DEVELOPMENT OF MULTIPLE FORGERY DETECTION AND LOCALIZATION TECHNIQUES FOR DIGITAL VIDEO

THESIS SUBMITTED

FOR THE AWARD OF DEGREE OF

DOCTOR OF PHILOSOPHY

by

NITIN ARVIND SHELKE

REGISTRATION NO. 951703013

UNDER THE GUIDANCE OF

DR. SINGARA SINGH KASANA

ASSOCIATE PROFESSOR

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

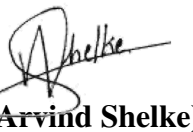
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY,

PATIALA, PUNJAB, INDIA-147004

AUGUST 2021

CERTIFICATE

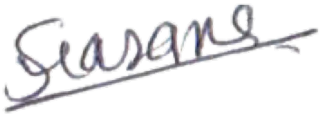
I hereby declared that the work being presented in this thesis titled as **"Development of Multiple Forgery Detection and Localization Techniques for Digital Video"**, in fulfillment of the requirement for the award of the degree of **DOCTOR OF PHILOSOPHY** submitted in the department of Computer Science and Engineering, Thapar Institute of Engineering and Technology Patiala Punjab India, is an authentic record of my own work carried out under the supervision of Dr. Singara Singh Kasana, Associate Professor, Computer Science and Engineering Department, and refer the other researcher works, which are properly referenced. This thesis content has not been submitted for the award of any other degree at this or any other university.



(Nitin Arvind Shelke)

Registration No. 951703013

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



(Dr. Singara Singh Kasana)

Associate Professor, CSED

TIET Patiala

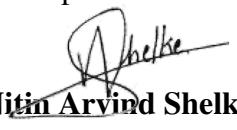
Supervisor

Acknowledgments

I dedicate this thesis to my PhD work at the TIET - Thapar Institute of Engineering and Technology Patiala, Punjab, India, to my beloved parents. The undertaking of this PhD has been a truly life-changing experience for me, and I would not have been able to complete it without the assistance and guidance of many people.

First and foremost, I wish to record a deep sense of gratitude to **Dr. Singara Singh Kasana**, my supervisor, for his constant support and guidance at all stages of my PhD. The work would not have been possible without his valuable inputs, advice, and support. His guidance and patience helped me in carrying out conduct by research effectively.

I would like to express my gratitude to my PhD committee, which includes **Dr. Maninder Singh**, Professor and Head, **Dr. Shailendra Shivani**, Assistant Professor, **Dr. Shrilekha Pandey**, Assistant Professor, TIET, Computer Science and Engineering Department, and **Dr. Vinay Kumar**, Associate Professor, Electronics & Communication Engineering Department, for keeping track of my progress and giving helpful suggestion to improve the work. I also extend my thanks to the entire staff (Teaching/Non-teaching) of the Computer Science and Engineering Department for their constant support and cooperation. I am thankful to my brother Rohan Shelke and my special friends Thakur, Sidharth, Sanjeev and Param for always being there for me and supporting me to keep going. Finally, my sincere gratitude goes to my parents **Arvind Shelke** and **Arti Shelke**. They dedicated their time, efforts, and resources to train me physically, mentally, and intellectually to become a person of importance and value to society which enabled me to complete this work. Aside from the individuals mentioned above, I would like to thank almighty, who has granted countless blessings, knowledge, and opportunity to pursue and complete this research so that I have been finally able to accomplish the thesis.



(Nitin Arvind Shelke)

Abstract

In recent years, video footages have played an important role in evidence shreds in various sectors, including forensic, surveillance, social networks, courtrooms, and news media. However, with the unlimited availability of advanced editing software, it has become quite easy for even inexperienced users to change/modify the video's actual content, resulting in an alarming rise in the number of forged videos. As a result, trust in videos is dwindling. There is a nudge to find a reliable solution to video forgery detection. Researchers across the world are working hard to come up with new solutions to the problem. There are two types of techniques for detecting video manipulation: active and passive. The active technique embeds footprint data in the form of watermarks in the digital video, either during recording or later, with the assistance of a specific application, allowing later verification of the originality of its content. However, the difficulty with the active method for video authentication is that it is only helpful in a restricted number of cases requiring special hardware. Passive techniques were developed in response to these constraints. It examines the integrity and authenticity of a video in the absence of an embedded watermark and relies on internal features. However, many of the passive approaches proposed so far can only identify one type of forgery at a time.

This doctoral research aims to advance the field of video forensics by developing passive techniques for detecting and localizing the multiple forgeries in the forged video. In this study, first, the prerequisite information for comprehending video forgery detection is impersonated. Following that, a detailed review of other contemporary passive forgery detection techniques for digital video is conducted. Based on this review, the limitations of the existing work are identified.

This study proposes techniques for passive video forgery detection. Each of these techniques concentrates on specific features or characteristics to identify the video forgeries. The first technique proposed in this work is based on correlation consistency between LBP-coded frames. This technique can detect and localizes the presence of forgeries in the videos and

tested on an MFVD-4 dataset. This technique provides an accuracy of 97.33% and is robust against the illumination change, GOP length, and background dependency.

This study offered a second passive technique to investigate multiple forgeries in a digital video using entropy-based texture features such as Two-Dimensional Distribution Entropy (DistrEn2D) and Bi-dimensional Multiscale Entropy (MSE2D), as well as the Outlier detection approach. This technique detects and locates the presence of forgeries when tested on an MFVD-2 dataset and provides an accuracy of 97.49% and 96.66% with DistrEn2D and MSE2D features, respectively. Moreover, the proposed technique is independent of GOP length and video background as well as robust to compression. The third passive technique detects multiple forgeries in a digital video using a VGG-16 neural network and Kernel Principal Component Analysis (KPCA). This technique achieves 97.24% accuracy when tested on an MFVD-3 dataset. Furthermore, this technique is independent of GOP length and video background and also robust to operations like noise addition and brightness, contrast, saturation, and hue modifications. Finally, a passive technique is proposed based on Principal Component Analysis (PCT) and Neighborhood Binary Angular Pattern (NBAP), followed by the GoogleNet model to detect and localize the multiple forgeries in the video. This technique has been tested and validated on the MFVD-1 dataset, and it is capable of identifying the forgeries in the video with 97.07% accuracy. Furthermore, the proposed technique is independent of GOP length and video background as well as robust to noise attacks.

Experimental results demonstrate the efficiency of the proposed techniques. Comparisons with existing passive forgery detection techniques show improved performance of the proposed techniques.

List of Publications

SCI Papers: Published

1. Nitin Arvind Shelke and Singara Singh Kasana, 2020,“A comprehensive survey on passive techniques for digital video forgery detection ”, *Multimedia Tools and Applications*, pp.6247-6310, Springer [I.F. 2.757].

2. Nitin Arvind Shelke and Singara Singh Kasana, 2021,“Multiple forgery detection and localization technique for digital video using PCT and NBAP ”, *Multimedia Tools and Applications*, pp.1-29, Springer [I.F. 2.757].

3. Nitin Arvind Shelke and Singara Singh Kasana, 2021,“Multiple forgeries identification in digital video based on correlation consistency between Entropy coded frames ”, *Multimedia System*, pp.1-14, Springer [I.F. 1.935].

SCI Papers: Under Review

1. Nitin Arvind Shelke and Singara Singh Kasana, 2021, “Multiple forgery detection in digital video using LBP features and outlier detection approach ”, *Journal of information security and applications*, Elsevier [I.F. 2.327].

2. Nitin Arvind Shelke and Singara Singh Kasana, 2021,“Multiple forgery detection in digital video with VGG-16-based deep neural network and KPCA ”, *Journal of Ambient Intelligence and Humanized Computing*, Springer [I.F. 7.104].

3. Nitin Arvind Shelke and Singara Singh Kasana, 2021,“Detection and localization of multiple forgeries in video using entropy features and 3-sigma rule ”, *Computer and Electrical Engineering*, Elsevier [I.F. 3.818].

Conference Paper: Presented

1. Nitin Arvind Shelke and Singara Singh Kasana, 2018 “Image Forgery Type and Identification: A Survey”, *International Conference on Electrical, Electronics, Computers, Communication, Mechanical and Computing, IEEE*.

Contents

Abstract **i**

List of Publications **iii**

List of Tables **ix**

List of Figures **xi**

List of Abbreviations **xiv**

List of Symbols **xvii**

1 Introduction **1**

1.1 Introduction 1

1.2 Video Forgeries 5

1.2.1 Intra-frame Video Forgery 5

1.2.2 Inter-frame Video Forgery 8

1.3 Video Forgery Detection Techniques 11

1.3.1 Active Techniques 12

1.3.2 Passive Techniques 12

1.4 Thesis Contribution 13

1.5 Methodology 13

1.6 Thesis Outline 15

2 Literature Survey **17**

2.1 Classification of Passive Forgery Detection Techniques 17

2.1.1 Compression Artifacts based Techniques 18

2.1.2 Noise Artifacts based Techniques 20
2.1.3 Motion Features based Techniques 23
2.1.4 Statistical Features based Techniques 27
2.1.5 Machine Learning based Techniques 34
2.2 Research Gaps 38
2.3 Research Objectives 40

3 Multiple Forgery Detection Technique for Digital video using LBP Features and Outlier Detection Approach 41
3.1 Background 42
3.2 Proposed Technique 43
3.2.1 Pre-processing and LBP Feature Extraction 43
3.2.2 Correlations Consistency 47
3.2.3 Outlier Detection 47
3.3 Experimental Analysis and Discussion 48
3.3.1 Dataset Description 48
3.3.2 Performance Evaluation 50
3.3.3 Experimental Results 52
3.3.4 Overall Comparison 52
3.4 Conclusion 56

4 Multiple Forgeries Detection Technique for Video based on Correlation Consistency between Entropy Coded Frames 57
4.1 Background 58
4.2 Proposed Technique 59
4.2.1 Pre-processing 60
4.2.2 Texture Feature Extraction 62
4.2.3 Correlation Consistency 64
4.2.4 Outlier Detection 65
4.3 Experimental Analysis and Discussion 67
4.3.1 Dataset Description 67
4.3.2 Visualization 69
4.3.3 Performance Evaluation 70

4.3.4 Comparison with Existing Techniques 74
4.4 Conclusion 77

5 Multiple Forgery Detection Technique for Digital Video using VGG-16 Neural Network and and KPCA 79

5.1 Background 80
5.2 Proposed Technique 81
5.2.1 Video Pre-processing 81
5.2.2 Feature Extraction using VGG-16 81
5.2.3 KPCA based Feature Selection 84
5.2.4 Investigation of Correlation Distribution 86
5.3 Experimental Analysis and Discussion 87
5.3.1 Experimental Setup 87
5.3.2 Dataset Description 90
5.3.3 Experimental Results 91
5.3.4 Performance Evaluation 91
5.4 Conclusion 99

6 Multiple Forgery Detection and Localization Technique using PCT and NBAP 101

6.1 Background 102
6.2 Proposed Technique 103
6.2.1 Pre-processing 103
6.2.2 PCT and NBAP Feature Extraction 103
6.2.3 GoogleNet Methodology 106
6.3 Experimental Result Ananlysis 109
6.3.1 Experimental Setup 109
6.3.2 Dataset Description 111
6.3.3 Experimental Evaluation 113
6.3.4 Overall Comparison 116
6.4 Overall Comparison of Proposed Techniques 118
6.5 Conclusion 120

7 Conclusion and Future Scope **121**

7.1 Conclusion 121

7.2 Future Work 123

References **125**

List of Tables

2.1 Pros and Cons of compression artifact-based forgery detection techniques . . . 19

2.2 Pros and Cons of noise artifacts-based forgery detection techniques 22

2.3 Pros and Cons of motion features-based forgery detection techniques 25

2.4 Pros and Cons of statistical features-based forgery detection techniques 31

2.5 Pros and Cons of machine learning-based forgery detection techniques 35

2.6 Details of forgery identification techniques 36

3.1 Description of forged videos (SF: Splicing Forgery, IF: Insertion Forgery, DF: Deletion Forgery) 50

3.2 Overall comparision (A: Forgeries identified, B: Multiple Forgeries Identification, C: GOP length, D: Background, E: Average Execution Time per video (in Sec), FIX: Fixed, VAR: Variable, SB: Static Background, MB: Moving Background, FS: Frame Splicing, FD: Frame Deletion, FDu: Frame Duplication) 56

4.1 Details of forged videos 68

4.2 Performance evaluation 70

4.3 Overall Comparison 77

5.2 Network Hyperparameter 89

5.1 VGG-16 Network details(ConvL: Convolution Layer, MPL: Max Polling Layer) 89

5.3 Forged videos details 90

5.4 Performance of proposed technique 95

5.5 Proposed *vs* existing techniques comparison 98

5.6 Comparision with existing techniques 98

6.1 Parameters of the GoogleNet model (ps: patch size, os: output size d: depth, r: reduce, incept: inception, params: parameters, ops: operations) 110
6.2 Network Hyperparameter 111
6.3 Sample forged videos details (RES: Resolution) 112
6.4 Performance of proposed technique 113
6.5 Comparative analysis 118
6.6 Overall comparison of Proposed techniques 119

List of Figures

1.1 Video surveillance system 2
1.2 Important aspects of video forensics 4
1.3 Classification of video forgeries 5
1.4 Copy-move forgery a) Flower in frame region is copied and pasted b) Keyboard removal from the video frame 6
1.5 Frame splicing forgery 7
1.6 Upscale crop a) Actual frame b) Upscale forged frame 7
1.7 Inter-frame forgeries a) Original sequence b) Frame insertion c) Frame deletion d) Frame duplication e) Frame Shuffling 8
1.8 Frame insertion forgery a) Actual frame sequence b) Resulted forged sequence after insertion forgery 9
1.9 Frame deletion manipulation a) Actual frame sequence b) Resulted manipulated sequence after frame removal (3^{rd} and 4^{th} frame is removed) 9
1.10 Frame duplication forgery a) Actual frame sequence b) Resulted forged sequence after frame duplication forgery 10
1.11 Frame mirroring type forgery a) Actual frame sequence b) Resulted forged sequence after frame mirroring forgery 11
1.12 Frame shuffling/replication forgery a) Actual frame sequence b) Resulted forged sequence after shuffling) 11

2.1 Categorization of passive techniques 18
2.2 Compression artifacts 18
2.3 Noise artifacts 21
2.4 Motion artifacts 23
2.5 Statistical features 28

3.1 Proposed technique flowchart . . . 44
3.2 LBP descriptor . . . 45
3.3 Center pixel calculation in LBP . . . 46
3.4 LBP encoded frame . . . 46
3.5 Performance of the proposed technique . . . 51
3.6 Evaluation against the GOP length . . . 51
3.7 Evaluation against the video background . . . 52
3.8 Experimental results on sample test videos . . . 54
3.9 Proposed technique *vs.* existing techniques in terms of Accuracy . . . 55

4.1 Flowchart of multiple video forgery detection and localization technique . . . 61
4.2 Snapshot of forged videos . . . 68
4.3 Visualization: DistrEn2D a) Successful b) Failure . . . 71
4.4 Visualization: MSE2D a) Successful b) Failure . . . 72
4.5 Performance against the varied GOP length videos . . . 73
4.6 Performance against the static/moving background videos . . . 74
4.7 Performance against the compression . . . 75
4.8 Comparison with the existing techniques . . . 76

5.1 Architecture: Proposed technique . . . 82
5.2 VGG-16 layers (a) The original VGG16 network architecture (b) Proposed fine-tuned VGG-16 network architecture (FC layers are Removed from VGG-16) . . . 85
5.3 Experimental results on sample forged videos . . . 94
5.4 Evaluation of the proposed technique against the different GOP length videos . . . 95
5.5 Evaluation of the proposed technique against the static and moving background video . . . 96
5.6 Evaluation of the proposed technique against the Post-processing operations . . . 97

6.1 Proposed technique flowchart . . . 104
6.2 Architecture of GoogleNet model . . . 107
6.3 Inception module (1×1 convolution) . . . 108
6.4 Snapshot of forged videos . . . 112
6.5 Detection and localization result . . . 114
6.6 Evaluation on videos with different GOP lengths . . . 115

6.7 Evaluation on videos with different backgrounds 116
6.8 Evaluation against the noise robustness 117

List of Abbreviations

ACC	Accuracy
AFCT	Adaptive Parameter based Fast Compression Tracking
AVIBE	Adaptive Parameter-based Visual Background
AVI	Audio Video Interleave
AWOB	Adjustable Width Object Boundary
BBVD	Block-wise Brightness Variance Descriptor
BoW	Bag of Words
CA	Cellular Automata
CCCoGV	Correlation Coefficients of Gray Values
CFA	Color Filter Array Artifacts
CNN	Convolutional Neural Network
DCT	Discrete Cosine Transform
DistrEn2D	Two-Dimensional Distribution Entropy
EFM	Exponential-Fourier Moments
ESD	Extreme Studentized Deviate
FLANN	Fast Library for Approximate Nearest Neighbors
GLCM	Gray Level Co-occurrence Matrices
GOP	Group of Pictures
GSSIM	Gradient Structure Similarity
H-DC	Duplicate Cluster Detection
HMRF	Huber Markov Random Field

HOG	Histogram of Oriented Gradients
HSV	Hue-Saturation-Value
JPEG	Joint Photographic Experts Group
KPCA	Kernel Principal Component Analysis
KNN	K-nearest neighbour
LBP	Local Binary Patterns
LDA	Linear Discriminant Analysis
LSTM	Long Short-term Memory
MACE	Minimum Average Correlation Energy
MAD	Median Absolute Deviation
MCEA	Motion Compensated Edge Artifacts
MEI	Motion Energy Image
MFVD	Multi-Forged Video Dataset
MI-SIFT	Mirror-invariant and Inversion-invariant SIFT
MLP	Multilayer Perceptron
MSE2D	Bi-dimensional Multiscale Entropy
MPEG	Moving Picture Experts Group
NBAP	Neighborhood Binary Angular Pattern
NIMB's	Number of Intra Macroblocks
NTF	Nonnegative Tensor Factorization
NLF	Noise Level Function
PCC	Pearson Correlation Coefficient
PCT	Polar Cosine Transform
PHT	Polar Harmonic Transform
PR	Prediction Residual
PSNR	Peak Signal to Noise Ratio

QCCoLBPs	Quotients of Consecutive Correlation Coefficients of LBPs
QCRI	Quantitative Correlation Rich Region
QoMSSIM	Quotient of Mean Structural Similarity
RMSE	Root Mean Square Error
SCREs	Spatially Constrained Residual Errors
SIFT	Scale-Invariant Features Transform
SPNC	Sensor Pattern Noise Correlation
SULFA	Surrey University Library for Forensic Analysis
SURF	Speeded-up Robust Features
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TIRI	Temporally Informative Representative Images
VGG	Visual Geometry Group
VPF	Variation of Prediction Footprint
VTD	Video Tampering Dataset
VTL	Video Trace Library
ZOCM	Zernike Opponent Chromaticity Moments

List of Symbols

I	Image
V	Video
F	Frame
h	Frame width
w	Frame height
(x, y)	Image Pixel
(X_c, Y_c)	Centre pixel value
μ	Mean
ρ	Correlation Coefficients
$E(X, Y)$	Cross-correlation between X and Y
σ_X	Variances of X
σ_Y	Variances of Y
p_i	Probability of Pixels Occurrence
γ, θ	Polar Coordinates of Image
M_{nl}	PCT Coefficients
$\sum$	Summation
$\bar{x}$	Mean of x
$\bar{y}$	Mean of y
C^F	Covariance Matrix
(m_h, m_w)	Embedding Vector
N_m	Total number of square windows within Image I

D	Distance Matrix
λ	Eigenvalues
v	Eigenvectors
K	Kernel Matrix
pc_k	Principal Components
$\phi(f)$	Kernel function
δ	Euclidean Distance

CHAPTER 1

Introduction

This Chapter provides an overview of forgery detection in video. First, the basic terminologies related to forgery detection are explained. The objectives and scope are then described. The thesis outline is presented at the end.

1.1 Introduction

A digital video a sequence of pictures/images, also known as video frames, arranged in a specific order. The creation of movies, news reporting, surveillance cameras, and admissible evidence are all examples of digital video applications. A digital video may be easily distributed using a low-cost internet service for various purposes, including video conferencing, media houses, surveillance systems, traffic lights monitoring, and hospital surveillance. On the other hand, the movie industry and media houses, uses robust and powerful editing software to create videos with high quality and appealing graphics. Moreover, smartphones gadgets and cam-

corders provide a smooth and easy way to record and store a video.

The usage of digital videos has noticeably increased in the lives of people from all walks of life. Videos are utilized in various fields and play a crucial role in conveying authenticity, particularly in the surveillance system, court evidence, medical field, cinema industry, product advertising, social media, and journalism. In general, common users trust any video and do not even question whether the information conveyed by it is true or not. This blind trust causes a slew of issues in our social, personal, and professional lives. The following are some of the influential societal areas that video manipulations can impact.

- **Video Surveillance:** In video surveillance, videos are collected from various locations such as office rooms, traffic signals/roads, hospitals, banks, industries, college/university premises, apartments, railway station, airport, and so on, in order to keep track of daily activities. Sometimes these collected video content may be manipulated/forged for the purpose of certain crimes by deleting, inserting, or editing some frames. As a result, these trusted mediums fail to reveal the truth. The video surveillance is shown in Figure 1.1.

Figure 1.1: Video surveillance system

- **Social Media and Journalism:** Since the advent of social networking websites like Facebook, WhatsApp, Twitter, YouTube, and Instagram, the phenomenon of the transfer of digital videos over the Internet has gained significant popularity among technology experts and amateurs alike. While social media has modernized producing and disseminating news, it has also become a breeding ground for false and fake news. Forged/manipulated digital videos disseminated *via* online social media have the potential to mislead their

spectators, imposing cognitive stress, abusing prior beliefs, or affecting persons' actions and decisions. Journalists seek out important and interesting local or global events that are taking place at any given time in order to report on them. It has been discovered that information/video gathered from social media is sometimes manifested in the news. Any forged video from social media broadcasted on the news has the potential to have a significant negative impact on society.

- **Court:** Video footage is considered as one of the most important pieces of evidence against a defendant in the court. It assists the judge in making the correct decision against the crimes committed. Inability to distinguish between authentic and forged video may lead to an incorrect decision, eroding public trust in law and order.

- **Politics Influence:** In recent years, some politicians have become deeply involved in undermining the face of opposition politician by spreading forged video *via* online social media. Videos of some political leaders' faces being replaced in videos of other incidents to infamy them on social media.

- **Religious Faith:** Many videos with religious themes can be found on social media. Often, a forged video steers people's minds toward religious faith, regardless of their basic senses.

With the availability of video editing devices and open source applications, digital video authenticity and integrity has become the major concern in the last decade. Besides, the incidents of manipulations of videos for criminal intent have also been increased. So in order to safeguard the video's content integrity and authenticity, it led to the birth of forensic science, especially Digital video forensic. Video forensics is a branch of forensic science that deals with the scientific examination, assessment, evaluation, and inspection of digital video to measure the trustworthiness, authenticity, and genuineness of video content (Ho & Li, 2015). The field of digital forensic is divided into six main categories (Pandey et al., 2016). Figure 1.2 shows the important aspects of video forensics.

- Source Classification: Its goal is to differentiate or classify videos based on their source, such as Nikon *vs* Canon.

- Device Identification: Its goal is to demonstrate that a given video is acquired from some specific digital device.
- Device Linking: It groups objects together based on their common source (To determine which common camera is used to capture videos).
- Processing History Recovery: Its primary goal is to restore the processing chain for digital videos.
- Forgery Detection: Its main intention is to detect forgeries in digital video and determine whether the video is authentic.
- Steganalysis: Its primary concern is to verify whether there are any existing hidden data present in the digital video, and its second concern is to discover that hidden message Q. Liu et al., 2008; C.-N. Yang et al., 2012.

Figure 1.2: Important aspects of video forensics

One of the important aspects of video forensics that we studied and explored in this work is the video forgery detection mechanism. Digital content such as images and videos have been

profoundly manipulated in recent year (Birajdar & Mankar, 2013; Ho & Li, 2015). Due to alterations in a digital video, it is getting more difficult to determine the integrity and authenticity of the digital videos with the naked eyes due to changes such as region manipulation, insertion/deletion of the frame, etc. Therefore, a technique for detecting forgeries present in digital videos is required to confirm the integrity/authenticity of input videos. A video forgery is a modification done in a video for falsification (Johnston & Elyan, 2019). Video forgery detection is a technique to expose the forgeries present in the video in a systematic manner.

1.2 Video Forgeries

This section defines the fundamental terms required to comprehend video forgery detection.

The forgeries present in the videos are categorized as 1) Intra-frame 2) Inter-frame. The classifications of forgeries in the video are depicted in Figure 1.3.

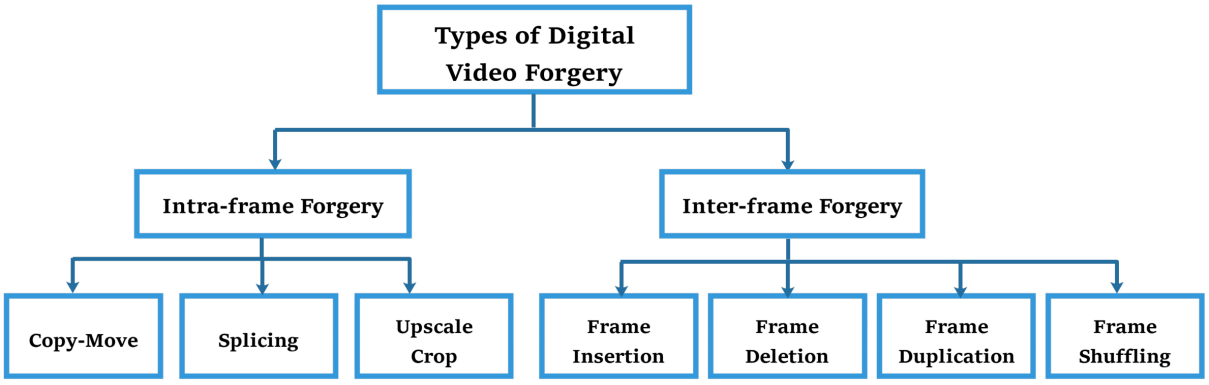


Figure 1.3: Classification of video forgeries

1.2.1 Intra-frame Video Forgery

In this type of forgeries, the specific video content of frames is modified. It is sometimes referred to as spatial-based video manipulations. Intra-frame forgeries can be performed at both the pixel and block levels of video frames. Some examples of intra-frame forgery are listed as.

a) Copy-move forgery: Copy-move is an example of video intra-frame forgery. Copy-move forgery is accomplished by copying an object from a frame in the video and pasting it to another location. A copy-move forgery allows an attacker to modify the video scene by removing or inserting the entire or portion of an object in a frame.

The methods used in copy-move forgeries can also be employed to conceal an area in the frame. (C. Chen et al., 2017; Fan et al., 2016). The case of copy-move manipulation is shown in Figure 1.4 wherein (a) A flower from the frame region is copied and inserted at some different location of the same video frame. The keyboard shown by a yellow mark is removed from the actual video frame, in Figure 1.4 (b),

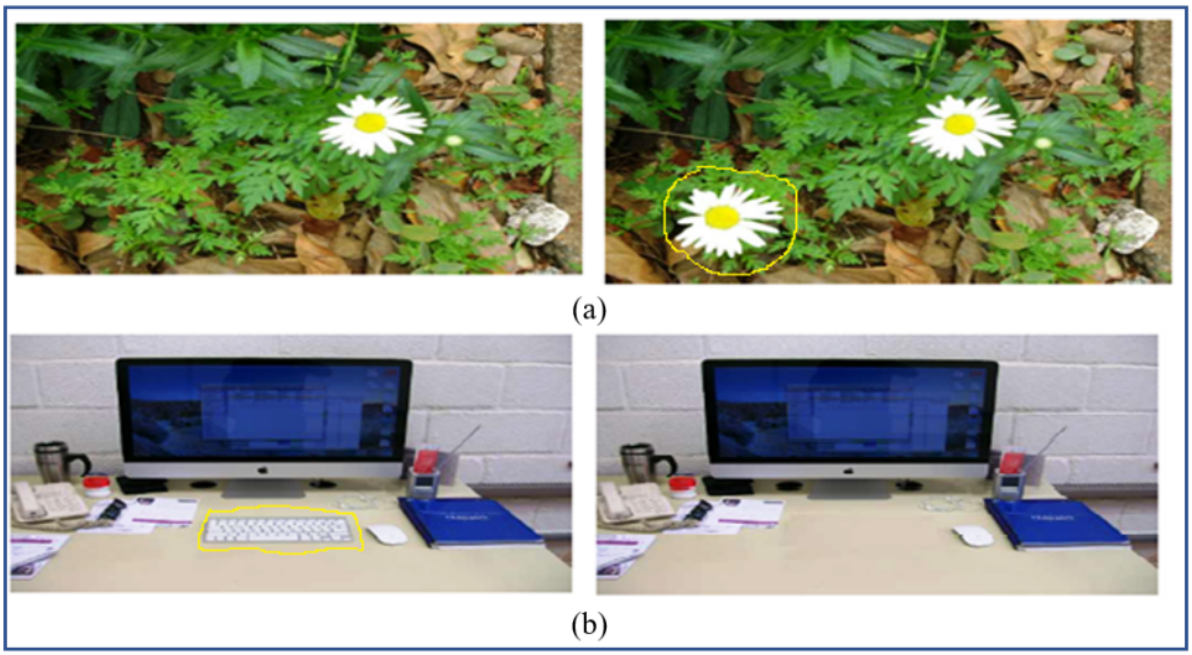


Figure 1.4: Copy-move forgery a) Flower in frame region is copied and pasted b) Keyboard removal from the video frame

b) Frame splicing:

Splicing is another example of intra-frame forgery. It is generated by combining parts of two or more video frames (Jaiswal & Srivastava, 2020 Upadhyay & Chhabra, 2021). Figure 1.5 shows video frame splicing forgery in which two frames are combined to create a new frame.

c) Upscale crop:

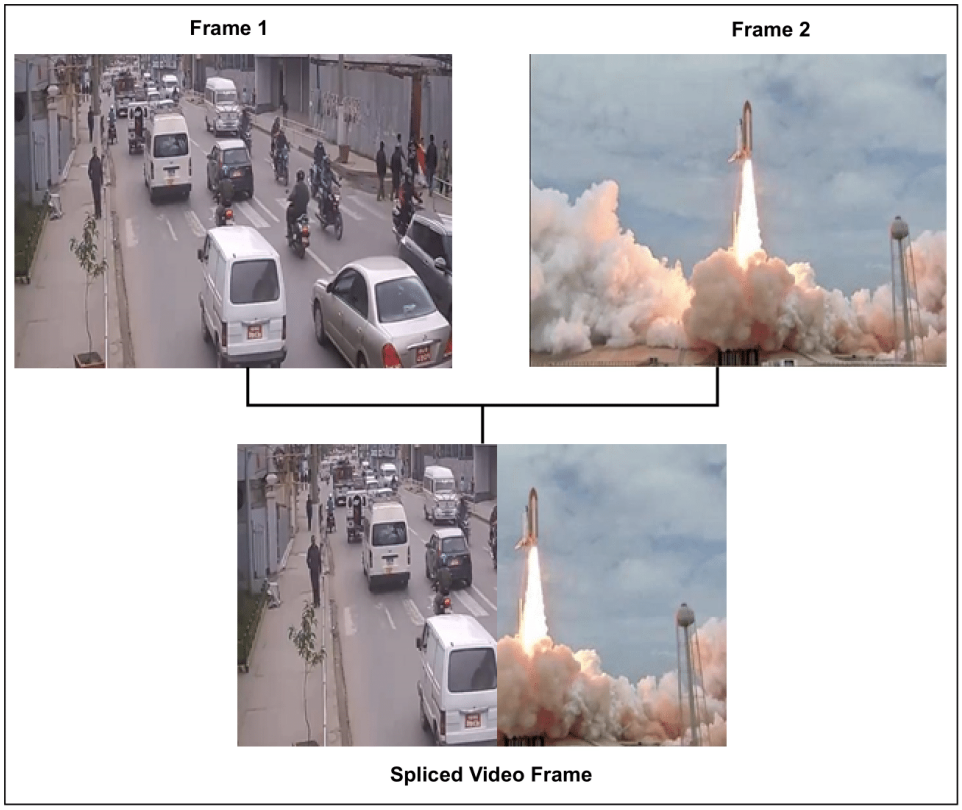


Figure 1.5: Frame splicing forgery

In upscale crop forgery, an outer part of the frame is trimmed off to delete some area or region R. D. Singh & Aggarwal, 2017b. The example of upscale crop forgery is shown in Figure 1.6 with (a) Actual frame, (b) Upscale forged frame (the lady is cropped off).

Figure 1.6: Upscale crop a) Actual frame b) Upscale forged frame

1.2.2 Inter-frame Video Forgery

In inter-frame forgery, the attackers disrupt the actual sequence of frames. It is also referred to as temporary manipulation. Frame insertion/addition, frame duplication, frame removal/deletion, and frame shuffling/replication are the types of video inter-frame forgery as depicted in Figure 1.7.

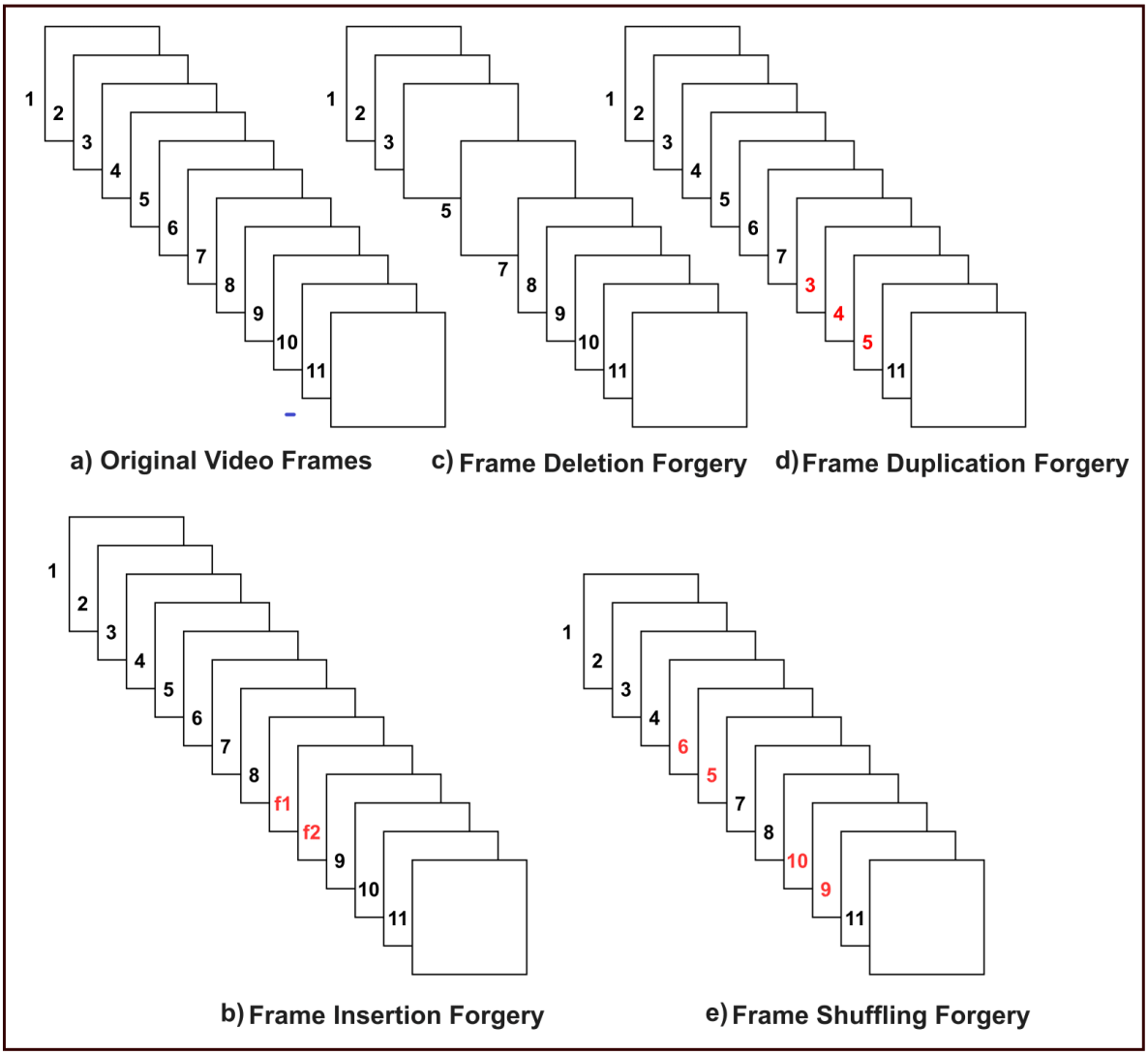


Figure 1.7: Inter-frame forgeries a) Original sequence b) Frame insertion c) Frame deletion d) Frame duplication e) Frame Shuffling

a) Frame insertion: Frame insertion is an example of inter-frame manipulation. Frames

taken from the same or different videos are purposely inserted at arbitrary positions. Figure 1.8 depicts the video frame insertion manipulation with (a) represents actual frame sequence (b) represents frames after the insertion forgery, which involves the insertion of I1 and I2 frames at the 2^{nd} and 3^{rd} positions of the original sequence.

