

A project report on

IMAGE AND VIDEO FORGERY DETECTION

Submitted in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE & ENGINEERING

By

CHIPINAPI KEERTHI SADHA (21BCE9540)

GOLI REVANTH KRISHNA (21BCE7852)

POKURU HARSHINI (21BCE9512)

Under the Guidance of

PROF. ARINDAM DEY



SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

VIT-AP UNIVERSITY

AMARAVATI – 522237

ANDHRA PRADESH, INDIA

APRIL 2025

DECLARATION

We hereby declare that the thesis entitled “**IMAGE AND VIDEO FORGERY DETECTION**” submitted by us, for the award of the degree of **Bachelor of Technology in Computer Science Engineering** at **VIT** is a record of bonafide work carried out by us under the supervision of **Prof. Arindam Dey**.

We further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Amaravati

Date: 13-05-2025

Signatures of the Candidates

Ch. Keerthi Sadha

P. Harshini


G. Revanth Krishna.

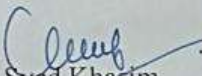
CERTIFICATE

This is to certify that the Capstone Project work titled “**IMAGE AND VIDEO FORGERY DETECTION**” that is being submitted by **CHIPINAPI KEERTHI SADHA (21BCE9540), GOLI REVANTH KRISHNA (21BCE7852)** and **POKURU HARSHINI(21BCE9512)**, is in partial fulfillment of the requirements for the award of Bachelor of Technology, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Prof. Arindam Dey
Guide

The thesis is satisfactory / unsatisfactory


Prof. Arindam Dey
Internal Examiner 1


Prof. Syed Khasim
Internal Examiner 2

Approved by

Dr. G Muneeshwari

HoD

School of Computer Science and Engineering

ABSTRACT

In the digital era, the rampant availability of fabricated media presents enormous risks to the veracity of information, with misrepresentative images and videos driving disinformation, political instability, and social conflict. As the tools used to forge become more sophisticated, spanning image joining and copy-move maneuvers to AI-produced deep fakes, there is a call for matching detection systems. Our project, Forgery Check, tackles this critical demand by creating an AI-driven web application that can identify digital forgeries in images and videos. The application employs state-of-the-art machine learning and deep learning algorithms to inspect media files and offer reliable, real-time evaluations of authenticity.

Developed through a Django-based framework, the application features a responsive, intuitive interface with robust support for uploading media and elaborate forgery inspection. To identify images, the preprocessed images training datasets is classified as original or fake using a Convolutional Neural Network (CNN) which also highlights forged areas via techniques like Error Level Analysis (ELA), detection of noise and edges. In case of video material, ResNet50 model architecture has been used in conjunction with frame-by-frame inspection in order to expose discrepancies as well as signs of deepfakes. These combined strategies provide comprehensive coverage and high accuracy on various types of media.

The project also prioritizes usability and accessibility, providing rich visual reports, confidence ratings, and support for a variety of file formats. Preliminary testing shows robust performance, with the image model reaching 87% validation accuracy and the video model successfully marking tampered frames. Beyond core functionality, the system is also built with future scalability in mind, such as supporting PDF forgery detection, mobile access, multi-file analysis, and improved user account management.

By merging technological advancement with real-world use, Forgery Check not only advances the cause of digital forensics but also equips users with the ability to fight the dissemination of false information. As the art of manipulating media continues to improve, this site is an anticipatory measure in protecting the credibility and genuineness of digital communication.

ACKNOWLEDGEMENT

It is our pleasure to express our deep sense of gratitude to **Prof. ARINDAM DEY**, Associate Professor, SCOPE, VIT-AP, for his constant guidance, continual encouragement, and understanding. More than anything, he taught us patience throughout our endeavor. Our association with him is not confined to academics alone; it has been a great opportunity for us to work with an intellectual and expert in the field of Computer Science and Engineering.

We would like to express our gratitude to Founder and Chancellor, **Dr. G. VISWANATHAN**, Vice-President, **Mr. SANKAR VISWANATHAN**, Vice-President, **Dr. SEKHAR VISWANATHAN**, Vice-President, **Dr. G. V. SELVAM**, Vice-Chancellor, **Dr. S. V. KOTA REDDY**, and **Dr. S. SUDHAKAR ILANGO**, SCOPE, for providing us with an environment conducive to learning and for their inspiration throughout the course.

In a jubilant mood, we wholeheartedly express our sincere thanks to **Dr. REEJA SR**, Head of the Department of Artificial Intelligence and Machine Learning, **Dr. AFZAL HUSSAIN SHAHID**, Year Coordinator, all the teaching staff, and members who are integral to our university for their selfless enthusiasm and timely encouragement, which enabled us to acquire the requisite knowledge to successfully complete our course.

We extend our heartfelt gratitude to our parents for their unwavering support and encouragement. It is indeed a pleasure to thank our friends who persuaded and encouraged us to take up and complete this task.

Last but not least, we express our gratitude and appreciation to all those who have helped us directly or indirectly toward the successful completion of this project.

Place: Amaravati

Date: 13-05-2025

Names of the students
Chipinapi Keerthi Sadha
Goli Revanth Krishna
Pokuru Harshini

CONTENTS

CONTENTS	VI
LIST OF FIGURES.....	IX
LIST OF TABLES.....	X
LIST OF ACRONYMS	XI

CHAPTER 1

INTRODUCTION.....	1
1.1 INTRODUCTION.....	1
1.2 OVERVIEW OF DIGITAL FORGERY MEDIA	2
1.3 IMPORTANCE OF FORGERY DETECTION.....	2
1.4 PROBLEM STATEMENT.....	3
1.5 OBJECTIVES	3
1.6 SCOPE OF THE PROJECT.....	4
1.7 ORGANIZATION OF THE REPORT.....	4

CHAPTER 2

BACKGROUND	5
2.1 INTRODUCTION.....	5
2.2 TYPES OF IMAGE AND VIDEO FORGERY	6
2.2.1 COPY-MOVE FORGERY	6
2.2.2 SPLICING.....	7
2.2.3 IMAGE RETOUCHING AND ENHANCEMENT	7
2.2.4 DEEPPAKES.....	8
2.2.5 FRAME DUPLICATION AND FRAME REMOVAL	8
2.2.6 OBJECT INSERTION AND REMOVAL.....	9
2.2.7 RE-ENCODING AND COMPRESSION ARTIFACTS	9
2.3 TECHNIQUES FOR FORGERY DETECTION.....	10
2.3.1 PIXEL-BASED ANALYSIS	10
2.3.2 BLOCK-BASED AND KEYPOINT-BASED METHODS	10
2.3.3 METADATA AND FILE STRUCTURE ANALYSIS	11
2.3.4 MACHINE LEARNING AND DEEP LEARNING APPROACHES	11

2.3.5 SENSOR AND CAMERA FINGERPRINT ANALYSIS	11
2.3.6 TEMPORAL AND MOTION ANALYSIS IN VIDEOS	12
2.3.7 HYBRID AND ENSEMBLE METHODS	12
2.3.8 HUMAN-IN-THE-LOOP AND EXPLAINABLE AI	12
2.4 REVIEW OF EXISTING TOOLS.....	13
2.4.1 IMAGE FORGERY DETECTION TOOLS	13
2.4.2 VIDEO FORGERY DETECTION TOOLS	14
2.4.3 INTEGRATED PLATFORMS AND AI-POWERED SOLUTIONS	14
2.4.4 LIMITATIONS AND AREAS FOR IMPROVEMENT	15
2.5 CHALLENGES IN MEDIA FORENSICS	16
2.6 SUMMARY OF RESEARCH GAPS	17
CHAPTER 3	
SYSTEM DESIGN AND METHODOLOGY	18
3.1 INTRODUCTION.....	18
3.2 PROPOSED SYSTEM ARCHITECTURE	19
3.3 WORKFLOW FOR IMAGE FORGERY DETECTION.....	21
3.4 WORKFLOW FOR VIDEO FORGERY DETECTION	22
3.5 DATASET COLLECTION AND PREPROCESSING.....	23
3.6 MODEL TRAINING.....	24
3.6.1 MODEL TRAINING FOR IMAGE FORGERY DETECTION	24
3.6.2 MODEL TRAINING FOR VIDEO FORGERY DETECTION	26
3.7 TOOLS AND TECHNOLOGIES USED.....	28
CHAPTER 4	
IMPLEMENTATION	30
4.1 INTRODUCTION.....	30
4.2 DJANGO WEB APPLICATION ARCHITECTURE.....	31
4.2.1 BACKEND ARCHITECTURE.....	32
4.2.2 FRONTEND INTERFACE	33
4.3 MODEL INTEGRATION INTO WEB APP	41
4.4 IMAGE ANALYSIS OUTPUT	42
4.4.1 FORGED IMAGE RESULT	42
4.4.2 AUTHENTIC IMAGE RESULT	46
4.5 VIDEO ANALYSIS OUTPUT.....	47
4.5.1 FORGED VIDEO RESULT	48
4.5.2 AUTHENTIC VIDEO RESULT	49

CHAPTER 5

TESTING AND RESULTS	50
5.1 INTRODUCTION.....	50
5.2 TEST ENVIRONMENT AND SETUP	50
5.3 FUNCTIONAL TESTING	51
5.4 MODEL ACCURACY AND PERFORMANCE METRICS.....	52

CHAPTER 6

DISCUSSION AND ANALYSIS	53
6.1 INTERPRETATION OF RESULTS	53
6.2 STRENGTHS OF THE PROPOSED SYSTEM	53
6.3 LIMITATIONS AND CHALLENGES.....	54
6.4 COMPARATIVE ANALYSIS WITH EXISTING SOLUTIONS	55

CHAPTER 7

CONCLUSION AND FUTURE WORK	56
7.1 CONCLUSION.....	56
7.2 FUTURE ENHANCEMENTS	56

REFERENCES.....	57
------------------------	-----------

APPENDICES	58
-------------------------	-----------

APPENDIX A: IMAGE FORGERY DETECTION MODEL ARCHITECTURE	58
APPENDIX B: IMAGE FORGERY DETECTION MODEL TRAINING LOG	59
APPENDIX C: VIDEO FORGERY DETECTION MODEL ARCHITECTURE.....	60
APPENDIX D: VIDEO FORGERY DETECTION MODEL TRAINING LOG	61
APPENDIX E: TECHNOLOGY STACK.....	62
APPENDIX F: DJANGO PROJECT FOLDER STRUCTURE	62
APPENDIX G: SOURCE CODE SNIPPETS.....	64

LIST OF FIGURES

Figure 1: System architecture of the web application	19
Figure 2: Workflow of the model.....	20
Figure 3: Workflow for image forgery detection	21
Figure 4: Workflow for video forgery detection.....	22
Figure 5: Training and validation loss and accuracy curves.....	25
Figure 6: Confusion matrix showing model performance on the test set.....	25
Figure 7: Confusion matrix for video forgery detection test results	27
Figure 8: Folder structure of the ‘ForgeryCheck’ in VS Code	31
Figure 9: Media forgery detection backend process	33
Figure 10: Homepage of the ForgeryCheck platform	34
Figure 11: User login page	34
Figure 12: User registration page.....	35
Figure 13: How It Works page – Video analysis workflow.....	35
Figure 14: How It Works page – Video analysis workflow.....	36
Figure 15: FAQs page	36
Figure 16: FAQs page – Expanded view.....	37
Figure 17: Contact Us page with form and contact details	38
Figure 18: User profile page	38
Figure 19: Analysis history page	39
Figure 20: Image upload page	39
Figure 21: Video upload page.....	40
Figure 22: Footer section with quick navigation links.....	40
Figure 23: Navigation bar and user interface elements.....	41
Figure 24: Forged image result page.....	42
Figure 25: Forged image – Results overview	43
Figure 26: Deep Learning Mask image.....	43
Figure 27: Error Level Analysis output.....	44
Figure 28: Edge Map output.....	44
Figure 29: Noise Analysis output.....	45
Figure 30: Copy-Move Detection output	45
Figure 31: Authentic image result page.....	46
Figure 32: Forged video result page.....	48
Figure 33: Authentic video result page	49

LIST OF TABLES

Table 1: Model Architecture for Image Forgery Detection	58
Table 2: Summary of Training Accuracy and Loss.....	59
Table 3: Model Architecture for Video Forgery Detection.....	60
Table 4: Summary of Training Accuracy and Loss.....	61
Table 5: Overview of Technologies and Libraries Utilized.....	62

LIST OF ACRONYMS

Acronym	Full Form
ELA	Error Level Analysis
AI	Artificial Intelligence
API	Application Programming Interface
ASGI	Asynchronous Server Gateway Interface
CNN	Convolutional Neural Network
CSS	Cascading Style Sheets
DB	Database
GUI	Graphical User Interface
HTML	HyperText Markup Language
JPEG	Joint Photographic Experts Group (image format)
JS	JavaScript
WSGI	Web Server Gateway Interface
ML	Machine Learning
PIL	Python Imaging Library
URL	Uniform Resource Locator
RGB	Red Green Blue (color model)
SIFT	Scale-Invariant Feature Transform
SQL	Structured Query Language
UI	User Interface

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The digital revolution has transformed the way we capture, share, and consume visual information. Images and videos are now an integral part of our daily communication, journalism, entertainment, and even legal proceedings. However, this widespread use of digital media has also introduced new challenges, particularly in the realm of authenticity. With the proliferation of sophisticated editing tools and artificial intelligence, it has become alarmingly easy to manipulate visual content in ways that are nearly undetectable to the human eye. The consequences of such manipulations are far-reaching. Forged images and videos can be used to spread misinformation, damage reputations, influence public opinion, and even serve as falsified evidence in courts of law. The rise of deepfakes AI-generated synthetic media that can convincingly mimic real people has further escalated concerns about the reliability of digital content. As a result, the demand for robust, automated systems capable of detecting forgeries in images and videos has never been greater.

This project addresses this urgent need by developing a comprehensive web-based application for image and video forgery detection. By leveraging advanced machine learning models and an intuitive user interface, the system aims to empower users, regardless of their technical background to verify the authenticity of visual media with confidence. The application not only detects manipulations but also provides detailed forensic analysis, making it a valuable tool for journalists, researchers, legal professionals, and the general public. Through this work, we hope to contribute to the broader effort of preserving truth and trust in the digital age.

In addition to its technical capabilities, Forgery Check is designed with a strong focus on user empowerment and trust. The goal isn't just to flag manipulated media, it's to help users understand how and why a piece of content might be deceptive. The platform breaks down complex AI outputs into simple, digestible insights, making it easier for users from all backgrounds to make informed decisions. Whether it's a journalist verifying a source, a student assessing an image in a research project, or an everyday user curious about a viral video, the application bridges the gap between cutting-edge forensics and real-world utility. By promoting digital literacy and fostering critical thinking, Forgery Check encourages a more discerning and responsible online community.

1.2 OVERVIEW OF DIGITAL FORGERY MEDIA

Digital media forgery refers to the deliberate manipulation or fabrication of images and videos with the intent to mislead viewers or distort reality. Over the past decade, the accessibility of powerful editing software and the rise of artificial intelligence have made it increasingly easy for individuals to alter digital content. Techniques such as copy-move, where a portion of an image is duplicated elsewhere within the same image, and splicing, which involves merging elements from different sources, are commonly used to create convincing forgeries. More recently, the emergence of deepfake technology where AI is used to generate hyper-realistic but entirely fake videos has further complicated the landscape, making it possible to fabricate events or statements that never actually occurred.

The impact of digital media forgery extends far beyond simple photo edits or harmless pranks. In the wrong hands, forged images and videos are capable of being used to convey false information, manipulate public opinion, or even serve as fabricated evidence in legal disputes. The rapid dissemination of manipulated content on social media platforms amplifies its reach and potential harm. With major ramifications for reporting, law enforcement, and the public at large, the capacity to identify and reveal digital forgeries has consequently emerged as a crucial field for research and development.

1.3 IMPORTANCE OF FORGERY DETECTION

Detecting forged media is crucial for maintaining the integrity of information in society. Automated forgery detection tools help prevent the spread of misinformation, protect individuals and organizations from reputational harm, and ensure that digital evidence used in legal contexts is trustworthy. By providing timely and accurate analysis, such systems support informed decision-making and foster greater confidence in digital communications.

Beyond these immediate benefits, forgery detection also plays a vital role in upholding ethical standards and accountability in the digital world. As manipulated images and videos become more sophisticated, the potential for misuse in areas such as journalism, politics, and social media increases significantly. Reliable detection mechanisms not only deter malicious actors but also empower users to critically evaluate the authenticity of the content they encounter. In this way, forgery detection contributes to a more transparent, informed, and responsible digital society.

1.4 PROBLEM STATEMENT

Despite the growing threat of digital forgeries, accessible and reliable detection tools remain limited, many are either too technical or narrowly focused on specific manipulations. This project bridges that gap with a user-friendly web application that integrates advanced machine learning models to detect various forgery types in both images and videos, making robust forensic analysis more accessible to everyday users. The rapid evolution of forgery techniques, including AI-generated deepfakes and subtle frame-level edits, demands adaptable detection systems. Many current tools lack the flexibility to keep pace with these developments. By prioritizing usability and extensibility, this project empowers users from diverse backgrounds to confidently verify digital content and stay ahead of emerging threats, regardless of technical expertise.

In addition, the lack of transparency and interpretability in many existing detection systems further limits their effectiveness and user trust. Users often receive binary results without any supporting evidence or explanation, making it difficult to understand or act upon the findings. This project addresses this gap by not only providing detection outcomes but also offering clear forensic insights and visual evidence, thereby enhancing user confidence and enabling informed decision-making in the face of digital forgery.

1.5 OBJECTIVES

The main objectives of this project are to:

- Develop a web-based platform for detecting forgeries in images and videos.
- Integrate state-of-the-art machine learning models for accurate analysis.
- Present results in a clear and interpretable manner, including forensic evidence and confidence scores.
- Ensure the application is accessible and easy to use for non-technical users.

In addition to these core goals, the project aspires to foster greater awareness about digital forgery and its implications. By providing transparent and detailed analysis, the system encourages users to critically assess the authenticity of media they encounter. The project also aims to lay a foundation for future enhancements, such as real-time detection, support for additional file formats, and the integration of new detection algorithms as the field advances.

1.6 SCOPE OF THE PROJECT

This project focuses on detecting digital forgeries in commonly used image and video formats such as JPEG, PNG, MP4, and AVI. Designed with accessibility in mind, the system is delivered through a web-based interface, allowing users to upload media files without needing specialized software or technical skills. Once submitted, each file is analyzed using advanced, pre-trained machine learning models tailored to detect manipulations like copy-move, splicing, and deepfakes. Results are clearly presented, accompanied by confidence scores and visual indicators that highlight suspicious areas, making the output easily understandable for non-experts. The intuitive interface ensures smooth navigation, while backend optimization allows for fast, real-time analysis without compromising accuracy.

The platform is built to serve a broad range of users—from journalists and legal professionals verifying digital evidence to individuals concerned about online misinformation. It prioritizes both precision and ease of use, closing the gap between complex forensic tools and practical, real-world needs. The system’s modular architecture allows for the integration of new detection techniques and support for additional formats in future updates. With scalability, security, and potential integration with other forensic databases in mind, the platform is not only prepared to meet current demands but is also positioned to evolve alongside the growing sophistication of digital manipulation tactics. Ultimately, the goal is to build digital resilience by enabling more people to critically assess the authenticity of the media they consume.

1.7 ORGANIZATION OF THE REPORT

The report is structured as follows:

- Chapter 2 reviews the background and literature on digital forgery detection.
- Chapter 3 details the system design and methodology.
- Chapter 4 describes the implementation of the web application.
- Chapter 5 presents testing procedures and results.
- Chapter 6 discusses findings, strengths, and limitations.
- Chapter 7 concludes the report and outlines future work.

CHAPTER 2

BACKGROUND

2.1 INTRODUCTION

The swift development of digital technology has transformed the process of producing, sharing, and consuming visual information. Images and videos are now the main tool for communication, storytelling, and recording both in our private and work life. But this digital revolution has also presented new weaknesses, mostly of visual media itself regarding authenticity and trustworthiness. The simplicity with which virtual content can be altered has bred a multitude of issues, rendering the identification of forgeries an urgent research and innovation topic.

In recent years, the proliferation of sophisticated editing tools and artificial intelligence has made it possible to alter images and videos in ways that are virtually undetectable to the naked eye. Techniques such as copy-move, splicing, and the emergence of deepfakes have blurred the line between genuine and fabricated content. These manipulations are not just technical curiosities; they have real-world implications, from spreading misinformation and influencing public opinion to undermining the credibility of digital evidence in legal proceedings.

The growing prevalence of digital forgeries has prompted researchers, technologists, and policymakers to seek effective solutions for detecting and mitigating the impact of manipulated media. The field of digital forensics has evolved rapidly, drawing on advances in computer vision, machine learning, and signal processing to develop tools and techniques capable of identifying subtle inconsistencies and artifacts introduced during the forgery process. Despite these advancements, the dynamic nature of forgery techniques continues to pose significant challenges, necessitating ongoing research and adaptation.

