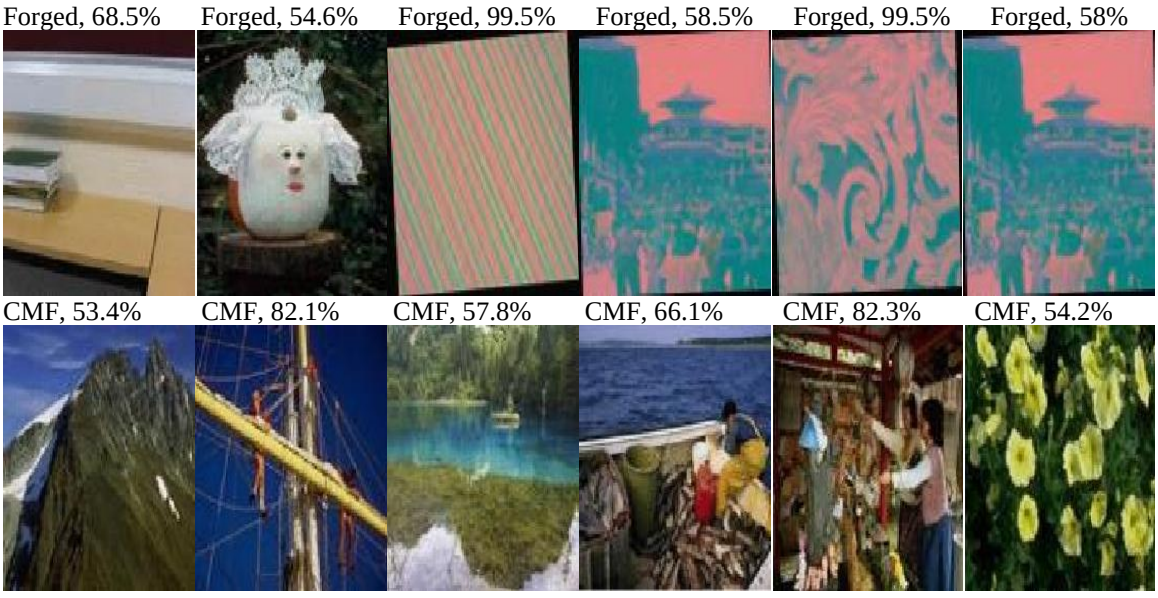


Fig. 5.18 Classification of three classes CMF, video frame forgery, and SF using DCNN model

Then we perform data augmentation and apply preprocessing before training the VGG16 network. Next, train the model with seventy percent of training data using given options. The model predicts color pixels of the test image from thirty percent of test data. The prediction results

show white color pixels for a forged category, black for the Not Forged group, and display the class names with the color. Finally, we calculate the overall performance of the model.

5.5.1 CHOOSING THE DATASET

To detect CMF and SF, we have used publically available datasets, i.e., 1. GRIP, 2. DVMM, 3. CMFD, 4. BSDS300.

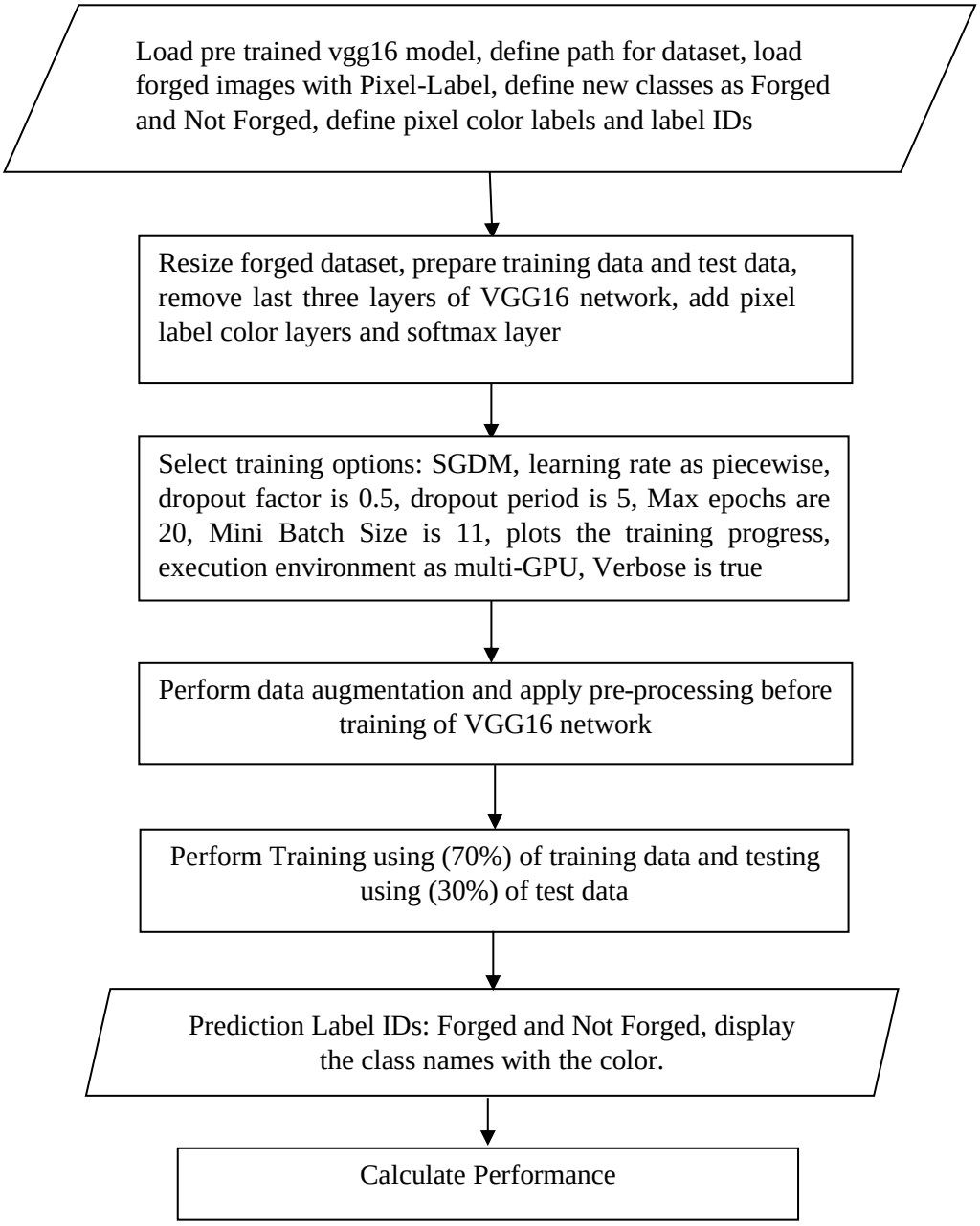


Fig. 5.19 flowchart of image forgery detection and localization using the proposed deep learning method

The color illumination, rotation at 2°, 4°, 6°, 8°, and 10° is applied to each image. In the CMFD image manipulation, dataset 48 images are available along with 48 ground truth images and color illumination, rotation at 2°, 4°, 6°, 8°, 10° is applied to each image. We combined all image datasets and applied geometric and color illumination attacks. The total images are 47566. The training images are 28540, and the testing images are 19026. The image frequency for forged images is 0.0805, and for not forged images is 0.9102. Fig. 5.19 shows the flowchart of image forgery detection and localization using the proposed deep learning method.

5.5.2 PROPOSED HYBRID TECHNIQUE BLOCK DIAGRAM

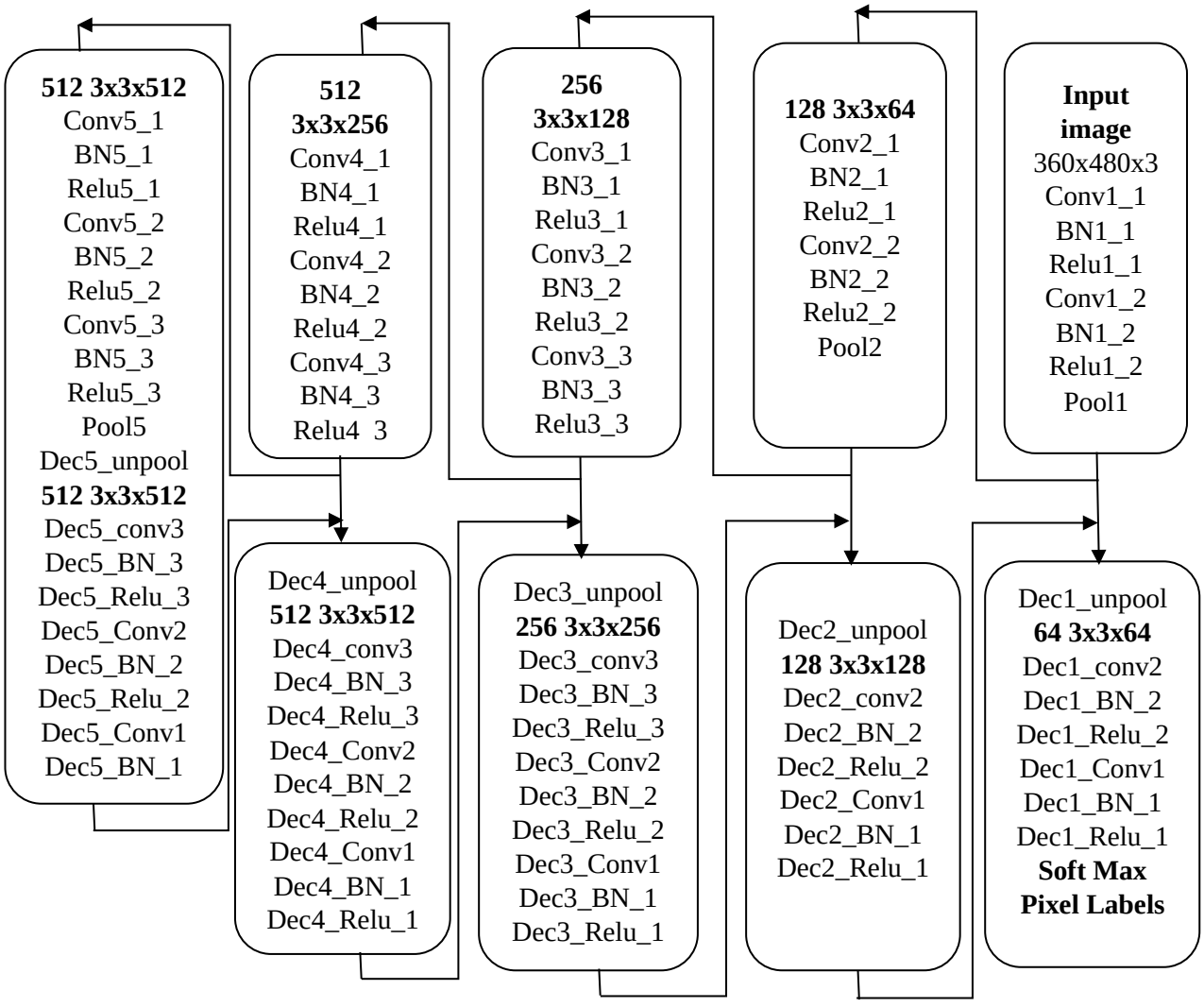


Fig. 5.20 The block diagram of the proposed 91-layer deep convolution neural network (DCNN)

The proposed forgery detection method is shown in Fig. 5.19. This algorithm comprises color illumination, a deep convolution neural network, and semantic segmentation. Color illumination is used to apply a color map in the preprocessing step. This step is used to increase data size and generate better features. The transfer learning approach is used to train VGG-16 with two classes using a deep convolution neural network. This algorithm classifies forged and not forged image pixels. These classified images with color pixel labels are prepared using semantic segmentation to localize forged pixels. The transfer learning approach is used to train data sets. 70% of images are used for training from the dataset, and 30% are used for testing. DCNN is used to train the model, and transfer learning is used to train the new classes. This model predicts two-pixel color levels as Not Forged and Forged.

The dataset with millions of images is used to train the network using deep CNN with computationally heavy GPUs and extensive memory resources. Otherwise, preparing an accurate model would be time-consuming. Also, concepts like overfitting and convergence often complicate the preparation of a precise model. Hence, we need high-end GPUs to deal with this trouble. Thus, deep learning from scratch sometimes proves to be a time-consuming task, so some researchers see an option to use the already trained models for further processing. Pre-processing is applied on the dataset using color illumination, scaling, rotation, and image resized into 360×480×3. With the help of preprocessing, the total images are 47566 in the dataset. From the dataset, the training images are 28540, and the testing images are 19026. The main reason to apply color illumination is that the edges and corners are detected quickly, which increases forgery detection accuracy.

Suppose, RGB colors of a pixel centered at (x, y) to be

$$f(x,y) = \{f_R(x,y), f_G(x,y), f_B(x,y)\} \tag{5.6}$$

where $f(x,y)$ is image value, (x,y) is a spatial coordinate in the image, f_R is red color value, f_G is green color value, and f_B is a blue color value

$$f(x,y) = \int_{\omega} \mathrm{e}(\lambda,x,y)\,\mathrm{s}(\lambda,x,y)\,\mathrm{c}(\lambda)\,\mathrm{d}\lambda \tag{5.7}$$

where $\mathrm{e}(\lambda,x,y)$ is Light source, $\mathrm{s}(\lambda,x,y)$ is a surface reflection, $\mathrm{c}(\lambda)$ is camera sensitivity

Chromatic reflection of the scene, when no reflection z=0

$$\iint s(\lambda,x,y)d_x d_y = 0 \tag{5.8}$$

Chromatic reflection of the scene, when no reflection z=1

$$\iint s(\lambda, x, y) d_x d_y = \iint d_x d_y \tag{5.9}$$

$$\iint \frac{s(\lambda, x, y) d_x d_y}{\iint d_x d_y} = z \tag{5.10}$$

$$f(x, y) = z \int_w e(\lambda, x, y)\, \mathrm{c}(\lambda)\, \mathrm{d}\lambda \tag{5.11}$$

$$f(x, y) = ze \tag{5.12}$$

where, $e = [\mathrm{R}_e \mathrm{G}_e \mathrm{B}_e]^T$

$$\iint \frac{s(\lambda, x, y) d_x d_y}{\iint d_x d_y} = \frac{1}{\iint d_x d_y} \iiint_w e(\lambda, x, y) s(\lambda, x, y) c(\lambda) d_\lambda d_x d_y \tag{5.13}$$

$$\iint \frac{s(\lambda, x, y) d_x d_y}{\iint d_x d_y} = \int_w e(\lambda, x, y) c(\lambda) \left(\frac{\iint s(\lambda, x, y)}{\iint d_x d_y} \right) d_\lambda$$

$$\iint \frac{s(\lambda, x, y) d_x d_y}{\iint d_x d_y} = z \int_w e(\lambda, x, y) c(\lambda) d_\lambda = ze$$

$$\max_{x,y} . f(x, y) = ze \tag{5.14}$$

$$\left[\frac{\int (f(x, y))^p \, dxdy}{\int dxdy} \right]^{\frac{1}{p}} = ze \tag{5.15}$$

The color of the illuminant e is denoted as

$$ke^{\eta p \sigma} = \left[\sum \left(\frac{\partial^\eta f^\sigma(x, y)}{dx^\eta dy^\eta} \right)^p \right]^{\frac{1}{p}} \tag{5.16}$$

where η is the n[th] derivative order, (x, y) is the pixel coordinate, |.| is the absolute value, and $f^\sigma(x)$ observed intensities at position x, p is the Minkowski norm, k is a scalar factor, σ is the Gaussian kernel. The input layer is the first layer. Typically, a large dataset with millions of small size images is used to train the network. A small size image can be processed faster. The first layer comprises pixel information of the pictures. The size of the image input is 360×480×3. In the convolutional layer, 64 high pass filters are used for the initialization of the weights. Each

layer of the Convolutional Neural Network is connected to another neuron. Some regions of the image will be selected, and it will then slide over the whole picture to perform the dot product of all of these pixels. The convolution layer splits into feature learning and classification classes when cognitive processes are performed on the respective layer. Then convolution, batch normalization, and pooling operations are performed. In deep networks, these operations are performed multiple times to achieve better features. These various layers of convolution, batch normalization, Relu, and pooling, create a deep neural network.

The proposed deep convolution neural network (DCNN) consists of 91 layers. First, an input image of size 360×480×3 is applied to the proposed system. Then, image forgery localization is performed using the encoder-decoder network. The encoder consists of multiple layers of convolution, batch normalization, Relu, and pooling layers. These layers are stacked one after another in the same manner. After the first, second, third, fourth, and fifth pooling operations, the output is 128 3x3x64, 256 3x3x128, 512 3x3x256, 512 3x3x512, and 512 3x3x1024, respectively. The decoder consists of multiple layers of deconvolution, batch normalization, Relu, and pooling layers. These layers are stacked one after another in the same manner. After the fifth, fourth, third, second, and first unspooling operations, the output is 512 3x3x512, 512 3x3x512, 256 3x3x256, 128 3x3x128, and 64 3x3x64, respectively. In the end, we use SoftMax and Pixel Labels layers to predict the forged pixel colors.

