

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University of Echahid Cheikh Larbi Tebessi – Tebessa
Faculty of Exact Sciences and Natural and Life Sciences
Department of Computer Science

MASTER'S THESIS

In Partial Fulfillment of the Requirements for the Degree of Master
Specialization: Networks and Computer Security

Video Tampering Detection Based on Deep Learning Techniques

Presented by: Rais Tameur

Academic Supervisor: Dr. Sahraoui Abdelatif

Academic Year: 2024-2025

Thesis Defense Date: June 10, 2025

Jury Members:

Prof. Mkhaznia Tahar	Jury President
Prof. Houam Mohamed Yassin	Examiner
Dr. Sahraoui Abdelatif	Academic Supervisor
Dr. Khalifa Boujemaa	Co-supervisor

Acknowledgment

I would like to express my deepest thanks and heartfelt gratitude to everyone who contributed, directly or indirectly, to the completion of this master thesis.

First and foremost, I wish to extend my profound love, appreciation, and sincere thanks to my beloved parents, who have been the pillars and the foundation of my life throughout this journey. Their endless patience, unconditional love, and constant encouragement have given me the strength and determination to keep moving forward.

I also offer my sincere gratitude to all my esteemed professors in Computer Science department at Sheikh Larbi Tebessi University. They enriched me with their knowledge and expertise, and stood by me during two academic years full of challenges and achievements. I am equally grateful to the honorable members of the examination committee for their valuable time and efforts in evaluating this work.

Special thanks go to my dear friends, who have been my companions on this path, supporting me with their positive spirit and constructive cooperation. They shared with me moments of tension and joy, and their presence was a source of strength and reassurance throughout this journey.

Finally, I express my deepest thanks and appreciation to my supervisor and mentor, Mr. Sahraoui Abdelatif, for his continuous support, wise guidance, and great patience, which had a profound impact on the success of this research.

This work is the result of a long effort, filled with learning, dedication, and passion. I am proud to present it and dedicate it to all those who love us and whom we love, and who stood by us and supported us.

As Isaac Newton once said: ***"Science is never complete; it is a continuous endeavor."***

شكر وتقدير

أتقدم بكامل الشكر والامتنان لكل من ساهم، بشكل مباشر أو غير مباشر، في إنجاز هذا البحث.

أولاً وقبل كل شيء، أود أن أعبر عن جزيل شكري وحبي وامتناني الكبير لوالديّ الحبيبين، اللذين كانا السند والعماد في حياتي وطوال مسيرتي، بصبرهما وحبهما اللا محدود، وتشجيعهما المتواصل الذي منحني القوة والإصرار على مواصلة هذا الطريق.

كما أتقدم بخالص الشكر والامتنان لجميع أساتذتي الكرام في قسم الإعلام الالي بجامعة الشهيد الشيخ العربي التبسي، الذين أغنوني بعلمهم وخبرتهم، ووقفوا معي خلال عامين دراسيين مليئين بالتحديات والنجاحات. وأتوجه بالشكر الجزيل لأعضاء لجنة المناقشة الموقرين على وقتهم الثمين وجهودهم القيمة في تقييم هذا العمل.

وأخص بالشكر أصدقائي الأعزاء، الذين كانوا رفقاء دربي، داعمين بروحهم الإيجابية وتعاونهم البنّاء، وشاركوا معي لحظات التوتر والفرح، فكان وجودهم نبعاً من القوة والطمأنينة طوال هذه الرحلة.

وفي الختام، أوجه جزيل شكري وتقديري لأستاذي المشرف والمؤطر، الأستاذ الصحراوي عبد اللطيف، على دعمه المستمر، وتوجيهاته الحكيمة، وصبره الكبير، الذي كان له الأثر البالغ في نجاح هذا البحث.

هذا العمل هو ثمرة جهد طويل، مليء بالتعلم والاجتهاد والشغف. أفتخر بأن أقدمه، وأهديه لكل من يحبوننا ونحبهم ووقفوا إلى جانبنا وساندونا.

كما قال إسحاق نيوتن: **"إن العلم لا يكتمل أبدًا، بل هو مسعى مستمر."**

Abstract

Dashcams have become standard in modern vehicles, capturing continuous video footage that can serve as vital digital evidence in legal disputes or insurance claims. However, the reliability of these recordings is threatened by various forms of **video tampering**, including **video forgery**, **frame deletion**, **frame insertion**, and even **deepfake**-based alterations. These challenges demand robust and intelligent **forgery detection** systems capable of identifying manipulations at both frame and video levels.

This master thesis presents a **deep learning**-based approach for automated **video tampering detection** in dashcam recordings. It begins with an in-depth theoretical analysis of **digital video forgery**, outlining its common forms and the limitations of traditional forensic methods. The study then focuses on the use of **Convolutional Neural Networks (CNNs)**, known for their effectiveness in **visual anomaly detection**, and proposes a novel detection pipeline built upon the **ResNet50** architecture.

The main contribution is the implementation of a custom model named **BinaryCNN**, a CNN-based architecture derived from **ResNet50** and fine-tuned to classify dashcam videos into "original" or "forged" (specifically for **frame deletion forgery**). The detection pipeline includes **video preprocessing**, **frame extraction**, individual **frame classification**, and score aggregation to determine video-level authenticity.

The system was trained and evaluated using the publicly available **Video Forgery Dataset** (Kaggle), which contains thousands of dashcam videos, divided into classes representing different **tampering techniques**: **frame deletion**, insertion, duplication, flipping, rotation, and zooming. For this work, we focused on a **binary classification** task: detecting **frame deletion** forgery vs. original. The dataset includes 1181 videos per class, split evenly between training and testing sets, and covers diverse driving conditions.

The proposed **BinaryCNN** model achieved an overall **accuracy of 79%**, with a **precision of 81%**, **recall of 78%**, and **F1-score of 79%**, demonstrating its effectiveness in identifying **forged videos** under real-world conditions. These results confirm the capability of CNN-based models to tackle **video forgery detection** with high reliability.

Additionally, a user-friendly web interface was developed using Flask to demonstrate the real-time application of this **deep learning video tampering detection** system. Users can upload a dashcam video and instantly receive a classification result.

This research contributes to the growing field of **automated forgery detection**

and highlights the potential of **deep learning techniques**, especially **ResNet50**-based **CNNs**, in building scalable, intelligent, and robust systems for **video tampering detection**.

key words: video tampering, video forgery, digital video forgery, video tampering detection, deepfake , ResNet50, Convolutional Neural Networks (CNNs), BinaryCNN, binary classification, deep learning, deep learning techniques, automated forgery detection.

Résumé

Les dashcams sont devenues courantes dans les véhicules modernes, enregistrant des séquences vidéo continues qui peuvent constituer des preuves numériques cruciales dans des contextes juridiques ou d'assurance. Toutefois, ces enregistrements sont de plus en plus exposés à des formes avancées de **falsification vidéo**, telles que la **supression d'images**, l'**insertion de séquences**, les **deepfakes** ou d'autres techniques de **manipulation vidéo**.

Ce travail présente une approche de **détection automatique de falsification vidéo** basée sur le **deep learning**, appliquée aux vidéos enregistrées par dashcam. Après une étude approfondie des différentes formes de **falsification vidéo** et des limites des méthodes traditionnelles, nous proposons un pipeline de détection basé sur les **réseaux de neurones convolutifs (CNN)**, en particulier le modèle pré-entraîné **ResNet50**, connu pour son efficacité dans la **détection d'anomalies visuelles**.

La principale contribution est l'implémentation du modèle **BinaryCNN**, un réseau basé sur **CNN** dérivé de **ResNet50**, conçu pour détecter la **falsification par suppression d'images** dans les vidéos dashcam. Le système comprend les étapes de **prétraitement vidéo**, d'**extraction d'images**, de **classification image par image** et d'**agrégation** pour produire une décision finale.

L'évaluation a été effectuée sur le **jeu de données Video Forgery Dataset** (Kaggle), contenant plus de 2300 vidéos dashcam réparties entre vidéos originales et falsifiées (suppression, insertion, duplication, retournement, rotation, zoom). L'étude s'est concentrée sur la tâche de **classification binaire** entre vidéos originales et falsifiées par **supression d'images**.

Le modèle **BinaryCNN** a atteint une **exactitude de 79%**, une **précision de 81%**, un **rappel de 78%** et un **F1-score de 79%**, confirmant son efficacité pour la **détection de falsification vidéo** dans des conditions réalistes.

Une interface web conviviale a également été développée via Flask, permettant aux utilisateurs de télécharger une vidéo dashcam et d'obtenir une classification immédiate entre vidéo "originale" ou "falsifiée".

Ce travail s'inscrit dans l'essor des techniques de **détection automatique de falsification vidéo** en utilisant le **deep learning**, et montre le potentiel des **CNNs comme ResNet50** dans des systèmes intelligents, robustes et adaptables.

Mots-clés : falsification vidéo, contrefaçon vidéo, falsification vidéo numérique, détection de falsification vidéo, deepfake, ResNet50, réseaux neuronaux convolutifs (CNN), BinaryCNN, classification binaire, apprentissage profond, techniques d'apprentissage profond, détection automatisée de falsification

ملخص

أصبحت كاميرات الداشكام جزءاً لا يتجزأ من السيارات الحديثة، حيث تقوم بتسجيل فيديوهات مستمرة يمكن أن تشكل دليلاً رقمياً مهماً في الحالات القانونية أو مطالبات التأمين. ومع ذلك، فإن هذه الفيديوهات قد تتعرض لأشكال متعددة من **تزوير الفيديو**، مثل **حذف الإطارات**، أو **إدراج مقاطع مزيفة**، أو استخدام تقنيات **التزييف العميق Deepfake**.

تهدف هذه المذكرة إلى تقديم نظام يعتمد على تقنيات **التعلم العميق Learning Deep** للكشف عن **تزوير الفيديو**، وبالاعتماد على **الشبكات العصبية الالتفافية (CNN)** وخاصة البنية الشهيرة **ResNet50**، للكشف عن **الأنماط غير الطبيعية** داخل فيديوهات الداشكام.

تم تطوير نموذج مخصص باسم **BinaryCNN** مبني على ResNet50، ومصمم خصيصاً لاكتشاف **التزوير عبر حذف الإطارات**. يتكون النظام من مراحل: **معالجة الفيديو المسبق، استخراج الإطارات، تصنيف كل إطار**، و**تجميع النتائج** لإصدار قرار نهائي حول صحة الفيديو.

تم تدريب وتقييم النموذج على مجموعة بيانات من Kaggle التي تحتوي على فيديوهات مقسّمة إلى أصلية ومزورة بأنواع مختلفة مثل: حذف، إدراج، تكرار، قلب، تدوير، وتكبير. ركزت هذه الدراسة على **مهمة تصنيف ثنائي** بين الفيديو الأصلي والمزور عن طريق حذف الإطارات.

حقق نموذج **BinaryCNN** نتائج ممتازة تمثلت في **دقة 79%**، و**دقة إيجابية 81%**، و**استدعاء 78%**، و**F1-Score 79%**، ما يدل على كفاءة النموذج في **كشف التزوير** في ظروف واقعية.

كما تم تطوير **واجهة ويب باستخدام Flask**، تسمح للمستخدم برفع فيديو وتحليل حالته بشكل فوري باستخدام تقنيات الكشف عن تزوير الفيديو بالتعلم العميق.

تُظهر هذه الدراسة فعالية **الشبكات العصبية ResNet50 وCNN** في تطوير أنظمة ذكية وموثوقة للكشف التلقائي عن **تزوير الفيديوهات**.

الكلمات المفتاحية: التلاعب بالفيديو، تزوير الفيديو، التزوير الرقمي للفيديو، كشف التلاعب بالفيديو، التزييف العميق، ResNet50، الشبكات العصبية الالتفافية (CNNs)، BinaryCNN، التصنيف الثنائي، التعلم العميق، تقنيات التعلم العميق، الكشف الآلي عن التزوير

Contents

General Introduction **14**

1 Introduction to Video Forgery **16**
1.1 Introduction 16
1.2 Digital Video and Basic Structure (Images, Frames, Codecs) 17
1.2.1 Definition of Digital Video 17
1.2.2 Video Codecs and Compression 18
1.3 History and Evolution of Video Forgery 19
1.4 Classic and Modern Video Forgery 20
1.4.1 Classic Video Forgery 20
1.4.2 Modern Video Forgery 22
1.5 Video Forgery Tools 26
1.5.1 Traditional Video Editing Tools 26
1.5.2 AI-Based Tools 28
1.6 Security Challenges 29
1.6.1 Forged Videos in Cybercrime and Fraud 29
1.6.2 Threats to Smart Surveillance and Critical Systems 30
1.6.3 Evasion of AI-Based Detection Models 30
1.6.4 Ethical, Privacy, and Policy Challenges 30
1.7 Consequences of Video Forgery 31
1.7.1 Judicial Investigations 31
1.7.2 Insurance and Legal Disputes 31
1.7.3 Psychologically and Socially 32
1.8 Conclusion 34

2 Video Forgery Detection – State of the Art **36**
2.1 Introduction 36
2.2 Classical Detection Methods 37
2.2.1 Compression Artifact Analysis 37
2.2.2 Sensor Pattern Noise (SPN) Analysis 38
2.2.3 Photo Response Non-Uniformity (PRNU) Analysis 39

2.2.4 Temporal and Spatial Consistency Analysis . . . 40
2.3 Limitations of Classical Approaches . . . 43
2.3.1 Sensitivity to Noise . . . 43
2.3.2 Ineffectiveness Against Subtle and Advanced Manipulations . . . 43
2.3.3 High Dependency on Manual Analysis . . . 44
2.3.4 Lack of Real-Time Detection Capabilities . . . 44
2.4 Deep Learning in Video Forgery Detection . . . 44
2.4.1 Video Forgery Detection Models . . . 44
2.4.2 Convolutional Neural Networks (CNNs) . . . 46
2.4.3 CNN Architectures for Video Forgery Detection . . . 47
2.4.4 Datasets for Video Forgery Detection . . . 48
2.5 Conclusion . . . 49

3 CNN Architecture for Forgery Detection 50
3.1 Introduction . . . 50
3.2 Presentation of the Proposed Solution . . . 51
3.2.1 Objective . . . 51
3.2.2 Pipeline Overview . . . 51
3.3 Dataset . . . 53
3.3.1 Source . . . 53
3.3.2 Dataset Structure and Composition . . . 53
3.3.3 Data Partitioning . . . 54
3.3.4 Data Preprocessing . . . 55
3.4 CNN Model Architecture . . . 56
3.4.1 Pretrained ResNet50 . . . 56
3.4.2 Modifications for Binary Classification and Regularization . . . 56
3.4.3 Functioning of BinaryCNN . . . 57
3.4.4 Video-Level Classification Strategy . . . 57
3.5 Model Training . . . 58
3.5.1 Parameters . . . 58
3.5.2 Loss Function . . . 59
3.5.3 Optimization . . . 59
3.5.4 Handling Class Imbalance with Weighted Loss . . . 60
3.6 Model Evaluation . . . 60
3.6.1 Evaluation Metrics . . . 60
3.6.2 Performance Interpretation . . . 62
3.7 Application Deployment . . . 63
3.7.1 User Interface . . . 63
3.7.2 Software Tools . . . 64