This chapter provides a comprehensive overview of the background and current state of digital forgery detection. It explores the various types of image and video manipulations, reviews the evolution of detection techniques, and examines the strengths and limitations of existing tools and methodologies. By understanding the landscape of digital forgery and the efforts made to combat it, we can better appreciate the significance of developing robust, accessible, and adaptable detection systems, Such as the one presented in this project.

2.2 TYPES OF IMAGE AND VIDEO FORGERY

The digital age has brought about an explosion in the creation and sharing of visual content. While this has enabled new forms of communication and creativity, it has also opened the door to increasingly sophisticated methods of manipulation. Image and video forgeries are now more prevalent and convincing than ever, posing significant challenges for individuals, organizations, and society as a whole. Understanding the various types of forgeries is essential for developing effective detection strategies and appreciating the risks associated with digital media. With the widespread availability of editing tools and artificial intelligence, even non-experts can now produce convincing forgeries with minimal effort.

Digital forgeries can range from simple edits, such as removing blemishes or adjusting lighting, to complex manipulations that completely alter the context or meaning of an image or video. Some techniques are designed to deceive viewers subtly, while others aim to fabricate entirely new events or narratives. The following sections describe the most common and impactful types of image and video forgery encountered in today's digital landscape. As these techniques continue to evolve, the line between authentic and manipulated content becomes increasingly blurred, making it more important than ever to stay informed about the latest forgery trends. By recognizing the signs and understanding the methods behind these manipulations, we can better protect ourselves and the integrity of the information we consume.

2.2.1 COPY-MOVE FORGERY

Copy-move forgery is perhaps the most common and secretly simple way of computer-generated deception. In this technique, a segment of an image or a video frame is copied and pasted elsewhere within the same file. The primary motivation behind this is often to conceal or duplicate certain elements—such as hiding a person in a crowd, removing sensitive information, or artificially increasing the number of objects in a scene. Because the copied region originates from the same image, it naturally matches the surrounding area in terms of color, texture, and noise, making it extremely difficult to spot with the naked eye. This subtlety is what makes copy-move forgery so effective and widespread. Detecting such manipulations requires advanced computational methods, such as block-based matching, keypoint detection, and pattern recognition algorithms, which can identify duplicated regions even when they have been rotated, scaled, or slightly altered to evade detection.

2.2.2 SPLICING

Splicing, also known as image or video composition, involves merging content from two or more separate sources to create a single, altered piece of media. Unlike copy-move forgery, where the manipulation is confined within the same file, splicing introduces entirely new elements that were not present in the original scene. This technique is often used to fabricate events, place individuals in different contexts, or combine objects in ways that never actually occurred. The process typically requires careful adjustment of lighting, shadows, and perspective to make the manipulation less obvious and more convincing. However, even the most skilled forgers may leave behind subtle inconsistencies, such as mismatched edges, unnatural transitions, or discrepancies in image metadata. Detecting splicing is a complex task that often involves analyzing lighting direction, edge artifacts, and other forensic clues to reveal the tampering. As splicing becomes more sophisticated, the need for robust detection tools becomes increasingly critical. In some high-profile cases, spliced images have been used to mislead audiences during political campaigns or spread false narratives on social media. With the ease of access to editing software, even non-experts can now produce convincing forgeries, making public awareness and detection systems more vital than ever.

2.2.3 IMAGE RETOUCHING AND ENHANCEMENT

Image retouching and enhancement are techniques that, while often used for legitimate purposes like improving visual appeal, can also serve deceptive goals. Retouching may involve modifying contrast, brightness, or color balance, as well as removing blemishes, smoothing skin, or altering facial features to misrepresent the appearance of individuals or scenes. In some cases, retouching is used to subtly change the emotional tone or credibility of an image, such as making a person appear happier, younger, or more trustworthy. These changes can be so subtle that they are nearly impossible to detect without specialized tools. But when applied with malicious intent, retouching distorts reality and deceives viewers. Detection methods for retouching often rely on analyzing inconsistencies in pixel distribution, examining noise patterns, or comparing the edited image with its original version if available. As digital editing tools become more accessible and powerful, the line between enhancement and deception continues to blur, making detection all the more important. This is particularly concerning in fields like e-commerce, online dating, and news media, where trust is essential and manipulated visuals can lead to false perceptions.

2.2.4 DEEPFAKES

Deepfakes represent a groundbreaking and deeply concerning advancement in digital forgery. Leveraging the power of artificial intelligence, particularly deep learning models like Generative Adversarial Networks (GANs), deepfakes can generate highly realistic synthetic media. The most common application involves swapping faces in videos or images, animating still photos, or generating entirely fabricated speech and expressions. Deepfakes can convincingly mimic real people, making them difficult to distinguish from authentic content. The potential for misuse is substantial, as deepfakes can be used to spread misinformation, impersonate individuals, or create fake evidence for malicious purposes. Detecting deepfakes is a rapidly evolving field that requires advanced analytical tools capable of identifying subtle artifacts, such as unnatural blinking, inconsistencies in facial geometry, or anomalies in audio-visual synchronization. As deepfake technology continues to improve, the arms race between forgers and forensic analysts is expected to intensify, underscoring the need for ongoing research and innovation in this field. In sensitive areas such as politics, journalism, and national security, the consequences of deepfake misuse can be particularly damaging. Therefore, building awareness and integrating detection capabilities into mainstream platforms is crucial to curbing the harmful impact of synthetic media.

2.2.5 FRAME DUPLICATION AND FRAME REMOVAL

Frame duplication and frame removal are techniques commonly used in video forgeries to manipulate the flow of action and alter the narrative. By duplicating frames, a forger can fabricate repetitive behavior or extend certain actions, while removing frames can hide key moments or events, effectively rewriting the story captured by the video. These manipulations can be subtle yet have a significant impact on the interpretation of an event, especially in contexts like surveillance footage or evidence in legal cases. Detecting such alterations requires careful analysis of temporal consistency, motion flow, and synchronization within the video sequence. Advanced algorithms can identify irregularities in frame transitions, abrupt changes in motion, or inconsistencies in audio-visual alignment. As video editing tools become more sophisticated, the ability to detect frame-level manipulations is becoming increasingly vital for ensuring the integrity of video evidence. Moreover, even small manipulations can change the perceived intent or outcome of an incident, making accurate detection essential in high-stakes scenarios. A robust detection system not only uncovers tampering but also helps preserve trust in digital documentation.

2.2.6 OBJECT INSERTION AND REMOVAL

Object insertion and removal are powerful techniques that can dramatically alter the meaning or narrative of an image or video. In object insertion, new elements are seamlessly added to a scene, often through meticulous blending and shadow matching to ensure they appear natural. Conversely, object removal involves erasing unwanted elements and filling in the gaps with content that matches the surrounding area. In videos, these processes are even more complex, requiring frame-by-frame editing to maintain consistency and realism. The motivations behind object insertion and removal can range from innocent photo enhancements to deliberate attempts at deception or misinformation. Detecting such manipulations involves looking for unnatural edges, perspective mismatches, or residual signals that do not align with the original scene's characteristics. As these techniques become more accessible, the risk of their misuse grows, highlighting the importance of robust detection methods. Even small changes like removing a single person or adding an object, can completely shift the interpretation of a scene. This makes it critical to educate users and develop forensic tools that can detect such manipulations before they influence public perception.

2.2.7 RE-ENCODING AND COMPRESSION ARTIFACTS

Re-encoding and compression artifacts are often unintended byproducts of digital forgery. When an image or video is edited and then saved again, especially in a lossy format like JPEG or MP4, the process can introduce visible inconsistencies not present in the original. These artifacts may manifest as changes in quality, blockiness, or unusual patterns in certain regions of the media. Sometimes, forgers attempt to mask their edits by re-encoding the entire file, but this can actually make the manipulation more detectable. Techniques such as Error Level Analysis (ELA) are used to highlight variations in compression levels, helping to pinpoint areas that may have been tampered with. While compression artifacts alone do not prove forgery, they can serve as important clues in a broader forensic investigation, guiding analysts to regions that warrant closer scrutiny. Additionally, analyzing these artifacts provides insight into the editing history and can reveal how many times a file has been altered or saved. This makes compression analysis a key step in validating the authenticity of digital content, especially in legal or journalistic contexts. Furthermore, repeated compression can degrade image quality unevenly, making certain tampered regions stand out under forensic scrutiny.

2.3 TECHNIQUES FOR FORGERY DETECTION

Due to the growing complexity of manipulation techniques and the pervasiveness of digital misinformation, the identification of image and video fakes has emerged as a crucial area of research and development. Over the years, a variety of methods have been developed to identify tampered media, ranging from traditional signal processing approaches to advanced machine learning and deep learning models. Each technique offers unique strengths and faces distinct challenges, often necessitating a combination of methods for robust and reliable detection.

2.3.1 PIXEL-BASED ANALYSIS

Pixel-based analysis is one of the foundational approaches in digital forensics. This technique involves examining the statistical properties of pixels within an image or video frame to identify anomalies that may indicate tampering. Methods such as Error Level Analysis (ELA) and noise inconsistency detection fall under this category. ELA, for example, works by analyzing the compression artifacts introduced when an image is saved in a lossy format like JPEG. Regions that have been altered often exhibit different error levels compared to untouched areas, making them stand out under analysis. Similarly, noise analysis detects inconsistencies in the noise pattern, as manipulations often disrupt the natural distribution of noise across an image. While pixel-based methods are effective for detecting certain types of forgeries, they can be sensitive to post-processing operations like resizing or re-compression, which may mask or mimic signs of tampering.

2.3.2 BLOCK-BASED AND KEYPOINT-BASED METHODS

Block-based techniques divide an image into overlapping or non-overlapping blocks and analyze their similarities to detect duplicated regions, a common sign of copy-move forgery. By comparing blocks using statistical measures or feature descriptors, these methods can identify areas that have been copied and pasted within the same image. Keypoint-based approaches, such as those utilizing SIFT or SURF, detect distinctive points in an image and match them to uncover duplicated or spliced regions. These methods are robust to geometric transformations like rotation and scaling, making them suitable for detecting more sophisticated manipulations. However, their effectiveness can be limited by the quality of the image and the presence of repetitive patterns.

2.3.3 METADATA AND FILE STRUCTURE ANALYSIS

Beyond the visual content, digital images and videos contain metadata—information about how, when, and with what device the file was created or modified. Metadata analysis involves examining this embedded information for inconsistencies that may suggest tampering. For instance, discrepancies in timestamps, camera model details, or editing software tags can raise red flags about the authenticity of a file. Additionally, analyzing the file structure and compression artifacts can reveal traces of manipulation, such as double JPEG compression or unusual encoding patterns. While metadata analysis can provide valuable clues, it is not foolproof, as metadata can be easily edited or stripped from files.

2.3.4 MACHINE LEARNING AND DEEP LEARNING APPROACHES

With the advent of artificial intelligence, machine learning and deep learning techniques have revolutionized forgery detection. These methods involve training models on large datasets of authentic and manipulated media to learn distinguishing features that may be imperceptible to humans. Convolutional Neural Networks (CNNs) are widely used for image forgery detection, capable of identifying subtle artifacts and inconsistencies introduced during tampering. In video forensics, Recurrent Neural Networks (RNNs) and 3D CNNs are employed to analyze temporal dynamics and spatial features across frames. Deep learning models can also be trained to detect specific types of forgeries, such as deepfakes, by learning the unique signatures left by generative algorithms. While these approaches offer high accuracy and adaptability, they require substantial computational resources and large, diverse training datasets to generalize effectively.

2.3.5 SENSOR AND CAMERA FINGERPRINT ANALYSIS

Every digital camera sensor leaves a unique pattern of noise, known as a Photo-Response Non-Uniformity (PRNU), on the images it captures. Sensor fingerprint analysis leverages this property to verify the source of an image or detect inconsistencies that may indicate tampering. By comparing the PRNU pattern of a suspicious image with that of a reference camera, forensic analysts can determine whether the image has been altered or originated from a different device. This technique is particularly useful for verifying the authenticity of images in legal and investigative contexts. However, it requires access to reference images and can be affected by heavy post-processing or compression.

2.3.6 TEMPORAL AND MOTION ANALYSIS IN VIDEOS

Video forgery detection often involves analyzing temporal and motion-related features to identify inconsistencies across frames. Techniques such as frame duplication detection, frame removal analysis, and motion flow examination are used to uncover manipulations that disrupt the natural progression of events in a video. For example, abrupt changes in motion vectors or irregularities in frame transitions can signal tampering.

2.3.7 HYBRID AND ENSEMBLE METHODS

Given the complexity and diversity of forgery techniques, no single detection method is universally effective. As a result, hybrid and ensemble approaches have gained popularity, combining multiple techniques to improve detection accuracy and robustness. For instance, a system may integrate pixel-based analysis, metadata examination, and deep learning predictions to cross-validate results and reduce false positives. Ensemble methods can leverage the strengths of different algorithms, providing a more comprehensive assessment of media authenticity.

2.3.8 HUMAN-IN-THE-LOOP AND EXPLAINABLE AI

Despite advances in automation, human expertise remains invaluable in media forensics. Human-in-the-loop systems combine automated detection with expert review, allowing analysts to interpret results, investigate ambiguous cases, and provide context-aware judgments. Additionally, the rise of explainable AI aims to make deep learning models more transparent, enabling users to understand the rationale behind detection decisions. This is particularly important in high-stakes scenarios, such as legal proceedings or journalistic investigations, where trust and accountability are paramount.

In summary, forgery detection is a rapidly evolving field that draws on a wide array of techniques, from traditional signal processing to cutting-edge artificial intelligence. The ongoing arms race between forgers and forensic analysts drives continuous innovation, with new methods emerging to address the challenges posed by increasingly sophisticated manipulations. As digital media continues to shape our world, the importance of reliable and adaptable forgery detection techniques cannot be overstated.

2.4 REVIEW OF EXISTING TOOLS

The rapid advancement of digital manipulation techniques has spurred the development of a wide array of tools and applications designed to detect image and video forgeries. These solutions range from open-source academic projects to commercial platforms, each offering unique features, detection capabilities, and user experiences. Understanding the landscape of existing tools is essential for identifying their strengths, limitations, and the gaps that still exist in the field of media forensics.

While many of these tools have made significant strides in identifying tampered content, their effectiveness often depends on the type of forgery, the quality of the media, and the level of user expertise required. Some tools are highly technical, offering detailed forensic analyses suited for researchers or professionals, while others focus on simplicity and accessibility for everyday users. Despite these advances, challenges remain. Such as handling new formats, detecting subtle edits, or distinguishing between benign enhancements and malicious alterations.

2.4.1 IMAGE FORGERY DETECTION TOOLS

A number of specialized tools have emerged to address the challenge of image forgery detection. Open-source solutions like FotoForensics provide users with access to techniques such as Error Level Analysis (ELA), which highlights areas of an image that may have been altered by comparing compression artifacts. Similarly, tools like JPEGsnoop allow for in-depth analysis of JPEG file structures and metadata, helping users uncover signs of tampering or double compression. Academic projects often focus on advanced methods, such as copy-move detection using block matching or keypoint analysis, and may provide code libraries or research prototypes for further experimentation.

Commercial platforms, on the other hand, tend to offer more user-friendly interfaces and integrate multiple forensic techniques. For example, Izitru and Serelay provide cloud-based verification services that assess image authenticity using a combination of metadata checks, noise analysis, and machine learning models. These platforms are designed for journalists, legal professionals, and organizations that require quick and reliable verification of digital images. However, commercial tools may come with usage fees or limitations on the types of files they can process.

2.4.2 VIDEO FORGERY DETECTION TOOLS

Detecting forgeries in videos presents additional challenges due to the temporal dimension and the sheer volume of data involved. Some tools, like Amber Video and Microsoft Video Authenticator, are specifically designed to analyze videos for signs of deepfakes and frame-level manipulations. These applications often leverage deep learning models trained on large datasets of authentic and manipulated videos, enabling them to spot subtle inconsistencies in facial movements, lighting, or audio-visual synchronization.

Academic research has also produced a variety of video forensic tools, many of which are available as open-source projects or research prototypes. These tools may focus on detecting frame duplication, splicing, or re-encoding artifacts, and often require technical expertise to operate effectively. While some solutions provide visualizations of detected anomalies, others output detailed reports highlighting suspicious segments within the video.

2.4.3 INTEGRATED PLATFORMS AND AI-POWERED SOLUTIONS

With the growing complexity of digital forgeries, integrated platforms that combine multiple detection techniques have become increasingly popular. AI-powered solutions, such as ForgeryCheck, utilize deep learning models alongside traditional forensic methods to deliver comprehensive analysis of both images and videos. These platforms often provide instant feedback, confidence scores, and visual evidence of detected manipulations, making them accessible to a broad range of users, from casual consumers to investigative professionals. By leveraging the strengths of both machine learning and classical analysis, these systems can adapt to new types of forgeries as they emerge, ensuring that detection capabilities remain robust. Moreover, the user-friendly design of such platforms means that even those without technical expertise can confidently assess the authenticity of digital media.

The integration of cloud computing and web-based interfaces has further democratized access to forensic tools. Users can now upload suspicious files through simple web portals and receive detailed analysis without the need for specialized hardware or software installations. This ease of use is crucial for empowering individuals and organizations to verify digital content in real time, especially in contexts where misinformation can spread rapidly.

2.4.4 LIMITATIONS AND AREAS FOR IMPROVEMENT

Despite the remarkable progress in the field of digital forgery detection, current tools and applications still face a range of significant limitations. One of the most persistent challenges is the difficulty in analyzing heavily compressed or low-resolution media. When images or videos are compressed, essential forensic traces—such as subtle noise patterns, compression artifacts, or pixel-level inconsistencies—can be lost or distorted. This makes it much harder for detection algorithms to distinguish between genuine content and manipulated regions. In real-world scenarios, where media is often shared and re-shared across various platforms, files are frequently resized or compressed multiple times, further complicating the detection process. As a result, even advanced systems may produce false negatives, missing manipulations that would otherwise be detectable in higher-quality originals. This limitation highlights the need for more resilient algorithms that can operate effectively across a wide range of media qualities and formats, ensuring reliable detection regardless of how the content has been processed.

Another major area of concern is the ongoing arms race between forgery creators and detection technologies, particularly in the realm of deepfakes. Generative models, such as those based on GANs (Generative Adversarial Networks), are evolving at a rapid pace, producing synthetic media that is increasingly difficult to distinguish from authentic content. Deepfake detection tools must constantly adapt to new techniques and artifacts, but the speed at which generative models improve often outpaces the development of detection methods. This dynamic creates a moving target for researchers and developers, who must not only keep up with the latest advances in forgery creation but also anticipate future trends. Moreover, the sophistication of deepfakes means that even trained experts can be fooled, raising the stakes for automated detection systems. The challenge is further compounded by the lack of large, diverse, and up-to-date datasets for training and evaluating detection models, which can limit their generalizability and robustness in real-world applications.

In summary, the current landscape of forgery detection tools is diverse and rapidly evolving. While many effective solutions exist for both images and videos, there is still a need for more robust, user-friendly, and adaptable platforms that can keep pace with emerging manipulation techniques. Ongoing research and collaboration between academia, industry, and the public sector will be essential for closing these gaps and ensuring the authenticity of digital media in the years to come.

2.5 CHALLENGES IN MEDIA FORENSICS

Media forensics faces a rapidly evolving landscape, with new challenges emerging as technology advances and digital content becomes more pervasive. One of the most significant hurdles is the sheer sophistication and accessibility of manipulation tools. Today, even individuals with limited technical expertise can create convincing forgeries using freely available software or AI-powered applications. This democratization of forgery creation means that the volume and complexity of manipulated media are growing at an unprecedented rate, making detection increasingly difficult.

Another major challenge is the diversity of manipulation techniques. From subtle retouching and copy-move forgeries to advanced deepfakes and synthetic media, each type of forgery leaves behind different traces and requires specialized detection methods. No single approach is universally effective, and forensics experts must constantly adapt their strategies to keep pace with new threats. Furthermore, the widespread use of compression, resizing, and social media sharing often degrades forensic evidence, obscuring the subtle artifacts that detection algorithms rely on.

The lack of standardized datasets and benchmarks also hampers progress in the field. Researchers and developers often work with different data sources and evaluation criteria, making it difficult to compare results or measure real-world effectiveness. This fragmentation slows the development of robust, generalizable solutions and can lead to inconsistent outcomes in practical applications.

Privacy and ethical considerations add another layer of complexity. Forensic analysis often involves handling sensitive or personal data, raising concerns about user consent, data security, and potential misuse of detection technologies. Striking a balance between effective forgery detection and the protection of individual rights is an ongoing challenge that requires careful policy and technical safeguards.

Finally, the speed at which misinformation can spread online means that timely detection is critical. Automated systems must not only be accurate but also fast and scalable, capable of analyzing large volumes of content in real time. Achieving this level of performance while maintaining high reliability remains a significant technical and operational challenge for the field.

2.6 SUMMARY OF RESEARCH GAPS

Despite notable advancements in media forensics, several research gaps persist that hinder the development of truly robust and universal detection systems. One of the most pressing gaps is the need for detection methods that can withstand heavy post-processing, such as compression, scaling, and format conversion, which are common in real-world media sharing. Many current algorithms lose effectiveness when forensic traces are degraded, highlighting the need for more resilient approaches.