Throughout the convolution and deconvolution operation, 3×3 filter size is used. The image pixel labels are defined as forged and not forged classes. The image frequency for forged pixels is 0.0805, and the not forged pixel is 0.9102. The class weights for forged pixels are 6.1550, and not forged pixels are 0.5442. The forged pixel count is 6.6145e+08, and the image pixel count is 8.2194e+09. The not forged pixel count is 7.481e+09, and the image pixel count is 8.2194e+09. In the encoder, 64, 128, 256, 512, and 512 filters are used in the 1^{st}, 2^{nd}, 3^{rd}, $4^{th,}$ and 5^{th} classifications. In the decoder, 512, 512, 256,128, and 64 filters are used in the 5^{th}, 4^{th}, 3^{rd}, $2^{nd,}$ and first classification. The first convolutional layer consists of three color channels, 64 output feature maps, with 3×3 filter size of weight matrices, then 128 weight matrices are required to initialize 64 fundamental filters. We define the piecewise learning rate with a learning rate drop factor of 0.5. The learn rate drop period is 5, Mini Batch Size is 11, and Max epochs are 20. To plots the training progress, we use a multi GPU execution environment with Verbose set to true.

5.5.2.1 FORGERY DETECTION AND LOCALIZATION ALGORITHM

Forgery detection and localization algorithm

Input: Take Forged image dataset and Ground Truth image dataset; **Output:** Detected forgery and performance matric

Load the Forged image dataset and Ground Truth image dataset.

Preprocess dataset using color illumination, rotation, scaling, and resize image size into 360×480×3.

Load pre-trained vgg16 model, define a path for the dataset, load forged images with Pixel-Label, identify new classes as Forged and Not Forged, determine pixel color labels and label IDs, prepare training data and test data, remove the last three layers of VGG16 network, add pixel label color layers and a softmax layer.

Select training options: SGDM, the learning rate is piecewise, drop out is 0.5, drop out period is 5, Max epochs are 20, Mini Batch Size is 11, plots the training progress, execution environment as multi-GPU, Verbose is true.

Perform data augmentation and apply preprocessing before training the VGG16 network.

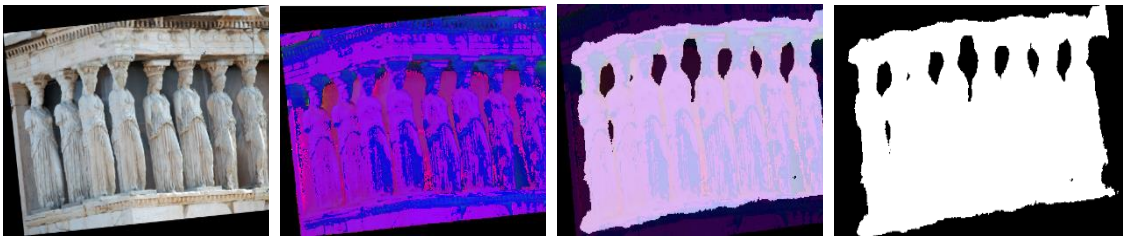
Perform training using (70%) of training data and testing using (30%) of test data

Prediction Label IDs: white for Forged and black for Not Forged; display the class names with the color.

Calculate Performance

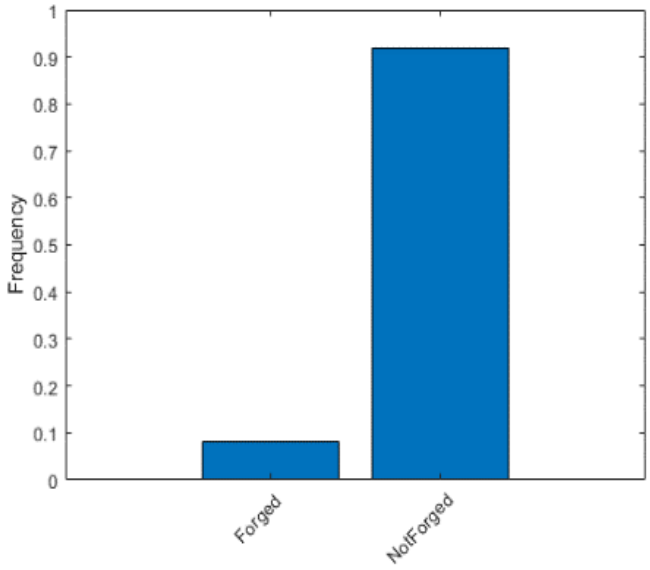
5.5.3 RESULTS OF THE PROPOSED ALGORITHM

In this section, numerous experiments are performed to test the proposed method. First, the dataset of 47566 images with 28540 training images and 19026 testing images is used. Then, to find copy-move forgery detection, splicing forgery detection, and rotation attack detection with different rotation factors, we use 47566 forged images from other datasets. Fig. 5.21 shows the forgery detection and localization results. Fig. 5.21(a) shows the input forged image, which undergoes the color illumination effect, as shown in Fig. 5.21(b). Next, we define the pixel color label for forged and not forged images, as shown in Fig. 5.21(c). These color labels are trained using deep convolution neural networks, and the final forgery segmented image is shown in Fig. 5.21(d).

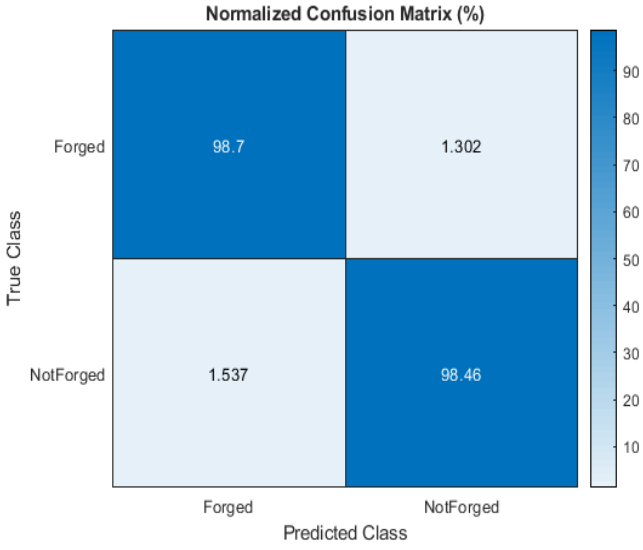


(a) (b) (c) (d)

Fig. 5.21 Forgery detection results. (a) Input forged image, (b) After color illumination, (c) Overlay image, and (d) Detected forgery.



(a)



(b)

Fig. 5.22 Bar graph of (a) Forged and not forged pixel frequency, and (b) Confusion matrix

Fig. 5.23 Intersection over the union between test image, truth, and prediction is IOU=1

Fig. 5.24 Intersection over the union between test image, truth, a prediction is IoU=0.28165

Fig. 5.22 shows the results in a bar graph of forged and not forged pixel frequency, confusion matrix, and image average IoU. Fig. 5.23 and 2.24 show the results for test image vs. truth vs. prediction and IoU results. Various experiments are performed using hyperparameter tuning. In the first experiment, training progress is shown in table 5.8 for 20 epochs, and its line graph between iteration and accuracy is demonstrated in Fig. 5.25. All training options are the same, as explained in algorithm 1.

Table 5.8 Training accuracy for twenty epochs using proposed deep learning methods

Epoch	Iteration	Time	Accuracy	Loss	Learning Rate
1	1	0:00:05	39.19%	0.7977	0.01
5	12950	8:43:13	91.63%	0.1741	0.01
10	25900	17:21:53	95.51%	0.0338	0.005
15	38900	25:59:42	94.09%	0.0482	0.0025
20	51880	34:43:26	95.16%	0.0375	0.0012

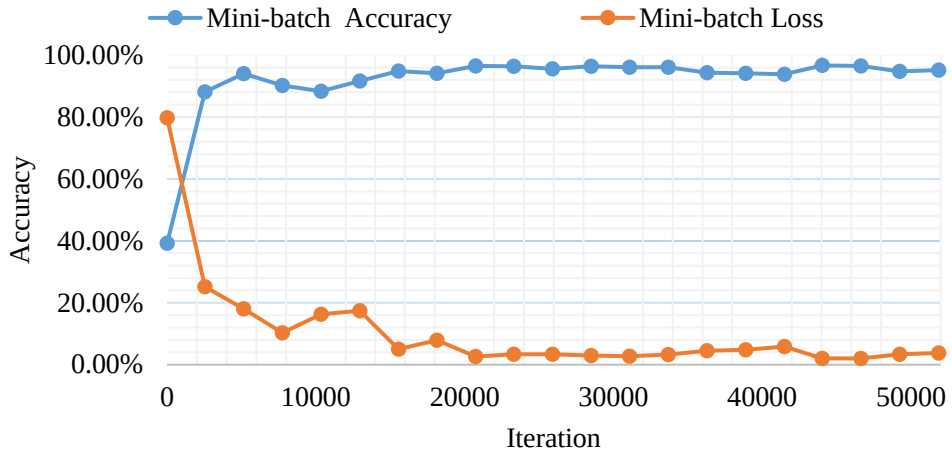


Fig. 5.25 Training plot between iteration and accuracy

Table 5.9 Test accuracy for twenty epochs using proposed deep learning methods.

Total Accuracy	Average Accuracy	Average IoU	Weighted IoU	Average Boundary F1 Score
0.98482	0.98581	0.91148	0.97193	0.86404

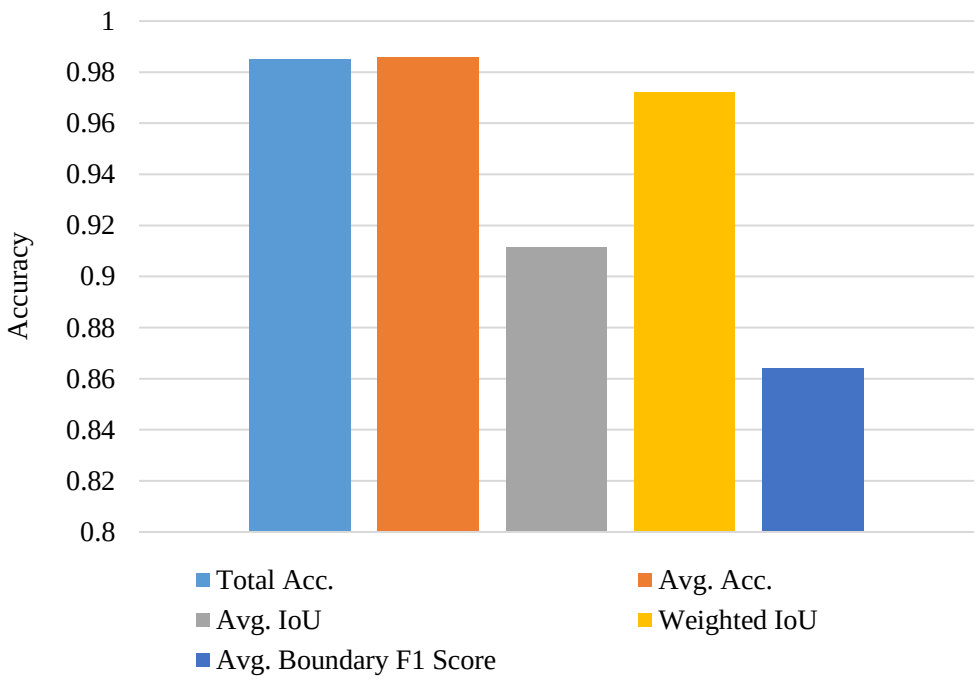


Fig. 5.26 Test accuracy using proposed deep learning methods

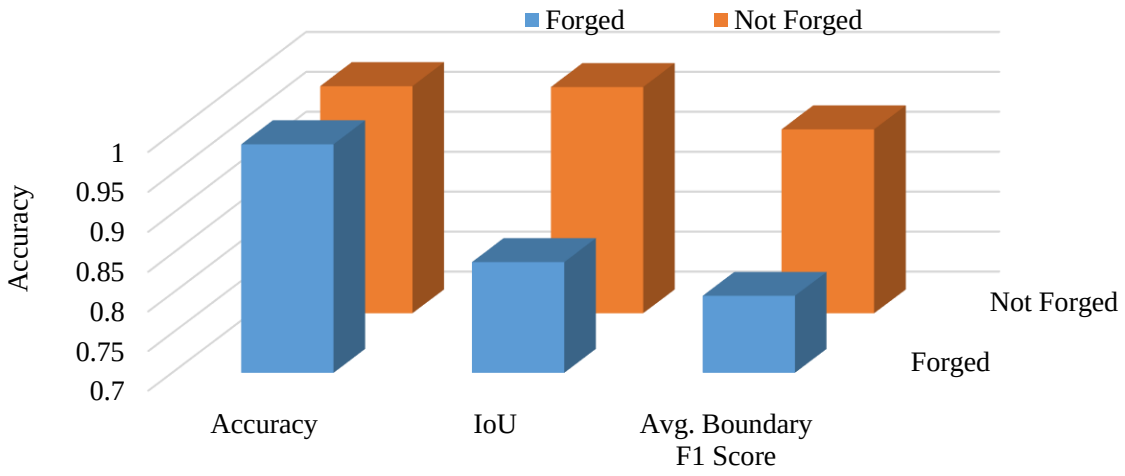


Fig. 5.27 Bar graph between forged, not forged pixels and test accuracy, IoU, average boundary F1 score

Table 5.10 Test accuracy, IoU, and average boundary F1 score using proposed deep learning methods

	Accuracy = TP / (TP + FN)	IoU = TP / (TP + FP + FN)	Avg. Boundary F1 Score
Forged	0.98698	0.83945	0.79709
Not Forged	0.98463	0.98351	0.93055

Table 5.9 shows the total accuracy 0.98482,average accuracy 0.98581, average IoU 0.91148, weighted IoU 0.97193, and average Boundary F1 Score 0.86404. Fig. 5.26 shows the bar graph of these values.

Table 5.11 Training accuracy for seven epochs using proposed deep learning methods

Epoch	Iteration	Time	Accuracy	Loss	Learning Rate
1	1	00:00:04	38.19%	0.8117	0.01
2	8150	3:40:29	94.96%	0.0667	0.009
4	16300	7:20:21	94.44%	0.0962	0.0073
6	24450	11:00:04	93.49%	0.1218	0.0059
7	28539	12:50:16	92.65%	0.0629	0.0053

Table 5.10 shows the Forged pixel accuracy 0.9869, IoU 0.83945, average Boundary F1 Score 0.79709, and Not Forged accuracy 0.98463, IoU 0.98351, Avg. Boundary F1 Score, 0.93055.

Fig. 5.27 shows the bar graph of these values. In the second experiment, training progress is shown in table 5.11 for seven epochs, and its line graph between iteration and accuracy is shown in Fig. 5.28.