3.7.3 Use Cases 66
3.8 Conclusion 67

General Conclusion **68**

bibliography **70**

List of Figures

1.1 Comparison of Frame Rates 30 Frames vs 60 Frames per Second. 17
1.2 Evolution of Video Forgery (1980 to 2025). 19
1.3 Video Frame Deletion. 20
1.4 Video Frame Insertion. 21
1.5 Video Frame Duplication. 21
1.6 Video Frame Transformations. 22
1.7 Illustration of how deepfake videos are created using facial tracking and GAN models [61]. 23
1.8 Illustration of Live Video Manipulation (Deepfake) [64]. 24
1.9 Visualization of an AI-powered script-to-video generation workflow [63]. . . 25

2.1 compression artifacts (blurring, ringing, blocking) in H.265/HEVC compressed video frames [58]. 38
2.2 CNN architecture illustrating feature learning and image classification [59]. 47

3.1 Video forgery detection pipeline using CNN and frame aggregation. 52
3.2 Structure of dataset folders showing original and forged classes. 54
3.3 Modified ResNet50 with binary classification and dropout regularization. . 56
3.4 Frame-level predictions aggregated via mean pooling for final video classification. 57
3.5 Main training loop used to optimize the BinaryCNN model. 58
3.6 Weighted cross-entropy loss to emphasize detection of forged frames. . . . 59
3.7 Flask web interface for uploading dashcam videos and displaying the classification result. 64
3.8 System architecture of the proposed video tampering detection solution. . 64
3.9 Flask and Pytorch [66]. 65
3.10 HTML5, JavaScript, and CSS3 [65]. 66

List of Tables

1.1 Comparison of Popular Video Codecs [60]. 18
1.2 Comparison Between Classic and Modern Video Forgery Techniques 26

2.1 Comparison of SPN vs PRNU Analysis 40
2.2 Comparative Overview of Temporal vs Spatial Consistency Detection Methods 42

3.1 Per-class performance metrics of BinaryCNN 62

General Introduction

In recent years, dashcam recordings have emerged as a crucial source of visual evidence in legal investigations, insurance claims, and traffic monitoring. These compact devices, mounted on vehicle dashboards, continuously capture events on the road, providing objective documentation of accidents, violations, or criminal incidents. Their growing adoption has made video footage a standard and trusted form of digital evidence in both private and institutional contexts [27, 28].

However, the credibility of dashcam videos is increasingly under threat due to the rise of video forgery techniques. Manipulations such as frame deletion, insertion, duplication, or advanced deepfake-based alterations have become more accessible and harder to detect with traditional methods [2, 7]. These forgeries can distort facts, mislead investigations, and compromise justice, especially when such videos are presented as primary evidence in legal or insurance procedures.

Given the limitations of classical detection techniques—based on compression artifact analysis, sensor noise patterns, or manual inspection—there is a pressing need for automated and intelligent systems capable of detecting subtle forgeries in complex, real-world conditions. Deep learning, particularly Convolutional Neural Networks (CNNs), has shown strong potential in this domain by learning discriminative features directly from raw video data and identifying tampering patterns without relying on handcrafted heuristics [46, 1].

This master thesis aims to contribute to this field by developing a deep learning-based system for detecting video forgery, with a specific focus on identifying frame deletion in dashcam videos. The approach leverages a fine-tuned ResNet50 CNN architecture within a customized pipeline named *BinaryCNN*, capable of performing frame-level classification and aggregating predictions to assess the authenticity of entire videos. The model is evaluated using the Kaggle Video Forgery Dataset, which includes various manipulation types under real driving conditions.

Our work is organized into three chapters. Chapter 1 introduces the problem of video forgery, its history, categories, consequences, and detection challenges. Chapter 2 presents a detailed review of traditional and deep learning-based detection methods, with an emphasis on CNN architectures and forgery datasets. Chapter 3 describes the design, training, and deployment of the proposed BinaryCNN model, and analyzes its

performance metrics.

By bridging forensic analysis with deep learning techniques, this research aims to improve the reliability of dashcam video evidence and contribute to the broader effort of securing visual data against manipulation in legal, insurance, and security contexts.

Chapter 1

Introduction to Video Forgery

1.1 Introduction

With the widespread availability of video recording devices and the increasing reliance on visual evidence in legal and security contexts, digital video has become a crucial medium in modern society. Among various sources, dashcams—cameras installed in vehicles—have gained prominence for their ability to capture continuous video streams during driving. These recordings serve as valuable evidence in the event of accidents, traffic violations, or criminal activities on the road [27].

However, the integrity of video content is no longer guaranteed. Advances in video editing software and AI-based manipulation techniques have made it relatively easy to modify or forge videos. Techniques such as frame deletion, insertion, duplication, flipping, or even deepfake generation pose serious challenges to the credibility of video evidence [7]. These alterations are often subtle and designed to deceive both human viewers and traditional forensic tools.

This chapter presents the fundamental concepts necessary to understand the structure and vulnerabilities of digital video. It begins by defining digital video in terms of its basic components—frames, frame rate, and codecs and explains how video compression affects the visual quality and forensic traceability. The second part of the chapter introduces the main categories of video forgeries and describes how each type manipulates the content to distort reality. The chapter concludes with a discussion on why detecting such manipulations has become an urgent task in the age of multimedia-based communication.

The knowledge provided in this chapter forms the theoretical foundation for understanding how video tampering works and sets the stage for analyzing existing detection methods and building intelligent forgery detection systems in subsequent chapters.

1.2 Digital Video and Basic Structure (Images, Frames, Codecs)

1.2.1 Definition of Digital Video

A digital video consists of a series of images (also called frames) captured at regular intervals. Each image captures a specific moment of a scene, and when presented consecutively, they create the illusion of movement. These videos are typically stored as digital files, allowing them to be viewed and manipulated on electronic devices such as computers, televisions, and smartphones [35].

The Frames

The basic elements of a digital video are the frames (or images). A video typically consists of hundreds or thousands of images, each capturing a specific moment in the filmed scene. The frame rate or fps (frames per second) refers to the speed at which these frames are presented. For example, a typical Full HD video displays 30 frames per second (30 fps), ensuring smooth motion on the screen [4, 35].

The Frame Rate (FPS)

The frame rate determines the number of frames shown per second [3]. The higher the frame rate (fps), the smoother the video appears. Videos can range from 15 frames per second for low-quality footage to 60 frames per second or higher for high-definition content, such as video games or 4K videos.

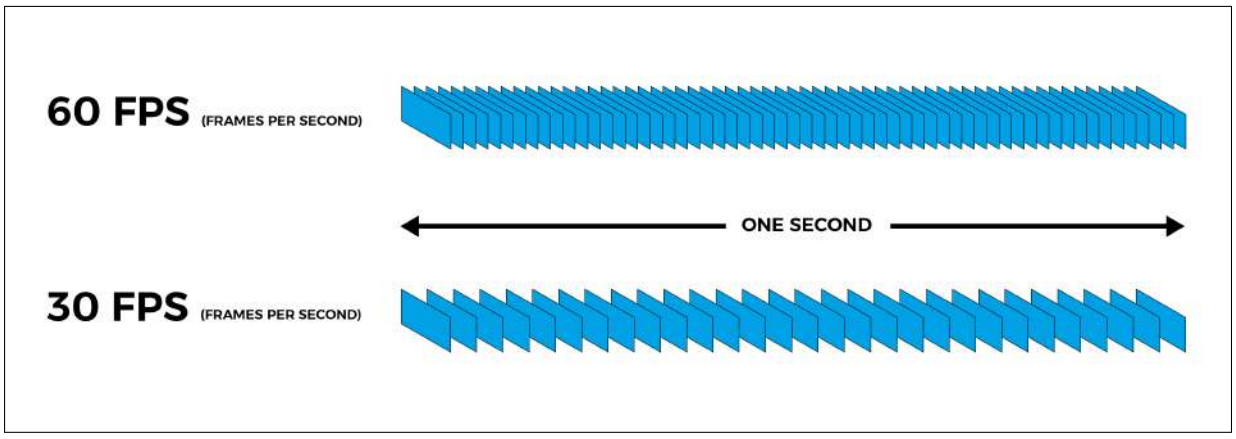


Figure 1.1: Comparison of Frame Rates 30 Frames vs 60 Frames per Second.

1.2.2 Video Codecs and Compression

To facilitate storage and transmission, digital videos are often compressed to reduce their file size. This compression is done using video codecs. A codec (compressor-decompressor) is a procedure used to reduce the size of video data for storage, while maintaining image and sound quality as much as possible.

Types de Codecs

The most commonly used video codecs include:

- **H.264 (AVC) :** This is the most widely used codec for online video streaming (for example, on YouTube or Netflix). It provides a good balance between compression and quality.
- **H.265 (HEVC) :** This codec is an improved version of H.264, offering better compression and quality for high-definition videos (4K) [4, 5].
- **VP9 :** A free video codec developed by Google, used for 4K streaming on YouTube.
- **AV1 :** A newer video codec that promises better compression than H.265 and VP9, while being royalty-free.

Codec	**Compression**	**Quality**	**File Size**	**Cost**	**Support**	**Best Use Case**
H.264 (AVC)	Moderate	Good	Moderate	Royalties	Widespread	General streaming, conferencing, widely supported
H.265 (HEVC)	Very high	Excellent	Much smaller	Royalties	Newer devices	4K/8K streaming, HDR video
VP9	High	Better	Smaller	Free	Good	YouTube, browser-based streaming
AV1	Highest	Best	Smallest	Free	Limited (growing)	Future-proof, open-source streaming platforms

Table 1.1: Comparison of Popular Video Codecs [60].

Video Compression

Video compression can be lossy or lossless. Codecs like H.264 and H.265 use lossy compression, where some information is discarded to reduce the file size. This can result in a decrease in quality, but it is usually undetectable to the naked eye. Lossless compression is often used for applications that require maximum quality preservation [3, 36], such as in professional videos.

1.3 History and Evolution of Video Forgery

Over the past few decades, video content manipulation has changed dramatically in tandem with advancements in digital technologies. Tampering in the analog era required physically cutting and splicing magnetic tapes, which was expensive and time-consuming and left visible artifacts. Software tools that enabled accurate frame-level adjustments, color corrections, and smooth transitions made editing easier with the introduction of digital video.

By the early 2000s, forgery had become easier and faster thanks to consumer video editors like Final Cut Pro, Sony Vegas, and Adobe Premiere. This signaled the start of traditional digital manipulation, including duplication, insertion, and deletion of frames. Because of variations in frame rate, compression, or noise patterns, these forgeries were frequently identifiable by forensic tools or skilled professionals.

AI-generated content became more popular in the 2010s, particularly after Generative Adversarial Networks (GANs) were introduced. These made it possible to produce deepfakes, which are incredibly lifelike videos in which a person's voice or face can be convincingly substituted. Even non-experts can now create extremely misleading video content thanks to open-source tools like DeepFaceLab.

Today, video forgery has advanced to the point where lightweight neural networks can manipulate videos in real time, posing a threat to digital security frameworks and legal systems alike.

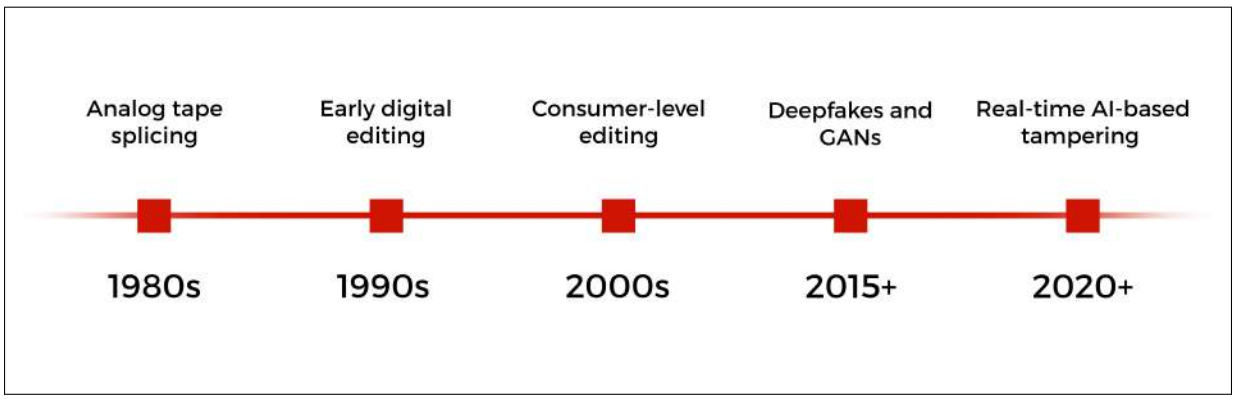


Figure 1.2: Evolution of Video Forgery (1980 to 2025).

1.4 Classic and Modern Video Forgery

1.4.1 Classic Video Forgery

Before the rise of AI, video tampering was performed using manual or semi-automated editing tools. These classic forgery techniques rely on human interaction and conventional software for manipulating temporal and spatial aspects of video content.

Frame Deletion

The intentional removal of one or more frames, or complete segments, from a video sequence is referred to as frame deletion as shown in Figure 1.3. This method is frequently employed to remove important visual evidence, such a traffic infraction, an accident, or an action that could be used against the accused. The forger modifies the chronological flow of events by removing particular segments from a movie, which may cause the spectator to perceive events differently. Careful examination can occasionally reveal minute temporal irregularities or strange transitions, even if the movie may still seem continuous. However, in many instances, this kind of fabrication is still invisible to the naked eye.

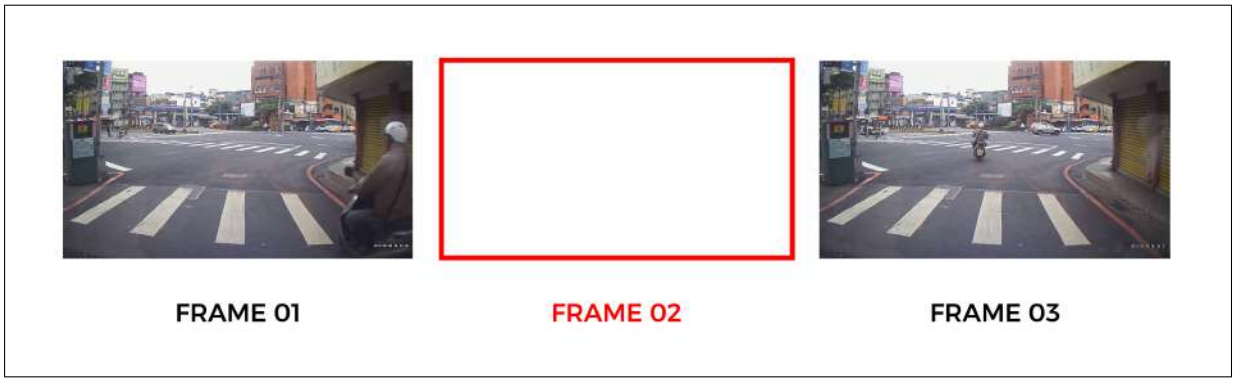


Figure 1.3: Video Frame Deletion.