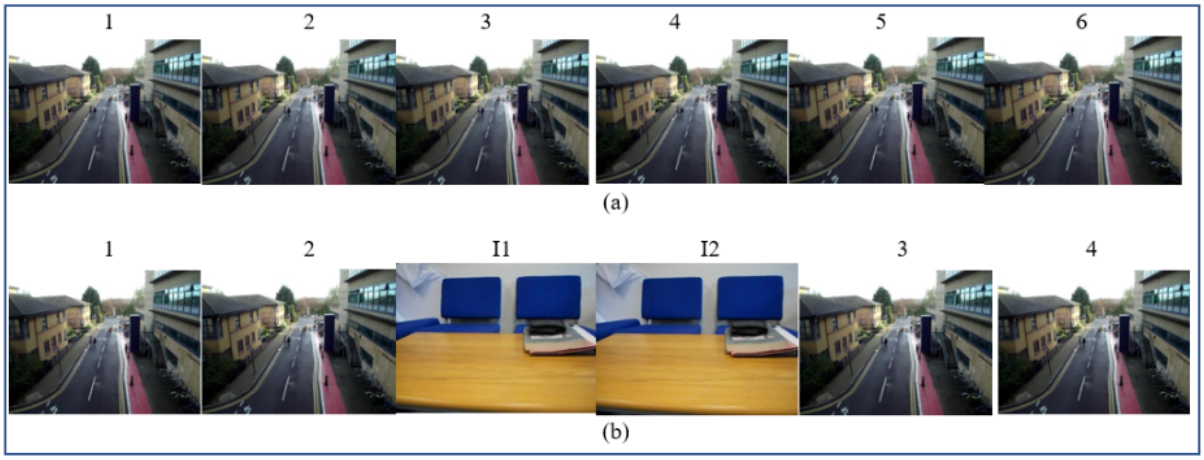


Figure 1.8: Frame insertion forgery a) Actual frame sequence b) Resulted forged sequence after insertion forgery

b) Frame deletion: Frame deletion is one type of video inter-frame forgery, in which certain frames are removed from the original sequence to perform the manipulation. Frame deletion forgery is depicted in Figure 1.9 with (a) representing the actual frame sequence, (b) representing the frame sequence following frame deletion manipulation.

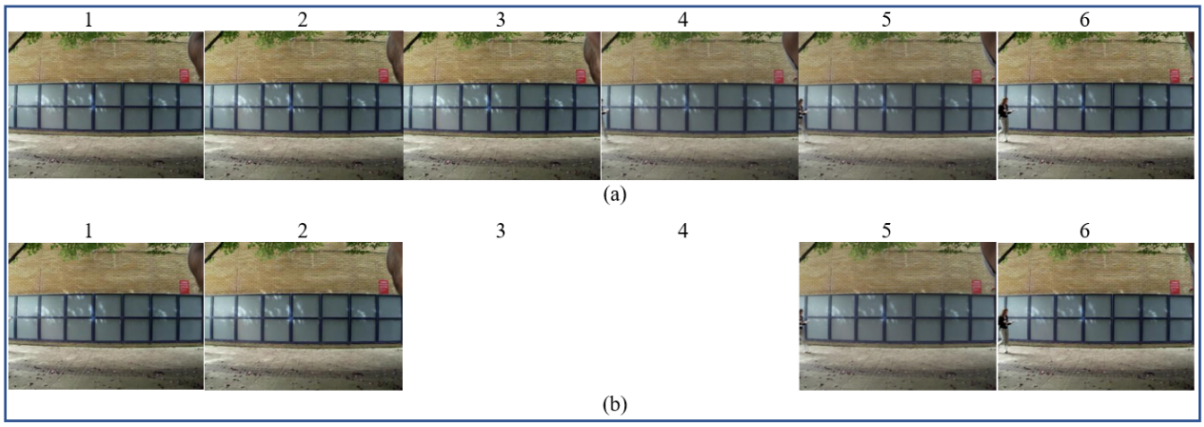


Figure 1.9: Frame deletion manipulation a) Actual frame sequence b) Resulted manipulated sequence after frame removal (3^{rd} and 4^{th} frame is removed)

c) Frame duplication: Frame duplication is an inter-frame manipulation technique in which a few frames in a video sequence are duplicated. Figure 1.10 depicts the video frame duplication manipulation. The actual frame sequence is shown in Figure 1.10 (a), and video frame duplication forgery is shown in Figure 1.10 (b) (Frame 6 is duplicated at 3^{rd} location).

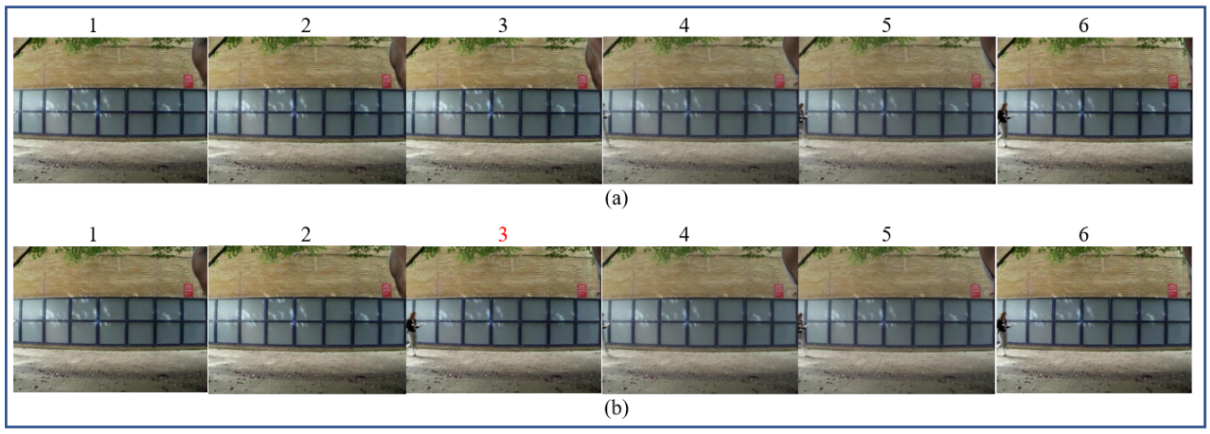


Figure 1.10: Frame duplication forgery a) Actual frame sequence b) Resulted forged sequence after frame duplication forgery

Frame-mirroring is a modified version of frame duplication forgery. It is generated by pasting a mirrored copy of the frame at a random location in the same video. Figure 1.11 (a) represents the actual frame sequence and Figure 1.11. (b) represents mirrored video frames (M2 denotes the mirrored frame, pasted at 2^{nd} position, and M6 denotes the mirrored frame, pasted at position 5)

d) Frame shuffling/replication:

When some frames from the input video are rearranged, it's called a shuffled video sequence. The frame shuffling forgery shuffles the video frames sequence, giving the original video a new meaning.

The frame shuffling forgery is presented in Figure 1.12. In Figure 1.12 (a) represent the actual frame sequence and (b) represent frame sequence after a shuffling type forgery in which the fourth video frame is replaced with the second.

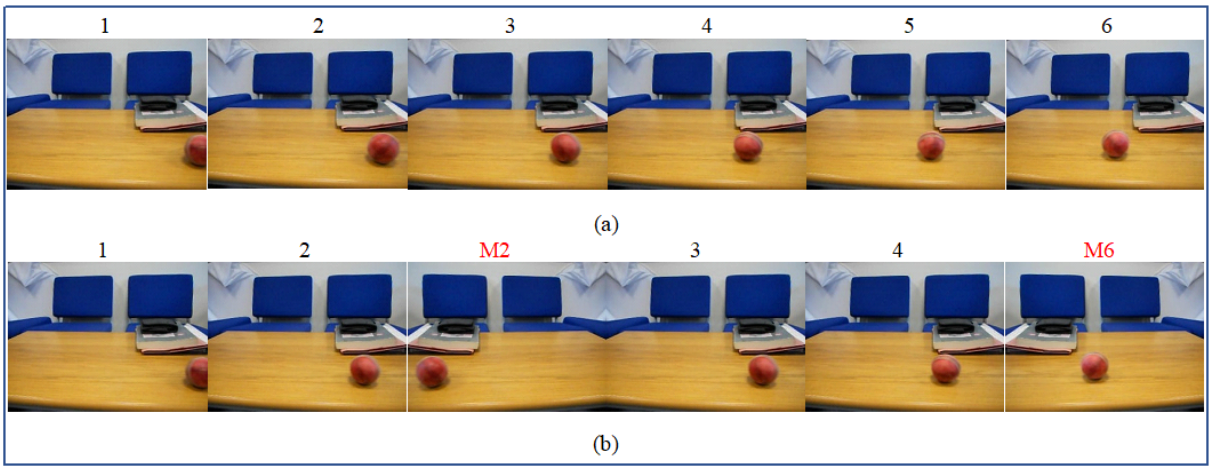


Figure 1.11: Frame mirroring type forgery a) Actual frame sequence b) Resulted forged sequence after frame mirroring forgery

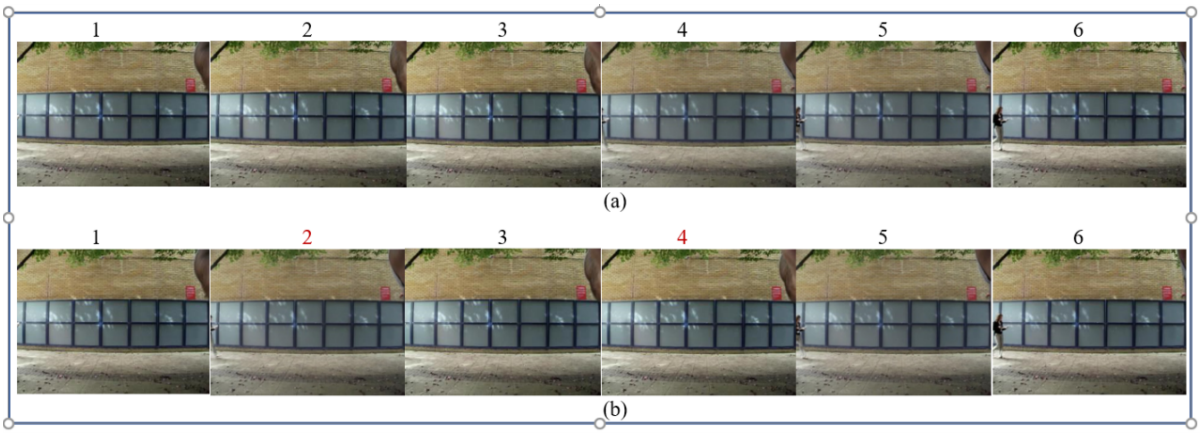


Figure 1.12: Frame shuffling/replication forgery a) Actual frame sequence b) Resulted forged sequence after shuffling)

1.3 Video Forgery Detection Techniques

The techniques for identifying the forgery in the video is categorized into two main part 1) Active techniques and 2) Passive techniques.

1.3.1 Active Techniques

The active approach insert footprint data in the form of watermarks in the digital video, either during recording or later, by taking the help of a specific application to allow verification of the originality, authenticity, and integrity of its content later (J. Zhang et al., 2007). The advantage of the active approach is that footprint data like a watermark/signature in a video makes forgery identification simple. However, videos downloaded from the Internet lack embedded information, so it is very hard to expose manipulation in such a situation. Another limitation of the active approach is that it necessitates additional hardware to embed information in each frame while capturing video. Moreover, with the use of the active approach, the pre-embedding watermarks or signatures degrade the video quality.

1.3.2 Passive Techniques

To address the limitations of the active approach, forensic researchers created a more robust approach, *i.e.,* passive approach for authenticating video content. It examines a video's integrity and authenticity in the absence of embedded evidence such as a signature or watermark. Working on passive techniques becomes difficult for the researcher if they are unaware of the pre-embedded information/data within the video. Therefore, video forgery detection gains a new dimension with passive techniques for digital video authentication and integrity. The assumption behind passive techniques is that videos consist of hidden parameters injected into them through the video processing steps. This technique is sometimes called the blind method because it depends upon the intrinsic artifacts/features of the video frames rather than the embedded evidence. The passive techniques look at the artifacts left behind after the video manipulations to discover the actual videos from the manipulated ones.

1.4 Thesis Contribution

This research study focused on video forensics and proposed passive forgery detection techniques to detect and localize multiple forgeries. The contributions to the field of digital video forensics are listed below:

- The conducted research provides the flaws in existing passive techniques for detecting the forgeries.
- Manipulations in the video are likely to disrupt the expected consistency of LBP extracted features which is to be considered an artifact in the first proposed technique to detect the forgeries. The use of this feature has improved the robustness against the GOP size and background dependency.
- A passive technique to detect the forgeries in the individual video is proposed using entropy based texture feature. Entropy-based texture feature is utilized in our second technique, which increases the robustness against compression quality factor.
- A passive technique to detect the video forgeries is proposed using the VGG-16 neural model. The VGG-16 is used in the third technique to perform the automatic feature extraction. The use of KPCA has reduced the dimension of the extracted features from VGG-16. The robustness against the brightness, contrast, and saturation/hue modifications has been improved by using this feature.
- To deal with the multiple forgeries present in the video, PCT and NBAP features are considered in the fourth technique. The robustness against rotation and noise attacks has been improved by using the PCT.

1.5 Methodology

- After reviewing the existing forgery detection techniques for the video based on passive techniques, four techniques are proposed to address the mentioned gaps in the existing

techniques. Certain features are considered in all the proposed techniques, which are then analyzed to check for any distortion if subjected to any forgeries.

- As mentioned in the survey and research gaps, the existing forgery detection techniques are only able to detect the single type of forgery present in the individual video. Thus, proposed techniques in this work are able to handle the multiple forgeries from the individual video.

- The robustness of the proposed forgery detection techniques is tested and proved against various post-processing attacks like noise addition, compression, brightness, contrast, saturation/hue modifications, and illuminations.

- The dependency of the proposed forgery detection technique is tested and proved against the change in GOP length and video background.

- As discussed in the survey and research gaps, no forged video dataset is available for testing the performance of the techniques against the multiple forgeries in the video. Thus, A dataset with multiple video forgeries is created with the use of video taken from the datasets such as SULFA (Qadir et al., 2012), (REWIND, 2013), (VTL, 2018), (VTD, 2017) in order to test the performance of the proposed techniques.

- The parameters such as accuracy, precision, recall, and F1-score are considered to evaluate the performance of the proposed techniques for the forgery identification.

$$Accuracy(ACC) = \frac{TP+TN}{TP+FN+TN+FP} \tag{1.1}$$

$$Precision(PR) = \frac{TP}{TP+FP} \tag{1.2}$$

$$Recall(RR) = \frac{TP}{TP+FN} \tag{1.3}$$

$$F1Score = 2 \times \frac{PR \times RR}{PR+RR} \tag{1.4}$$

Here, TP: True Positive, TN; True Negative, FP: False Positive, and FN-False Negative. ACC: Accuracy, PR: Precision, RR: Recall.

- **Accuracy:** It is calculated by dividing the total number of correctly detected samples by the total number of samples in the dataset.
- **Precision:** It is determined by dividing the correctly predicted positive samples by by all the observation in predicted class.
- **Recall:** It is determined by dividing the true positive samples by all the observations in an actual class.
- **F1-score:** It is termed as a harmonic mean of recall and precision.

1.6 Thesis Outline

✓ **Chapter 1: Introduction**

Chapter-1 contributes an overview of video forgery detection and discusses the foundation of important concepts that need to be known to understand the thesis work better. Different types of video forgeries followed by performance evaluation metrics are discussed in detail. This Chapter further introduces the research objectives.

✓ **Chapter 2: Literature Survey**

Chapter-2 explores the overall work done by various researchers across the globe related to video forgery detection techniques under the passive scheme. Categorization of passive video forgery identification techniques is included in this Chapter. This Chapter provides an in-depth investigation of compression artifact-based forgery detection techniques, Noise artifacts-based techniques, Motion features-based techniques, Statistical features-based techniques, and techniques based on machine learning. Finally, based on the study, some of the research gaps associates with existing techniques are also discussed in this Chapter.

✓ **Chapter 3: Multiple Forgery Detection Technique for Digital video using LBP Features and Outlier Detection Approach**

This Chapter presents the forgery detection technique using the LBP feature and the Outlier detection approach. In this Chapter, Section 3.1 discuss the background. Section 3.2 describes the overall flow of the proposed technique in this Chapter. Section 3.3 presents the experimental analysis and discussion, and the conclusion of the Chapter is discussed in Section 3.4

✓ **Chapter 4: Multiple Forgeries Detection Technique for Video-based on Correlation Consistency between Entropy Coded Frames**

This Chapter presents the forgery detection technique using entropy-based texture features and the Outlier detection approach. In this Chapter, Section 4.1 discuss the background. The proposed technique is explained in Section 4.2. Experimental analysis and discussion are presented in Section 4.3, and the conclusion of the Chapter is discussed in Section 4.4.

✓ **Chapter 5: Multiple Forgery Detection Technique for Digital Video using VGG-16 Neural Network and and KPCA**

In this Chapter, the forgery detection technique for digital video using a VGG-16 neural network and KPCAis presented. In this Chapter, Section 5.1 discuss the background. The proposed technique is explained in Section 5.2. Experimental analysis and discussion are presented in Section 5.3, and the conclusion of the Chapter is discussed in Section 5.4

✓ **Chapter 6: Multiple Forgery Detection and Localization Technique using PCT and NBAP**

This Chapter discusses a technique for video forgery identification using the PCT and NBAP followed by the GoogleNet model. In this Chapter, Section 6.1 discuss the background. Section 6.2 explains the overall flow of the proposed technique; Section 6.3 presents the experimental result analysis, Section 6.4 gives the overall comparison of all the proposed techniques, and the conclusion of the Chapter is discussed in Section 6.5.

✓ **Chapter 7: Conclusion and Future Scope**

In this Chapter, the conclusion of this thesis is presented. All the proposed multiple forgeries detection and localization techniques are concluded in this Chapter. A direction for the future scope of this research work is also discussed.

CHAPTER 2

Literature Survey

The study of existing passive techniques for forgery detection in the video is the focus of this Chapter, which is followed by research gaps. Based on these gaps, the objectives of the proposed work have been identified.

2.1 Classification of Passive Forgery Detection Techniques

A number of noteworthy research work has been done on passive forgery detection in videos in the last few years (Aghamaleki & Behrad, 2016; Ho & Li, 2015; Kobayashi et al., 2009; W. Wang & Farid, 2009; Zampoglou et al., 2019). The categorization based on features used in passive video forgery identification techniques is depicted in Figure 2.1

Contents of the work presented in this Chapter have been published in *Multimedia Tools and Applications*, pp.6247-6310, 2020, Springer. (SCI Indexed)

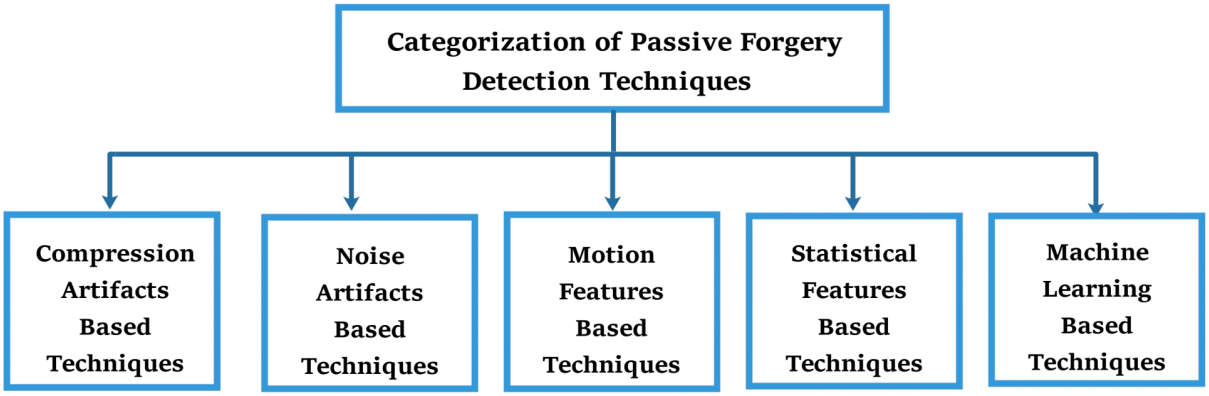


Figure 2.1: Categorization of passive techniques

2.1.1 Compression Artifacts based Techniques

To save space and time, videos are often compressed using a variety of coding standards such as MPEG-4, H-264, and others. Compression feature-based techniques utilize coding evidence or artifacts obtained through the video compression process to identify manipulations . Compression artifacts look at video attributes such as compression traits, periodic features, changes in Discrete Cosine Transform (DCT) parameters, quantization parameters, and Group of Pictures (GOP) characteristics. The compression artifacts employed in forgery detection are depicted in Figure 2.2.

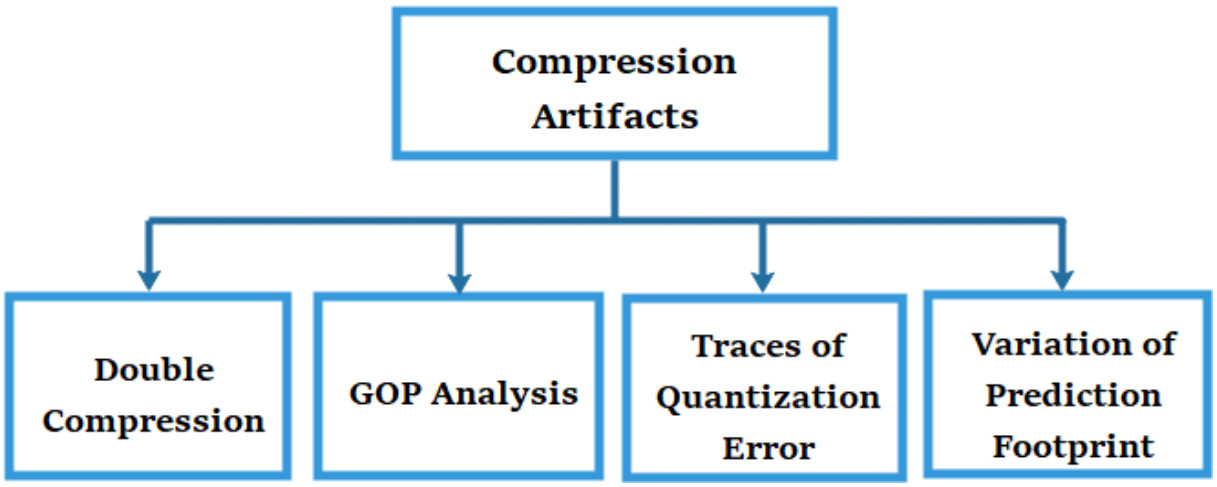


Figure 2.2: Compression artifacts

(W. Wang & Farid, 2006) observed the manipulation by analyzing static/temporal artifacts added in the video when subjected to double compression with MPEG. They made changes and developed a method for determining whether a video is doubly compressed. Using Histogram of Oriented Gradients (HOG) features and video compression attributes, (Subramanyam & Emmanuel, 2012) developed a passive approach to expose copy-move forgery. Furthermore, they designed an approach to recognize the copy-move forgery from a video using double compression and pixel estimation in GOP. (Labartino et al., 2013) Introduced a method to expose the video copy-move forgery with the help of footprint variation prediction, double quantization evidence, and histogram of DCT coefficients. A forgery detection technique using Variation of Prediction Footprint (VPF) and DCT coefficients for the examination of frame insertion and deletion was developed by (Gironi et al., 2014). (H. Liu et al., 2014) designed a system with the help of order of the residual average of *P*-frames to identify the deleted frames. The approach based on Spatially Constrained Residual Errors (SCREs) was presented by (Aghamaleki & Behrad, 2016) to expose frame deletion and insertion. They also designed another approach using the Time-domain analysis of residual errors to identify frame deletion and insertion forgery. (S. M. Fadl et al., 2018) designed a method using the notion of residual frames and entropy of DCT coefficients to recognize the video inter-frame duplication. The limitations of compression artifact-based forgery detection techniques are shown in Table 2.1.

Table 2.1: Pros and Cons of compression artifact-based forgery detection techniques

Ref.	**Pros**	**Cons**
(W. Wang & Farid, 2006)	Exposed frame insertion and deletion	Noise Sensitive. Sensitive for GOP size change. Localization of forgery is not done.
(W. Wang & Farid, 2009)	Exposed frame insertion and deletion. Localization is done but not precise	Depends on the compression rate. Works with a fixed GOP size and static background videos.
(Subramanyam & Emmanuel, 2012)	Exposed region manipulation	Localization of forgeries is not performed. Works on fixed GOP size and static background videos. Accuracy depends on the forged region size.
(Subramanyam & Emmanuel, 2013)	Exposed region manipulation	Localization of forgeries is not performed. Works with a fixed GOP size and static background videos. Quantization Scale ratio affects the accuracy.

(Labartino et al., 2013)	Exposed region manipulation from MPEG-2 coded videos. Localization is done.	Estimating the GOP is impossible. Only Works with a fixed GOP size and static background videos.
(Gironi et al., 2014)	Exposed frame insertion and deletion. Localization is done	Forgery localization is not accurate. Performance is depended on GOP structure size.
(H. Liu et al., 2014)	Exposed frame deletion.	Handle only one type of forgery. Work with GOP videos that are fixed in size. There is no localization.
(Aghamaleki & Behrad, 2016)	Exposed frame insertion and deletion. Localization is done.	Performance is changed for the video to have a low compression ratio and moving background.
(Aghamaleki & Behrad, 2017)	Exposed frame insertion, deletion and duplication. Localization is done.	Performance relies on the video background.
(S. M. Fadl et al., 2018)	Exposed frame duplication with Recall of 98.3%. Localization is done.	Single forgery is investigated. Not useful for varied-size GOP video.

2.1.2 Noise Artifacts based Techniques

The noise is an inherent trait or proof that can be used to identify tampering from a video. Techniques based on noise artifacts rely on sensor artifacts generated by digital cameras. This approach may also be termed as a camera-based detection approach. This is because a digital camera normally leaves a unique fingerprint/trace in the form of noise which the researcher can use to expose the video's forgeries. Figure 2.3 depicts the noise artifacts utilized in video forgery identification. Several noises are used to identify a forgery in the video, including fixed pattern noise, photon shot noise, quantization noise, photo response non-uniformity noise, and sensor pattern noise.

(Mondaini et al., 2007) used photo response non-uniformity noise and fixed pattern noise in the video to handle the copy-move and frame insertion forgery. The system based on the

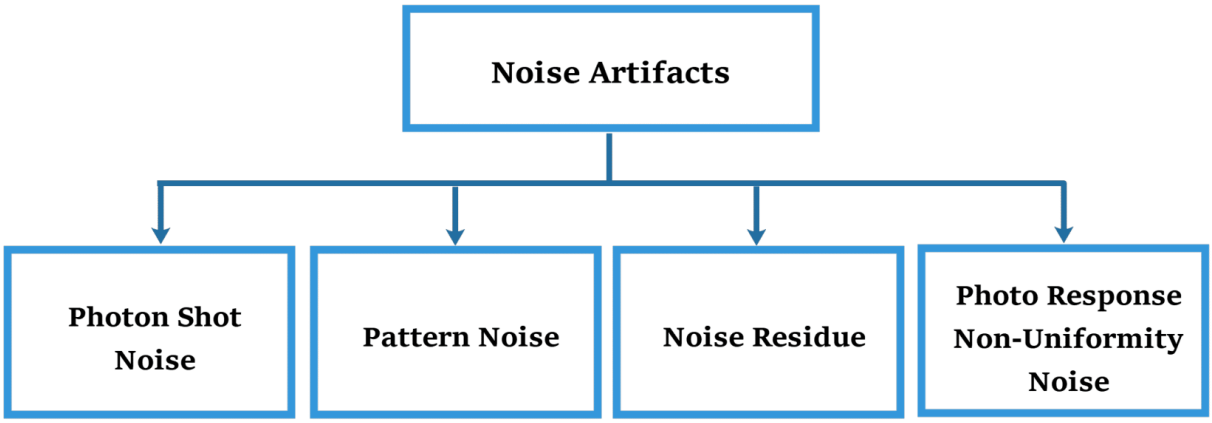


Figure 2.3: Noise artifacts

Noise Level Function (NLF) and inconsistencies in photon shot noise was presented by (Kobayashi et al., 2009) for identifying manipulated regions in the video. Moreover, they modified the existing work and suggested a new approach using Non-linear NLF photon shot noise inconsistencies to expose the region manipulations. The passive system using the extraction of quantization residue and noise features to handle the copy-move forgery was presented by (Chetty et al., 2010). A scheme designed by (Hyun et al., 2013) used sensor pattern noise and Minimum Average Correlation Energy (MACE) filter to detect the partial manipulation and upscale cropping in the video by estimating the scalar factor and correlation coefficient. (Ravi et al., 2014) proposed the modified Huber Markov Random Field (HMRF) model to handle copy move and inter-frame deletion in the video by investigating compression noise. (Pandey et al., 2014) developed a strategy for the investigation of video frame duplication tampering by the use of noise residue and wavelet denoising. Extrinsic camera parameters were utilized by (X. Hu et al., 2015) to develop a forgery detection system to identify the forged region. (R. D. Singh & Aggarwal, 2017b) proposed a technique to expose upscale-crop and splicing forgery with the help of noise-inconsistency and pixel-correlation investigation. (R. D. Singh & Aggarwal, 2017a) extended their technique to develop a scheme using Sensor Pattern Noise Correlation (SPNC), Color Filter Array Artifacts (CFA-V), and Duplicate Cluster Detection Scheme (H-DC) to handle the video copy-move forgery. (Fayyaz et al., 2020) suggested the method to handle copy-move forgery using sensor pattern noise. To detect the forgeries, the residue noise patterns were extracted from each frame and matched with the noise accumulated using adaptive DCT filtering. The limitations of noise artifact-based forgery detection techniques are shown in Table 2.2.

Table 2.2: Pros and Cons of noise artifacts-based forgery detection techniques

Ref.	**Pros**	**Cons**
(Mondaini et al., 2007)	Exposed copy move and frame insertion from MPEG compressed videos.	Adequate for fixed GOP size and static background video. Post-processing operations and compression change performance. Inadequate validation. Missing localization.
(Kobayashi et al., 2009)	Exposed region manipulation. Localization is done.	Reliable only on videos compressed using lossless huffyuv codec. Not beneficial for varied size GOP and moving scene videos. Not robust against the brightness.
(Kobayashi et al., 2010)	Exposed region manipulation. Localization is done.	Accuracy depends on the video codec. Not beneficial for varied size GOP and moving scene videos. Not robust to operations like compression, change in brightness and contrast.
(Chetty et al., 2010)	Exposed copy move with accuracy of 92%.	Suited to the moving background video. Single forgery investigation.
(Hyun et al., 2013)	Exposed upscale-crop from MPEG-4 oded videos. Localization is done	Moving background or objects in the video affect the performance. Average performance. Recommended for MPEG-4 videos only.
(Ravi et al., 2014)	Exposed copy move, frame duplication & deletion.	Performance is depends upon quantization scale. Missing Localization.
(Pandey et al., 2014)	Exposed frame duplication. Localization is done	Not robust to compression and postprocessing operations. Not recommended for varied size GOP and moving background videos.
(X. Hu et al., 2015)	Exposed region manipulation. Localization is done.	Camera parameters affect performance. Single forgery investigation. Inadequate validation
(R. D. Singh & Aggarwal, 2017b)	Exposed upscale-crop and splicing. Localization is done.	Not beneficial for varied size GOP and moving scene videos.

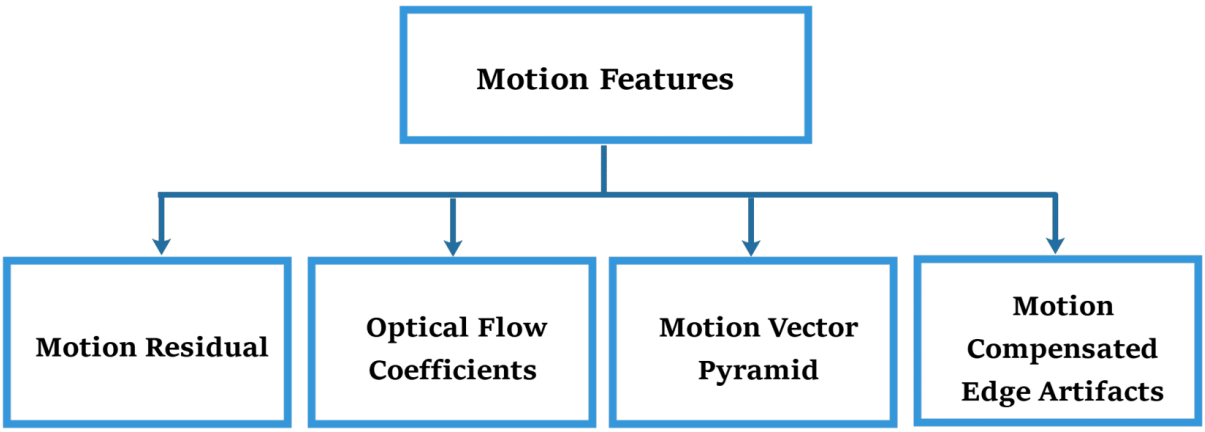


Figure 2.4: Motion artifacts

(R. D. Singh & Aggarwal, 2017a)	Exposed copy move. Localization is done.	Single forgery detection. Not beneficial for varied size GOP and moving scene videos.
(Fayyaz et al., 2020)	Exposed region manipulation. Localization is done.	Single forgery detection. Video background dependency.

2.1.3 Motion Features based Techniques

Motion features are time-dependent and determine the relationships between consecutive frames (Harit & Chaudhury, 2007). When someone tampers with a video, the motion attributes/features and relationships between consecutive frames are altered, which can be used as a trace of tampering. The motion features are represented by optical flow coefficients, motion residual, motion-compensated edge artifacts (MCEA), and motion vector pyramid. Motion artifacts/features for video forgery detection are depicted in Figure 2.4.

(Y. Su et al., 2009) proposed forgery detection system based on MCEA motion feature for video inter-frame deletion. Another passive framework was devised by (Dong et al., 2012), which is also based on the MCEA motion artifact to handle the frame insertion and deletion from a video. (L. Li et al., 2012) proposed a block-based motion estimation approach to investigate the video copy-move forgery. They noticed that removing a specific object from a video sequence changed the motion vector in order to detect the forgery. Motion evidence such

as a motion vector was obtained from adjacent frames of the video. (Bestagini et al., 2013) proposed a scheme using residual footprints that detects forgery, such as region manipulation in videos. Apart from that, they also expanded the size of the SULFA (Qadir et al., 2012) dataset by including additional forged videos.

An optical flow is one of the motion-based features that has been used in many research articles to detect forgeries in the video. In an actual video, optical flow differences among continuous frames appear to be less or more constant; however, when a video is subjected to inter-frame tampering, the optical flow begins to expose irregularities that is to be used as a fingerprint. (Chao et al., 2012) presented a forgery identification system using a window-based detection approach for the video by analyzing the optical flow features consistency to handle video inter-frame deletion and insertion. However, for detecting frame deletion forgery, a double adaptive threshold-based approach was used. (W. Wang et al., 2013) presented a forgery detection model based on continuity in the optical flow distribution. They found discontinuity points by extracting optical flow sequences from neighbouring frames and detecting video tampering (frame deletion, duplication or insertion). An algorithm based on fluctuating feature of motion residual and adaptive threshold scheme to detect and localizes the video inter-frame deletion was proposed by (Feng et al., 2014). (Wu et al., 2014) used Velocity Field Sequence (VFS), block-based cross-correlation and Generalized Extreme Studentized Deviate (ESD) to detect tampering such as frame duplication/frame deletion. (Q. Wang et al., 2014b) presented a forgery detection method based on an inconsistency to handle frame insertion/deletion from the video. The scheme based on GOP structure analysis and residual motion features for object-based forgery (copy-move forgery) identification was presented by (Tan et al., 2015). Bidokhti & Ghaemmaghami, 2015 proposed a technique to examine video tamperings such as copy-move and frame duplication incorporating optical flow features. (Z. Zhang et al., 2015) used the motion vector pyramid consistency and discontinuity points in the variation factor sequence to identify the frame duplication and frame deletion. (Yu et al., 2016) suggested a strategy based on the feature such as Prediction Residual (PR) and Number of Intra Macroblocks (NIMB's) to expose frame deletion from a video by analyzing sudden changes in a video sequence. An algorithm using motion residual features was developed by (S. Chen et al., 2016) to examine the region manipulation (like removal/insertion of objects) in the video. An optical flow and prediction residual artifacts were employed by (R. D. Singh & Aggarwal, 2017c) to examine the inter-frame forgeries. They emphasized the gradient brightness parameters of optical flow

for the frame deletion/addition detection from a video. (Kingra et al., 2017) proposed a passive technique to investigate video tampering (frame duplication, insertion, and deletion) by using optical flow artifacts and the prediction residuals. They evaluated how the temporal correlations between consecutive frames were disturbed when a video was forged. (Sitara & Mehtre, 2017) developed a scheme to examine the video inter-frame insertion, duplication, deletion, and shuffling using a variation of prediction footprint and velocity field consistency. They designed a generalized Extreme Studentized Deviate (ESD) algorithm to locate the forged points. A method based on Quantitative Correlation Rich Region (QCRI) concept, optical flow feature, and Gradient Structure Similarity (GSSIM) feature to examine the frame deletion forgery is recommended by (Pu et al., 2019). The limitations of motion features-based forgery detection techniques are shown in Table 2.3.

Table 2.3: Pros and Cons of motion features-based forgery detection techniques

Ref.	**Pros**	**Cons**
(Y. Su et al., 2009)	Exposed frame deletion. Localization is done.	Slow-motion videos and videos with a variable-length GOP size are not recommended. If the entire GOP is removed, the experiment fails.
(Dong et al., 2012)	Exposed frame insertion and deletion. Localization is done.	Changes in the GOP length, background and frame deletion count of the video affect the performance. Compression attacks affect the accuracy.
(L. Li et al., 2012)	Exposed region manipulation. Localization is done.	Detection of rigid object removal is its sole focus. Work on videos that have a fixed- GOP size and a motionless background.
(Bestagini et al., 2013)	Exposed region manipulation. Localization is done.	The quantization parameter affects performance, which is affected when quantization is recognized in the second encoding phase. Not recommended for varied size GOP and moving scene videos.
(Chao et al., 2012)	Exposed frame insertion and deletion. Loclaization is done.	Low accuracy for frame deletion detection. Not recommended for varied size GOP and moving background videos.
(W. Wang et al., 2013)	Exposed frame deletion duplication, insertion. Localization is done.	The optical estimation method requires improvement. Not recommended for a moving background and a variety of length GOP videos.