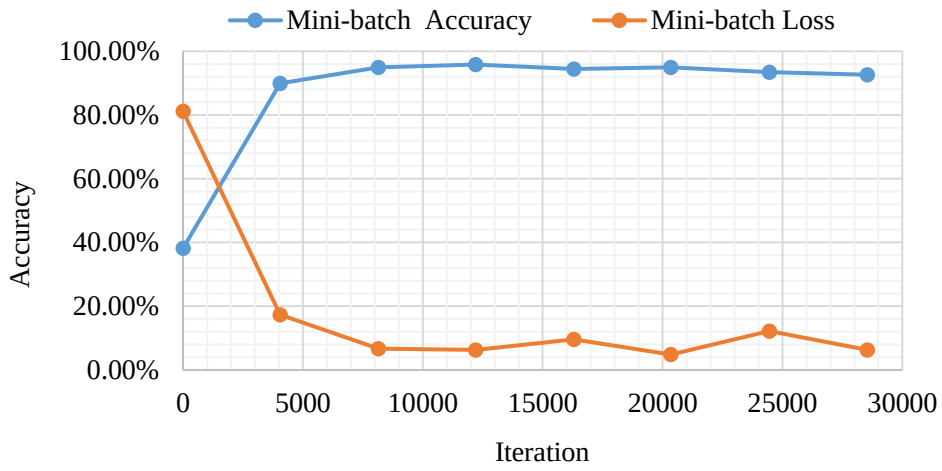


Fig. 5.28 Training plot between iteration and accuracy

Table 5.12 Test accuracy for seven epochs using proposed deep learning methods

Total Accuracy	Average Accuracy	Average IoU	Weighted IoU	Average Boundary F1 Score
0.95994	0.97407	0.81094	0.93307	0.74183

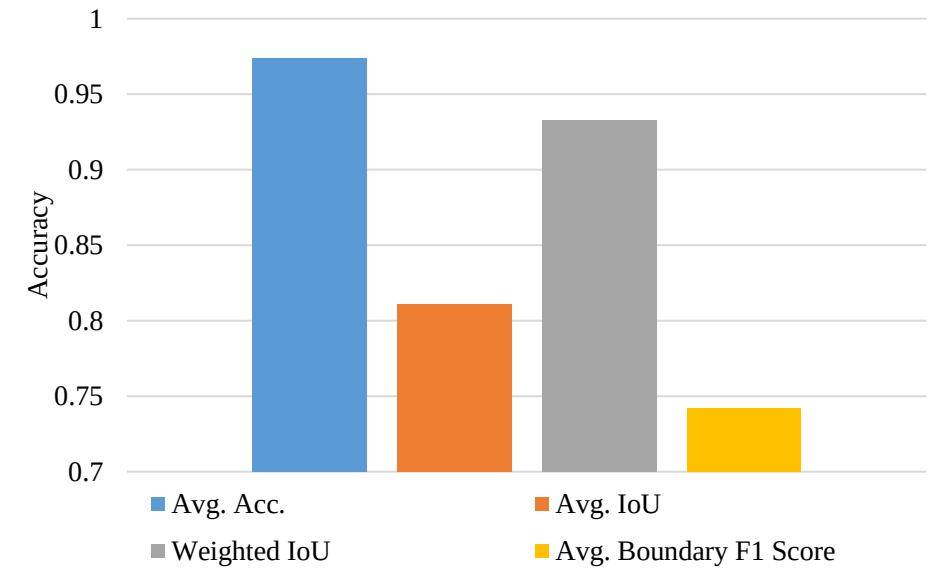


Fig. 5.29 Test accuracy using proposed Deep Learning Methods

Table 5.13 Test accuracy, IoU, and average boundary F1 score using proposed deep learning methods

	Accuracy = TP / (TP + FN)	IoU = TP / (TP + FP + FN)	Avg. Boundary F1 Score
Forged	0.9909	0.66541	0.62758
Not Forged	0.95723	0.95647	0.85577

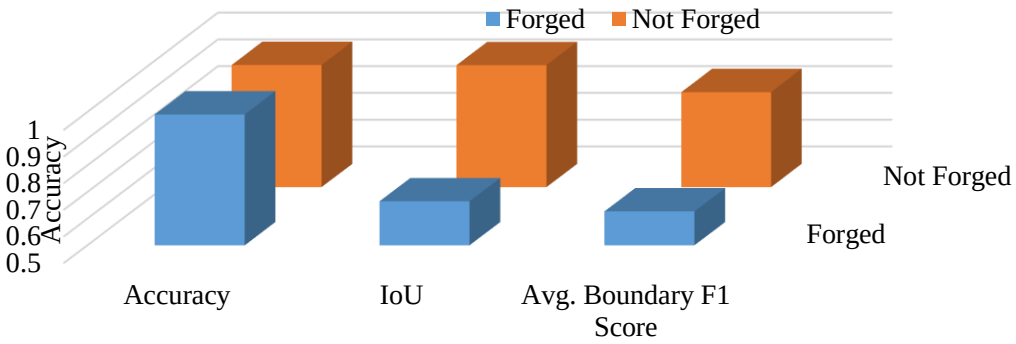


Fig. 5.30 Bar graph between forged, not forged pixels and test accuracy, IoU, average boundary F1 score

Table 5.12 shows the total accuracy 0.95994, average accuracy 0.97407, average IoU 0.81094, weighted IoU 0.93307, and average boundary F1 Score 0.74183. Fig. 5.29 shows the bar graph of these values. Table 5.13 shows the Forged pixel Accuracy 0.9909, IoU 0.66541, average boundary F1 Score 0.62758, and Not Forged accuracy 0.95723, IoU 0.95647, average boundary F1 score of 0.85577. Fig. 5.30 shows the bar graph of these values. In the third experiment, training progress is shown in table 5.14 for ten epochs, and its line graph between iteration and accuracy is demonstrated in Fig. 5.31.

Table 5.14 Training accuracy using proposed deep learning methods

Epoch	Iteration	Time	Accuracy	Loss	Learning Rate
1	1	0:00:04	38.06%	0.7908	0.01
2	5700	3:09:42	92.50%	0.108	0.009
4	11400	6:20:16	95.87%	0.0468	0.0073
6	17100	9:30:34	94.22%	0.0481	0.0059
8	22800	12:41:09	96.70%	0.0232	0.0048
10	28540	15:52:47	94.97%	0.0393	0.0039

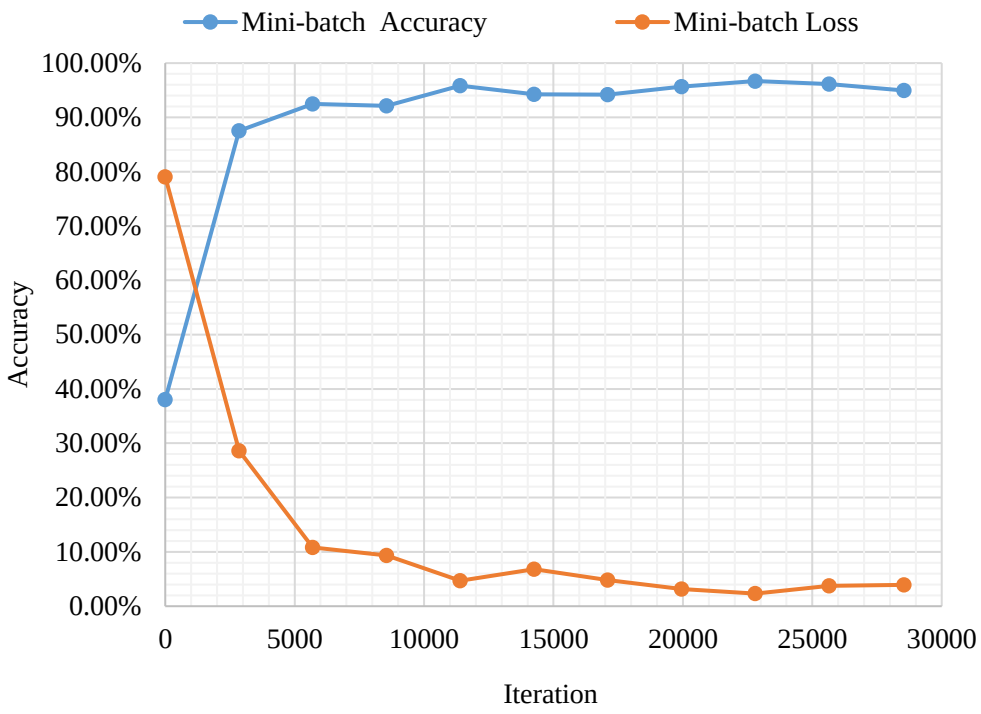


Fig. 5.31 Training plot between iteration and accuracy

Table 5.15 Test accuracy using proposed deep learning methods

Total Accuracy	Average Accuracy	Average IoU	Weighted IoU	Average Boundary F1 Score
0.9804	0.98224	0.89013	0.96447	0.82468

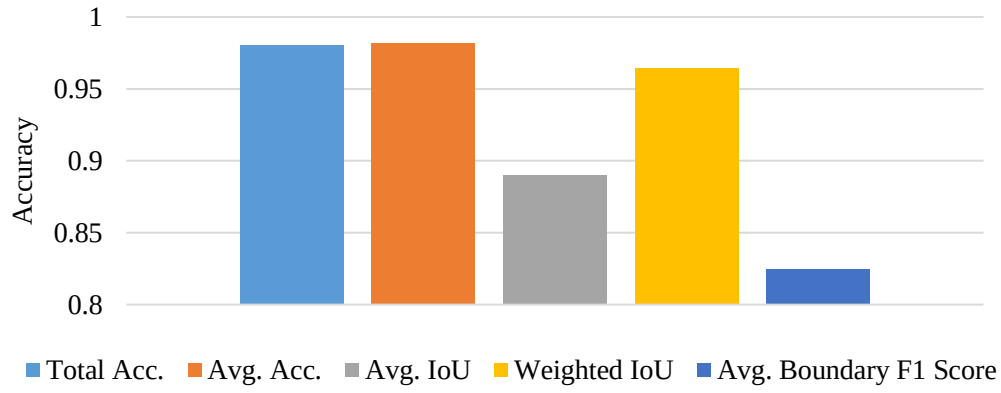


Fig. 5.32 Test accuracy using proposed deep learning methods

Table 5.16 Test accuracy, IoU, and average boundary F1 score using proposed deep learning methods

	Accuracy = TP / (TP + FN)	IoU = TP / (TP + FP + FN)	Avg. Boundary F1 Score
Forged	0.98442	0.80155	0.73837
Not Forged	0.98005	0.97872	0.91032

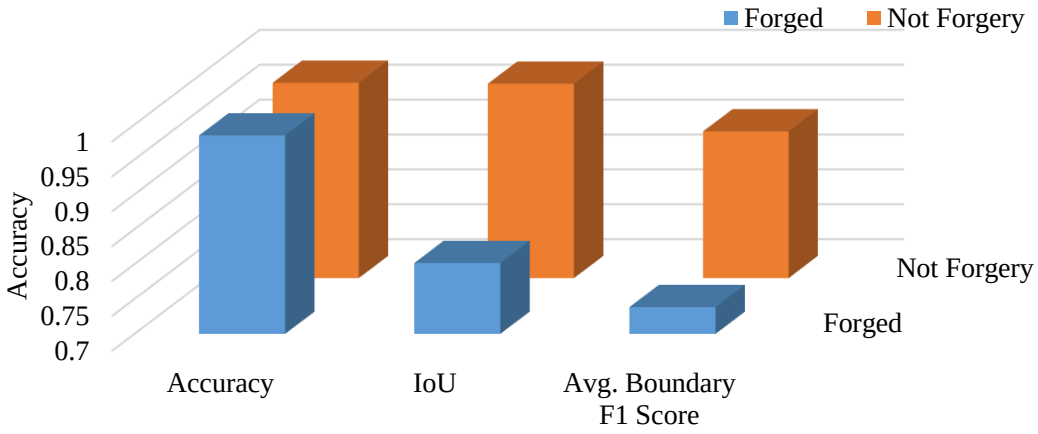


Fig. 5.33. Bar graph between forged, not forged pixels and test accuracy, IoU, average boundary F1 score.

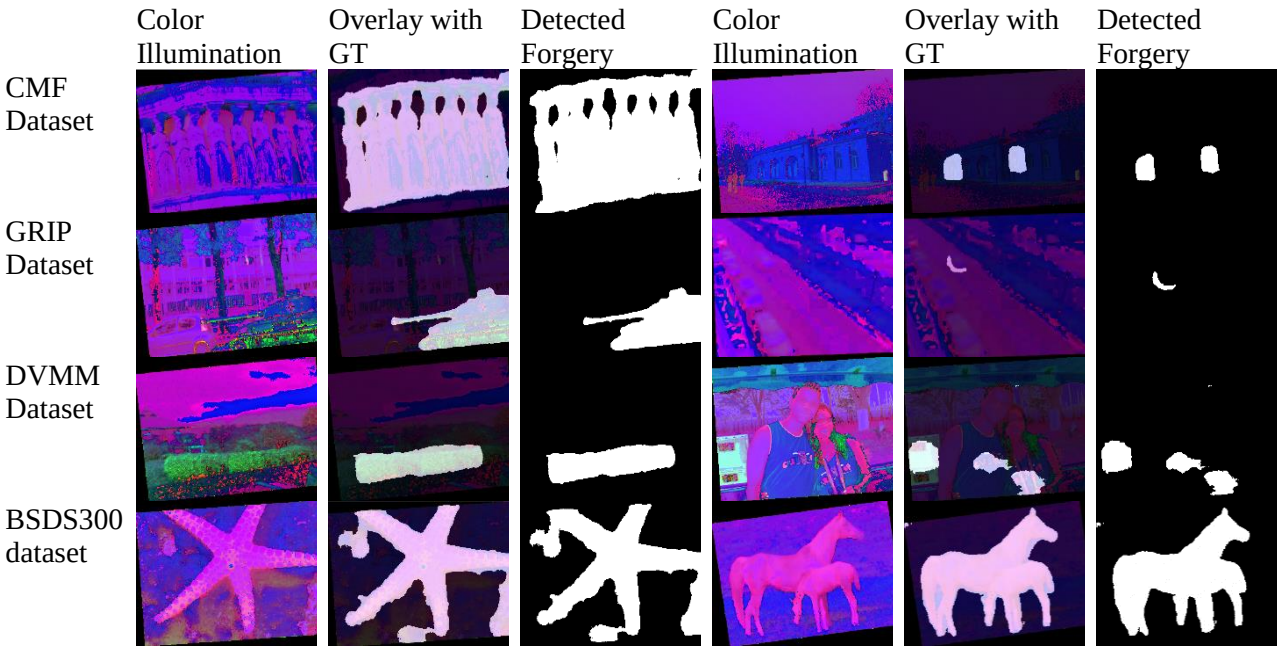


Fig. 5.34. Forgery detection results on color illuminated and rotated images using the proposed deep convolution neural network method.

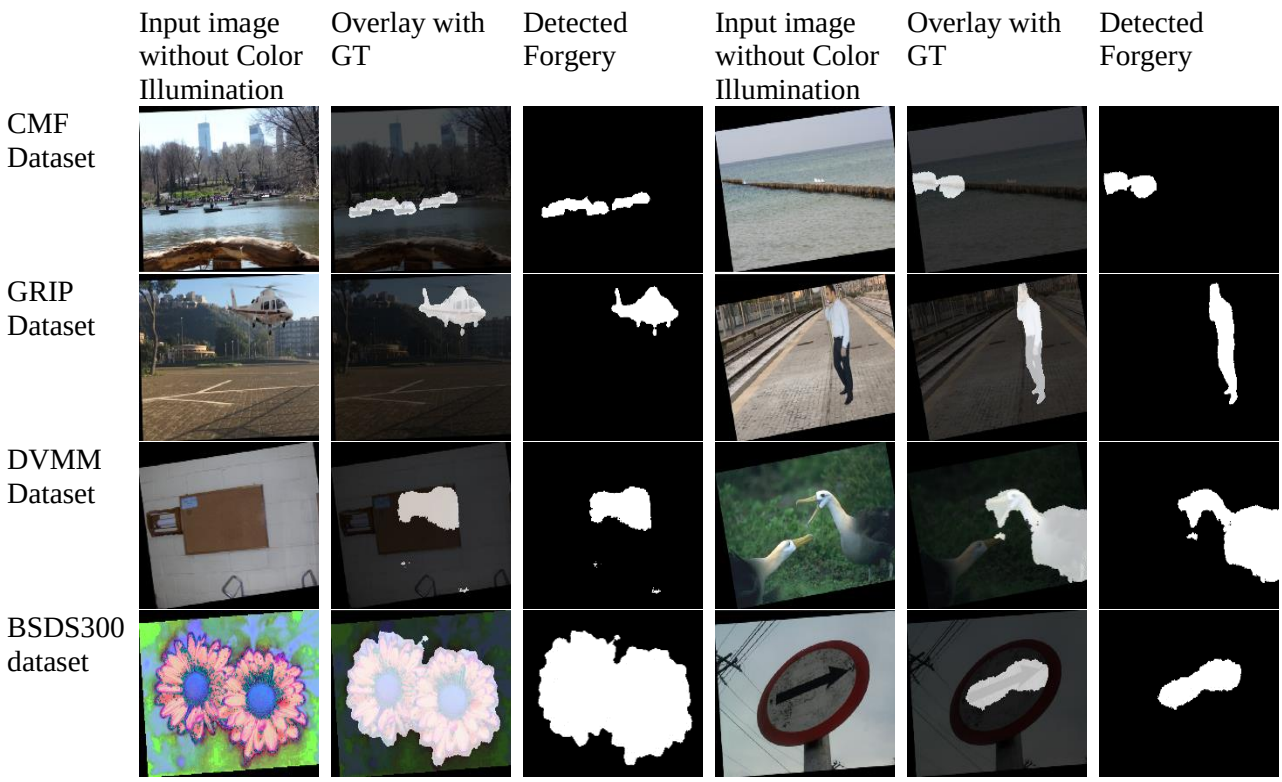


Fig. 5.35 Forgery detection results on without color illuminated and rotated images using the proposed deep convolution neural network method

Table 5.15 shows the total accuracy 0.9804, average accuracy 0.98224, average IoU 0.89013, weighted IoU 0.96447, and average boundary F1 score 0.82468. Fig. 5.32 shows the bar graph of these values. Table 5.16 shows the Forged pixel accuracy 0.98442, IoU 0.80155, average boundary F1 score 0.73837, and Not Forged accuracy 0.98005, IoU 0.97872, average boundary F1 score 0.91032. Fig. 5.33 shows the bar graph of these values. From these three experiments, we conclude that the training performed for a longer time gives better results. The learning rate drop factor and learning rate drop period also affect the performance of the model. We achieve good accuracy as compared with the state-of-the-art algorithms Liu [42], Bappy [43], Chen [44], Yang [45]. All these algorithms are based on a deep convolution neural network. These algorithms detect copy-move and splicing forgery with rotation attacks. The proposed algorithm is based on a deep convolution neural network, which uses a transfer learning approach to train a pre-trained VGG-16 network. This algorithm gives better forgery detection and localization results, as shown in Fig. 5.34 and Fig. 5.35. Fig. 5.34 shows the forgery detection results on color illuminated and rotated images. Fig. 5.35 shows the forgery detection results without color illuminated and rotated

images.