Frame Insertion

Adding frames or segments to a video that weren't included in the initial recording is known as frame insertion as shown in Figure 1.4. These extra components could be artificially created to present fictitious occurrences or they could be taken from other film. Falsification of this kind can create settings that trick viewers into thinking that things happened when they didn't. The natural flow of action, spatial coherence, or illumination consistency are frequently broken by frame insertion, while the forgery's quality may mask these inconsistencies. It is especially dangerous in settings where video recordings are used as factual or legal proof.



Figure 1.4: Video Frame Insertion.

Frame Duplication

The process of reproducing and repeating a frame sequence inside a single video is known as frame duplication as shown in Figure 1.5. This method can be used to create the appearance of continuity, cover up short but important activities, or artificially lengthen time periods. For instance, a forger can conceal abrupt movements or scene changes that would normally draw attention by replaying a neutral video clip. As a result, the timeframe is altered, gradually changing the story that the video tells. To find odd patterns of recurrence or temporal irregularities, sophisticated detection methods can be needed.



Figure 1.5: Video Frame Duplication.

Transformations

Transformational forgery refers to geometric or spatial modifications applied to video frames. Common transformations include rotation, zoom, cropping, scaling, and flipping as shown in Figure 1.6. These modifications are often used to distort the visual orientation of the video or to obscure specific elements such as faces, vehicle plates, or geographic markers. In some cases, transformations are employed to bypass automated object detection or facial recognition systems. Although these changes may not affect the content of

the video in terms of sequence or duration, they can significantly alter its interpretability and authenticity.



Figure 1.6: Video Frame Transformations.

1.4.2 Modern Video Forgery

Modern video forgery techniques rely heavily on machine learning and deep learning, which allow attackers to create hyper-realistic videos with minimal effort. These methods surpass classic editing by leveraging neural networks, especially Generative Adversarial Networks (GANs) and temporal modeling systems, to automate forgery tasks.

Below are the most widely known and impactful methods in this category:

Deepfakes

One of the most sophisticated and possibly hazardous types of video fabrication is represented by deepfakes. This term refers to synthetically generated or altered videos where a person's face, body, or voice is replaced, modified, or generated entirely from scratch using deep learning techniques, primarily Generative Adversarial Networks (GANs) [32, 33].

Deepfakes generate hyper-realistic videos by training a generator network to mimic the facial expressions, speech patterns, and movements of a target person. The results are often indistinguishable from authentic footage, especially when the forged content is short, well-lit, or in compressed formats.

Use cases :

- Replacing faces in videos (face swapping) as shown in Figure 1.7
- Synchronizing someone's lips with new audio (lip-syncing)
- Generating entirely fabricated videos of people doing or saying things they never did



Figure 1.7: Illustration of how deepfake videos are created using facial tracking and GAN models [61].

While deepfakes are not yet widespread in surveillance or dashcam footage, their accessibility via open-source tools like DeepFaceLab or FaceSwap makes them a rising concern in digital forensics .

Lip-Sync and Audio-Driven Video Manipulation

This method involves creating or modifying a person's mouth movements in a video so that they appear to be speaking a different script, synchronized with altered or fake audio.

Using models like Wav2Lip or SyncNet, the forger can:

- Modify mouth movements frame-by-frame to match synthetic audio
- Replace dialogue or inject fake speech into existing footage
- Generate false video confessions or testimonials with convincing sync

These methods are often used alongside deepfakes to improve credibility in manipulated videos where voice and visual coherence are critical [29].

Real-Time Video Manipulation (Live Deepfakes)

Modern forgery systems are increasingly capable of manipulating video streams in real-time. This involves applying deep learning filters during live video calls or streams using tools like Avatarify or Xpression Camera.

Key Features:

- Real-time face replacement or animation as shown in Figure 1.8

- Live application of emotions, movements, or voices
- Can be used during video conferences, streaming, or television broadcasts

Use Cases:

- Impersonating someone on a live Zoom call
- Broadcasting altered personas
- Real-time censorship evasion



Figure 1.8: Illustration of Live Video Manipulation (Deepfake) [64].

This method is particularly dangerous as it enables live social engineering, deepfake impersonation, and undetectable deception in dynamic settings [7].

AI-Based Object and Scene Manipulation

Unlike deepfakes that focus on humans, this category targets the scene content itself. Techniques such as video inpainting, object removal, and video interpolation are used to:

- Remove objects or people from a scene
- Insert new synthetic elements
- Smooth transitions between forged segments

These manipulations are done using tools like:

- Runway ML (video inpainting)

- Sora / Synthesia (AI-generated scene/video from text)
- DAIN / RIFE (interpolation and frame synthesis)

These approaches do not rely on manual edits but use context-aware AI to "hallucinate" missing content, often blending it seamlessly with surrounding video data [30, 31].

Text-to-Video Generation

A more recent development in video manipulation is the ability to generate entire videos from text descriptions using large AI models as shown in Figure 1.9. Models like Sora (by OpenAI) or Pika Labs can generate short, coherent video clips based solely on textual input, often with realistic physics, lighting, and motion. Though still in early stages, these tools present a significant challenge for video authenticity verification.

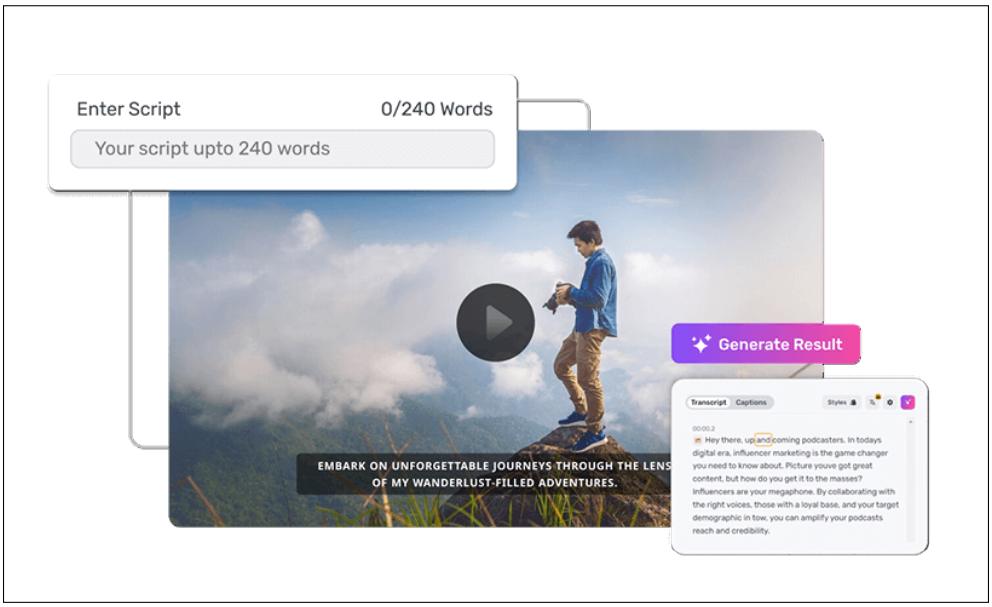


Figure 1.9: Visualization of an AI-powered script-to-video generation workflow [63].

Feature	Classic Techniques	Modern Techniques (AI-Based)
Tools Used	Video editors, FFmpeg	GANs, deep learning frameworks
Skills Required	Manual editing skills	AI knowledge or use of pre-trained models
Realism	Moderate	Very high (deepfakes)
Automation	Mostly manual	Fully automated
Detection Difficulty	Moderate to high	Very high
Examples	Cutting, duplicating, color editing	Deepfake speeches, face swaps

Table 1.2: Comparison Between Classic and Modern Video Forgery Techniques

1.5 Video Forgery Tools

The process of video forgery relies on a wide variety of software tools, ranging from traditional video editing programs to sophisticated artificial intelligence (AI) frameworks. These tools enable attackers to manipulate both the temporal (frames, motion) and spatial (appearance, objects) aspects of a video with varying degrees of subtlety and realism. Below is a categorization of commonly used tools for video forgery.

1.5.1 Traditional Video Editing Tools

These tools are widely available and primarily used for manual or timeline-based editing. They form the basis of classic video forgery techniques, such as frame deletion, duplication, or insertion.

Adobe Premiere Pro

A professional-grade video editing suite used for frame-level manipulation, audio synchronization, and visual effects. Forgers can:

- Delete specific scenes
- Rearrange sequences
- Overlay fake elements
- Manipulate lighting and contrast

Sony Vegas Pro / Final Cut Pro

Similar to Adobe Premiere, these tools offer powerful editing capabilities and are widely used in media production. Their timeline editing makes it easy to:

- Splice different video segments
- Add smooth transitions
- Insert unrelated frames

VirtualDub

An open-source video capture and processing tool known for its simplicity and speed. VirtualDub is effective for:

- Cutting and merging AVI videos
- Removing frames
- Changing frame rates

It lacks advanced visual effects, but its lightweight nature makes it popular for quick tampering.

FFmpeg

A powerful command-line tool for manipulating multimedia content. Forgers can use FFmpeg to:

- Extract or replace video segments
- Convert between formats/codecs
- Compress or re-encode videos (which can also hide traces of tampering)

Its scriptability makes it ideal for batch processing or automation.

After Effects (Adobe)

Used for inserting digital effects, masking, object removal, or motion tracking. It's particularly helpful for:

- Altering visual content frame-by-frame
- Hiding license plates, faces, or backgrounds
- Seamless transitions and compositing

1.5.2 AI-Based Tools

The rise of deep learning has enabled a new class of tools that can automatically manipulate or synthesize video content using pre-trained models, especially those based on Generative Adversarial Networks (GANs).

DeepFaceLab

One of the most widely used tools for creating deepfake videos. It allows for:

- Facial swapping between individuals
- Facial reenactment and synthesis
- Automatic facial landmark detection

It supports GPU acceleration, facial segmentation, and training with large datasets.

FaceSwap

An open-source deepfake framework that offers a user-friendly interface for:

- Replacing faces in videos using autoencoders
- Aligning and training facial models
- Generating high-quality deepfakes with minimal manual input

Avatarify

Enables real-time face animation during video conferencing. By tracking facial landmarks and syncing expressions, it allows users to impersonate another person live.

- Used on Zoom, Skype, or OBS Studio
- Popular in real-time impersonation fraud

Wav2Lip

A state-of-the-art model for generating lip-synced video from audio. The model modifies a person's mouth movements to match any speech input.

- Useful in fake video confessions
- Hard to detect due to precise alignment

Runway ML

A platform offering AI tools for:

- Video inpainting: removing or replacing objects in video
- Style transfer and background changes
- Used for cinematic alterations or falsifying contextual cues in videos

Synthesia / Sora (Emerging Tools)

Advanced AI platforms that generate synthetic videos from text prompts or avatars. These can be used to:

- Create fake spokespersons or witnesses
- Generate realistic actors who never existed
- Alter timelines or events entirely without real footage

1.6 Security Challenges

Significant security risks are presented by the growing incidence of video forgery in a number of areas, such as cybersecurity, legal systems, digital surveillance, and privacy protection. Since video content is increasingly used for sharing information, presenting evidence, and making decisions, its authenticity and integrity are essential to preserving trust in digital ecosystems. Whether created using conventional editing methods or artificial intelligence (AI)-based manipulation (such as deepfakes), manipulated videos can be used for nefarious activities like fraud, defamation, cyberattacks, and even terrorism [10, 45].

1.6.1 Forged Videos in Cybercrime and Fraud

The use of video forgery in cybercrime is one of the most pressing risks it poses. Attackers can pose as people in phishing, scams, and identity theft by using edited videos. One tactic already seen in sophisticated CEO fraud schemes is the use of a deepfake video to pretend to be a company executive requesting a money transfer from an employee. Similar to this, videos displaying false confessions, incriminating conduct, or improper acts can be produced and used for reputational or blackmail attacks, leaving people and public figures open to online pressure [9, 33].

1.6.2 Threats to Smart Surveillance and Critical Systems

Modern security systems including smart surveillance, autonomous vehicles, and military drones rely heavily on video inputs for real-time decision-making. When such systems are fed tampered or forged video streams, they may fail to recognize threats or respond incorrectly. For example:

- A forged dashcam video could alter the timeline of a traffic incident, misleading investigators or insurance agencies.
- A tampered surveillance feed might hide the presence of an intruder, allowing physical attacks or sabotage to occur undetected.
- AI-based recognition systems (e.g., facial or object detection) can be fooled by adversarial video manipulations, leading to false negatives (missed threats) or false positives (wrong alerts).

This vulnerability highlights the urgent need to secure video pipelines in IoT and smart environments, where decisions are increasingly automated and reactive [27, 28].

1.6.3 Evasion of AI-Based Detection Models

As forgery detection itself becomes automated through deep learning, attackers are developing methods to intentionally bypass detection systems. To manipulate videos in ways that avoid machine learning models, even those trained on sizable forgery datasets, methods like adversarial examples, GAN blending, and noise injection are employed.

Attackers and defenders engage in a risky arms race as a result, with forgery tools becoming more sophisticated and challenging to identify with each new development in detection technology. Constant updates, robust models, and a variety of training data are essential for the long-term security of AI-based video authentication systems [38, 20, 47].

1.6.4 Ethical, Privacy, and Policy Challenges

Detecting video forgery at scale requires access to and analysis of personal and sensitive video content, including user-generated footage on social media platforms. This raises ethical and privacy concerns, especially when detection tools are deployed in surveillance, law enforcement, or public security domains.

- Mass surveillance without consent under the guise of forgery detection.
- False positives labeling real videos as fake, which can damage trust and reputations.
- Abuse of detection systems by governments or corporations to censor content under the pretense of authenticity verification.

Thus, forging a balance between security and civil rights is critical in the deployment of forgery detection frameworks [34, 10].

1.7 Consequences of Video Forgery

The falsification of video content has profound implications across multiple domains. As visual recordings increasingly serve as key sources of evidence and decision-making tools, any compromise in their integrity can severely impact the credibility of institutions and the outcomes of critical processes. This section examines the consequences of video forgery in three primary contexts: judicial investigations, insurance and legal disputes, and public trust in digital systems.

1.7.1 Judicial Investigations

Video recordings are frequently used as direct evidence in court to substantiate or disprove allegations, reconstruct incidents, or identify criminals. Incorporating fabricated video footage into court cases can have disastrous results. It may cause erroneous convictions or acquittals, mislead investigators, and impede the administration of justice.

Removing a few frames or changing timestamps are examples of little modifications that can skew the timeline and influence how events are interpreted. Modified footage might change the perceived legality or guilt of the persons involved in situations when choices are based on split-second actions, such as when law enforcement uses force or when there are traffic infractions.