(Feng et al., 2014)	Exposed frame insertion. Localization is done.	Not robust to the abrupt zooming and lightning change. Bit rate affects the accuracy. Not fit for moving scene videos. Mange to handle one type of forgery.
(Wu et al., 2014)	Exposed frame deletion and duplication.	Compression and quantization scale affects the accuracy. Adequate for videos with a static background.
(Q. Wang et al., 2014b)	Exposed frame insertion and deletion	Missing Localization. Inappropriate for the varied GOP size and moving scene videos. Computationally ineffective.
(Tan et al., 2015)	Exposed region manipulation.	Low accuracy. Recommended for fixed GOP size and stationary background videos. Localization is not done.
(Bidokhti & Ghaemmaghami, 2015)	Exposed copy move and frame duplication.	The selection of the Region of Interest is extremely sensitive. Not recommended for GOPs with a range of lengths and a lot of motion.
(Z. Zhang et al., 2015)	Exposed frame deletion and duplication.	Suitable for MPEG-2 coded videos, including stationary-background and fixed-length GOP. Frame duplication accuracy is low. Not robust to videos with other encoding standards.
(Yu et al., 2016)	Exposed frame deletion. Localization is done.	Performance depends on frame count. Dependent on the background of the videos.
(S. Chen et al., 2016)	Exposed region manipulation. Localization is done.	Localization is not precise. Not useful for high bit rate and high-resolution videos. Work on fixed-length GOP videos. Not suggested for the moving camera videos.
(R. D. Singh & Aggarwal, 2017c)	Exposed frame insertion, replication and deletion. Localization is done.	Useful for videos with stationary background. Not suitable for multiple compressed videos.

(Kingra et al., 2017)	Exposed frame insertion, deletion and duplication. Localization is done. Suggested for static background videos.	Not robust to high illumination and varying length GOP.
(Sitara & Mehtre, 2017)	Exposed frame insertion, replication, deletion and duplication. Localization is done.	Not robust against compression and sudden zooming. Not recommended for the moving background videos.
(Pu et al., 2019)	Exposed frame deletion. Localization is done.	Concentrate solely on one type of forgery. Not useful for night videos. Dependent on the background and GOP length.

2.1.4 Statistical Features based Techniques

Techniques based on the statistical features examine objects, including statistical features, pixel-level variance, texture feature and correlation. The statistical feature-based method is also termed as Geometric/physical consistencies-based techniques because it works with consistency parameters in video frames (like brightness, lighting, zooming, shadows). After performing the forgeries in the video, the statistical properties may be altered, which are then examined to identify the manipulations. The statistical features utilized in forgery detection are shown in Figure 2.5.

(W. Wang & Farid, 2007) adopted the correlation coefficient as a standard measure to identify the forged area. (R. Chen et al., 2012) employed the statistical features of the video using the Adjustable Width Object Boundary (AWOB) algorithm to identify forged regions. To identify the forgery, the gradients information and contourlet coefficient features were obtained from the frames. A passive forgery detection technique developed by (Y. Hu et al., 2012) was used to expose inter-frame duplication with the help of Temporally Informative Representative Images (TIRI), DCT coefficients and hamming distance. (Lin & Chang, 2012) developed another passive technique for handling video inter-frame duplication using histogram difference among contiguous frames in RGB components as a forensic feature. A spatial similarity

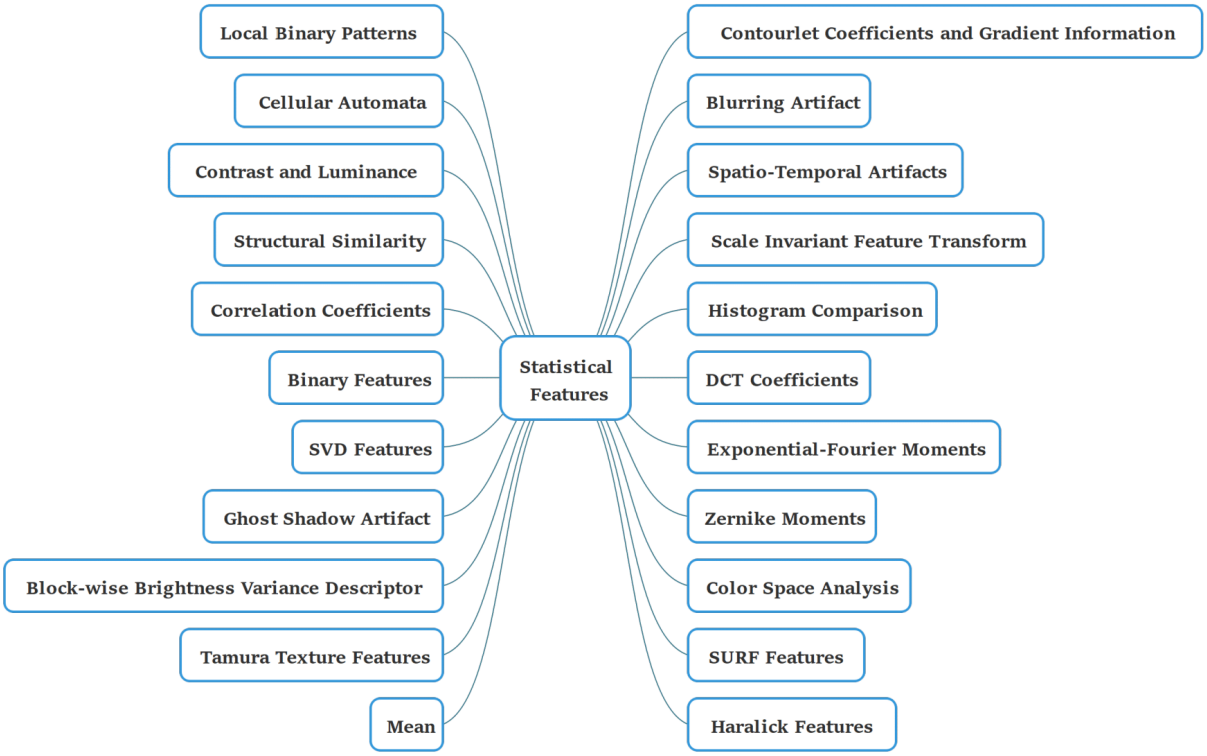


Figure 2.5: Statistical features

analysis was also carried out, in which the block-based algorithm detected a sudden correlation change among the two frames. Based on Tamura texture features (Tamura et al. (1978)) (like contrast, roughness, directionality *etc.*), (Liao & Huang, 2013) proposed a method for handling the video inter-frame duplication. A scheme to identify the video inter-frame duplication proposed by (F. Li & Huang, 2014) using the structural similarity concept. (Zheng et al., 2014) developed a system for handling frame insertion forgery with the use of Block-wise Brightness Variance Descriptor (BBVD) algorithm. Correlation Coefficients of Gray Values (CCCoGV) feature among the video frames were used by (Q. Wang et al., 2014a) to design a passive technique for the investigation of frame deletion/insertion video forgery. (Yin et al., 2014) devised an approach based on time-dimension attribute inconsistency and Nonnegative Tensor Factorization (NTF) to investigate frame deletion/insertion video forgery. (Chittapur et al., 2014) analyzed temporal difference between each video frame and suggested a method to expose forged regions based on statistical properties such as pixel comparison and mean of the pixels in the frame. (Tralic et al., 2014) employed statistical features such as Local Binary Patterns (LBP) and Cellular Automata (CA) to investigate the video frame duplication. The frames were divided into blocks, and LBP and CA features were then obtained from each block, and

the tampering was detected using a histogram. (V. K. Singh et al., 2015) proposed a scheme to handle video frame duplication using block-based features (*i.e.,* the mean and residue of block) of a frame. Next, the lexicographical sorting concept was employed to the collected features to obtain a group of similar frames. Finally, the Root Mean Square Error (RMSE) value and correlation among the adjacent frame were calculated to identify the tampering. (Pandey et al., 2014) developed a passive scheme to examine video copy-move tampering using SIFT feature. A compressive sensing scheme based on SVD feature and k-means clustering was proposed by (L. Su et al., 2015) to investigate the removal of moving foreground in the video with a stationary background. By extracting blurring artifacts the cross-correlation between the video foreground blocks and background was computed by (Bagiwa et al., 2016) to design a passive approach for detecting the chroma key forgery. The correlation coefficients were used by (Xu et al., 2016) to expose the inter-frame forgeries. (Z. Li et al., 2016) investigated the facts that Quotient of Mean Structural Similarity (QoMSSIM) features were consistent in the case of an original sequence and fluctuating in the tampered video. As a result, they designed a passive approach to identify frame deletion/insertion forgery. Their method was resistant to recompression as well as white Gaussian noise. (Mathai et al., 2016) presented a passive technique for the identification of content duplication in video using moment features such as prediction-error array and cross-correlation. (J. Yang et al., 2016) developed a passive method for frame duplication forgery detection in video using the SVD features. There were two stages to the method. In the first stage, the SVD feature extraction algorithm was employed to collect the frame features. The distance between the features of each frame was then calculated using Euclidean distance. The duplications in the video were identified in the second stage by using random block matching. Zernike Opponent Chromaticity Moments (ZOCM) feature was used by (Y. Liu & Huang, 2017) to develop a passive technique to reveal the inter-frame tampering from a video. (BOZKURT et al., 2017) developed a scheme for detecting video frame duplication forgery using a DCT and Hough transform. Based on the binary features, Euclidean distance metrics, and Peak Signal to Noise Ratio (PSNR), an approach was designed by (Ulutas et al., 2017) to examine the inter-frame tampering such as frame duplication and frame mirroring. (Ulutas et al., 2018) designed another technique to check the presence of frame duplication forgery with the use of the Bag of Words (BoW) model. This model produced visual information and built a dictionary from the SIFT feature of frames to detect duplications. A patch-based algorithm was proposed by (D'Amiano et al., 2018) to discover the video copy-move

temparing using the Zernike moments feature. The Speeded-up Robust Features (SURF), Hue Saturation Value (HSV) features, Fast Library for Approximate Nearest Neighbors algorithms, and color histogram comparison approach were used by (Zhao et al., 2018) to detect the video inter-frame forgeries. Exponential Fourier Moments (EFM) features were used by L. Su et al., 2018 to design an approach to identify video copy-move tampering using Post-Verification Scheme (PVS) and Adaptive Parameter based Fast Compression Tracking (AFCT). Their approach was efficiently adopted for the tampered region having a mirroring attack. (L. Su & Li, 2018) proposed a new approach based on spatio-temporal context learning to expose video copy-move tampering with the help of Mirror-invariant and Inversion-invariant SIFT (MI-SIFT) feature. Furthermore, (L. Su et al., 2019) designed another approach to expose the video region manipulation forgery using variable bit-rate compression. An Adaptive Parameter-based Visual Background Extractor (AVIBE) was adopted to develop the energy factor-based approach to examine the manipulations from a video. Their approach was resistant to noise addition and brightness change. (Wei et al., 2019) proposed a passive algorithm by using multi-scale standardized mutual information to detect the inter-frame video forgery. (Bakas et al., 2019) proposed a framework for the video forgery detection using Harlick features to detect the video inter-frame forgery. Based on Spatio-temporal analysis (Bai et al., 2019) presented an approach to detect the video in-painting forgery using LBP feature. (Aparicio-Díaz et al., 2019) proposed a block correlation matrix scheme to examine the copy-move and frame duplication forgery. The block correlation matrix was utilized to store both the pixels' spatial-temporal information to detect forgery. (Saddique et al., 2019) developed a scheme to identify the forged region by exploiting texture inconsistency, chrominance value and LBP features. (Aloraini et al., 2019) developed a method based on Spatio-temporal filter, Laplacian pyramid, and estimation of object movement for detecting the region manipulation in the video. Furthermore, (Aloraini et al., 2020) modified the existing method and developed a new method using Sequential and Patch analysis to identify video region manipulation. The SIFT features and motion vectors were utilized by (Kharat & Chougule, 2020) to design an approach for investigating the video inter-frame duplication. Motion Energy Image (MEI) and HOG features were used by (S. Fadl et al., 2020) in their proposed passive technique to expose the inter-frame forgery. The limitations of statistical features-based forgery detection techniques are shown in Table 2.4.

Table 2.4: Pros and Cons of statistical features-based forgery detection techniques

Ref.	**Pros**	**Cons**
(W. Wang & Farid, 2007)	Exposed frame duplication and copy move. Localization is done but not precise.	Changes in bit rate and forged region area have an impact on accuracy. Videos with a varied GOP size and moving scene are not recommended.
(R. Chen et al., 2012)	Exposed region manipulation. Localization is done.	Only work on the video with a static background. There is a lack of localization.
(Y. Hu et al., 2012)	Exposed frame duplication.	Works with videos that have a fixed GOP length and a static background. There is no localization.
(Lin & Chang, 2012)	Exposed frame duplication.	To improve efficacy, other features must be combined. Average performance. Videos with a varied GOP size and moving scene are not recommended.
(Liao & Huang, 2013)	Exposed frame duplication.	To reduce computation time, other features must be combined. Only detect one type of forgery.
(F. Li & Huang, 2014)	Exposed frame duplication.	Videos with a long-time static scene, a moving background, and a variable GOP structure are not suitable. High computational time.
(Zheng et al., 2014)	Exposed frame insertion.	Improvement is needed for Localization. Count of the forged frame affects the accuracy. Videos with a varied GOP size and moving scene are not recommended.
(Q. Wang et al., 2014a)	Exposed frame insertion and deletion	Changes in GOP length and video background affect the performance.
(Yin et al., 2014)	Exposed frame insertion and deletion	Efficiency depends on frame deletion count. Dependent on video background and GOP structure.
(Chittapur et al., 2014)	Exposed region manipulation. Localization is done.	Only useful for fixed GOP structure & a static video background. Handle single forgery detection at a time. Validation is insufficient.
(Tralic et al., 2014)	Exposed frame duplication. Localization is done.	When multiple frames are duplicated, it does not work properly. Depends on the GOP size. Not useful for moving the video background.

(V. K. Singh et al., 2015)	Exposed frame duplication	Handle only one type of forgery. It only works with videos that have a fixed-length GOP structure.
(Pandey et al., 2014)	Exposed copy move	As compression increases, accuracy decreases. Not suitable for varied GOP length and moving background videos. No localization.
(L. Su et al., 2015)	Exposed region manipulation. Localization is done.	If the deleted object is fast-moving and small, detection accuracy suffers. Computationally expensive. Depends on the background and GOP size.
(Bagiwa et al., 2016)	Exposed chroma Key.	If the video's background is blue/green color, the performance suffers. Depends on the GOP size. Not useful for moving the video background.
(Xu et al., 2016)	Exposed frame insertion, deletion & duplication.	Accuracy depends on the count of frames. Not robust to the compression, video background, and GOP length. No forgery localization.
(Z. Li et al., 2016)	Exposed frame insertion, deletion	The number of frames inserted or deleted affects performance. Depends on the GOP size. Not useful for moving the video background. There is no localization.
(Mathai et al., 2016)	Exposed copy move. Localization is done.	Accuracy is affected by the window size. Localization needs to be improved. Suited to those videos having varied GOP size and moving background scene.
(J. Yang et al., 2016)	Exposed frame duplication. Localization is done.	Single forgery detection. GOP dependent. Not robust to moving background. Failed when duplicated frames are less than the size of the window.
(Y. Liu & Huang, 2017)	Exposed frame insertion, deletion & duplication.	Not robust to moving background and GOP length. Missing localization.
(BOZKURT et al., 2017)	Exposed frame duplication. Localization is done.	Identify only one type of forgery. Not recommended for videos having a moving background or of varying GOP lengths.
(Ulutas et al., 2017)	Exposed frame duplication. Localization is done.	Not recommended for videos having a moving background or of varying GOP lengths.

(Ulutas et al., 2018)	Exposed frame duplication. Localization is done.	Handle only one type of forgery.
(D'Amiano et al., 2018)	Exposed copy move. Localization is done.	Not recommended for videos having a moving background or of varying GOP lengths.
(Zhao et al., 2018)	Exposed frame insertion, deletion & duplication. Localization is done.	Not suggested for videos having a moving scene or of varying GOP lengths.
(L. Su et al., 2018)	Exposed copy move. Localization is done.	Single forgery investigation. Not suited to videos with varying GOP lengths & a moving scene.
(L. Su & Li, 2018)	Exposed copy move with mirrored operation.	Handle only one type of forgery. Videos with a dynamic background and variable GOP size are not recommended.
(L. Su et al., 2019)	Exposed region manipulation.	Performance depends on the size of the forged area. Only deal with one type of forgery. Localization isn't exact.
(Wei et al., 2019)	Exposed frame insertion, deletion & duplication. Localization is done.	Not recommended for videos having a moving background or of varying GOP lengths.
(Bakas et al., 2019)	Exposed frame insertion, deletion & duplication. Localization is done. Independent on GOP length and video background.	Accuracy is concerned with a fast-moving scene. Post-processing operations on forged video affect the performance.
(Bai et al., 2019)	Exposed region manipulation. Localization is done.	Performance is affected by shaking and slight rotation. Not fit for moving background and varying GOP lengths video.
(Aparicio-Díaz et al., 2019)	Exposed copy move & duplication.	Not suggested for moving background and varying GOP lengths video.
(Saddique et al., 2019)	Exposed copy move & splicing. Localization is done.	Not suited to videos with varying GOP lengths and a moving scene.
(Aloraini et al., 2019)	Exposed region manipulation. Localization is done.	Not useful for moving background video. GOP dependent. Support single forgery detection.

(Aloraini et al., 2020)	Exposed region manipulation. Localization is done.	Not robust to change in GOP size and video background. Support single forgery detection.
(Kharat & Chougule, 2020)	Exposed frame duplication. Localization is done.	Depends on the GOP size. Not useful for moving the video background. Single forgery detection.
(S. Fadl et al., 2020)	Exposed frame insertion, deletion & duplication. Localization is done.	Not robust to change in GOP size. At a time one forgery is detected.

2.1.5 Machine Learning based Techniques

These techniques are data-driven, and they can automatically learn the complex features/artifacts required to identify forgery from a video. (Yao et al., 2017) devised a deep Convolutional Neural Network (CNN) model to investigate the copy-move tampering from a video. Their model was based on a Caffe deep learning framework (Jia et al., 2014) training on labelled training data set (SYSU-OBJFORG video dataset (S. Chen et al., 2016). (Long, Smith, Basharat, & Hoogs, 2017) used the 3D-CNN model to identify the frame deletion tampering by utilizing the Spatio-temporal correlation. Their model tested on videos taken from the YFCC100m (YFCC, 2019) and Nimble Challenge 2017 dataset (NIMBLE, 2017). (D'Avino et al., 2017) proposed a neural model using autoencoder and recurrent neural networks to detect frame splicing manipulation. The model is trained on extracted residual features from each of the frames. (Kono et al., 2018) investigated Spatio-temporal consistency in video and suggested a hybrid model by combining CNN and Long Short-term Memory (LSTM) model to identify the region manipulation from a video. The CNN was used to determine the video's spatial aspects, while the LSTM model determined the videos' temporal aspects. (Hong et al., 2019) proposed a technique to identify the frame deletion tampering. Their system was divided into two components. The first component extracted valuable characteristics from compressed coding data. The second component employed classification algorithms such as Linear Discriminant Analysis (LDA), Multilayer Perceptron (MLP), and KNN to determine the video's authenticity. (Johnston et al., 2020) proposed a technique for identifying tampered regions from the video. They implemented the CNN model to determine the compression parameters. The CNN model was also used by (Zampoglou et al., 2019) to investigate the region manipulation and frame insertion in video.

They devised a forensics filter using quantization error and DCT. The outputs of these forensics filters were then fed into the CNN, which used them to distinguish between original and tampered video. The limitations of machine learning-based forgery detection techniques are shown in Table 2.5.

Table 2.5: Pros and Cons of machine learning-based forgery detection techniques

Ref.	**Pros**	**Cons**
(Shanableh, 2013)	Exposed frame deletion from MPEG-2 coded videos.	Exact localization is not carried out. Handle only one type of forgery.
(Yao et al., 2017)	Exposed copy move	There is no localization. Moving background videos with high resolution and bit rate are not suitable. High computational cost. Labelled video data is required.
(Long et al., 2017)	Exposed frame deletion	High computational time is required. Useful with GOP video that has a fixed length. Useful with a single-shot video. Single forgery investigation. Labelled video data is required.
(D'Avino et al., 2017)	Exposed frame splicing. Localization is done.	More computational time is required. Not robust against the change in video background and GOP length.
(Kono et al., 2018)	Exposed region manipulation. Localization is done.	Not ideal for variable-length GOP videos. Handle only one kind of forgery.
(Hong et al., 2019)	Exposed frame deletion	Work with static background video having fixed GOP size.
(Johnston et al., 2020)	Exposed region manipulation, copy move and splicing.	Multiple manipulations are not detectable. Not robust against the change in video background and GOP length.
(Zampoglou et al., 2019)	Exposed inter-frame forgeries and region manipulation	Missing Localization. Labelled dataset is needed. Adequate for static background videos with a fixed GOP size.

The complete details of the state of the art passive video forgery identification techniques are shown in Table 2.6 (1- Copy-move forgery, 2-Frame splicing forgery, 3-Region tamper-

ing, 4-Frame insertion forgery, 5-Frame deletion forgery, 6-Frame duplication forgery, 7-Frame replication forgery, 8-In-painting forgery, 9- Upscale crop forgery, 10-Mirror invariant forgery, 11-Detection, 12-Localization, 13-Fixed GOP size, 14-Variable GOP size, 15-Static video background & 16: Moving video background).

Table 2.6: Details of forgery identification techniques

Ref.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
(W. Wang & Farid, 2006)	×	×	×	✓	✓	×	×	×	×	×	✓	×	✓	×	✓	×
(W. Wang & Farid, 2009)	×	×	×	✓	✓	×	×	×	×	×	✓	✓	✓	×	✓	×
(Subramanyam & Emmanuel, 2012)	×	×	✓	×	×	×	×	×	×	×	✓	×	✓	×	✓	×
(Subramanyam & Emmanuel, 2013)	×	×	✓	×	×	×	×	×	×	×	✓	×	✓	×	✓	×
(Labartino et al., 2013)	×	×	✓	×	×	×	×	×	×	×	✓	×	✓	×	✓	×
(Gironi et al., 2014)	×	×	×	✓	✓	×	×	×	×	×	✓	✓	✓	✓	✓	×
(H. Liu et al., 2014)	×	×	×	×	✓	×	×	×	×	×	✓	×	✓	×	✓	×
(Aghamaleki & Behrad, 2016)	×	×	×	✓	✓	×	×	×	×	×	✓	✓	✓	✓	✓	×
(Aghamaleki & Behrad, 2017)	×	×	×	✓	✓	×	×	×	×	×	✓	✓	✓	✓	✓	×
(S. M. Fadl et al., 2018)	×	×	×	×	×	✓	×	×	×	×	✓	✓	✓	×	✓	×
(Mondaini et al., 2007)	✓	×	✓	✓	×	×	×	×	×	×	✓	×	✓	×	✓	×
(Kobayashi et al., 2009)	×	×	✓	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Kobayashi et al., 2010)	×	×	✓	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Chetty et al., 2010)	✓	×	×	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Hyun et al., 2013)	×	×	×	×	×	×	×	×	✓	×	✓	✓	✓	×	✓	×
(Ravi et al., 2014)	✓	×	×	×	✓	×	×	×	×	×	✓	×	✓	×	✓	×
(Pandey et al., 2014)	×	×	×	×	×	✓	×	×	×	×	✓	✓	✓	×	✓	×
(X. Hu et al., 2015)	×	×	✓	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(R. D. Singh & Aggarwal, 2017b)	×	✓	×	×	×	×	×	×	✓	×	✓	✓	✓	×	✓	×
(R. D. Singh & Aggarwal, 2017a)	✓	×	×	×	×	×	×	×	×	×	✓	✓	✓	✓	✓	✓
(Fayyaz et al., 2020)	×	×	×	×	×	×	×	✓	×	×	✓	✓	✓	×	✓	×
(Y. Su et al., 2009)	×	×	×	×	✓	×	×	×	×	×	✓	✓	✓	×	✓	×
(Dong et al., 2012)	×	×	×	✓	✓	×	×	×	×	×	✓	✓	✓	✓	✓	×
(L. Li et al., 2012)	×	×	✓	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Bestagini et al., 2013)	×	×	✓	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Chao et al., 2012)	×	×	×	✓	✓	×	×	×	×	×	✓	×	✓	×	✓	×

(W. Wang et al., 2013)	×	×	×	✓	✓	✓	×	×	×	×	✓	✓	✓	×	✓	×
(Feng et al., 2014)	×	×	×	×	✓	×	×	×	×	×	✓	✓	✓	✓	✓	×
(Wu et al., 2014)	×	×	×	×	✓	✓	×	×	×	×	✓	✓	✓	✓	✓	×
(Q. Wang et al., 2014b)	×	×	×	✓	✓	×	×	×	×	×	✓	×	✓	×	✓	×
(Tan et al., 2015)	×	×	✓	×	×	×	×	×	×	×	✓	×	✓	×	✓	×
(Bidokhti & Ghaemmaghami, 2015)	✓	×	×	×	×	✓	×	×	×	×	✓	✓	✓	×	✓	×
(Z. Zhang et al., 2015)	×	×	×	×	✓	✓	×	×	×	×	✓	✓	✓	×	✓	×
(Yu et al., 2016)	×	×	×	×	✓	×	×	×	×	×	✓	✓	✓	✓	✓	×
(S. Chen et al., 2016)	×	×	✓	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(R. D. Singh & Aggarwal, 2017c)	×	×	×	✓	✓	×	✓	×	×	×	✓	✓	✓	×	✓	×
(Kingra et al., 2017)	×	×	×	✓	✓	✓	×	×	×	×	✓	✓	✓	×	✓	×
(Sitara & Mehtre, 2017)	×	×	×	✓	✓	✓	✓	×	×	×	✓	✓	✓	✓	✓	×
(Pu et al., 2019)	×	×	×	×	✓	×	×	×	×	×	✓	✓	✓	×	✓	×
(W. Wang & Farid, 2007)	✓	×	×	×	×	✓	×	×	×	×	✓	×	✓	×	✓	×
(R. Chen et al., 2012)	×	×	✓	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Y. Hu et al., 2012)	×	×	×	×	×	✓	×	×	×	×	✓	×	✓	×	✓	×
(L. Li et al., 2012)	×	×	×	×	×	✓	×	×	×	×	✓	✓	✓	×	✓	×
(Liao & Huang, 2013)	×	×	×	×	×	✓	×	×	×	×	✓	✓	✓	×	✓	×
(F. Li & Huang, 2014)	×	×	×	×	×	✓	×	×	×	×	✓	✓	✓	×	✓	×
(Zheng et al., 2014)	×	×	×	✓	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Q. Wang et al., 2014a)	×	×	×	✓	✓	×	×	×	×	×	✓	✓	✓	×	✓	×
(Yin et al., 2014)	×	×	×	✓	✓	×	×	×	×	×	✓	✓	✓	×	✓	×
(Chittapur et al., 2014)	×	×	✓	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Tralic et al., 2014)	×	×	×	×	×	✓	×	×	×	×	✓	✓	✓	×	✓	×
(V. K. Singh et al., 2015)	×	×	×	×	×	✓	×	×	×	×	✓	✓	✓	×	✓	✓
(Pandey et al., 2014)	✓	×	×	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(L. Su et al., 2015)	×	×	✓	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Bagiwa et al., 2016)	×	✓	×	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Xu et al., 2016)	×	×	×	✓	✓	✓	×	×	×	×	✓	×	✓	×	✓	×
(Z. Li et al., 2016)	×	×	×	✓	✓	×	×	×	×	×	✓	×	✓	×	✓	×
(Mathai et al., 2016)	✓	×	×	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(J. Yang et al., 2016)	×	×	×	×	×	✓	×	×	×	×	✓	✓	✓	×	✓	×

(Y. Liu & Huang, 2017)	×	×	×	✓	✓	✓	×	×	×	×	✓	×	✓	×	✓	×
(BOZKURT et al., 2017)	×	×	×	×	×	✓	×	×	×	×	✓	✓	✓	×	✓	×
(Ulutas et al., 2017)	×	×	×	×	×	✓	×	×	×	✓	✓	✓	✓	×	✓	×
(Ulutas et al., 2018)	×	×	×	×	×	✓	×	×	×	×	✓	✓	✓	×	✓	✓
(D'Amiano et al., 2018)	✓	×	×	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Zhao et al., 2018)	×	×	×	✓	✓	✓	×	×	×	×	✓	✓	✓	×	✓	×
(L. Su et al., 2018)	✓	×	×	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(L. Su & Li, 2018)	✓	×	×	×	×	×	×	×	×	✓	✓	✓	✓	×	✓	×
(L. Su et al., 2019)	×	×	✓	×	×	×	×	×	×	×	✓	✓	✓	×	✓	✓
(Wei et al., 2019)	×	×	×	✓	✓	✓	×	×	×	×	✓	✓	✓	×	✓	×
(Bakas et al., 2019)	×	×	×	✓	✓	✓	×	×	×	×	✓	✓	✓	✓	✓	✓
(Bai et al., 2019)	×	×	×	×	×	×	×	✓	×	×	✓	✓	✓	×	✓	✓
(Aparicio-Díaz et al., 2019)	✓	×	×	×	×	✓	×	×	×	×	✓	✓	✓	×	✓	×
(Saddique et al., 2019)	✓	✓	×	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Aloraini et al., 2019)	×	×	✓	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Aloraini et al., 2020)	×	×	✓	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Kharat & Chougule, 2020)	×	×	×	×	×	✓	×	×	×	×	✓	✓	✓	×	✓	×
(S. Fadl et al., 2020)	×	×	✓	✓	×	×	×	×	×	×	✓	×	✓	×	✓	×
(Shanableh, 2013)	×	×	×	×	✓	×	×	×	×	×	✓	✓	✓	✓	✓	×
(Yao et al., 2017)	✓	×	✓	×	×	×	×	×	×	×	✓	×	✓	×	✓	×
(Long et al., 2017)	×	×	×	×	✓	×	×	×	×	×	✓	✓	✓	×	✓	✓
(D'Avino et al., 2017)	×	✓	×	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Kono et al., 2018)	×	×	✓	×	×	×	×	×	×	×	✓	✓	✓	×	✓	×
(Hong et al., 2019)	×	×	×	×	✓	×	×	×	×	×	✓	✓	✓	×	✓	×
(Johnston et al., 2020)	✓	✓	✓	×	×	×	×	×	×	×	✓	✓	✓	×	✓	✓
(Zampoglou et al., 2019)	×	×	✓	✓	×	×	×	×	×	×	✓	×	✓	×	✓	×

2.2 Research Gaps

On the basis of the literature survey, the following gaps have been identified

- The majority of the existing techniques are able to handle the single forgery at a time.

They are unable to examine multiple forgeries from a single video.

- Currently, most video encoders use a video with a different GOP size. Plenty of the approaches are effective for detecting forgery in GOP videos having a fixed length, only a few of them are effective for detecting tampering in videos with a varied size GOP.

- Different sorts of Statistical texture features such as Harlick, LBP, Zernike Moments and many more were used by the existing techniques to design the forgery detection techniques, and these features are not robust against the noise attack. Also, it has been somewhat challenging for academics to design new approaches that can cope with new types of noise (such as Gaussian noise, specal noise, salt and peeper noise) since there has been a noteworthy change in video noise over the previous 15 years (Birajdar & Mankar, 2013).

- Most of the existing techniques can detect manipulation in a video having a stationary background. Very few techniques have been proposed to identify tampering in a moving background video, so researchers will need to provide some solutions to this issue.

- Many forgery detection techniques are dependent on the codecs (like H.264, MPEG-2/3/4 *etc.*). These techniques could fall short for uncompressed forged videos. With these existing forgery detection techniques, accuracy drops as the compression ratio rises. Also, changes in video bitrates and quantization scale ratio affect the performance of forgery detection methods. The compression attribute in the video degrades the detection system's performance.

- In order to detect inter-frame forgery, the majority of current techniques are based on the number of frames that have been deleted, inserted, or duplicated.

- Many techniques for discovering tampering are only suitable for short-duration and low-resolution videos. As a result, researchers have a greater opportunity to design better methods for exposing the forgeries in long-run videos.

- The main limitation of existing forgery detection techniques is the necessity of forged video datasets for exploratory investigation. The existing datasets primarily include videos with one forgery only. Moreover, few databases are made up of a forged video having moving background. There are currently no publicly available forged video datasets that contain Intra and inter-frame forgeries.

- Most of the forgery identification schemes proposed in the literature have not considered robustness to operations like compression, noise addition, or brightness.
- Fewer techniques that use machine learning methods, particularly deep learning, have been developed. The use of these models in video forensics motivates researchers to develop automated forgery detection methods.

2.3 Research Objectives

The research objectives are as follows,

- To study and review existing approaches for the digital image and video forgery detection.
- To design the forgery detection technique to detect the multiple forgeries in the digital video.
- To design the technique that can localize the multiple forgeries in the digital video.
- To test and validate the proposed technique for better performance on video forensic dataset.

CHAPTER 3

Multiple Forgery Detection Technique for Digital video using LBP Features and Outlier Detection Approach

In this chapter, a passive technique based on the inter-frame correlation consistency between LBP-coded frames for video forgery detection is presented. In this Chapter, Section 3.1 discusses the background of LBP texture feature and Pearson correlation. Section 3.2 describes the proposed technique. Section 3.3 gives the experimental analysis and discussion, and the Conclusion is discussed in Section 3.4.

3.1 Background

In this section, the LBP texture feature and Pearson correlation are discussed. The texture is an important feature that is used in several computer vision applications (Silva et al., 2018). The primary goal of feature extraction is to extract the pertinent information from the data and interpret it in a lower-dimensional space. LBP is a texture description operator that calculates a texture description at the local level. This local description is built by comparing each pixel to its immediate neighborhood of pixels. LBP utilizes the signs of differences between neighboring and central pixels to describe the texture, examines the points surrounding a central point, and determines if the surrounding points are more or less than the central point.

The term correlation is commonly used to describe a relationship between two or more objects. The Pearson coefficient is a correlation coefficient describing the association among two variables measured on the likewise interval or ratio scale. It is used to determine the strength of the association between two variables. For linear relationships between two normally distributed variables, the Pearson correlation coefficient is probably the most commonly used measure, often just called the correlation coefficient. Usually, through a Least-Squares fit, the Pearson coefficient is obtained, and a value of 1 signifies a positive relationship, -1 signifies a negative relationship, and 0 signifies the lack of a relationship between variables. Let X and Y be two random variables. The following equation determines the Pearson correlation coefficient:

$$\rho(X,Y) = \frac{E(X,Y)}{\sigma_X \sigma_Y} \tag{3.1}$$

Where $E(X,Y)$ denotes the cross-correlation between X and Y. σ_X and σ_Y denotes the variances of X and Y.

3.2 Proposed Technique

This section covers the overall implementation of the technique for detecting forgeries from a video. The proposed technique work utilizes LBP texture features for the identification of intra-frame splicing and inter-frame insertion and deletion forgeries from the individual video. The LBP is one of the approaches of texture features extraction. There are also other image texture features like Zernike moments, Haralick features, and Scale Invariant Feature Transform (SIFT). Zernike moments features are computationally ineffective due to complex Zernike moments calculations. The disadvantages of Haralick features are their high dimensionality and high correlation between their features. Whereas SIFT feature one features are considered suitable for the identification of intra-frame forgery, specifically copy-move. The main advantages of LBP are its low calculation cost, resistance to fluctuations, and inability to be affected by variations in illumination. Therefore, LBP is selected in this work to extract features. The technique consists of four components, 1) Pre-processing, 2) LBP feature extraction, 3) Correlation computation 4) Outlier detection. The flow of the proposed technique is shown in Figure 3.1. The input video is first divided into a series of frames. Following that, the RGB frames are then transformed into a grayscale frame and the LBP features of each frame are obtained. The correlation coefficients between the LBP-coded frames are next computed in order to identify the forged video. Finally, the outlier detection approach detects and locates the multiple forged points present in the video. The overall steps of the proposed multiple forgeries detection and localization technique are given in the algorithm 3.1.

3.2.1 Pre-processing and LBP Feature Extraction

The input video is first separated into frames *i.e.,* $f_1, f_2, f_3 ..... f_n$ as part of the pre-processing procedure, the RGB frames are then transformed to grayscale frames. The formula to convert the RGB to grayscale conversion (Güneş et al., 2016) for a frame F is given by the following equation (3.2):

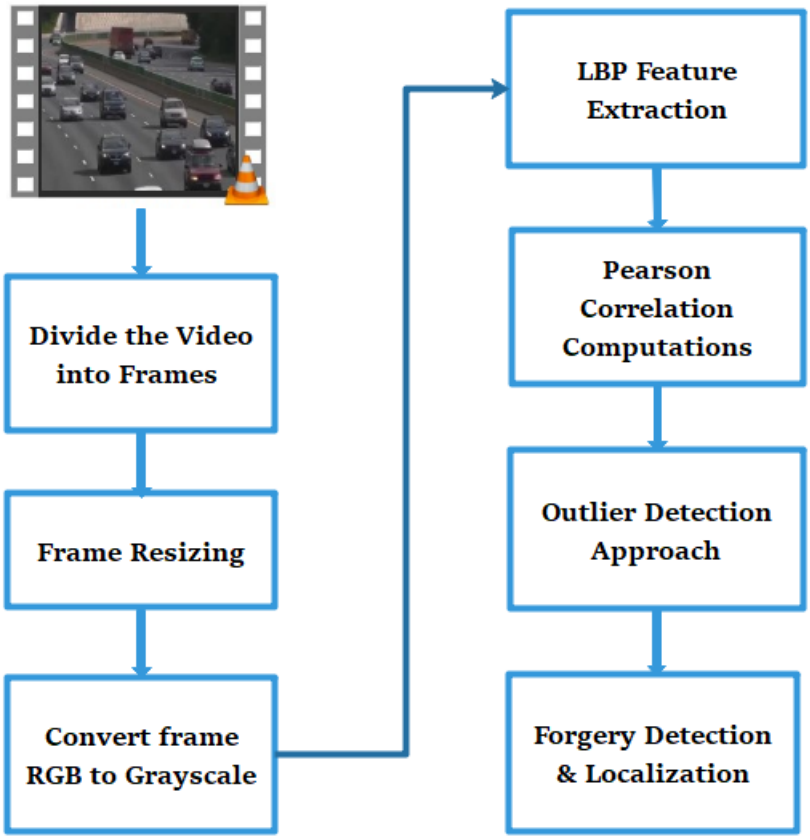


Figure 3.1: Proposed technique flowchart

$$F(i,j) = 0.2999 \times R(i,j) + 0.587 \times G(i,j) + 0.114 \times B(i,j) \tag{3.2}$$

Where $R(i,j)$, $G(i,j)$ and $B(i,j)$ indicates the red, green and blue components of the frame, respectively.