Table 5.17 The CMFD Dataset rotation attack results compared with other methods [2]

	SIFT [79]	SURF [80]	Circle [78]	RSIFT [2]	Bravo [77]	Pun [2]	Our
Rotation	46.7	52.8	68.7	80.8	87.3	83.8	91.1

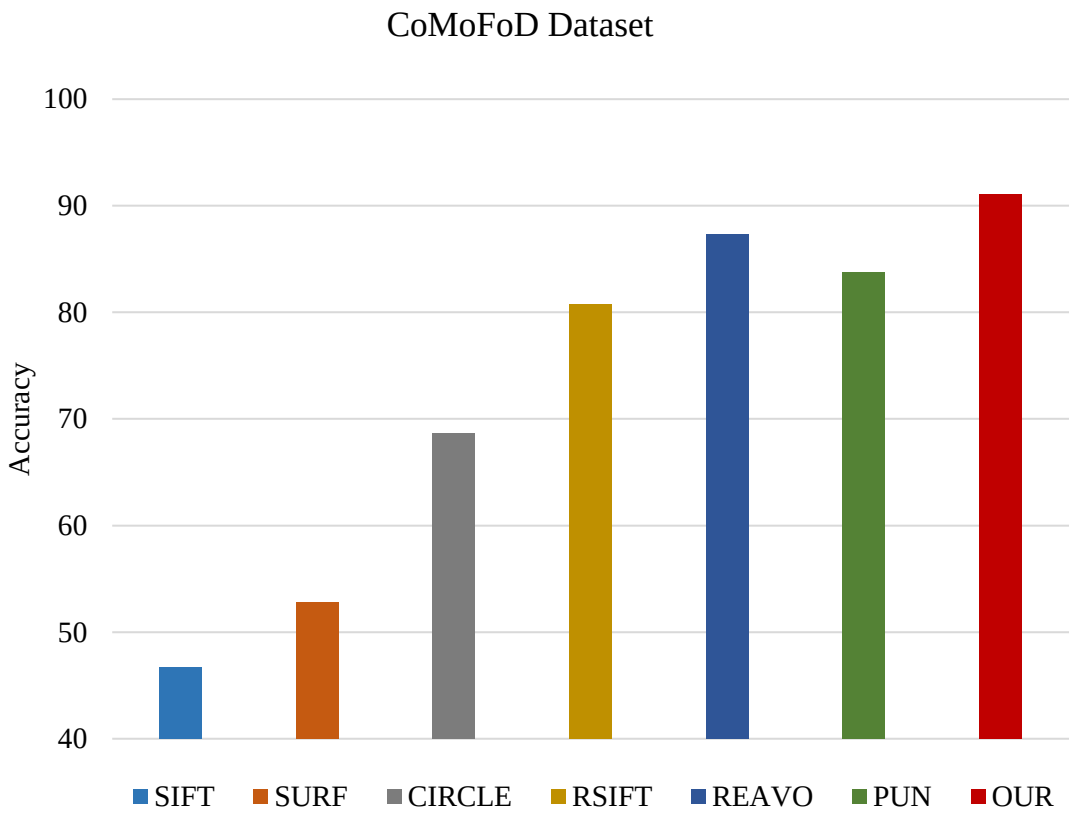


Fig. 5.36 Rotation attack results on CMFD Dataset comparison with existing methods [2]

Table 5.18 Comparison of the proposed method with other methods [33]

DVMM Dataset	
Authors	Accuracy (%)
Our	98.69
Yuan Rao *et al.*, [33]	96.38
He *et al.*, [40]	93.55
Zhao *et al.*, [42]	93.36

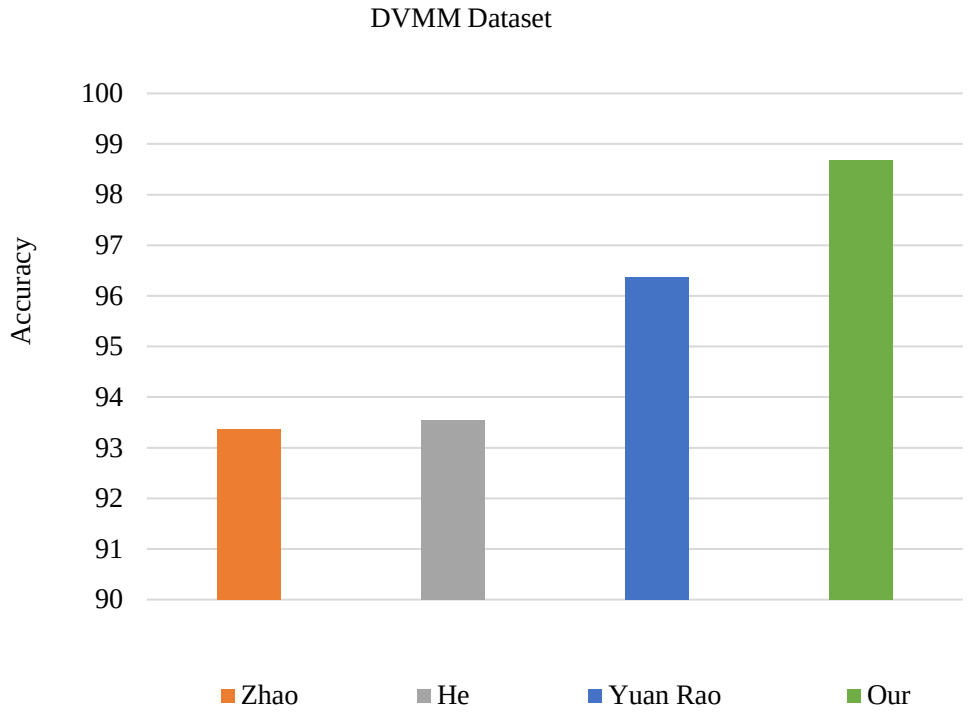


Fig. 5.37 CMF and SF detection result in DVMM dataset comparison Rao *et al.*, [33]

From the comparison table 5.17 and 5.18, our method improved to detect and localize image forgeries. Table 5.17, the CMFD dataset comparison, shows that our approach gives average accuracy (91.1%) at the pixel level. The graphical representation is shown in Fig. 5.36. Table 5.18, the DVMM dataset comparison, shows that our method gives global accuracy (98.69%) at the pixel level. The graphical representation is shown in Fig. 5.37.

5.5.4 PROPOSED MODEL ADVANTAGES, LIMITATIONS, AND OPEN PROBLEMS

The first step after classification is forgery localization. Classification simply means predicting what kind of forgery is present in an image. As we have seen, it is straightforward to classify fraud using transfer learning and DCNN. On the other hand, localization tells us in which part of the image forgery is present. It is a much more complex problem, which we have achieved using transfer learning DCNN and semantic segmentation. Visual Geometry Group is the research group that invented the VGG network in 2014. The VGG network is trained on the image net, which follows the architecture of the LeNet network. The pre-trained model is advantageous because it is trained on millions of images. Moreover, it saves a lot of time consumed by the

training process, reducing dimensionality.

Furthermore, it reduces the number of parameters, which reduces the chance of overfitting. Therefore, a deep neural network performs well, expresses more, and gives better generalization. We take the VGG pre-trained model and replace its last classification layer of 1000 classes. We used a multi-layer DCNN 91 layer model for training each pixel of the image. The 91 layers DCNN model generally performs better with image forgery detection and localization. The major advantage of transfer learning is that it works on small data and retrains its model. First, we take a shallow autoencoder and afterward just train it to reproduce the input. Next, we chop off the other half and use the first half of the initial layer of the neural network. In the second step, we do the same thing again. Still, instead of reproducing the original images, we try to reproduce the transformation from the previous layer. The same process repeatedly follows until we get the desired transformation. The training time and computation power requirement are major problems with a deep neural network.

Social networking platform has generated an enormous volume of image and video data over the last decade. Recently, deep learning techniques have begun dramatically shifting the way researchers develop new forensic algorithms. The CNNs learn forensic features from data, which a clever hacker can use to their advantage. GANs Yan *et al.*, [88] itself creates visually authentic images and poses a more significant threat to forensically authentic images. In particular, an attacker may use a GAN Goel *et al.*, [89] to train a generator that can falsify the forensic traces. Design forensic methods that use deep learning to protect against or track future attacks of this kind. CNN's capable of learning forensic features automatically, and they were developed to address several open issues in forensics, such as detecting fraud. The open-set picture forgery problems are designed to identify and locate content-conscious healing for duplication and splicing, clone stamping, deep learning adversarial assaults, facial or biometric spoofing identification, and forensic video and image deep learning.

5.5.5 PROPOSED MODEL COMPARATIVE EVALUATION

This chapter performs different experiments on image and video frame forgery using the DCNN model and semantic segment. The comparison of the proposed algorithm is given in table 5.19. The proposed algorithm achieves global accuracy (98.69%) at the pixel level, which is best compare to the state-of-the-art methods.

Table 5.19 Comparison of the proposed method with other methods

Ref.	Technique Used	Application	Accuracy (%)
[20]	CNN, VGG	DSO-I, SwapMe, Combined FaceSwap	Accuracy for DSO-I is 67.02, SwapMe is 77.43, Combined FaceSwap is 77.43, 71.43
[27]	k-Nearest Neighbor (kNN), SVM	illuminated images data set DSO-1 and DSI-1	DSO-1 achive 94% and DSI-1 achieve 84%
[90]	Deep Convolution Neural Networks, VGG	Signature dataset of (GPDS-160)	97.26%
[39]	VGG Face deep learning model	Face database	The equal error rate for S Gesture is 0.2157
Proposed	DCNN, Semantic Segmentation	BSDS300, DVMM, GRIP, and CMFD dataset	100% classification accuracy for CMF, SP, Frames. In Pixel level forgery localization 91.1% in CMFD, and 98.69% in DVMM

5.6 SUMMARY

Hybrid deep learning and machine learning approaches for passive image forensic are proposed in this chapter. In addition, the DCNN classifies images into forged and not forged categories. The key features of the current chapter are:

- The performance accuracy is calculated on the CASIA v1.0 validation set, and the test set is 98% and 99%, respectively. The performance accuracy is calculated on the CASIA v2.0 validation set, and the test set is 98% and 98%, respectively. The DVMM dataset forgery detection accuracy is 97%. The BSDS300 dataset forgery detection accuracy is 98%. The proposed algorithm is tested on image-level on CMFD dataset and achieved performance accuracy, i.e. Precision (P) = 98%, Recall (R) = 100% and F1 = 99%.
- The experiment result shows that total accuracy of 98.482%, an average accuracy of 98.581%, average IOU of 91.148%, weighted IOU of 97.193 %, average boundary F1 score of 86.404%, forged pixel accuracy is 98.698%, forged pixel IoU is 83.945 %, forged pixel average boundary F1 score is 79.709 %, Not Forged pixel accuracy is 98.463%, Not Forged pixel IoU is 98.351 %, Not Forged pixel average boundary F1 score is 93.055%. Thus, experiment results show that forged pixel and not forged detection accuracy are above 98%, which is better than art methods.

CHAPTER 6
GEOMETRICAL ATTACKS

Nowadays, a digital camera is available in the market at a meager cost, resulting in more digital images on social networks. These images are downloaded and modified by onlookers with user-friendly software. It becomes effortless to alter these images and create false propaganda. In some cases, such images are created as proof during some legal hearings. But these images are only considered after forensic investigation. The detection and localization of these types of forgery are vital problems. Therefore, to make the forensic analysis more trustworthy, so we require active research for value addition. The main reason to create CMF is to hide helpful information and make false propaganda Carvalho *et al.*, [12]. Image splicing is the type of forgery in which two images are merged into a single image. In this type of image, forgery items are composed of more than one picture. For example, in Fig. 6.1, some objects from one location are copied and pasted to another location on the same image.

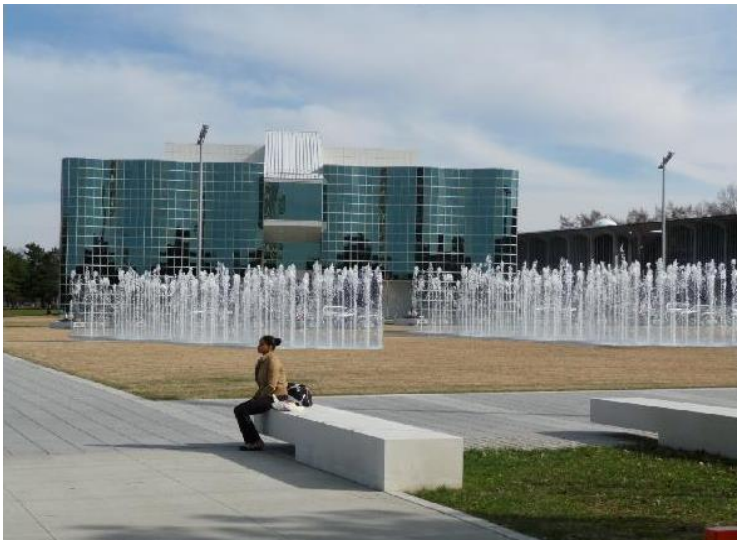


Fig. 6.1 Copy move forged image

The experimental results on various attacks will prove the efficacy of the proposed algorithm. Several performance parameters have been evaluated on attacks, and as an outcome, it has been observed that proposed algorithms are more robust than the existing techniques.

6.1 CMF, SF, AND GEOMETRICAL ATTACKS DETECTION

Nowadays, people send and receive digital pictures as communication on social media. Therefore, there is a chance to download and modify their pictures by anyone. Therefore, to detect image authenticity, active research is needed in this field for value addition. Editing software creates copy-move forgery (CMF) and splicing forgery (SF) very quickly Pun *et al.*, [2]. These are the most commonly used image forgeries. In this chapter, image forgery classification between these two categories is performed on different data sets. In CMF, few patches are copied and pasted in the same picture. In SF, two images are merged into a single image. Fig. 6.1 and 6.2 depict the CMF and SF. In this chapter, image forgery classification and detection, an algorithm is developed using Machine Learning (ML) and Color Illumination (CI).

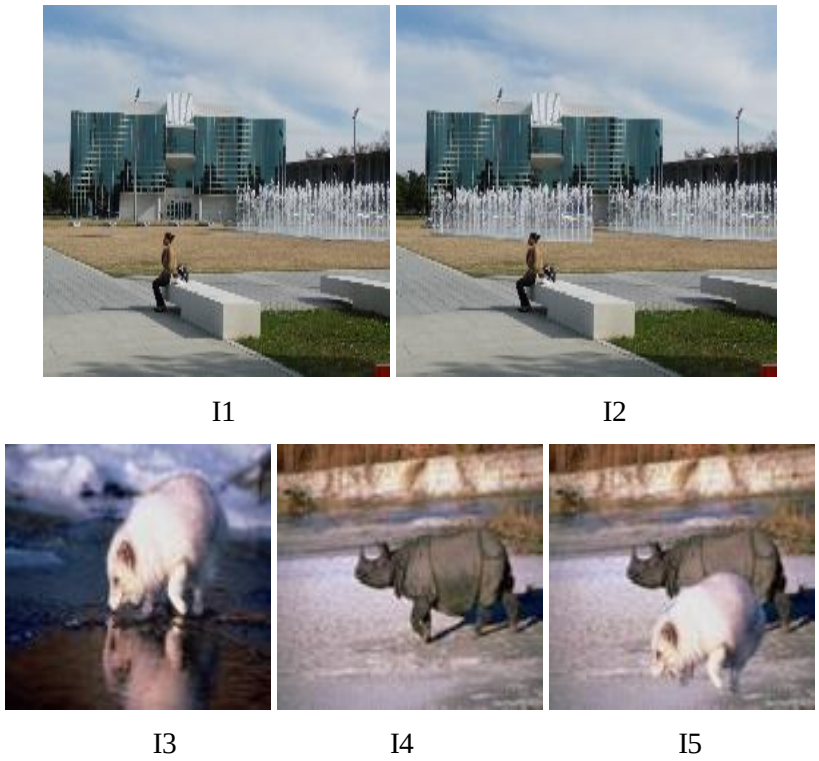


Fig. 6.2 Original images (I1), (I3), and (I4). CMF Forged Image (I2), Spliced image (I5)

In section two, firstly, the classification between forged v's authentic and CMF v's SF is performed. Secondly, the classification between scale v's rotation is performed, and lastly, the localization of CMF is performed. In section three, CMF attacks (JPEG, Scale, and Rotation) detection and localization are covered. Section four concludes the chapter by highlighting the salient features of DCNN and Color Illumination methods. Fig. 6.2 depicts the splicing forgery

in (I3) and CMF in (I4). Some objects from one image (I2) are copied and pasted on the other image (I1), which gives the spliced image in (I3).

6.2 FORGERY CLASSIFICATION AND LOCALIZATION ALGORITHM

The evolution and exponential growth in machine learning are becoming the most trending field of the 21st century. We have two main categories of problems. We have regression problems and have classification problems. Regression predicts a continuous quantity, whereas classification predicts between two classes with a true/false decision. In this chapter, we are using the deep CNN classification algorithm, as shown in Table 6.1. In this deep CNN classification algorithm, 86508 authentic images and 32028 spliced images are present. The first pseudo code explains the classification between forged v's authentic image categories and copy-move v's splicing image categories.

