Image Forgery Detection using Deep Learning

Submitted in partial fulfilment of the

requirements of the degree of

Bachelor of Engineering (B.E.)

in

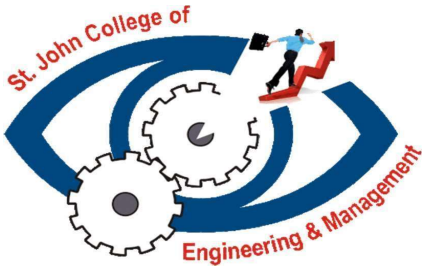
INFORMATION TECHNOLOGY

by

Yash Aman Kamble	**EU1214003**
Aditya Rakesh Pandey	**EU1214021**
Bhavesh Bheraram Kumavat	**EU1214047**

Under the guidance of

Mr. Manthan Surti



Department of Information Technology

St. John College of Engineering and Management, Palghar

(Autonomous)

University of Mumbai

2024-2025

CERTIFICATE

This is to certify that the B.E. project entitled "**Image Forgery Detection using Deep Learning"** is a bonafide work of "**Yash Aman Kamble" (EU1214003), "Aditya Rakesh Pandey" (EU1214021),** and **"Bhavesh Bheraram Kumavat" (EU1214047)** submitted to University of Mumbai in partial fulfilment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Information Technology"** during the academic year 2024-2025.

Mr. Manthan Surti

Project Guide

Dr. Arun Saxena

Head of Department

Dr. Kamal Shah

Principal

B.E. Project Report Approval

This project report entitled ***Image Forgery Detection using Machine Learning*** by ***Yash Aman Kamble, Aditya Rakesh Pandey,*** and ***Bhavesh Bheraram Kumavat*** is approved for the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

Examiners

1._____

2._____

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature
Yash Aman Kamble (EU1214003)

Signature
Aditya Rakesh Pandey (EU1214021)

Signature
Bhavesh Bheraram Kumavat (EU1214047)

Date:

Place:

Abstract

In the digital age, image manipulation has become increasingly accessible and sophisticated, posing significant risks to the authenticity and reliability of visual content in various domains, including journalism, law enforcement, social media, and digital forensics. The proliferation of advanced editing tools has made detecting forged images a critical challenge, as traditional manual methods are time-consuming and often ineffective in identifying subtle manipulations. This paper presents a comprehensive approach to image forgery detection using machine learning techniques, specifically Convolutional Neural Networks (CNN) and Error Level Analysis (ELA), to address these challenges.

CNNs are employed to automatically extract features from images by analyzing pixel-level inconsistencies, textures, and artifacts that often indicate manipulation. The deep learning capabilities of CNNs enable the model to detect complex and subtle image forgeries, such as changes in lighting, texture, and pixel blending. ELA, on the other hand, provides a complementary approach by highlighting areas of inconsistent compression, which typically suggest tampered regions. By identifying discrepancies in compression error levels, ELA allows the system to focus on potentially manipulated sections of the image.

The integration of CNN and ELA techniques ensures a robust and scalable solution that can detect a wide range of image manipulations, including splicing, copy-move forgery, and deepfakes. The system is designed to process large volumes of images in real-time, making it suitable for applications where timely detection is critical. Additionally, the project emphasizes the development of a user-friendly interface to facilitate easy deployment across different sectors. This research contributes to the broader effort to ensure the integrity of visual media and combat the spread of misinformation, offering an effective tool for real-world image forgery detection.

Keywords— *Image Forgery Detection, Convolutional Neural Networks (CNN), Error Level Analysis (ELA), Deep Learning, Splicing, Copy-Move Forgery, Deepfakes, Pixel-level Inconsistencies, Real-time Detection*

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CNN	Convolutional Neural Network
DFD	Data Flow Diagram
ELA	Error Level Analysis
CMFD	Copy-Move Forgery Detection
RSA	Reptile Search Algorithm
NASNet	Neural Architecture Search Network
XGBoost	Extreme Gradient Boosting
ResNet50v2	Residual Network 50 Version 2
YOLO	You Only Look Once
CASIA	Chinese Academy of Sciences Institute of Automation
SAE	Stacked Autoencoder
JPEG	Joint Photographic Experts Group
TIFF	Tagged Image File Format
GOA	Genetic Optimization Algorithm
RSADTL-CMPD	Random Sample and Adaptive Dissimilarity-based Two-Level Content Manipulation Detection
CM	Confusion Matrix
DCT	Discrete Cosine Transform
GB	Gradient Boosting
WBS	Work Breakdown Structure
VGG16	Visual Geometry Group 16
GAN	Generative Adversarial Network
API	Application Programming Interface
RAM	Random Access Memory
OS	Operating System

Chapter 1

Introduction

In today's digital world, the manipulation of visual content has become increasingly widespread and sophisticated, posing serious challenges to the authenticity of images shared across various platforms. With the advent of powerful image editing tools, anyone can easily alter or forge images, making it difficult to discern between genuine and manipulated content. This rise in image forgery has significant implications in areas such as journalism, where doctored images can mislead the public; law enforcement, where manipulated evidence can obstruct justice; and social media, where altered images contribute to the spread of misinformation.