After Pre-processing, the LBP feature of a frame F, $F = (f_{i,j})_{j=1,2,\ldots,h}^{i=1,2,\ldots,w}$ is calculated, where, h and w are the height and width of the frame F, respectively. A frame pixel neighbourhood of size R encircling the center pixel is selected. For the center pixel, an LBP value is computed and saved in a $2D$ array of the same size as the input frame. The primary LBP descriptor works on a 3×3 fixed-sized neighbourhood of pixels as shown in Figure 3.2. Following that, the center pixel (marked in green color) is chosen against its neighbourhood pixels. If the intensity of the centre pixel is greater than or equal to that of its neighbor, the value is set to 1; otherwise, it is set to 0. There are a total of $2^8 = 256$ LBP code combinations possible for

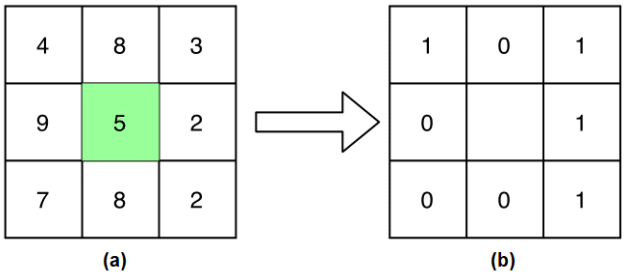


Figure 3.2: LBP descriptor

8 neighbouring pixels. The equation of LBP is given by (3.3) :

$$LBP(X_c, Y_c) = -\sum_{n=0}^{7} 2^n S(F_n - F(X_c, Y_c)) \tag{3.3}$$

Wherein $LBP(X_c, Y_c)$ is a LBP value at center pixel (X_c, Y_c). F_n: values of neighbour pixel. $F(X_c, Y_c)$ values of centre pixel. n: Index of neighbour pixel.

The function $S(x)$ where $x = (F_n - F(X_c, Y_c)$ can be expressed by the equation (3.4) :

$$S = \begin{cases} 1, & \text{if } x >= 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{3.4}$$

Then using a binary test, the center pixel LBP value is calculated, which is surrounded by 8 neighbours. The output of this binary test is fed into the 8-bit array, which is then converted to decimal, yielding a value of 71, as shown in Figure 3.3. The encoded frame comprises the LBP value of the 8×8 neighbourhood around the corresponding pixel in the original frame f, as shown in Figure 3.4. After encoding every pixel in the frame, an LBP-coded frame of the same size as the original frame is obtained.

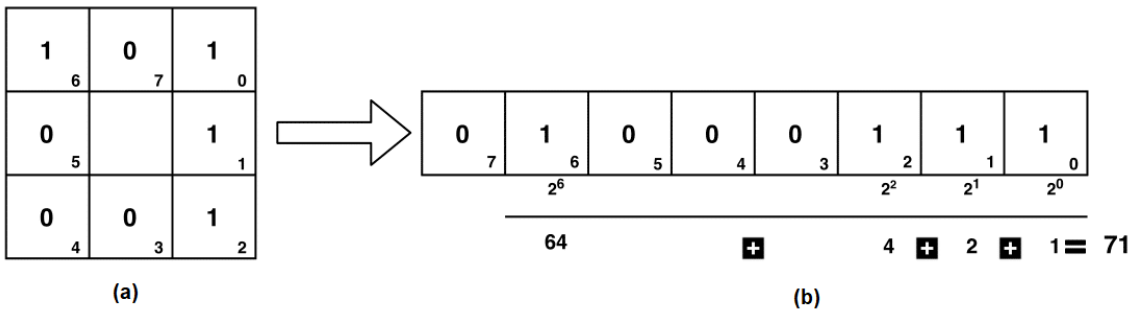


Figure 3.3: Center pixel calculation in LBP

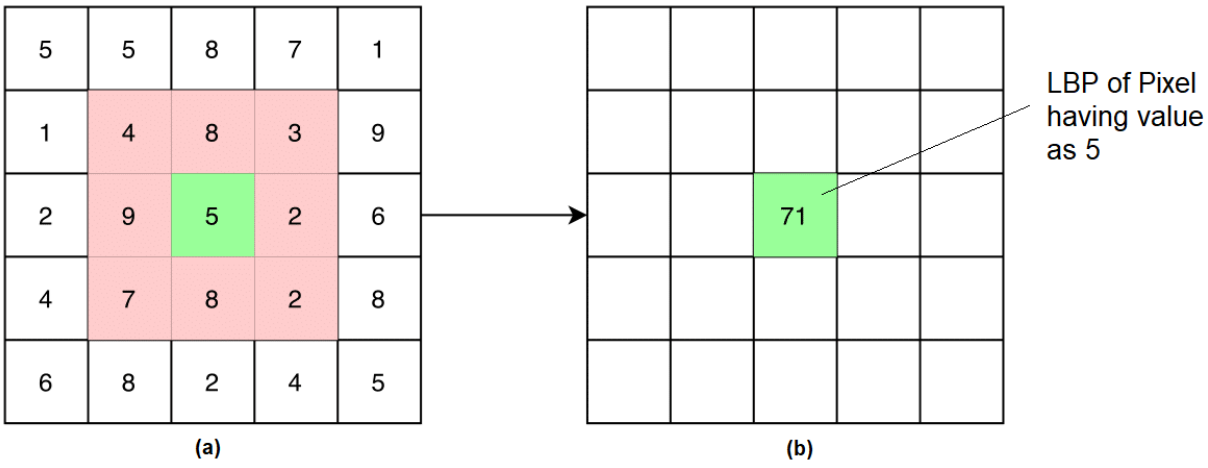


Figure 3.4: LBP encoded frame

3.2.2 Correlations Consistency

The correlation consistency is a similarity measure used in this stage to check the consistency of the video. In the previous section, LBP features from each frame of the input video are extracted. In this part, Pearson's correlation coefficient is employed to compute the correlation between each of the adjacent LBP coded frames. The video is considered as a sequence of LBP coded frames $F1, F2, \ldots Fn$ indexed at a time t, with a dimension of $m \times n$ pixels. . The Pearson's correlation coefficient ρ, is defined by equation (3.5). Where, $F^t(i,j)$ is the intensity of the $(i,j)^{th}$ pixel of LBP coded frame at time t, $F^{t+1}(i,j)$ is the intensity of the $(i,j)^{th}$ pixel of LBP coded frame at time $t+1$, $\bar{F}_t$ is the mean intensity of LBP coded frame at time t, and $\bar{F}_{t+1}$ is the mean intensity of LBP coded frame at time $t+1$

$$\rho = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(F^t(i,j)-\bar{F}_t)(F^{t+1}(i,j)-\bar{F}_{t+1})}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(F^t(i,j)-\bar{F}_t)^2}\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(F^{t+1}(i,j)-\bar{F}_{t+1})^2}} \tag{3.5}$$

In equation 3.5, m denotes the frame's width, n denotes the frame's height, t denotes time.

3.2.3 Outlier Detection

The outliers detection approach is used to investigate the forgeries in the video. The median (M) is a measure of central tendency that plays an important role in the outliers detection and has the advantage of being relatively unaffected by outliers (Leys et al., 2013). The outlier detection uses the Median Absolute Deviation (MAD), which examines the variability of a univariate sample of data. (Miller, 1991) proposed an outliner exclusion that can be adopted for MAD with a range of values, from 3 (best), 2.5 (good) to 2 (poor). A MAD scale of 3 is used in the proposed technique to examine the forgeries in the video. The *MAD* is defined as:

$$MAD = K \times MEDIAN(|A_i - MEDIAN(A)|) \tag{3.6}$$

Where A denotes the random variable vector consisting of N scalar observations In equation 3.6, K denotes the scaling factor, and it is defined by equation 3.7

$$K = \frac{-1}{\sqrt{2} \times erfcinv(3/2)} \tag{3.7}$$

Where, $erfcinv$ denotes the inverse complementary error function.

3.3 Experimental Analysis and Discussion

We discuss the experimental dataset, performance evaluation, and an overall comparison with the existing techniques in this section. The proposed technique is implemented on MATLAB 2020A using an i5 8^{th} Generation Processor, 16GB RAM, 1TB SSD, and 16GB GPU (GTX 1070Ti).

3.3.1 Dataset Description

A dataset with multiple video forgeries is created in order to perform the experiment with proposed technique. Dataset development commenced with the selection of videos from three available datasets: (Qadir et al., 2012 SULFA), (REWIND, 2013) and (VTL, 2018). The videos from these datasets are subjected to various tampering techniques such as splicing, frame deletion and insertion. Furthermore, some videos from the dataset have been subjected to post-processing operations such as illuminations. These videos have a format of MP4 and AVI coded by H.264 and Motion JPEG encoding standard, respectively, with different GOP lengths using FFmpeg software (FFmpeg, 2019). We termed it MFVD-4. A video with both static and moving backgrounds is included in this dataset. Description of sample tampered videos in the dataset is presented in Table 3.1

Algorithm 3.1 Proposed Technique

Read the input video V

Split the V into frames $f_1, f_2, f_3, f_4 \ldots\ldots f_n$

Count the total number of frames and stored it into $totalFrames$

for $j = 1$ to $totalFrames$ **do**

    Read the frame

    Resize the frame into 480×640

    Convert the RGB frame to Grayscale

    $LBPcodedframe$ = Extract LBP feature of grayframe

end for

Get the value of n by substracting 1 from total number of frames: $n = totalFrames - 1$

Initialize the ρ with 0: $\rho = zeros(1,n)$

for $i = 1$ to n **do**

    $f_1(x,y) =$ Read the $LBPcodedframe(i)$

    $f_2(x,y) =$ Read the $LBPcodedframe(i+1)$

    $F^t(x,y) = f_1(x,y) - \mu(f_1(x,y))$

    $F^{t+1}(x,y) = f_2(x,y) - \mu(f_2(x,y))$

    $\rho(1,i) = \frac{\sum\sum(F^t(x,y) \times F^{t+1}(x,y))}{\sqrt{\sum\sum(F^t(x,y) \times F^{t+1}(x,y))} \times \sqrt{\sum\sum(F^t(x,y) \times F^{t+1}(x,y))}}$

end for

$p = n - 1$

Initialize CD = 0 ▷ //CD: Correlation Differnce

for $i = 1$ to p **do**

    $CD(1,i) = \rho(1,i) - \rho(1,(i+1))$

end for

Apply outlier detection approach

Plot on CD

Table 3.1: Description of forged videos (SF: Splicing Forgery, IF: Insertion Forgery, DF: Deletion Forgery)

Test Videos	**Forgery Details**
Forged Video 1-5	SF:9&17; IF:31-40&111-120; DF:151-175
Forged Video 6-10	SF:14&35; IF:60-69&122-131; DF:275-299
Forged Video 11-15	SF:10&27; IF:41-50&115-124; DF:190-229
Forged Video 16-20	SF:11&22; IF:42-51&107-116; DF:144-168
Forged Video 21-25	SF:12&25; IF:51-60&113-122 DF:354-378
Forged Video 26-30	SF:14&23; IF:39-48&98-107; DF:220-259
Forged Video 31-35	SF:10&27; IF:41-50&115-124; DF:190-229
Forged Video 36-40	SF:11&22; IF:42-51&107-116; DF:144-168
Forged Video 41-45	SF:9&17; IF:31-40&111-120; DF:151-175
Forged Video 46-50	SF:14&35; IF:60-69&122-131; DF:275-299

3.3.2 Performance Evaluation

The proposed technique is successfully tested on an MFVD dataset-4, and performance parameters such as accuracy, recall, precision, and F1-Score are computed and illustrated in Figure 3.5. This figure shows that the proposed technique performs well and provides the accuracies of 99%, 100%, and 93% for the identification of splicing, frame insertion, and frame deletion forgeries, respectively. At the same time, accuracy achieves approximately 97.33% in the case of identification of multiple forgeries in the video.

3.3.2.1 Evaluation based on GOP Length

The dependency of the proposed technique against the changes in the GOP length is investigated in this section. Most of the existing techniques are tested on standard GOP sizes such as 10,12 and 16. Therefore we have also tested our technique on varied lengths of GOP, such as 10,12 and 16. The evaluation against the varied lengths of GOP is shown in Figure 3.6. From this Figure 3.6 it is demonstrated that the proposed technique performs better and provides the same accuracy of 97.33% for the identification of multiple forgeries, even when the GOP length of

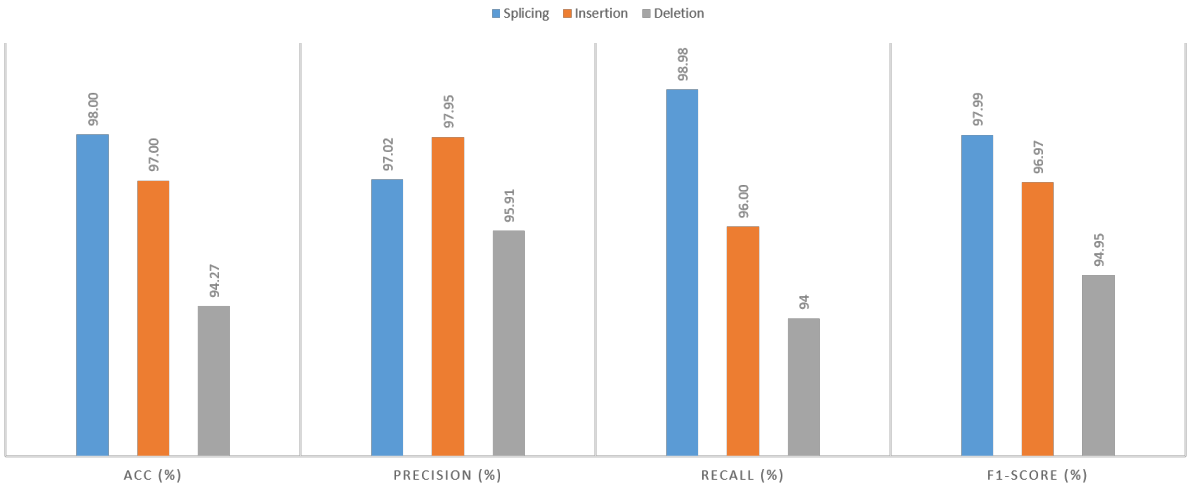


Figure 3.5: Performance of the proposed technique

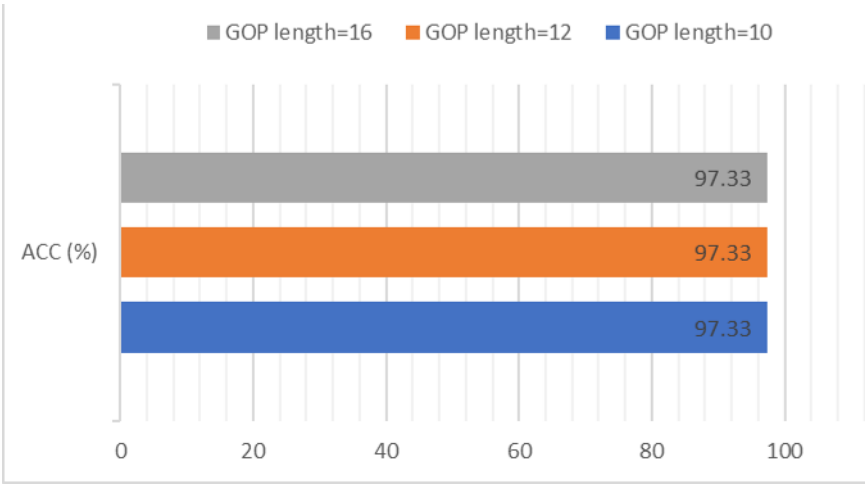


Figure 3.6: Evaluation against the GOP length

the videos is different. From this, it is clear that the proposed multiple forgeries identification technique does not depend on the different lengths of the GOP.

3.3.2.2 Evaluation based on Background

The dependency of the proposed technique against the changes in the video backgrounds is examined. The videos with the static and moving background are considered in order to evaluate the performance against the background dependency of the proposed technique, which is shown in Figure 3.7. From this figure it is inferred that the proposed forgery detection technique works well for forgeries detection from both static as well as moving background videos. Though, in the case of a moving background, the accuracy is affected slightly due to a fast-moving video

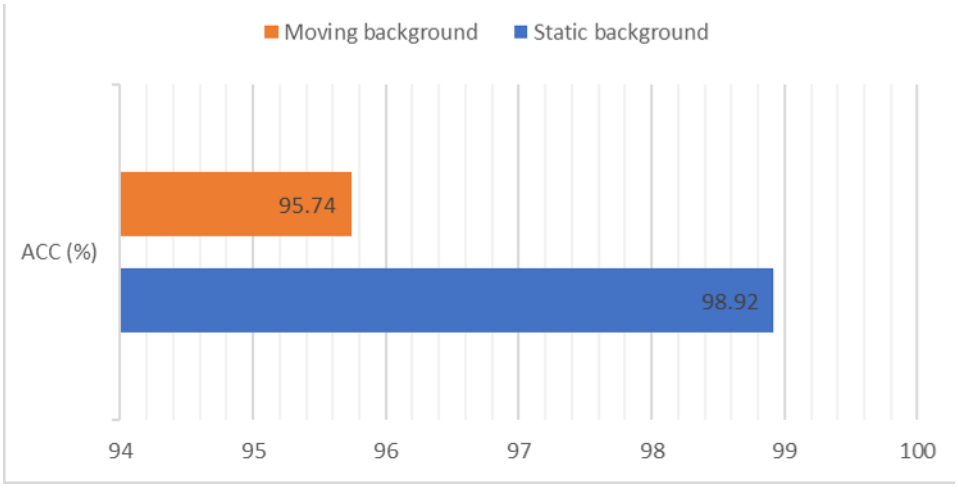


Figure 3.7: Evaluation against the video background

scene. In the case of moving background, the proposed technique achieves an accuracy of 95.74%. While in the case of static background videos, however, an accuracy of approximately 98.72% was achieved.

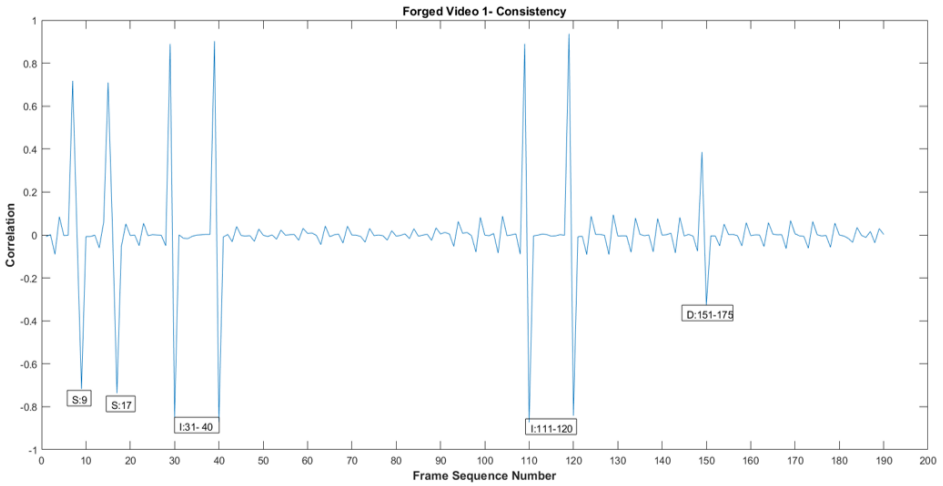
3.3.3 Experimental Results

In this part, we present the experimental results of the proposed technique on self-created sample multiple forged video (*i.e.,* for the videos 1,6,11,16 and 21). The experimental results are shown in Figure 3.8. One can see that the consistency among the LBP-coded adjacent frames is affected when the video is subjected to forgeries.

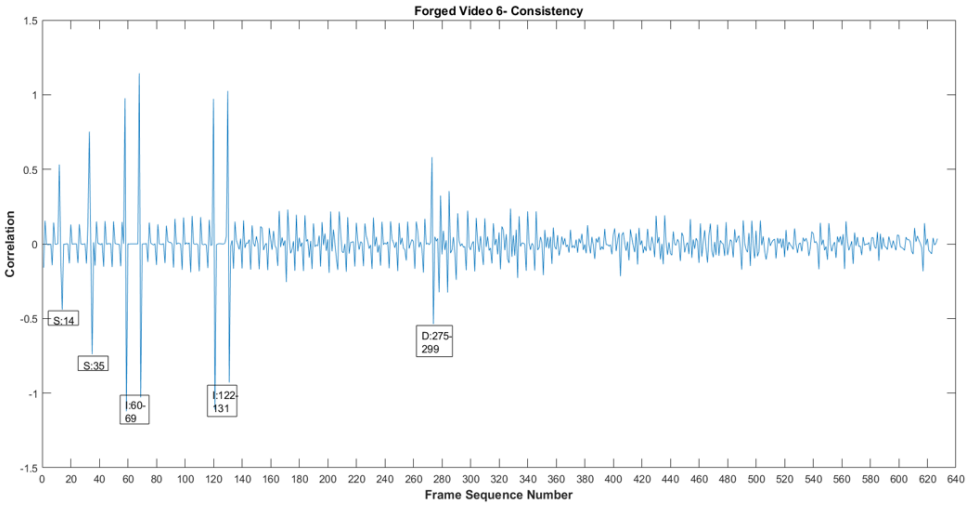
3.3.4 Overall Comparison

This comparative analysis section has selected the latest state-of-the-art, relevant techniques to compare our techniques. To the best of our knowledge, the current literature doesn't contain multiple forgery detection techniques. So we have compared our work with the state of the art techniques in (Wei et al., 2019) and (Bakas et al., 2019) that can detect the single forgery at a time. In the background, we have implemented and tested the existing techniques and our technique against the individual forgery on the multi-forged video dataset for a fair comparison.

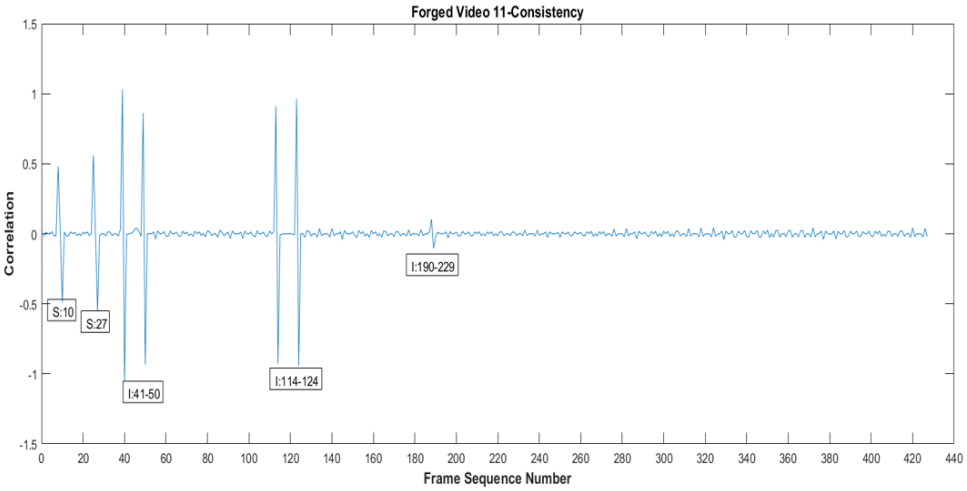
(a) Forged Video 1



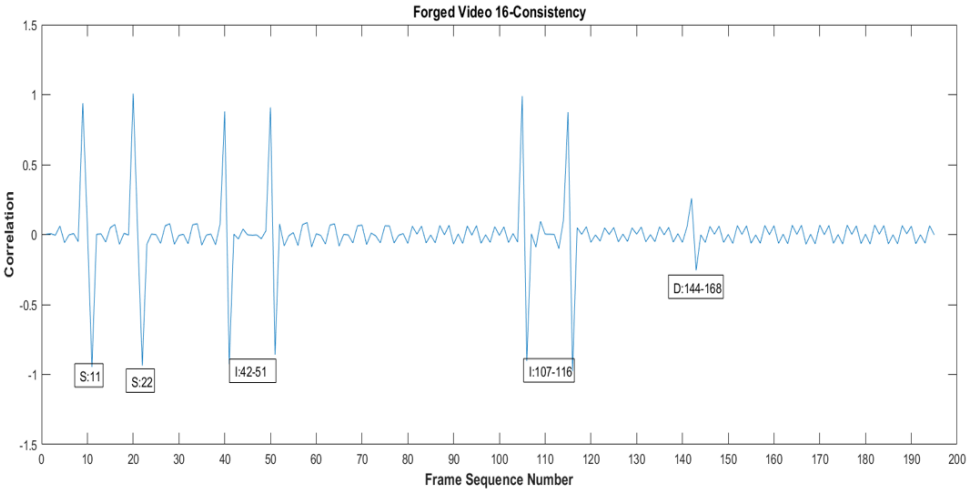
(b) Forged Video 6



(c) Forged Video 11



(d) Forged Video 16



(e) Forged Video 21

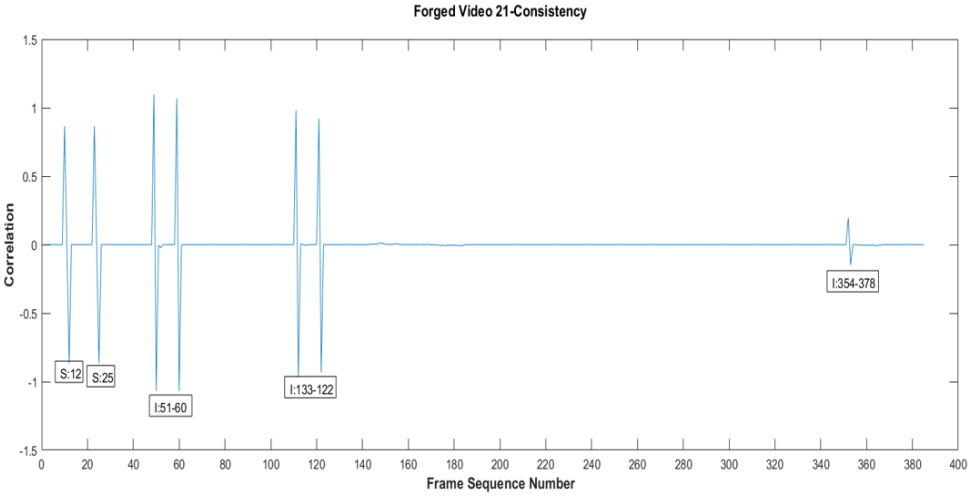


Figure 3.8: Experimental results on sample test videos

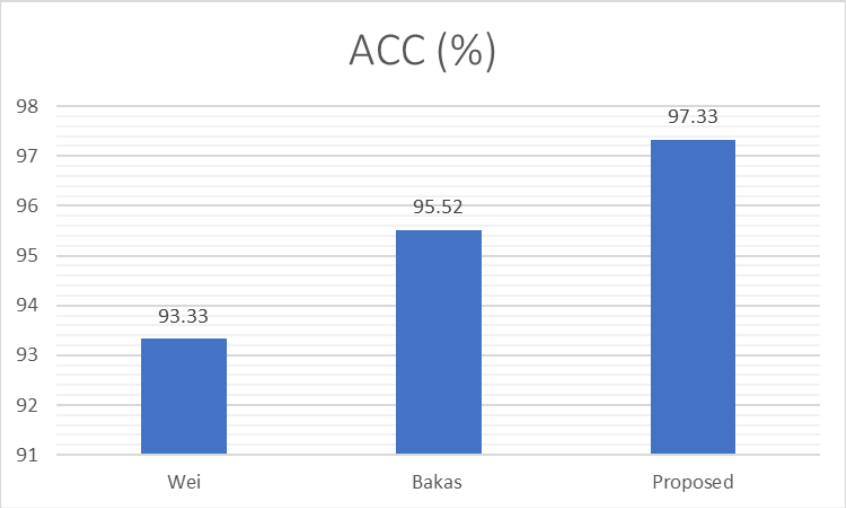


Figure 3.9: Proposed technique *vs.* existing techniques in terms of Accuracy

The comparison has been made to show the effectiveness of our techniques in terms of accuracy, multiple-forgery detection or not, execution time, GOP and background dependency. The proposed and existing techniques are evaluated on tampered videos taken from the MFVD-4 dataset. The comparison shows that the technique outperforms the existing techniques with respect to several different parameters. Figure 3.9 shows the proposed technique's performance in comparison to existing approaches. The experimental outcomes confirm that the proposed technique has high detection and localization accuracy regardless of GOP structure or video background. It is evident from Figure 3.9 that by comparing with existing techniques, our proposed technique achieves a better accuracy, *i.e.,* it provides 97.33%.

The complete comparison analysis in between proposed and existing techniques using several parameters such as forgeries identified, multiple forgeries detection and localization, GOP length dependency, background dependency, average execution time per video (in a second) is shown in Table 3.2. The existing techniques in (Bakas et al., 2019; Wei et al., 2019) are only capable of detecting the individual forgery. However, the proposed technique is able to examine the multiple forgeries in the video. Moreover, the technique in (Wei et al., 2019) is reliant on the changes in GOP length and background. However, our technique and the method in (Bakas et al., 2019) are entirely independent of changes in GOP length and video background. In addition, we measured and compared the average execution time for all three techniques mentioned. The LBP feature used in the proposed technique is computationally more efficient as compared to the features employed in the other two techniques. Therefore, it is observed that our technique

Table 3.2: Overall comparision (A: Forgeries identified, B: Multiple Forgeries Identification, C: GOP length, D: Background, E: Average Execution Time per video (in Sec), FIX: Fixed, VAR: Variable, SB: Static Background, MB: Moving Background, FS: Frame Splicing, FD: Frame Deletion, FDu: Frame Duplication)

Ref	A	B	C	D	E
(Wei et al., 2019)	FI, FD & FDu	NO	FIX	SB	179
(Bakas et al., 2019)	FI, FD & FDu	NO	FIX/VAR	SB/MB	40
Proposed Technique	FS, FI & FD	YES	FIX/VAR	SB/MB	17

is computationally effective as it requires less time than these two techniques.

3.4 Conclusion

In this chapter, a passive technique was proposed to identify the multiple forgeries in the individual video. The technique used LBP features and calculated the correlations between LBP-coded frames, followed by an outlier detection approach. This technique was able to detect frame splicing, frame insertion, and deletion forgeries. The technique was tested on the MFVD-4 dataset. The experimental results showed that accuracy could achieve 97.33%.

CHAPTER 4

Multiple Forgeries Detection Technique for Video based on Correlation Consistency between Entropy Coded Frames

In this Chapter, the passive technique based on the inter-frame correlation consistency between entropy-coded frames for video forgery detection is presented. In this Chapter, Section 4.1 discussed the background. The proposed technique's flow is described in Section 4.2. The experimental analysis and discussion are given in Section 4.3, and the Conclusion of the Chapter is discussed in Section 4.4.

Contents of the work presented in this Chapter have been published in *Multimedia System*, pp.1-14, 2020, Springer. (SCI Indexed)

4.1 Background

In this section, the entropy texture feature, Pearson correlation, and median absolute deviation are discussed. The texture is crucial for human visual perception and is used in a variety of image/video processing applications (Silva et al., 2018). The texture measures the variation of the intensity of pixels in the frames. The feature extraction selects the useful features/characteristics that are useful for further analysis in the video. Entropy is a mathematical measure of randomness that can describe the texture of an image. An image is considered as a bag of pixels, and the following equation (4.1) computes entropy for the image.

$$Entropy = -\sum_{i=1}^{n} p_i \times \log_2(p_i) \tag{4.1}$$

Where p_i value denotes the probability occurrence of pixels, which is calculated as

$$p_i = \frac{Number\ of\ occurrence\ of\ intensity\ level\ i\ in\ image}{Number\ of\ intensity\ levels} \tag{4.2}$$

The availability of a wide range of entropy variants prompted us to investigate the applicability preference of these variants for particular applications. The entropy texture features like DistrEn2D and MSE2D as adopted as a part of the feature extraction in this work.

The term correlation is commonly used to describe a relationship between two or more objects. The Pearson coefficient is a correlation coefficient describing the association among two variables measured on the likewise interval or ratio scale. It is used to measure the strength of the relationship between two variables. For linear relationships between two normally distributed variables, the Pearson correlation coefficient is most commonly used measure, often just called the correlation coefficient. The Pearson correlation coefficient is defined by the following equation,

$$\rho(X,Y) = \frac{E(X,Y)}{\sigma_X \sigma_Y} \tag{4.3}$$

Where $E(X,Y)$ denotes the cross-correlation between X and Y. σ_X and σ_Y denotes the variances of the X and Y.

The Median Absolute Deviation(MAD), recognized as a statistical methodology, is a robust analogue to a more commonly used outlier technique, which uses standard deviation from the mean. MAD uses variation from the median, which is less susceptible to distortion caused by outlying values. The MAD is a reliable measure of how evenly distributed a batch of data is. The standard deviation and variance are also measurements of spread, but they are more influenced by extremely low/high values, as well as non-normality. Outliers have a negligible effect on the median than they do on the mean; hence outliers have less impact on MAD. For univariate data $A_1, A_2, A_3 .... A_n$, The MAD formula is given by the following equation, Where $\bar{A} = median(A)$.

$$MAD = median(|A_i - \bar{A}|) \tag{4.4}$$

4.2 Proposed Technique

The implementation process to propose a technique is explained in this section. This technique is used for exposing the multiple forgeries from the video. The technique utilizes entropy texture features. When a video is modified, the consistency of entropy-coded frames gets affected. This inconsistency is utilized to identify the presence of forgeries. The main idea of our proposed technique depends on the concept of consecutive inter-frame correlation between the entropy-coded adjacent frame. This technique can identify the video forgeries (like splicing forgery, frame insertion, and frame deletion forgery).

There are other image texture features like Zernike moment and Haralick features used by the researchers (Y. Liu & Huang, 2017), and (Bakas et al., 2019) respectively. The tech-

nique proposed using Zernike moments feature is not robust to the GOP size and background dependency. Also, the computational cost is very high with the Zernike moments feature due to complex moments calculations. The high dimensionality of the matrix and the strong correlation among its features are the downsides of Haralick features. Also, both of these features are not robust against compression. To solve these problem, entropy texture features such as DistrEn2D (Azami et al., 2017) and MSE2D (Costa et al., 2002) is adopted in our work. With the use of these features, the proposed technique's robustness against the GOP length, video background, and compression attacks is improved.

The proposed technique comprises four stages 1) Pre-processing, 2) Texture feature extraction, 3) Correlation consistency, and 4) Outlier detection approach. First, the video is divided into frames. Frames are then transformed into grayscale frames to discard the irrelevant data as the RGB frame contains lots of data. With RGB frame to grayscale conversion, a significant amount of information is discarded, which is not required for further processing. In the second stage, entropy features of each frame are extracted. The inter-frame correlation among the entropy-coded frames is then computed in the third stage. An outlier detection method is applied in the last stage to detect and locate the forgeries. The proposed technique flow chart is given in Figure 4.1. The overall process of the proposed technique is described in the algorithm 4.1.

4.2.1 Pre-processing

The input video is separated into frames *i.e.,* $f_1, f_2, f_3 ..... f_n$ as part of the pre-processing procedure. Following that, the RGB frames are transformed to grayscale frames. The formula to convert the RGB to grayscale (citealpgunecs2016optimizing) for a frame F is given by the following equation (4.5)

$$F(i,j) = 0.2999 \times R(i,j) + 0.587 \times G(i,j) + 0.114 \times B(i,j) \tag{4.5}$$

Where $R(i,j)$, $G(i,j)$ and $B(i,j)$ indicates the red, green and blue components of the

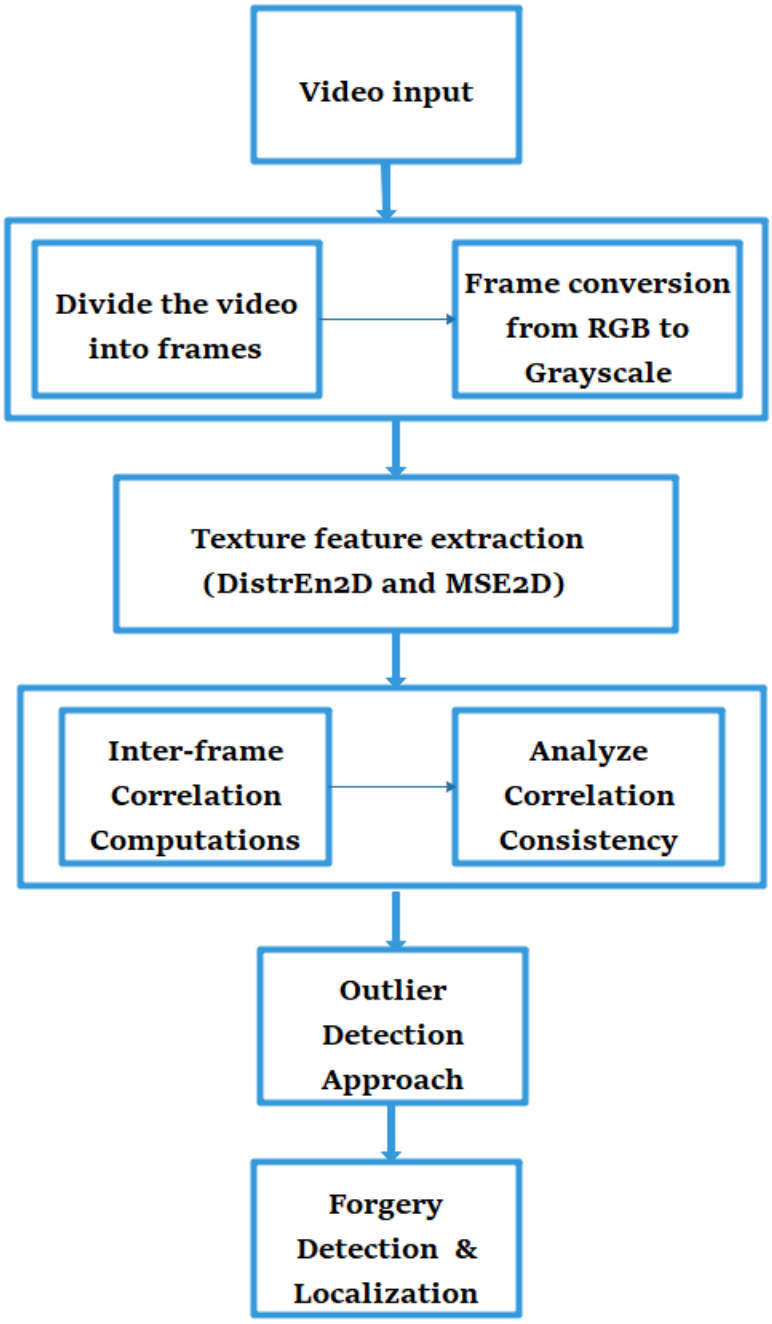


Figure 4.1: Flowchart of multiple video forgery detection and localization technique

frame, respectively.