Another gap lies in the detection of emerging and hybrid manipulation techniques. As forgers combine multiple methods—such as blending deepfakes with traditional splicing or using AI to erase forensic footprints—existing tools may fail to recognize these complex forgeries. Research into multi-modal and ensemble detection strategies is still in its early stages and requires further exploration.

There is also a shortage of large, diverse, and up-to-date datasets that reflect the full spectrum of real-world manipulations. Most available datasets are limited in scope or quickly become outdated as new forgery techniques emerge. This lack of representative data restricts the training and evaluation of advanced detection models, particularly those based on deep learning. Interpretable and user-friendly detection systems are another area in need of development. Many current solutions provide technical outputs that are difficult for non-experts to understand or act upon. Bridging the gap between technical accuracy and practical usability is essential for empowering a broader range of users, from journalists to everyday consumers.

Lastly, ethical and privacy considerations are often underexplored in research. As detection technologies become more powerful, ensuring that they are used responsibly and with respect for individual rights is crucial. Future research should address not only technical challenges but also the societal implications of widespread media forensic analysis.

CHAPTER 3

SYSTEM DESIGN AND METHODOLOGY

3.1 INTRODUCTION

In an era where digital content is created, shared, and consumed at an unprecedented scale, the authenticity of images and videos has become a critical concern. The ease with which media can be manipulated, whether through simple editing tools or advanced artificial intelligence, poses significant challenges for individuals, organizations, and society as a whole. As misinformation and digital deception become more sophisticated, there is an urgent need for reliable systems that can detect and flag forged content before it spreads.

This chapter presents the design and methodology behind the proposed forgery detection system, which aims to address these challenges by leveraging a combination of traditional forensic techniques and state-of-the-art machine learning models. The system is built to analyze both images and videos, providing users with clear, actionable insights into the authenticity of their digital media. By integrating multiple detection methods and offering a user-friendly interface, the solution is intended to be accessible to a wide range of users.

The following sections will outline the overall architecture of the system, detailing how each component interacts to deliver robust and accurate forgery detection. The workflows for both image and video analysis will be described step by step, highlighting the processes involved in data acquisition, preprocessing, model inference, and result interpretation. Special attention is given to the strategies employed for dataset collection and preparation, as the quality and diversity of training data are crucial for building effective detection models.

Furthermore, this chapter will discuss the methodologies adopted for training separate models for images and videos, recognizing that each media type presents unique challenges and requires tailored approaches. The selection of tools and technologies that underpin the system will also be covered, providing insight into the practical considerations and innovations that drive the solution. By the end of this chapter, readers will gain a comprehensive understanding of the system's design philosophy, the rationale behind key methodological choices, and the steps taken to ensure that the proposed solution is both effective and adaptable in the face of evolving forgery techniques.

3.2 PROPOSED SYSTEM ARCHITECTURE

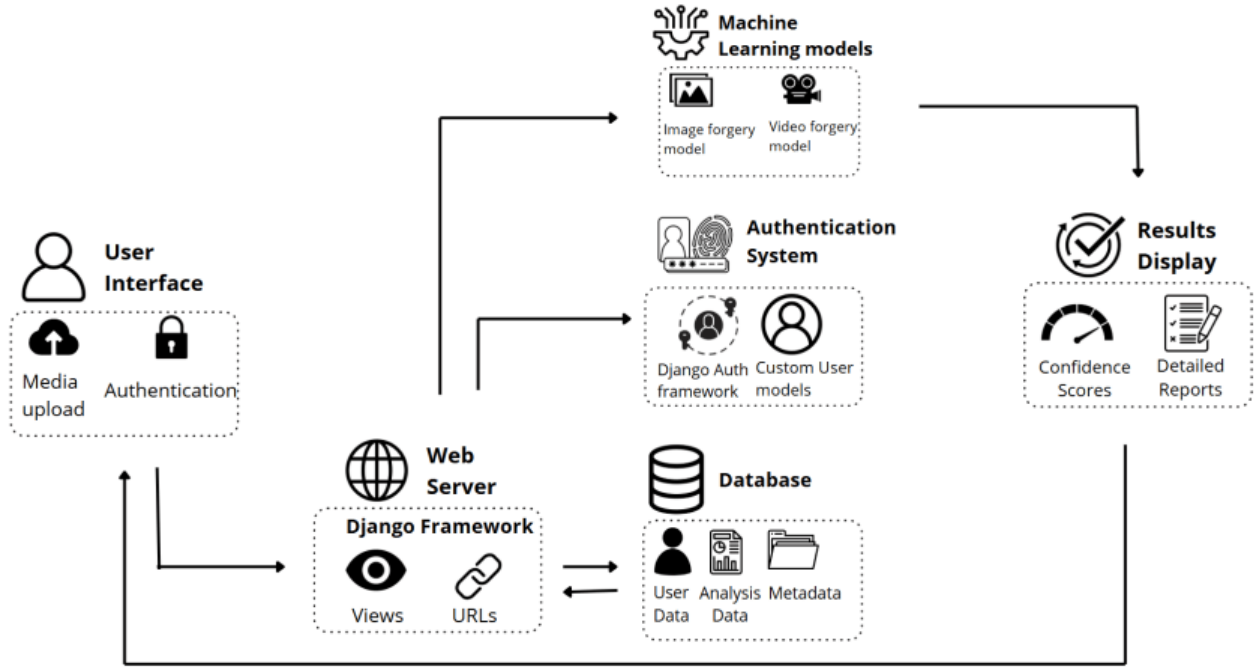


Figure 1: System architecture of the web application

The proposed system architecture is designed to make image and video forgery detection both accessible and reliable for users. At the front end, users interact with a simple web interface to upload media and access their results. The backend, built on the Django framework, manages all requests, handles authentication, and coordinates the analysis process. Uploaded files are securely stored in a database, along with user and analysis data. This setup ensures that users can quickly receive feedback on their submissions without needing technical expertise. The architecture is also flexible, allowing for future enhancements and integration of additional detection features as technology advances.

The core of the system is the forgery detection module, which uses advanced machine learning models to analyze images and videos for signs of manipulation. Once the analysis is complete, results, including confidence scores and detailed reports, are displayed back to the user through the interface. Each component works together seamlessly, ensuring that the process is secure, efficient, and user-friendly. The modular design also allows for easy updates and integration of new detection techniques as the field evolves. This ensures that users can always rely on the platform to keep pace with emerging threats and maintain the integrity of their digital content.

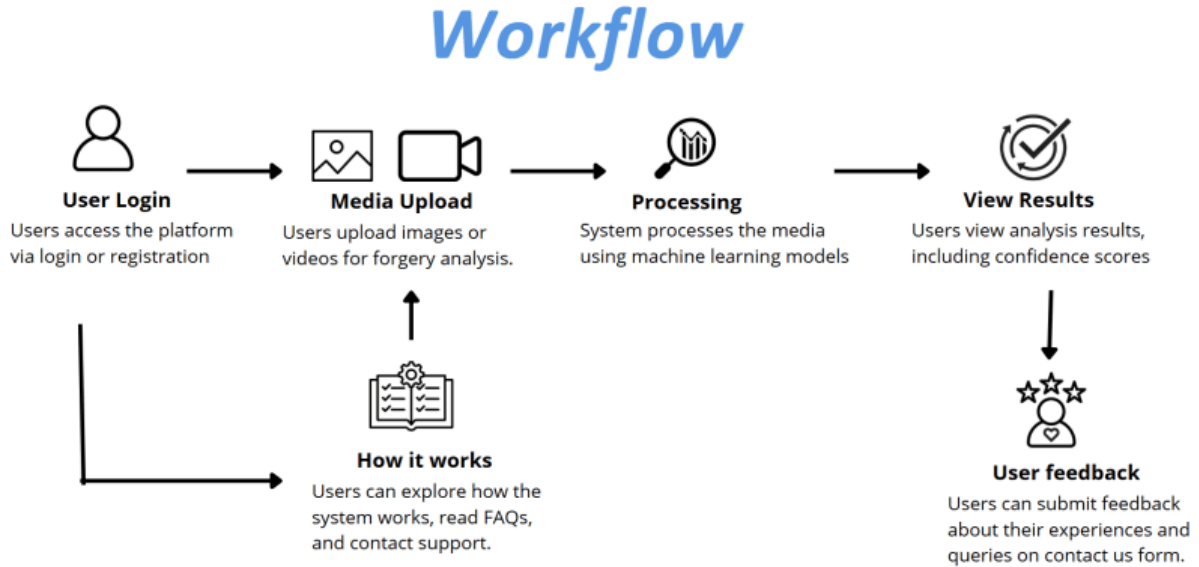


Figure 2: Workflow of the model

The workflow for the forgery detection platform is designed to be straightforward and user-friendly. Users begin by logging into the system, either through registration or their existing credentials. Once authenticated, they can upload images or videos that they want to check for authenticity. The uploaded media is then processed by the system, which uses advanced machine learning models to analyze the content for any signs of tampering or forgery.

After processing, users are presented with clear and detailed results, including confidence scores that indicate the likelihood of manipulation. The platform also encourages users to provide feedback about their experience or any questions they may have, helping to improve the service further. Additionally, users can access a dedicated section to learn more about how the system works, explore frequently asked questions, and reach out for support if needed. This workflow ensures that every step, from login to feedback, is smooth and transparent for all users.

The platform also encourages users to provide feedback about their experience or any questions they may have, helping to improve the service further. Additionally, users can access a dedicated section to learn more about how the system works, explore frequently asked questions, and reach out for support if needed. This workflow ensures that every step, from login to feedback, is smooth and transparent for all users, fostering a sense of trust and ease throughout their experience.

3.3 WORKFLOW FOR IMAGE FORGERY DETECTION

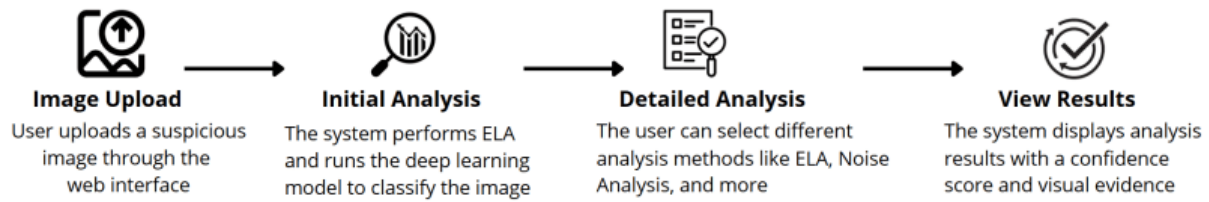


Figure 3: Workflow for image forgery detection

The workflow for image forgery detection in this platform is carefully structured to guide users through every step, ensuring both ease of use and technical robustness. The process starts with the Image Upload stage, where users submit a suspicious image through a straightforward web interface. This upload mechanism is designed to handle various image formats and provides immediate feedback, confirming that the file has been received and is ready for analysis. By making the upload process seamless, the platform encourages users to verify their digital content without any technical barriers.

Once the image is uploaded, the system initiates the Initial Analysis phase. Here, the platform automatically applies Error Level Analysis (ELA), a technique that highlights areas of potential manipulation by comparing the original image to a recompressed version. Simultaneously, a deep learning model is run to classify the image as either authentic or forged, providing a quick and data-driven first impression.

Following the initial assessment, users are given the opportunity to conduct a Detailed Analysis. The platform offers a suite of advanced forensic tools, such as noise analysis and other specialized techniques, allowing users to dive deeper into the image's integrity. For example, noise analysis examines inconsistencies in the image's noise patterns, which can be a telltale sign of tampering. Users can select which methods to apply, tailoring the analysis to their specific needs and concerns. This flexibility empowers users to gain a comprehensive understanding of the image's authenticity. Finally, in the View Results stage, the system presents the findings in a clear and accessible format. Users receive a detailed report that includes a confidence score, visual evidence such as highlighted regions of potential forgery, and a summary of the analysis performed. The results are designed to be easily interpretable, even for those without a technical background, ensuring that everyone can make informed decisions based on the evidence provided.

3.4 WORKFLOW FOR VIDEO FORGERY DETECTION

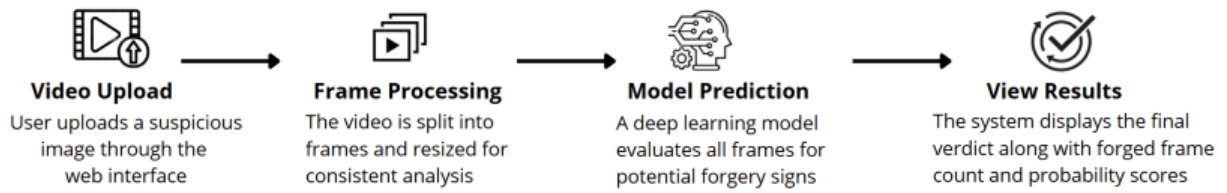


Figure 4: Workflow for video forgery detection

The workflow for video forgery detection is designed to address the complexities of video analysis while remaining user-friendly. It starts with the Video Upload step, where users can easily submit a suspicious video file through the web interface. Once the video is uploaded, the system confirms receipt and prepares the file for further processing.

During the Frame Processing stage, the system automatically splits the uploaded video into individual frames. Each frame is resized and standardized to ensure that the analysis is consistent and reliable across the entire video. This approach allows the system to examine every moment of the footage, increasing the chances of detecting even subtle manipulations that might be missed in a more superficial scan. By breaking the video down into frames, the platform ensures a thorough and granular analysis. The next step, Model Prediction, leverages advanced deep learning models to evaluate each frame for signs of forgery. The model scans for anomalies, inconsistencies, or patterns that suggest digital tampering, such as frame splicing, deepfake insertions, or other forms of manipulation. The results from all frames are then aggregated to provide a holistic assessment of the video's authenticity. In the View Results phase, users receive a comprehensive report that includes the final verdict, the number of frames flagged as forged, and detailed probability scores. The results are presented in a clear, accessible format, empowering users to make informed decisions about the integrity of their video content.

Additionally, the platform is designed to support transparency and user engagement throughout the process. Users can review visual evidence for flagged frames, helping them understand exactly where and why the system detected potential forgery. The workflow also encourages users to provide feedback or seek support if they have questions about their results. This commitment to clarity and user support ensures that the video forgery detection process is not only technically robust but also approachable and trustworthy for everyone.

3.5 DATASET COLLECTION AND PREPROCESSING

A comprehensive and well-prepared dataset is the cornerstone of any successful forgery detection system. For this project, both image and video datasets were meticulously curated to ensure a balanced representation of authentic and manipulated samples. The collection process involved gathering publicly available datasets as well as generating additional forged examples using various tampering techniques. This approach ensured that the models would be exposed to a wide range of real-world forgery scenarios, including copy-move, splicing, and deepfake manipulations.

For images, the preprocessing pipeline began with organizing the data into clear categories, authentic and forged. Each image was resized to a standard dimension to maintain consistency during model training and evaluation. Advanced preprocessing techniques, such as Error Level Analysis (ELA), were applied to highlight subtle artifacts introduced during manipulation. These processed images were then normalized and converted into arrays suitable for input into deep learning models. Data augmentation strategies, such as rotation, flipping, and color adjustments, were also employed to increase the diversity of the training set and improve model robustness.

The video preprocessing workflow was designed to handle the unique challenges of temporal data. Each video was decomposed into individual frames, with every frame resized and standardized to ensure uniformity across the dataset. The frame extraction process was carefully monitored to maintain the correct sequence and avoid data loss. Once extracted, frames were labeled according to their source, authentic or forged, and further normalized for model compatibility. The dataset was then split into training and testing subsets, ensuring that the evaluation metrics would accurately reflect real-world performance.

Throughout the preprocessing phase, visualizations and statistical analyses were used to verify the quality and balance of the dataset. Sample images and frames were inspected to confirm the effectiveness of augmentation and preprocessing steps. Distribution plots were generated to ensure an even split between authentic and forged samples, minimizing the risk of model bias. This rigorous approach to dataset collection and preprocessing laid a strong foundation for the subsequent stages of model training and evaluation, ultimately contributing to the reliability and accuracy of the forgery detection platform. In addition, correlation matrices and class-wise summary statistics were examined to detect potential anomalies or imbalances, ensuring that the dataset represented diverse patterns across both classes.

3.6 MODEL TRAINING

The model training phase is a critical component of the forgery detection system, as it determines the system's ability to accurately distinguish between authentic and manipulated content. To address the unique characteristics of images and videos, separate training pipelines were established for each modality, ensuring that the models are optimized for their respective tasks.

3.6.1 MODEL TRAINING FOR IMAGE FORGERY DETECTION

The image forgery detection training process began with assembling a balanced dataset of authentic and tampered images from established sources. All images were resized to 128x128 pixels for consistency, and Error Level Analysis (ELA) was applied to highlight subtle manipulation artifacts, aiding feature learning. The dataset was then shuffled and split into training, validation, and test sets to ensure fair performance evaluation and generalization. To enhance model robustness, data augmentation techniques such as random rotations, shifts, and flips were used to increase input variability.

MODEL ARCHITECTURE

A custom convolutional neural network (CNN) was designed for the forgery detection task. The architecture consisted of multiple convolutional layers with increasing filter sizes, interleaved with max pooling layers to progressively extract hierarchical features from the input images. Dropout layers were included to reduce overfitting by randomly deactivating a fraction of neurons during training. The network concluded with dense layers and a sigmoid activation function, outputting a probability score indicating whether an image was authentic or forged.

TRAINING PROCESS

Because the classification task was binary, binary cross-entropy was chosen as the loss function, and the model was assembled using the Adam optimizer with a carefully selected learning rate. Training was conducted over 50 epochs with a batch size of 19, and early stopping was implemented to halt training thus preventing overfitting. Throughout training, performance metrics such as accuracy and loss were monitored on both training and validation sets to ensure optimal model convergence. Additionally, the training process was closely observed to ensure the model not only learned effectively but also maintained generalization, striking a balance between learning from the data and avoiding memorization.

During training, both the loss and accuracy for the training and validation sets were monitored and plotted. These plots show that as training progressed, the loss decreased and accuracy increased for both training and validation sets, indicating effective learning and good generalization.

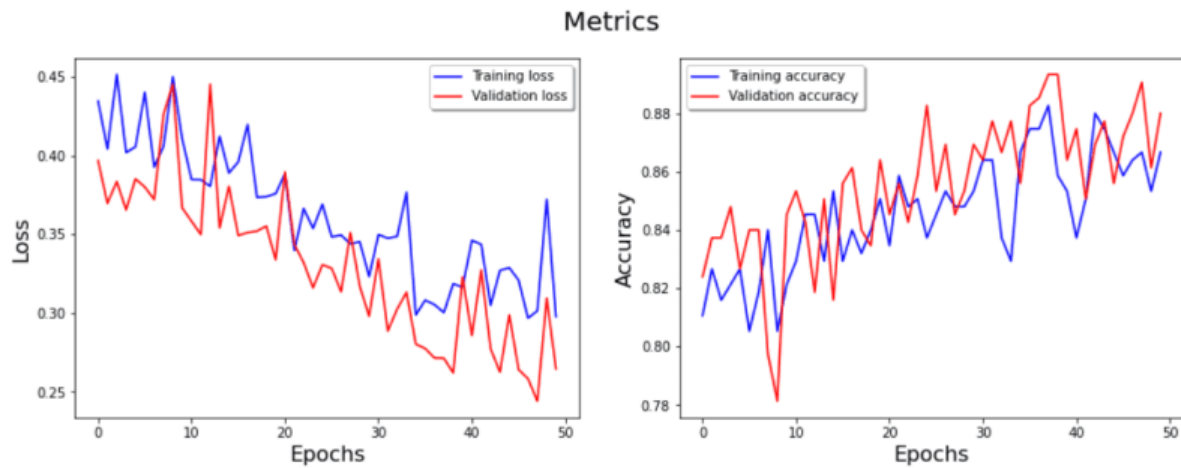


Figure 5: Training and validation loss and accuracy curves

MODEL EVALUATION

After training, the model's performance was rigorously evaluated on the test set. Predictions were compared against ground truth labels to compute key metrics such as accuracy, precision, recall, and F1-score. The confusion matrix reveals the number of true positives, true negatives, false positives, and false negatives, offering insight into the model's strengths and areas for improvement.

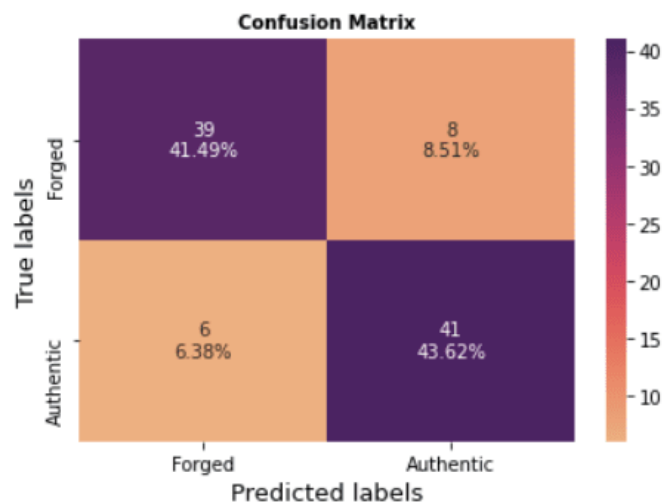


Figure 6: Confusion matrix showing model performance on the test set

The results demonstrated that the model was able to accurately distinguish between authentic and forged images, with high precision and recall values. The confusion matrix further confirmed that the majority of images were correctly classified, with only a small number of misclassifications.