Furthermore, legal professionals find it challenging to distinguish genuine film without the use of specialized forensic tools [45] due to the growing realism of deepfakes and other sophisticated alterations. This makes already delicate investigations more complicated, expensive, and unpredictable, which eventually calls into question the validity of digital video as a source of truth in the legal system.

1.7.2 Insurance and Legal Disputes

Dashcam footage is now routinely used in the insurance sector to handle claims pertaining to property damage, auto accidents, and responsibility issues. It is possible to falsify damages, hide at-fault behavior, or stage fictitious incidents by forging such footage [9]. Because of this, insurers might be duped into accepting false claims, which would result in monetary losses and systemic inefficiencies.

Additionally, the existence of fabricated movies may make dispute settlement more difficult. When one party presents evidence that has been manipulated, it creates a situation where more resources are needed to look into the material's legitimacy. This can

damage trust between clients, insurers, and third-party adjusters in addition to delaying the settlement process.

Furthermore, the possibility of manipulation casts doubt on the footage and reduces its persuasive power in court proceedings where it is used to bolster or refute a legal argument. Its legitimacy may be contested by opposing parties, necessitating forensic analysis and expert intervention—procedures that are time-consuming and frequently inconclusive.

1.7.3 Psychologically and Socially

The effects of video forgery have spread well beyond the technical and security spheres, especially since the introduction of deepfakes. These fake videos have the psychological potential to skew people's understandings of reality, increase anxiety, and erode public confidence. Socially, they can be used to sway public opinion, spark disputes, or harm the reputations of individuals and organizations. It is essential to comprehend these non-technical effects since they affect how societies respond to misrepresented media and how laws should be created to prevent abuse.

Erosion of Public Trust

The decline in public confidence in digital systems is arguably the most extensive effect of video counterfeiting. The authenticity of this video serves as the foundation for societal truth and accountability in a time when dashcams, smartphones, and surveillance cameras record a significant amount of our everyday lives.

The public may become widely skeptical of journalism, the media, police recordings, and even private film if they start to believe that video content is easily altered. The ability of editing software and artificial intelligence to convincingly create situations that never happened challenges the idea that "seeing is believing" [10].

Because edited recordings can be used as a weapon to sway public opinion, stir strife, or discredit people and organizations, this lack of trust can have a chilling effect on investigative journalism, whistleblowing, and even democratic processes.

Forged or altered information can also deceive algorithms in technical systems, including automated surveillance or facial recognition systems, resulting in missed threats or erroneous detections. Both the safety and security of vital infrastructure as well as individual privacy are seriously threatened by this.

Propagation of Misinformation

Forged videos are increasingly used in disinformation campaigns designed to manipulate public perception or create social unrest. These can include:

- Fake videos during elections to discredit candidates.
- Misleading clips to incite ethnic or political violence.
- Fabricated footage of public figures engaging in scandalous behavior.

Due to the visual realism and emotional impact of video, manipulated content spreads faster and is more persuasive than text-based misinformation. According to a 2020 report by the Brookings Institution, deepfake videos can amplify polarization and radicalization in online communities [12].

Psychological Manipulation and Emotional Harm

On an individual level, forged videos can cause:

- Anxiety and paranoia, especially among public figures or activists.
- Emotional trauma in cases where individuals are falsely portrayed in pornographic or criminal contexts.
- Social exclusion or harassment, particularly when circulated on social media.

Studies in media psychology show that false visual memories can be formed when viewers are repeatedly exposed to forged videos, especially when the content is emotionally charged [13]. This makes it harder for viewers to distinguish fact from fiction, even after the forgery is debunked.

Impact on Journalism and Media Credibility

Journalists and news organizations increasingly face challenges verifying video content, particularly in fast-moving crisis situations. The rise of deepfakes may lead media outlets to:

- Hesitate before broadcasting video content.
- Rely more on textual sources or expert verification.
- Lose public credibility when caught sharing forged content.

This undermines the freedom of the press and the ability to expose corruption or injustice with audiovisual evidence.

Gendered and Targeted Abuse

Forged videos are disproportionately used to target women and minorities. A large portion of deepfake content online involves non-consensual pornography, with women being the primary victims [14]. These forgeries are used to:

- Humiliate individuals.
- Destroy reputations.
- Blackmail victims into silence.

Such attacks reflect broader social patterns of digital gender-based violence, raising the need for strong legal and ethical countermeasures.

Societal Desensitization and Fatigue

As forged videos become more common, people may begin to doubt everything they see even genuine footage of violence, injustice, or disaster. This leads to:

- Desensitization to real events.
- Decreased civic engagement.
- An increase in "truth apathy" a tendency to stop caring about what's real.

This undermines collective decision-making, especially in democratic societies, where an informed public is vital.

1.8 Conclusion

This chapter has examined the nature, techniques, and consequences of video forgery from both technical and societal perspectives. We began by outlining the fundamental structure of digital video, emphasizing its frame-based composition, reliance on codecs such as H.264 and H.265 for compression [4], and the inherent vulnerabilities that make it susceptible to tampering at various stages of the digital pipeline.

We then explored a typology of video manipulation methods, categorizing them into direct content alterations (e.g., frame deletion, insertion, duplication), geometric transformations (e.g., flipping, zooming, rotation), and modern AI-based forgeries such as deepfakes [7]. Each category presents unique challenges in terms of detection difficulty, intent, and level of realism. Tools facilitating these forgeries range from standard editing software to advanced AI models like GANs, which can generate highly convincing content that is difficult to detect through manual inspection.

Beyond the technical aspects, we also considered the broader implications of video forgery. In judicial contexts, tampered videos can obstruct investigations, lead to miscarriages of justice, or cast doubt on legitimate evidence. In insurance claims, they may enable fraudulent behavior, delay resolution, and inflate costs. On a societal scale, the proliferation of forged videos undermines public trust in digital content, challenges the credibility of surveillance systems, and contributes to misinformation and social tension.

Given the high stakes associated with forged video content, especially in domains such as law enforcement, insurance, and digital media, developing reliable and intelligent detection mechanisms has become a critical research priority [45]. The next chapter will explore the landscape of forgery detection techniques, reviewing both classical forensic methods and modern deep learning-based models, and highlighting their strengths, limitations, and relevance to real-world tampering detection scenarios. This review will serve as a foundation for designing and implementing the detection system proposed in this master thesis.

Chapter 2

Video Forgery Detection – State of the Art

2.1 Introduction

The rapid advancement of video editing tools and artificial intelligence has led to increasingly sophisticated forms of video tampering. Traditional forensic methods, while foundational, are often inadequate in detecting subtle or adversarial manipulations that modern techniques can produce. Given the stakes involved—especially in law enforcement, insurance claims, and digital evidence validation—detecting forged content has become a critical area of research.

This chapter provides a comprehensive overview of existing video forgery detection techniques, structured into two major parts. The first section explores classical approaches, including methods based on sensor pattern noise (SPN), photo-response non-uniformity (PRNU), compression artifact analysis, and spatial/temporal consistency checks [45, 15, 37]. These techniques leverage low-level signals embedded in video content to detect signs of manipulation, yet they suffer from sensitivity to noise, compression, and limited generalizability.

The second part focuses on deep learning-based methods, which have emerged as powerful alternatives due to their ability to learn complex feature representations automatically. In particular, Convolutional Neural Networks (CNNs) and their extensions—3D CNNs, hybrid CNN-RNN models, and transformer-based architectures—have shown superior performance in detecting both spatial and temporal anomalies in video sequences [20, 52, 53]. These models, supported by large annotated datasets such as FaceForensics++, DFDC, and the Kaggle Video Forgery Dataset, enable scalable and more accurate tampering detection.

By comparing the strengths and weaknesses of traditional and modern techniques, this chapter sets the stage for the architecture and implementation of the proposed CNN-based detection model presented in the next chapter.

2.2 Classical Detection Methods

In the past, forensic methods based on statistical analysis, signal processing, and domain-specific expertise in camera hardware and video encoding have been crucial in the detection of video forgeries. These traditional techniques, which frequently take advantage of the digital and physical artifacts that are challenging for forgers to completely duplicate or remove, are intended to spot minute irregularities and discrepancies left by tampering activities. An extensive summary of the main traditional methods for detecting video forgeries is given in this section.

2.2.1 Compression Artifact Analysis

The lossy process of digital video compression, used by standards like MPEG-2, H.264/AVC, and HEVC, minimizes file size by taking advantage of temporal and spatial redundancy in video data. The utilization of macroblocks, which are tiny blocks of pixels usually 16x16 in size, is a key component of these codecs. They are the essential unit for compression operations like motion correction and discrete cosine transform (DCT) [3, 36].

These macroblocks encode spatial and temporal continuity, which is broken when a video is altered by frame insertion, deletion, or rearranging [11]. For example, altered segments may generate unnatural discontinuities at block boundaries and change the quantization parameters or motion vector distribution. Analyzing residual error statistics, motion vector coherence, and macroblock patterns are necessary to find these discrepancies.

In order to identify compression fingerprints, advanced forensic technologies analyze anomalous motion vector fields, inconsistent quantization artifacts, and differences in block border smoothness. These techniques take advantage of the fact that tampering typically results in compression discrepancies because of re-encoding or improper splicing of video portions that have been compressed using different codecs or parameters.

However, when videos go through several recompression rounds or high-quality re-encoding, which could mask or eliminate recognizable artifact patterns, compression-based solutions encounter difficulties. Additionally, they may not work as well for videos encoded using contemporary codecs [16] that aim to reduce visual artifacts and necessitate familiarity with the compression standard in use.

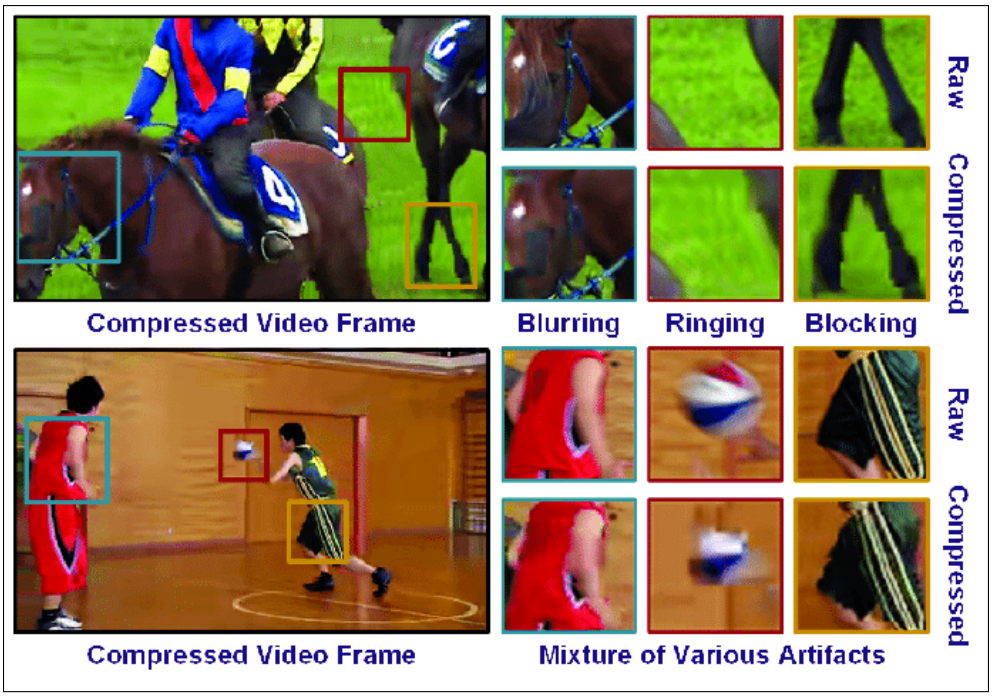


Figure 2.1: compression artifacts (blurring, ringing, blocking) in H.265/HEVC compressed video frames [58].

2.2.2 Sensor Pattern Noise (SPN) Analysis

Sensor Pattern Noise (SPN) refers to the unique and consistent noise characteristics introduced by a camera's image sensor during video or image capture [37]. It is caused by microscopic imperfections and inconsistencies in the sensor's manufacturing process, resulting in a fixed-pattern noise that remains stable across frames captured by the same device. This noise acts as a digital fingerprint of the camera.

SPN is widely used in multimedia forensics as a passive technique to verify the authenticity and origin of video content [15]. When a forged video contains frames spliced from different sources or recorded by a different device, the sensor noise pattern is disrupted, allowing forensic analysts to detect inconsistencies. In such cases, SPN patterns across frames do not correlate as expected, revealing possible signs of tampering.

To extract SPN, denoising filters are applied to suppress scene content and isolate the residual fixed-pattern noise. Correlation analysis is then used to compare this residual noise across frames or against a reference signature [45]. A significant drop in correlation or mismatch with expected patterns may indicate frame-level tampering or manipulation.

Despite its usefulness, SPN analysis is sensitive to several real-world factors. Compression artifacts, resizing, filtering, stabilization, or motion blur can weaken or distort the sensor noise. This can lead to reduced accuracy or false conclusions if not carefully accounted for. Nonetheless, SPN remains a foundational tool for source identification and integrity verification in forensic video analysis [15].

2.2.3 Photo Response Non-Uniformity (PRNU) Analysis

Photo Response Non-Uniformity (PRNU) is the dominant and most widely used component of SPN in forensic video analysis. PRNU represents the pixel-level variations in sensor sensitivity to light and is embedded in all images and video frames captured by a specific camera. It behaves like a unique digital fingerprint of the recording device [37, 15].

PRNU analysis is especially effective in identifying frame-level manipulations. When a video contains frames inserted from a different device or tampered regions, the PRNU noise pattern in those areas will differ from the rest of the footage. By isolating the PRNU noise and correlating it with a reference fingerprint from the same camera, inconsistencies can be revealed that suggest forgery [15].

To extract PRNU, forensic analysts usually apply denoising algorithms (such as BM3D or wavelet filters) and subtract the denoised image from the original. The residual represents the noise, which includes PRNU. Cross-correlation with reference noise patterns helps to localize and confirm tampering.

PRNU analysis remains one of the most accurate forensic techniques due to its reliance on hardware-level features that are nearly impossible to replicate. However, it requires access to either a clean reference PRNU fingerprint or a large number of untampered frames. Moreover, PRNU reliability degrades when videos are heavily compressed, stabilized, or resampled [45].

Aspect	Sensor Pattern Noise (SPN)	Photo Response Non-Uniformity (PRNU)
Origin	General fixed-pattern noise from sensor defects	Specific pixel-level light sensitivity variation
Use Case	Source camera identification, basic tampering detection	High-precision frame authentication and tampering localization
Detection Method	General denoising and correlation across frames	Residual noise extraction and reference fingerprint correlation
Sensitivity	Moderate — affected by compression and noise	High — but degraded by resizing or filtering
Requirements	May not require reference pattern	Often needs reference PRNU or original frame sequence
Strengths	Simple to extract; useful for general integrity check	Accurate; based on hardware-specific characteristics
Limitations	Less robust to post-processing or format changes	Fragile under compression or stabilization

Table 2.1: Comparison of SPN vs PRNU Analysis

2.2.4 Temporal and Spatial Consistency Analysis

Video authenticity can also be assessed by examining the temporal and spatial coherence of visual content within and between frames [46, 45].