Traditional methods of image forgery detection, such as manual inspection, are labor-intensive and often ineffective, especially when dealing with subtle and highly advanced manipulations. As image forgery techniques continue to evolve, there is an urgent need for automated, efficient, and scalable solutions that can reliably detect such forgeries in real-time.

This project aims to address these challenges by developing an image forgery detection system using machine learning techniques, particularly Convolutional Neural Networks (CNN) and Error Level Analysis (ELA). CNNs, widely recognized for their effectiveness in image processing tasks, are leveraged to extract key features from images, learning to recognize pixel-level anomalies and patterns indicative of manipulation. Meanwhile, ELA complements the CNN by detecting inconsistent compression levels across different regions of an image, a common sign of forgery. Together, these techniques form the backbone of a robust and accurate image forgery detection system.

The proposed system is designed to be highly scalable and capable of analyzing large volumes of images in real-time, making it suitable for a range of applications, from verifying the authenticity of media in news outlets to detecting manipulated content on social media platforms. Moreover, a user-friendly interface will be developed to facilitate easy access to the system, ensuring that it can be used by individuals and organizations with minimal technical expertise. By combining machine learning techniques with practical usability, this project aims to make significant contributions to the field of digital image forensics and play a role in combating the growing threat of misinformation in the digital era.

1.1 Motivation

In an era where digital media plays a critical role in shaping public opinion, information dissemination, and legal evidence, the authenticity of visual content is more important than ever. With the increasing availability of sophisticated image editing tools, the threat of manipulated media—whether through simple alterations or advanced techniques like deepfakes—has grown significantly. This manipulation can lead to misinformation, loss of public trust, and serious consequences for individuals, organizations, and society at large. The motivation for this project stems from the urgent need to develop an automated, scalable, and reliable system for detecting image forgeries. By utilizing machine learning, we aim to create a solution that not only enhances the accuracy and speed of forgery detection but also helps in maintaining the integrity of visual content. This project seeks to contribute to the broader effort of combating digital disinformation and ensuring that images, which are often used as evidence or trusted sources of information, remain authentic and credible.

1.2 Problem Statement

Through extensive research and analysis, we identified the following issues in current image forgery detection systems that this project will address:

- Difficulty in detecting sophisticated image manipulations such as deepfakes and complex forgeries.
- Inadequacy of traditional manual methods, which are time-consuming and prone to human error.
- Lack of scalability in existing solutions to handle the large volume of daily generated visual content.
- Limited accuracy of automated systems in detecting diverse types of image forgeries. Inability of current systems to provide real-time image forgery detection.

1.3 Objective

The objectives are as follows:

- Develop a robust and automated image forgery detection system using machine learning techniques.
- Accurately detect various types of image manipulations, including splicing, copy-move forgeries, and deepfakes.
- Ensure scalability and efficiency for processing and analyzing large volumes of images in real-time.
- Train machine learning models on a diverse dataset of authentic and manipulated images for real-world applicability.
- Create a user-friendly interface accessible to non-technical users in journalism, law enforcement, and social media.
- Contribute to the fight against digital misinformation by ensuring the integrity of visual content.

1.4 Scope

The scope of this project encompasses the development of an image forgery detection system using machine learning to identify and detect manipulated images. The project involves collecting and preprocessing a large dataset of both authentic and manipulated images, followed by designing and training machine learning models capable of identifying forgeries. Ensuring real-time scalability is a key focus, enabling the system to process large volumes of images efficiently. This system can be applied across multiple domains where the authenticity of visual content is crucial. In journalism, it can help verify the authenticity of images before publication, preventing the spread of manipulated news. In law enforcement, the system can assist in analyzing photographic evidence to ensure its integrity in investigations and court proceedings. For social media platforms, the system can be integrated to detect and flag manipulated images, helping reduce the dissemination of false information. Moreover, it can be used in digital forensics, media organizations, and academic research to safeguard the authenticity of visual data. By developing a user-friendly interface, the project ensures that users from various fields can easily access and benefit from the tool.

Chapter 2

Review of Literature

2.1 Modeling of Reptile Search Algorithm with Deep Learning Approach for Copy-Move Image Forgery Detection

In 2023, Mashael Maashi et al. introduced a novel Copy-Move Forgery Detection (CMFD) method using the Reptile Search Algorithm (RSA) combined with deep learning. The method utilizes NASNet for feature extraction and RSA for hyperparameter tuning, followed by XGBoost for classification. This approach enhances the detection of forged regions, outperforming other models when tested on benchmark datasets.

2.2 A Survey on Image Forgery Detection Using Different Forensic Approaches

In 2020, Akram Hatem Saber, Mohd Ayyub Khan, and Basim Galeb Mejbel reviewed various digital image forgery detection techniques, including active methods (e.g., digital watermarking) and passive methods (e.g., copy-move and splicing detection). The paper provides a comparative analysis of deep learning approaches such as convolutional neural networks (CNNs) in detecting tampered images. The study highlights the advantages and limitations of different forensic technologies and identifies areas for future research in improving detection algorithms.