INTERPRETATION AND INSIGHTS

The combination of ELA preprocessing, data augmentation, and a well-structured CNN architecture contributed to the model's strong performance. The training and validation curves indicated that the model did not suffer from significant overfitting, and the confusion matrix highlighted its reliability in real-world scenarios. These results suggest that the model is well-suited for deployment in practical image forgery detection applications. By following this structured and rigorous approach, the image forgery detection model achieved robust and reliable results, laying a strong foundation for further enhancements and real-world deployment.

3.6.2 MODEL TRAINING FOR VIDEO FORGERY DETECTION

The process of training a video forgery detection model begins with the careful preparation of the dataset. Videos are collected in both original (authentic) and forged (tampered) categories. Each video is systematically decomposed into individual frames using OpenCV, ensuring that every frame is captured for analysis. This granular approach allows the model to learn subtle differences between authentic and manipulated content at the frame level. The total number of frames extracted from both forged and original videos is balanced to prevent bias during training. Once extracted, frames are resized to a standard dimension (e.g., 240x320 pixels) to maintain consistency across the dataset. Each frame is labeled according to its source video—either as "original" or "forged." The dataset is then split into training, validation, and test sets, ensuring that the model is evaluated on unseen data for a fair assessment of its generalization ability.

MODEL ARCHITECTURE

For the task of video forgery detection, a deep convolutional neural network based on the ResNet50 architecture is employed. ResNet50 is chosen for its proven ability to capture complex spatial features and its robustness in handling large-scale image data. The model is initialized with random weights and adapted for binary classification, outputting a probability score for each frame indicating whether it is likely to be forged or authentic.

TRAINING PROCESS

The model is trained using the Adam optimizer and binary cross-entropy loss function, which are well-suited for binary classification problems. Training is conducted over multiple epochs, with the model learning to minimize the difference between predicted and actual labels. Data augmentation techniques, such as random rotations, flips, and zooms, are applied to the training frames to increase dataset diversity and improve the model's ability to generalize to new, unseen videos.

MODEL EVALUATION

After training, the model's performance is rigorously evaluated on the test set. Each frame in the test set is passed through the model, and the predicted labels are compared to the ground truth. The results are summarized using a confusion matrix, which provides a clear visualization of true positives, true negatives, false positives, and false negatives.

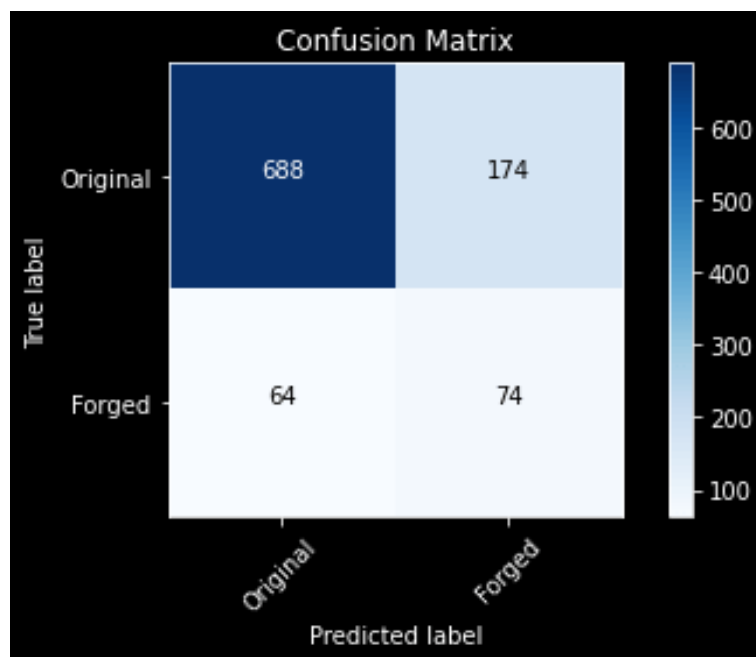


Figure 7: Confusion matrix for video forgery detection test results

In addition to the confusion matrix, other metrics such as accuracy, precision, recall, and F1-score are calculated to provide a comprehensive assessment of the model's effectiveness. These metrics help identify the model's strengths and areas for improvement, such as its ability to detect subtle manipulations or its tendency to misclassify certain types of frames.

AGGREGATING FRAME-LEVEL PREDICTIONS

Since video forgery detection ultimately requires a verdict at the video level, frame-level predictions are aggregated. For each video, the proportion of frames classified as forged is calculated. If this proportion exceeds a certain threshold, the entire video is flagged as potentially manipulated. This approach ensures that even localized tampering within a video can be detected, providing a robust solution for real-world scenarios.

INTERPRETATION AND INSIGHTS

The combination of a powerful ResNet50-based architecture, comprehensive data augmentation, and rigorous evaluation ensures that the video forgery detection model is both accurate and reliable. The confusion matrix and performance metrics demonstrate the model's ability to distinguish between authentic and forged frames, while the aggregation strategy enables effective video-level decision-making.

3.7 TOOLS AND TECHNOLOGIES USED

The development of the Forgery Check platform leverages a robust stack of modern tools and technologies, each carefully selected to address specific functional and performance requirements in image and video forgery detection. These technologies also support seamless web application development and ensure an optimal user experience across various devices. Below is an overview of the key components that power the system:

1. PROGRAMMING LANGUAGES

- **Python:** The core logic for image and video analysis, machine learning model training, and backend processing is implemented in Python. Its extensive ecosystem of libraries makes it ideal for rapid prototyping and scientific computing.
- **JavaScript, HTML, CSS:** These are used for crafting the interactive and responsive frontend, ensuring a seamless user experience.

2. WEB FRAMEWORK

- **Django:** The web application is built using Django, a high-level Python web framework. Django provides a secure, scalable, and maintainable structure for handling user authentication, routing, template rendering, and database management.

3. MACHINE LEARNING & DEEP LEARNING

- **TensorFlow & Keras:** These frameworks are used for designing, training, and deploying deep learning models for both image and video forgery detection. Keras offers a user-friendly API, while TensorFlow provides the computational backbone.
- **scikit-learn:** Utilized for data preprocessing, model evaluation, and generating metrics such as confusion matrices and classification reports.

4. IMAGE AND VIDEO PROCESSING

- **OpenCV:** Essential for frame extraction from videos, image resizing, and various preprocessing tasks required before feeding data into the models.
- **Pillow (PIL):** Used for image manipulation tasks such as Error Level Analysis (ELA), format conversion, and resizing.

5. VISUALIZATION AND REPORTING

- **Matplotlib & Seaborn:** These libraries are used to visualize training progress, model performance, and analysis results, including loss/accuracy curves and confusion matrices.
- **TQDM:** Provides progress bars for data processing and training loops.

6. DATABASE AND FILE MANAGEMENT

- **SQLite:** The default database for Django, used to store user data and other records.
- **FileSystemStorage:** Handles secure storage and retrieval of uploaded media files.

7. FRONTEND LIBRARIES AND FRAMEWORKS

- **Bootstrap 5:** Ensures a modern, responsive, and visually appealing user interface.

8. OTHER UTILITIES

- **hachoir:** Used for extracting metadata from uploaded files.
- **Streamlit:** Employed for rapid prototyping and testing of certain features during development.

By integrating deep learning, a robust backend, and an intuitive frontend, Forgery Check offers accurate forgery detection and a seamless user experience. This synergy enables efficient forensic analysis of digital media.

CHAPTER 4

IMPLEMENTATION

4.1 INTRODUCTION

This chapter delves into the practical realization of the Forgery Check platform, translating the system design and methodologies into a fully functional web application. The implementation phase bridges the gap between conceptual planning and a tangible product, focusing on how each component is brought to life using modern technologies and best practices.

The primary goal of this stage is to create a seamless, secure, and efficient environment where users can easily upload images and videos, initiate forgery detection, and interpret the results with confidence. The implementation covers both the backend and frontend aspects of the application, ensuring that complex machine learning models and forensic algorithms are integrated smoothly into a user-friendly interface. Attention is also given to optimizing system responsiveness and minimizing latency during file processing. Moreover, clear visual feedback and result summaries are provided to enhance user understanding and trust in the detection process.

Key areas addressed in this chapter include the structure of the Django web application, the organization of backend logic and frontend presentation, and the mechanisms for integrating advanced detection models into the workflow. Special attention is given to the flow of data from the moment a user uploads a file, through the analysis pipeline, to the delivery of clear and actionable results. The chapter also highlights how the system manages different types of media, handles user interactions, and maintains security and privacy throughout the process. Additionally, it explains the modular design that facilitates future scalability and the use of APIs to streamline communication between components. Emphasis is placed on maintaining system reliability and responsiveness under varying workloads.

By the end of this chapter, readers will gain a comprehensive understanding of how the Forgery Check platform operates in practice, from its architectural foundations to the user experience it delivers. This sets the stage for evaluating the system's performance and exploring its real-world impact in subsequent chapters. The chapter also lays the groundwork for future enhancements and potential integration with other forensic tools and platforms.

4.2 DJANGO WEB APPLICATION ARCHITECTURE

The Forgery Check platform is architected as a modular Django web application, designed to deliver a seamless and secure experience for users seeking to verify the authenticity of images and videos. The project leverages Django's Model-View-Template (MVT) pattern, which cleanly separates the data models, business logic, and user interface components. This structure not only enhances maintainability but also allows for rapid development and easy integration of new features. At the core of the application is the `website` app, which encapsulates all the main functionalities, including user authentication, media upload, analysis workflows, and result presentation. The project directory is organized to keep code, templates, static assets, and machine learning models logically separated. This clear organization ensures that both developers and future maintainers can easily navigate and extend the system. Key directories and their purposes include:

ForgeryCheck/: The main Django project folder containing settings, URLs, and configuration files.

website/: The primary app where all business logic, views, forms, and detection modules reside.

templates/: Houses all HTML templates, organized into subfolders for authentication, detection, and informational pages.

static/: Contains static assets such as CSS, JavaScript, and images for the frontend.

media/: Stores user-uploaded files, including images and videos to be analyzed.

ml_models/: Stores pre-trained machine learning models used for forgery detection.

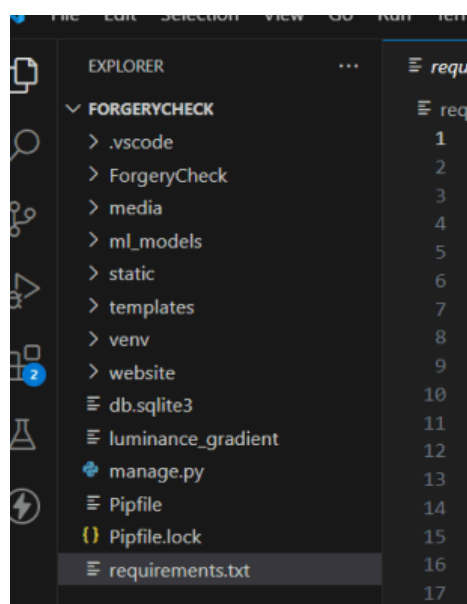


Figure 8: Folder structure of the 'ForgeryCheck' in VS Code

4.2.1 BACKEND ARCHITECTURE

The backend of Forgery Check is built on Django's robust framework, providing a secure and scalable foundation for all server-side operations. The backend is responsible for orchestrating the entire workflow from handling user requests and managing file uploads to invoking advanced machine learning models for forgery detection. It seamlessly connects the user interface with intelligent decision-making, ensuring a smooth and responsive experience. By leveraging Django's flexibility, the system can easily adapt to evolving project needs while maintaining reliability and efficiency. This architecture empowers the application to deliver accurate results in real time, enhancing user trust and overall system performance.

Key Components:

- **Views:** The `views.py` file in the website app contains all the logic for handling HTTP requests. Each view function corresponds to a specific route, such as image or video upload, analysis, and result display. The backend ensures that user actions are processed efficiently and that appropriate feedback is provided.
- **Forms:** Custom Django forms are used for user registration and media uploads, ensuring data validation and a smooth user experience.
- **Models:** The backend defines models for storing analysis records, user data, and other relevant information. This enables users to view their analysis history and ensures traceability of results.
- **Media Handling:** Uploaded files are securely stored in the `media/` directory using Django's `FileSystemStorage`. The backend manages file paths, ensures only supported formats are accepted, and cleans up temporary files as needed.
- **Machine Learning Integration:** The backend seamlessly integrates with pre-trained deep learning models for both image and video forgery detection. When a user submits a file, the backend invokes the appropriate detection pipeline, processes the results, and prepares them for display on the frontend.
- **Security:** Django's built-in authentication and permission system is leveraged to protect user data and restrict access to sensitive operations. CSRF protection, input validation, and secure file handling are enforced throughout the backend.
- **Metadata Extraction:** For both images and videos, the backend extracts metadata (such as file size, format, creation date, and technical properties) to provide users with additional context about their uploads.

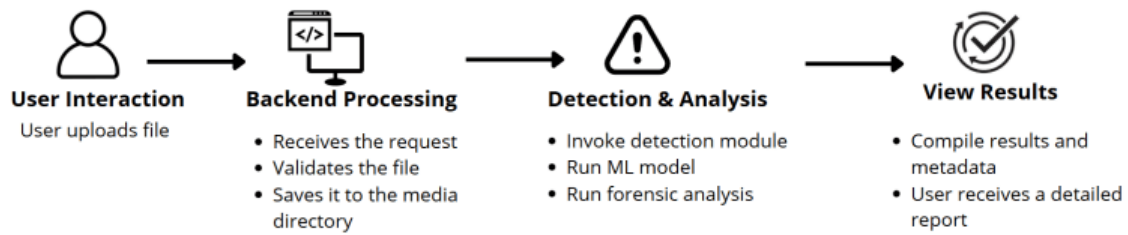


Figure 9: Media forgery detection backend process

The Django web application structure and backend architecture of Forgery Check are thoughtfully designed to balance security, scalability, and user-friendliness. By leveraging Django’s strengths and integrating advanced machine learning models, the backend ensures that users receive fast, accurate, and reliable forgery detection services.

4.2.2 FRONTEND INTERFACE

The frontend interface of the Forgery Check platform is designed to be intuitive, visually appealing, and highly interactive, ensuring that users of all backgrounds can easily navigate the system and access its powerful forgery detection features. Built using Django’s templating engine, the frontend leverages modern web technologies and design principles to deliver a seamless experience across devices.

USER EXPERIENCE AND DESIGN

The user interface adopts a clean, modern aesthetic with a focus on clarity and usability. Consistent color schemes, iconography, and responsive layouts are implemented using Bootstrap 5 and Font Awesome, making the platform accessible on both desktop and mobile devices. The navigation bar provides quick access to core features such as image and video analysis, user authentication, and informational pages like FAQs and "How It Works".

KEY PAGES AND FEATURES

1. Homepage

Upon visiting the homepage, users are greeted by a striking hero section that introduces the platform’s mission empowering users to verify the authenticity of images and videos. A bold visual, concise messaging, and a clear call-to-action invite users to begin their analysis journey. Below, the features section showcases core capabilities.

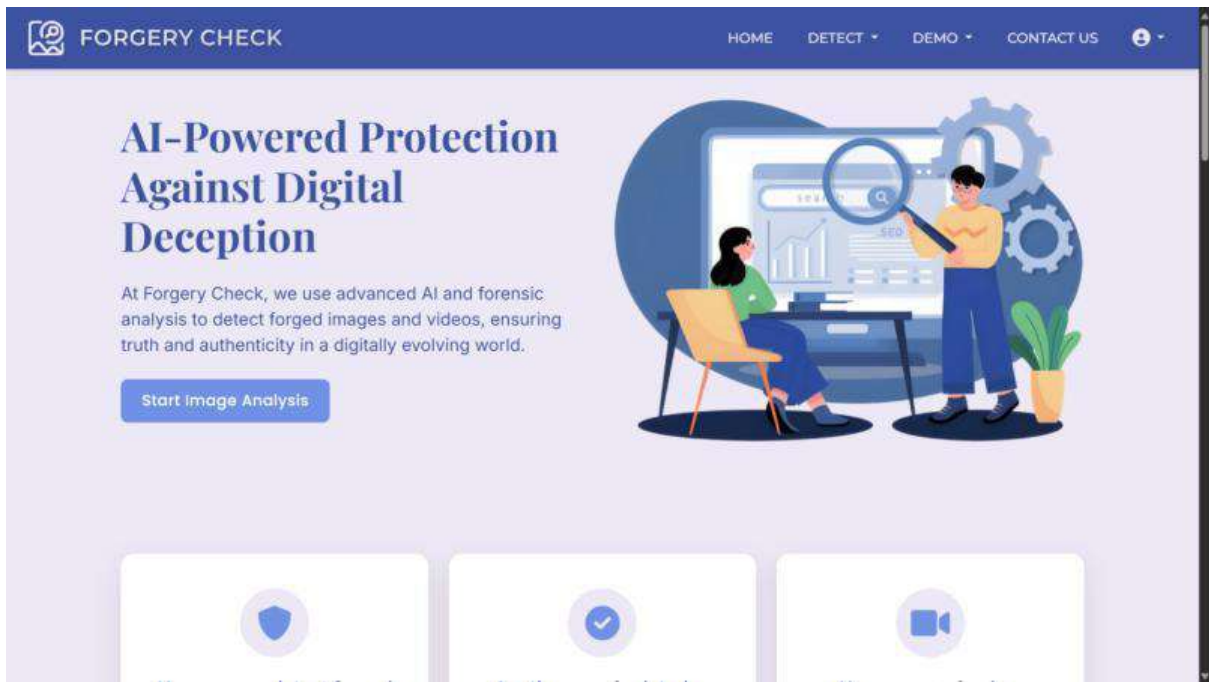


Figure 10: Homepage of the ForgeryCheck platform

2. Authentication: Login and Register Pages

The platform provides dedicated pages for user authentication. The login page offers a simple, secure form for existing users to access their accounts, while the registration page allows new users to sign up by providing essential details. Both pages are styled for clarity and include helpful prompts or error messages to guide users through the process.

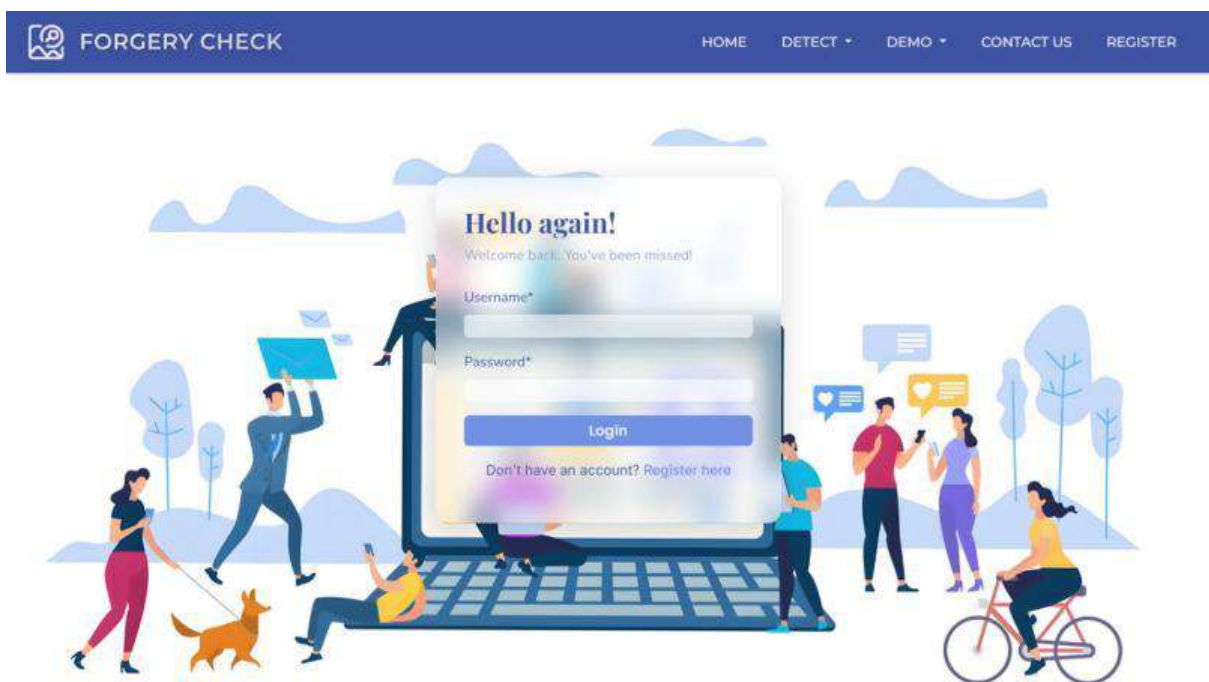


Figure 11: User login page

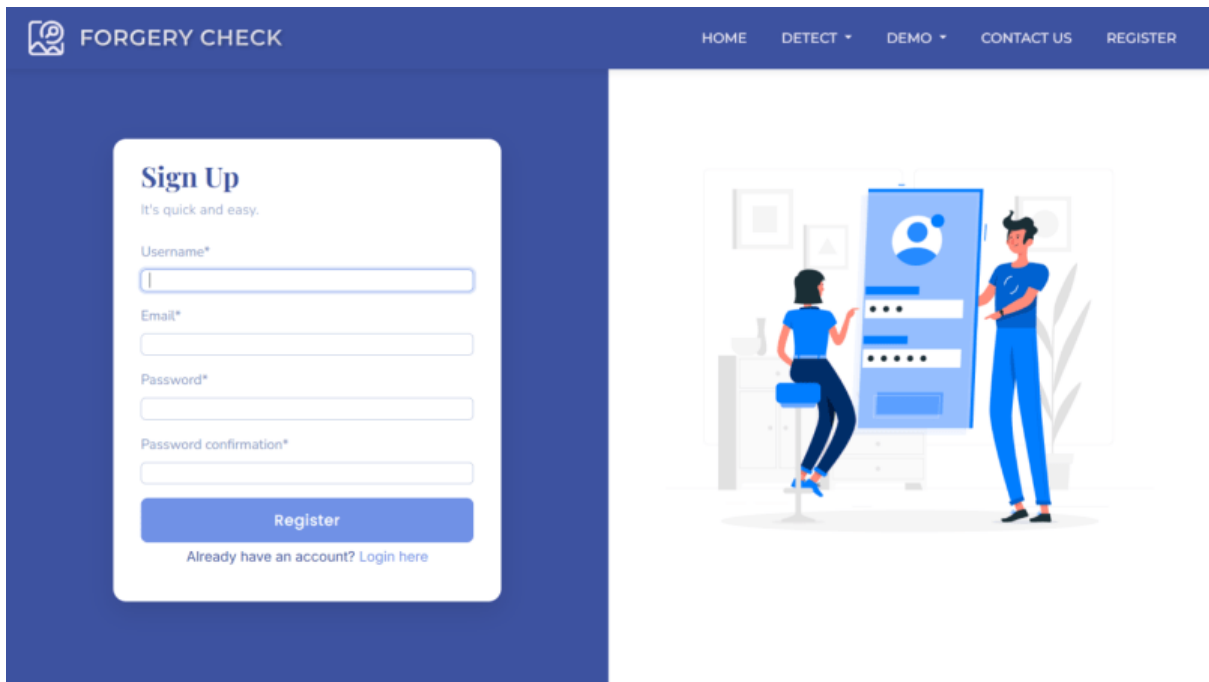


Figure 12: User registration page

3. How It Works Page

To demystify the technology, the "How It Works" page uses step-by-step cards and visuals to explain the detection process for both images and videos. This page walks users through uploading their media, the forensic analysis steps, and how results are generated, building trust and transparency.