Temporal Consistency

This method compares successive frames to assess the smoothness and believability of motion, changes in illumination, and object trajectories. Unexpected temporal discontinuities are frequently caused by tampering operations like frame deletion or insertion [46, 38], and they can be identified using methods like motion vector analysis, temporal differencing, or optical flow estimates.

Pixel-wise motion between frames is estimated via optical flow methods [39], for instance. Motion vectors with abrupt pauses or strange jumps could be signs of missing or modified frames. Similarly, tampering-introduced anomalies can be found in statistical models of illumination changes or temporal noise.

Spatial Consistency

By examining characteristics including lighting direction, shadow alignment, texture patterns, and noise distribution, spatial coherence is investigated within individual frames [1]. Spatial irregularities are frequently introduced by forged content—for example, shadows from inserted objects might not match the ambient lighting, or sections that are copied might have different textures or noise levels.

Methods include examining the geometry of shadows [40], utilizing texture descriptors to identify cloned areas, or use reflectance models to analyze illumination consistency. Some techniques use frequency analysis or edge detection to find blending artifacts or irregular transitions [45].

Aspect	Temporal Consistency Detection	Spatial Consistency Detection
Focus	Analyzes motion and appearance changes across consecutive frames.	Examines visual coherence within individual frames.
Key Indicators	• Optical flow irregularities • Abrupt motion discontinuities • Temporal luminance or object appearance changes	• Lighting and shadow inconsistencies • Texture and noise anomalies • Edge and blending artifacts
Detection Techniques	• Optical flow estimation • Temporal differencing • Motion vector analysis from compression codecs	• Shadow geometry analysis • Texture descriptors and frequency domain analysis • Cloning and duplication detection • Edge detection methods
Strengths	• Effective at revealing frame-level tampering like deletion or insertion. • Captures dynamic inconsistencies in motion patterns.	• Good at detecting spatial forgeries such as object insertion or cloning. • Can identify lighting or shadow manipulations that temporal methods miss.
Limitations	• Can produce false positives in videos with rapid motion or scene changes. • Sensitive to camera shakes and natural motion noise.	• Performance affected by heavy compression or video noise. • May struggle with complex lighting or natural shadows.
Typical Applications	• Frame deletion/insertion detection • Video temporal integrity verification	• Object insertion/cloning detection • Scene lighting authenticity verification

Table 2.2: Comparative Overview of Temporal vs Spatial Consistency Detection Methods

While temporal and spatial consistency checks are helpful for revealing a range of manipulations, they might not work as well in videos with complex motions, abrupt scene changes, or a lot of compression noise. Furthermore, complex filtering and model calibration are required to prevent false positives caused by natural events like camera tremors or changes in ambient lighting.

2.3 Limitations of Classical Approaches

Although classical approaches to video forgery detection laid the foundation for early forensic analysis, they suffer from several important limitations [46]. With the increasing complexity of modern video content, and the rise of sophisticated tampering techniques, these traditional methods often fail to keep up. In this section, we highlight the most common and significant drawbacks of classical methods.

2.3.1 Sensitivity to Noise

Classical detection techniques typically rely on the analysis of specific low-level signals such as sensor noise patterns, motion inconsistencies, or compression artifacts. However, these signals are extremely sensitive to common forms of video degradation.

Real-world videos often undergo compression (e.g., H.264, HEVC), stabilization, filtering, resizing, or re-encoding before they are shared or stored. These processes can introduce new types of noise or distort the very signals that forensic tools rely on. As a result, traditional algorithms may fail to detect forgery when the evidence is hidden behind post-processing operations. In some cases, they may even mistake compression noise for tampering, producing false positives [45].

Additionally, environmental factors like low lighting, motion blur, or camera shaking can weaken visual consistency, further reducing the reliability of noise-based detection methods.

2.3.2 Ineffectiveness Against Subtle and Advanced Manipulations

Modern tampering techniques are no longer limited to simple frame insertion or cropping. Today's attackers use advanced tools such as generative adversarial networks (GANs), deepfake models, temporal smoothing, and content-aware editing. These techniques can generate forged content that looks visually consistent and natural, making it extremely difficult for traditional methods to detect anomalies.

For instance, frame interpolation can create smooth transitions between frames, hiding the evidence of insertion or deletion. AI-based face replacement can preserve lighting and motion consistency, bypassing many classical forensic checks.

Because traditional detection systems focus on low-level artifacts (noise, macroblocks, motion vectors), they often fail to catch these subtle manipulations [47]. As forgery methods become more intelligent, there is a growing need for adaptive, learning-based systems that can detect deep-level inconsistencies beyond what classical algorithms are capable of [46].

2.3.3 High Dependency on Manual Analysis

Another drawback is that many classical techniques require manual tuning, visual inspection, or expert interpretation. For example, some methods may highlight inconsistencies in compression patterns or detect motion vector irregularities but it is up to a forensic analyst to decide whether those inconsistencies are due to tampering or natural causes.

This not only increases the workload but also introduces subjectivity and potential for human error. In contrast, modern AI-based systems are increasingly capable of automating such decisions with better consistency and scalability [46].

2.3.4 Lack of Real-Time Detection Capabilities

Most classical techniques are not designed for real-time use. They often require extracting frames, computing statistical models, and analyzing patterns manually or semi-manually. This makes them unsuitable for modern use cases like live video monitoring, surveillance, or on-the-fly verification of dashcam videos.

As the demand grows for faster and more scalable detection, especially in the context of streaming media or smart city applications, the inability of classical methods to operate in real-time becomes a serious bottleneck [45].

2.4 Deep Learning in Video Forgery Detection

Deep learning has revolutionized the field of multimedia forensics [46, 45] , particularly in the detection of video tampering. Unlike classical methods, which rely on manually engineered features and domain-specific heuristics, deep learning models can automatically learn meaningful patterns directly from raw data. Convolutional Neural Networks (CNNs), in particular, have demonstrated outstanding performance in identifying both spatial and temporal inconsistencies in forged videos.

2.4.1 Video Forgery Detection Models

Recent advancements in deep learning have led to the development of a wide range of models for detecting forged content in videos. These models are tailored to capture either spatial anomalies in individual frames or temporal inconsistencies across frame sequences [1]. Based on their architecture and processing strategy, deep learning models for video forgery detection can be classified into the following categories:

2D Convolutional Neural Networks (2D CNNs)

2D CNNs operate on individual frames extracted from a video, treating each frame as a separate image. These models are effective at capturing low-level spatial anomalies such

as unnatural textures, blur, color mismatch, and artifacts introduced during tampering. Commonly used architectures include VGG16, ResNet50, and EfficientNet.

- These models are typically fine-tuned using transfer learning from large-scale image datasets (e.g., ImageNet).
- Their ability to detect visual inconsistencies makes them suitable for detecting frame insertion, frame deletion, and localized deepfake regions.
- They are often used as the first step in more complex pipelines that integrate temporal analysis later.

Despite their simplicity, 2D CNNs lack temporal awareness, making them less effective for manipulations involving frame order or motion inconsistencies.

3D Convolutional Neural Networks (3D CNNs)

Unlike 2D CNNs, 3D CNNs process a sequence of frames simultaneously, applying 3D convolutional kernels across both spatial and temporal dimensions. This enables the model to learn motion cues, object trajectories, and scene dynamics that are often disturbed during forgery [17].

- 3D CNNs are particularly effective for detecting duplications, deletions, and tampered segments that disrupt temporal flow.
- Models like C3D and I3D have been successfully applied to forgery detection tasks due to their ability to model short-term temporal dependencies.
- They require more computational resources but offer a deeper understanding of motion irregularities.

CNN-RNN Hybrid Models

Hybrid models combine Convolutional Neural Networks (CNNs) with Recurrent Neural Networks (RNNs), such as LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit), to capture long-range temporal dependencies.

- CNNs are used to extract spatial features from each frame or block of frames.
- The resulting features are passed to RNNs that model how these features evolve over time, detecting inconsistent motion patterns, frame-level tampering, or scene changes.
- Models like CLRNet and ConvLSTM have demonstrated high performance in spatiotemporal forgery detection [20, 51].

These architectures are more adaptable to real-world video data, where tampering may span across long durations and vary in intensity.

Transformer-Based and Attention-Driven Models

With the rise of vision transformers [62], several forgery detection systems have begun using self-attention mechanisms to model long-range dependencies across frames.

- Transformers excel at identifying subtle manipulations by learning context-aware relationships across distant regions in a video.
- Models such as GenConViT incorporate convolutional layers for local feature extraction and attention layers for global reasoning [52].
- Attention mechanisms also improve interpretability by highlighting suspicious regions in frames.

These models offer a balance between spatial and temporal modeling, often outperforming pure CNN-based architectures, especially on challenging datasets.

Hybrid and Optimization-Based Models

In pursuit of better performance, some researchers have introduced hybrid pipelines that integrate multiple models or apply optimization techniques to fine-tune parameters.

- A notable example is the Jaya-optimized CNN, which applies a metaheuristic optimization algorithm to adjust weights for improved convergence and accuracy [53].
- Ensemble learning techniques combine predictions from various architectures (e.g., 2D CNN + LSTM + Transformer) to boost robustness.
- Such systems are particularly useful in forensic applications where detection sensitivity is critical.

These models are typically more complex but offer enhanced accuracy, stability, and resistance to adversarial forgeries.

2.4.2 Convolutional Neural Networks (CNNs)

A class of deep learning models called convolutional neural networks was created especially to process data with a grid-like topology, like pictures and videos. From basic edges and textures in the early layers to intricate shapes and semantic patterns in the deeper layers, CNNs are made up of several layers that automatically learn hierarchical features [17].

The main components of CNNs are fully connected layers that carry out high-level reasoning for classification or regression tasks, pooling layers that reduce spatial dimensions while maintaining crucial information, and convolutional layers that extract spatial features using learnable filters as shown in Figure 2.2.

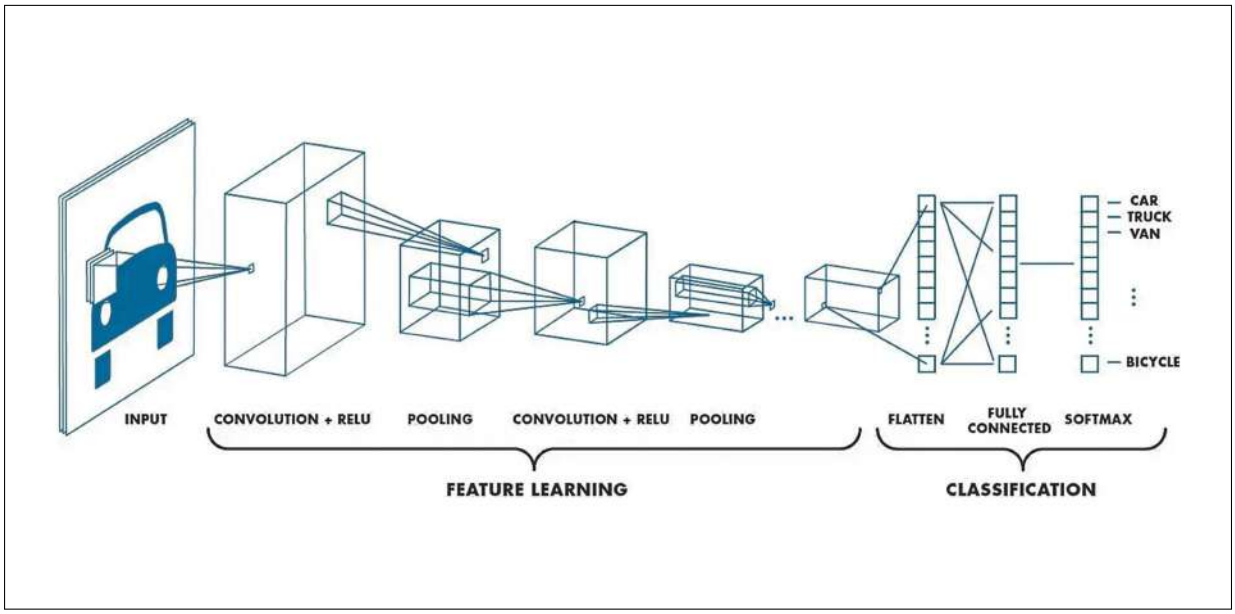


Figure 2.2: CNN architecture illustrating feature learning and image classification [59].

Because CNNs perform better than hand-crafted feature-based methods and can generalize across different visual conditions, they have become the foundation of many state-of-the-art computer vision applications.

2.4.3 CNN Architectures for Video Forgery Detection

The detection of video forgery fundamentally involves identifying subtle and often complex anomalies in visual content that are difficult to explicitly define or model. CNNs are well-suited for this task for several reasons:

Automatic Feature Extraction

Unlike classical methods that rely on manually engineered features, CNNs automatically learn discriminative features optimized for the detection task during training, enabling them to capture intricate patterns of manipulation.

Robustness to Variability

CNNs can learn to be invariant to various transformations, lighting conditions, and noise, which are common challenges in real-world video data.

Spatial and Temporal Modeling

When extended to process sequences of frames or combined with temporal models, CNNs can capture both spatial inconsistencies within frames and temporal anomalies across frames, enhancing detection capabilities [46, 1].

Scalability and Adaptability

CNN-based methods can be trained on large datasets encompassing diverse forgery types, making them adaptable to new and evolving manipulation techniques.

2.4.4 Datasets for Video Forgery Detection

High-quality datasets are essential for training and evaluating deep learning models:

Kaggle Video Forgery Dataset

This dataset includes over 9,000 dashcam videos categorized as Original or Forged, with forgery types such as deletion, insertion, duplication, flipping, zooming, and rotation. It is particularly useful for vehicle surveillance and real-world tampering scenarios.

FaceForensics++

A large-scale dataset with different face manipulation techniques (DeepFake, Face2Face, NeuralTextures), annotated at the frame level [22].

DeepFake Detection Challenge (DFDC)

Released by Facebook AI, DFDC contains 100,000+ deepfake videos generated using various face synthesis methods [23].

Celeb-DF v2 and ViFake

Celeb-DF v2 offers high-resolution facial forgery videos with greater realism and challenging conditions. ViFake provides full-frame manipulation annotations, supporting both classification and localization tasks.

UADFV

An early dataset containing real and fake videos, often used for benchmarking binary classifiers.

These datasets provide standard benchmarks for evaluating deep learning systems based on metrics such as accuracy, F1-score, and AUC.

2.5 Conclusion

This chapter has presented a structured overview of video forgery detection methods, ranging from traditional forensic techniques to advanced deep learning approaches. Classical methods such as PRNU-based fingerprinting [15], compression artifact analysis, and temporal-spatial consistency checks [46] have played a foundational role in the early stages of digital forensics. These techniques rely on low-level signals and assumptions about the acquisition and encoding process, which can be effective in controlled settings but struggle with modern challenges.