2.3 Deep Learning-Based Digital Image Forgery Detection System

In 2022, Emad Ul Haq Qazi, Tanveer Zia, and Abdulrazaq Almorjan proposed a deep learning-based system using ResNet50v2 architecture and YOLO CNN for detecting splicing image forgeries. The model was tested on the CASIA_v1 and CASIA_v2 datasets, achieving 99.3% accuracy with transfer learning and 81% without transfer learning on CASIA_v2. This approach improves splicing detection accuracy by utilizing transfer learning with pre-trained models.

2.4 Image Region Forgery Detection: A Deep Learning Approach

In 2016, Ying Zhang et al. proposed a deep learning-based method to detect image region forgery across various formats. The two-stage approach uses a Stacked Autoencoder (SAE) for feature learning and contextual information integration to detect tampered regions. The method outperforms previous techniques on JPEG and TIFF images, achieving 91.09% accuracy on the CASIA dataset. This approach allows for more accurate tampered region localization, addressing limitations of previous methods that were format-specific or targeted only JPEG images.

2.5 Image Forgery Detection

Hany Farid, in 2009, conducted a survey on various image forgery detection techniques, categorizing them into pixel-based, format-based, camera-based, physically-based, and geometry-based methods. The paper discusses how advancements in digital manipulation tools have made forgeries easier to create, but forensic techniques have evolved to counter them. Farid acknowledges the ongoing challenge of creating undetectable forgeries, stating that while it's increasingly difficult, it is not impossible.

2.6 An Analysis of Image Forgery Detection Techniques

Chandandeep Kaur and Navdeep Kanwal, in 2019, reviewed various passive image forgery detection techniques. They focused on methods like copy-move, splicing, and retouching detection. The study highlighted that many existing techniques require human oversight and are often specific to particular forgery types. The authors emphasize the need for more automated, generalized detection systems and better differentiation between malicious tampering and benign retouching.

2.7 The Effect of Error Level Analysis on Image Forgery Detection Using Deep Learning

This paper explores the use of Error Level Analysis (ELA) as a preprocessing technique in conjunction with deep learning models to detect image forgery. The study concludes that ELA can enhance the performance of deep learning models in identifying subtle image manipulations. However, the effectiveness of ELA depends on the dataset and model quality, and there is still room for optimization of model architecture and dataset diversity for better generalization across different types of forgeries.

2.8 Detecting Image Forgery over Social Media Using U-NET with Grasshopper Optimization

In 2023, Niousha Ghannad and Kalpdrum Passi proposed an enhanced U-Net model, optimized by the Grasshopper Optimization Algorithm (GOA), to detect image forgery on social media platforms. The study focuses on improving segmentation accuracy and achieving high precision, recall, and F1 scores for detecting forgeries. Tested on the CASIA dataset, the model demonstrates superior performance in identifying image manipulations, such as splicing and copy-move forgery, making it suitable for real-time use in social media forensics.

Table 2.1 Literature Review

Sr. No.	Paper Title [Ref.]	Author names	Conclusion	Research Gaps
1	Modeling of Reptile Search Algorithm With Deep Learning Approach for Copy Move Image Forgery Detection [2023]	1. Mashael Maashi 2. Hayam Alamro 3. Heba Mohsen 4. Noha Negm	RSADTL-CMFD uses NASNet, RSA, and XGBoost for CM region detection, showing improved results. Future work involves hybrid DL methods.	1. Complexity 2. Resource Intensive 3. Dependency on pre-trained models
2	A Survey on Image Forgery Detection Using Different Techniques [2020]	1. Akram Hatem Saber 2. Mohd Ayyub Khan 3. Basim Galeb Mejbel	The survey highlighted the need for robust, efficient detection methods, noting trade-offs in complexity, accuracy, and robustness among techniques.	1. High complexity 2. High false positives 3. Heavy compression 4. Poor robustness
3	Deep Learning-Based Digital Image Forgery Detection System [2022]	1. Emad Ul Haq Qazi 2. Tanveer Zia 3. Abdulrazaq	Deep learning approach using ResNet50v2 and YOLO CNN achieved 99.30% accuracy on CASIA datasets for image forgery detection.	1. Dependency on Large Database 2. Computationally Intensive 3. Specificity to Splicing
4	Image Region Forgery Detection: A Deep Learning Approach [2016]	1. Ying Zhang 2. Jonathan Goh 3. Lei Lei Win 4. Vrizlynn Thing	Deep learning approach achieved 91.09% accuracy in detecting tampered images; future work involves DCT input and exploring new architectures.	1. Training Complexity 2. Imbalanced Data 3. Limited to Patch 4. Level Analysis.

Sr. No.	Paper Title [Ref.]	Author names	Conclusion	Research Gaps
5	Image Forgery Detection [2009]	1. Hany Farid	The paper surveys various image forgery detection techniques, categorizing them into pixel-based, format-based, camera-based, physically-based, and geometry-based methods.	1. Comprehensive Categorization 2. Application to Different Scenarios 3. Advancement in Forensics Tools
6	An Analysis of Image Forgery Detection Techniques [2019]	1. Chandandeep Kaur 2. Navdeep Kanwal	The paper reviews passive forgery detection methods and highlights the challenges of differentiating forgery types. It emphasizes the need for effective solutions due to the ease of digital manipulation.	1. Comprehensive Review 2. Dataset Coverage 3. Identification of Forgery Types
7	The Effect of Error Level Analysis on Image Forgery Detection Using Deep Learning [2019]	1. Wina Permana Sari 2. Hisyam Fahmi	The paper finds that using Error Level Analysis (ELA) as a preprocessing step can enhance deep learning-based forgery detection but is dependent on dataset quality and model.	1. Combines Traditional and Modern Techniques 2. Improved Detection of Subtle Forgeries 3. Potential for Automation
8	Detecting Image Forgery over Social Media Using U-NET with Grasshopper Optimization [2023]	1. Niousha Ghannad 2. Kalpdrum Passi	The paper presents an enhanced U-Net optimized with Grasshopper Optimization Algorithm (GOA) for better image forgery detection on social media, showing improved accuracy on the CASIA dataset	1. Improved Accuracy 2. Real-Time Usability 3. Versatility 4. Enhanced Model Architecture