4.2.2 Texture Feature Extraction

The texture features like DistrEn2D and MSE2D are adopted as a part of the feature extraction in this stage.

4.2.2.1 DistrEn2D Feature Extraction

In this part, the DistEn2D features of a pre-processed frame $F = (f_{i,j})_{j=1,2,...,h}^{i=1,2,...,w}$ is calculated. The steps for calculations is given below.

- Frame F pixels are normalized in the range 0 to 1 to make into a single scale.
- Compose 2-D matrices $(m_h \times m_w)$ $X_{k,l}^m$ $(k = 1,2,...,h-(m_h-1))$ and $(l = 1,2,...,w-(m_w-1))$ given by equation (4.6) where $m = [m_h, m_w]$ denotes the embedding vector.

$$X_{k,l}^m = \begin{bmatrix} f_{k,l} & f_{k,l+1} & \cdots & f_{k,l+(m_w-1)} \\ f_{k+1,l} & f_{k+1,l+1} & \cdots & f_{k+1,l+(m_w-1)} \\ f_{k+2,l} & f_{k+2,l+1} & \cdots & f_{k+2,l+(m_w-1)} \\ \vdots & \vdots & \vdots & \vdots \\ f_{k+(m_h-1),l} & f_{k+(m_h-1),l+1} & \cdots & f_{k+(m_h-1),l+(m_w-1)} \end{bmatrix} \tag{4.6}$$

- Calculate the distance matrix $D = (d_{k,l})_{k=1,2,...,h-(m_h-1)}^{l=1,2,...,w-(m_w-1)}$ as the largest element of the absolute difference between $X_{k,l}^m$ and $X_{a,b}^m$. wherein a vary from 1 to $h-(m_h-1)$ and b vary from 1 to $w-(m_w-1)$.
- The empirical probability density function of matrix D is computed using the histogram approach with M bins. Where $p_{t(t=1,2,3...,M)}$ represent the probability of each bin.
- DistEn2D is computed using equation (4.7)

$$DistEn2D(F,m,M) = -\sum_{t=1}^{M} p_t \times \log_2(p_t) \tag{4.7}$$

4.2.2.2 MSE2D Feature Extraction

One more entropy-based feature, such as MSE2D, is also utilized in this technique to detect video forgeries. Multiscale Entropy (MSE) feature is used by (Costa et al., 2002) for the estimation of time series uncertainty *i.e.,* fluctuations in time-series signal is identified. MSE works for both 1-dimensional data *i.e.,* MSE1D and 2-dimensional data *i.e.* MSE2D (Silva et al., 2018). In our work, the data is in the form of 2D images, so we use MSE2D. MSE2D is a modified form of MSE1D, and is derived from the MSE1D equation, which is given below.

$$MSE1D(x,t,m,r) = -\log \frac{A_t^m(r)}{B_t^m(r)} \tag{4.8}$$

Where $A_t^m(r)$ and $B_t^m(r)$ are measured from the coarse-grained time-series using scale factor t.

The following two steps are required to calculate the MSE2D.

1. For 2-D arbitrary video frame F, the coarse-grained frame is computed by the equation (4.9)

$$Y_{i,j}^t = \frac{1}{t^2} \sum_{k=(i-1)t+1;l=(j-1)t+1}^{k=it;l=jt} F_{k,l} \tag{4.9}$$

where $1 \leq i \leq \frac{h}{t}$ and $1 \leq j \leq \frac{w}{t}$, $Y_{i,j}^t$ is a coarse-grained of given frame at scale t. Where h and w denote the height and width of the frame.

2. Sample entropy for each coarse-graining frame with a certain threshold r is computed by equation (4.10)

$$SampEn2D(F,m,r) = -log(\frac{U^{m+1}(r)}{U^m(r)}) \tag{4.10}$$

$$U^m(r) == \frac{1}{N_m} \sum_{i=1;j=1}^{i=h-m;j=w-m} U_{i,j}^m(r) \tag{4.11}$$

$$U^{m+1}(r) == \frac{1}{N_m} \sum_{i=1;j=1}^{i=h-m;j=w-m} U_{i,j}^{m+1}(r) \tag{4.12}$$

$$U_{i,j}^m(r) == \frac{\#\ of\ x_m(a,b) | d[x_m(i,j), x_m(a,b)] \leq r}{N_m - 1} \tag{4.13}$$

$$U_{i,j}^{m+1}(r) == \frac{\#\ of\ x_{m+1}(a,b) | d[x_{m+1}(i,j), x_{m+1}(a,b)] \leq r}{N_m - 1} \tag{4.14}$$

where $x_m(i,j)$ is the set of pixels of m-length square window having origin at $U(i,j)$: changing from j to $j+m-1$ and i to $i+m-1$. N_m is the total sum of square window within frame F that can produce for both m and $m+1$ size, $N_m = (w-m) \times (h-m)$. Wherein, distance d is computed by equation (4.15)

$$d[x_m(i,j), x_m(a,b)] = max(\quad \text{mod}\ [F(i+k, j+l) - F(a+k, b+l)]) \tag{4.15}$$

where k and l vary from 0 to $m-1$

Finally, MSE2D with a scale factor t is calculated by equation (4.16)

$$MSE2D(F,t,m,r) = SampEn2D(Y^t, m, r) \tag{4.16}$$

4.2.3 Correlation Consistency

Correlation consistency is one of the similarity measures used in this work for examining the correlation between the adjacent frames in the video. In the previous section, entropy features of each video frame are obtained. In this part, the Pearson's correlation coefficient $PCC(\rho)$ is

used to calculate the inter-frame correlation between each of the adjacent Entropy coded frames. The video is considered as a sequence of Entropy coded frames $F1, F2 \ldots Fn$. The $PCC(\rho)$ is defined by equation (4.17)

$$PCC(\rho) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(F^{t}(i,j) - \bar{F}_t)(F^{t+1}(i,j) - \bar{F}_{t+1})}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(F^{t}(i,j) - \bar{F}_t)^2}\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(F^{t+1}(i,j) - \bar{F}_{t+1})^2}} \tag{4.17}$$

Where in equation 4.17, $F^{t}(i,j)$ is the entropy coded frame at time t, $F^{t+1}(i,j)$ is the entropy coded frame at time $t+1$, $\bar{F}_t$ is the Mean intensity of $F^{t}(i,j)$ frame. $\bar{F}_{t+1}$ is the Mean intensity of $F^{t+1}(i,j)$ frame. m is the width of the frame and n is the height of the frame.

4.2.4 Outlier Detection

In this stage, the outliers detection approach is used to investigate the forgeries in the video. An outlier is a sample that differs so significantly from the other samples that it raises the possibility that a unique mechanism has produced it. The median (M) is a measure of central tendency that plays an important role in the outliers detection and has the advantage of being relatively unaffected by outliers (Leys et al., 2013). The outlier detection uses the scaled Median Absolute Deviation (MAD), which measures the variability of a univariate sample of data. A forgery is examined when a forged point element is greater than three scaled MAD aside from the M. The scaled MAD is given by equation (4.18), where random vector A is formed by N scalar observation.

$$MAD = K \times MEDIAN(|A_i - MEDIAN(A)|) \tag{4.18}$$

$$K = \frac{-1}{\sqrt{2} \times erfcinv(3/2)} \tag{4.19}$$

The K denotes scaling factor, given by equation (4.19) where erfcinv represents an inverse complementary error function.

Algorithm 4.1 Proposed Technique

Input: Digital Video

Output: Forgeries identification

1: Read the input video V

2: Split the video V into frames $f_1, f_2, f_3, f_4 \ldots\ldots f_n$

3: Count the total number of frames and stored it into $totalFrames$

4: **for** $j = 1$ to $totalFrames$ **do**

5: Read the frame

6: Resize the frame into 480×640

7: Convert the RGB frame to $grayscaleframe$

8: Extract entropy feature from each frame and stored it into $ent\_codedframe$

9: **end for**

10: $n = totalFrames - 1$

11: Initialize the PCC with 0

12: **for** $i = 1$ to n **do**

13: $f_1(x,y) =$ Read the $ent\_codedframe(i)$

14: $f_2(x,y) =$ Read the $ent\_codedframe(i+1)$

15: $F^t(x,y) = f_1(x,y) - \mu(f_1(x,y))$

16: $F^{t+1}(x,y) = f_2(x,y) - \mu(f_2(x,y))$

17: $PCC(1,i) = \frac{\sum\sum(F^t(x,y) \times F^{t+1}(x,y))}{\sqrt{\sum\sum(F^t(x,y) \times F^{t+1}(x,y))} \times \sqrt{\sum\sum(F^t(x,y) \times F^{t+1}(x,y))}}$

18: **end for**

19: $p = n - 1$

20: Initialize $corDiff = 0$

21: **for** $i = 1$ to p **do**

22: Calculate the correlation difference: $corDiff(1,i) = PCC(1,i) - PCC(1,(i+1))$

23: **end for**

24: Apply outlier detection approach

25: Plot on $corDiff$

4.3 Experimental Analysis and Discussion

In this part, we discuss the experimental dataset to test the proposed technique and experimental results followed by an overall analysis. The proposed technique is performed on MATLAB 2019A using an i5 (8th generation processor), 8GB RAM (DDR4), 128Gb SSD, and 4GB GPU (GTX 1050Ti) with 768 CUDA cores.

4.3.1 Dataset Description

To evaluate the performance of the proposed technique, a multi-forged video dataset has been created. Dataset development commenced with the selection of videos from the available datasets: SULFA dataset (Qadir et al., 2012), (REWIND, 2013) and (VTD, 2017). The selected videos are exposed to tamperings such as frame splicing, frame insertion, deletion, and duplication. We termed it MFVD-2, which includes 30 forged and 30 original videos with static as well as moving backgrounds. Videos are in MP4 and AVI format coded using H.264 and Motion JPEG encoding standard with fixed and variable (*i.e.,* 10, 12, and 16) GOP size by FFmpeg software (FFmpeg, 2019). Figure 4.2 depicts a snapshot of forged video from the dataset. The information of forged videos is presented in Table 4.1.

Figure 4.2: Snapshot of forged videos

Table 4.1: Details of forged videos

Videos	**Format**	**Length (sec)**	**Resolution**	**Forgery**
FV 1	MP4	6	640 × 480	
FV 2	MP4	9	640 × 480	
FV 3	MP4	9	640 × 480	
FV 4	MP4	9	640 × 480	
FV 5	MP4	18	640 × 480	
FV 6	AVI	14	320 × 240	
FV 7	AVI	20	320 × 240	FS:15; FI:30-40;
FV 8	AVI	19	320 × 240	FDu:(25,29)&(125,139);FD:157-
FV 9	AVI	14	320 × 240	186;
FV 10	AVI	9	320 × 240	
FV 11	AVI	10	320 × 240	
FV 12	AVI	11	320 × 240	
FV 13	AVI	10	320 × 240	
FV 14	AVI	11	320 × 240	
FV 15	AVI	11	320 × 240	

FV 16	MP4	9	640×480	
FV 17	MP4	7	640×480	
FV 18	MP4	9	640×480	
FV 19	MP4	9	640×480	
FV 20	MP4	7	640×480	
FV 21	AVI	12	640×480	
FV 22	AVI	9	640×480	FS:10;FI:50-60;
FV 23	AVI	8	640×480	FDu:(35,42)&(118,127);FD:146-
FV 24	AVI	17	640×480	175
FV 25	AVI	6	640×480	
FV 26	MP4	11	640×480	
FV 27	MP4	18	640×480	
FV 28	MP4	7	640×480	
FV 29	MP4	14	640×480	
FV 30	MP4	6	640×480	

4.3.2 Visualization

This section visualizes the proposed technique experimentation using both entropy features such as DistrEn2D and MSE2D. For that, both successful and failure cases are considered, which is displayed in Figure 4.3 and Figure 4.4. To test the performance of the proposed technique, sample forged videos such as 1, 13, 7, and 28 are selected randomly from the MFVD-2 dataset for visualization.

The successful cases are shown in Figure 4.3 (a) and Figure 4.4 (a). From these figures, it shows that the proposed forgery detection technique can identify the intra-frame splicing forgery as well as inter-frame forgeries such as insertion, duplication, and deletion in the forged video 1 using DistrEn2D and forged video 7 using MSE2D. The splicing forgery is detected at the 15^{th} frame and inserted frames at frame locations 30 to 40. Frame duplication is observed at 25 and 139, and frame deletion is exposed after 156^{th} frame, where 30 frames are deleted.

The failure cases are shown in Figure 4.3 (b) and Figure 4.4 (b). From Figure 4.3 (b), it is observed that the proposed forgery detection technique using the DistrEn2D is not able to

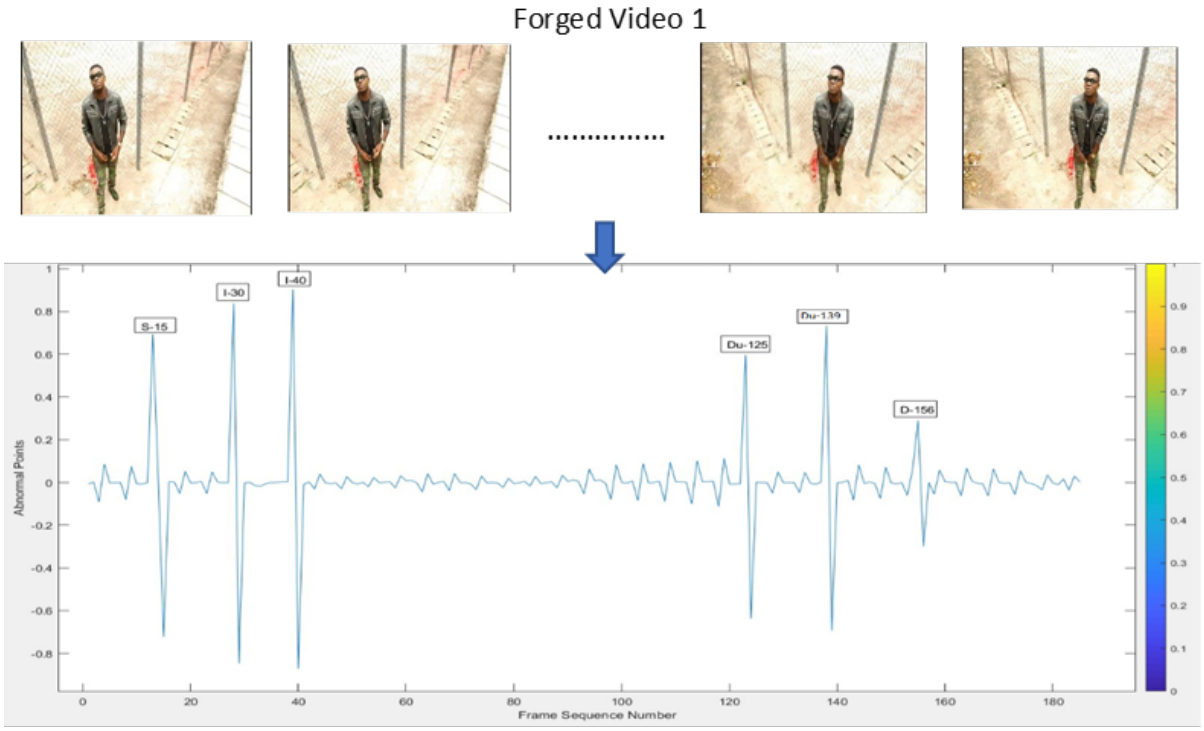
Table 4.2: Performance evaluation

	DistrEn2D				MSE2D			
Forgery	**PR**	**RR**	**F1 Score**	**ACC**	**PR**	**RR**	**F1 Score**	**ACC**
Insertion	100	100	100	100	100	100	100	100
Splicing	98.33	97.87	98.09	96.66	98.33	97.87	98.09	96.66
Duplication	100	100	100	100	98.33	97.87	98.09	96.66
Deletion	97.33	96.85	97.09	93.33	97.33	96.85	97.09	93.33
Multiple	98.91	98.68	98.79	97.49	98.49	98.14	98.31	96.66

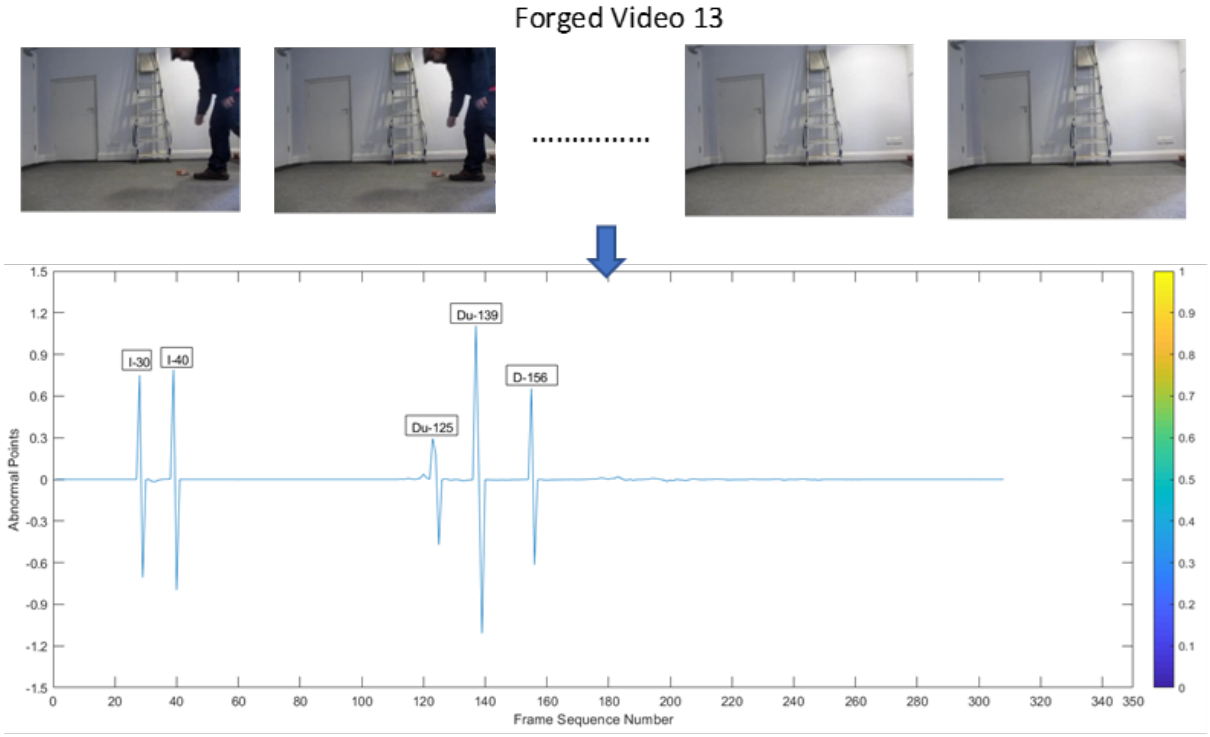
recognize the frame splicing present in the forged video 13. However, it can detect the frame insertion in between 30 to 40, frame deletion after 156^{th} frame, and duplicated frame at 125 and 139. In the failure case shown in Figure 4.4 (b), it is analyzed that the proposed technique using the MSE2D feature can expose forgery from the forged video 28, wherein the frame splicing is detected at the 10^{th} frame and inserted frame in between 50 to 60. However, it cannot reveal the deleted and duplicated frames due to less consistency among entropy-coded frames.

4.3.3 Performance Evaluation

The proposed technique performance is evaluated on videos taken from the MFVD dataset-2. The performance evaluation measure such as accuracy, precision, recall, and F1-score, using DistrEn2D and MSE2D features are shown in as mentioned above Table 4.2. This table shows that the forgery detection technique using DistrEn2D and MSE2D can handle the forgeries in the individual videos with good performance. For cases of multiple forgeries identification using DistrEn2D feature, the proposed technique provides PR of 98.91%, RR of 98.68%, F1-Score of 98.79%, and accuracy of 97.49%. Whereas with the use of the MSE2D feature, the proposed technique for detecting the multiple forgeries provides 98.49% PR, 98.14% RR, and 98.31% F1 Scores, and accuracy reaches up to 96.66 %. However, in the case of overall accuracy with multiple forgeries is concerned, the proposed technique with the DistrEn2D gives comparatively better accuracy than the MSE2D.

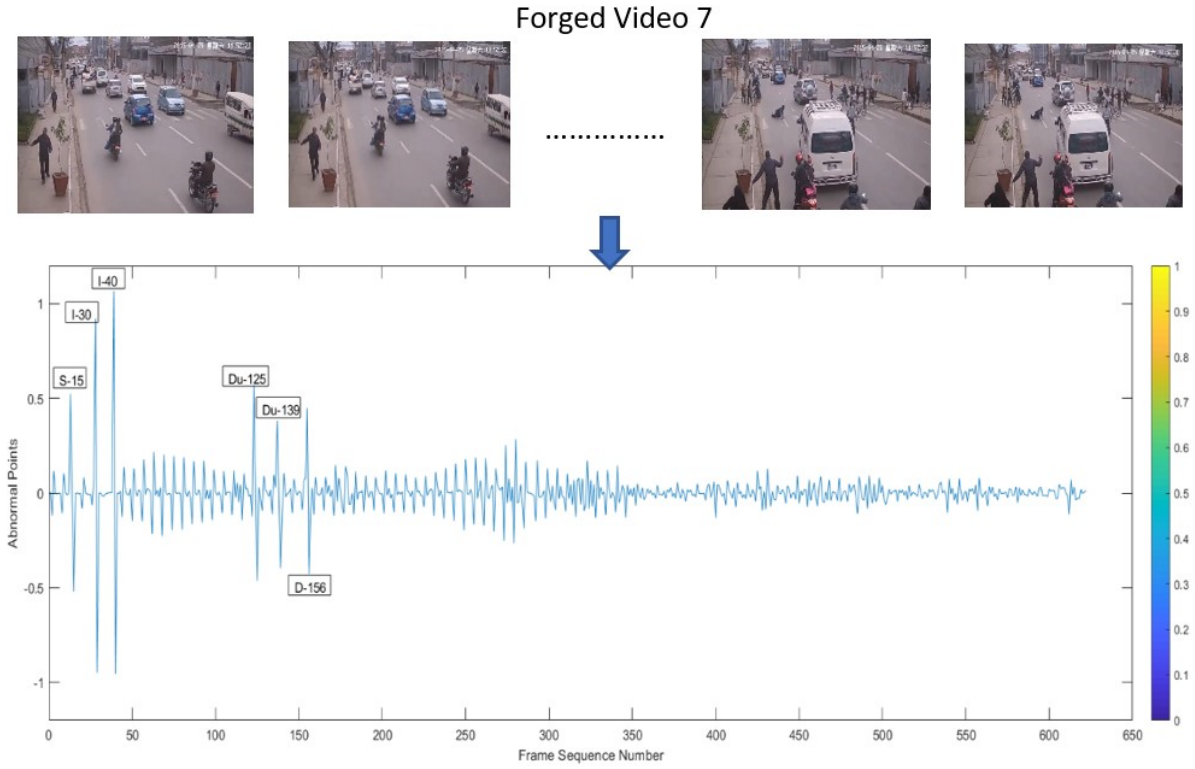


(a)

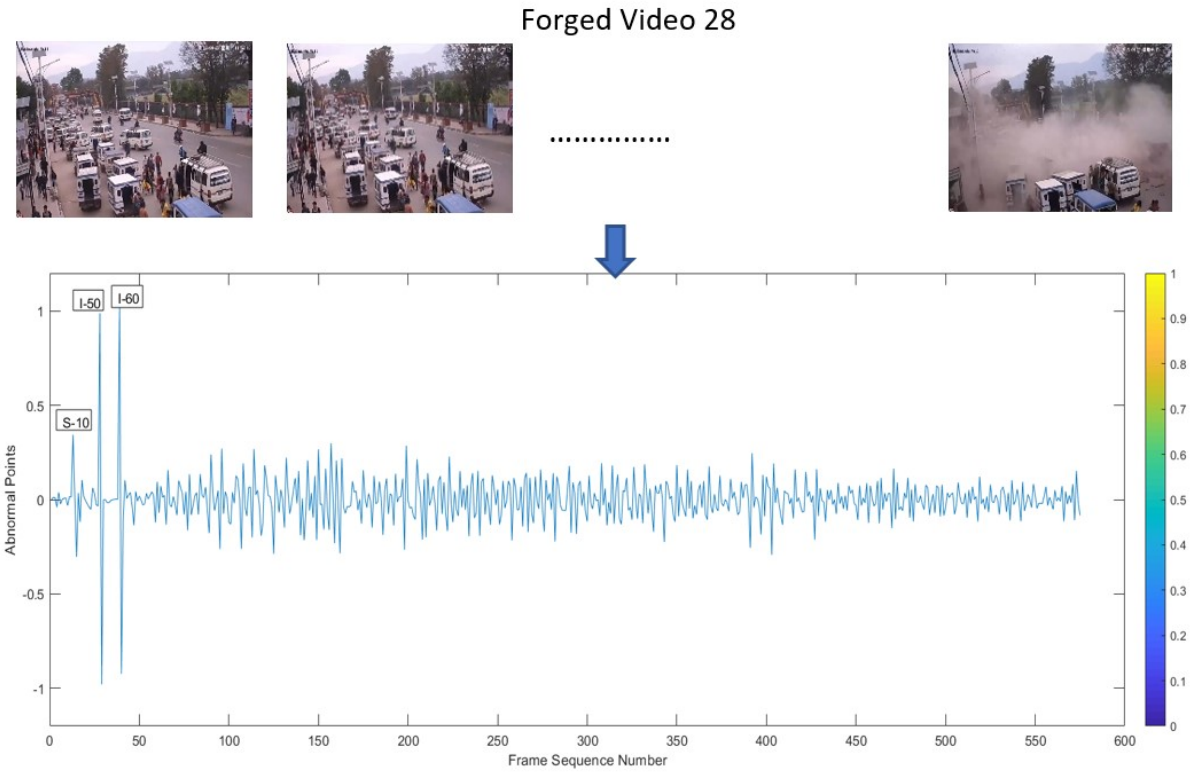


(b)

Figure 4.3: Visualization: DistrEn2D a) Successful b) Failure



(a)



(b)

Figure 4.4: Visualization: MSE2D a) Successful b) Failure

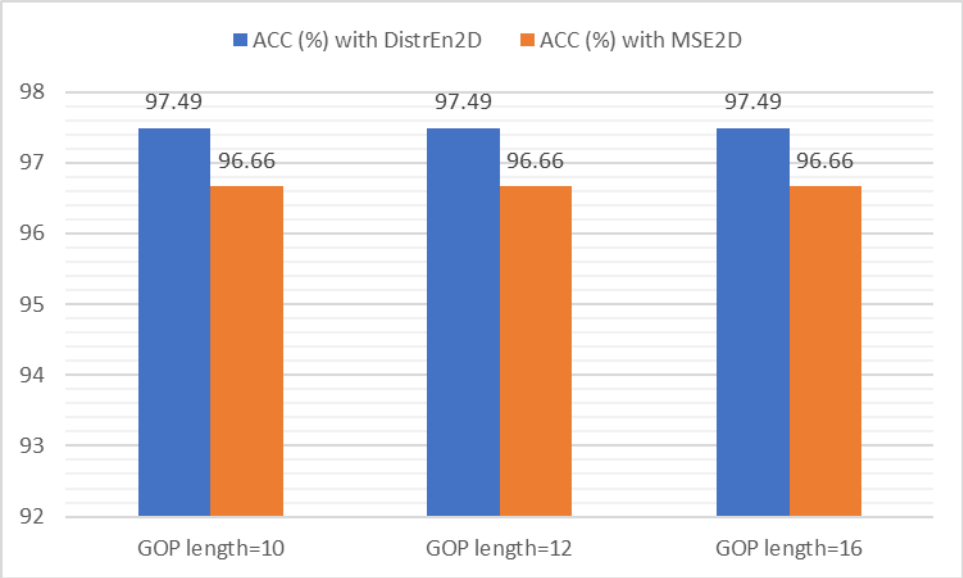


Figure 4.5: Performance against the varied GOP length videos

4.3.3.1 GOP Dependancy

The GOP length dependency of the proposed technique is analyzed in this section. Our technique is fully based on investigating the correlation among the entropy-coded neighboring frames of video, and it cannot use any artifacts such as prediction or residual error produced based on GOP. Due to this reason, the proposed technique performance is not affected by the changes in GOP length. To evaluate the technique, videos with varied lengths of GOP, such as 10,12, and 16, are considered. The evaluation against the GOP is shown in Figure 4.5. This figure shows that the proposed technique works well against the varied lengths of GOP videos and is fully independent of GOP length.

4.3.3.2 Video Background

We tested the effectiveness of our technique with varied video backgrounds, *i.e.,* static/moving. Figure 4.6 shows the accuracy of the technique for the videos in both categories. From this figure, it is observed that the proposed forgery detection technique is able to identify forgeries in static as well as moving video backgrounds. However, in the case of moving video background, the accuracy is marginally affected due to a sudden observed change between the consistency of extracted feature of frames in the video sequence.

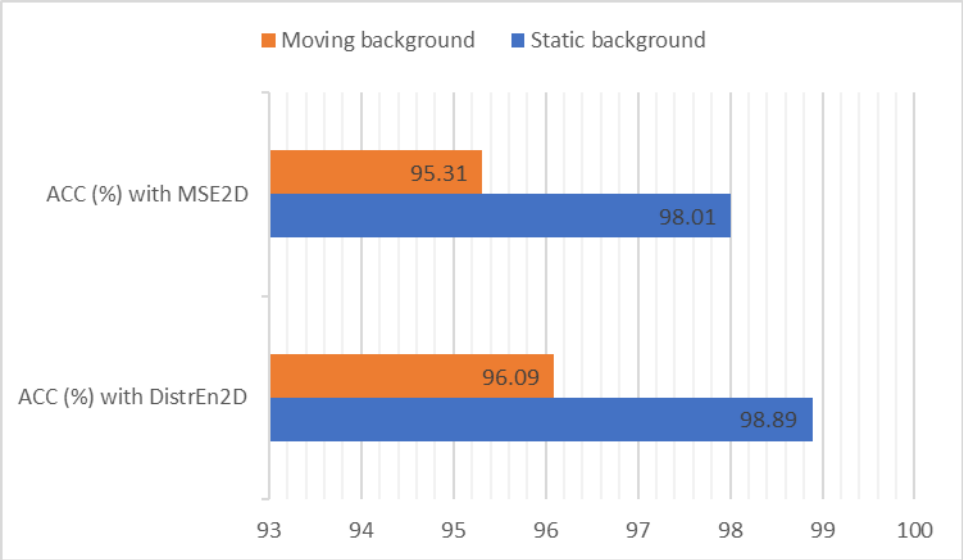


Figure 4.6: Performance against the static/moving background videos

4.3.3.3 Compression Attack

In this section, the robustness of the suggested technique against compression attacks is evaluated. The forged videos are compressed using different compression quality factors (CQF = 1, 5, 15, and 20), wherein CQF = 1 being the best quality. Accuracy (ACC) is estimated to prove the technique's robustness (using both DistrEn2D & MSE2D) against the compression. ACC values against the compression quality factors are shown in Figure 4.7. The proposed technique with the DistrEn2D feature gives 95.21% ACC even for the worst situation (CQF = 20), whereas ACC values become 97.49% for the CQF=1. The proposed technique with the MSE2D feature gives 94.5% ACC even for the worst situation (CQF = 20), whereas ACC values become 96.66% for the CQF=1. The results reveal that the proposed forgery detection technique is robust to compression.

4.3.4 Comparison with Existing Techniques

In the comparative analysis section, we have selected the only latest state-of-the-art, relevant techniques to compare with the proposed techniques. To the best of our knowledge, the current literature doesn't contain multiple forgery detection techniques. So we have compared our work with the state of the art techniques (J. Yang et al., 2016), (Wei et al., 2019), and (S. Fadl

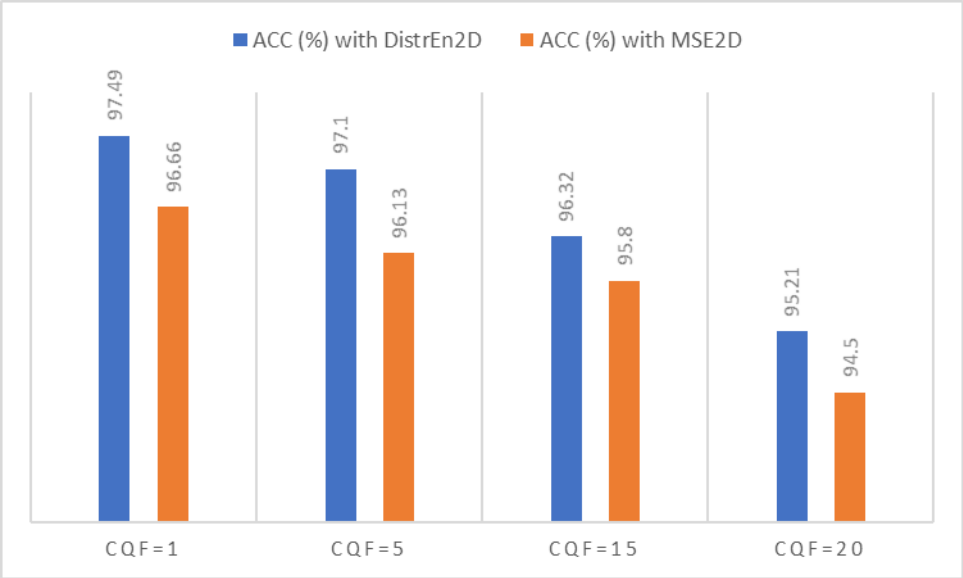


Figure 4.7: Performance against the compression

et al., 2020) that can detect the single forgery at a time. In the background, we have tested the proposed and existing techniques against the single forgery detection on the multi-forged video dataset MFVD-2 for a fair comparison. The comparison has been made to show the effectiveness of our techniques in terms of accuracy, multiple-forgery detection or not, GOP and background dependency and robustness against the compression. We tested and compared the performance of the technique using the DistrEn2D feature and MSE2D feature with some of the existing mentioned techniques. The comparison shows that the technique outperforms the existing techniques with respect to several different parameters. The comparative performance is shown in Figure 4.8. It is evident that our techniques forgery detection performance is significantly higher compared to the existing techniques. The proposed technique using the DistrEn2D feature provides an accuracy of 97.49%, whereas using the MSE2D feature, it achieves 96.66% accuracy. This comparison also illustrates that a technique using the MSE2D feature provides a slightly lower accuracy than one with the DistrEn2D feature.

An overall comparative analysis using several parameters such as forgeries identified, multiple forgeries identification, GOP length, and background dependency, and execution time are presented in Table 4.3, Wherein (A: Multiple forgeries handled B: GOP length, C: Background D: Compression Robustness, E: Execution time (in Sec), FIX: Fixed, VAR: Variable, SB: Static background, MB: Moving background). The existing techniques such as (S. Fadl et al., 2020; Wei et al., 2019; J. Yang et al., 2016) can only handle one type of forgery from the individual

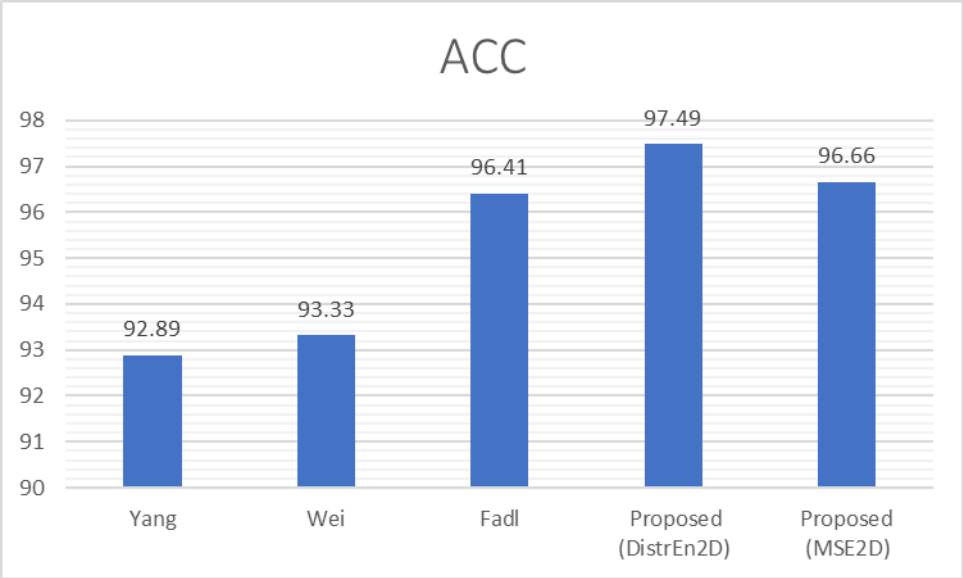


Figure 4.8: Comparison with the existing techniques

video. These techniques are not suitable for videos with different GOP lengths and are not robust against compression attacks. On the other hand, our technique can detect the multiple forgeries present in the video. It is robust to the compression attack, and it can work for videos having different GOP lengths. The techniques in (Wei et al., 2019; J. Yang et al., 2016) are not suitable for the moving background. However, the proposed technique and technique in (S. Fadl et al., 2020) are independent of the video background, *i.e.,* works for both static as well as moving background videos. The average execution times for the proposed and existing techniques are also recorded. In comparison to the features employed in existing techniques, the entropy features used in the proposed technique is computationally efficient. It is evident from Table 4.3 the proposed technique using both DistrEn2D and MSE2D features is computationally efficient compared to existing techniques. However, the proposed technique using the MSE2D feature is comparatively computationally effective than the DistrEn2D feature.