As video manipulation techniques have evolved—particularly with the emergence of AI-generated forgeries and high-quality compression pipelines—these traditional approaches have shown limitations. They are often sensitive to post-processing, lack real-time applicability, and fail to generalize across diverse devices and formats [47].

In response to these limitations, the field has increasingly embraced deep learning. Convolutional Neural Networks (CNNs), along with their extensions such as 3D CNNs, CNN-RNN hybrids, and attention-based models, have demonstrated superior capabilities in capturing complex spatial and temporal features. Supported by large-scale datasets like DFDC, Celeb-DF, and the Kaggle Video Forgery Dataset, these models offer better scalability, automation, and performance in real-world conditions [20, 52, 53].

Building upon the insights discussed in this chapter, the next part of this master thesis introduces the design and implementation of a deep learning model specifically developed to detect frame deletion in dashcam videos. It details the architecture, training strategy, and evaluation process, aiming to bridge the gap between theoretical advancements and practical forgery detection applications.

Chapter 3

CNN Architecture for Forgery Detection

3.1 Introduction

Following the review of traditional and deep learning-based video forgery detection techniques, this chapter presents the practical implementation of a deep learning system designed to detect frame deletion in dashcam videos. The proposed approach builds upon the theoretical insights discussed in previous chapters and focuses on a binary classification task: distinguishing between original and tampered videos.

To address this challenge, we developed a model named **BinaryCNN**, which leverages a pre-trained **ResNet50** architecture as a visual feature extractor. The model has been adapted for binary classification by modifying the final fully connected layer to distinguish between the "Original" and "Forgery Deletion" classes. This chapter details each component of the system architecture, from preprocessing to prediction.

The process begins with extracting frames from input videos and applying transformations such as resizing, normalization, and data augmentation. These preprocessed frames are then passed through the CNN model, which performs frame-level classification. To reach a final video-level decision, the model aggregates the frame predictions, producing a single output label for each video.

The training and evaluation were performed on a subset of the Kaggle Video Forgery Dataset, which contains both original and forged videos categorized by manipulation type. We focused specifically on detecting **frame deletion forgeries**, which are among the most subtle and challenging types of tampering. The dataset was carefully balanced and split into training and testing sets, with performance measured using accuracy, precision, recall, and F1-score.

In addition to the AI model itself, this chapter also describes the implementation of a web-based user interface using Flask. The interface allows users to upload dashcam videos and receive a real-time verdict on whether the video is authentic or manipulated. The

entire pipeline—from frame extraction to classification and result presentation—has been designed to be scalable and user-friendly, with potential applications in law enforcement and insurance sectors.

3.2 Presentation of the Proposed Solution

3.2.1 Objective

The major goal of our project is to create an automated system that can detect frame deletion in dashcam video, a prevalent form of video manipulation. Intentionally removing one or more frames from a video sequence in order to change the visual narrative is known as frame deletion, and it can cause problems for investigations or insurance claims. Because manipulated videos can erode trust and result in incorrect decisions, it is imperative to ensure the authenticity of dashcam footage for legal, forensic, and insurance applications.

Convolutional neural networks (CNNs), a type of deep learning, are used in the suggested solution to overcome this difficulty because of their demonstrated capacity to extract intricate visual patterns and features that more conventional handcrafted approaches might overlook. The system's purpose is to analyse videos at the frame level and determine whether they are "Original" (unaltered) or contain "Forgery Deletion" (frames that have been deleted).

3.2.2 Pipeline Overview

The detection process is structured as a sequential pipeline that transforms raw dashcam videos into a reliable authenticity decision through the following key stages:

Frame Extraction

Every dashcam video is broken down into separate frames, or pictures. A set number of frames, up to 10, are extracted uniformly throughout the video timeline to ensure consistency across videos of different lengths. The final frame is copied to make up the difference if a video has fewer than ten frames. This guarantees that the model's input size is constant.

Preprocessing of Frames

To get the extracted frames ready for the neural network, they go through a number of preprocessing steps:

- **Resizing :** To match the input size required by the backbone CNN (ResNet50), frames are resized to 224×224 pixels.

- **Normalization :** To stabilise and speed up model training, pixel values are normalised using the mean and standard deviation of the ImageNet dataset.
- **Data Augmentation :** Random augmentations like rotation, horizontal flipping, colour jittering (brightness, contrast, saturation), Gaussian blur, and random erasing (simulating occlusions) are used during training to improve model robustness and decrease overfitting.

Feature Extraction and Frame-Level Classification

The CNN model, which is based on the pre-trained ResNet50 architecture optimised for this task, receives each preprocessed frame separately. ResNet50 is a potent feature extractor that can identify minute irregularities suggestive of frame deletion as well as intricate spatial patterns. Each frame's output from the model is a prediction score for either the "Original" or "Forgery Deletion" classes.

Aggregation of Frame Predictions

A final video-level decision is generated by combining individual frame-level predictions because a video is represented by multiple frames. In particular, a strong consensus that takes temporal consistency into account is produced by averaging the prediction probabilities across all frames. By reducing the impact of any noisy or unclear frames, this averaging produces a more accurate classification of the full video.

Final Decision

The system determines whether the dashcam footage is authentic or tampered with based on the total score. Stakeholders like law enforcement, insurance companies, or forensic analysts can use this binary result to evaluate the credibility of videos.

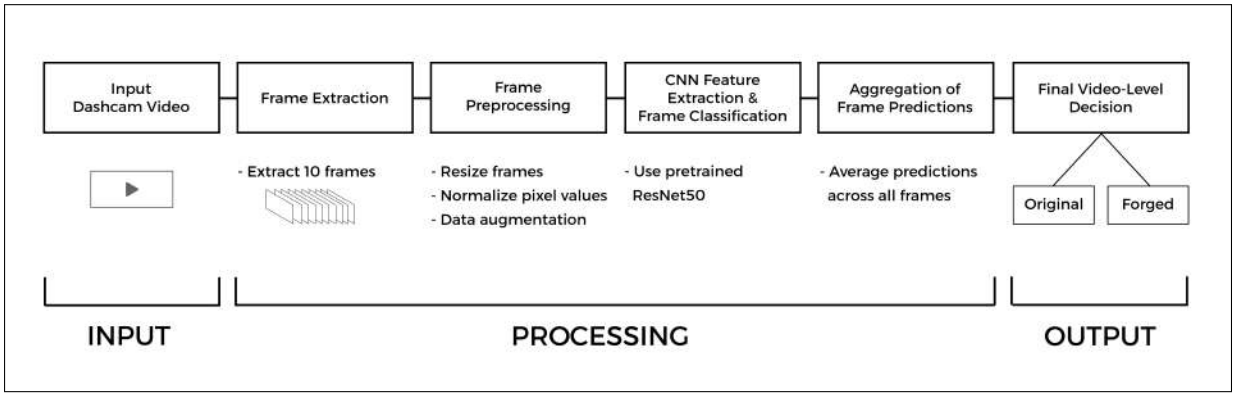


Figure 3.1: Video forgery detection pipeline using CNN and frame aggregation.

3.3 Dataset

3.3.1 Source

The Video Forgery Dataset is a publicly available collection hosted on Kaggle that was used in this study. This dataset, which comprises a large range of dashcam videos that represent real and manipulated footage, was especially selected for the task of video tampering detection. It is an appropriate benchmark for creating and assessing deep learning models targeted at video forgery detection because of its accessibility and thoroughness [57].

3.3.2 Dataset Structure and Composition

The dataset is structured hierarchically to reflect different classes and types of video tampering. Its main components are as follows:

Original Videos

This folder contains unaltered dashcam videos recorded under various real-world conditions, including different lighting, weather, and traffic scenarios. The total number of original videos is approximately 1181. These videos serve as the ground truth class and are critical for training the model to recognize genuine video characteristics.

Forgery Videos

The forgery folder is subdivided into several subdirectories, each corresponding to a distinct tampering technique applied to the original videos as shown in Figure 3.2. The primary subfolders include:

- **Frame Deletion :** Approximately 1181 videos where one or more frames have been intentionally removed to simulate tampering.
- **Frame Insertion :** Videos with additional frames inserted into the sequence.
- **Frame Duplication :** Videos containing duplicated frames to manipulate temporal continuity.
- **Transformations :** Videos altered by geometric transformations such as rotation, flipping, or zooming. For this study, the focus is exclusively on the Frame Deletion subset, which directly aligns with the detection objective.

Each forged video is derived from a corresponding original video, maintaining the same base content but with the specific tampering applied. This setup allows for precise

supervised learning and comparative analysis.
For this study, the focus is exclusively on the Frame Deletion subset, which directly aligns with the detection objective.

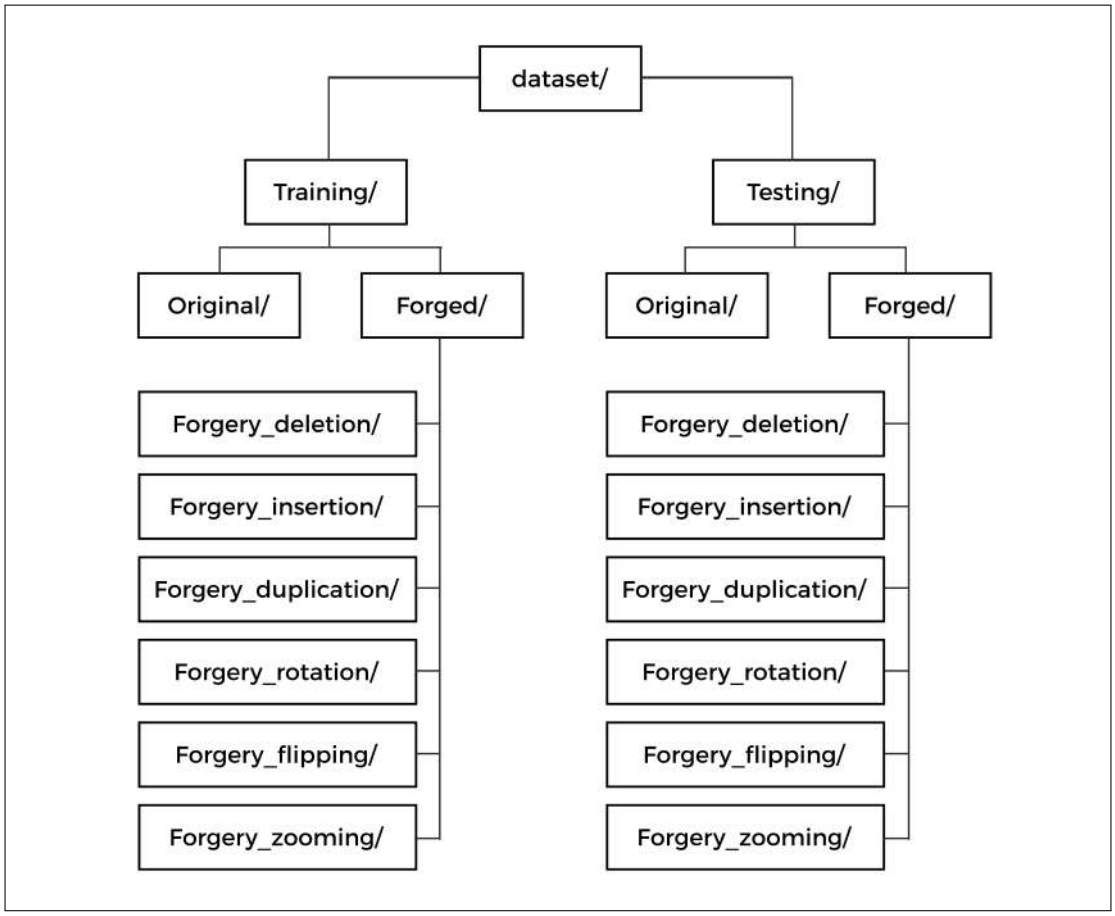


Figure 3.2: Structure of dataset folders showing original and forged classes.

3.3.3 Data Partitioning

To train and evaluate the model effectively, the dataset is partitioned into training and testing subsets with a balanced distribution:

Training Set

includes 363 original videos and 363 forged videos (frame deletion) for a total of 726 videos. This balanced split facilitates unbiased representation learning by guaranteeing that the model is exposed to an equal amount of tampered and authentic data during learning.

Testing Set

contains 726 videos as well, with the original and forged classes remaining split 50/50. For a fair and thorough assessment of the model's classification performance on unseen

data, this balance is essential.

3.3.4 Data Preprocessing

Before feeding the video data into the deep learning model, a comprehensive preprocessing pipeline is applied to standardize inputs and enhance the model's robustness.

Frame Extraction

Videos are collections of pictures, or frames. To obtain a representative sample of temporal content, up to 10 frames are randomly selected from each video. To ensure uniform input size and facilitate batch processing, videos with fewer than 10 frames have their last extracted frame duplicated to reach the fixed count.

Frame Resizing

To match the input resolution required by the core ResNet50 architecture, all extracted frames are resized to 224 x 224 pixels. This resizing standardises the input size while maintaining important spatial features.

Normalization

The mean ([0.485, 0.456, 0.406]) and standard deviation ([0.229, 0.224, 0.225]) of the ImageNet dataset are used to normalise pixel intensity values in frames. By bringing the input data distribution into line with the expectations of the pretrained ResNet50 model, this normalisation enhances training stability and convergence speed.

Data Augmentation (Training Phase Only)

Several random augmentations are used throughout the training process to lessen overfitting and enhance model generalisation, including:

- Horizontal flipping to simulate viewpoint variations.
- Random rotations up to 30 degrees to mimic camera tilts.
- Adjustments to brightness, contrast, and saturation to account for different lighting conditions.
- Gaussian blur to simulate motion or focus degradation.
- Random erasing to emulate occlusions or noise.

3.4 CNN Model Architecture

3.4.1 Pretrained ResNet50

The base architecture chosen for this work is ResNet50 [24, 25], a 50 layer deep convolutional neural network that uses residual learning to efficiently train very deep architectures, forms the foundation of our model. ResNet50, which was first presented by He et al.(2016), learns complex hierarchical features from input images by utilising skip connections, or residual blocks, which mitigate the vanishing gradient issue.

The ImageNet dataset, which comprises more than a million labelled images in 1000 classes, was used to pretreat ResNet50. Through transfer learning, the model can extract general visual features like edges, textures, shapes, and object parts from this pretraining and apply them to other vision tasks. Given the small size of the video forgery dataset, using this pretrained model greatly improves performance and cuts down on training time.

3.4.2 Modifications for Binary Classification and Regularization

To adapt ResNet50 to the specific task of binary classification between "Original" and "Frame Deletion Forgery," we have modified the final layers of the network as follows:

```
# Model Definition
class BinaryCNN(nn.Module):
    def __init__(self):
        super(BinaryCNN, self).__init__()
        self.base_model = models.resnet50(weights=models.ResNet50_Weights.IMAGENET1K_V1)
        self.base_model.fc = nn.Linear(self.base_model.fc.in_features, 2)  # 2 classes: Original + Forgery_deletion
        self.dropout = nn.Dropout(0.5)
```

Figure 3.3: Modified ResNet50 with binary classification and dropout regularization.