Chapter 3

Requirements Gathering and Planning

3.1 Hardware Requirements

Operating System: Windows 10 Home Single Language

Processor: Intel(R) Core(™) i5-7200U CPU @ 2.50GHz 2.70 GHz

Hard Disk Storage: 1 TB

Installed RAM: 8.00 GB (7.87 GB usable)

3.2 Software Requirements

1. Google Colab
2. Different libraries
3. 64-bit Windows OS / macOS
4. Dataset: The dataset in the form of images
5. Python

3.3 Functional Requirements

1. Collect dataset from various sources.
2. Extracted data stored in .csv file format.
3. Process data using Python libraries.

3.4 Non-Functional Requirements

Python workspace should be always updated with all the libraries that are required for running the process. Appropriate dataset should be given as an input.

Chapter 4

Design

4.1 Dataflow Diagram (DFDs)

4.1.1 Level 0 DFD

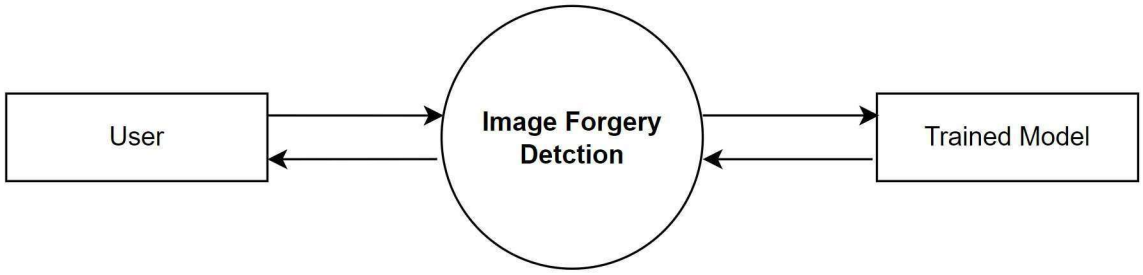


Figure 4.1.1 Level 0 DFD

4.1.2 Level 1 DFD

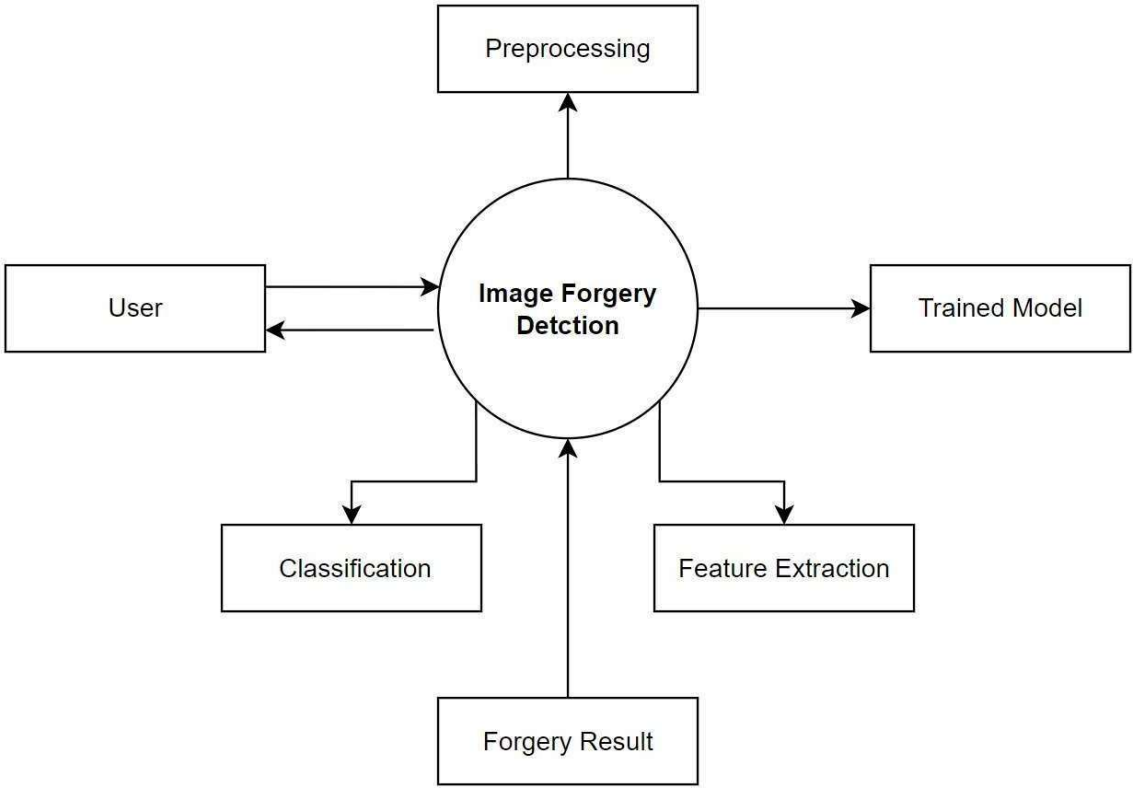


Figure 4.1.2 Level 1 DFD

4.1.3 Level 2 DFD

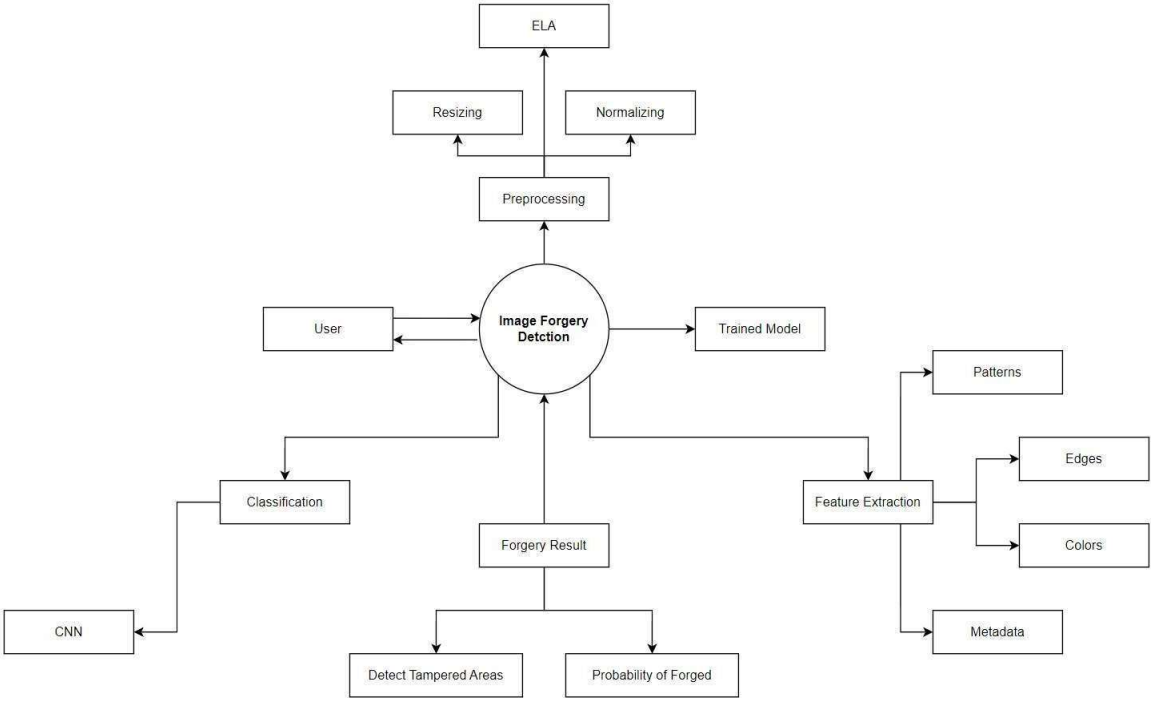


Fig 4.1.3 Level 2 DFD

All the above DFD diagrams represent the flow of data in an image forgery detection system. The system revolves around key entities like the user and the trained model, allowing them to interact with different modules such as preprocessing, feature extraction, classification, and forgery result generation. The system's main function is to analyze an uploaded image and return its authenticity by processing it through multiple stages. It utilizes a trained model for accurate detection, while various components handle tasks like detecting tampered areas and analyzing patterns and metadata.