Table 6.1 Deep CNN layer's specification

S/N	Layers	Filter Size	Stride	Padding
1	Image Input	114x114x3		
2	Convolution	32 5x5x3	[1 1]	[4 4 4 4]
3	ReLU			
4	Maximum Pooling	5x5	[2 2]	[0 0 0 0]
5	Convolution	64 5x5	[1 1]	[4 4 4 4]
6	ReLU			
7	Maximum Pooling	5x5	[2 2]	[0 0 0 0]
8	Convolution	64 5x5	[1 1]	[2 2 2 2]
9	ReLU			
10	Maximum Pooling	3x3	[2 2]	[0 0 0 0]
11	2 Fully Connected			
12	ReLU			
13	2 Fully Connected			
14	Softmax Layer			
15	Classification Output	Cross-Entropy		

6.2.1 CLASSIFICATION OF VARIOUS GEOMETRIC ATTACKS

6.2.1.1 PSEUDOCODE FOR THE PROPOSED ALGORITHM

Input: Suspicious image (I).

Output: Detected result. Forged (Fo) or Authentic (Au); and (CMF) or Spliced (Sp).

Procedure

Apply Color Illumination

for each input image (I) **do**

Original = read Image (I)

Y= resize input image into [116 116]

cform = make cform ('srgb2lab')

J = apply cform (Y, cform)

Save each image into data set having two categories Authentic (Au) and Forged (Fo)

end for

for image classification between Au and Fo **do**

Apply deep convolution for feature extraction

Convolution:

$$h(x)=f \otimes g=\int f(x-u)g(u)du \quad (6.1)$$

$$h(x)=F^{-1}(\sqrt{2}\pi F|f|F[g]) \quad 6.2)$$

Feature map extraction:

$$\text{feature map}=\text{input} \otimes \text{kernel} \quad (6.3)$$

$$=\sum_{x=0}^{Row} \text{Input}(x-a, y-b)\text{kernel}(x,y)) \quad (6.4)$$

$$=F^{-1}(\sqrt{2}\pi F|\text{inputf}|F[\text{kernel}]) \quad 6.5)$$

$$\sum_{y=0}^{C} ((\sum_{x=0}^{R} \text{Input}(x-a, y-b)\text{kernel}(x,y)) \quad (6.6)$$

Probability value:

$$S(yi)=\frac{e^{yi}}{\sum_j e^{yi}} \quad (6.7)$$

Mean square error (MSE):

$$MSE=\frac{1}{n}\sum_{i=1}^{n} (ti-yi)^2 \quad (6.8)$$

Train the DCNN model with 70% training images

end for

for Test the DCNN model **do**

Test the DCNN model with 30% test images

end for

if forged **then**

Show forged images results and save them

else

Show authentic images results and save them

end if

for image classification between CMF and SF **do**

Apply deep convolution for feature extraction as shown above

Train the DCNN model with 80% training images

end for

for testing of DCNN model, **do**

Test the DCNN model with 20% test images

end for

if CMF **then**

Show CMF images results and save them

else

Show spliced images results and save them

end if

end procedure

The classification output is obtained into two categories, such as authentic or forged. The forged image category is divided into two sub-categories, such as copy-move and splicing.

6.2.1.2 PREPARE DATA SET FOR TRANSFER LEARNING

In this step, forged images of copy-move and splicing categories with various geometric attacks are collected from a different data set. The rotation and Scaling attacks are applied on CASIA-1 Dong *et al.*, [85], CASIA-2 (Dong *et al.*, [85], CoMoFoD Tralic *et al.*, [84] and DVMM Chang *et al.*, [90] data sets. The color information is extracted, and all the images are passed through

color illumination to detect edges better (Pun *et al.*, [2]. The data set is labeled into two categories as copy-move and splicing. All the images are resized into the same size and channels. Then training and validation data set is generated. This image data set with Copy Move and Spliced Categories (80% for training and 20% for validation) is given in our model. In this classification phase, the training is performed on an Intel i7 CPU of base speed 2.90 GHz with 16 GB of RAM, two 1070 STRIX NVIDIA graphic cards of 8 GB each CUDA capable. The MATLAB software is used for the computation, training, and validation of the DCNN network. In this software, a different MATLAB deep learning toolbox is used.

6.2.1.3 CLASSIFICATION OF CM AND SP IMAGES

The classification of copy-move v's splicing images with scaling attack and rotation attack achieved proper training and validation accuracy. A total of 2802 spliced images and 2658 copy-move images were used for training under scale attack. The average accuracy achieved with the training data set is 97.86, in which 96/4368 wrong classifications are predicted. The average accuracy on the validation set is 97.86, in which 24/1092 false classifications are predicted.

For copy move and splicing images under ROTATION attack total of 86508 spliced images and 32028 copy-move images under rotation attacks were given for training. The average accuracy achieved with the training data set is 99.23, in which 827/94828 wrong classifications are predicted. The average accuracy on the validation set is 99.29, in which 210/23708 false categories are predicted.

Table 6.2 depicts that there is a difference between the proposed algorithm and the existing algorithm. In this comparison, CASIA V1.0, V2.0, and DVMM data set are evaluated using a DCNN model on the validation data set. As a result, the achieved validation accuracy is better, as shown in Table 6.2.

Table 6.2 Comparison of validation accuracy for DVMM, CASIA V1.0, and CASIA V2.0

Methods	Proposed	Yuan Rao [33]	Muhammad [22]	Zhao [86]	He [87]
CASIA v1.0 [85] Acc. (%)	97.35	98.04	94.89	-	-
CASIA v2.0 [85] Acc. (%)	97.93	97.83	97.33	-	89.76
DVMM [90] Acc. (%)	97.86	96.38	-	93.36	93.55

6.3 LOCALIZATION OF VARIOUS GEOMETRIC ATTACKS

6.3.1 SETUP FOR FEATURE EXTRACTION

The input image is pre-processed using color illumination Carvalho *et al.*, [12] scaling and rotation with equal size and stored in the database. These images are classified as forged and authentic first and then copy move and spliced categories. The primary goal is to localize copy-move forgery using machine learning. In the second, third, and fourth steps, the image is divided into blocks using SLICO. Then, the image patterns are detected using SIFT, extracted features matched with each block using block-matching, and highlight these pixels using morphological operation. The pseudocode for localization of detected forgery regions is given below.

6.3.2 PSEUDOCODE FOR FORGERY LOCALIZATION

Input: Suspicious image (I).

Output: Localization of forged object (FD)

Procedure: Localization of Copy Move Image (I)

[Row, Column, Channel]=size (I)

If Channel=3 **then**

Let f(pix)=(f_{red}(pix), f_{green}(pix),f_{blue}(pix))^T ; RGB color

$$F(pix)=\int_{\Omega} e(\lambda, pix)s(\lambda, pix)c(\lambda)d\lambda \quad (6.9)$$

$$F(pix)=\int_{\Omega} e(\lambda, pix)s(\lambda, pix)c(\lambda)d\lambda \quad (6.10)$$

$$ke^{n,p,\sigma} = \left(\int \left|\frac{\partial^n f^{\sigma}(\text{pix})}{\partial x^n}\right|^p d\text{pix}\right)^{\frac{1}{p}} \quad (6.11)$$

Image=rgb2gray (I)

end

Block Size=Row × Column

Calculate initial size $S = Q \times R$ of the host image

$$ELF = \sum |CA4| \quad (6.12)$$

$$EHF = \sum \left(\sum |\text{CDi}| + \sum |\text{CHi}| + \sum |\text{CVi}|\right) \quad (6.13)$$

$$PLF = \frac{\text{ELF}}{(\text{ELF} + \text{EHF})} * 100\ \% \quad (6.14)$$

$$S = \sqrt{(0.02 \times \text{Q} \times \text{R})};\ PLF > 50\% \quad (6.15)$$

$$S = \sqrt{(0.01 \times Q \times R)};\ PLF \leq 50\% \quad (6.16)$$

N=Block Partition using SLICO (Image)

for i=1: N

for j=1: N

Lm_Z=dct2(N)

end

end

Do Feature extraction using SIFT algorithm

Do Lexicographical Sorting

Do knn search for Euclidean distance (Ed) between each feature vector

if | Ed| <Threshold range1 (0-1)? **then**

F1=Compute SIFT vector length (S_L)

else if S_L>Threshold range2 (20-30)? **then**

F2=feature matching knn (F1)

Apply SVM to classify (Matched block/Not matched block)

F3=Matched Block (f2)

F4=morphological Operation (F3)

FD= extracted feature vectors (F4)

else

Ignore these feature vectors

end if

end procedure

6.3.3 CLASSIFIER SELECTION AND MODELING

The SVM classifier extracts the exact boundaries of the forged object and divides each object pixel into different pixel color categories. These color categories distinguish each object boundary and classify them into object categories. Thus, the authentic and forged types are classified correctly using this classifier. In this chapter, we have developed this algorithm for huge data set to classify a particular type of forgery, and if it is present, we localize the forged region. This algorithm reduces the computation time to find a reproduced image for a sizeable data set.

6.4 EXPERIMENTAL RESULTS

Precision, recall, and F1 values are used to evaluate the model's performance. Precision (P) is the

ratio of detected localized forged pixels to an overall number of localized pixels. Recall (R) is a ratio of correctly localized fake pixels to a total number of fake pixels. F1 is the ratio of twice the product of precision and recall to the sum of precision and recall.

$$R=\frac{\text{correctly localized forged pixels}}{\text{number of forged pixels}} \tag{6.17}$$

$$P=\frac{\text{correctly localized forged pixels}}{\text{total number of localized forged pixels}} \tag{6.18}$$

$$F1=2\times\frac{R\times P}{R+P} \tag{6.19}$$

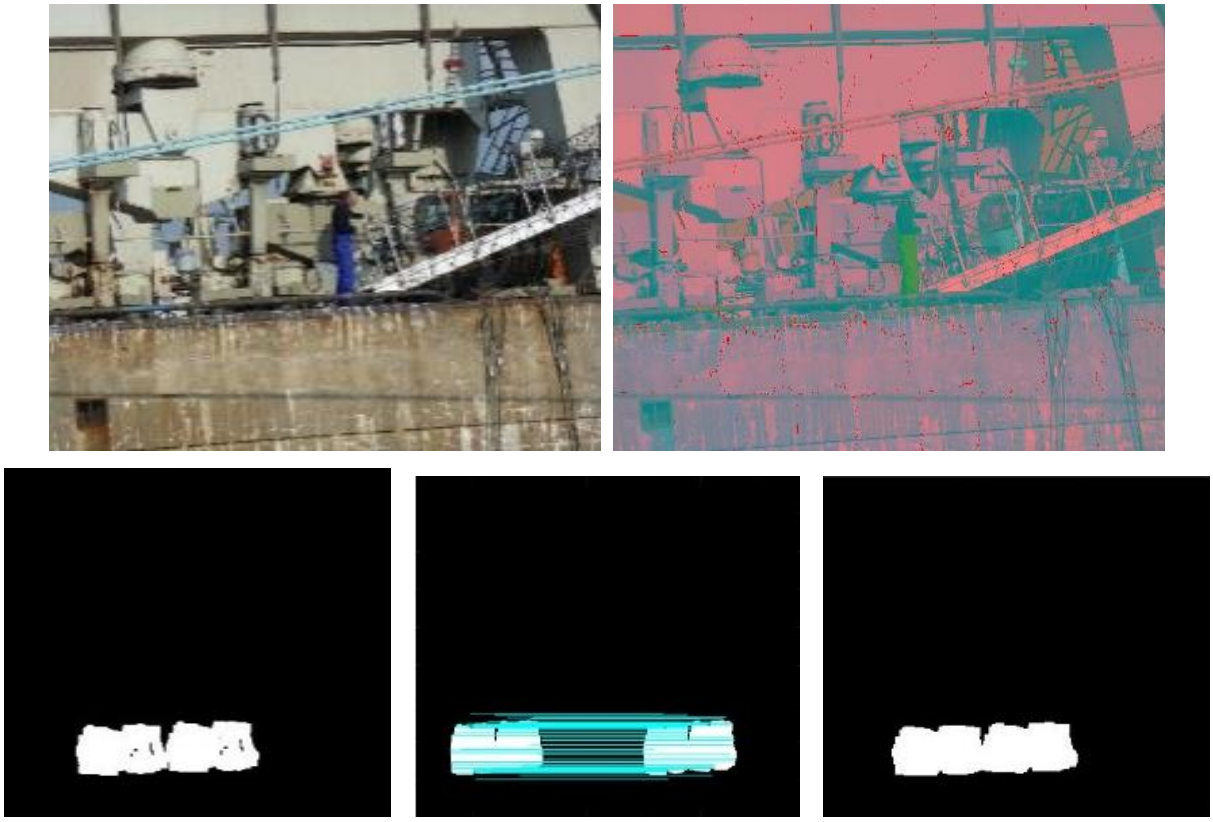


Fig. 6.3 Step by step execution results of the input image

Fig. 6.3 depicts the step-by-step execution results of the input image. The first image is the input applied to the model. The second, third, fourth, and fifth image SLICO, SIFT, feature matching, and image forgery localization results are shown.

Table 6.3 Copy-move forgery localization results in comparison with existing methods

Methods	Precision	Recall	F1 Value
Bravo [77]	87.2	100	93.2
Wang [78]	92.3	100	96
SIFT [79]	88.3	79.1	83.5
SURF [80]	90.4	89.5	90.5
Pun [2] Fix	95.9	97.9	96.9
Pun [2]Ad.	96	100	97.9
Proposed	97.2	100	98.5

Table 6.3 depicts a comparison of copy-move forgery localization results with an existing algorithm for different parameters. The achieved P, R, and F1 values of the proposed algorithm are better, as shown in table 6.3.

6.4.1 JPEG ATTACK LOCALIZATION

The proposed algorithm is tested on CoMoFoD Tralic *et al.*, [84] JPEG compressed images. All the pictures are JPEG compressed from J-20 to J-100 in the step of 20 are shown in a 1st row. All these 48 images are evaluated using a machine learning-based color illumination method and find out the average precision is 71.44%, average recall is 58.44%, and average F1 is 63.77%.