Table 4.3: Overall Comparison

Ref	Forgery	A	B	C	D	E
(J. Yang et al., 2016)	Du	NO	FIX	SB	NO	40
(Wei et al., 2019)	FI, FDu & FD	NO	FIX	SB	NO	188
(S. Fadl et al., 2020)	FI, FDu & FD	NO	FIX	SB/MB	NO	74
Proposed (DistrEn2D)	FS, FI, FDu & FD	YES	FIX/VAR	SB/MB	YES	15.42
Proposed (MSE2D)	FS, FI, FDu & FD	YES	FIX/VAR	SB/MB	YES	14.07

4.4 Conclusion

In this Chapter, the forgery detection technique was proposed using entropy features such as DistrEn2D and MSE2D. The correlation consistency and outlier detection approach was applied after the extraction of features. This technique was able to examine the presence of forgeries such as intra-frame forgery, *i.e.,* splicing and inter-frame forgeries, *i.e.,* frame insertion, deletion, and duplication from the digital videos. The technique was tested on an MFVD-2 dataset. The experimental result provided an accuracy of 97.49% and 96.66% with DistrEn2D and MSE2D features, respectively. The experimental outcomes also revealed that the proposed technique performed better when compared with the relevant existing techniques. The advantage of this technique was its robustness against the compression attack, GOP length, and video background. However, it is not robust against the different types of noise when added in the video.

CHAPTER 5

Multiple Forgery Detection Technique for Digital Video using VGG-16 Neural Network and and KPCA

-

This Chapter proposes the multiple video forgery detection technique based on the VGG-16 deep neural network and and KPCA. Section 5.1 discusses the background of the VGG-16 deep neural network and KPCA. The flow of the proposed technique is outlined in Section 5.2. The experimental analysis and discussion are given in Section 5.3, Finally, the Conclusion is discussed in Section 5.4.

5.1 Background

In this section, VGG-16 and PCA are discussed, which are used in the proposed forgery detection technique. VGG-16, a deep learning architecture, has recently attracted a lot of interest in real-world applications. The model is trained on millions of images taken from the ImageNet Dataset. As a result, the network with pre-trained weights has picked up rich feature representations for a variety of images. VGG16 is based on the Convolutional Neural Network Model (CNN) (Simonyan & Zisserman, 2014). CNN is composed of input, many hidden layers and the final output layer. The convolution, pooling and fully connected layers are the part of hidden layers. It filters input volumes to higher levels of abstraction using several convolution layers. By adding pooling layers for limited translation and rotation invariance, CNNs increase their detection capabilities for strangely situated objects. The pooling layer in CNN also permits the use of more convolutional layers by lowering memory consumption. Normalization layers are used to standardize part of input regions by shifting all inputs in a layer towards a mean equal to zero and variance equal to one. Fully connected layers contain neurons that function similarly to convolutional layers but differ in that they are linked to all activations in the preceding layer.

Principal component analysis (PCA) is most common dimension reduction approach that uses unsupervised learning (Martinez and Kak (2001)). Dimensionality reduction is the process of transforming raw from a higher dimension to a lower dimension while retaining some substantial elements of the original data, ideally near to its intrinsic dimension. PCA, the most used linear technique for dimensionality reduction, produces a linear mapping of the data to a low dimensional vector space with the goal of maximizing the data's variance in the low dimension representation. In order to select the reduced dimension features from the given data, The data's covariance matrix is generated, and the matrix's eigenvectors are determined. The eigenvectors that correspond to the highest eigenvalues, which are called the principal components, are obtained. Kernel PCA is the version of PCA used as a non-linear dimension reduction technique.

5.2 Proposed Technique

The implementation of the proposed technique to investigate the multiple forgeries found in videos is described in this section. This technique uses the VGG-16 transfer learning model for automatic feature extraction. Manipulations in the video will likely disrupt the expected consistency of extracted features and will be considered as an artifact to detect the forgeries in the video. The working of the technique can be categorized into four main sections: 1) Pre-processing, 2) VGG-16 Model, 3) Feature selection, 4) Correlation Consistency. The architecture of the proposed technique for forgery (*i.e.,* intra-frame and inter-frame both) detection and localization is shown in Figure 5.1. The complete pseudo code to design the multiple forgery detection and localization technique is given in the algorithm 5.1. The first stage is video preprocessing, which involves splitting the input video into multiple frames. The second stage is the extraction of features using VGG-16, which extracts feature vectors of every corresponding frame. The spatial visual features vf_1, vf_2, ..., vf_n of the frames are extracted with a modified VGG-16 model for the identification of video forgeries. Then, Kernel PCA-based feature selection methodology is applied in order to reduce the dimension of extracted features. These feature vectors are then analyzed with the help of Pearson Correlation for the identification of multiple forgeries in videos in the final stage.

5.2.1 Video Pre-processing

The aim of preprocessing stage is to prepare data and facilitate further processing activities. The input video is first divided into a series of frames. Following that, all these video frames are resized into a size of $224 \times 224 \times 3$.

5.2.2 Feature Extraction using VGG-16

This stage aims to extract the features from the preprocessed frame using the VGG-16 model.

VGG-16 is the 16 layers Deep CNN Model developed by the Visual Geometry Group

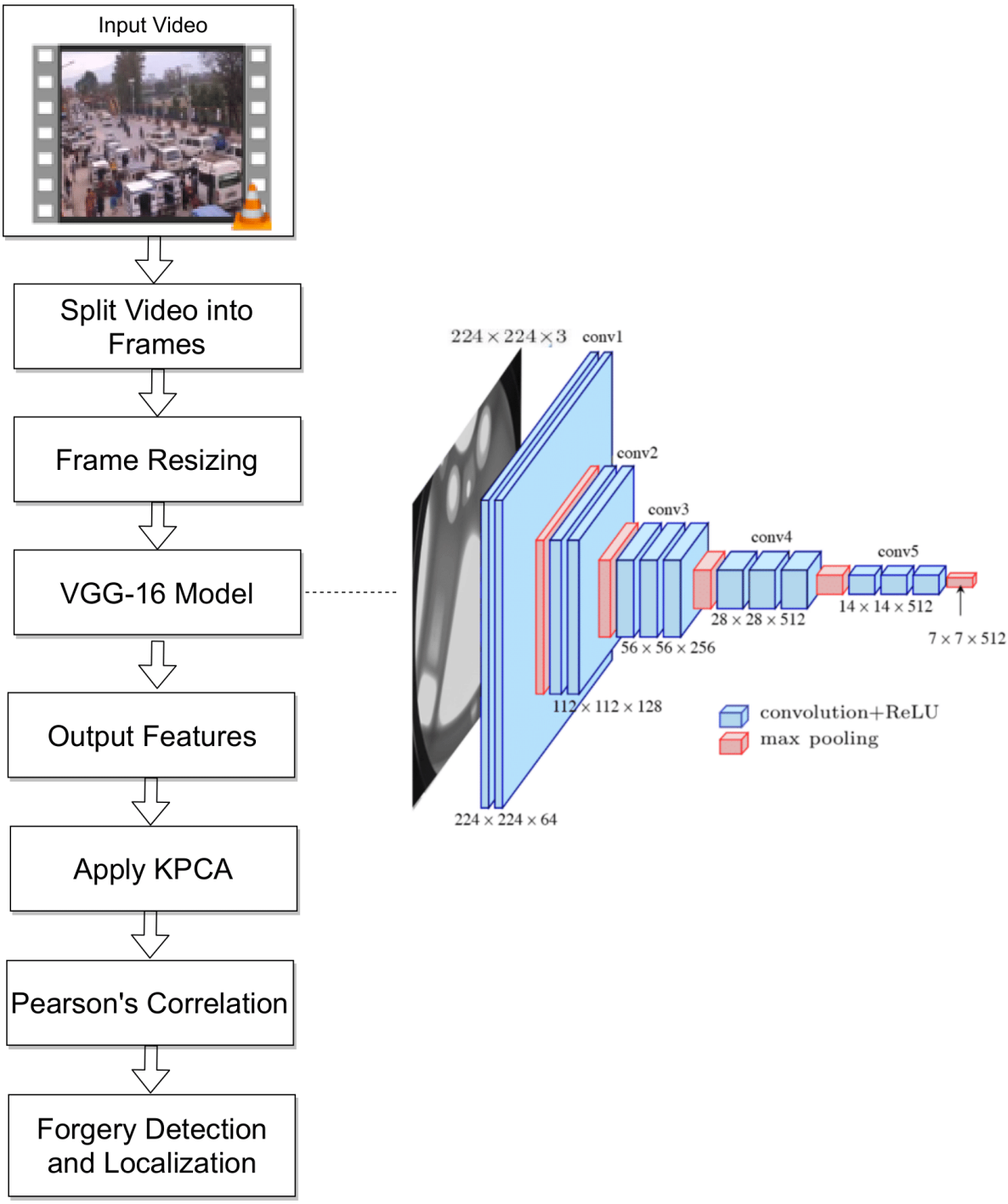


Figure 5.1: Architecture: Proposed technique

at Oxford University (Simonyan & Zisserman, 2014). The model is trained on millions of images taken from the ImageNet Dataset. As a result, the network with pre-trained weights has picked up rich feature representations for a variety of images. VGG-16 pre-trained transfer learning model is used for the automatic features extraction from the input video frames from the coevolutionary base of the model. The key reason for its appeal is that it can automatically extract features, eliminating the requirement for manual feature extraction. The original VGG16 network architecture is depicted in Figure 5.2 (a).

- In VGG-16 model, 16 indicates the total number of layers.
- Input layer takes color frames (Fixed size $224 \times 224 \times 3$) as an input.
- Convolution Layer: The frames are passed through a series of convolution layers. The convolutions layers have different channels with a filter size of 3×3.
- Max pooling Layer: It uses a max-pool window of the size of 2×2 with stride 2.
- Fully connected layer: Three Fully-Connected layers follow a stack of convolutional layers.
- Soft-max layer is the final layer. This layer uses the softmax function is given by the equation.

$$softmax(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{5.1}$$

- The activation function used in the hidden layers is ReLU.
- The VGG-16 network is quite large, with around 138 million parameters.

In the proposed technique, the existing VGG-16 model is modified for detecting the forgeries in the videos. This modification is done in order to perform feature extraction. In the modified VGG-16 model, the fully connected nodes are removed at the end of the network. Then fully connected nodes are replaced with current initialized ones to extract specific features. The modified architecture of the VGG-16 network is depicted in Figure 5.2 (b), in which fully connected (FC) layers are removed and returned to the final POOL layer. The last max-pooling layer's output (labelled by $7 \times 7 \times 512$) is considered a feature extraction part of the

model. This output is in the form of feature vectors later given to the next stage to identify the multiple forgeries. This modified pre-trained model is considered a feature extractor when performing the deep feature extraction, allowing the input frames to perpetuate forward direction, stopping at the pre-defined layer, and returning the feature vectors as output.

5.2.3 KPCA based Feature Selection

In this stage, kernel PCA is used to select representative features that reveal potentially complex structures in the extracted features from VGG-16. Kernel PCA uses a nonlinear mapping function (*i.e.,* kernel function) to project each feature vector having a lower-dimensional space into a new set of a higher-dimensional feature vector. The nonlinear mapping of a sample f can be denoted as $\phi(f)$, which is called kernel function. This technique transforms higher dimensional input feature vectors $f_1, f_2, \ldots\ldots, f_n$ into lower dimensional vector space $pc_1, pc_2, \ldots\ldots pc_n$. The covariance of the mapped input feature in the feature space C^F is give by (5.2) (Lee et al., 2004)

$$C^F = \frac{1}{N}\sum_{j=1}^{N} \phi(f_j)\phi(f_j)^T \tag{5.2}$$

$$\lambda v = C^F v \tag{5.3}$$

The eigenvalue and eigenvectors of the covariance matrix C^F are calculated from the equation (5.3). Where, $\lambda \geq 0$ denotes the eigenvalues and v denotes the eigenvectors.

By putting equation (5.2) (*i.e.,* value of C^F) in equation (5.3) gives equation (5.4)

$$C^F = \left(\frac{1}{N}\sum_{j=1}^{N} \phi(f_j)\phi(f_j)^T\right) v = \frac{1}{N}\sum_{j=1}^{N} \langle \phi(f_j), v\rangle \phi(f_j)) \tag{5.4}$$

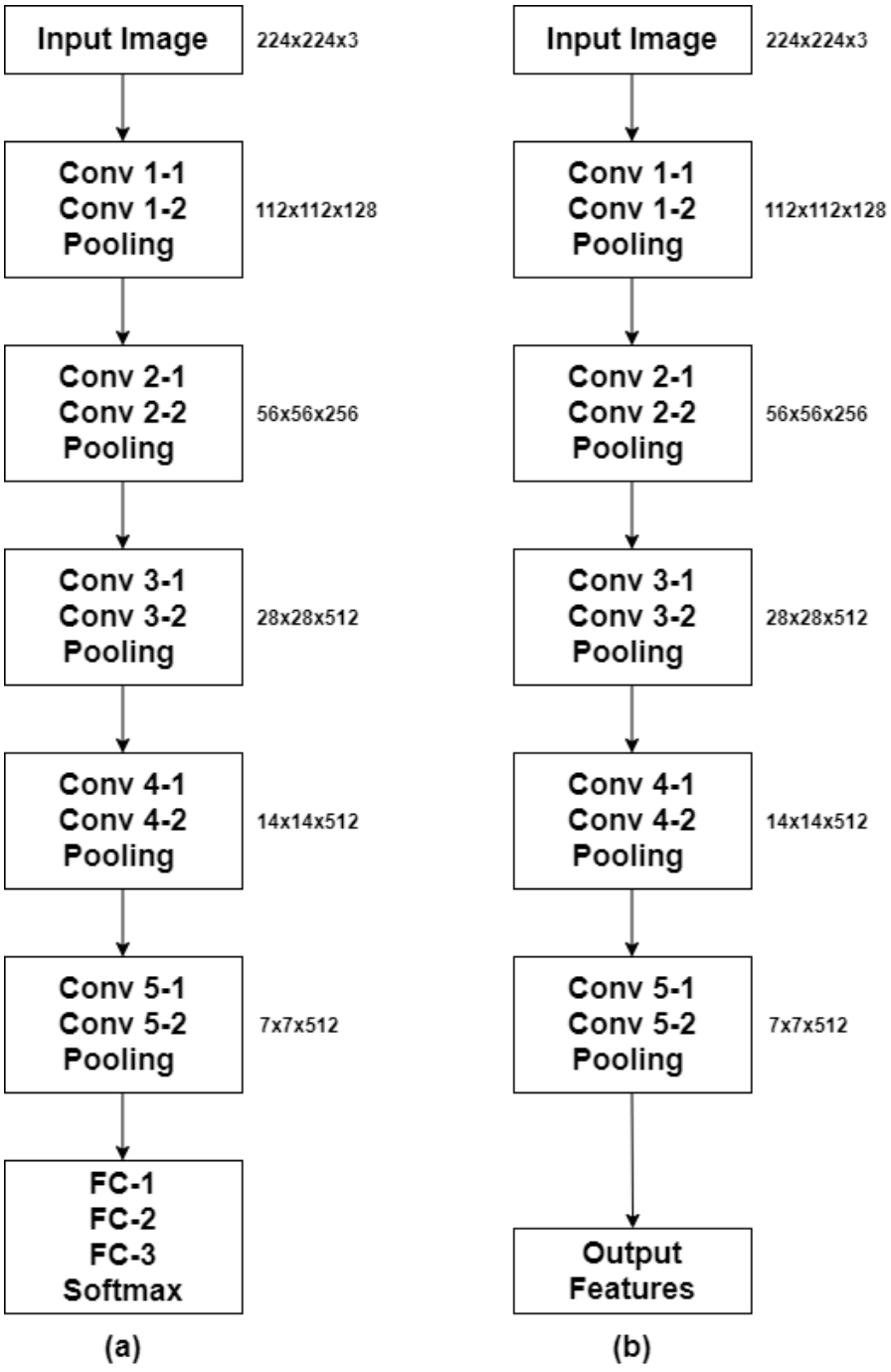


Figure 5.2: VGG-16 layers (a) The original VGG16 network architecture (b) Proposed fine-tuned VGG-16 network architecture (FC layers are Removed from VGG-16)

Recognizing that all solutions v with $\lambda \neq 0$ fall into the range of $\phi(f_1)$, $\phi(f_2)$, $\ldots \phi(x_N)$, coefficients $\alpha_i (i = 1,2,\ldots,N)$ that provide $v = \sum_{i=1}^{N} \alpha_i \phi(f_i)$.

After that, equation (5.5) is given by

$$\lambda \sum_{i=1}^{N} \alpha_i \langle \phi(f_k), \phi(f_i) \rangle = \frac{1}{N} \sum_{i=1}^{N} \alpha_i \left\langle \phi(f_k), \sum_{j=1}^{N} \phi(f_j) \right\rangle \langle \phi(f_j), \phi(f_i) \rangle \tag{5.5}$$

for all $k = 1,2,3 \ldots N$. By adding kernel matrix K with $[K]_{i,j} = K_{i,j} = \langle \phi(f_i), \phi(f_j) \rangle$. So equation (5.5) now becomes

$$\lambda NK\alpha = K^2\alpha \tag{5.6}$$

Where $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$. For new sample f, the principal component is calculated by equation (5.7)

$$pc_k = \langle v_k, \phi(f) \rangle = \sum_{i=1}^{N} \alpha_i^k \langle \phi(f_i), \phi(f) \rangle \tag{5.7}$$

where $k = 1,2,\ldots,p$, where p gives the number of principal components.

5.2.4 Investigation of Correlation Distribution

In this section, the correlation distribution between adjacent feature vectors collected from the previous stage (*i.e.,* after feature selection using KPCA) is investigated. Correlation consistency is a similarity metric that can be used to investigate the similarity between the feature vectors. We have extracted features from each frame of the input video in the previous step. The correlation distribution among the extracted features is then calculated. We have incorporated it as

a similarity metric for the detection of manipulations. The Pearson correlation coefficient ρ is determined by equation (5.8)

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 (y_i - \bar{y})^2}} \tag{5.8}$$

5.3 Experimental Analysis and Discussion

In this section, we discuss the experimental dataset to test the proposed technique and experimental results followed by an overall analysis. The proposed technique is implemented on the Anaconda and MATLAB 2020A using a Core i5 8th Generation Processor, SSD 128Gb, 8GB RAM (DDR4), and 4GB GPU-NVIDIA GeForce (GTX 1050Ti).

5.3.1 Experimental Setup

As stated earlier, a pre-trained deep learning model was used to train the network, *i.e.,* VGG-16. The VGG-16 architecture, including all specifications considered in this study, is shown in Table 5.1. The Table 5.2 below highlights the hyperparameters utilized in this work. Each of these parameters ranges are indicated within square brackets.

Algorithm 5.1 Proposed technique

Input: Digital Video

Output: Multiple Forgery Identification

Load the input video V

Divide the input video V into frame sequence $f_1, f_2, f_3, f_4 ..... f_n$

Count the number of frames in the input video and stored it into $totalFrames$

for $j = 1$ to $totalFrames$ **do**

    Read the frame

    Resize the frame into $224 \times 224 \times 3$

end for

Initialize the VGG-16 Model: $model \leftarrow Sequential()$

$Data \leftarrow$ Feature Extracted From VGG-16

Apply KPCA on $Data$

Calculate covariance matrix C^F for $Data$ according to Eq (5.2)

Compute the eigenvalue and eigenvectors of C^F using decomposition Eq (5.3)

Calculate the principal component pc_k using Eq (5.7)

$Data = pc_1, pc_2, ... pc_n.$

$[r, c] =$ Shape of the $Data$

Initialize the ρ with 0

for $i = 1$ to length of c **do**

    $PCC(1,i) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 (y_i - \bar{y})^2}}$

end for

$p = n - 1$

$corDiff = 0$

for $i \leftarrow 1$ to p **do**

    $corDiff(1,i) = \rho(1,i) - \rho(1,(i+1))$

end for

Plot on $corDiff$

Table 5.2: Network Hyperparameter

Hyperparameter	**Abbreviation**	**Range**
Learning rate	learning_rates	[0.001, 0.002]
Batch Size	batch_sizes	[16, 32]
Optimizer	optimizer	[RMS Prop, Adam, SGD]
Drop Out	drop_out	[0.2, 0.3]

Table 5.1: VGG-16 Network details(ConvL: Convolution Layer, MPL: Max Polling Layer)

Layer	**Feature Map**	**Kernel Size**	**Stride**
2 × ConvL	64	3 × 3	1
1 × MPL	64	3 × 3	2
2 × ConvL	128	3 × 3	1
1 × MPL	64	3 × 3	2
2 × ConvL	256	3 × 3	1
1 × MPL	256	3 × 3	2
3 × ConvL	512	3 × 3	1
1 × MPL	512	3 × 3	2
3 × ConvL	512	3 × 3	1
1 × MPL	512	3 × 3	2

We have used various hyperparameters such as learning rate, drop out rate, optimizer, the batch size for building the learning models. All hyperparameters were chosen based on experimental trials from the Table 5.2. During training, an Adam optimizer was used to refine the network parameters. The learning rate of 0.001 was selected. To regularize the deep models, a dropout ratio of 0.2 was chosen. The different batch sizes were trialled during learning, the batch sizes were chosen from the range [16,32], and the optimal batch size we obtained was 32.

5.3.2 Dataset Description

A multi-forged video dataset is created to perform experimentation with the proposed technique combining videos from four datasets: SULFA (Qadir et al., 2012), (VTD, 2017), (REWIND, 2013) and (VTL, 2018). The videos in this library are selected and exposed to tamperings such as copy-move, frame splicing, frame insertion, and frame deletion. Also, some videos have undergone operations like noise additions and brightness, contrast, saturation, and hue modifications with the use of Clideo software (Clideo, 2020). There are 30 videos with a stationary and dynamic background in the created multiple forged video library. We termed it a MFVD-3. The video formats of this dataset consist of MP4 and AVI coded by H.264 and Motion JPEG encoding standards, respectively, with different GOP lengths using (FFmpeg, 2019). We used the varied length of GOP, *i.e.,* 10, 12, and 16 to test the proposed technique. The full description of 30 forged videos is shown in Table 5.3.

Table 5.3: Forged videos details

Videos	**Format**	**Resolution**	**Details**
FV 1	MP4	1280×720	CM:15;FS:28;FI:67-77;FD:122-161
FV 2	MP4	1280×720	CM:15;FS:28;FI:67-77;FD:122-161
FV 3	MP4	1280×720	CM:15;FS:28;FI:67-77;FD:122-161
FV 4	MP4	1280×720	CM:15;FS:28;FI:67-77;FD:122-161
FV 5	MP4	1280×720	CM:15;FS:28;FI:67-77;FD:122-161
FV 6	AVI	640×480	CM:26;FS:15;FI:40-50;FD:167-196
FV 7	AVI	640×480	CM:26;FS:15;FI:40-50;FD:167-196
FV 8	AVI	640×480	CM:26;FS:15;FI:40-50;FD:167-196
FV 9	AVI	640×480	CM:26;FS:15;FI:40-50;FD:167-196
FV 10	AVI	640×480	CM:26;FS:15;FI:40-50;FD:167-196
FV 11	AVI	320×240	CM:28;FS:5;FI:50-60;FD:147-187
FV 12	AVI	320×240	CM:28;FS:5;FI:50-60;FD:147-187
FV 13	AVI	320×240	CM:28;FS:5;FI:50-60;FD:147-187
FV 14	AVI	320×240	CM:28;FS:5;FI:50-60;FD:147-187
FV 15	AVI	320×240	CM:28;FS:5;FI:50-60;FD:147-187
FV 16	MP4	1280×720	CM:9;FS:27;FI:60-70;FD:113-147
FV 17	MP4	1280×720	CM:9;FS:27;FI:60-70;FD:113-147

FV 18	MP4	1280×720	CM:9;FS:27;FI:60-70;FD:113-147
FV 19	MP4	1280×720	CM:9;FS:27;FI:60-70;FD:113-147
FV 20	MP4	1280×720	CM:9;FS:27;FI:60-70;FD:113-147
FV 21	AVI	640×480	CM:9;FS:11;FI:32-42;FD:113-147
FV 22	AVI	640×480	CM:9;FS:11;FI:32-42;FD:113-147
FV 23	AVI	640×480	CM:9;FS:11;FI:32-42;FD:113-147
FV 24	AVI	640×480	CM:9;FS:11;FI:32-42;FD:113-147
FV 25	AVI	640×480	CM:9;FS:11;FI:32-42;FD:113-147
FV 26	MP4	640×480	CM:17;FS:41;FI:56-66;FD:120-154
FV 27	MP4	640×480	CM:17;FS:41;FI:56-66;FD:120-154
FV 28	MP4	640×480	CM:17;FS:41;FI:56-66;FD:120-154
FV 29	MP4	640×480	CM:17;FS:41;FI:56-66;FD:120-154
FV 30	MP4	640×480	CM:17;FS:41;FI:56-66;FD:120-154

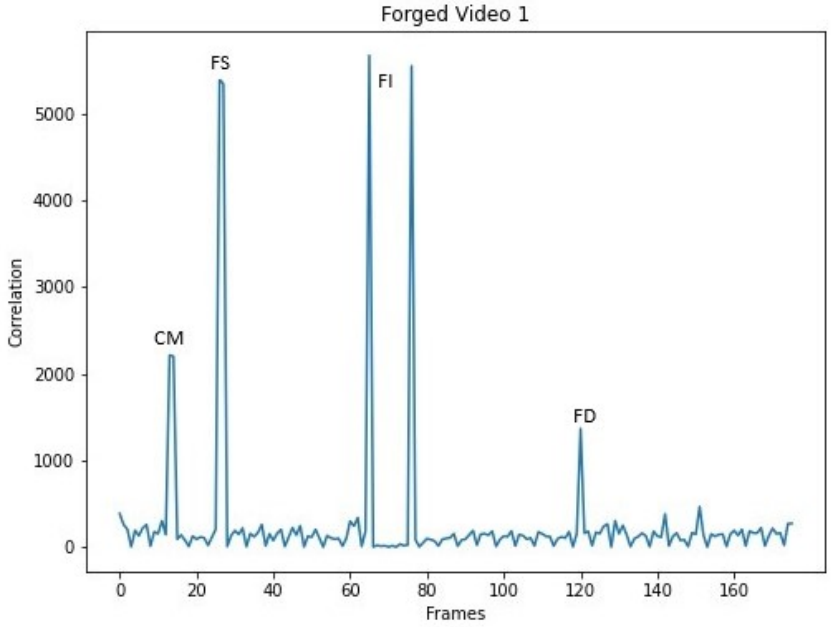
5.3.3 Experimental Results

In this section, we present the experimental results of the proposed technique on self-created sample multiple forged video (*i.e.,* for the videos 1,6,11,16,21 & 26). The experimental results are depicted in Figure 5.3.

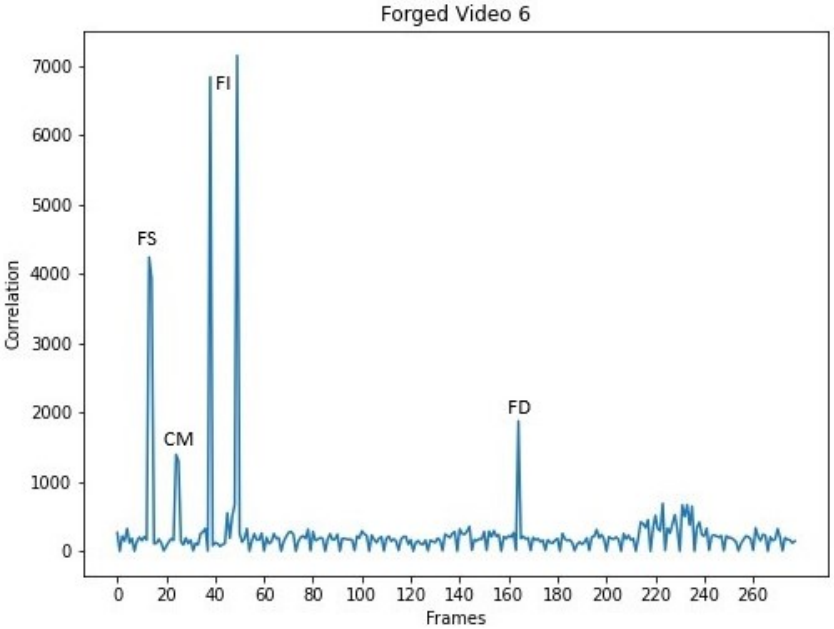
5.3.4 Performance Evaluation

The proposed technique is successfully tested and validated on MFVD-3. The performance parameters, such as accuracy, precision, recall, and F1-score, are tabulated in Table 5.4. From this table, one can observe that the accuracy for frame insertion is 100%, while for copy-move and splicing forgery, the accuracy is 97.04% and 98.90%, respectively. Frame insertion and deletion forgery, on the other hand, have an accuracy of 98.88% and 94.30%, respectively. The accuracy for multiple forgery is 97.24%.

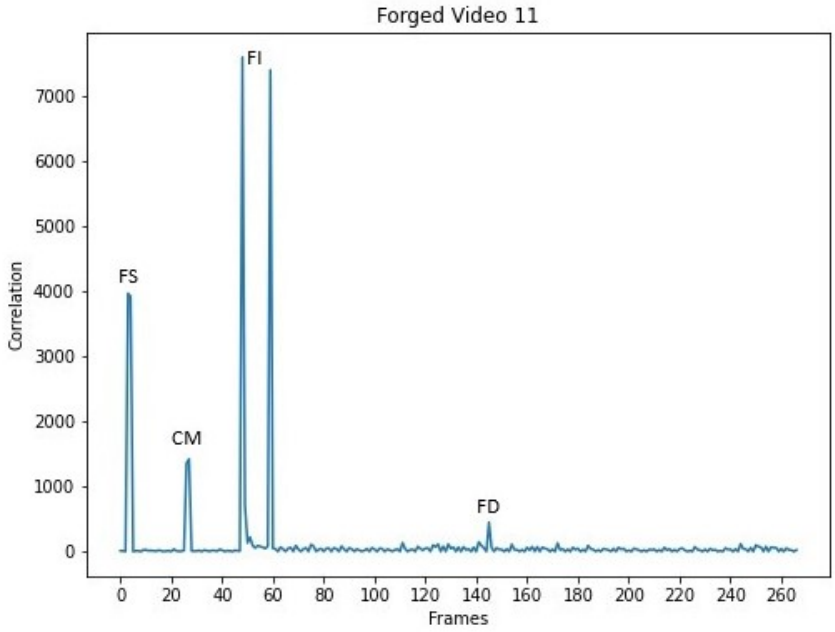
(a) FV 1


Forged Video 1
FS
FI
CM
FD
Correlation
Frames
0
1000
2000
3000
4000
5000
0
20
40
60
80
100
120
140
160

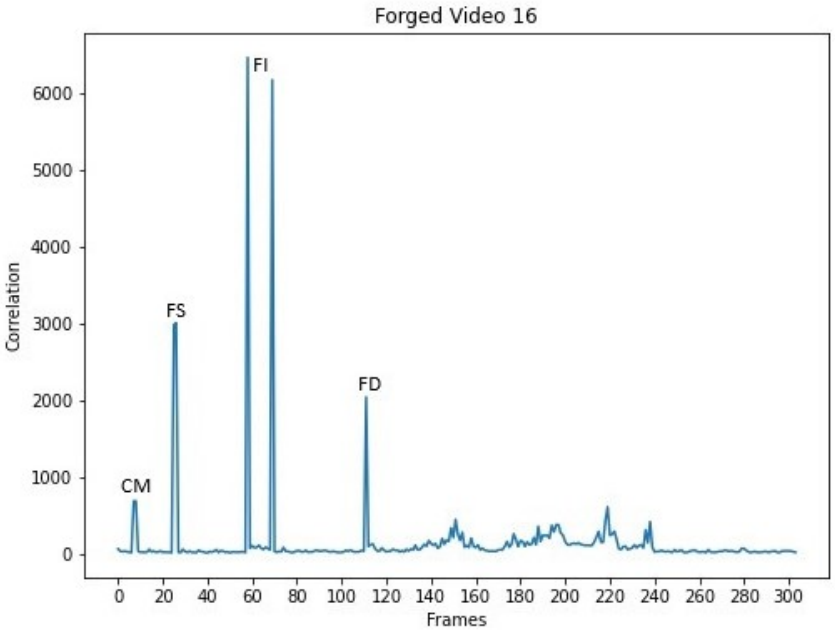
(b) FV 6


Forged Video 6
FS
FI
CM
FD
Correlation
Frames
0
1000
2000
3000
4000
5000
6000
7000
0
20
40
60
80
100
120
140
160
180
200
220
240
260

(c) FV 11

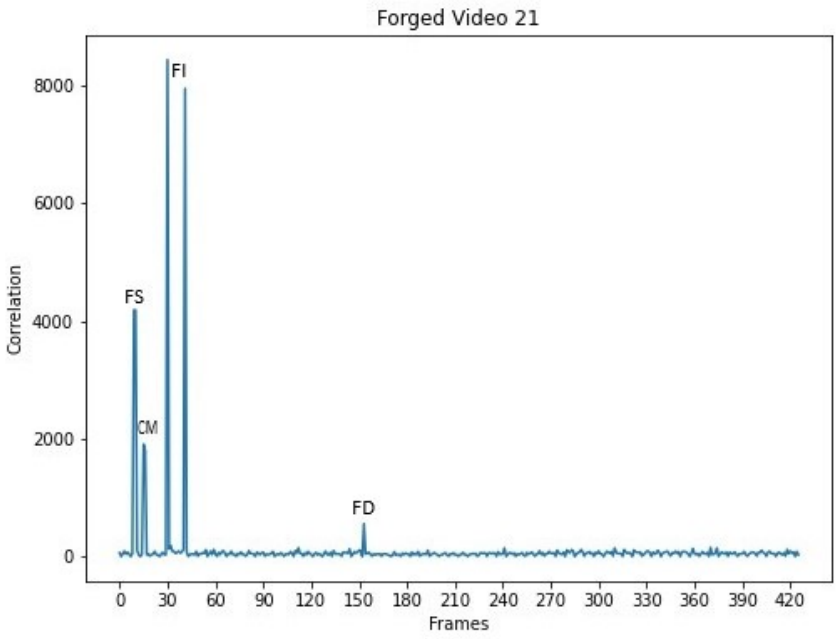

Forged Video 11
FI
FS
CM
FD
Correlation
Frames
0
1000
2000
3000
4000
5000
6000
7000
0 20 40 60 80 100 120 140 160 180 200 220 240 260

(d) FV 16


Forged Video 16
FI
FS
CM
FD
Correlation
Frames
0
1000
2000
3000
4000
5000
6000
0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300

(e) FV 21

(f) FV 26



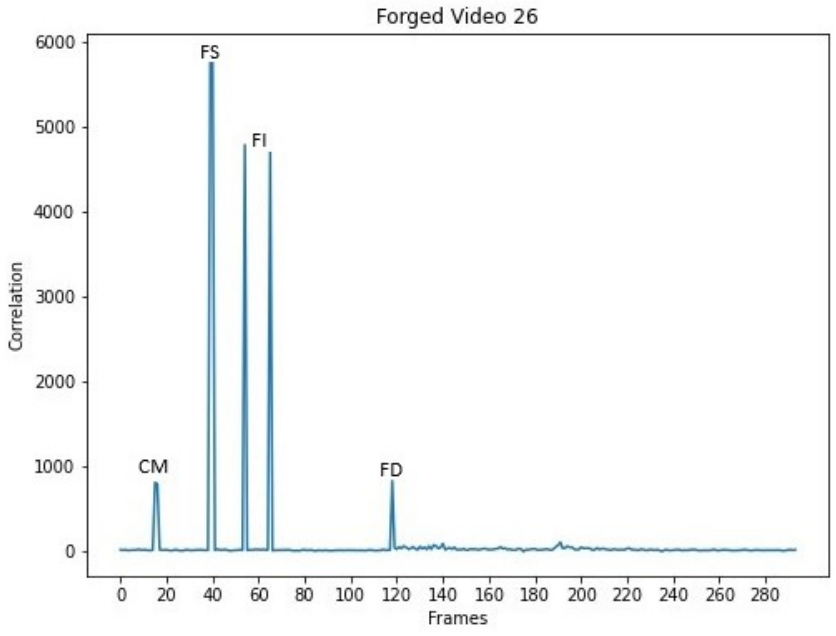


Figure 5.3: Experimental results on sample forged videos

Table 5.4: Performance of proposed technique

Forgery Type	ACC	PR	RR	F1-score
Copy-move	97.04	96.06	94.80	95.43
Frame Splicing	98.90	98.81	95.48	97.12
Frame Insertion	98.88	98.95	96.89	97.91
Frame Deletion	94.30	93.81	93.58	93.19
Multiple Forgeries	97.24	96.86	94.81	95.82

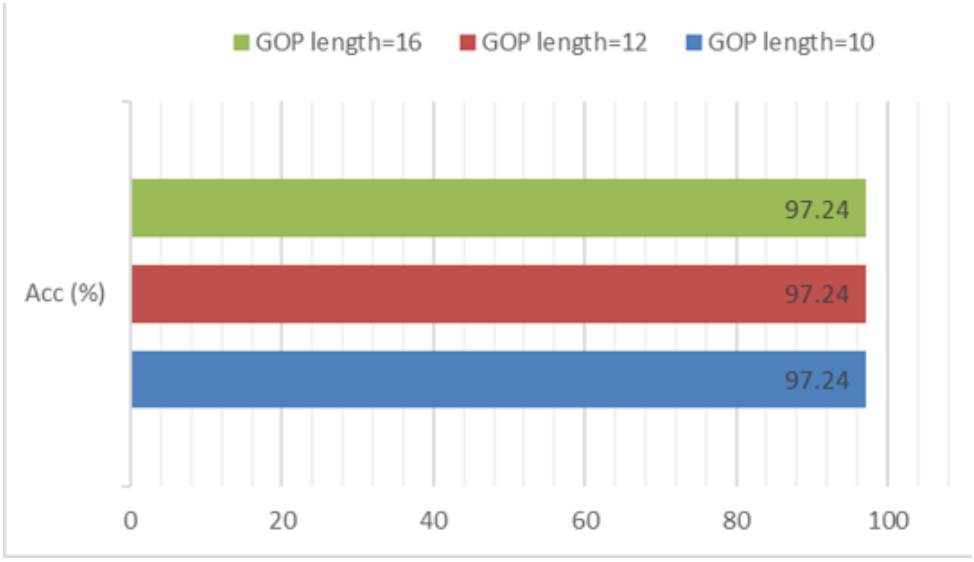


Figure 5.4: Evaluation of the proposed technique against the different GOP length videos

5.3.4.1 Evaluation based on GOP length

The dependency of the proposed video forgeries detection technique against the changes in the GOP length is examined. Most of the existing techniques are tested on standard GOP sizes such as 10,12 and 16 (Bakas et al., 2019). Therefore we have also tested our technique on varied lengths of GOP, such as 10,12 and 16. The evaluation against the varied lengths of GOP is depicted in Figure 5.4. This figure is evident that the proposed forgery detection technique is independent of change in GOP length and provides the same accuracy for identifying multiple forgeries, even when the GOP length of the videos is different.