4.2 UML Diagrams

4.2.1 Use Case Diagram

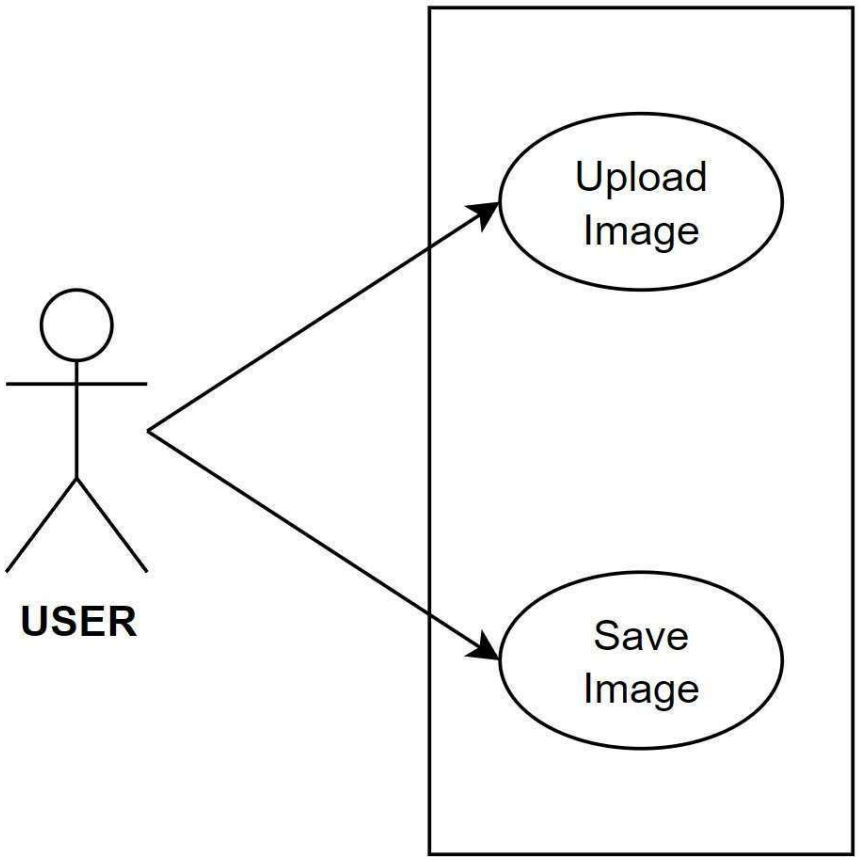


Fig 4.2.1 Use Case Diagram

The use case diagram illustrates the primary interactions between the user and the image forgery detection system. The user is able to perform two main actions: uploading an image for forgery detection and saving the processed image. These functions represent the core operations the system offers to the user, facilitating easy image analysis and management within the application. Through these features, the user can efficiently interact with the system, ensuring seamless image processing and storage.

4.2.2 Class Diagram

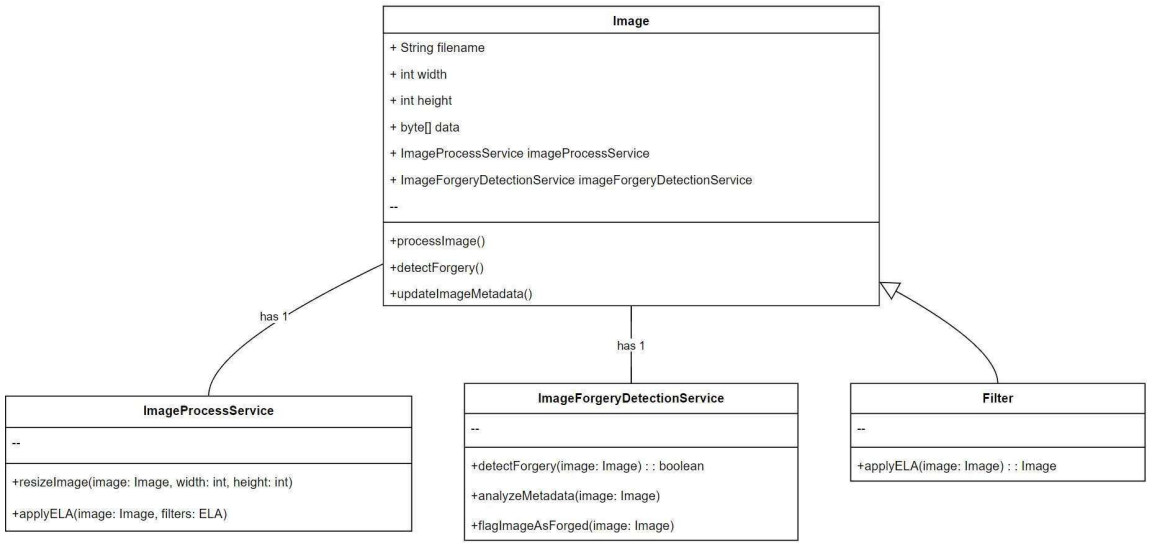


Fig 4.2.2 Class Diagram

The class diagram illustrates the data and features of the image forgery detection system. It consists of four entities, each with its own data and functions. The arrows represent relationships between the entities. For example, the `Image` class has a "has-a" relationship with both `ImageProcessService` and `ImageForgeryDetectionService`, meaning the `Image` class utilizes these services to process and detect forgery in images. The `ImageProcessService` handles resizing and applying ELA filters, while `ImageForgeryDetectionService` manages forgery detection, metadata analysis, and flagging forged images. The `Filter` class applies the ELA filter for further analysis.

4.2.3 Activity Diagram

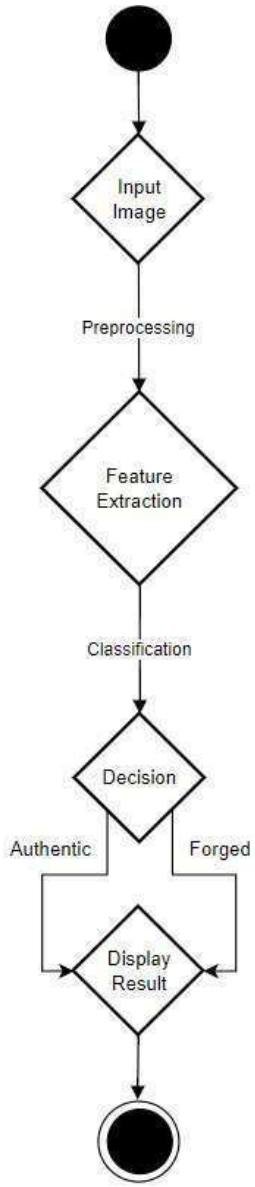


Fig 4.2.3 Activity Diagram

The activity diagram illustrates the flow of the image forgery detection process. It starts with an input image and proceeds through preprocessing, where the image is prepared for analysis. The next step is feature extraction, where key characteristics of the image are identified. This is followed by the classification stage, where the system makes a decision based on the extracted features. The decision block classifies the image as either authentic or forged. Finally, the result is displayed, and the process ends.