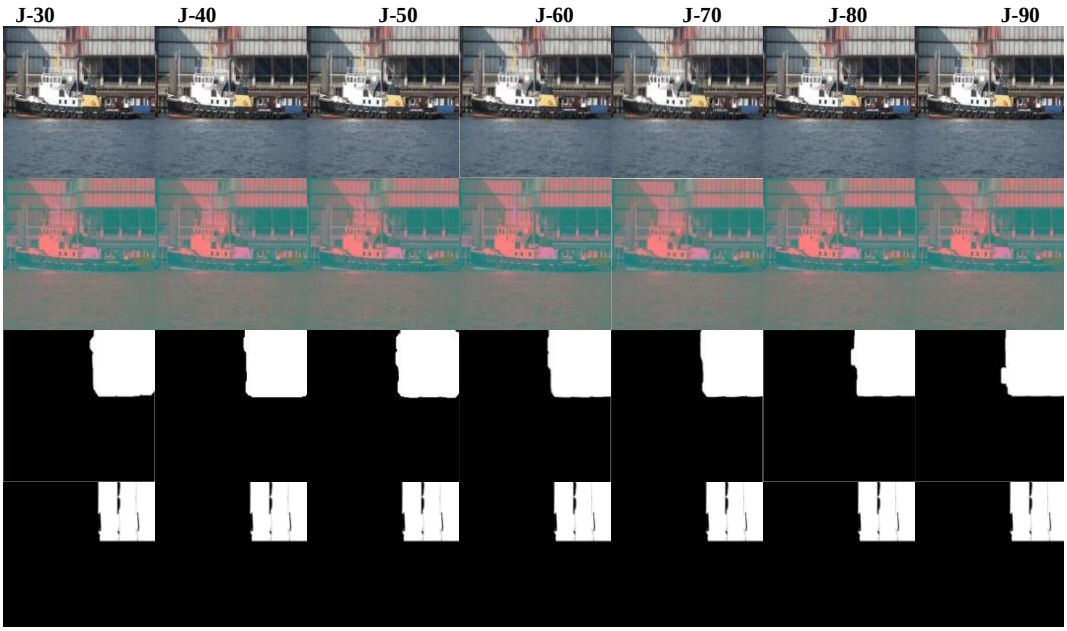
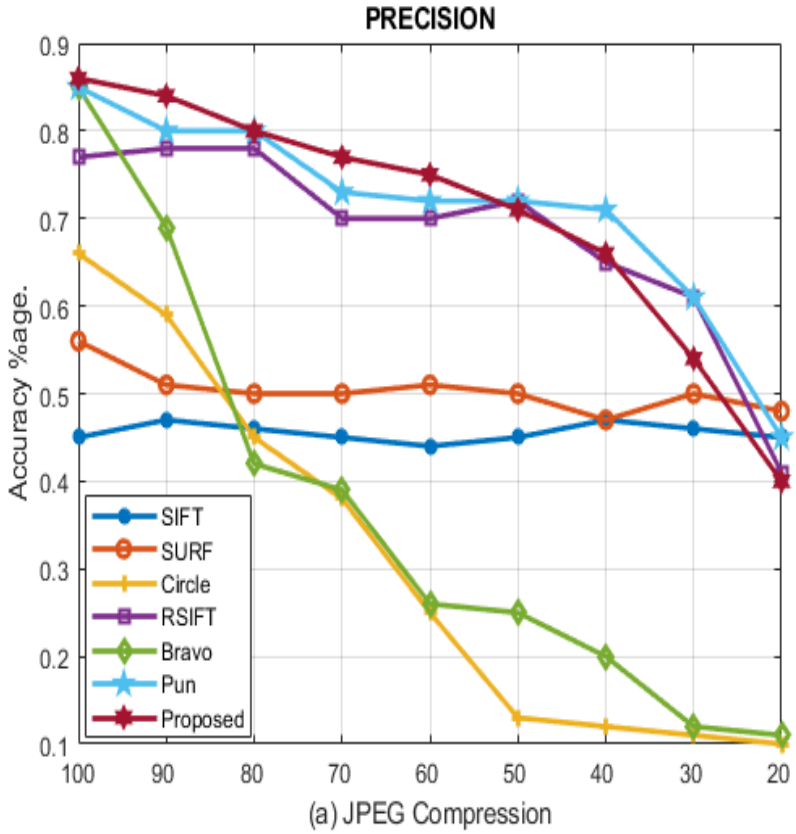
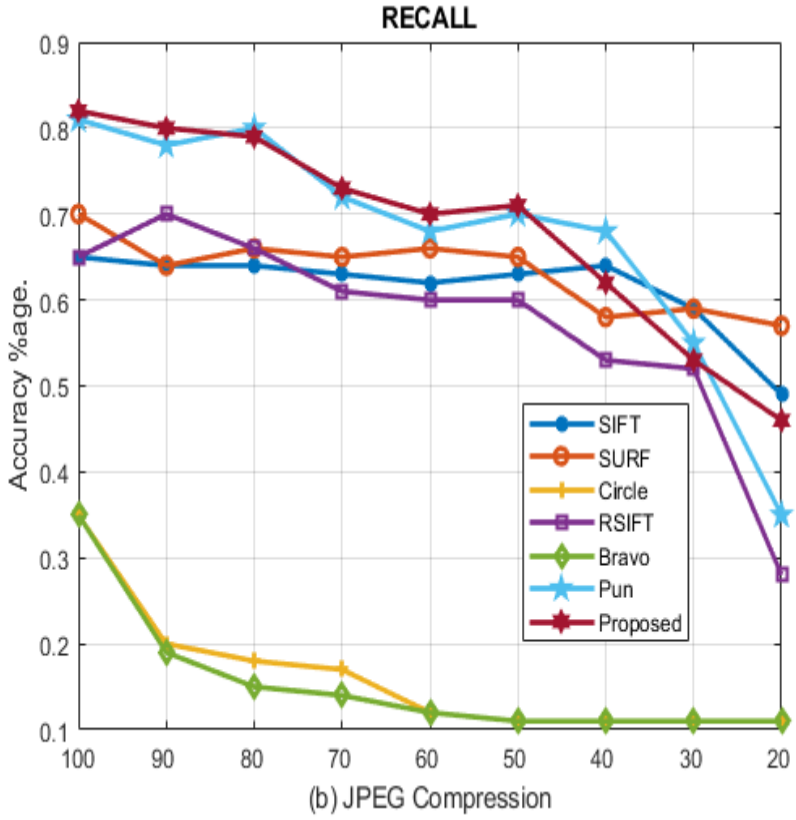


Fig. 6.4 Row-1 depicts the JPEG compression factor from J-30 to 90; Row-2 depicts color illumination for the corresponding image; Row-3 depicts the results of detecting forgery of a similar picture. Row-4 describes the ground truth image


PRECISION
Accuracy %age.
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
100
90
80
70
60
50
40
30
20
SIFT
SURF
Circle
RSIFT
Bravo
Pun
Proposed
(a) JPEG Compression


RECALL
Accuracy %age.
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
100
90
80
70
60
50
40
30
20
SIFT
SURF
Circle
RSIFT
Bravo
Pun
Proposed
(b) JPEG Compression

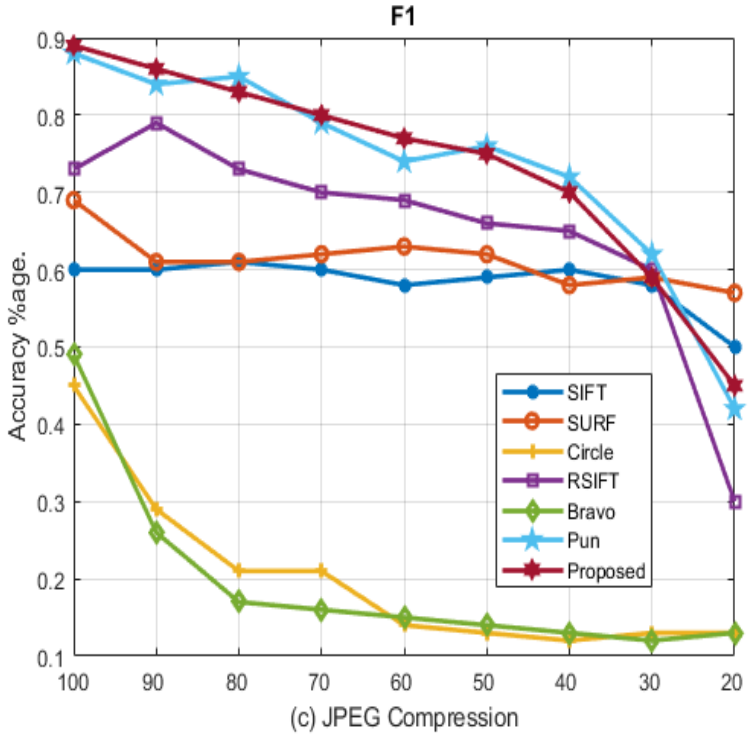


Fig. 6.5 Line graph between JPEG Compression attack and accuracy: (a) Precision, (b) Recall, and (c) F1

Fig. 6.4 depicts JPEG compression attack results at J-20 to J-100. The average precision is 71.44%, average recall is 58.44%, and average F1 is 63.77%. Fig. 6.5 depicts a line graph between JPEG Compression attack and accuracy. Fig. 6.5 shows the JPEG compression output of line graph (a) Precision, (b) Recall, and (c) F1.

6.4.2 SCALE ATTACK LOCALIZATION

The proposed algorithm tested on a CoMoFoD Tralic *et al.*, [84] scale attacked images. All the photos are scaled from S-91 to S-109 in step 2. Row 1 depicts the input images under various scale attacks. All images are scaled into proper size and channel. Then color illumination is applied to these images. Row 2 depicts color illuminated images. Color illuminated images are processed for feature extraction. Features are extracted using a scale-invariant feature extraction algorithm. These extracted features are analyzed using a support vector machine to predict forged pixels. Row 3 depicts detected forgery images. Row 4 describes ground truth images. Fig. 6.7 line graph depicts that the proposed algorithm performs better, and all the parameters are improved using the proposed method. All these 48 images are evaluated using a machine learning-based color illumination method and find out the Avg. Precision=85.2%; Avg. Recall=74.8%; Avg. F1=79.1%.

Line graph between scale attack and accuracy shown in Fig. 6.7. Fig. 6.7 (a) represents precision, (b) represents recall, and (c) represents F1. All these parameters are calculated for different algorithms. The line graph represents between scale factor (91-109) and accuracy. This line graph was plotted for scale-invariant feature transform, speeded up robust features, Circle, RSIFT, Bravo, Pun, and Proposed algorithm. All the algorithms are plotted with different colors.

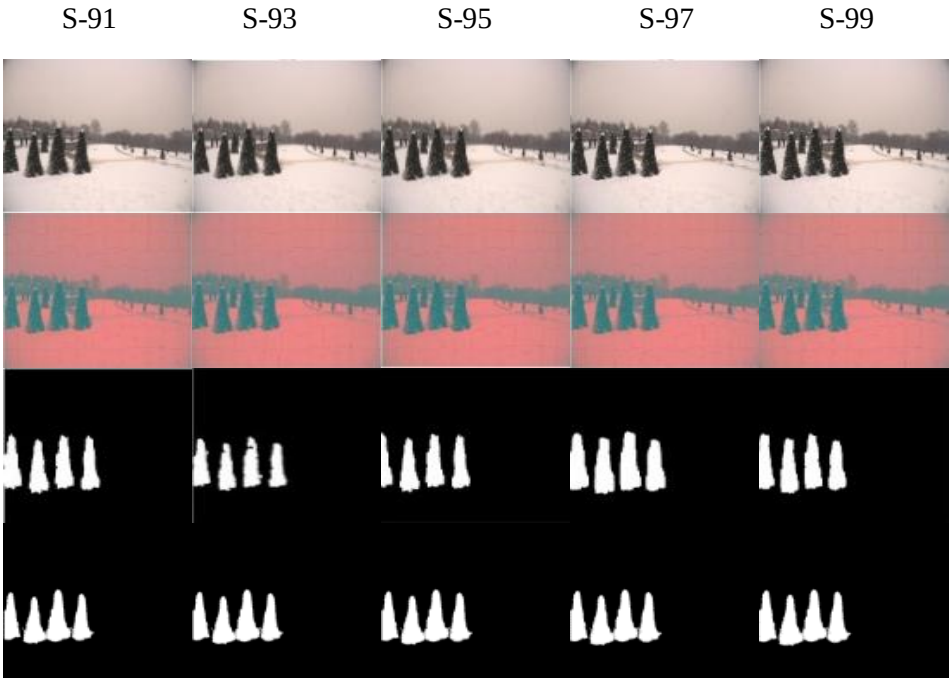
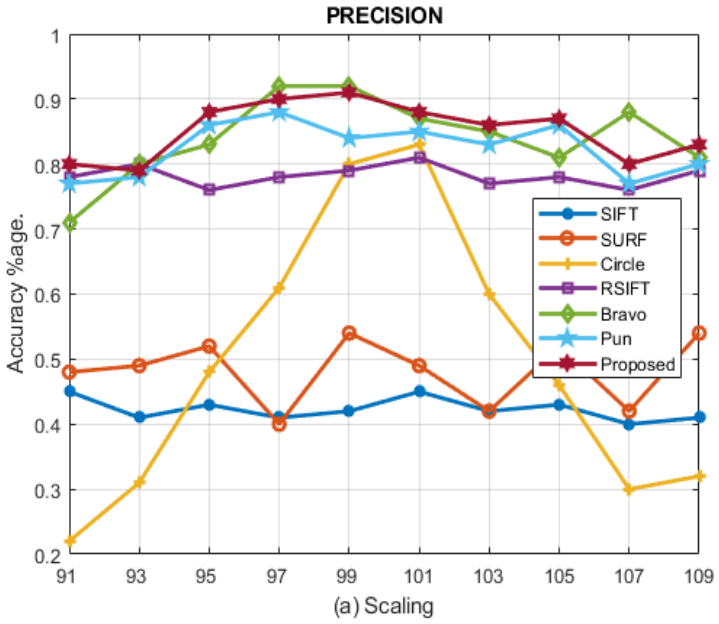


Fig. 6.6 Row-1 depicts the forged images with scale factor varies from S-91 to S-99; Row-2 depicts the color illumination; Row-3 depicts the forgery detection results; Ground truth images are shown in Row-4



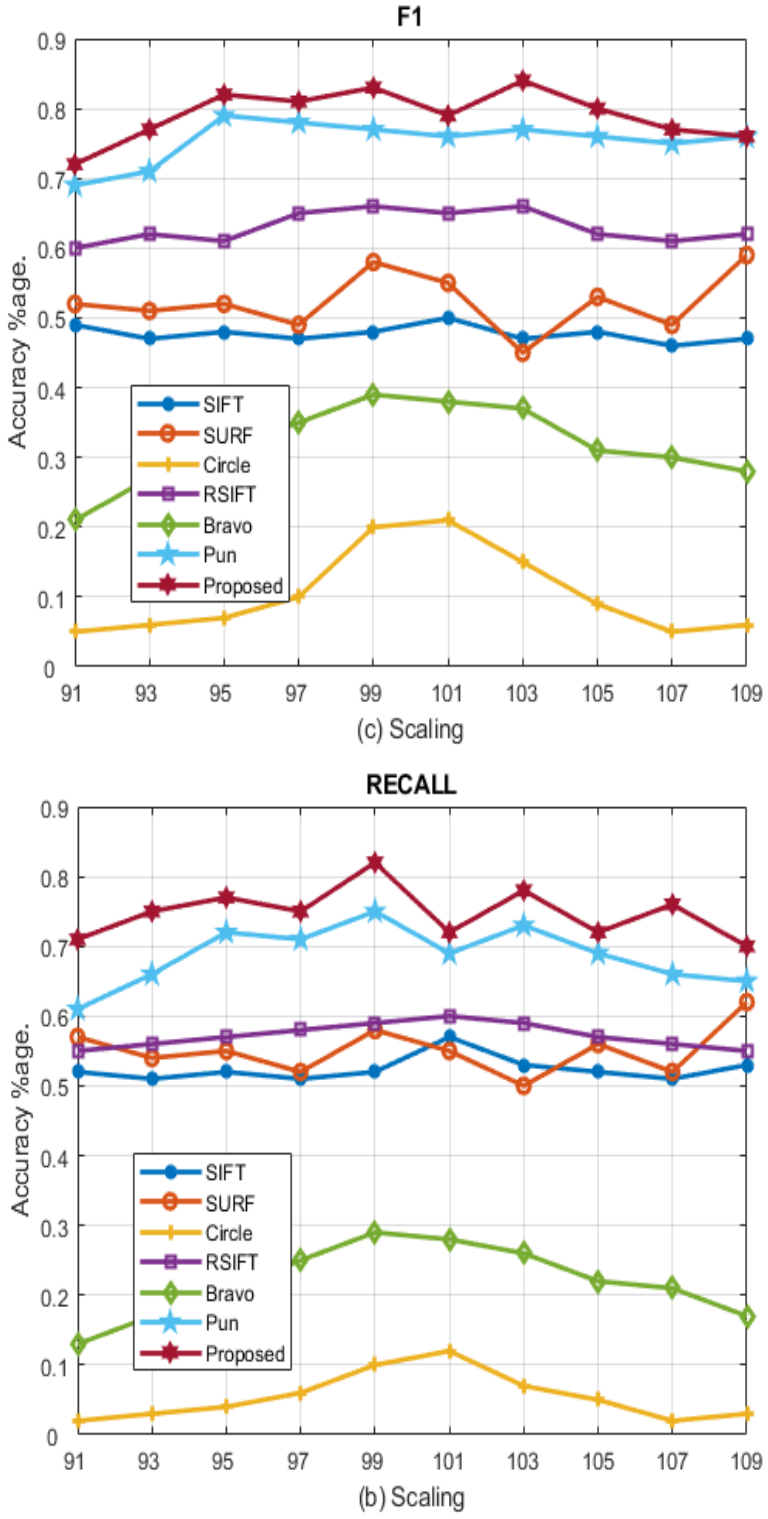


Fig. 6.7 Line graph between Scale attack and accuracy: (a) Precision, (b) Recall, and (c) F1

6.4.3 ROTATION ATTACK LOCALIZATION

The proposed algorithm was tested on CoMoFoD11 rotation attack images. All images are rotated and resized into the proper size and channel. Row 1 depicts the input images under various rotation attacks. Then color illumination is applied to these images. Row 2 depicts color illuminated images. Color illuminated images are processed for feature extraction. Features are extracted using the SIFT algorithm. These extracted features are analyzed using a support vector machine to predict forged pixels. Row C3 depicts detected forgery images. Column 4 depicts ground truth images. All the images rotated with 2°, 4°, 6°, 8°, and 10°. All these 48 images are evaluated using a machine learning-based color illumination method and find out the Avg. Precision=87.83%.; Avg. Recall=76.33%; Avg. F1=86.16%. The forgery localization results are shown in Fig. 6.8.

Table 6.4 depicts the comparison between existing methods to the proposed method. In geometric attack localization, we have tested scale attack, rotation attack, and JPEG compression attack. All these results were evaluated on the CoMoFoD data set. This data set is freely available. There are two types of the version available for this data set. The first type contains large-size images, and the second type includes small-size photos.