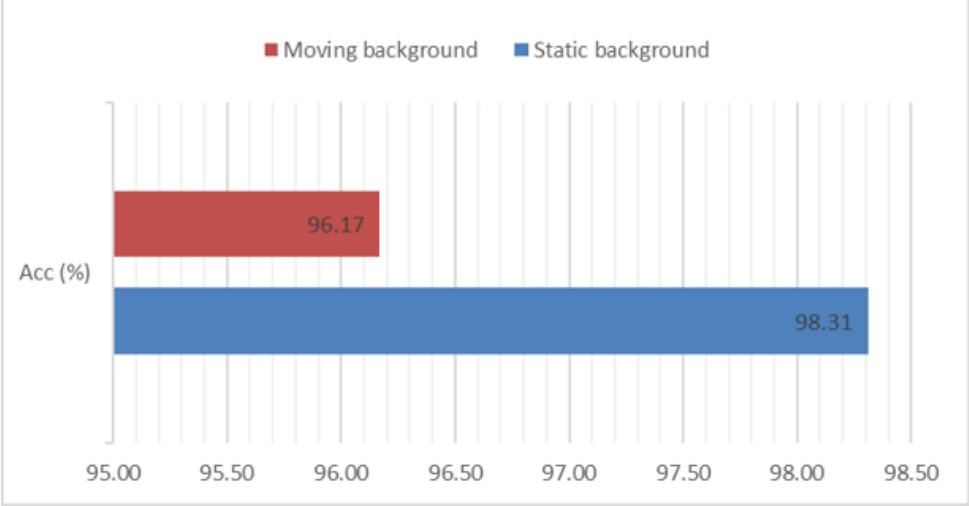


Figure 5.5: Evaluation of the proposed technique against the static and moving background video

5.3.4.2 Evaluation based on Background

The dependency of the proposed technique against the changes in the video backgrounds is examined. The videos with the static and moving background are considered in order to evaluate the performance against the background dependency of the proposed technique, which is shown in Figure 5.5. From this figure, it is inferred that the technique works well for forgeries detection from both static as well as moving background videos. Although, in the case of a moving background, the accuracy is affected slightly due to a sudden change observed between the consistency of extracted feature of frames in the video sequence.

5.3.4.3 Evaluation based on Post-processing Operations

This section analyzes the proposed technique performance under different video post-processing operations like noise additions and brightness, contrast, saturation, and hue modifications. In the background, we have considered different sorts of ranges: Noise attack range was [Gaussian noise: 5db, 10db, 15db and 30db], Contrast range was [High, Medium, Low] and Hue/saturation range was [High, Medium, Low]. We tested our techniques against the post-processing operations with different ranges and estimated the average accuracy over a set of ranges. It is clear from the Figure 5.6 that our technique is effective in forgeries detection under several post-processing operations. For the post-processing attacks like Hue, contrast, brightness change,

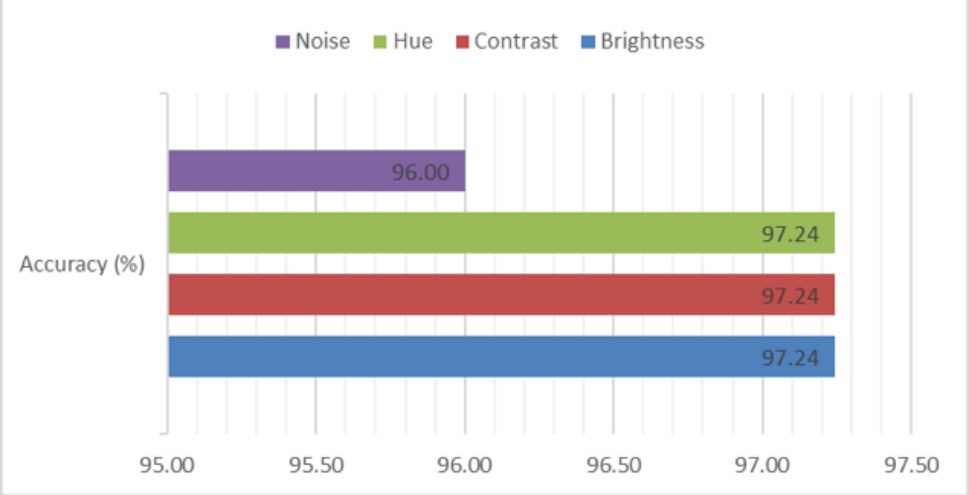


Figure 5.6: Evaluation of the proposed technique against the Post-processing operations

the proposed technique provides 97.24% accuracy(i.e., not affected by operations). However, for the noise attacks, our technique gives an accuracy of 96 %.

5.3.4.4 Overall Comparison with Existing Techniques

In the comparative analysis section, we have selected the only latest state-of-the-art, relevant techniques to compare with the proposed techniques. To the best of our knowledge, the current literature doesn't contain multiple forgery detection techniques. So we have compared our work with the state of the art techniques (L. Su et al., 2018), (Wei et al., 2019) and (Bakas et al., 2019) that can detect the single forgery at a time. In the background, we have tested the proposed and existing techniques against the individual forgery on the multi-forged video dataset MFVD-2 for a fair comparison. The comparison has been made to show the effectiveness of our techniques in terms of accuracy, multiple-forgery detection or not, GOP and background dependency, robustness against the post-processing operations. The comparison shows that the technique outperforms the existing techniques with respect to several different parameters. The existing techniques in (Bakas et al., 2019; L. Su et al., 2018; Wei et al., 2019) can only handle a single forgery at a time. On the other hand, the proposed technique can handle multiple forgeries. The comparative output is given in Table 5.5. This Table indicates that the proposed technique gives 97.24% accuracy, which is higher than existing techniques.

Table 5.6 presents the comparison concerning the GOP length and video background dependency, post-processing robustness, and execution time. The techniques given in (L. Su et

Table 5.5: Proposed *vs* existing techniques comparison

Techniques	**Forgeries**	**ACC (%)**
(L. Su et al., 2018)	CM	92.02
(Wei et al., 2019)	FI, FD, FDU	92.89
(Bakas et al., 2019)	FI, FD, FDU	96.06
Proposed	CM, FS, FI, FD	97.24

Table 5.6: Comparision with existing techniques

Techniques	**GOP robustness**	**background robustness**	**Post-processing operations robustness**	**Execution time/ video (sec)**
(L. Su et al., 2018)	NO	NO	NO	71
(Wei et al., 2019)	NO	NO	NO	153
(Bakas et al., 2019)	YES	YES	NO	110
Proposed	YES	YES	YES	38

al., 2018; Wei et al., 2019) are dependent upon GOP length and background. However, our technique and technique specified in (Bakas et al., 2019) are not dependent on the GOP length and video background. Besides, the proposed technique is robust to the operations like noise addition and brightness, contrast, saturation, and hue modifications in comparison with existing techniques in (Bakas et al., 2019; L. Su et al., 2018; Wei et al., 2019). Another comparison is based on the execution time shown in Table 5.6. The proposed technique automatically extracts the required feature with the use of the VGG-16 model; nevertheless, existing techniques consider hand-crafted features, requiring more computational time. Therefore, it is observed that the proposed forgery detection technique is computationally effective as it requires less time than other existing techniques. Although our technique is superior to existing techniques in (Bakas et al., 2019; Wei et al., 2019), but it cannot expose the frame duplication forgery.

5.4 Conclusion

In this Chapter, a passive technique for multiple forgery detection in the video is proposed using the VGG-16 neural network and KPCA. This technique can detect and localize the forgeries such as copy-move, splicing, insertion, and deletion present in the individual video. This technique is tested on the MFVD-3 dataset. Experimental results demonstrate the efficiency of the proposed scheme in terms of robustness against several post-processing operations, GOP structure, and background. The proposed technique outperforms existing techniques and can handle multiple tampering in the video with an accuracy of 97.24%.

CHAPTER 6

Multiple Forgery Detection and Localization Technique using PCT and NBAP

In this Chapter, a technique using the PCT, NBAP, and GoogleNet model is presented to detect and localize multiple forgeries from the video. In this Chapter, Section 6.1 discussed the background. Section 6.2 explains the overall flow of the proposed technique, Section 6.3 presents the experimental result analysis, Section 6.4 gives the overall comparison of proposed techniques and the Conclusion of the chapter is discussed in Section 6.5.

Contents of the work presented in this Chapter have been published in *Multimedia Tools and Applications*, pp.1-21, 2021, Springer. (SCI Indexed)

6.1 Background

In this section, a hybrid model of PCT, NBAP and GooglNet is discussed. Polar Cosine Transform (PCT) is one types of polar harmonic transforms. The Polar Harmonic Transform (PHT) describes a group of transforms whose kernels are both fundamental and harmonic (L. Li, 2012). They represent an invariant image pattern representations for 2-D image retrieval and pattern recognition tasks. PHT gives orthogonal foundations for describing rotation-invariant patterns. In PCT, The orthogonal kernel in the unit circle domain comprises a radial and circular component, and the generalized orthogonal moments are obtained by projecting the image onto the orthogonal kernel function, represented by H_{nl}.

$$H_{nl}(\gamma,\theta) = R_n(\gamma)e^{il\theta} \tag{6.1}$$

Where n:order, $R_n(\gamma)$: radial component, l:repetition and $e^{il\theta}$: circular component.

Recently, in areas like image processing and computer vision, the latest concept is to collect binary angular patterns for various sets of applications. NBAP is type of angular pattern features. The key idea in NBAP is to capture the information among neighbourhood frames and then build rich texture descriptors using such features. These features are typically invariant to similarity transformation, including rotation and scaling, which implies neither shifting nor normalization is needed when analyzing related descriptors.

GoogLeNet is a type of pre-trained deep learning model which is based on Convolutional Neural Network, trained over the ImageNet dataset (Szegedy et al., 2015). CNNs are a subset of Artificial neural networks or multi-layer neural networks that are meant to understand patterns directly from pixels of the images with minimal preprocessing (Albawi et al., 2017). CNN is composed of input, one or more hidden layers and the final output layer. The convolution, pooling and fully connected layers are the part of hidden layers. It filters input volumes to higher levels of abstraction using several convolution layers. By adding pooling layers for limited translation and rotation invariance, CNNs increase their detection capabilities. The pooling layer in CNN also permits the use of more convolutional layers by lowering memory

consumption. Normalization layers are used to standardize part of input regions by shifting all inputs in a layer towards a mean equal to zero and variance equal to one. Fully connected layers contain neurons that function similarly to convolutional layers but differ in that they are linked to all activations in the preceding layer.

6.2 Proposed Technique

The overall implementation of the proposed technique for detecting forgeries from a video is described in this section. It has three stages: a) Pre-processing, b) Feature extraction, and c) GoogleNet model. In order to propose the techniques, PCT and NBAP features are considered. When the video is subjected to any manipulation, consistency in these features is affected, which is further used as a forensic clue. This clue can be quantified and used as evidence of tampering. The overall flow of the proposed technique is depicted in Figure 6.1.

6.2.1 Pre-processing

The pre-processing stage aims to prepare data and to facilitate processing activities. It includes video splitting, frame resizing, and RGB to grayscale conversion. The input video is first divided into a series of frames. Following that, the RGB video frames are then transformed to grayscale video frames. The raw video needs to be pre-processed before it can be given to the next stage.

6.2.2 PCT and NBAP Feature Extraction

In this part, PCT and NBAP features are obtained from the pre-processed video frames. First, The PCT is applied to the pre-processed video frames to obtain the necessary features. The following are the steps used by PCT to extract features.

- Read the grayscale input frame after the pre-processing stage.

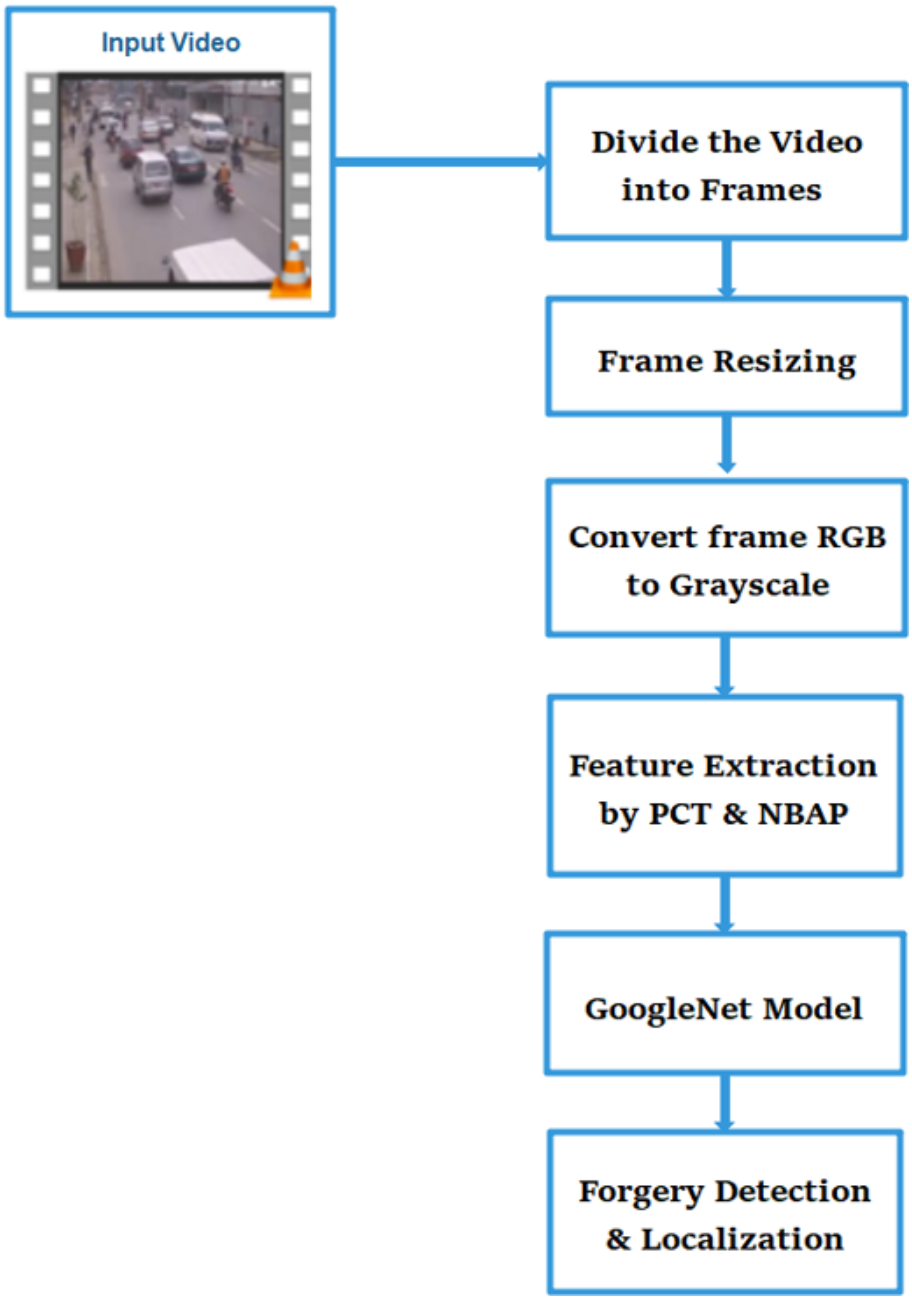


Figure 6.1: Proposed technique flowchart

- Image ($I_{(i,j)}$) is converted into polar coordinates ($I_{(\gamma,\theta)}$),

$$\gamma = \sqrt{i^2 + j^2} \tag{6.2}$$

$$\theta = \arctan \frac{i}{j} \tag{6.3}$$

Where γ: radius and θ: azimuth.

- The value of γ set on a unit circle and modified to the function $H_{nl}(\gamma,\theta)$ as,

$$f(\gamma,\theta) = \sum_{n=0}^{\infty} \sum_{l=-\infty}^{\infty} [M_{nl} H_{nl}(\gamma,\theta)] \tag{6.4}$$

- PCT kernel is represented as a radial and circular component.

$$H_{nl}(\gamma,\theta) = R_n(\gamma) e^{il\theta} = e^{i2\pi nr^2} e^{il\theta} \tag{6.5}$$

Where n:order, l:repetition

- The PCT coefficients can be expressed as,

$$M_{nl} = \Omega_n \int_0^{2\pi} \int_0^1 [e^{i2\pi nr^2} e^{il\theta}]^* \gamma d\gamma d\theta \tag{6.6}$$

Where $|n|$, $|l|$ = 0,1,...,∞, $[.]^*$ is the complex conjugate operation

$$\Omega_n = \begin{cases} \frac{1}{\pi} & \text{if } n = 0 \\ \frac{2}{\pi} & \text{if } n = 1 \end{cases} \tag{6.7}$$

- The orthogonal kernel condition is represented as,

$$\int_0^{2\pi} \int_0^1 H_{nl}(\gamma,\theta)[H_{n'l'}(\gamma,\theta) \times \gamma d\gamma d\theta] = \pi \delta_{nn'} \delta_{ll'} \tag{6.8}$$

- The PCT discrete form is expressed as,

$$M_{nl} = \frac{4}{\pi MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [H_{nl}(i,j) \times I(i,j)] \tag{6.9}$$

Following that, the NBAP is implemented immediately after PCT extraction. Along with

the PCT extracted features, this NBAP implementation contributes some additional features. NBAP's feature extraction steps are outlined in the algorithm 6.1.

Algorithm 6.1 Neighborhood Binary Angular Pattern

Input: Video Frames

Output: NBAP Features

1: Read the gray scale Image I
2: Size of Image $= size(I,1)$
3: **for** $x = 2$ to Size of Image - 1 **do**
4: **for** $y = 2$ to Size of Image - 1 **do**
5: Create a window of size 3×3 and Extract 8 neighbhoord with respective to (x,y) as a center
6: Extract center pixel value of window created in the step 5
7: **for** $i = 2$ to window size -1 **do**
8: **for** $j = 2$ to window size -1 **do**
9: Compute the deviation of windows centre pixel with the computed average
10: **if** Value obtained in the previous step is +ve **then** replace non-diagonal elements of the window by 1 otherwise 0
11: **end if**
12: Flatten the result obtained in the step 10
13: **end for**
14: **end for**
15: Features = $\sum$ Pixel values obtained in the step 12
16: **end for**
17: **end for**
18: Return Features

6.2.3 GoogleNet Methodology

The pre-trained deep learning model such as GoogleNet is used in the proposed technique. GoogleNet is pre-trained CNN based deep learning model with 22 layers. By removing the last layer, we fine-tuned the pre-trained model. It is a type of CNN in which the deep neural

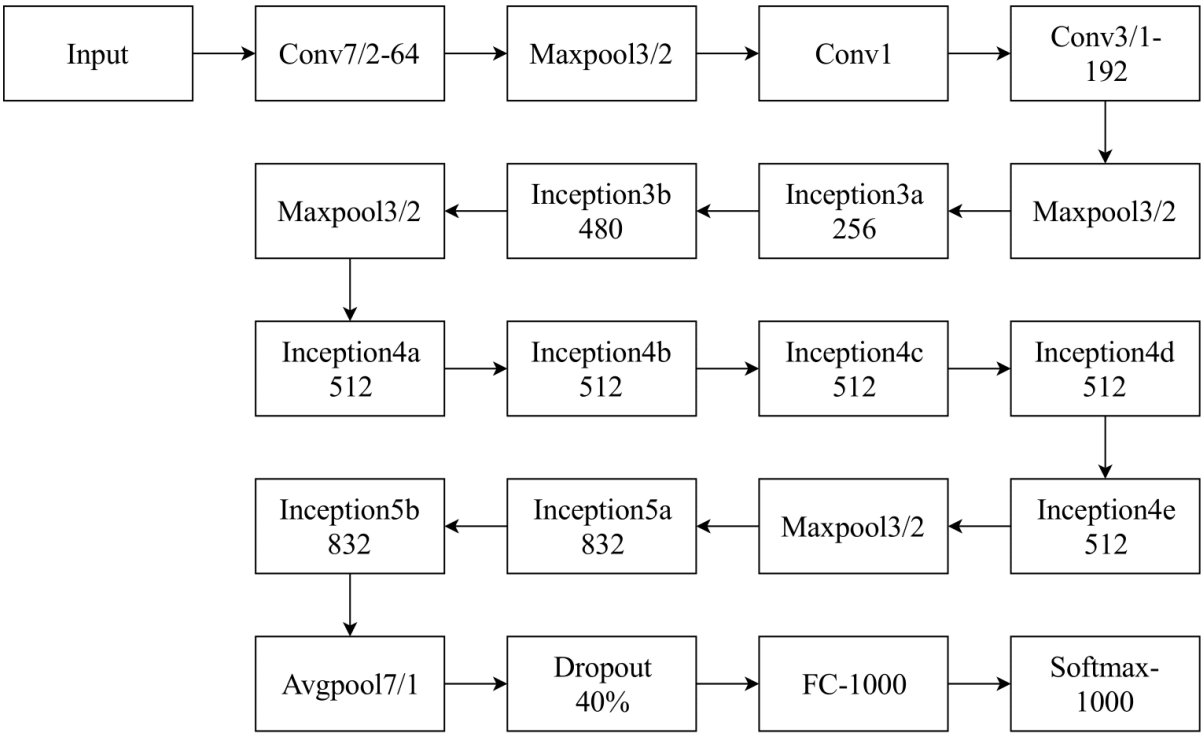


Figure 6.2: Architecture of GoogleNet model

network uses several convolution filters of varied sizes. A CNN consists of a set of neurons having learning biases and weights. Each neuron in CNN takes a set of inputs, calculates a weighted sum, applies the activation function, and outputs the result. The typical GoogleNet architecture is depicted in Figure 6.2. The typical components used in GoogleNet architecture are as follows,

- Convolution:- 1×1 convolution is used by GoogleNet. It is used to lessen the parameters counts(like biases, weights, *etc.*) in the GoogleNet transfer learning model. 1×1 convolution, employed by GoogleNet, reduces the parameters. The purpose of this reduction is to significantly increase the dimension of the architecture.

- Inception:- GoogleNet uses the inception module as a fundamental layer to improve the computational speed. Each layer applies a parallel filter process to the input from the previous layer. The entire architecture of GoogleNet is made up of 9 Inception modules. Figure 6.3 shows the module inception having 1×1 convolution.

- Max-pooling: The convolutions go through a max-pooling layer before being given to the subsequent inception module. Some inception modules have two max-pooling lay-

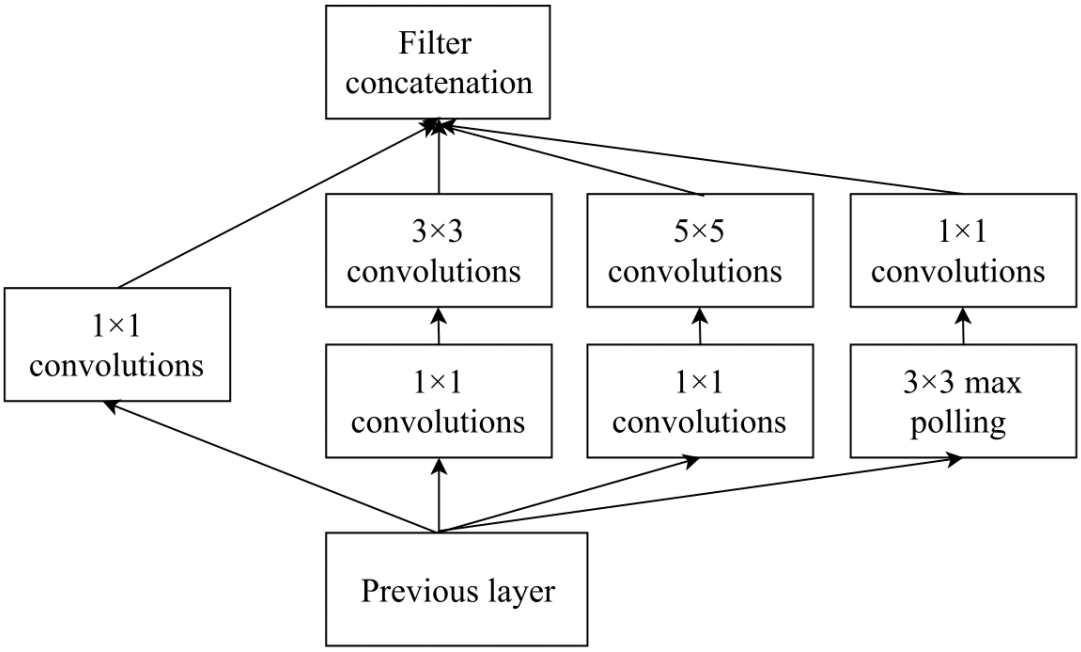


Figure 6.3: Inception module (1×1 convolution)

ers. These max-pooling layers' job is to downsample the input as it travels through the network.

- Average pooling:- The average pooling layer is present just after the set of inception layers. This layer is adopted to lower the dimensionality of a three-dimensional tensor. The 7×7 feature map is averaged to 1×1 by the pooling layer.
- Dropout:- Dropout layer is applied before the linear layer. It is a regularisation method that is utilized to prevent the overfitting of the network during training.
- FC-1000:- Fully connected (FC) layer is used at the end of the network, consisting of 1000 hidden units.
- Softmax:- The final layer is the softmax layer. This layer uses the softmax function is given by the equation.

$$softmax(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{6.10}$$

The extracted image features(*i.e.,* by PCT and NBAP) of the video are combined and fed into the GoogleNet model, and then it is adopted to detect forgeries. Multiple forgeries localization is determined during the evaluation for each video based on inconsistencies between

frames based on the extracted features. The squared Euclidean distance method is used to calculate frame inconsistencies. It is calculated using the following equation:

$$\delta = || f(n) - f(n+1) ||^2 \tag{6.11}$$

6.3 Experimental Result Ananlysis

This section includes a description of datasets, as well as findings and observations from the implementation of the proposed technique. The proposed technique has been executed on MATLAB 2019A using an i5 (8^{th} generation processor), 8GB RAM (DDR4), and 8GB GPU (NVIDIA 1050Ti) with 768 CUDA cores.

6.3.1 Experimental Setup

As asserted earlier, transfer learning was utilized for training the model, *i.e.,* GoogleNet. The GoogleNet architecture, including all specifications considered in this work, is shown in Table 6.1. Table 6.2 below highlights the hyperparameters utilized in this study. Each of these parameters ranges are indicated within square brackets.

Table 6.1: Parameters of the GoogleNet model (ps: patch size, os: output size d: depth, r: reduce, incept: inception, params: parameters, ops: operations)

Type	**ps**	**os**	**d**	**#1×1**	**#3×3 r**	**#3×3**	**#5×5 r**	**#5×5**	**poll proj**	**params**	**ops**
Conv	7×7/2	112×112×64	1							2.7K	34M
max poll	3×3/2	56×56×64	0								
conv	3×3/1	56×56×192	2		64	192				112K	360M
max poll	3×3/2	28×28×192	0								
incept(3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
incept(3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max poll	3×3/2	14×14×480	0								
incept(4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
incept(4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
incept(4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
incept(4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
incept(4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max poll	3×3/2	7×7×832	0								
incept(5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
incept(5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg poll	7×7/1	1×1×1024	0								
drop .4		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

The model parameters in the pre-trained GoogleNet model are fine-tuned. All hyperparameters were selected based on experimental trials from the Table Table 6.2. The minimum batch size is set to 16, an initial learning rate to 0.001, which will be changed for each iteration and set max epochs as 100. The optimizer is set to Stochastic Gradient Descent (SGD). Also, we utilize momentum to escape the local optimum and fix it as 0.9; at the same time, the dropout ratio and weight decay are considered as 0.4 and 0.0008, respectively.

Table 6.2: Network Hyperparameter

Hyperparameter	Abbreviation	Range
Optimizer	optimizer	[RMS Prop, Adam, SGD]
Batch Size	batch_sizes	[16, 32]
Learning rate	learning_rates	[0.001, 0.003]
Momentum	momentum	[0.9, 0.95]
Drop Out	drop_out	[0.2, 0.3,0.4]

6.3.2 Dataset Description

There is no video dataset available to test and validate the multiple forgery detection methodologies. A dataset with multiple video forgeries is created in order to perform the experiment with the proposed technique. Dataset development commenced with the selection of videos from three available datasets: SULFA dataset (Qadir et al., 2012), (REWIND, 2013) dataset and (VTL, 2018). In order to generate a multiple forged video dataset, the selected videos are later exposed to multiple tamperings such as splicing, copy-move, and frame insertion, duplication, and deletion. Furthermore, some videos from the dataset have been subjected to post-processing operations like noise addition. The videos from the dataset are having a format of MP4 and AVI coded by H.264 and Motion JPEG encoding standard with different GOP length using FFmpeg software (FFmpeg, 2019). We termed it an MFVD-1. Figure 6.4 depicts a snapshot of forged video from the dataset. Description of sample tampered videos in the dataset is presented in Table 6.3 wherein FDU: frame duplication, FI: frame insertion, FDE: frame deletion, CM: copy-move, SPL: splicing.

Figure 6.4: Snapshot of forged videos

Table 6.3: Sample forged videos details (RES: Resolution)

Forged Videos	**Format**	**RES**	**Forgery**
FV_1	MP4	320×240	SPL: 9,17; CM: 15,25; FI: 31-40 & 51-60; FDU: (71,91) & (81,101); FDE: 111-140
FV_2	MP4	320×240	SPL: 9,17; CM: 15,25; FI: 31-40 & 51-60; FDU: (71,91) & (81,101); FDE: 111-140
FV_3	MP4	320×240	SPL: 9,17; CM: 15,25; FI: 31-40 & 51-60; FDU: (71,91) & (81,101); FDE: 111-140
FV_4	MP4	320×240	SPL: 9,17; CM: 15,25; FI: 31-40 & 51-60; FDU: (71,91) & (81,101); FDE: 111-140
FV_5	MP4	320×240	SPL: 9,17; CM: 15,25; FI: 31-40 & 51-60; FDU: (71,91) & (81,101); FDE: 111-140
FV_6	AVI	640×480	SPL: 15,25; CM: 5,35; FI: 41-50 & 61-70; FDU: (92,192) & (100,201); FDE:141-170
FV_7	AVI	640×480	SPL: 15,25; CM: 5,35; FI: 41-50 & 61-70; FDU: (92,192) & (100,201); FDE:141-170

FV_8	AVI	640×480	SPL: 15,25; CM: 5,35; FI: 41-50 & 61-70; FDU: (92,192) & (100,201); FDE:141-170
FV_9	AVI	640×480	SPL: 15,25; CM: 5,35; FI: 41-50 & 61-70; FDU: (92,192) & (100,201); FDE:141-170
FV_10	AVI	640×480	SPL: 15,25; CM: 5,35; FI: 41-50 & 61-70; FDU: (92,192) & (100,201); FDE:141-170

6.3.3 Experimental Evaluation

The proposed technique is successfully tested on an MFVD-1 dataset. The performance parameters, such as accuracy, recall, precision and F1-score, are tabulated in Table 6.4. Table 6.4 shows that for the copy-move and splicing forgery, the accuracy is 97.06% and 98.78%, respectively. However, for the frame insertion, frame deletion, and frame duplication, the accuracy is 98.28%, 94.30%, and 98.56%, respectively. Whereas for multiple forgeries, the accuracy reaches 97.07%. The sample detection and localization result for the test video using the proposed technique is depicted in Figure 6.5. The forgeries such as copy-move (15^{th} and 25^{th}), splicing (9^{th} and 12^{th}) and frame inertion (31-40 and 51-60), deletion (111-140) and duplication (91,101) are identified from the test video.

Table 6.4: Performance of proposed technique

Forgery Type	**Accuracy**	**Precision**	**Recall**	**F1-score**
Copy-move	97.06	96.26	93.85	95.04
Frame Splicing	98.78	98.81	95.48	97.12
Frame Insertion	98.28	98.87	95.29	97.05
Frame Deletion	94.30	93.58	93.81	93.69
Frame Duplication	98.56	97.58	94.56	96.05
Multiple Forgeries	97.07	96.82	94.40	95.59

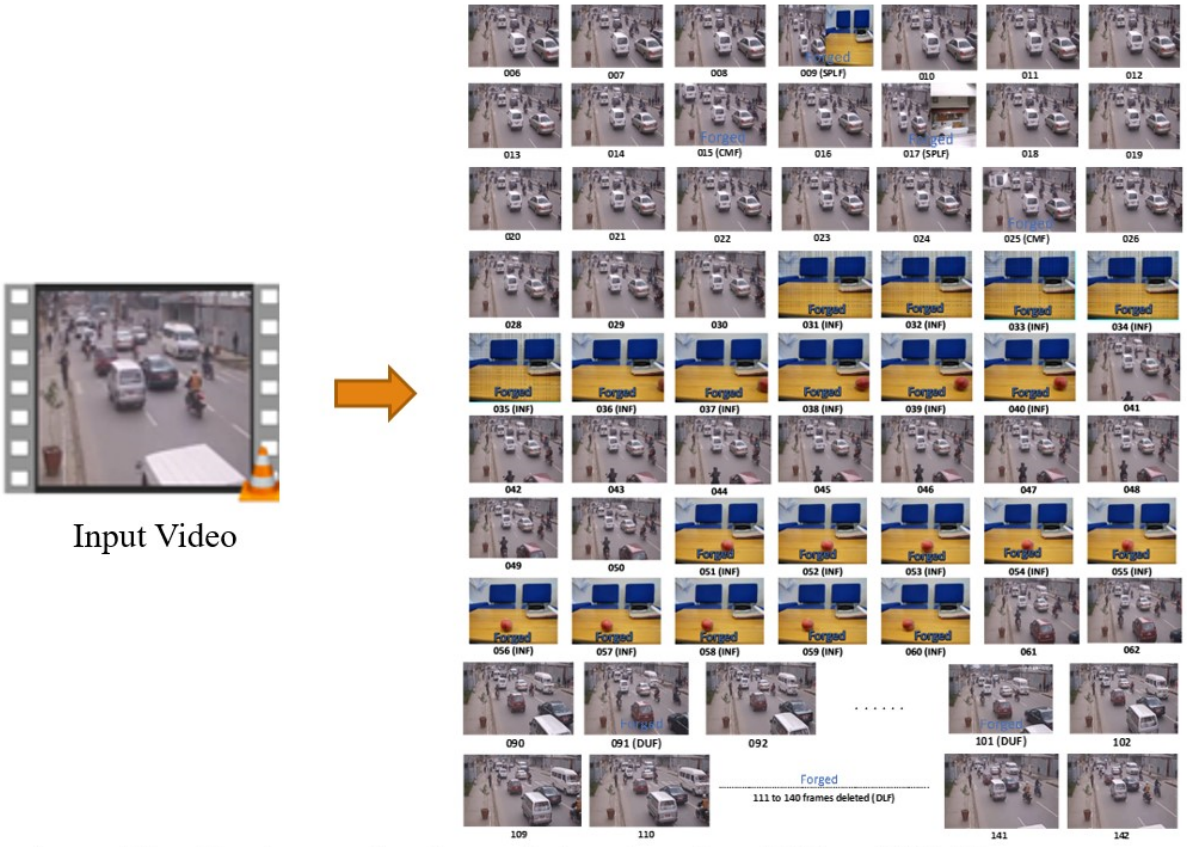


Figure 6.5: Detection and localization result

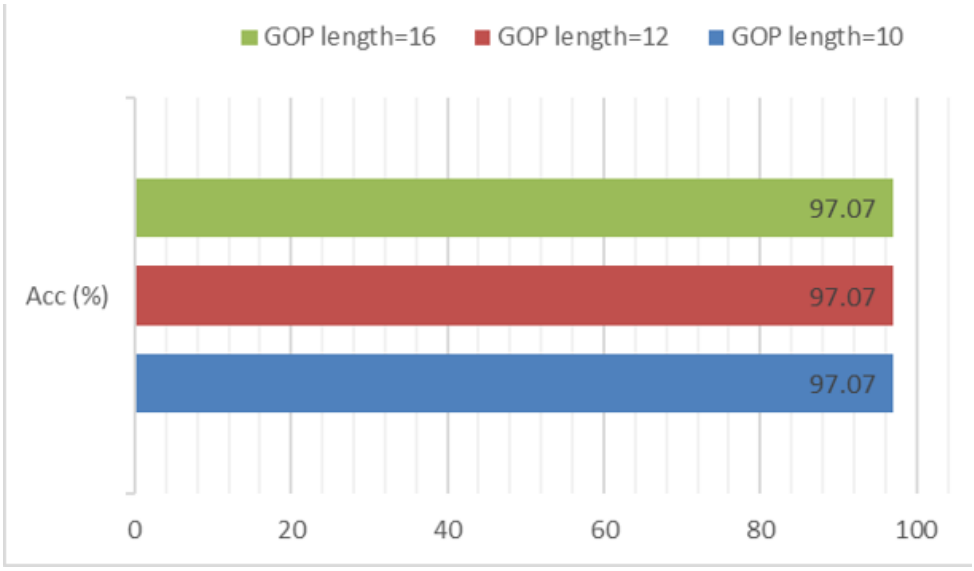


Figure 6.6: Evaluation on videos with different GOP lengths

6.3.3.1 Evaluation based on GOP Length

The proposed forgery detection technique is investigated for checking the GOP length dependency. The proposed technique is tested on the videos having variable GOP lengths *i.e.,* 10,12, and 16 and the evaluation performance against the variable GOP lengths is shown in Figure 6.6. This figure clearly shows that our technique performs efficiently and is not affected by GOP length.