4.2.4 Sequence Diagram

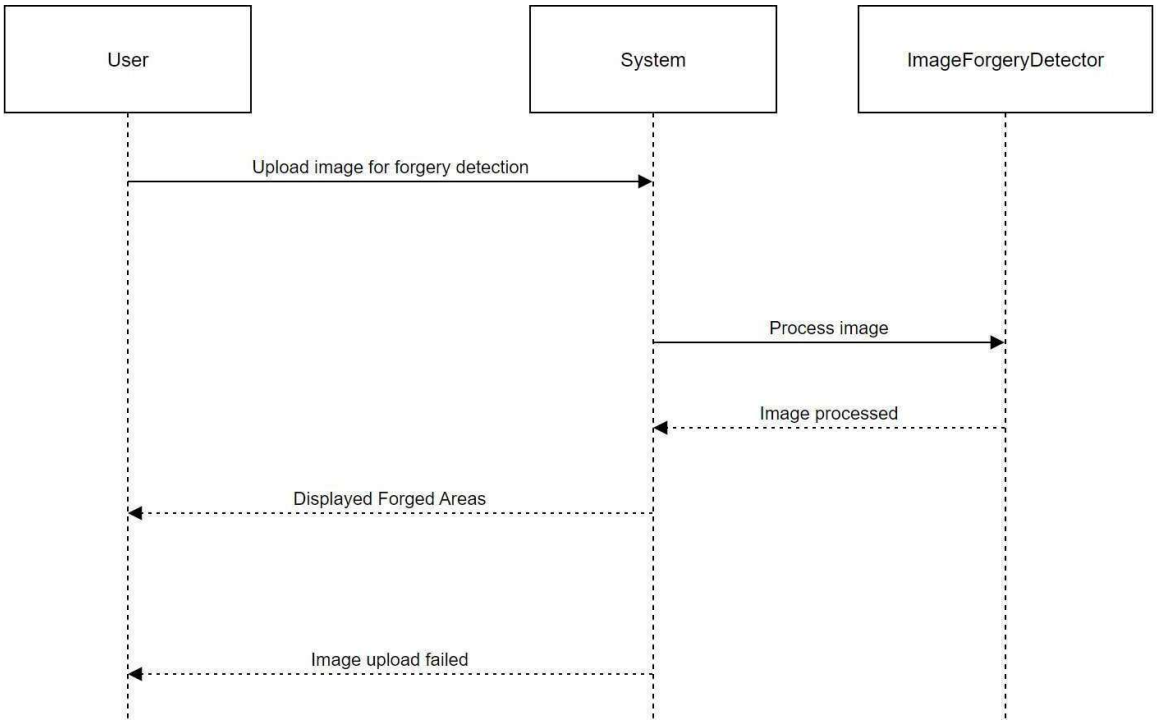


Fig. 4.2.4 Sequence Diagram

The above sequence diagram provides an overview of how the image forgery detection system operates. It demonstrates the interaction between the user, system, and the ImageForgeryDetector. Key functions include:

- **Upload image for forgery detection**: The user uploads an image to the system for analysis.
- **Process image**: The system forwards the image to the ImageForgeryDetector for processing.
- **Image processed**: The detector processes the image and sends the results back to the system.
- **Displayed Forged Areas**: The system displays the identified forged areas to the user if the image is manipulated.
- **Image upload failed**: The system informs the user if the upload fails.