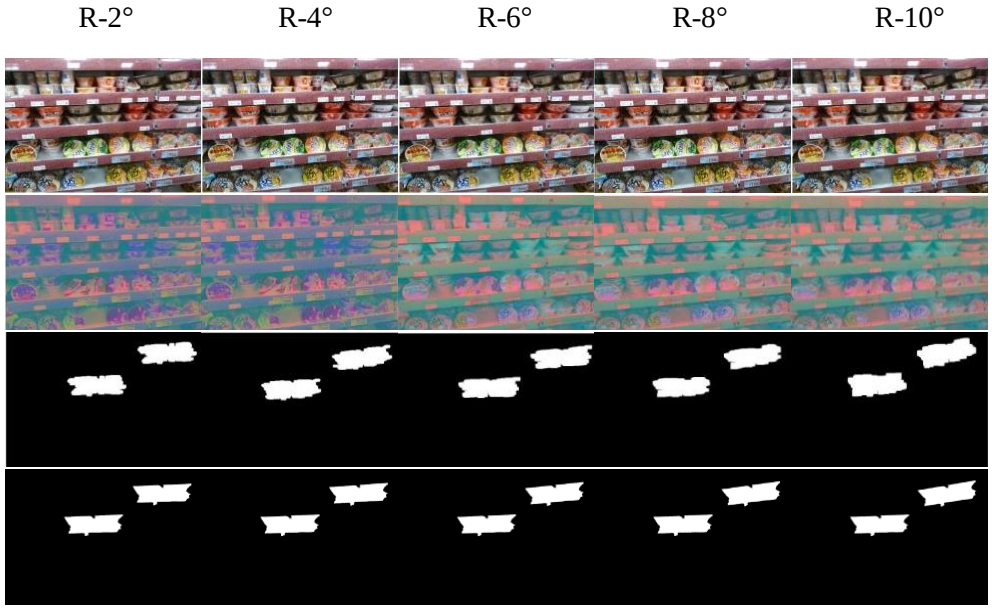
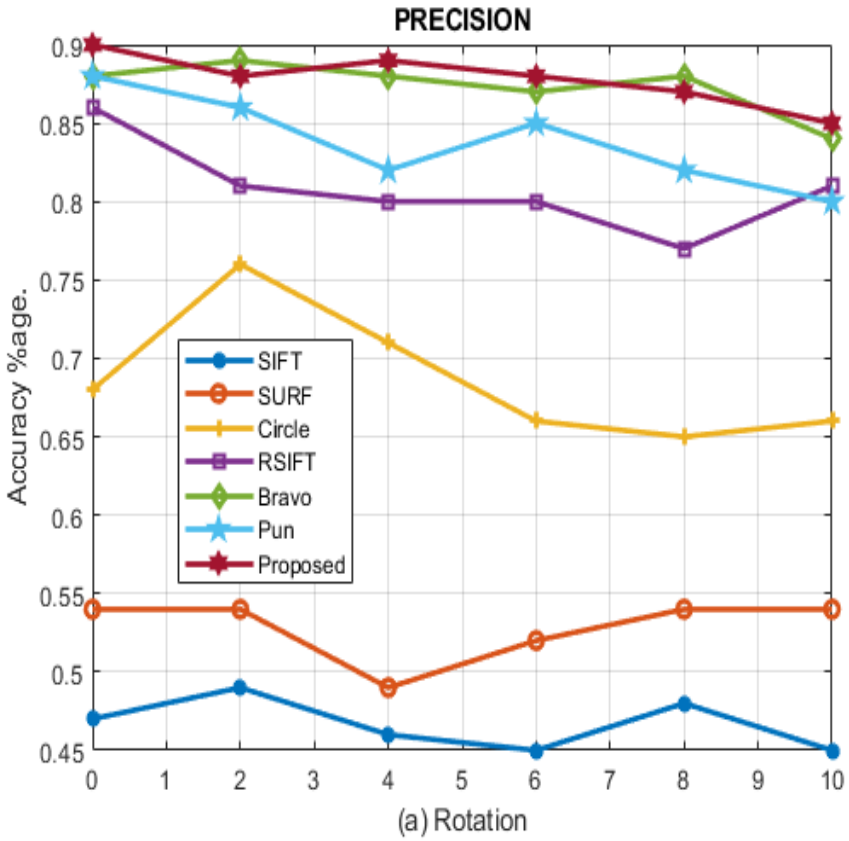
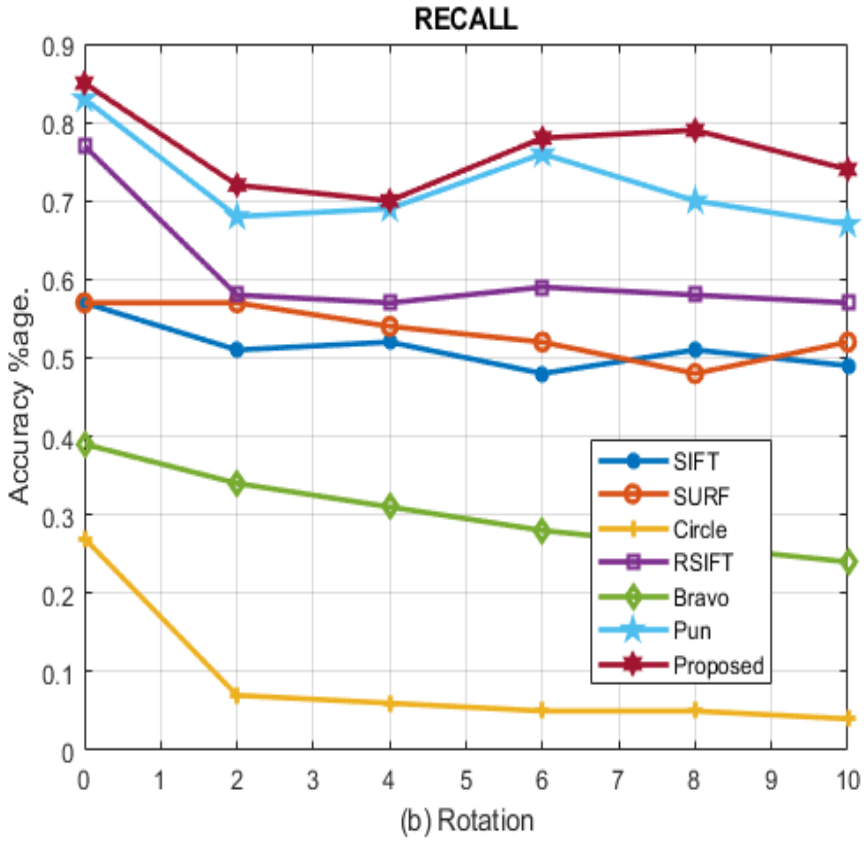


Fig. 6.8 The first row depicts forged images with a rotation factor from 2° to 10°; the second row represents the color illumination of the corresponding image; the third row illustrates the reproduced object localization result of the rotation attack; the fourth row depicts the GT images


PRECISION
Accuracy %age.
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0
1
2
3
4
5
6
7
8
9
10
SIFT
SURF
Circle
RSIFT
Bravo
Pun
Proposed
(a) Rotation


RECALL
Accuracy %age.
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
7
8
9
10
SIFT
SURF
Circle
RSIFT
Bravo
Pun
Proposed
(b) Rotation

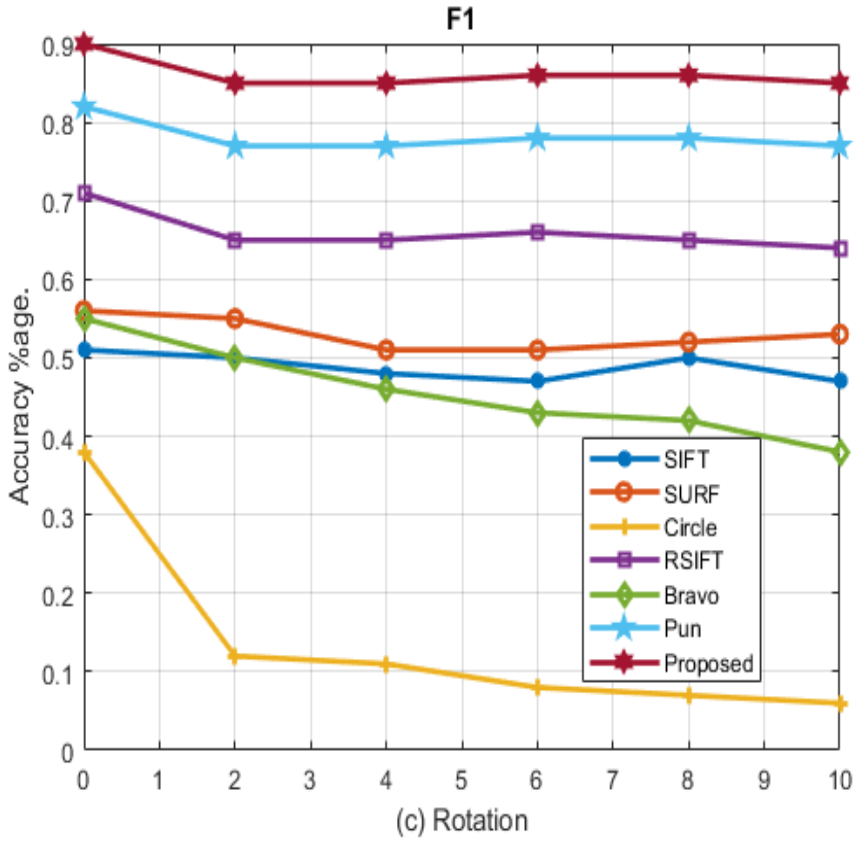


Fig. 6.9. Line graph between rotation attack and accuracy: (a) Precision, (b) Recall, and (c) F1.

Table 6.4 Geometric attack localization results in comparison with existing methods

Methods	Scale	Rotation	JPEG	Scale	Rotation	JPEG	Scale	Rotation	JPEG
	F1 Value			RECALL			PRECISION		
SIFT [79]	47.7	48.8	48.4	52.4	51.3	51.4	42.3	46.7	45.6
SURF [80]	52.3	53.0	50.3	55.1	53.3	53.3	48.2	52.8	50.3
Circle [24]	10.4	13.7	10.1	5.4	9.0	6.2	49.3	68.7	31.0
RSIFT [2]	63.0	66.0	55.0	57.2	61.0	47.2	78.2	80.8	68.0
Bravo [77]	31.7	45.7	9.4	21.9	30.3	5.4	84.0	87.3	84.0
Pun [2]	75.4	82.0	63.6	68.7	72.2	57.4	82.4	83.8	71.0
Proposed	79.1	86.2	63.8	74.8	76.3	58.4	85.2	87.8	71.4

6.5 SUMMARY

- The experimental results show better Precision, Recall, and F1 as compared with existing methods. The plain CMF attack detection results are: P=97.25%; R=100% and F1=98.53%. The

JPEG CMF attack detection results are: P=71.44%; R=58.44% and F1=63.77%. The scale CMF attack detection results are: P=85.2%; R=74.8% and F1=79.1%. The rotation CMF attack results are: P=87.83%; R=76.33% and F1=86.16%.

- In this chapter, color illumination-based block, a key point-based approach, is used to detect copy-move forged patches from the CMFD image manipulation dataset. Using this technique, rotation, scaling, and JPEG attacks are detected.
- After detection, forged patches, precision, recall, and F1 values are calculated. Also, a comparison is made between proposed methods of forgery detection with existing algorithms.
- The proposed work can be implemented in real-world applications in the future for national, social, and economic benefit. Nowadays, social media is widely used to communicate information in the form of images. However, these images may be prime fake news sources due to free photo editing software like GNU Gimp, Adobe Photoshop, etc. So, before any action is taken based on a received image, the authenticity must be verified. Proposed work can be used to find out single and multiple forgeries in the images to check the integrity of images. Some other applications are banking, military, etc.

CHAPTER 7

CONCLUSIONS AND FUTURE SCOPE

From the analysis and simulated results being performed and reported in the previous chapters, conclusions have been drawn here. The decisive notes are encompassed in the following section, along with the future scope of the work.

7.1 CONCLUSION

The color illumination, block, key point, machine learning, and deep learning algorithms are proposed for image forgery classification and localization. Several methods are available in the literature to classify and localize image forgery. The traditional algorithms have time complexity for the detection of large image datasets. The machine and deep learning-based algorithms learn from training and have less time complexity. During training, these algorithms take much more time, but when the model is trained, they work very fast compared to the traditional algorithms. In chapter 2, a literature survey is given on block, key point, machine learning, and deep learning-based image forgery detection techniques. In chapter 3, the basics of image forgery, ML, and DL are given. The step-by-step supervised and unsupervised approaches are discussed. In chapter 4, the copy-move forgery detection using color illumination, block, and key point-based technique is proposed. Finally, the SVM classifies image forgery correctly.

The precision, recall, and F1 values are compared with other methods and show good improvements. In chapter 5, the CMF and SF are detected using a deep convolution neural network. Two different approaches are proposed in this chapter. In the first approach, hybrid DL and ML methods are used to classify and detect a forged patch. In the second approach, DCNN with Semantic segmentation use VGG 16 model to classify and localize forgery. Geometrical attack detection and comparative analysis to justify the superiority of the proposed approach over the existing one is included in chapter 6. The experimental results show better Precision, Recall, and F1 as compared with existing methods. The plain CMF attack detection results are: P=97.25%; R=100% and F1=98.53%. The JPEG CMF attack detection results are: P=71.44%; R=58.44% and F1=63.77%. The scale CMF attack detection results are: P=85.2%; R=74.8% and F1=79.1%. The rotation CMF attack results are: P=87.83%; R=76.33% and F1=86.16%. Finally,

ML and DCNN are efficient, more flexible, and more powerful tools for image forgery classification and localization.

7.2 FUTURE SCOPE OF WORK

In the future, more efficient ways for detection of CMF, SF, and geometrical attack forgery can be searched. Moreover, an effort should also be made to find video forgery detection using ML and DL-based methods.

Similar to ML, the DL algorithms can be developed to efficiently detect digital image forgery applications, such as copy-move forgery, splicing forgery, retouching, scaling, rotation, and JPEG image compression forgery of the image.

The ML and DL can also be used to analyze video forging for other applications such as interframes and intraframes.

In the future, several other types of forgery, such as splicing, can be implemented on adaptive over-segmentation, color illumination, and feature-point matching on different kinds of media, for example, video and audio. In addition, future work may focus on increasing the accuracy rate of the proposed algorithm in video and audio forgery detection. Finally, although the usage of this system is generally limited to forensics, in the future, this system can also be implemented to filter out the content on social media to eliminate fake news and malicious content.

REFERENCES

[1] I. C. Chang, J. C. Yu, and C. C. Chang, "A forgery detection algorithm for exemplar-based inpainting images using multi-region relation," *Image Vis. Comput.*, vol. 31, no. 1, pp. 57–71, 2013, doi: 10.1016/j.imavis.2012.09.002.

[2] 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.

[3] A. Singh, G. Singh, and K. Singh, "A Markov based image forgery detection approach by analyzing CFA artifacts," *Multimed. Tools Appl.*, vol. 77, no. 21, pp. 28949–28968, 2018, doi: 10.1007/s11042-018-6075-5.

[4] L. Su, C. Li, Y. Lai, and J. Yang, "A Fast Forgery Detection Algorithm Based on Exponential-Fourier Moments for Video Region Duplication," *IEEE Trans. Multimed.*, vol. 20, no. 4, pp. 825–840, 2018, doi: 10.1109/TMM.2017.2760098.

[5] Y. Yuan, X. Yang, W. Wu, H. Li, Y. Liu, and K. Liu, "A fast single-image super-resolution method implemented with CUDA," *J. Real-Time Image Process.*, vol. 16, no. 1, pp. 81–97, 2019, doi: 10.1007/s11554-018-0774-z.

[6] L. D'Amiano, D. Cozzolino, G. Poggi, and L. Verdoliva, "A PatchMatch-based dense-field algorithm for video copy-move detection and localization," *arXiv*, vol. 29, no. 3, pp. 669–682, 2017.

[7] D. B. Tariang, R. S. Chakraborty, and R. Naskar, "Countering Antiforensic Attack on Median," *IEEE Signal Process. Lett.*, vol. 26, no. 8, pp. 1132–1136, 2019, doi: 10.1109/LSP.2019.2922498.

[8] 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.

[9] G. Nirmala and K. K. Thyagharajan, "A modern approach for image forgery detection using brich clustering based on normalised mean and standard deviation," *Proc. 2019 IEEE Int. Conf. Commun. Signal Process. ICCSP 2019*, pp. 441–444, 2019, doi: 10.1109/ICCSP.2019.8697951.

[10] P. Kakar, N. Sudha, and W. Ser, "Exposing digital image forgeries by detecting

discrepancies in motion blur," *IEEE Trans. Multimed.*, vol. 13, no. 3, pp. 443–452, 2011, doi: 10.1109/TMM.2011.2121056.

[11] V. Christlein, C. C. Riess, J. Jordan, C. C. Riess, and E. Angelopoulou, "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.

[12] T. J. De Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. D. R. 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.

[13] 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.

[14] A. Costanzo, I. Amerini, R. Caldelli, and M. Barni, "Forensic analysis of SIFT keypoint removal and injection," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 9, pp. 1450–1464, 2014, doi: 10.1109/TIFS.2014.2337654.

[15] Q. Liu, B. Zhou, A. H. Sung, and M. Qiao, "Exposing inpainting forgery in JPEG images under recompression attacks," *Proc. - 2016 15th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2016*, pp. 164–169, 2017, doi: 10.1109/ICMLA.2016.93.