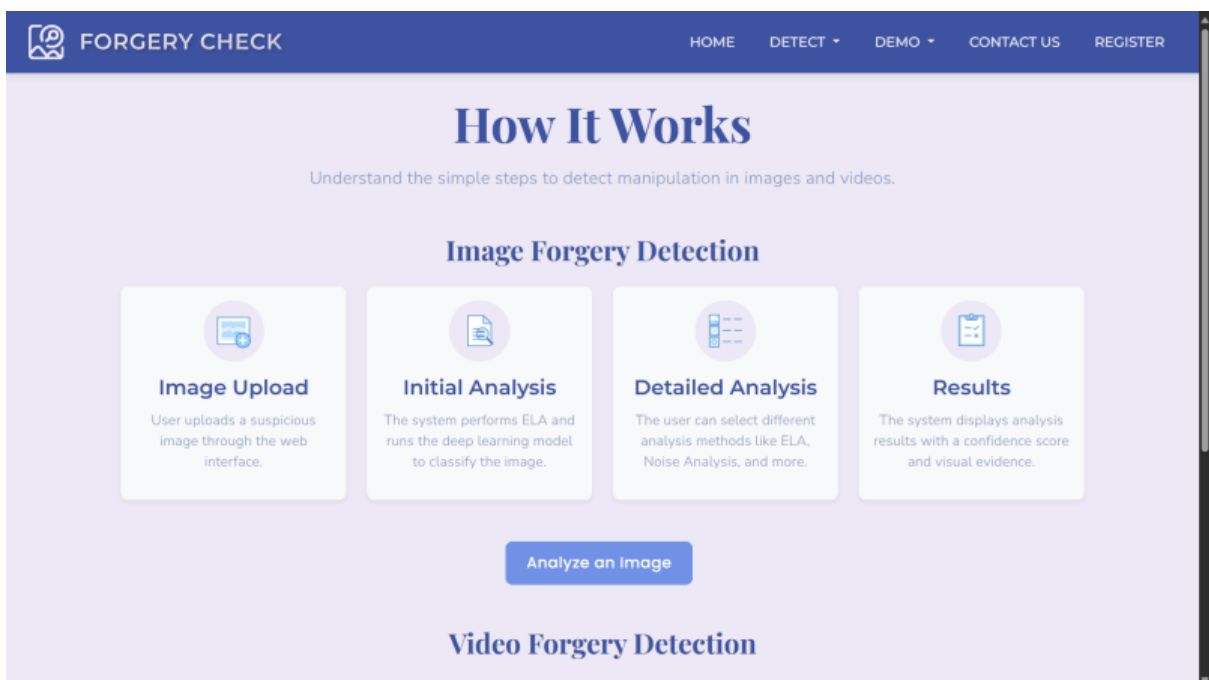


Figure 13: How It Works page – Video analysis workflow

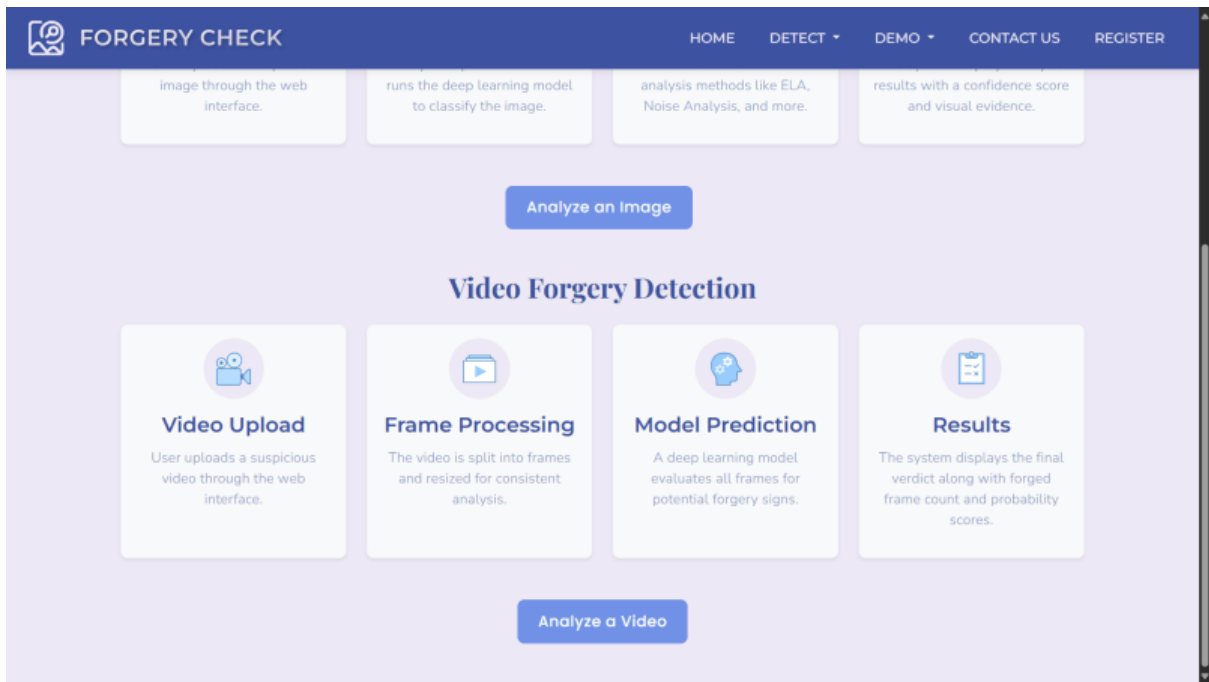


Figure 14: How It Works page – Video analysis workflow

4. FAQs Page

The FAQs page addresses common questions about the platform, such as supported file types, privacy, account management, and analysis capabilities. Organized in an accordion layout, users can easily browse and expand questions relevant to their needs.

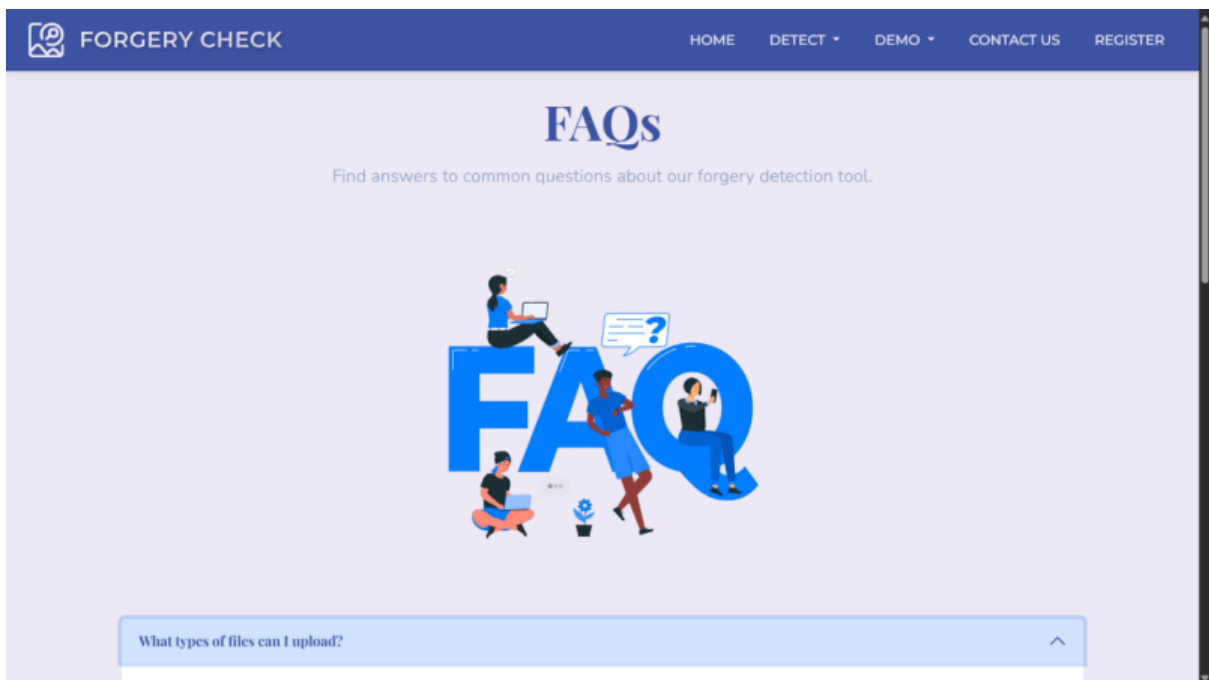


Figure 15: FAQs page

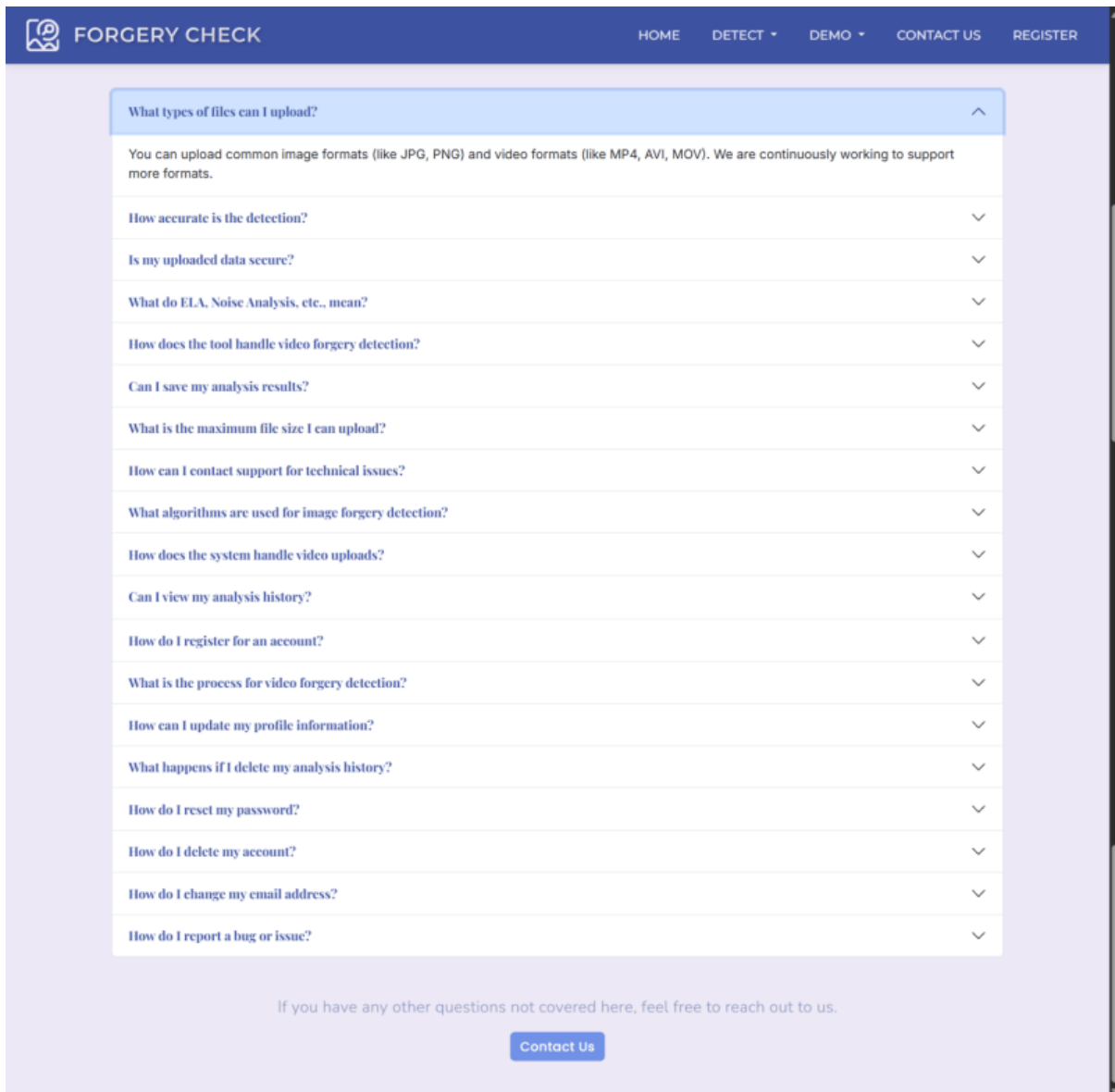


Figure 16: FAQs page – Expanded view

5. Contact Us Form

The Contact Us page of Forgery Check is designed with a warm, welcoming tone to make users feel supported and heard. At the top, a friendly headline “How can we help you?” sets the stage for open communication. Below, users find a simple, intuitive form where they can enter their name, email, and message, making it easy to get in touch for support, feedback, or general inquiries.

Accompanied by a clean layout and a helpful illustration, the page also includes key contact details such as the official email address, phone number, and the physical location at VIT-AP University. Whether users have technical questions or just want to share suggestions, the page ensures that every message is directed to the right team for a timely response.

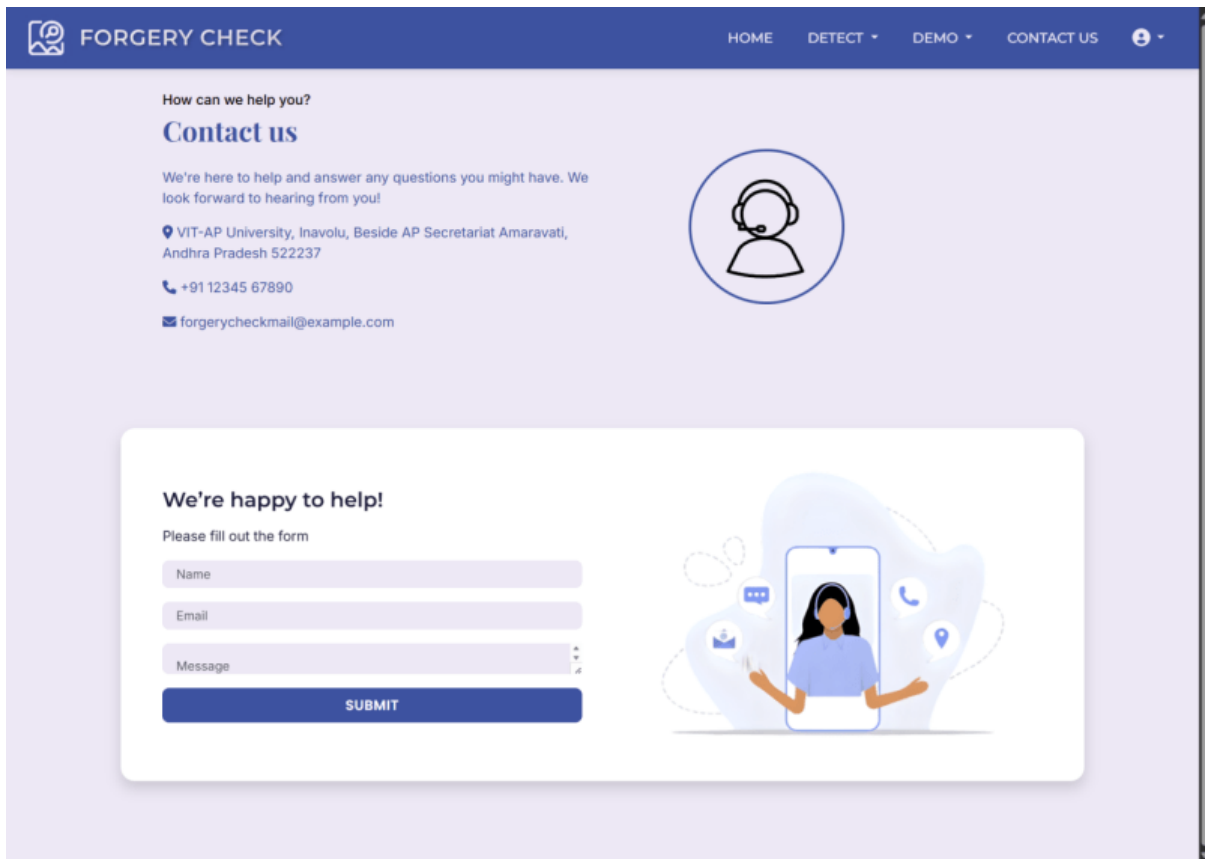


Figure 17: Contact Us page with form and contact details

6. My Profile Page

Logged-in users have access to a personalized profile page. Here, users can view and update their account information, such as email and name. The profile page features a visually appealing card layout, with clear sections for profile details and account settings.

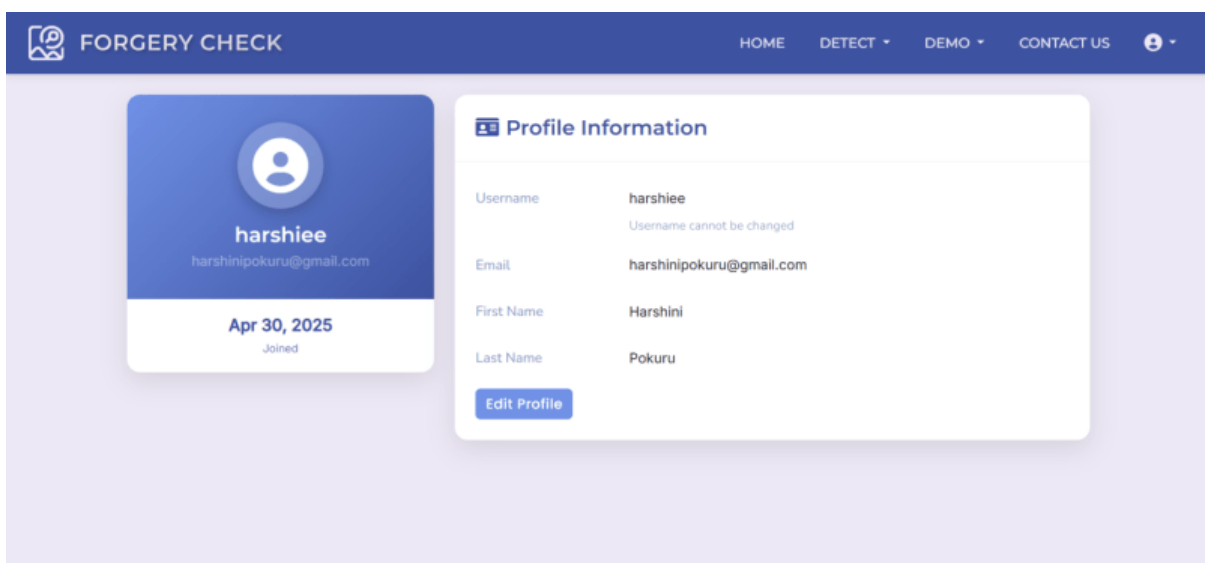


Figure 18: User profile page

7. Analysis History Page

While the analysis history feature is planned for future enhancement, the current page is designed to eventually display a user's past analyses. The layout anticipates showing each analysis as a card, with details. For now, users see a friendly message and illustration if no history is available, along with quick links to start a new analysis.