Final Fully Connected Layer Replacement

A new linear layer with two output neurones, representing the two target classes, was used in place of the original fully connected (FC) layer, which was intended to classify 1000 ImageNet classes. During training and inference, a softmax function transforms the logits that are directly produced by this output layer into probabilities.

Dropout Layer Addition

A dropout layer with a dropout rate of 0.5 was added before the final classification layer to avoid overfitting and enhance the model's capacity for generalisation. During training, dropout randomly sets half of the neurone activations to zero, which forces the model to create redundant representations and lessens its dependence on particular neurones.

3.4.3 Functioning of BinaryCNN

Our model, named BinaryCNN, takes as input individual video frames extracted from dashcam footage. Each frame undergoes preprocessing steps, including:

- Resizing to 224 × 224 pixels to comply with the input size expected by ResNet50.
- Normalization using the mean and standard deviation values from the ImageNet dataset to standardize pixel intensity distribution.
- Data augmentation techniques (during training only), such as random horizontal flips, rotations, brightness and contrast adjustments, Gaussian blur, and random erasing, are applied to increase the model's robustness and reduce overfitting.

The preprocessed frames are fed into the network, which processes them through the ResNet50 layers followed by the dropout and final binary classification layer. The output is a probability distribution over the two classes, indicating the likelihood of the frame being "Original" or containing "Forgery Deletion."

3.4.4 Video-Level Classification Strategy

Since videos contain multiple frames, classifying the entire video based on a single frame may not be reliable, particularly because manipulations may affect only some frames. To address this, the model uses a frame-level classification aggregation strategy as follows:

```
def forward(self, x):
    batch_size, num_frames, channels, height, width = x.size()
    x = x.view(batch_size * num_frames, channels, height, width)  # Flatten frames
    outputs = self.base_model(x)  # Shape: (batch_size * num_frames, 2)
    outputs = outputs.view(batch_size, num_frames, -1)  # Shape: (batch_size, num_frames, 2)
    (variable) outputs: Any
    outputs = outputs.mean(dim=1)  # Average predictions across frames
    return outputs
```

Figure 3.4: Frame-level predictions aggregated via mean pooling for final video classification.

- Up to 10 frames are uniformly sampled from each video to represent its content.
- Each frame is individually passed through the BinaryCNN model to obtain frame-level class probabilities.
- The final video-level prediction is computed by averaging the predicted probabilities across all sampled frames.
- The class with the highest average probability is assigned as the predicted class for the video.

This approach balances computational efficiency and temporal coverage, allowing the model to leverage multiple snapshots from the video without requiring complex temporal models. It also reduces the impact of noisy or ambiguous frames on the overall prediction.

3.5 Model Training

3.5.1 Parameters

The training of the BinaryCNN model was conducted with carefully chosen hyperparameters to ensure stable convergence and effective learning, given the complexity of the task and the dataset characteristics:

```
# Training Loop
for epoch in range(num_epochs):
    model.train()
    running_loss = 0.0
    for frames, labels in tqdm(train_loader, desc=f"Epoch {epoch+1}/{num_epochs}"):
        frames = frames.to(device)
        labels = labels.to(device)
        (variable) outputs: Any
        outputs = model(frames)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
```

Figure 3.5: Main training loop used to optimize the BinaryCNN model.

Batch Size

16 videos were selected as the batch size. After preprocessing, each video contributes 10 frames, so the model can process 160 frames in a batch. This size strikes a compromise between the statistical dependability of gradient estimates and memory limitations. While larger batches may require more memory and result in overfitting on stable gradients, smaller batches may produce noisy gradients.

Epochs

Forty epochs were used to train the model. Initial experiments using the training and validation loss curves allowed for the empirical determination of this number. Training with fewer epochs produced underfitting, while training with 30 epochs produced diminishing returns and mild overfitting.

Learning Rate

0.00005 (5×10^{-5}) was the initial learning rate. Because ResNet50 is pretrained, a low learning rate is necessary to delicately adjust the weights without erasing the pretrained

information. In subsequent work, the rate could be dynamically adjusted based on validation performance using a learning rate scheduler (e.g., StepLR or ReduceLROnPlateau).

Weight Decay

A tiny weight decay (L2 regularisation) factor of 1e-4 was applied to lessen overfitting. By penalising large weights, this encourages simpler models and regularises the network parameters.

3.5.2 Loss Function

The model optimizes the **Cross-Entropy Loss**, which measures the difference between predicted class probabilities and true class labels [17].

```
class_weights = torch.tensor([1.0, 1.5]).to(device)  # Higher weight for Forgery_deletion
criterion = nn.CrossEntropyLoss(weight=class_weights)
```

Figure 3.6: Weighted cross-entropy loss to emphasize detection of forged frames.

The weighted Cross-Entropy Loss L for a batch of N samples is defined as:

$$L = -\frac{1}{N} \sum_{i=1}^{N} \left(w_0 \, y_i^{(0)} \log(p_i^{(0)}) + w_1 \, y_i^{(1)} \log(p_i^{(1)}) \right)$$

where:

- $y_i^{(c)} \in \{0, 1\}$ indicates whether the i-th frame belongs to class c,
- $p_i^{(c)}$ is the predicted probability for class c of the i-th frame,
- w_0 and w_1 are class weights for the "Original" and "Forgery Deletion" classes, respectively.

Class weights help to balance the influence of each class during training [42], especially important when classes are imbalanced or when one class is more critical.

3.5.3 Optimization

The Adam algorithm, which is an extension of stochastic gradient descent and estimates the first and second moments of the gradients to maintain adaptive learning rates for each parameter, was used to optimize the model parameters [43].

Key advantages of Adam include:

- Faster convergence due to adaptive learning rates,

- Stability on sparse gradients,
- Less need for manual tuning compared to vanilla SGD.

The Adam hyper-parameters were set as follows:

- Initial learning rate: 0.00005
- Exponential decay rates for moment estimates: $\beta_1 = 0.9$, $\beta_2 = 0.999$
- Epsilon (to prevent division by zero): $\epsilon = 10^{-8}$

These values follow the recommended defaults from Kingma and Ba (2015).

3.5.4 Handling Class Imbalance with Weighted Loss

Although the dataset was balanced in terms of the number of original and forged videos, **frame-level imbalance** can still occur due to the following reasons:

- Not all frames in forged videos exhibit obvious tampering signals,
- Preprocessing and augmentation can shift class distributions,
- Some classes may be inherently harder to learn.

To mitigate this, the loss function incorporates **class weights** to give higher importance to the "Forgery Deletion" class during training. This encourages the model to reduce **false negatives**, which is crucial in forensic applications where missing a forged video has significant consequences.

The weights were determined empirically by analyzing class distribution and performance on validation data. Typically, the weight for the forgery class w_1 is set higher than the weight for the original class w_0, for example:

$$w_0 = 1, \quad w_1 = \frac{\text{Number of original frames}}{\text{Number of forgery frames}}$$

This weighting scheme balances the effective contribution of each class to the loss.

3.6 Model Evaluation

3.6.1 Evaluation Metrics

Several common metrics that are frequently used in binary classification problems, particularly in the field of video forensics, were used to evaluate the BinaryCNN model's performance [44]. A thorough understanding of the model's performance on each class

is provided by these metrics, which are especially helpful in security applications where incorrect classifications can have serious repercussions. They also aid in assessing overall accuracy.

Accuracy

Accuracy measures the overall proportion of correct predictions over all predictions made:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

The accuracy achieved on the test set of 726 videos was:

$$\boxed{\text{Accuracy} = 79\%}$$

This indicates that approximately 4 out of 5 videos were correctly classified.

Precision

Precision focuses on the quality of the positive predictions how many of the videos predicted as "forged" actually are forged:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Forgery Deletion:** $\boxed{81\%}$
- **Original:** $\boxed{76\%}$

Recall (Sensitivity)

Recall indicates the model's ability to detect all relevant case how many of the actual forged videos were correctly detected:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **Forgery Deletion:** $\boxed{80\%}$
- **Original:** $\boxed{77\%}$

F1-Score

The F1-score combines both precision and recall into a single metric, especially useful when both false positives and false negatives matter:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Forgery Deletion:** 80.5%
- **Original:** 76.5%

Results Summary Table

Metric	Original	Forgery Deletion
Precision	76%	81%
Recall	77%	80%
F1-Score	76.5%	80.5%

Table 3.1: Per-class performance metrics of BinaryCNN

Overall Accuracy: 79%

3.6.2 Performance Interpretation

The BinaryCNN model performs well in both classes and has a high degree of accuracy. Its over 80% precision and recall in detecting forged videos makes it especially useful in real-world situations like insurance fraud detection or police investigations.

Strengths

- High recall for forgeries (80%) ensures most tampered videos are correctly detected.
- Strong precision (81%) means few original videos are wrongly flagged.
- Balanced performance across classes confirms model reliability.

Limitations

- Slightly weaker performance on the "Original" class, indicating conservative predictions to avoid missing forgeries.
- Evaluation is limited to frame deletion generalization to other forgery types remains untested.

Future Improvements

- Extend the model to detect additional types of video forgery, such as frame insertion, duplication, geometric transformations, and **deepfake** manipulations, to broaden applicability.
- Enhance model performance to achieve higher accuracy and more robust detection across diverse forgery scenarios.

- Integrate temporal models, such as Long Short-Term Memory (LSTM) networks or 3D Convolutional Neural Networks (3D-CNN), to better capture temporal dependencies and contextual information within video sequences.
- Develop a more user-friendly and feature-rich interface to facilitate real-time analysis and improve usability for end-users, such as law enforcement and insurance investigators.

3.7 Application Deployment

3.7.1 User Interface

A user-friendly web interface was created to enable end users, including forensic analysts, insurance investigators, and law enforcement officers, to access the video forgery detection system. The interface's primary features include:

Video Upload

Users can upload dashcam video files in common formats (e.g., MP4, AVI) through a clean and intuitive upload form.

Result Display

After uploading, the system processes the video in the background and displays the final prediction result, indicating whether the video is Original or contains Forgery Deletion.

Privacy Considerations

The interface does not display the video content itself, ensuring that sensitive footage remains confidential during the analysis.

Professional Layout

The UI is designed to be clear, professional, and straightforward, minimizing user training and maximizing usability as shown in Figure 3.8.

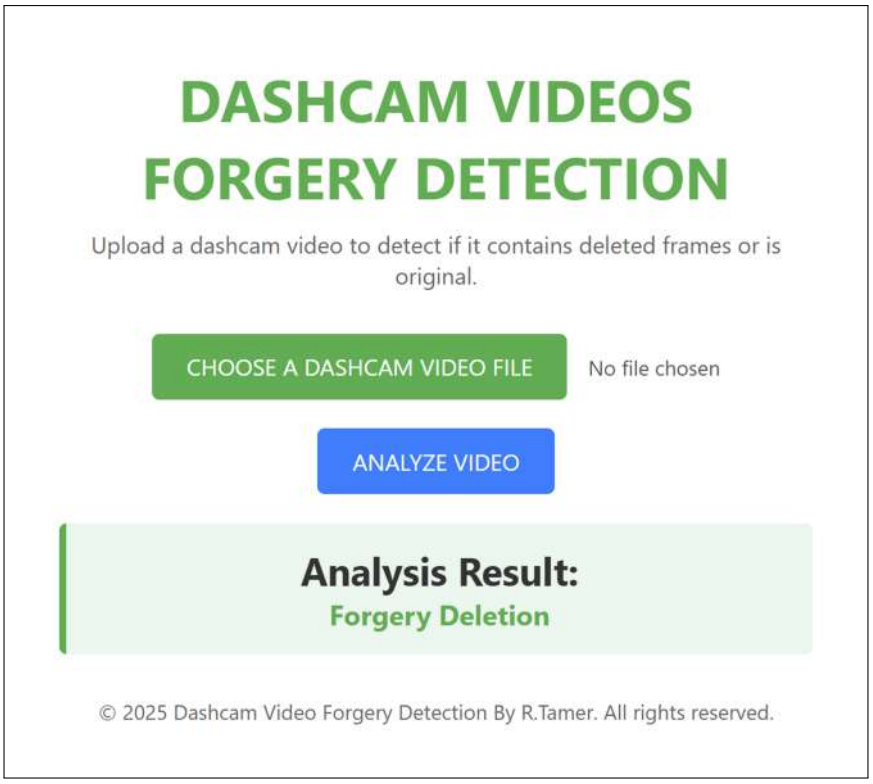


Figure 3.7: Flask web interface for uploading dashcam videos and displaying the classification result.

3.7.2 Software Tools

To ensure efficient deployment, scalability, and ease of use, the proposed system relies on a lightweight and modular software stack that bridges deep learning models with an interactive web interface. This architecture allows both technical and non-technical users—such as law enforcement agents or insurance investigators—to easily verify the authenticity of dashcam videos.

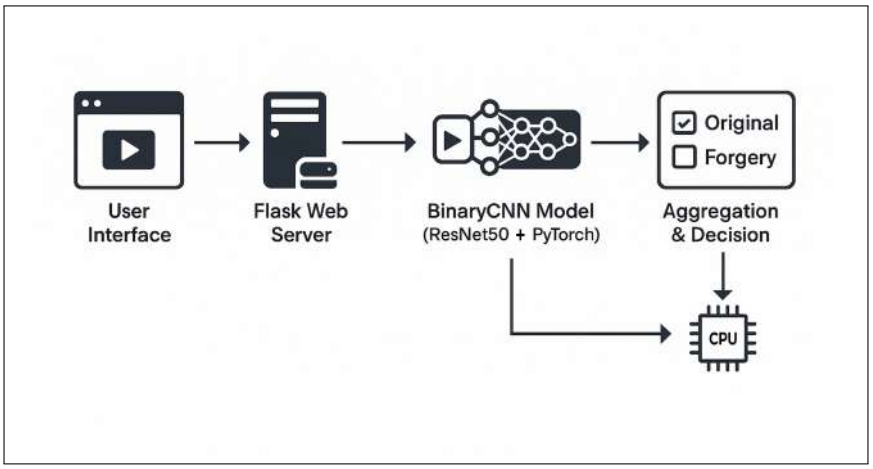


Figure 3.8: System architecture of the proposed video tampering detection solution.

Back-End: Flask and PyTorch Integration

The back-end is developed using **Flask**, a Python-based micro web framework known for its simplicity, flexibility, and support for RESTful APIs [48]. Flask is responsible for:

- Receiving and validating video files uploaded by users.
- Preprocessing the videos (extracting and resizing frames).
- Loading the pre-trained deep learning model (**BinaryCNN**) built using **PyTorch** [49].
- Performing inference on extracted frames.
- Aggregating frame-level predictions into a final decision.
- Sending the results back to the user interface.



Figure 3.9: Flask and Pytorch [66].

Thanks to Flask's minimal overhead, the system remains fast and responsive, making it ideal for real-time or near-real-time video analysis.