6.3.3.2 Evaluation based on Background

The efficiency of the proposed technique is evaluated against video backgrounds, *i.e.,* static /moving background. It is clear from Figure 6.7, the proposed technique effectively detects multiple forgeries from the video having a different background. However, the performance is slightly impacted by a fast-moving background video.

6.3.3.3 Evaluation based on Noise

The performance of the proposed forgery detection technique is analyzed to identify the multiple forgeries against the noise robustness. For this, some videos from the dataset have been subjected to Gaussian noise addition, and after that, the performance is evaluated for a different value of noise. Figure 6.8 depicts the evaluation of the proposed technique robustness to

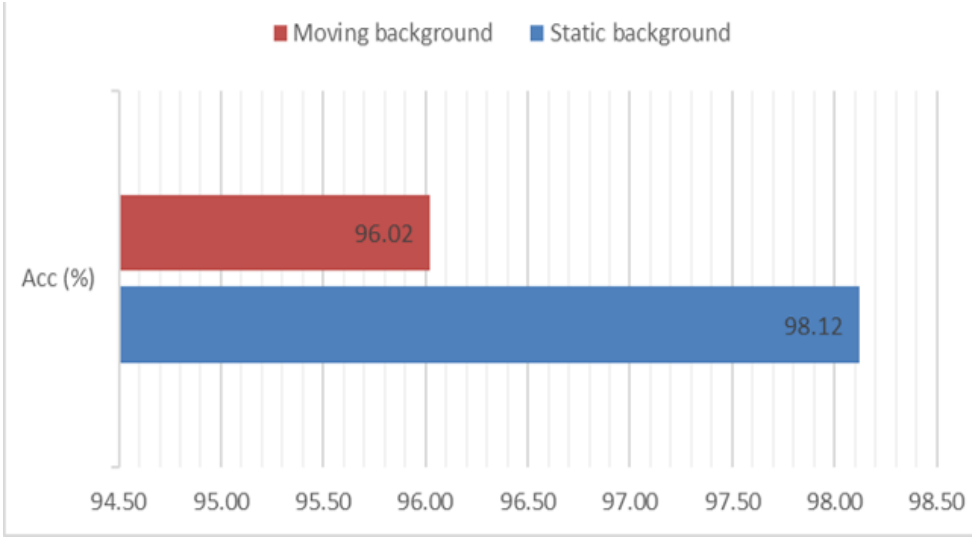


Figure 6.7: Evaluation on videos with different backgrounds

the noise. Our technique provides almost the same accuracy of 97.07% for the noise attack of 5dB, 10dB, and 15dB noise. However, even in the worst-case scenario, *i.e.,* with 15dB and 30dB noise, the technique gives the accuracy of 96% and 94% respectively. From this thorough analysis, it has been demonstrated that our technique is robust to noise addition.

6.3.4 Overall Comparison

This section compares the proposed forgery detection technique performance with the existing techniques mentioned in (Wei et al., 2019), (Bakas et al., 2019), (Aloraini et al., 2020). The comparative analysis section has selected the latest state-of-the-art, relevant techniques to compare with the proposed techniques. To the best of our knowledge, the current literature doesn't contain multiple forgery detection techniques. So we have compared our work with the state of the art techniques that can detect the single forgery at a time. In the background, we have tested the proposed and existing techniques against the single forgery detection on the multi-forged video dataset MFVD-1 for a fair comparison. The comparison has been made to show the effectiveness of our techniques in terms of accuracy, multiple-forgery detection or not, GOP and background dependency and robustness against the noise attacks. The comparison shows that the technique outperforms the existing techniques with respect to several different parameters. Mentioned existing techniques are only effective for a single sort of tampering at a time, *i.e.*, fail

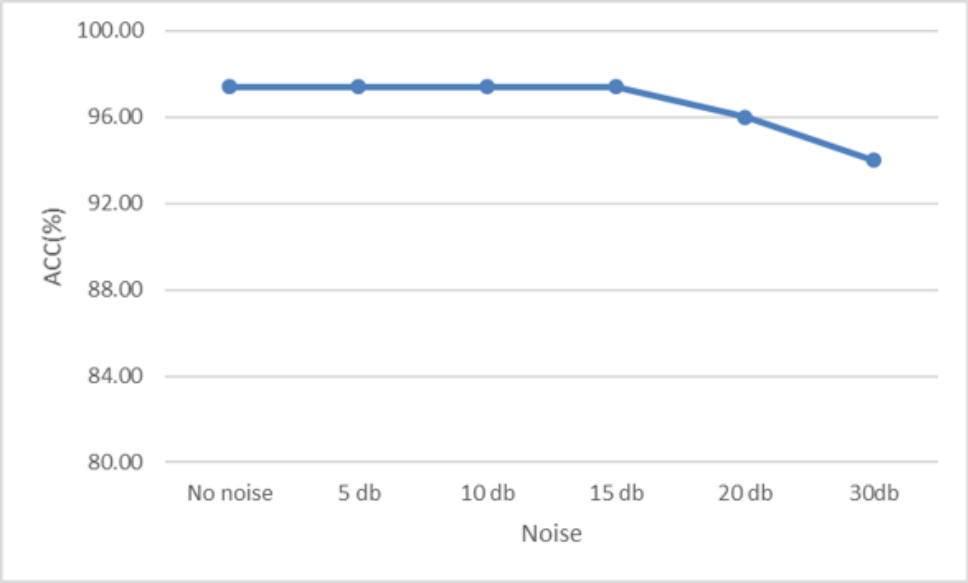


Figure 6.8: Evaluation against the noise robustness

to detect multiple forgeries. The proposed technique, on the other hand, can identify multiple forgeries. Table 6.5 shows the comparative analysis between proposed and existing techniques with respect to forgeries identified, accuracy, multiple forgery detection and localization, GOP length, background, noise robustness, and execution time, Wherein (A: Forgeries, B: Accuracy, C: Multiple Forgeries Detection and Localization, D: GOP Robustness, E: Suitable for Static(S) & Moving (M) background, F: Noise Robustness, G: Computational time/video (Sec)). The proposed technique gives an accuracy of 97.07%, which is better than the techniques mentioned. The technique given in (Aloraini et al., 2020; Wei et al., 2019) are reliant on the GOP length and background. However, our technique and approach in (Bakas et al., 2019), on the other hand, is not reliant on GOP length and background. Furthermore, when compared to existing techniques, our technique is more resistant to noise addition. Another point of comparison is the computational time. For that, the execution time for our technique and stated techniques are recorded, which is shown in Table 6.5. The technique used in (Aloraini et al., 2020) is faster than the proposed technique, but it can only detect only one forgery from individual video. However, the features used in the feature extraction process of the proposed forgery detection technique is computationally inexpensive as compared to the features employed in techniques (Bakas et al., 2019; Wei et al., 2019). Due to this, our technique is computationally effective as compared to the other techniques and can handle multiple forgeries.

Table 6.5: Comparative analysis

Techniques	**A**	**B**	**C**	**D**	**E**	**F**	**G**
(Wei et al., 2019)	FI, FD, FDU	92.86	NO	YES	S	NO	175
(Bakas et al., 2019)	FI, FD, FDU	96.52	NO	NO	S/M	NO	151
(Aloraini et al., 2020)	CM	97.02	NO	YES	S	NO	40
Proposed	CM, SPL, FI, FD, FDU	97.07	YES	NO	S/M	YES	90

6.4 Overall Comparison of Proposed Techniques

This section comprises the comparison of the proposed multiple forgeries detection techniques. Table 6.6 shows the comparison between the proposed schemes (wherein D: GOP & Background Robustness, E: Noise Robustness, F: Compression Robustness, G:Robustness against other Post-processing operations). The table shows that technique 1 can detect frame splicing, insertion, and deletion forgeries and is robust against GOP and background. Technique 2 can detect splicing, frame insertion, deletion and duplication and is robust against GOP, Background and Compression. Technique 3 can detect copy-move, splicing, insertion, deletion forgeries and is not robust against compression. Technique 4 can detect copy-move, splicing, frame insertion, deletion, duplication and is robust against GOP, background and post-processing operations.

Table 6.6: Overall comparison of Proposed techniques

Technique	**Features/ method**	**Forgery**	**ACC**	**D**	**E**	**F**	**G**
Technique 1	LBP	FS, FI, FD	97.33	YES	NO	NO	NO
Technique 2	DistrEn2D, MSE2DFI	FS, FI, FD, FDU	97.49, 96.66	YES	NO	YES	NO
Technique 3	VGG-16, KPCA	FS, CM, FI, FD	97.24	YES	YES	NO	YES
Technique 4	PCT, NBAP, GoogleNet	CM, SPL, FI, FD, FDU	97.07	YES	YES	NO	NO

Detailed analysis for the Table 6.6 is as follows:

- In the first multiple forgeries detection technique, LBP features and outlier detection approach is used. This technique can detect frame splicing, frame insertion, and deletion forgeries. The technique is tested on an MFVD-4 dataset. The experimental results show that accuracy reaches up to 97.33%. The experimental results reveal the efficiency of the proposed technique in terms of GOP and video background dependency. However, This technique is not able to detect the copy-move and frame duplication forgery. Moreover, this technique is not robust against the compression, noise and other post-processing operation like Hue and Brightness change.

- In the second multiple forgeries detection technique, entropy-based features such as DistrEn2D and MSE2D are used. This technique can examine the presence of forgeries such as intra-frame forgery, *i.e.,* splicing and inter-frame forgeries, *i.e.,* frame insertion, deletion, and duplication from the digital videos. The technique is tested on an MFVD-2 dataset. The experimental result provides 97.49% and 96.66% accuracy with DistrEn2D and MSE2D features, respectively. The advantage of this technique is that it is robust to the compression attack, GOP length, and video background. However, robustness against the noise attack is not tested.

- In the third multiple forgeries detection technique, VGG-16 neural network and KPCA is used. This technique can detect and localize the forgeries such as copy-move, splicing,

insertion, and deletion present in the individual video. The VGG-16 deep neural model is fine-tuned with various hyperparameters such as learning rate, drop out rate, optimizer and batch size in order to extract the relevant features. The technique is tested on an MFVD-3 dataset. Experimental results demonstrate the efficiency of the proposed scheme in terms of robustness against several post-processing operations (like noise, hue and brightness changes), GOP structure, and background. The proposed technique provides an accuracy of 97.24%. However, This technique cannot detect the frame duplication forgery and is not robust to the compression attack.

- In the fourth multiple forgeries detection technique, PCT, NBAP, and GoogleNet models is used. This technique can handle the presence of intra-frame forgeries, *i.e.,* copy-move and splicing, as well as inter-frame forgeries, *i.e.,* frame insertion, deletion, and duplication from the digital videos. The model parameters in the pre-trained GoogleNet model are fine-tuned. All hyperparameters such as optimizer, learning rate, Momentum, drop out rate, and batch size was selected based on experimental trials to build the model. The technique is tested on an MFVD-1 dataset. The proposed technique provides an accuracy of 97.07%. Experimental results demonstrate the efficiency of the proposed scheme in terms of robustness against noise attack, GOP structure length, and video background. However, robustness against the compression attack is not tested. The advantage of this technique it can identify the five types of forgeries. This technique is better than all the above-proposed techniques in terms of the number of forgeries detected.

6.5 Conclusion

In this Chapter, a forgery detection technique for the video is proposed using PCT, NBAP, and GoogleNet models. This technique could handle the presence of intra-frame forgeries, *i.e.,* copy-move and splicing, as well as inter-frame forgeries, *i.e.,* frame insertion, deletion, and duplication from the digital videos. This technique was tested on an MFVD-1 dataset. The proposed technique provides an accuracy of 97.07%. Experimental results illustrated the capability of the proposed scheme in terms of noise attack robustness, GOP structure length, and video background.

CHAPTER 7

Conclusion and Future Scope

7.1 Conclusion

The entire focus of the thesis is on the development of passive techniques for detecting and localizing multiple forgeries in the videos. A multiple forgery detection technique for digital video using LBP features is proposed that calculates correlations between LBP-coded frames followed by an outlier detection approach. The technique is tested on an MFVD-4 dataset. This technique can detect frame splicing, frame insertion, and deletion forgeries. The experimental results show that accuracy reaches up to 97.33%, and the technique is efficient in terms of GOP and video background dependency. This technique is not able to detect the copy-move and frame duplication forgery. Moreover, this technique is not robust against the compression attack, noise and other post-processing operations like hue and brightness change. Thus, a multiple forgeries detection technique for video based on correlation consistency between entropy-coded frames is proposed that uses entropy-based features like DistrEn2D and MSE2D.

The technique is tested on an MFVD-2 dataset. The experimental result provides 97.49% and 96.66% accuracy with DistrEn2D and MSE2D features, respectively. This technique can examine the presence of forgeries such as intra-frame forgery, *i.e.,* splicing and inter-frame forgeries, *i.e.,* frame insertion, deletion, and duplication from the digital videos. The technique is robust to the compression attack, GOP length, and video background. However, robustness against the noise attack is not tested. Another forgery detection technique for digital video using VGG-16 neural Network and KPCA is proposed to detect and localize the forgeries such as copy-move, splicing, insertion, and deletion present in the individual video. Experimental results demonstrate the efficiency of the proposed scheme in terms of robustness against several post-processing operations like noise, hue and brightness changes; GOP structure, and background. The proposed technique provides an accuracy of 97.24%. The use of deep neural networks to extract the features automatically in comparison to the handcrafted feature extraction in the previous two techniques helps this technique to outperform the previously proposed two schemes. However, this technique is not able to detect the frame duplication forgery and is not robust to the compression attack. Thus a multiple forgery detection and localization technique using PCT, NBAP and GoogleNet model is developed to detect and locate the forgeries in the video. It can identify the presence of intra-frame forgeries, *i.e.,* copy-move and splicing, as well as inter-frame forgeries, *i.e.,* frame insertion, deletion, and duplication from the digital videos. Experimental results demonstrate the efficiency of the proposed scheme in terms of robustness against noise attack, GOP structure length, and video background. The accuracy of this technique is 97.07%. This technique can identify the five types of forgeries, and it proves to be better than the previously proposed techniques in terms of the number of forgeries detected.

7.2 Future Work

This work has focused on passive forgery detection techniques for digital video. Four different techniques have been proposed to reduce the gaps in existing techniques. However, there is still have a number of areas for of improvements. This section discusses possible future work in the field of video forensics:

- More research is needed to detect and identify other complex video forgeries from individual videos.
- Deepfake detection in digital videos is currently attracting a lot of attention in the video forensics.
- The need to address the limitation of the proposed techniques as they are not suitable for real-time video streaming.
- Besides the visual content of the digital video, the audio component also plays an important role in legal matters. The proposed techniques concentrated only on visual content, *i.e.,* scope to the audio content of the video is open for the research.

References

Aghamaleki, J. A., & Behrad, A. (2016). Inter-frame video forgery detection and localization using intrinsic effects of double compression on quantization errors of video coding. *Signal Processing: Image Communication*, *47*(1), 289–302.

Aghamaleki, J. A., & Behrad, A. (2017). Malicious inter-frame video tampering detection in mpeg videos using time and spatial domain analysis of quantization effects. *Multimedia Tools and Applications*, *76*(20), 20691–20717.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *In Proceedings of International Conference on Engineering and Technology in Turkey*, 1–6.

Aloraini, M., Sharifzadeh, M., Agarwal, C., & Schonfeld, D. (2019). Statistical sequential analysis for object-based video forgery detection. *Electronic Imaging*, *19*(5), 543–550.

Aloraini, M., Sharifzadeh, M., & Schonfeld, D. (2020). Sequential and patch analyses for object removal video forgery detection and localization. *IEEE Transactions on Circuits and Systems for Video Technology*, *31*(3), 917–930.

Aparicio-Díaz, E., Cumplido, R., Gort, P., Lázaro, M., & Feregrino-Uribe, C. (2019). Temporal copy-move forgery detection and localization using block correlation matrix. *Journal of Intelligent & Fuzzy Systems*, *36*(5), 5023–5035.

Azami, H., Escudero, J., & Humeau-Heurtier, A. (2017). Bidimensional distribution entropy to analyze the irregularity of small-sized textures. *IEEE Signal Processing Letters*, *24*(9), 1338–1342.

Bagiwa, M. A., Wahab, A. W. A., Idris, M. Y. I., Khan, S., & Choo, K.-K. R. (2016). Chroma key background detection for digital video using statistical correlation of blurring artifact. *Digital Investigation*, *19*(1), 29–43.

Bai, S., Yao, H., Ni, R., & Zhao, Y. (2019). Detection and localization of video object removal by spatio-temporal lbp coherence analysis. *In Proceedings of International Conference on Image and Graphics*, 244–254.

Bakas, J., Naskar, R., & Dixit, R. (2019). Detection and localization of inter-frame video forgeries based on inconsistency in correlation distribution between haralick coded frames.

Multimedia Tools and Applications, *78*(4), 4905–4935.

Bestagini, P., Milani, S., Tagliasacchi, M., & Tubaro, S. (2013). Local tampering detection in video sequences. *In Proceedings of IEEE International Workshop on Multimedia Signal Processing in Italy*, 488–493.

Bidokhti, A., & Ghaemmaghami, S. (2015). Detection of regional copy/move forgery in mpeg videos using optical flow. *In Proceedings of International Symposium on Artificial Intelligence and Signal Processing in Iran*, 13–17.

Birajdar, G. K., & Mankar, V. H. (2013). Digital image forgery detection using passive techniques: A survey. *Digital Investigation*, *10*(3), 226–245.

BOZKURT, I., Bozkurt, M. H., & ULUTAŞ, G. (2017). A new video forgery detection approach based on forgery line. *Turkish Journal of Electrical Engineering & Computer Sciences*, *25*(6), 4558–4574.

Chao, J., Jiang, X., & Sun, T. (2012). A novel video inter-frame forgery model detection scheme based on optical flow consistency. *In Proceedings of International Workshop on Digital Watermarking*, 267–281.

Chen, C., Chen, L.-Y., & Lin, Y.-J. (2017). Block sampled matching with region growing for detecting copy-move forgery duplicated regions. *Journal of Information Hiding and Multimedia Signal Processing*, *8*(1), 86–96.

Chen, R., Dong, Q., Ren, H., & Fu, J. (2012). Video forgery detection based on non-subsampled contourlet transform and gradient information. *Information Technology Journal*, *11*(10), 1456–1462.

Chen, S., Tan, S., Li, B., & Huang, J. (2016). Automatic detection of object-based forgery in advanced video. *IEEE Transactions on Circuits and Systems for Video Technology*, *26*(11), 2138-2151.

Chetty, G., Biswas, M., & Singh, R. (2010). Digital video tamper detection based on multimodal fusion of residue features. *In Proceedings of International Conference on Network and System Security in Australia*, 606–613.

Chittapur, G. B., Murali, S., Prabhakara, H., & Anami, B. S. (2014). Exposing digital forgery in video by mean frame comparison techniques. *In Proceedings of Springer Emerging Research in Electronics, Computer Science and Technology*, 557–562.

Clideo. (2020). (Software available online at:: `https://clideo.com/editor/adjust-video/`)

Costa, M., Goldberger, A. L., & Peng, C.-K. (2002). Multiscale entropy analysis of complex physiologic time series. *Physical Review Letters*, *89*(6), 1-4.

D'Amiano, L., Cozzolino, D., Poggi, G., & Verdoliva, L. (2018). A patchmatch-based dense-field algorithm for video copy–move detection and localization. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(3), 669–682.

D'Avino, D., Cozzolino, D., Poggi, G., & Verdoliva, L. (2017). Autoencoder with recurrent neural networks for video forgery detection. *Electronic Imaging*, *17*(7), 92–99.

Dong, Q., Yang, G., & Zhu, N. (2012). A mcea based passive forensics scheme for detecting frame-based video tampering. *Digital Investigation*, *9*(2), 151–159.

Fadl, S., Han, Q., & Qiong, L. (2020). Exposing video inter-frame forgery via histogram of oriented gradients and motion energy image. *Multidimensional Systems and Signal Processing*, *31*(4), 1365–1384.

Fadl, S. M., Han, Q., & Li, Q. (2018). Authentication of surveillance videos: detecting frame duplication based on residual frame. *Journal of forensic sciences*, *63*(4), 1099–1109.

Fan, Y., Zhu, Y.-S., & Liu, Z. (2016). An improved sift-based copy-move forgery detection method using t-linkage and multi-scale analysis. *Journal of Information Hiding and Multimedia Signal Processing*, *7*(2), 399–408.

Fayyaz, M. A., Anjum, A., Ziauddin, S., Khan, A., & Sarfaraz, A. (2020). An improved surveillance video forgery detection technique using sensor pattern noise and correlation of noise residues. *Multimedia Tools and Applications*, *79*(9), 5767–5788.

Feng, C., Xu, Z., Zhang, W., & Xu, Y. (2014). Automatic location of frame deletion point for digital video forensics. *In Proceedings of ACM workshop on Information Hiding and Multimedia Security*, 171–179.

FFmpeg. (2019). (Software available online at:: `https://www.ffmpeg.org/`)

Gironi, A., Fontani, M., Bianchi, T., Piva, A., & Barni, M. (2014). A video forensic technique for detecting frame deletion and insertion. *In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing in Italy*, 6226–6230.

Güneş, A., Kalkan, H., & Durmuş, E. (2016). Optimizing the color-to-grayscale conversion for image classification. *Signal, Image and Video Processing*, *10*(5), 853–860.

Harit, G., & Chaudhury, S. (2007). Video shot characterization using principles of perceptual prominence and perceptual grouping in spatio–temporal domain. *IEEE Transactions on Circuits and Systems for Video Technology*, *17*(12), 1728–1741.

Ho, A. T., & Li, S. (2015). Handbook of digital forensics of multimedia data and devices. *John Wiley & Sons*, 1–653.

Hong, J. H., Yang, Y., & Oh, B. T. (2019). Detection of frame deletion in hevc-coded video in the compressed domain. *Digital Investigation*, *30*(1), 23–31.

Hu, X., Ni, J., & Pan, R. (2015). Detecting video forgery by estimating extrinsic camera parameters. *In Proceedings of International Workshop on Digital Watermarking*, 28–38.

Hu, Y., Li, C.-T., Wang, Y., & Liu, B.-B. (2012). An improved fingerprinting algorithm for detection of video frame duplication forgery. *International Journal of Digital Crime and Forensics*, *4*(3), 20–32.

Hyun, D.-K., Lee, M.-J., Ryu, S.-J., Lee, H.-Y., & Lee, H.-K. (2013). Forgery detection for surveillance video. *In Proceedings of Era of Interactive Media, Springer*, 25–36.

Jaiswal, A. K., & Srivastava, R. (2020). A technique for image splicing detection using hybrid feature set. *Multimedia Tools and Applications*, *79*(17), 11837–11860.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *In Proceedings of ACM International Conference on Multimedia*, 675–678.

Johnston, P., & Elyan, E. (2019). A review of digital video tampering: from simple editing to full synthesis. *Digital Investigation*, *29*(1), 67–81.

Johnston, P., Elyan, E., & Jayne, C. (2020). Video tampering localisation using features learned from authentic content. *Neural computing and applications*, *32*(16), 12243–12257.

Kharat, J., & Chougule, S. (2020). A passive blind forgery detection technique to identify frame duplication attack. *Multimedia Tools and Applications*, *79*(11), 8107–8123.

Kingra, S., Aggarwal, N., & Singh, R. D. (2017). Inter-frame forgery detection in h. 264 videos using motion and brightness gradients. *Multimedia Tools and Applications*, *76*(24), 25767–25786.

Kobayashi, M., Okabe, T., & Sato, Y. (2009). Detecting video forgeries based on noise characteristics. *In Proceedings of Pacific-Rim Symposium on Image and Video Technology*, 306–317.

Kobayashi, M., Okabe, T., & Sato, Y. (2010). Detecting forgery from static-scene video based on inconsistency in noise level functions. *IEEE Transactions on Information Forensics and Security*, *5*(4), 883-892.

Kono, K., Yoshida, T., Ohshiro, S., & Babaguchi, N. (2018). Passive video forgery detection

considering spatio-temporal consistency. *In Proceedings of International Conference on Soft Computing and Pattern Recognition*, 381–391.

Labartino, D., Bianchi, T., De Rosa, A., Fontani, M., Vázquez-Padín, D., Piva, A., & Barni, M. (2013). Localization of forgeries in mpeg-2 video through gop size and dq analysis. *In Proceedings of IEEE International Workshop on Multimedia Signal Processing in Italy*, 494–499.

Lee, J.-M., Yoo, C., Choi, S. W., Vanrolleghem, P. A., & Lee, I.-B. (2004). Nonlinear process monitoring using kernel principal component analysis. *Chemical engineering science*, *59*(1), 223–234.

Leys, C., Ley, C., Klein, O., Bernard, P., & Licata, L. (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, *49*(4), 764–766.

Li, F., & Huang, T. (2014). Video copy-move forgery detection and localization based on structural similarity. *In Proceedings of International Conference on Multimedia Technology*, 63–76.

Li, L. (2012). Geometrically invariant image watermarking using polar harmonic transforms. *Information Sciences*, *199*(1), 1–19.

Li, L., Wang, X., Zhang, W., Yang, G., & Hu, G. (2012). Detecting removed object from video with stationary background. *In Proceedings of International Workshop on Digital Watermarking*, 242–252.

Li, Z., Zhang, Z., Guo, S., & Wang, J. (2016, November). Video inter-frame forgery identification based on the consistency of quotient of mssim. *Security and Communication Networks*, *9*(17), 4548–4556.

Liao, S.-Y., & Huang, T.-Q. (2013). Video copy-move forgery detection and localization based on tamura texture features. *In Proceedings of IEEE International Congress on Image and Signal Processing in China*, 864–868.

Lin, G.-S., & Chang, J.-F. (2012). Detection of frame duplication forgery in videos based on spatial and temporal analysis. *In Proceedings of International Journal of Pattern Recognition and Artificial Intelligence*, *26*(07), 1250017.

Liu, H., Li, S., & Bian, S. (2014). Detecting frame deletion in h. 264 video. *In Proceedings of International Conference on Information Security Practice and Experience*, 262–270.

Liu, Q., Sung, A. H., Chen, Z., & Xu, J. (2008). Feature mining and pattern classification

for steganalysis of lsb matching steganography in grayscale images. *Pattern Recognition*, *41*(1), 56–66.

Liu, Y., & Huang, T. (2017). Exposing video inter-frame forgery by zernike opponent chromaticity moments and coarseness analysis. *Multimedia Systems*, *23*(2), 223–238.

Long, C., Smith, E., Basharat, A., & Hoogs, A. (2017). A c3d-based convolutional neural network for frame dropping detection in a single video shot. *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops in USA*, 1898–1906.

Martinez, A. M., & Kak, A. C. (2001). Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(2), 228–233.

Mathai, M., Rajan, D., & Emmanuel, S. (2016). Video forgery detection and localization using normalized cross-correlation of moment features. *In Proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation in USA*, 149–152.

Miller, J. (1991). Reaction time analysis with outlier exclusion: Bias varies with sample size. *Quarterly Journal of Experimental Psychology*, *43*(4), 907–912.

Mondaini, N., Caldelli, R., Piva, A., Barni, M., & Cappellini, V. (2007). Detection of malevolent changes in digital video for forensic applications. *In Proceedings of Security, Steganography, and Watermarking of Multimedia Contents IX*, 65050-65061.

NIMBLE. (2017). (Nimble Challenge Dataset: [Online] `https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation`)

Pandey, R. C., Singh, S. K., & Shukla, K. (2014). Passive copy-move forgery detection in videos. *In Proceedings of IEEE International Conference on Computer and Communication Technology, in India*, 301–306.

Pandey, R. C., Singh, S. K., & Shukla, K. K. (2016). Passive forensics in image and video using noise features: A review. *Digital Investigation*, *19*, 1–28.

Pu, H., Huang, T., Guo, G., Weng, B., & You, L. (2019). Video tampering detection algorithm based on spatial constraints and stable feature. *UK Workshop on Computational Intelligence*, 541–553.

Qadir, G., Yahaya, S., & Ho, A. T. S. (2012). Surrey university library for forensic analysis (sulfa) of video content. *In Proceedings IET Conference on Image Processing*, 1-6.

Ravi, H., Subramanyam, A. V., Gupta, G., & Kumar, B. A. (2014). Compression noise based video forgery detection. *In Proceedings of IEEE International Conference on Image Processing in France*, 5352–5356.

REWIND. (2013). (Datset: [Online]. https://sites.google.com/site/rewindpolimi/downloads/datasets/video-copy-move-forgeries-datase)

Saddique, M., Asghar, K., Bajwa, U. I., Hussain, M., & Habib, Z. (2019). Spatial video forgery detection and localization using texture analysis of consecutive frames. *Advances in Electrical and Computer Engineering*, *19*(3), 97–108.

Shanableh, T. (2013). Detection of frame deletion for digital video forensics. *Digital Investigation*, *10*(4), 350–360.

Silva, L. E., Duque, J. J., Felipe, J. C., Murta Jr, L. O., & Humeau-Heurtier, A. (2018). Two-dimensional multiscale entropy analysis: Applications to image texture evaluation. *Signal Processing*, *147*(1), 224–232.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, R. D., & Aggarwal, N. (2017a). Detection and localization of copy-paste forgeries in digital videos. *Forensic Science International*, *281*, 75–91.

Singh, R. D., & Aggarwal, N. (2017b). Detection of upscale-crop and splicing for digital video authentication. *Digital Investigation*, *21*(1), 31–52.

Singh, R. D., & Aggarwal, N. (2017c). Optical flow and prediction residual based hybrid forensic system for inter-frame tampering detection. *Journal of Circuits, Systems and Computers*, *26*(07), 1750107.

Singh, V. K., Pant, P., & Tripathi, R. C. (2015). Detection of frame duplication type of forgery in digital video using sub-block based features. *In Proceedings of International Conference on Digital Forensics and Cyber Crime*, 29–38.

Sitara, K., & Mehtre, B. (2017). A comprehensive approach for exposing inter-frame video forgeries. *In Proceedings of IEEE International Colloquium on Signal Processing & its Applications in Malaysia*, 73–78.

Su, L., Huang, T., & Yang, J. (2015). A video forgery detection algorithm based on compressive sensing. *Multimedia Tools and Applications*, *74*(17), 6641–6656.

Su, L., & Li, C. (2018). A novel passive forgery detection algorithm for video region duplication. *Multidimensional Systems and Signal Processing*, *29*(3), 1173–1190.

Su, L., Li, C., Lai, Y., & Yang, J. (2018). A fast forgery detection algorithm based on exponential-fourier moments for video region duplication. *"IEEE Transactions on Multimedia"*, *20*(4), 825-840.

Su, L., Luo, H., & Wang, S. (2019). A novel forgery detection algorithm for video foreground removal. *IEEE Access*, *7*(1), 109719–109728.

Su, Y., Zhang, J., & Liu, J. (2009). Exposing digital video forgery by detecting motion-compensated edge artifact. *In Proceedings of International Conference on Computational Intelligence and Software Engineering in China*, 1–4.

Subramanyam, A. V., & Emmanuel, S. (2012). Video forgery detection using hog features and compression properties. *In Proceedings of IEEE International Workshop on Multimedia Signal Processing in Canada*, 89–94.

Subramanyam, A. V., & Emmanuel, S. (2013). Pixel estimation based video forgery detection. *In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing in Canada*, 3038–3042.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., & Rabinovich, A. (2015). Going deeper with convolutions. *In Proceedings of In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.

Tamura, H., Mori, S., & Yamawaki, T. (1978). Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, *8*(6), 460–473.

Tan, S., Chen, S., & Li, B. (2015). Gop based automatic detection of object-based forgery in advanced video. *In Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference in China*, 719–722.

Tralic, D., Grgic, S., & Zovko-Cihlar, B. (2014). Video frame copy-move forgery detection based on cellular automata and local binary patterns. *In Proceedings of International Symposium on Telecommunications in Herzegovina*, 1–4.

Ulutas, G., Ustubioglu, B., Ulutas, M., & Nabiyev, V. (2017). Frame duplication/mirroring detection method with binary features. *IET Image Processing*, *11*(5), 333-342.

Ulutas, G., Ustubioglu, B., Ulutas, M., & Nabiyev, V. V. (2018). Frame duplication detection based on bow model. *Multimedia Systems*, *24*(5), 549–567.

Upadhyay, P., & Chhabra, J. K. (2021). Multilevel thresholding based image segmentation using new multistage hybrid optimization algorithm. *Journal of Ambient Intelligence and Humanized Computing*, *12*(1), 1081–1098.

VTD. (2017). (Video Tampering Dataset: [Online] `https://www.youtube.com/channel/UCZuuu-iyZvPptbIUHT9tMrA`)

VTL. (2018). (Video Trace Library: [Online] `http://trace.eas.asu.edu/`)

Wang, Q., Li, Z., Zhang, Z., & Ma, Q. (2014a). Video inter-frame forgery identification based on consistency of correlation coefficients of gray values. *Journal of Computer and Communications*, *2*(04), 51-57.

Wang, Q., Li, Z., Zhang, Z., & Ma, Q. (2014b). Video inter-frame forgery identification based on optical flow consistency. *Sensors & Transducers*, *166*(3), 229–234.

Wang, W., & Farid, H. (2006). Exposing digital forgeries in video by detecting double mpeg compression. *In Proceedings of Workshop on Multimedia and Security*, 37–47.

Wang, W., & Farid, H. (2007). Exposing digital forgeries in video by detecting duplication. *In Proceedings of Workshop on Multimedia & security*, 35–42.

Wang, W., & Farid, H. (2009). Exposing digital forgeries in video by detecting double quantization. *In Proceedings of ACM workshop on Multimedia and security*, 39–48.

Wang, W., Jiang, X., Wang, S., Wan, M., & Sun, T. (2013). Identifying video forgery process using optical flow. *In Proceedings of International Workshop on Digital Watermarking*, 244–257.

Wei, W., Fan, X., Song, H., & Wang, H. (2019). Video tamper detection based on multi-scale mutual information. *Multimedia Tools and Applications*, *78*(19), 27109–27126.

Wu, Y., Jiang, X., Sun, T., & Wang, W. (2014). Exposing video inter-frame forgery based on velocity field consistency. *In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing in Italy*, 2674–2678.

Xu, J., Liang, Y., Tian, X., & Xie, A. (2016). A novel video inter-frame forgery detection method based on histogram intersection. *In Proceedings of IEEE International Conference on Communications in China*, 1–6.

Yang, C.-N., Ouyang, J.-F., & Harn, L. (2012). Steganography and authentication in image sharing without parity bits. *Optics Communications*, *285*(7), 1725–1735.

Yang, J., Huang, T., & Su, L. (2016). Using similarity analysis to detect frame duplication forgery in videos. *Multimedia Tools and Applications*, *75*(4), 1793–1811.

Yao, Y., Shi, Y., Weng, S., & Guan, B. (2017). Deep learning for detection of object-based forgery in advanced video. *Symmetry*, *10*(1), 3–12.

YFCC. (2019). (Datset: [Online]. `http://www.yfcc100m.org`)

Yin, L., Bai, Z., & Yang, R. (2014). Video forgery detection based on nonnegative tensor factorization. *In Proceedings of IEEE International Conference on Information Science and Technology in China*, 148–151.

Yu, L., Wang, H., Han, Q., Niu, X., Yiu, S.-M., Fang, J., & Wang, Z. (2016). Exposing frame deletion by detecting abrupt changes in video streams. *Neurocomputing*, *205*(1), 84–91.

Zampoglou, M., Markatopoulou, F., Mercier, G., Touska, D., Apostolidis, E., Papadopoulos, S., ... Kompatsiaris, I. (2019). Detecting tampered videos with multimedia forensics and deep learning. *In Proceedings of International Conference on Multimedia Modeling*, 374–386.

Zhang, J., Ho, A. T., Qiu, G., & Marziliano, P. (2007). Robust video watermarking of h. 264/avc. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *54*(2), 205–209.

Zhang, Z., Hou, J., Li, Z., & Li, D. (2015). Inter-frame forgery detection for static-background video based on mvp consistency. *In Proceedings of International Workshop on Digital Watermarking*, 94–106.

Zhao, D.-N., Wang, R.-K., & Lu, Z.-M. (2018). Inter-frame passive-blind forgery detection for video shot based on similarity analysis. *Multimedia Tools and Applications*, *77*(19), 25389–25408.

Zheng, L., Sun, T., & Shi, Y.-Q. (2014). Inter-frame video forgery detection based on block-wise brightness variance descriptor. *In Proceedings of International Workshop on Digital Watermarking*, 18–30.