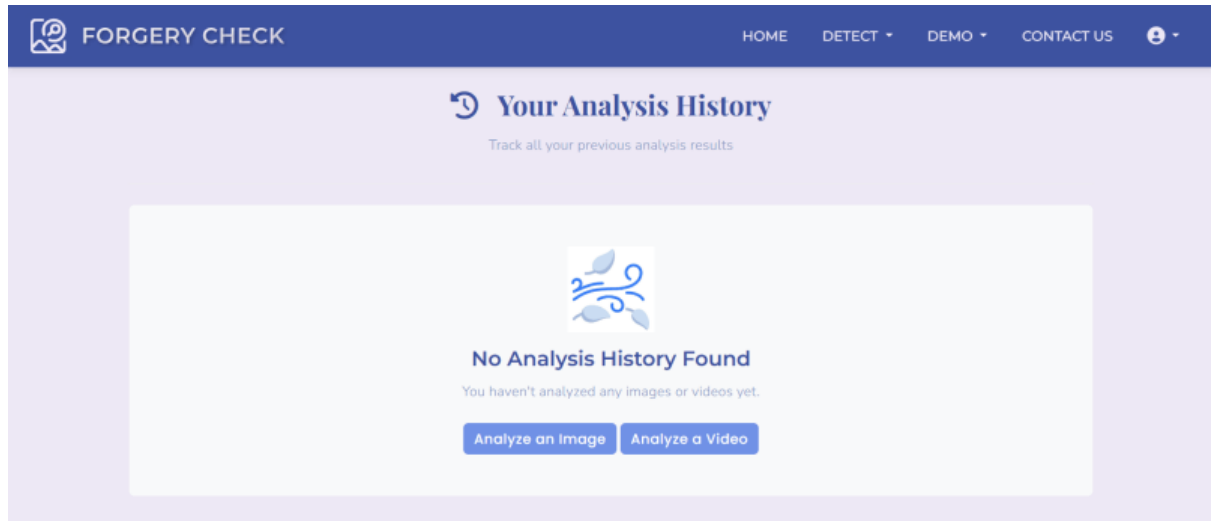


Figure 19: Analysis history page

8. Image Upload and Analysis

The image upload page offers a drag-and-drop interface or file browser for submitting images. Upon upload, users see a loading overlay indicating that analysis is in progress. Once complete, the results page presents the original image, forensic masks, and a detailed report, all organized for easy interpretation.

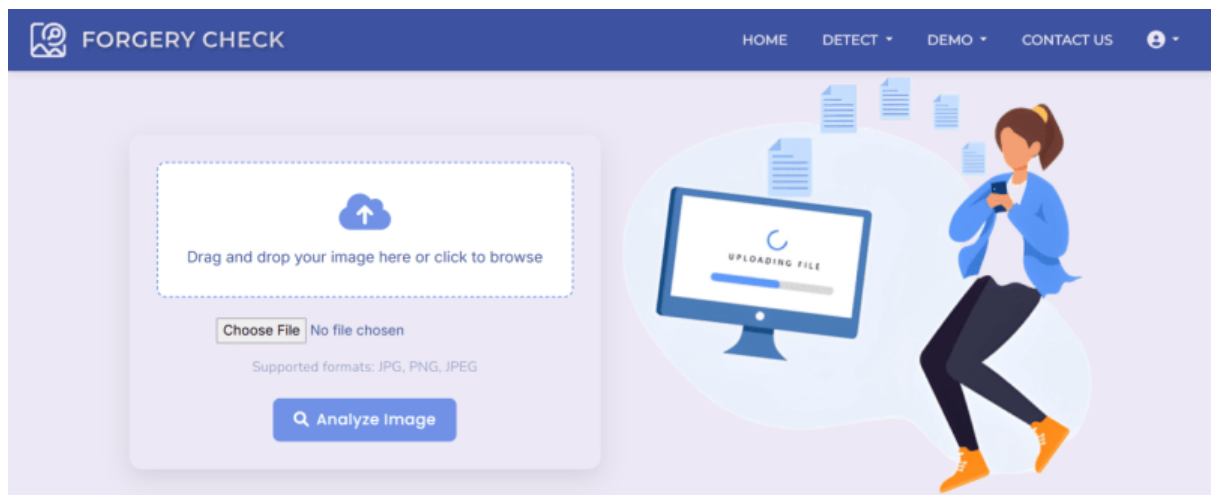


Figure 20: Image upload page

9. Video Upload and Analysis

Similarly, the video upload page allows users to submit videos for analysis. The interface guides users on supported formats and provides feedback during processing. The results page summarizes the analysis verdict, frame-by-frame findings, and relevant metadata.

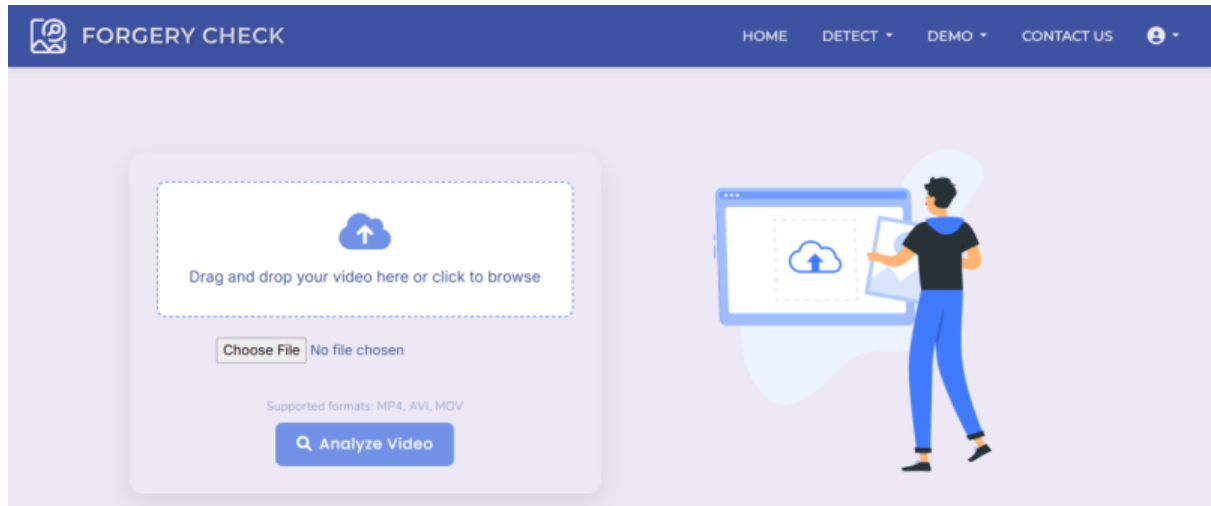


Figure 21: Video upload page

10. Footer with Direct Links

Every page features a consistent footer, providing direct links to essential sections. This ensures users can easily navigate the platform from anywhere.

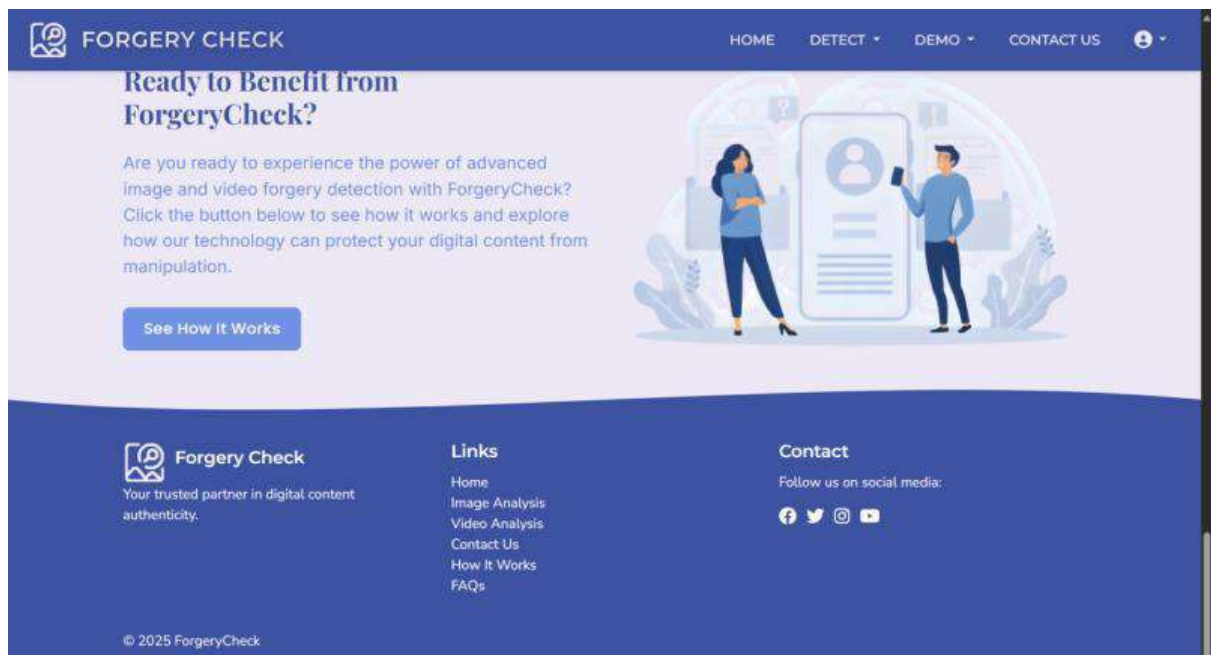


Figure 22: Footer section with quick navigation links

11. Navigation and User Experience

The navigation bar adapts based on authentication status, offering quick access to profile, analysis history, and logout for logged-in users. Interactive elements like loading spinners, progress bars, and clear feedback messages enhance the overall user experience.



Figure 23: Navigation bar and user interface elements

The Forgery Check frontend is designed to be welcoming, informative, and efficient. From the first visit to in-depth analysis, every page and feature is built to support users in verifying digital content with confidence. Including screenshots of the key pages and states described above will greatly enrich your documentation and showcase the platform's strengths.

4.3 MODEL INTEGRATION INTO WEB APP

Integrating advanced machine learning models into the Forgery Check web application is a pivotal aspect of its functionality. The backend, built with Django, acts as the bridge between user interactions and the deep learning models responsible for analyzing images and videos. When a user uploads a file for analysis, the backend securely stores the media and invokes the appropriate detection pipeline. For images, the system leverages pre-trained neural networks and forensic algorithms to assess authenticity, while for videos, frames are extracted and processed through specialized models capable of identifying manipulations such as deepfakes or frame tampering. The integration is handled through modular Python scripts and model files stored in the `ml_models` directory, ensuring maintainability and scalability.

This architecture allows the application to deliver rapid, accurate results directly to the user interface. The models are loaded only when needed, optimizing server resources and response times. By abstracting the complexity of machine learning, the platform empowers users to access cutting-edge forgery detection with just a few clicks, making advanced digital forensics accessible to everyone.

4.4 IMAGE ANALYSIS OUTPUT

Once a user uploads an image for analysis, they are directed to a dedicated results page. At the top, a prominent heading as "Image Analysis" immediately informs the user of the report's purpose. This is followed by a summary verdict, which concisely states whether the image is likely authentic or forged, often accompanied by a confidence score.

The results page is organized into two main columns. On the left, the original uploaded image is displayed, allowing users to recall the content they submitted. On the right, a dynamic section presents processed images highlight different forensic techniques applied, such as Mask, Edge Map, Error Level Analysis (ELA), noise analysis, and copy-move detection. This visual comparison allows users to see exactly where potential manipulations may have occurred.

Below the visual outputs, a detailed findings section provides textual explanations of the results. This section includes:

- A breakdown of which forensic techniques detected anomalies.
- Technical metadata about the image (such as file type, dimensions, and creation date).

4.4.1 FORGED IMAGE RESULT



Figure 24: Forged image result page

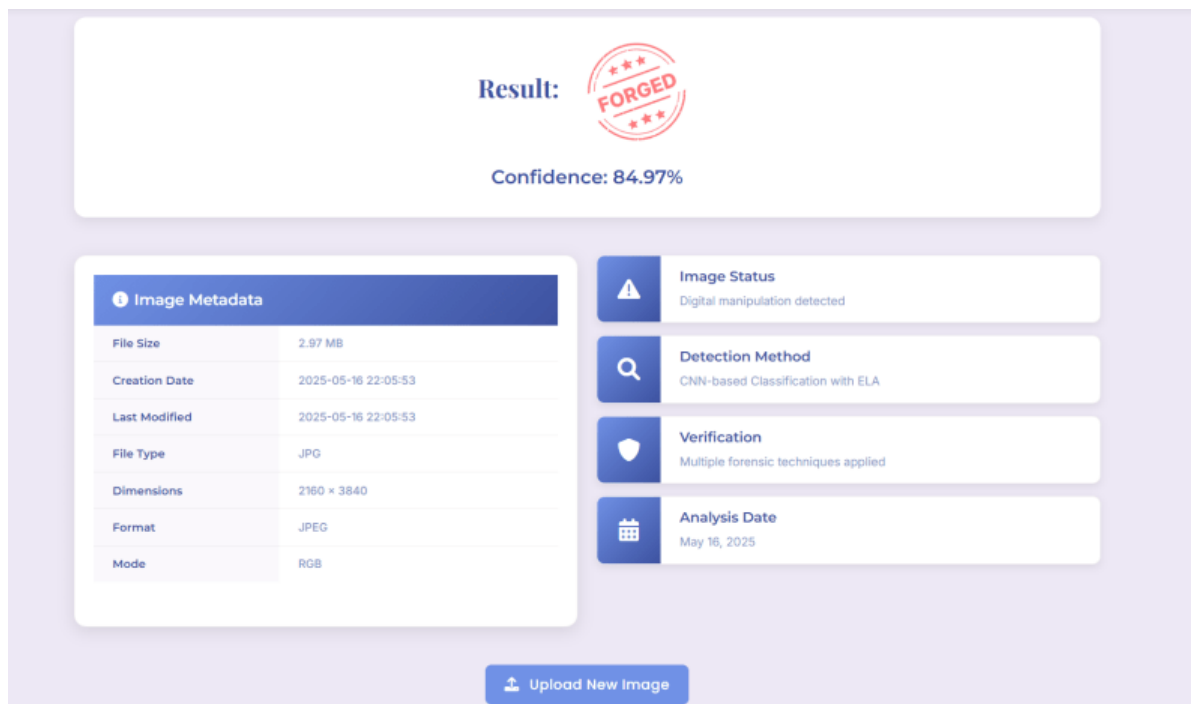


Figure 25: Forged image – Results overview

The core of this experience is a set of buttons, each corresponding to a different forensic technique. These allow users to visualize suspected forged regions using multiple scientific approaches. Below, each method is described with a suggested heading, explanatory text, and where to place example images.

1. **Mask:** The Mask button uses a deep learning model to segment and highlight manipulated regions. This mask overlays the original image, visually marking areas the AI suspects have been altered.

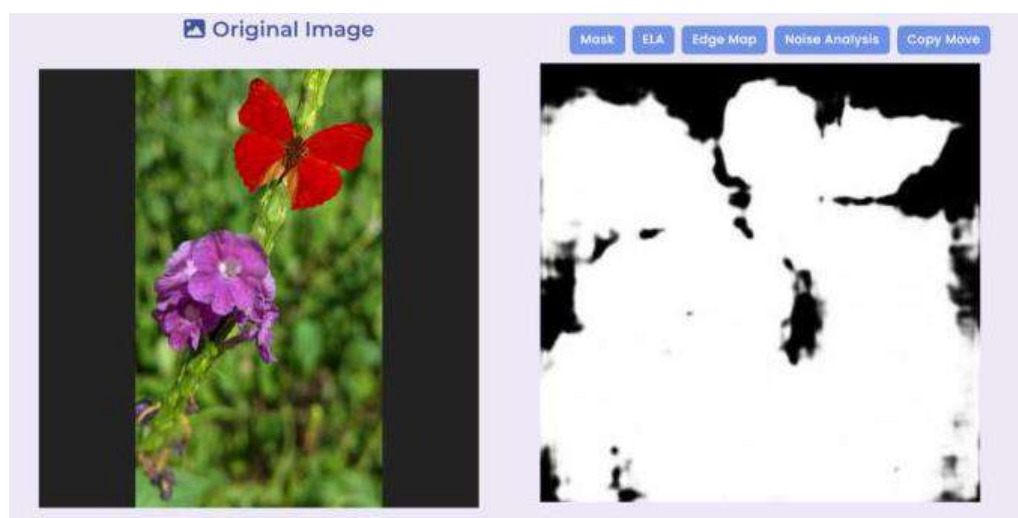


Figure 26: Deep Learning Mask image

2. **Error Level Analysis (ELA):** ELA highlights areas of an image that have different compression levels, which often indicate digital tampering. When you click the ELA button, regions with inconsistent compression are visually emphasized, making it easier to spot possible edits.

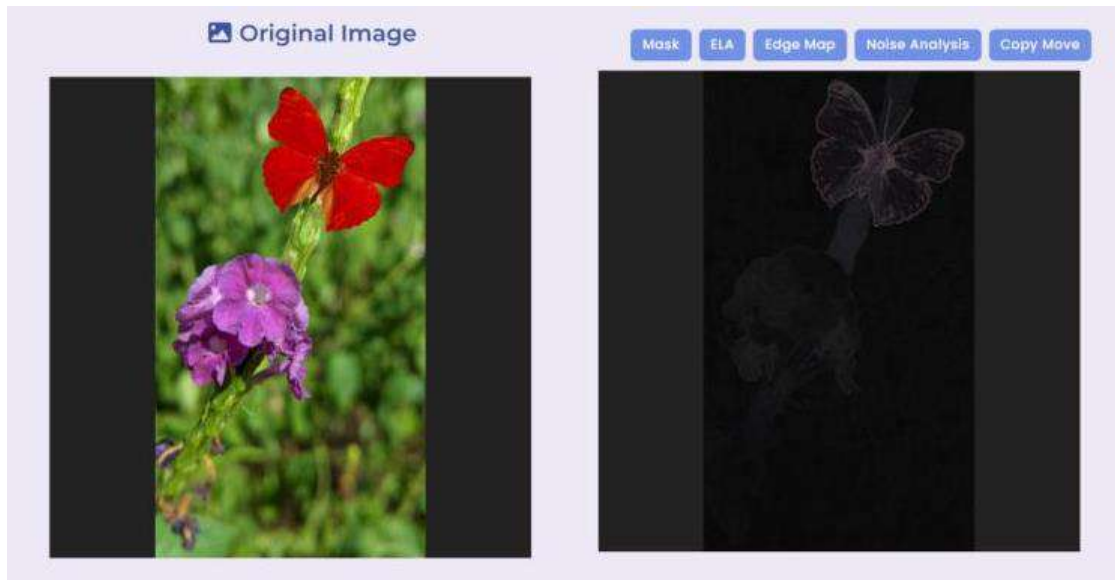


Figure 27: Error Level Analysis output

3. **Edge Map Analysis:** By clicking the Edge Map button, users can see the result of edge detection algorithms. Inconsistencies in edges such as abrupt changes or unnatural boundaries can be signs of tampering.

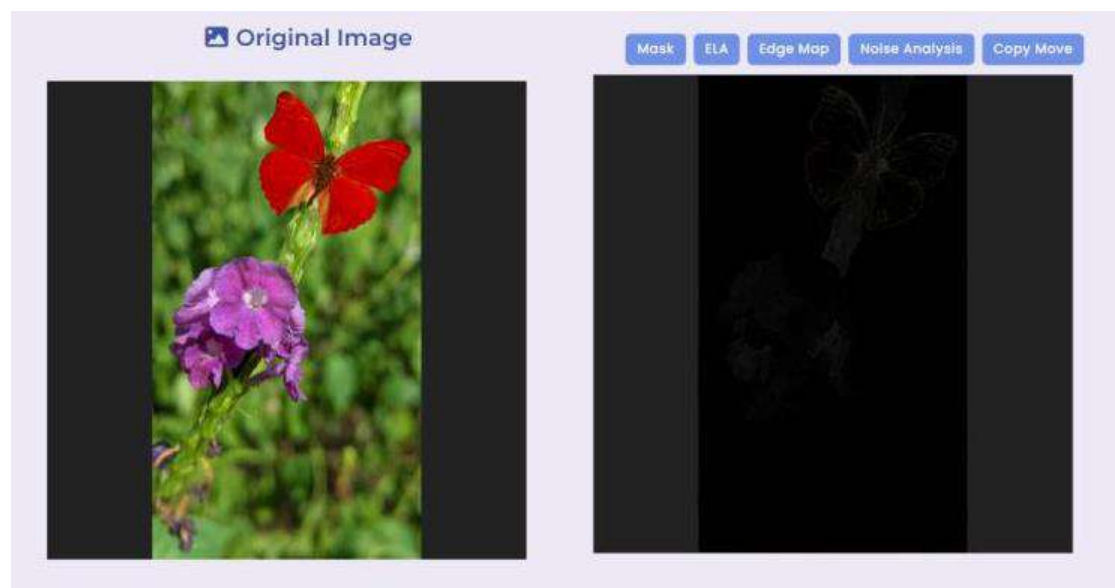


Figure 28: Edge Map output

4. **Noise Variance Analysis:** The Noise Analysis button reveals variations in the noise pattern across the image. Forged regions often have different noise characteristics compared to the rest of the image, which this analysis can expose.