Front-End: Web Interface

The front-end of the system is implemented using standard web technologies:

- **HTML5** is used for page structure and form elements that allow video file uploads.
- **CSS3** provides styling and layout, ensuring a professional and user-friendly design.
- **JavaScript** adds interactivity and handles asynchronous requests (e.g., using AJAX or Fetch API) to communicate with the Flask server.



Figure 3.10: HTML5, JavaScript, and CSS3 [65].

This combination ensures compatibility across modern browsers and provides a smooth experience across desktops and mobile devices. The user interface was designed to require no technical knowledge, making the tool accessible to forensic analysts and field agents alike.

3.7.3 Use Cases

Several practical scenarios illustrate the utility of the deployed application:

Police Investigations

Officers can upload dashcam footage from traffic incidents to quickly verify the authenticity of videos before using them as evidence.

Insurance Claims Verification

Insurance adjusters can analyze submitted dashcam videos for tampering, reducing fraudulent claims and speeding up settlement processes.

Forensic Analysis

Digital forensics experts can use the tool as an initial screening step to flag suspicious videos for more in-depth examination.

Public Awareness and Education

The interface can be adapted for educational purposes, demonstrating how video tampering can be detected and raising awareness about digital evidence integrity.

3.8 Conclusion

This chapter has detailed the design, training, and evaluation of a CNN-based system for detecting frame deletion in dashcam videos. The model, named BinaryCNN, uses a pre-trained ResNet50 as its backbone and was tailored to perform binary classification between original and tampered videos. By extracting ten representative frames per video, applying preprocessing steps (including resizing, normalization, and augmentation), and aggregating predictions at the video level, the system was able to achieve strong results in terms of detection accuracy.

The evaluation conducted on a well-structured subset of the Kaggle Video Forgery Dataset demonstrated the model's ability to generalize well to unseen data. With an overall classification accuracy of **79%**, along with precision, recall, and F1-scores above **76%** for both classes, the system shows promising performance in realistic scenarios. The weighted loss function and data augmentation techniques helped address issues related to class imbalance and overfitting, contributing to the model's robustness [20].

Furthermore, the deployment of a web-based interface enhances the system's usability by enabling users to upload videos and receive authenticity evaluations without requiring technical expertise. This makes the tool especially valuable for stakeholders such as traffic police, insurers, and legal professionals who need quick and reliable verification of video evidence.

While the system is currently focused on detecting frame deletion, the modularity of the architecture allows for future extensions to include other types of video forgeries such as insertion, duplication, and even deepfakes. Potential improvements also include integrating temporal modeling techniques such as LSTM or 3D CNNs to better capture motion inconsistencies and expanding training with additional datasets to enhance generalization.

Through this implementation, the thesis moves from theoretical analysis to practical contribution, demonstrating the feasibility of deep learning approaches in real-world video forensics.

General Conclusion

This master thesis addressed the critical issue of video tampering in the context of dashcam footage, focusing on the detection of frame deletion using deep learning techniques. Given the growing reliance on video evidence in legal and insurance processes, the integrity of such footage has become a vital concern. Manipulated videos can distort events, obscure truth, and mislead investigations, thereby undermining public trust in digital media.

The first chapter explored the technical foundation and societal implications of video forgery. It categorized tampering techniques into classical manipulations (deletion, insertion, duplication) and modern AI-driven methods (deepfakes, GAN-based alterations), and discussed their potential consequences for judicial, social, and cybersecurity domains [26, 7].

Chapter 2 reviewed the existing detection methods, comparing classical forensic techniques with modern deep learning approaches. While traditional methods—such as PRNU, SPN, and compression artifact analysis—offer valuable insights, they are often insufficient against subtle or AI-generated manipulations. This gap justifies the use of Convolutional Neural Networks (CNNs), which can extract rich visual features and detect inconsistencies more effectively [46, 20].

In Chapter 3, we proposed and implemented the *BinaryCNN* model—a CNN-based solution derived from a pretrained ResNet50. This model was trained and evaluated on the Kaggle Video Forgery Dataset, focusing on binary classification between original and frame-deleted videos. The preprocessing pipeline, model structure, and classification strategy were thoroughly explained. The model achieved promising results, including an overall accuracy of 79%, precision of 81%, recall of 78%, and an F1-score of 79%, confirming its effectiveness in detecting frame-level forgery under real-world conditions.

Additionally, a user-friendly web interface was developed using Flask, allowing stakeholders (e.g., law enforcement or insurers) to upload dashcam videos and receive real-time feedback on their authenticity. This deployment bridges the gap between research and practical application, making the system accessible and usable by non-experts.

Despite these achievements, some limitations remain. The current system focuses only on frame deletion. Future research could extend detection to other manipulation types, such as insertion, duplication, and deepfakes. Incorporating temporal models like LSTMs or 3D CNNs may also improve performance. Moreover, evaluating the model on larger,

more diverse datasets would further validate its robustness.

In conclusion, this master thesis contributes a functional and scalable approach to video forgery detection using CNNs. It highlights the effectiveness of deep learning in forensic video analysis and offers a solid foundation for future improvements aimed at safeguarding the authenticity of video evidence in critical decision-making environments.

Bibliography

[1] X. Zhou, R. Han, V. Chandrasekhar, "A Survey on Deepfake Detection," *IEEE Transactions on Multimedia*, vol. 23, pp. 2028–2046, 2021.

[2] D. Afchar, V. Nozick, J. Yamagishi, I. Echizen, "MesoNet: a Compact Facial Video Forgery Detection Network," in *Proc. IEEE International Workshop on Information Forensics and Security (WIFS)*, 2020.

[3] I. E. Richardson, A. Vetro, "Video Compression: Techniques and Standards," *IEEE Signal Processing Magazine*, vol. 39, no. 1, pp. 10–20, 2022.

[4] G. J. Sullivan et al., "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 3–21, 2020.

[5] J. Bankoski, R. L. Fowler, "The AV1 Codec Overview and Performance," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, 2020.

[6] T. T. Nguyen, T. Le, J. Shin, "Deepfake Detection Using Deep Learning: A Survey," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–42, 2021.

[7] P. Korshunov, S. Marcel, "Deepfakes: A New Threat to Face Recognition? Assessment and Detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.

[8] L. Verdoliva, "Media Forensics and Deepfakes: An Overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 910–932, 2020.

[9] L. Guarnera, G. Poggi, L. Verdoliva, "Video Forgery Detection for Insurance Claims: Challenges and Methods," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3545–3558, 2021.

[10] J. Kietzmann et al., "Deepfakes and Trust in the Digital Age," *Journal of Business Ethics*, vol. 175, pp. 365–381, 2022.

[11] X. Li et al., "Compression Fingerprint-Based Video Forgery Detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, pp. 234–245, 2020.

[12] Brookings Institution, "The Deepfake Dilemma: Technology, Transparency and Trust," 2020. [Online]. Available: `https://www.brookings.edu/techstream/the-deepfake-dilemma/`. Accessed: May 2025.

[13] S. Frenda, E. Nichols, and E. Loftus, "Current Issues and Advances in Misinformation Research," *Current Directions in Psychological Science*, vol. 22, no. 1, pp. 36–41, 2013.

[14] H. Ajder et al., "The State of Deepfakes: Landscape, Threats, and Impact," Deeptrace Report, 2019. [Online]. Available: `https://www.sensity.ai/deepfake-report/`. Accessed: May 2025.

[15] G. Chierchia et al., "PRNU-Based Video Forgery Detection: Recent Advances and Challenges," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 365–378, 2020.

[16] S. Hussain et al., "Limitations of Classical Video Forgery Detection Techniques," *Journal of Visual Communication and Image Representation*, vol. 74, p. 103071, 2021,

[17] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2021.

[18] P. Zhou et al., "Learning Rich Features for Image Forgery Detection Using CNNs," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1183–1196, 2020.

[19] B. Bayar, M. Stamm, "A Deep Learning Approach to Detecting Tampered Images," in *IEEE Workshop on Information Forensics and Security (WIFS)*, 2020.

[20] T. Nguyen et al., "3D CNN and LSTM Hybrid Models for Video Tampering Detection," *IEEE Transactions on Multimedia*, vol. 24, pp. 3456–3468, 2022.

[21] C. Hsu et al., "Attention Mechanisms for Robust Video Forgery Detection," *Pattern Recognition*, vol. 132, p. 108902, 2023.

[22] A. Rossler et al., "FaceForensics++: Learning to Detect Manipulated Facial Images," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2020.

[23] B. Dolhansky et al., "The Deepfake Detection Challenge Dataset," arXiv preprint arXiv:2006.07397, 2020.

[24] K. He et al., "Deep Residual Learning for Image Recognition," in *Proc. CVPR*, 2016.

[25] X. Huang et al., "Deep Video Forgery Detection Using ResNet Architectures," *IEEE Access*, 2020.

[26] M. Chowdhury et al., "Recent Advances in Video Forgery Detection: A Review," *IEEE Access*, 2021.

[27] J. Shin, H. Kim, "Dashcam Video Analysis for Incident Detection and Forensics," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[28] R. Agarwal, M. B. Green, "Review of Dashcam Usage and Video Forgery Risks in Insurance and Legal Sectors," *Journal of Forensic Sciences*, 2022.

[29] K. R. Prajwal et al., "Wav2Lip: Accurately Lip-syncing Videos In The Wild," in *Proc. ACM International Conference on Multimedia (MM)*, 2020.

[30] S. Niklaus, L. Mai, and F. Liu, "Video Frame Interpolation via Adaptive Convolution," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[31] J. Xu et al., "Investigating Visual Hallucination in Video Inpainting Models," in *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2022.

[32] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection," *Information Fusion*, vol. 64, pp. 131–148, 2020.

[33] M. Westerlund, "The Emergence of Deepfake Technology: A Review," *Technology Innovation Management Review*, vol. 9, no. 11, pp. 39–52, 2019.

[34] M. Singer, "Generative AI for Video: Risks and Opportunities," *AI Ethics Journal*, vol. 2, no. 1, pp. 22–33, 2023.

[35] A. C. Bovik, "The Essential Guide to Video Processing," Academic Press, 3rd ed., 2022.

[36] A. Vetro and H. Sun, "Overview of Video Coding Standards," *IEEE Signal Processing Magazine*, 2021.

[37] J. Lukáš, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, 2006.

[38] E. Sabir et al., "Recurrent Convolutional Strategies for Temporal Consistency Analysis in Video Forensics," *IEEE Transactions on Image Processing*, 2021.

[39] E. Ilg et al., "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. CVPR*, 2017.

[40] E. Kee and H. Farid, "Exposing digital forgeries from 3-D lighting environments," in *Information Hiding*, Springer, 2013.

[41] Y. Zhao et al., "A Survey on Deep Learning-Based Video Forgery Detection," *IEEE Access*, 2021.

[42] M. Buda, A. Maki, and M. A. Mazurowski, "A Systematic Study of the Class Imbalance Problem in Convolutional Neural Networks," *Neural Networks*, 2018.

[43] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. ICLR*, 2015.

[44] M. Sokolova and G. Lapalme, "A Systematic Analysis of Performance Measures for Classification Tasks," *Information Processing & Management*, 2009.

[45] L. Verdoliva, "Media Forensics and DeepFakes: An Overview," *IEEE Journal of Selected Topics in Signal Processing*, 2020.

[46] B. Bayar, M. Stamm, "On the Robustness of Convolutional Neural Networks for Camera Model Identification," *IEEE Transactions on Information Forensics and Security*, 2021.

[47] H. Zhao, Y. Zhou, and W. Wang, "Detecting AI-generated Video Forgeries Using Deep Neural Networks," *IEEE Access*, 2021.

[48] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, O'Reilly Media, Inc., 2018.

[49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[50] A. Farah, "Deepfake Detection Using CNNs," GitHub Repository, 2023. [Online]. Available: `https://github.com/AhmedFarah5/deepfake-detection-cnn`. Accessed: May 28, 2025.

[51] S. Tariq, S. Lee, and S. S. Woo, "A Convolutional LSTM based Residual Network for Deepfake Video Detection," arXiv preprint arXiv:2009.07480, 2020.

[52] D. Wodajo, S. Atnafu, and Z. Akhtar, "Deepfake Video Detection Using Generative Convolutional Vision Transformer," arXiv preprint arXiv:2307.07036, 2023.

[53] Z. F. Hussain and H. R. Ibraheem, "Novel Convolutional Neural Networks based Jaya algorithm Approach for Accurate Deepfake Video Detection," Mesopotamian Journal of CyberSecurity, vol. 2023, pp. 35–39, 2023.

[54] Facebook AI, "DeepFake Detection Challenge Dataset," 2020. [Online]. Available: `https://ai.facebook.com/datasets/dfdc`. Accessed: May 28, 2025.

[55] Y. Li, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 3207–3216.

[56] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2019, pp. 1–11.

[57] R. Shah, "Video Forgery Dataset," Kaggle, 2020. [Online]. Available: `https://www.kaggle.com/datasets/rajshah1/video-forgery-dataset`. Accessed: May 28, 2025.

[58] J. Deng et al., "Illustration of Compression Artifacts in H.265/HEVC Videos," [Online]. Available: `https://www.researchgate.net/publication/342540674`. Accessed: May 2025.

[59] S. Patil, "Understanding Convolutional Neural Networks (CNN)," Medium, 2023. [Online]. Available: `https://miro.medium.com/v2/resize:fit:1358/1*vkQ0hXDaQv57sALXAJquxA.jpeg`. Accessed: May 2025.

[60] Muvi, "Best Video Codec for Compression and Quality in 2019," Muvi Blog, July 2019. [Online]. Available: `https://www.muvi.com/wp-content/uploads/2019/07/Best-video-codec.png`. Accessed: May 28, 2025.

[61] Didit.me, "Deepfake: What It Is, How It's Created, and Why You Should Be Cautious," [Online]. Available: `https://didit.me/blog/deepfake-what-it-is-how-its-created-and-why-you-should-be-cautious`. Accessed: May 28, 2025.

[62] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *Proc. ICLR*, 2021.

[63] Simplified, "AI Script to Video Generator," Simplified Website, 2024. [Online]. Available: `https://lp.simplified.com/siteimages/video-editor/ai-script-to-video.png`. Accessed: May 28, 2025.

[64] S. Patil, "Understanding Convolutional Neural Networks (CNN)," Medium, 2023. [Online]. Available: `https://miro.medium.com/v2/resize:fit:720/format:webp/0*CTVHsEgahMep_Pb0.png`. Accessed: May 28, 2025.

[65] Medium, "Deep Learning Workflow Illustration," available at: `https://miro.medium.com/v2/resize:fit:4800/format:webp/1*l4xICbIIYlz1OTymWCoUTw.jpeg`, accessed June 2025.

[66] P. Paudel, "Flask and PyTorch background remover illustration," available at: https://raw.githubusercontent.com/pawanpaudel93/flask-pytorch-background-remover/master/static/flask-pytorch.jpeg, accessed June 2025.