[16] Y. Wo, K. Yang, G. Han, H. Chen, and W. Wu, "Copy-move forgery detection based on multiradius PCET," *IET Image Process.*, vol. 11, no. 2, pp. 99–108, 2017, doi: 10.1049/iet-ipr.2016.0229.

[17] Y. Zhang and V. L. L. Thing, "A semi-feature learning approach for tampered region localization across multi-format images," *Multimed. Tools Appl.*, vol. 77, no. 19, pp. 25027–25052, 2018, doi: 10.1007/s11042-018-5756-4.

[18] F. Yang, J. Li, W. Lu, and J. Weng, "Copy-move forgery detection based on hybrid features," *Eng. Appl. Artif. Intell.*, vol. 59, no. December 2016, pp. 73–83, 2017, doi: 10.1016/j.engappai.2016.12.022.

[19] K. M. Hosny, H. M. Hamza, and N. A. Lashin, "Copy-for-duplication forgery detection in colour images using QPCETMs and sub-image approach," *IET Image Process.*, vol. 13, no. 9, pp. 1437–1446, 2019, doi: 10.1049/iet-ipr.2018.5356.

[20] D. S. Vidyadharan and S. M. Thampi, *Evaluating color and texture features for forgery localization from illuminant maps*, vol. 77, no. 16. Multimedia Tools and Applications,

2018.

[21] 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.

[22] 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.

[23] H. U. Neenu and J. Cheriyan, “Image forgery detection based on illumination inconsistencies & intrinsic resampling properties,” *2014 Annu. Int. Conf. Emerg. Res. Areas Magn. Mach. Drives, AICERA/iCMMD 2014 - Proc.*, 2014, doi: 10.1109/AICERA.2014.6908192.

[24] E. Ardizzone, A. Bruno, and G. Mazzola, “Copy-Move Forgery Detection by Matching Triangles of Keypoints,” *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 10, pp. 2084–2094, 2015, doi: 10.1109/TIFS.2015.2445742.

[25] T. H. N. Le, K. Luu, and M. Savvides, “Fast and robust self-training beard/moustache detection and segmentation,” *Proc. 2015 Int. Conf. Biometrics, ICB 2015*, pp. 507–512, 2015, doi: 10.1109/ICB.2015.7139066.

[26] 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.

[27] T. Carvalho, F. A. Faria, H. Pedrini, R. S. Da Torres, and A. Rocha, “Illuminant-based transformed spaces for image forensics,” *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 4, pp. 720–733, 2016, doi: 10.1109/TIFS.2015.2506548.

[28] A. Ferreira *et al.*, “Behavior Knowledge Space-Based Fusion for Copy-Move Forgery Detection,” *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4729–4742, 2016, doi: 10.1109/TIP.2016.2593583.

[29] R. Cristin, J. P. Ananth, and V. C. Raj, “Illumination-based texture descriptor and fruitfly support vector neural network for image forgery detection in face images,” *IET Image Process.*, vol. 12, no. 8, pp. 1439–1449, 2018, doi: 10.1049/iet-ipr.2017.1120.

[30] G. Muzaffer and G. Ulutas, “A new deep learning-based method to detection of copy-move forgery in digital images,” *2019 Sci. Meet. Electr. Biomed. Eng. Comput. Sci. EBBT 2019,*

pp. 1–4, 2019, doi: 10.1109/EBBT.2019.8741657.

[31] Q. Liu, “An Improved Approach to Exposing JPEG Seam Carving under Recompression,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 1907–1918, 2019, doi: 10.1109/TCSVT.2018.2859633.

[32] J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” *Intas Polivet*, vol. 10, no. 2, pp. 227–228, 2009.

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

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

[35] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, “Analyzing features learned for Offline Signature Verification using Deep CNNs,” *Proc. - Int. Conf. Pattern Recognit.*, vol. 0, pp. 2989–2994, 2016, doi: 10.1109/ICPR.2016.7900092.

[36] H. Y. Choi *et al.*, “Detecting composite image manipulation based on deep neural networks,” *Int. Conf. Syst. Signals, Image Process.*, pp. 0–4, 2017, doi: 10.1109/IWSSIP.2017.7965621.

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

[38] A. Alotaibi and A. Mahmood, “Deep face liveness detection based on nonlinear diffusion using convolution neural network,” *Signal, Image Video Process.*, vol. 11, no. 4, pp. 713–720, 2017, doi: 10.1007/s11760-016-1014-2.

[39] O. C. Kurban, T. Yildirim, and A. Bilgic, “A multi-biometric recognition system based on deep features of face and gesture energy image,” *Proc. - 2017 IEEE Int. Conf. Innov. Intell. Syst. Appl. INISTA 2017*, pp. 361–364, 2017, doi: 10.1109/INISTA.2017.8001186.

[40] N. Huang, J. He, and N. Zhu, “A Novel Method for Detecting Image Forgery Based on Convolutional Neural Network,” *Proc. - 17th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. 12th IEEE Int. Conf. Big Data Sci. Eng. Trust. 2018*, pp. 1702–1705, 2018, doi: 10.1109/TrustCom/BigDataSE.2018.00255.

[41] Y. Zhang and V. L. L. Thing, “A multi-scale noise-resistant feature adaptation approach for image tampering localization over Facebook,” *2017 IEEE 2nd Int. Conf. Signal Image Process. ICSIP 2017*, vol. 2017-Janua, pp. 272–276, 2017, doi: 10.1109/SIPROCESS.2017.8124547.

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

[43] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury, “Hybrid LSTM and encoder-decoder architecture for detection of image forgeries,” *arXiv*, vol. 28, no. 7, pp. 3286–3300, 2019, doi: 10.1109/TIP.2019.2895466.

[44] Y. Chen, X. Kang, Y. Q. Shi, and Z. J. Wang, “A multi-purpose image forensic method using densely connected convolutional neural networks,” *J. Real-Time Image Process.*, vol. 16, no. 3, pp. 725–740, 2019, doi: 10.1007/s11554-019-00866-x.

[45] B. Yang, Z. Li, and T. Zhang, “A real-time image forensics scheme based on multi-domain learning,” *J. Real-Time Image Process.*, vol. 17, no. 1, pp. 29–40, 2020, doi: 10.1007/s11554-019-00893-8.

[46] M. A. Kutay and H. M. Ozaktas, “Optimal image restoration with the fractional Fourier transform,” *J. Opt. Soc. Am. A*, vol. 15, no. 4, p. 825, 1998, doi: 10.1364/josaa.15.000825.

[47] A. Koc, B. Bartan, E. Gundogdu, T. Çukur, and H. M. Ozaktas, “Sparse representation of two- and three-dimensional images with fractional Fourier, Hartley, linear canonical, and Haar wavelet transforms,” *Expert Syst. Appl.*, vol. 77, pp. 247–255, 2017, doi: 10.1016/j.eswa.2017.01.046.

[48] B. Meher, S. Agrawal, R. Panda, L. Dora, and A. Abraham, “A novel region-based multimodal image fusion technique using improved dictionary learning,” *Int. J. Imaging Syst. Technol.*, vol. 30, no. 3, pp. 558–576, 2020, doi: 10.1002/ima.22395.

[49] B. Meher, S. Agrawal, R. Panda, and A. Abraham, “A survey on region based image fusion methods,” *Inf. Fusion*, vol. 48, no. December 2017, pp. 119–132, 2019, doi: 10.1016/j.inffus.2018.07.010.

[50] S. K. Mahana and R. K. Aggarwal, “Image Steganography: Analysis & Evaluation of Secret Communication,” *SSRN Electron. J.*, pp. 1936–1943, 2019, doi: 10.2139/ssrn.3358094.

[51] Neha and R. K. Aggarwal, “Study of single image fog removal techniques in low visibility foggy images,” *Proceeding - IEEE Int. Conf. Comput. Commun. Autom. ICCCA 2017*, vol. 2017-Janua, pp. 1114–1118, 2017, doi: 10.1109/CCAA.2017.8229963.

[52] V. B. Kukkala and S. R. . Iyengar, “Identifying Influential Spreaders in a Social Network (While Preserving Privacy),” *Proc. Priv. Enhancing Technol.*, vol. 2020, no. 2, pp. 537–557, 2020, doi: 10.2478/popets-2020-0040.

[53] H. Cherifi, B. Goncalves, R. Menezes, and R. Sinatra, “Preface,” *Stud. Comput. Intell.*, vol. 644, pp. v–vi, 2016, doi: 10.1007/978-3-319-30569-1.

[54] P. S. Rajawat, D. K. Gupta, S. S. Rathore, and A. Singh, “Predictive Analysis of Medical Data using a Hybrid Machine Learning Technique,” *ICSCCC 2018 - 1st Int. Conf. Secur. Cyber Comput. Commun.*, pp. 228–233, 2018, doi: 10.1109/ICSCCC.2018.8703302.

[55] S. Kumari, D. K., and R. Mohan, “A Robust Method for Vehicle License Plate Recognition based on Harries Corner Algorithm and Artificial Neural Network,” *Int. J. Comput. Appl.*, vol. 148, no. 4, pp. 16–19, 2016, doi: 10.5120/ijca2016911075.

[56] G. Varshney, M. Misra, and P. Atrey, “Secure authentication scheme to thwart RT MITM, CR MITM and malicious browser extension based phishing attacks,” *J. Inf. Secur. Appl.*, vol. 42, pp. 1–17, 2018, doi: 10.1016/j.jisa.2018.07.001.

[57] G. Varshney, M. Misra, and P. K. Atrey, “A survey and classification of web phishing detection schemes,” *Secur. Commun. Networks*, vol. 9, no. 18, pp. 6266–6284, 2016, doi: 10.1002/sec.1674.

[58] N. B. A. Warif, A. W. A. Wahab, M. Y. I. Idris, R. Salleh, and F. Othman, “SIFT-Symmetry: A robust detection method for copy-move forgery with reflection attack,” *J. Vis. Commun. Image Represent.*, vol. 46, pp. 219–232, 2017, doi: 10.1016/j.jvcir.2017.04.004.

[59] L. Bondi, S. Lameri, D. Guera, P. Bestagini, E. J. Delp, and S. Tubaro, “Tampering Detection and Localization Through Clustering of Camera-Based CNN Features,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2017-July, pp. 1855–1864, 2017, doi: 10.1109/CVPRW.2017.232.

[60] O. Mayer and M. C. Stamm, “Accurate and Efficient Image Forgery Detection Using Lateral Chromatic Aberration,” *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 7, pp. 1762–1777, 2018, doi: 10.1109/TIFS.2018.2799421.

[61] M. D. Ansari, S. P. Ghrera, and V. Tyagi, “Pixel-Based Image Forgery Detection: A

Review," *IETE J. Educ.*, vol. 55, no. 1, pp. 40–46, 2014, doi: 10.1080/09747338.2014.921415.

[62] G. K. Birajdar and V. H. Mankar, "Digital image forgery detection using passive techniques: A survey," *Digit. Investig.*, vol. 10, no. 3, pp. 226–245, 2013, doi: 10.1016/j.diin.2013.04.007.

[63] 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.

[64] M. T. H. Majumder and A. B. M. Alim Al Islam, "A tale of a deep learning approach to image forgery detection," *Proc. 2018 5th Int. Conf. Networking, Syst. Secur. NSysS 2018*, pp. 1–9, 2019, doi: 10.1109/NSysS.2018.8631389.

[65] H. G. Kim, J. S. Park, D. G. Kim, and H. K. Lee, "Two-stream neural networks to detect manipulation of JPEG compressed images," *Electron. Lett.*, vol. 54, no. 6, pp. 354–355, 2018, doi: 10.1049/el.2017.4444.

[66] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," *Media Watermarking, Secur. Forensics 2015*, vol. 9409, p. 94090J, 2015, doi: 10.1117/12.2083479.

[67] J. Ouyang, Y. Liu, and M. Liao, "Copy-move forgery detection based on deep learning," *Proc. - 2017 10th Int. Congr. Image Signal Process. Biomed. Eng. Informatics, CISP-BMEI 2017*, vol. 2018-Janua, pp. 1–5, 2018, doi: 10.1109/CISP-BMEI.2017.8301940.

[68] B. Bayar and M. C. Stamm, "Design principles of convolutional neural networks for multimedia forensics," *IS T Int. Symp. Electron. Imaging Sci. Technol.*, pp. 77–86, 2017, doi: 10.2352/ISSN.2470-1173.2017.7.MWSF-328.

[69] J. A. Redi, W. Taktak, and J. L. 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.

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

[71] S. R. Sreela and S. M. Idicula, "Modified densely connected convolutional network for content generation in automatic image description generation system," *TENSYMP 2017* -

IEEE Int. Symp. Technol. Smart Cities, 2017, doi: 10.1109/TENCONSpring.2017.8070036.

[72] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Two-Stream Neural Networks for Tampered Face Detection," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2017-July, pp. 1831–1839, 2017, doi: 10.1109/CVPRW.2017.229.

[73] M. Bashar, K. Noda, N. Ohnishi, and K. Mori, "Exploring Duplicated Regions in Natural Images," *IEEE Trans. Image Process.*, no. c, pp. 1–40, 2019, doi: 10.1109/TIP.2010.2046599.

[74] M. Akbarpour Sekeh, M. A. Maarof, M. F. Rohani, and B. Mahdian, "Efficient image duplicated region detection model using sequential block clustering," *Digit. Investig.*, vol. 10, no. 1, pp. 73–84, 2013, doi: 10.1016/j.diin.2013.02.007.

[75] H. C. Nguyen and S. Katzenbeisser, "Detection of copy-move forgery in digital images using radon transformation and phase correlation," *Proc. 2012 8th Int. Conf. Intell. Inf. Hiding Multimed. Signal Process. IIH-MSP 2012*, vol. 1, pp. 134–137, 2012, doi: 10.1109/IIH-MSP.2012.38.

[76] J. van de Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2207–2214, 2007, doi: 10.1109/TIP.2007.901808.

[77] S. B.-S. and A. K. Nandi, "EXPOSING DUPLICATED REGIONS AFFECTED BY REFLECTION , Department of Electrical Eng . & Electronics . The University of Liverpool," pp. 1880–1883, 2011.

[78] J. Wang, G. Liu, H. Li, Y. Dai, and Z. Wang, "Detection of image region duplication forgery using model with circle block," *1st Int. Conf. Multimed. Inf. Netw. Secur. MINES 2009*, vol. 1, pp. 25–29, 2009, doi: 10.1109/MINES.2009.142.

[79] 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.

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

[81] M. Gong, T. Zhan, P. Zhang, and Q. Miao, "Superpixel-based difference representation learning for change detection in multispectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2658–2673, 2017, doi: 10.1109/TGRS.2017.2650198.

[82] C. Chen, S. McCloskey, and J. Yu, “Image splicing detection via camera response function analysis,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1876–1885, 2017, doi: 10.1109/CVPR.2017.203.

[83] H. Zuo, H. Fan, E. Blasch, and H. Ling, “Combining Convolutional and Recurrent Neural Networks for Human Skin Detection,” *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 289–293, 2017, doi: 10.1109/LSP.2017.2654803.

[84] D. Tralic, I. Zupancic, S. Grgic, and M. Grgic, “CoMoFoD - New database for copy-move forgery detection,” *Proc. Elmar - Int. Symp. Electron. Mar.*, no. September, pp. 49–54, 2013.

[85] J. Dong, W. Wang, and T. Tan, “CASIA image tampering detection evaluation database,” *2013 IEEE China Summit Int. Conf. Signal Inf. Process. ChinaSIP 2013 - Proc.*, pp. 422–426, 2013, doi: 10.1109/ChinaSIP.2013.6625374.

[86] 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.

[87] 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.

[88] Y. Yan, W. Ren, and X. Cao, “Recolored Image Detection via a Deep Discriminative Model,” *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 1, pp. 5–17, 2019, doi: 10.1109/TIFS.2018.2834155.

[89] A. Goel and P. Moulin, “Random ensemble of locally optimum detectors for detection OF Amish Goel Pierre", Moulin University of Illinois Urbana Champaign, *2018 IEEE Glob. Conf. Signal Inf. Process.*, pp. 1189–1193, 2018.

[90] Y.-F. H. and S.-F. Chang, “Detecting image splicing using geometry invariants and camera characteristics consistency", Department of Electrical Engineering Columbia University, *IEEE Int. Conf. Multimed. Expo*, pp. 549–552, 2006.

[91] Y. Bengio, G. Hinton, and Y. Lecun, “FATHERS OF THE DEEP LEARNING REVOLUTION RECEIVE ACM A.M. TURING AWARD Bengio, Hinton and LeCun Ushered in Major Breakthroughs in Artificial Intelligence,” pp. 1–5, 2019, [Online]. Available: https://awards.acm.org /binaries /content /assets /press-releases /2019/march

/turing-award-2018.pdf.

[92] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, “Recent advances in convolutional neural network acceleration,” Neurocomputing, vol. 323, pp. 37–51, 2019, doi: 10.1016/j.neucom.2018.09.038.