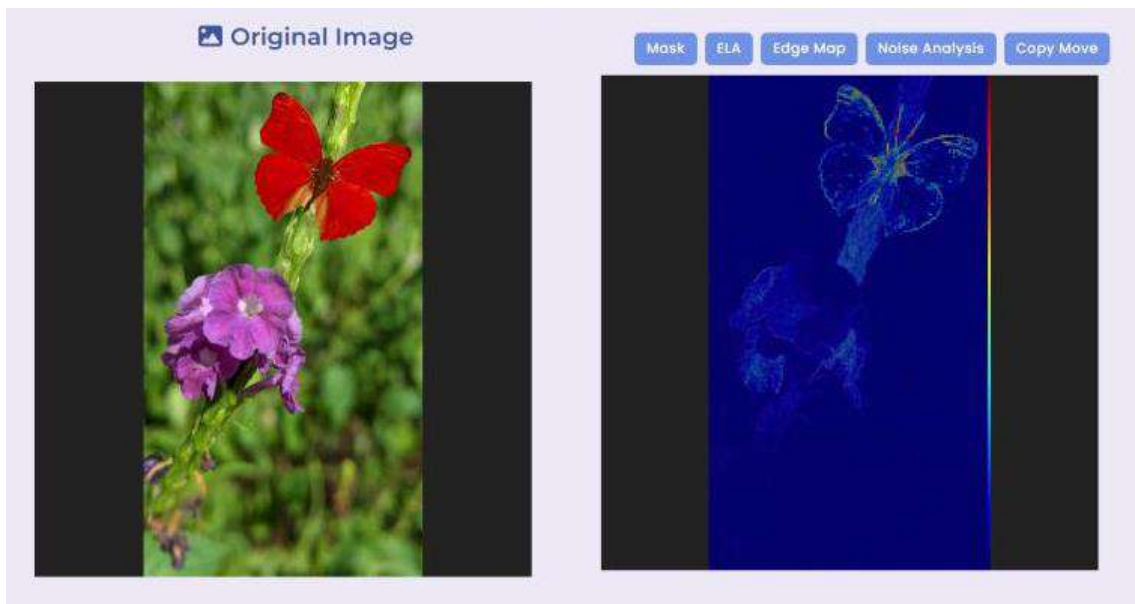


Figure 29: Noise Analysis output

5. **Copy-Move Forgery Detection (SIFT):** The Copy-Move button leverages SIFT feature matching to detect duplicated regions within the same image a common forgery technique. Detected regions are connected by lines or highlighted contours.

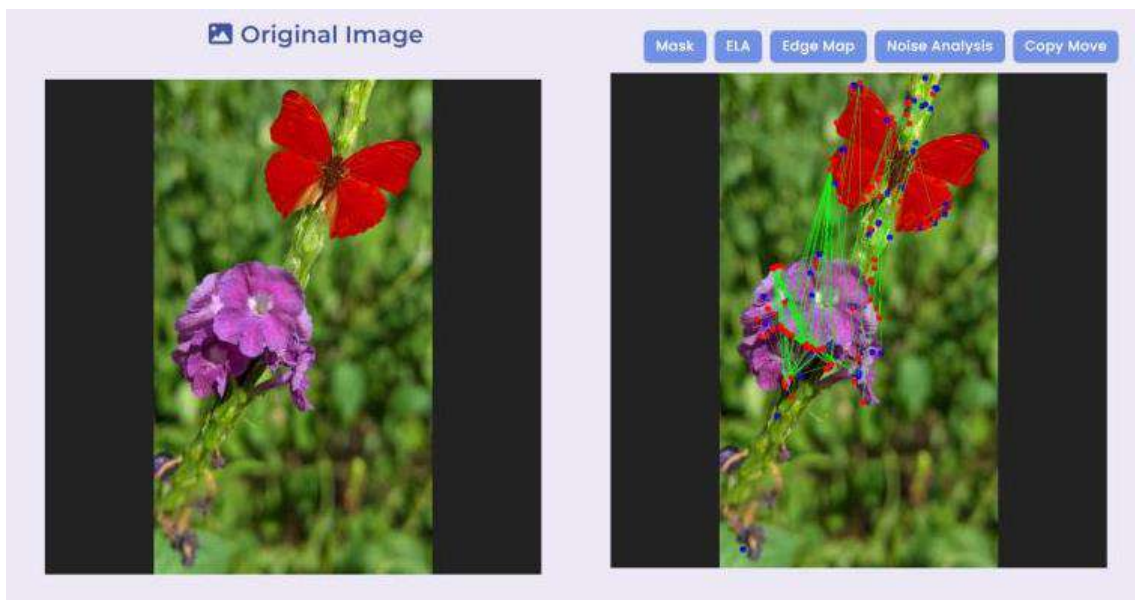


Figure 30: Copy-Move Detection output

4.4.2 AUTHENTIC IMAGE RESULT

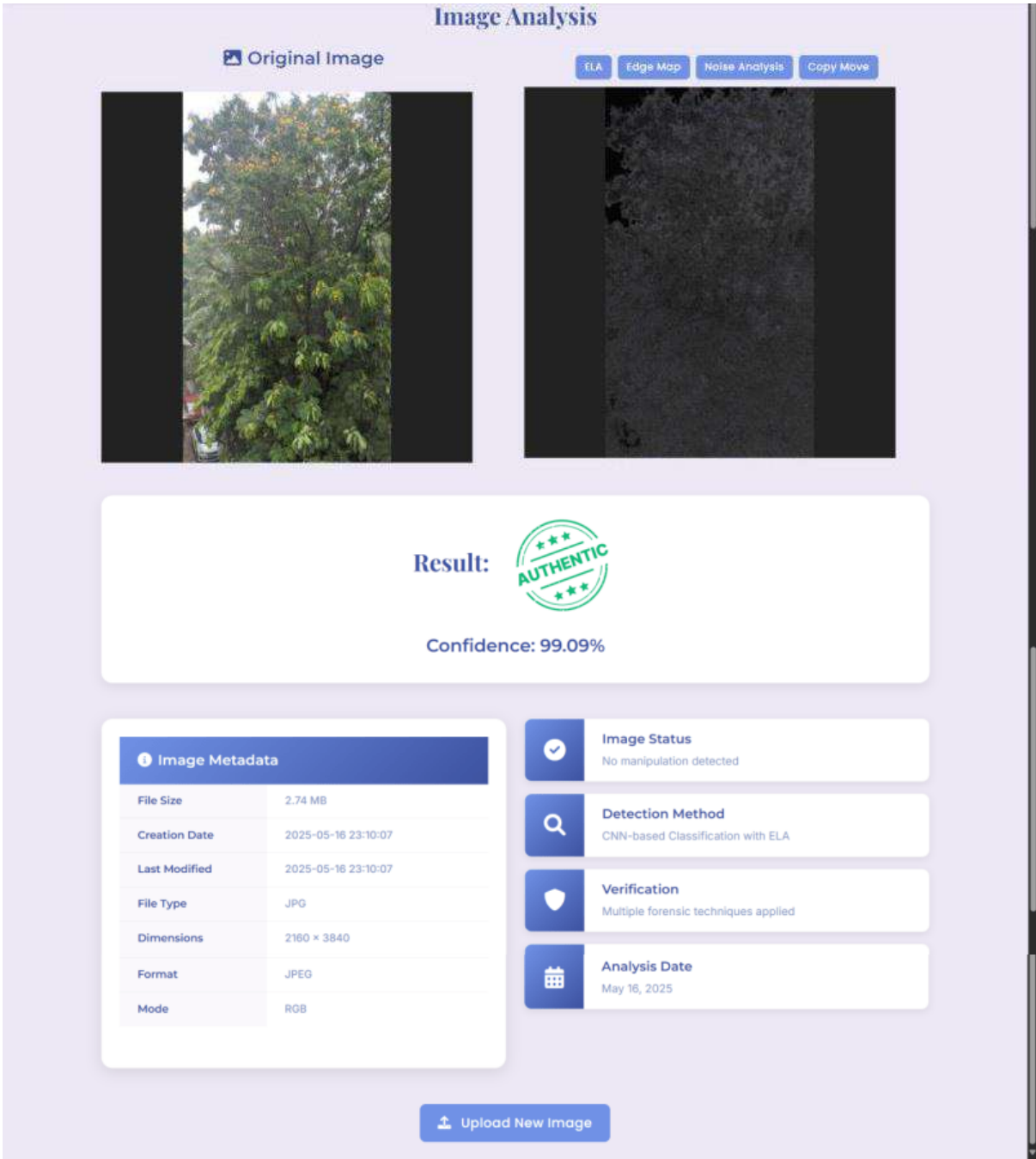


Figure 31: Authentic image result page

The image analysis output page in Forgery Check is designed to bridge the gap between advanced digital forensics and user-friendly reporting. Through clear headings, interactive forged region buttons, side-by-side image displays, and detailed textual findings, users are empowered to confidently interpret the authenticity of their images.

4.5 VIDEO ANALYSIS OUTPUT

The video analysis output page in Forgery Check is more than just a verdict. It's a comprehensive, interactive, and user-friendly forensic report. By combining detailed visual evidence, technical metadata, confidence metrics, and actionable guidance, the platform empowers users to confidently assess the authenticity of their videos and take appropriate next steps. Once a user uploads a video for analysis, they are directed to a dedicated results page that clearly presents the outcome of the forensic examination. At the top, a prominent heading such as Video Analysis immediately informs the user of the report's purpose. This is followed by a summary verdict, which concisely states whether the video is likely authentic or forged, often accompanied by supporting details such as the number of manipulated frames detected.

The results page is thoughtfully organized to help users understand the findings at a glance. On the left, the original uploaded video is displayed, allowing users to review the content they submitted. On the right, a dynamic section presents key analysis results, including frame-by-frame breakdowns and visualizations of detected manipulations. This side-by-side layout makes it easy for users to compare the original footage with the forensic evidence.

Below the visual outputs, a detailed findings section provides textual explanations of the results. This section includes:

- A breakdown of which frames or segments were flagged as manipulated.
- Technical metadata about the video (such as file type, duration, resolution, and frame rate).
- The detection method used (e.g., deep learning frame analysis).

Beyond the technical details, the findings section is crafted to be approachable for users of all backgrounds. Each flagged frame or segment is clearly listed, making it easy to pinpoint exactly where potential tampering has occurred. The metadata offers transparency, helping users understand the context of the video. By specifying the detection method, the system builds trust, showing that advanced, state-of-the-art techniques were used in the analysis.

4.5.1 FORGED VIDEO RESULT



Figure 32: Forged video result page

The core of the video analysis experience is a set of visual and textual cues that highlight where and how manipulation was detected. The system analyzes the video frame by frame using advanced neural networks, flagging any frames that show signs of tampering. The results page displays:

- A timeline or list of frames identified as forged.
- A summary indicating the total number of manipulated frames.

1. **Frame-by-Frame Analysis:** Each frame is examined for inconsistencies such as deepfake faces, unnatural transitions, or digital artifacts.
2. **Results Summary:** A clear summary states the verdict (e.g., "Digital manipulation detected") and quantifies the extent of forgery (e.g., "12 frames show evidence of manipulation").
3. **Metadata and Detection Method:** The findings section lists technical details about the video and explains the detection approach used.

4.5.2 AUTHENTIC VIDEO RESULT

The screenshot displays a web interface for video analysis. At the top, a 'Video Preview' section shows a street scene with a car. Below this, a 'Result' section features a green circular stamp with the word 'AUTHENTIC' and the text 'Video is'. To the left, a 'Video Metadata' table lists file details. To the right, a sidebar contains four status boxes: 'Video Status' (No manipulation detected), 'Detection Method' (Deep Learning Frame Analysis), 'Verification' (Frame-by-frame analysis shows no tampering), and 'Analysis Date' (May 16, 2025). An 'Upload New Video' button is at the bottom.

Video Metadata	
File Name	IAVFD_Au_VI_25.mp4
File Size	1.63 MB
Creation Date	2025-05-17 00:12:15
Last Modified	2025-05-17 00:12:15
File Type	MP4

	Video Status No manipulation detected
	Detection Method Deep Learning Frame Analysis
	Verification Frame-by-frame analysis shows no tampering
	Analysis Date May 16, 2025

[Upload New Video](#)

Figure 33: Authentic video result page

CHAPTER 5

TESTING AND RESULTS

5.1 INTRODUCTION

This chapter presents a comprehensive overview of the testing strategies and results for the ForgeryCheck platform. The goal is to demonstrate the reliability, effectiveness, and real-world applicability of both the web application and the underlying machine learning models.

Testing was conducted in two main phases. First, the core deep learning models for image and video forgery detection were trained and evaluated using Jupyter notebooks in Google Colab. These environments provided the computational resources necessary for large-scale data processing and model experimentation, ensuring that the models achieved robust performance before integration. Second, the fully integrated Django web application was subjected to thorough functional testing. This included verifying user interactions, file uploads, analysis workflows, and the clarity of the output reports. The aim was to ensure that users receive accurate, timely, and understandable results, regardless of their technical background.

Throughout this chapter, we detail the test environment, describe the procedures used to validate both the backend models and the frontend user experience, and present key performance metrics. The results highlight not only the accuracy of the detection algorithms but also the usability and reliability of the overall system.

5.2 TEST ENVIRONMENT AND SETUP

To ensure the accuracy and reliability of the ForgeryCheck platform, a carefully structured test environment was established for both model development and web application validation. The core machine learning models for image and video forgery detection were developed and evaluated using Google Colab notebooks. Google Colab provided a cloud-based environment with access to high-performance GPUs, which was essential for handling large datasets and accelerating deep learning workflows. All necessary libraries such as TensorFlow, Keras, NumPy, OpenCV, and scikit-learn, were installed within the Colab runtime, ensuring a consistent and reproducible setup for training and testing.

The data preprocessing phase involved organizing and preparing both forged and authentic samples. Videos were split into individual frames, and images were standardized in size and format. This step was crucial for building robust models capable of generalizing to real-world scenarios. For the web application, the Django-based ForgeryCheck system was deployed and tested on a Windows machine. The environment included Python 3.9, Django 3.2, and all required dependencies as specified in the project's configuration files. Test media files were stored in dedicated directories to simulate real user uploads, and the application was accessed via a local server for end-to-end testing.

By combining model training and evaluation with local web application deployment, the test environment closely mirrored practical usage conditions. This approach ensured that both the backend algorithms and the user-facing features were thoroughly validated, providing confidence in the platform's readiness for real-world use.

5.3 FUNCTIONAL TESTING

To ensure that ForgeryCheck delivers a seamless and reliable experience for users, the web application underwent comprehensive functional testing. The primary goal was to verify that all features, from file upload to result visualization, work as intended and provide clear, actionable feedback to users.

Testing began with the core user flows. Both image and video uploads were tested using a variety of authentic and manipulated files. The system was observed for its ability to handle different file formats, sizes, and edge cases such as corrupted or unsupported files. Each upload triggered the appropriate backend analysis pipeline, and the results were checked for accuracy and clarity. Special attention was given to the user interface. The results pages were reviewed to ensure that verdicts, such as Authentic or Forged were displayed prominently, along with supporting details like confidence scores, frame-by-frame breakdowns, and technical metadata. The application was intentionally provided with problematic inputs to confirm that it gracefully informs users of issues such as invalid file types or processing errors without crashing or producing confusing messages.

Overall, the functional testing process confirmed that ForgeryCheck is robust, user-friendly, and capable of guiding users through the process of digital media verification with confidence.

5.4 MODEL ACCURACY AND PERFORMANCE METRICS

The effectiveness of ForgeryCheck hinges on the accuracy and reliability of its underlying machine learning models. Both the image and video forgery detection models were rigorously evaluated using dedicated Jupyter notebooks.

Image Model:

Training and evaluation were conducted in ``ImageForgeryClassification.ipynb``. The model was trained on a diverse dataset of authentic and forged images, utilizing techniques such as Error Level Analysis (ELA) and convolutional neural networks (CNNs). Performance was measured using standard metrics including accuracy, precision, recall, and F1-score. The image forgery detection model achieved an accuracy of 84.84% , demonstrating strong performance in distinguishing subtle manipulations from genuine content.

Video Model:

The video forgery detection pipeline was developed and tested using ``Data_Preprocessing.ipynb``, ``Model_Training.ipynb``, ``Model_Testing.ipynb``, and ``VideoForgeryDetection.ipynb``. Videos were split into frames, preprocessed, and fed into a deep learning model (e.g., ResNet-based architecture). The model's predictions were evaluated on a held-out test set, with metrics such as frame-level accuracy, forged frame detection rate, and overall video classification accuracy. The results demonstrated that the model could reliably flag manipulated segments, even in challenging cases involving deepfakes or subtle edits. The video forgery detection model achieved an accuracy of 76.2% , reliably flagging manipulated segments, even in challenging cases involving deepfakes or subtle edits.

These results confirm that ForgeryCheck's models are both accurate and practical, providing users with trustworthy insights into the authenticity of their digital media. Ongoing work includes expanding the training datasets, refining model architectures, and incorporating new detection techniques to boost performance even further. By continuously evolving, ForgeryCheck aims to stay ahead of emerging forgery methods and provide users with even greater confidence in their digital content verification.

CHAPTER 6

DISCUSSION AND ANALYSIS

6.1 INTERPRETATION OF RESULTS

The results obtained from the ForgeryCheck system provide valuable insights into its effectiveness in detecting manipulated images and videos. A closer look at the performance metrics reveals that the image model excels at identifying subtle alterations, thanks to the combination of Error Level Analysis (ELA) and deep learning techniques. The high accuracy suggests that the model can generalize well across a variety of image forgeries, including those involving minor edits or sophisticated manipulations. For video analysis, the slightly lower accuracy reflects the inherent complexity of video forgery detection, where manipulations can occur at the frame level and may involve advanced techniques such as deepfakes. Despite these challenges, the model demonstrates a strong ability to flag suspicious segments and provide users with actionable results.

Overall, the results confirm that ForgeryCheck offers a practical and effective solution for digital media verification. The system's confidence scores and visual outputs further enhance user trust, making it a valuable tool for anyone seeking to validate the authenticity of images and videos.

6.2 STRENGTHS OF THE PROPOSED SYSTEM

The ForgeryCheck platform demonstrates several notable strengths that set it apart as a robust solution for digital media verification. One of its primary advantages is the integration of advanced machine learning techniques, including deep learning models and forensic analysis methods such as Error Level Analysis (ELA) and noise pattern examination. This multi-faceted approach enables the system to detect a wide range of manipulations, from simple edits to sophisticated forgeries.

Another key strength lies in the user-friendly design of the web application. The platform offers an intuitive interface that guides users through the process of uploading media, selecting analysis methods, and interpreting results. Visual outputs, such as highlighted regions of suspected tampering and clear confidence scores, make the findings accessible even to users without technical expertise.

ForgeryCheck also excels in its adaptability and scalability. The modular architecture allows for easy updates and the integration of new detection techniques as the field of digital forensics evolves. Additionally, the system is designed to handle both images and videos, providing comprehensive coverage for various types of digital content. Finally, the platform places a strong emphasis on privacy and security. Uploaded files are processed securely, and user data is handled with care, ensuring that sensitive information remains protected throughout the analysis process.

6.3 LIMITATIONS AND CHALLENGES

Evolving Forgery Techniques: The system must keep pace with the rapid development of new manipulation strategies, including highly realistic deepfakes and AI-driven forgeries that are becoming harder to detect.

Dataset Limitations: Detection accuracy is closely tied to the variety and quality of training data; if the dataset lacks certain forgery styles, the model's reliability may decrease for those cases.

Resource Demands: Processing high-resolution images or lengthy videos can be computationally intensive, sometimes leading to slower analysis and impacting the user experience.

Borderline Cases: Some analysis results may remain inconclusive, making it necessary for human experts to review and interpret subtle or ambiguous manipulations.

Adversarial Attacks: There is always a possibility that forgeries are deliberately engineered to evade detection, which means the system needs regular updates and monitoring.

Generalization: The models might not deliver consistent performance across all types of media or manipulation techniques, and may require further adjustment for new formats or sources.

These challenges highlight the need for continuous improvement and regular updates to maintain the effectiveness and reliability of ForgeryCheck.

6.4 COMPARATIVE ANALYSIS WITH EXISTING SOLUTIONS

When evaluating ForgeryCheck against other digital forgery detection tools, several distinctions become clear in terms of technology, usability, and overall effectiveness.

Technological Approach: Most digital forgery detection tools rely on single-method techniques like Error Level Analysis (ELA) or metadata inspection, which are limited against complex manipulations. ForgeryCheck stands out by integrating multiple forensic methods—including ELA, noise analysis, copy-move detection, and deep learning-based classification. This multi-pronged strategy enables detection of both simple edits and sophisticated deepfakes, offering a comprehensive assessment of media authenticity.

Model Performance: While many tools perform well on specific datasets, they often struggle with generalization. ForgeryCheck’s models are trained on diverse datasets, achieving robust accuracy rates, demonstrating strong generalizability across various manipulation styles and formats an advantage over narrowly focused solutions.

User Experience: Traditional forensic tools typically require technical expertise and often display raw or complex data. ForgeryCheck, by contrast, offers a user-friendly web interface that guides users through the process, presents clear confidence scores, and visually highlights tampered regions—making it accessible to both experts and non-experts.

Adaptability and Extensibility: Some tools are limited to image analysis or lack scalability as forgery techniques evolve. ForgeryCheck supports both image and video analysis and features a modular architecture, allowing seamless integration of new forensic algorithms and ensuring adaptability to emerging manipulation trends.

Privacy and Security: Unlike some platforms that store or share user-uploaded content, ForgeryCheck prioritizes privacy. Media is processed securely and deleted post-analysis unless the user opts to save results, a crucial consideration for sensitive or confidential content.

Limitations of Other Solutions: Many existing tools lack video support, struggle with large or high-resolution files, or offer outputs that are difficult to interpret. ForgeryCheck fills these gaps with support for varied file types and sizes, and delivers results that are both actionable and easy to understand.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

ForgeryCheck has demonstrated itself as a practical and effective solution for detecting digital media manipulation. By combining advanced forensic techniques such as Error Level Analysis, noise pattern examination, and deep learning models, the system is able to identify a wide range of forgeries in both images and videos. Its user-friendly interface ensures that even those without technical expertise can easily upload files and interpret results, making digital verification accessible to a broader audience.

Throughout the development and evaluation process, ForgeryCheck has shown strong performance in terms of accuracy and usability. The platform's ability to provide clear visual evidence and confidence scores empowers users to make informed decisions about the authenticity of digital content. Additionally, the system's focus on privacy and security helps build trust with users who are concerned about the handling of sensitive files.

While there are still challenges to address such as keeping pace with evolving forgery methods and optimizing performance for large files, ForgeryCheck lays a solid foundation for reliable and accessible media verification. Its comprehensive approach positions it as a valuable tool in the ongoing fight against misinformation and digital fraud.

7.2 FUTURE ENHANCEMENTS

Future enhancements for ForgeryCheck aim to broaden its functionality and improve user experience. Planned updates include support for tampering detection in PDF documents, extending the platform's reach beyond images and videos. A secure password recovery system using email or OTP verification will enhance account security and ease of access. To boost mobility, a dedicated mobile app is in development, enabling on-the-go forgery detection. Users will also gain the ability to store, track, and compare past analysis reports, creating a detailed verification history. Additionally, batch processing will be introduced, allowing multiple files such as images, videos, or PDFs, to be analyzed simultaneously for greater efficiency. Continuous improvements based on user feedback will ensure ForgeryCheck remains adaptive, effective, and user-centered.

REFERENCES

- Cozzolino, D., Poggi, G., & Verdoliva, L. (2015). Splicebuster: A new blind image splicing detector. *IEEE International Workshop on Information Forensics and Security (WIFS)*, 1–6. <https://doi.org/10.1109/WIFS.2015.7368561>
- Zhou, P., Han, X., Morariu, V. I., & Davis, L. S. (2018). Learning rich features for image manipulation detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1053–1061. <https://doi.org/10.1109/CVPR.2018.00117>
- Afchar, D., Nozick, V., Yamagishi, J., & Echizen, I. (2018). MesoNet: A compact facial video forgery detection network. *IEEE International Workshop on Information Forensics and Security (WIFS)*, 1–7. <https://doi.org/10.1109/WIFS.2018.8630761>
- Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). FaceForensics++: Learning to detect manipulated facial images. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1–11. <https://doi.org/10.1109/ICCV.2019.00010>
- Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., & Natarajan, P. (2019). Recurrent convolutional strategies for face manipulation detection in videos. *arXiv preprint arXiv:1905.00582*. <https://arxiv.org/abs/1905.00582>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- Nguyen, T. T., Nguyen, C. M., Nguyen, D. T., Nguyen, D. T., & Nahavandi, S. (2019). Deep learning for deepfakes creation and detection: A survey. *arXiv preprint arXiv:1909.11573*. <https://arxiv.org/abs/1909.11573>
- Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). FaceForensics++: Learning to detect manipulated facial images. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1–11. <https://doi.org/10.1109/ICCV.2019.00010>

APPENDICES

APPENDIX A: Image Forgery Detection Model Architecture

Layer (Type)	Output Shape	Number of Parameters
Conv2D (32 filters)	(124, 124, 32)	2,432
MaxPooling2D	(62, 62, 32)	0
Conv2D (64 filters)	(60, 60, 64)	18,496
MaxPooling2D	(30, 30, 64)	0
Conv2D (128 filters)	(28, 28, 128)	73,856
Conv2D (128 filters)	(26, 26, 128)	147,584
MaxPooling2D	(13, 13, 128)	0
Conv2D (256 filters)	(11, 11, 256)	295,168
Conv2D (256 filters)	(11, 11, 256)	590,080
Flatten	(30,976)	0
Dense (64 units)	(64)	1,982,528
Dropout	(64)	0
Dense (128 units)	(128)	8,320
Dropout	(128)	0
Dense (Output - 1 unit)	(1)	129
Total Parameters		3,118,593
Trainable params		3,118,593
Non-trainable params		0

Table 1: Model Architecture for Image Forgery Detection

This CNN architecture is tailored for image forgery detection, progressively extracting features through multiple Conv2D and MaxPooling layers. Starting with 32 filters, the model captures basic patterns, then deepens to 64, 128, and 256 filters to identify complex manipulations. Pooling layers reduce spatial dimensions, enhancing efficiency. After feature extraction, the Flatten layer transitions the data to Dense layers (64 and 128 units), enabling deep pattern learning. Dropout layers prevent overfitting, and the final Dense layer with 1 unit outputs the forgery classification.

APPENDIX B: Image Forgery Detection Model Training Log

EPOCH-WISE SUMMARY TABLE

Epoch	Train Accuracy	Validation Accuracy	Train Loss	Validation Loss
1	81.07%	82.40%	0.4346	0.3967
5	82.00%	83.20%	0.4221	0.3804
10	82.13%	84.53%	0.4105	0.3668
15	83.40%	85.47%	0.3923	0.3485
20	85.07%	86.40%	0.3758	0.3338
25	85.20%	86.67%	0.3509	0.3095
30	85.33%	86.93%	0.3231	0.2980
35	85.33%	86.67%	0.3200	0.3104
40	85.33%	86.40%	0.3164	0.3228
50	86.67%	88.00%	0.2977	0.2646

Table 2: Summary of Training Accuracy and Loss

The above table presents a condensed overview of the model’s training performance across 10 key epochs during the training phase. The data highlights a steady improvement in both training and validation accuracy, starting from 81.07% in the first epoch and reaching up to 86.67% by the 50th epoch. Simultaneously, the training and validation losses gradually decreased, indicating that the model was successfully learning the underlying patterns in the dataset while minimizing overfitting.

By epoch 50, the training loss reduced to 0.2977 and validation loss reached 0.2646, accompanied by an increase in validation accuracy to 88.00%, which reflects strong generalization capability on unseen data. This outcome implies that the model not only performed well on the training set but also maintained robustness and reliability when evaluated on separate validation samples.

The progression of these metrics is visually represented in *Figure 5*, which includes the training and validation loss and accuracy curves. These plots provide a clear illustration of the model’s learning trajectory and convergence over time.

APPENDIX C: Video Forgery Detection Model Architecture

Layer (Type)	Output Shape	Number of Parameters
Conv2D (conv4_block6_3)	(15, 20, 1024)	263,168
BatchNorm (conv4_block6_3)	(15, 20, 1024)	4,096
Add (conv4_block6_add)	(15, 20, 1024)	0
Activation (conv4_block6_out)	(15, 20, 1024)	0
Conv2D (conv5_block1_1)	(8, 10, 512)	524,800
BatchNorm (conv5_block1_1)	(8, 10, 512)	2,048
Conv2D (conv5_block1_2)	(8, 10, 512)	2,359,808
Conv2D (conv5_block1_0)	(8, 10, 2048)	2,099,200
Conv2D (conv5_block1_3)	(8, 10, 2048)	1,050,624
BatchNorm (conv5_block1_0)	(8, 10, 2048)	8,192
BatchNorm (conv5_block1_3)	(8, 10, 2048)	8,192
Add (conv5_block1_add)	(8, 10, 2048)	0
Activation (conv5_block1_out)	(8, 10, 2048)	0
Conv2D (conv5_block2_1)	(8, 10, 512)	1,049,088
BatchNorm (conv5_block2_1)	(8, 10, 512)	2,048
Conv2D (conv5_block2_2)	(8, 10, 512)	2,359,808
BatchNorm (conv5_block2_2)	(8, 10, 512)	2,048
Conv2D (conv5_block2_3)	(8, 10, 2048)	1,050,624
BatchNorm (conv5_block2_3)	(8, 10, 2048)	8,192
Add (conv5_block2_add)	(8, 10, 2048)	0
Activation (conv5_block2_out)	(8, 10, 2048)	0
Conv2D (conv5_block3_1)	(8, 10, 512)	1,049,088
BatchNorm (conv5_block3_1)	(8, 10, 512)	2,048
Conv2D (conv5_block3_2)	(8, 10, 512)	2,359,808
BatchNorm (conv5_block3_2)	(8, 10, 512)	2,048
Conv2D (conv5_block3_3)	(8, 10, 2048)	1,050,624
BatchNorm (conv5_block3_3)	(8, 10, 2048)	8,192
Add (conv5_block3_add)	(8, 10, 2048)	0
Activation (conv5_block3_out)	(8, 10, 2048)	0
Dropout	(8, 10, 2048)	0
GlobalAveragePooling2D	(2048)	0
Dense (64 units)	(64)	131,136
Dense (Output - 1 unit)	(1)	65
Total Parameters		23,718,913
Trainable Parameters		23,665,793
Non-trainable Parameters		53,120

Table 3: Model Architecture for Video Forgery Detection

APPENDIX D: Video Forgery Detection Model Training Log

EPOCH-WISE SUMMARY TABLE

Epoch	Train Accuracy	Validation Accuracy	Train Loss	Validation Loss
1	75.39%	79.29%	0.5070	0.4265
2	76.89%	79.71%	0.4442	0.4276
3	77.79%	77.29%	0.4215	0.4090
4	76.93%	77.71%	0.4117	0.4516
5	77.32%	78.14%	0.4058	0.4211

Table 4: Summary of Training Accuracy and Loss

The table above summarizes the model's training performance over five epochs. Starting from a training accuracy of 75.39% and validation accuracy of 79.29%, the model showed a consistent learning trend during the initial stages. By the fifth epoch, the training accuracy improved to 77.32%, and the validation accuracy reached 78.14%, reflecting modest but steady performance gains. Although the jump in accuracy wasn't dramatic, the incremental improvements suggest that the model was steadily learning meaningful patterns from the video data. These results indicate a relatively smooth training phase with no signs of severe overfitting or underfitting.

Loss values also followed a generally downward trend, with training loss decreasing from 0.5070 in epoch 1 to 0.4058 by epoch 5. While the validation loss showed some fluctuations likely due to the varying complexity and motion in video frames, it remained within a close range and did not diverge significantly. This consistency hints at a stable learning process and suggests that the model was able to generalize reasonably well across unseen video samples. The fluctuations in validation loss may also reflect natural noise in the video data rather than an actual decline in performance. Overall, the model demonstrates a strong baseline for further tuning or scaling.

APPENDIX E: Technology Stack

Category	Technology/Library	Purpose
Backend Framework	Django (Python)	Web application backend, server-side logic
Frontend	HTML, CSS, Bootstrap, JavaScript	User interface design and interactivity
Machine Learning	TensorFlow, Keras, scikit-learn, scikit-image	Model building, training, evaluation, and image-based ML
Image Processing	OpenCV, Pillow	Image manipulation, enhancement, and analysis
Data Handling	NumPy, Pandas	Numerical computation and data manipulation
Utility Libraries	TQDM, Hachoir, Streamlit	Progress visualization, file metadata extraction, rapid app prototyping

Table 5: Overview of Technologies and Libraries Utilized

APPENDIX F: Django Project Folder Structure

The project follows a modular directory structure to ensure clarity, scalability, and maintainability:

ForgeryCheck Directory: This is the main Django project folder containing core configuration files such as *settings.py*, *urls.py*, and *wsgi.py*. These files handle global settings, URL routing, and server deployment interfaces.

website App: This directory contains the main application logic, including Django views and the image forgery detection modules. It manages user requests, handles model predictions, and renders appropriate responses.

ml_models Directory: Stores pre-trained machine learning models used for forgery detection. These models are loaded at runtime and enable the system to analyze uploaded media.

media Directory: Used for storing uploaded files (images or videos) that are temporarily held for processing and result generation.

static and templates Directories: *static* includes CSS, JavaScript, and image assets, while *templates* holds the HTML files for rendering the web pages. These ensure a responsive and interactive user interface.

The following outlines the main folder and file structure of the ForgeryCheck Django project as implemented for this work:

```
ForgeryCheck/
├── manage.py
├── requirements.txt
├── Pipfile
├── db.sqlite3
├── ForgeryCheck/
│   ├── __init__.py
│   ├── asgi.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
├── ml_models/
│   ├── forgery_model_me.hdf5
│   ├── proposed_ela_50_casia_fidac.h5
│   └── segmenter_weights.h5
├── media/
│   └── (uploaded files)
├── static/
│   ├── assets/
│   ├── css/
│   ├── img/
│   └── js/
├── templates/
│   ├── authentication/
│   ├── detection/
│   ├── pages/
│   └── base.html
└── website/
    ├── ImageForgeryDetection/
    ├── VideoForgeryDetection/
    ├── admin.py
    ├── apps.py
    ├── forms.py
    ├── migrations/
    ├── models.py
    ├── tests.py
    ├── urls.py
    └── views.py
```

APPENDIX G: Source Code Snippets

This appendix presents selected source code excerpts from the ForgeryCheck project. These snippets illustrate key functionalities, including user registration, forgery detection logic, and the integration of machine learning models within the Django framework.

1. User Registration Form with Email Requirement (*forms.py*)

Adding an email field to the registration form ensures every user provides a valid contact method, improving security and communication. This code extends Django's built-in `UserCreationForm` by making the email mandatory. It simply adds the email field and updates the form metadata to include it. This way, users must enter their email when signing up, enhancing account verification and user management. It's a straightforward but important step for more reliable user registration.

```
from django import forms
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.models import User

class RegistrationForm(UserCreationForm):
    email = forms.EmailField(required=True) # Make email required

    class Meta(UserCreationForm.Meta):
        model = User
        fields = UserCreationForm.Meta.fields + ('email',)
```

2. Home Page View Function (*views.py*)

This basic view function handles requests to the home page, serving as the main entry point of the app. It uses Django's *render* shortcut to deliver the *index.html* template to users. Keeping it simple, this function ensures that when users visit the root URL, they see the home page content. It's a foundational piece that connects the URL routing to the frontend display. This setup makes navigation intuitive and user-friendly from the very start.

```
def index(request):
    return render(request, "index.html")
```

This simple Django view renders the home page of the application, serving as the entry point for users.

3. Image Forgery Prediction Using Deep Learning (*FakeImageDetector.py*)

This function loads a pre-trained model and predicts the authenticity of the uploaded image.

```
def predict_result(self, fname):
    model_path = os.path.join(PROJECT_ROOT, 'ml_models',
                              'proposed_ela_50_casia_fidac.h5')
    model = load_model(model_path)
    class_names = ['Forged', 'Authentic']
    test_image = self.prepare_image(fname)
    if test_image is None:
        return ("Error", "Could not prepare image for prediction")
    test_image = test_image.reshape(-1, 128, 128, 3)
    y_pred = model.predict(test_image)
    y_pred_class = int(round(y_pred[0][0]))
    prediction = class_names[y_pred_class]
    confidence = f'{{{(1 - y_pred[0][0]) if y_pred <= 0.5 else y_pred[0][0]}
* 100:0.2f}}'
    return (prediction, confidence)
```

3. Main Entry Point for Django Management (*manage.py*)

This is the main entry point for running Django administrative commands, such as starting the server or applying migrations.

```
def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE',
                          'ForgeryCheck.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
```

4. Initializing the Image Forgery Detection Class (*FakeImageDetector.py*)

This class prepares an image for analysis, converting it to the required format and size for the machine learning model.

```
class FID:

    def prepare_image(self, fname):

        image_size = (128, 128)

        temp_prepare_path = os.path.join(TEMP_MEDIA_PATH,
        'temp_prepare.jpg')

        ela_image = self.convert_to_ela_image(fname, 90, temp_prepare_path)

        if ela_image is None:

            print("Error: Failed to prepare ELA image.")

            return None

        return np.array(ela_image.resize(image_size)).flatten() / 255.0
```

5. Noise Analysis for Image Forgery Detection (*noise_variance.py*)

This function estimates the noise level in an image, which can help identify tampered regions.

```
def estimate_noise(I):

    H, W = I.shape

    M = [[1, -2, 1], [-2, 4, -2], [1, -2, 1]]

    sigma = np.sum(np.sum(np.absolute(signal.convolve2d(I, M))))

    sigma = sigma * math.sqrt(0.5 * math.pi) / (6 * (W-2) * (H-2))

    return sigma
```

6. Double JPEG Compression Detection (*double_jpeg_compression.py*)

This function detects whether an image has undergone double JPEG compression, a common sign of manipulation.

```
def detect(input):

    firstq = 30

    secondq = 40

    thres = 0.5

    image = cv2.imread(input)

    shape = image.shape

    if(k==3):

        if peak_count>=20: return True

        else: return False
```

7. Copy-Move Forgery Detection Using SIFT (*copy_move_sift.py*)

This class leverages the SIFT algorithm to identify similar regions within an image, helping uncover potential copy-move forgeries. By extracting keypoints and matching features, it highlights duplicated areas that may signal tampering. It's effective due to SIFT's robustness to scale and rotation changes.

```
class CopyMoveSIFT:
    resize_percentage = 100
    max_dist = 250

    def __init__(self, path, output_path=None):
        # ... (initialization and error handling code) ...

        self.keypoints_sift, self.descriptors =
sift.detectAndCompute(self.img_gray, None)

        # ... (matching and drawing code) ...
```

8. Video Forgery Detection Function (*detect_video.py*)

This function analyzes video files frame by frame to detect possible forgeries using a trained deep learning model. It first validates and extracts frames, then evaluates them to determine authenticity. The result indicates whether the video is forged or authentic, along with the count of forged frames.

```
def detect_video_forgery(vid_src):
    vid = []
    sumFrames = 0
    cap = cv2.VideoCapture(vid_src)
    if not cap.isOpened():
        print(f"Error: Could not open video file: {vid_src}")
        return {'result': 'Error', 'f_frames': 0, 'message': 'Could not
open video file.'}

    # ... (frame extraction and prediction code) ...

    if no_of_forged <=0:
        return {'result':'Authentic','f_frames':0}
    else:
        return {'result':'Forged','f_frames':no_of_forged}
```

9. Django Settings: Installed Apps and Database Configuration (*settings.py*)

This section of the *settings.py* file lists the Django applications that are currently active in the project. Alongside Django's built-in apps for authentication, admin, and sessions, it includes third-party packages like *crispy_forms* and *crispy_bootstrap5* to enhance form rendering and styling. The custom *website* app is also registered here, indicating it contains core features of the project. These apps work together to deliver both the frontend and backend functionality of the application.

Below the app configuration, the *DATABASES* section defines the default database used for the project. In this case, SQLite is chosen for its simplicity and ease of use, making it ideal for lightweight applications and development purposes. The database file is stored locally in the base directory of the project. This setup allows seamless data management and supports key features like user authentication, session storage, and content handling.

Additionally, SQLite requires no external dependencies, making deployment and testing straightforward for small-scale projects.

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'website',
    'crispy_forms',
    'crispy_bootstrap5',
]

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

10. Django Management Utility Script (*manage.py*)

This script serves as the entry point for various administrative commands in a Django project. It configures the environment settings and delegates commands like running the development server or applying migrations to Django's command-line interface. The `main()` function ensures that the appropriate settings module is loaded and attempts to execute commands, handling import errors if Django isn't properly installed. This file is automatically created when a Django project is initialized and is essential for managing project-level tasks.

```
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE',
                          'ForgeryCheck.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
```

The source code snippets compiled in *Appendix G* provide a practical foundation for understanding the core components of the video and image forgery detection system built using Django and deep learning techniques. Together, these scripts showcase not only the backend logic and structure but also the thoughtful integration of machine learning models into a web-based interface, reflecting a blend of practical implementation and purposeful design.

BIODATA



Name : Chipinapi Keerthi Sadha
Mobile Number : +91 70326 39808
Email : keerthisadha.21bce9540@vitapstudent.ac.in
Address : Manubolupadu, Dagadarthi, Nellore Dist,



Name : Goli Revanth Krishna
Mobile Number : +91 79954 20997
Email : revanth.21bce7852@vitapstudent.ac.in
Address : Visalakshi Nagar, Samalkot, Kakinada Dist,



Name : Pokuru Harshini
Mobile Number : +91 70329 06580
Email : harshini.21bce9512@vitapstudent.ac.in
Address : East Street, Gudur, Tirupati Dist, 524101