4.2.5 Timeline / Gantt Chart

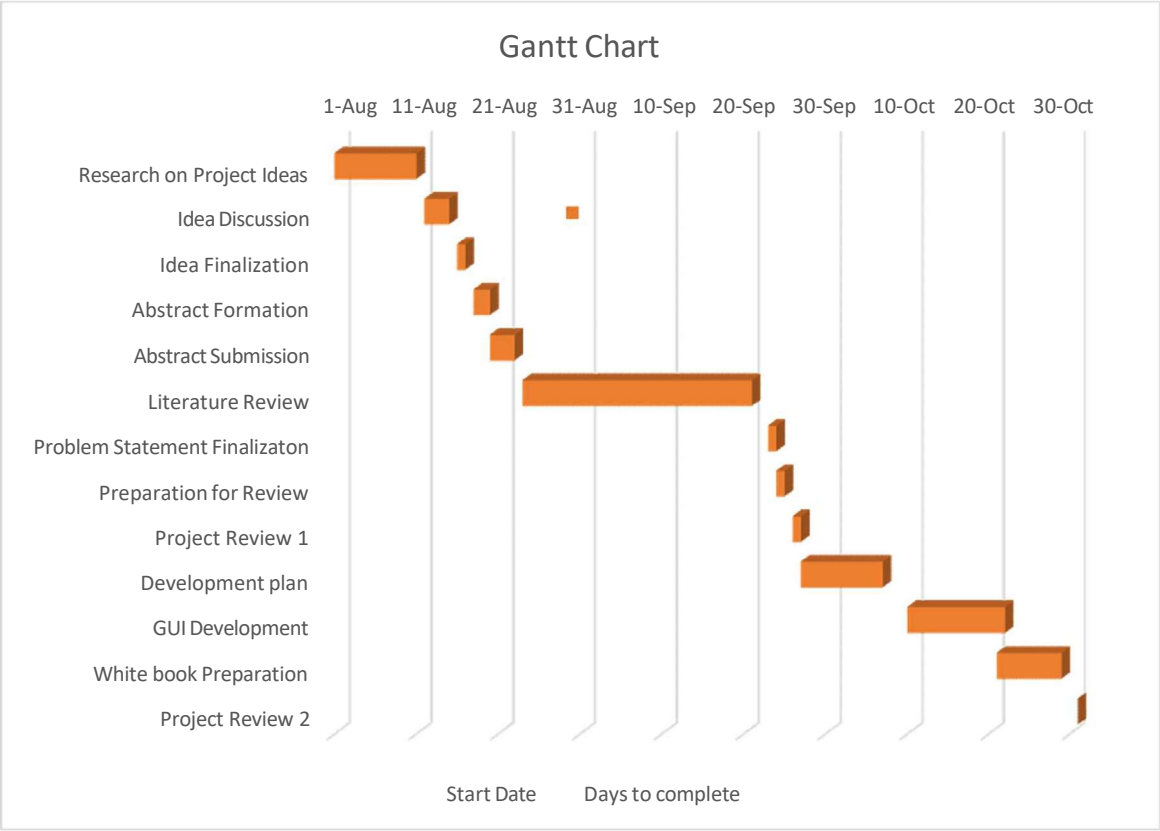


Fig 4.2.5 Gantt Chart

Table 4.2.5 Gantt Chart table

Task	Start Date	Days to complete
Research on Project Ideas	01-Aug	10
Idea Discussion	12-Aug	3
Idea Finalization	16-Aug	1
Abstract Formation	18-Aug	2
Abstract Submission	20-Aug	3
Literature Review	24-Aug	28
Problem Statement	23-Sep	1
Preparation for Review	24-Sep	1
Project Review 1	26-Sep	1
Development plan	27-Sep	10
GUI Development	10-Oct	12
White book Preparation	21-Oct	8
Project Review 2	31-Oct	1

4.2.6 Work Breakdown Structure (W.B.S) Chart

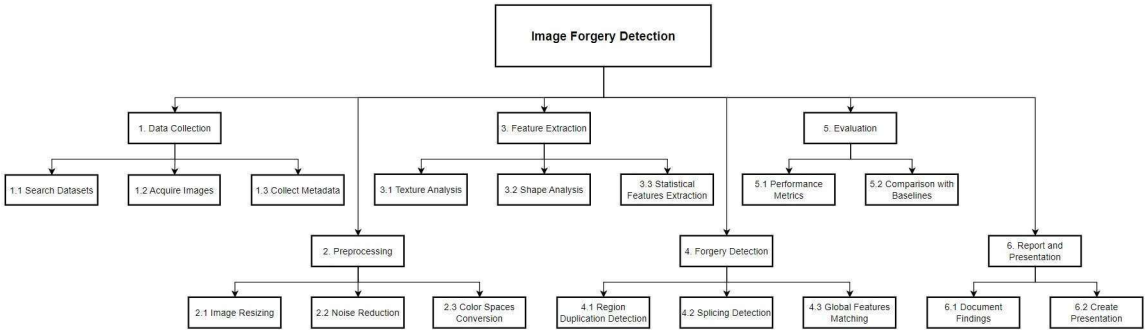


Fig 4.2.6 WBS Chart

The Work Breakdown Structure (WBS) chart above outlines the development process for the image forgery detection system. It starts with data collection, including searching for datasets, acquiring images, and collecting metadata. The next phase is preprocessing, involving steps like image resizing, noise reduction, and color space conversion. Following preprocessing, the feature extraction process begins, focusing on texture, shape, and statistical features. The forgery detection stage then identifies region duplication, splicing, and global feature matching. Evaluation comes next, assessing performance metrics and comparing with baselines. The final phase covers reporting and presentation, documenting findings and creating presentations to summarize the results.

Chapter 5

Report on Present Investigation

5.1 Methodology/Proposed System

The proposed system for this project is an automated image forgery detection platform that will utilize machine learning techniques for front-end processing and Python-based deep learning models for backend analysis. This system will enable users to upload images for forgery detection, where the platform will analyze the image using a combination of techniques such as error level analysis (ELA), statistical feature extraction, and texture analysis. The system will detect various types of image manipulations, including splicing, copy-move forgeries, and deepfakes. The platform will provide real-time feedback by displaying forged areas if detected. Additionally, a user-friendly interface will be developed to ensure ease of use for non-technical users. The system will also include features for image resizing, noise reduction, and color space conversions as part of the preprocessing stage, making it a robust tool for image authenticity verification across various sectors, such as journalism, law enforcement, and social media.

5.1.1 Architecture/Block Diagram of System

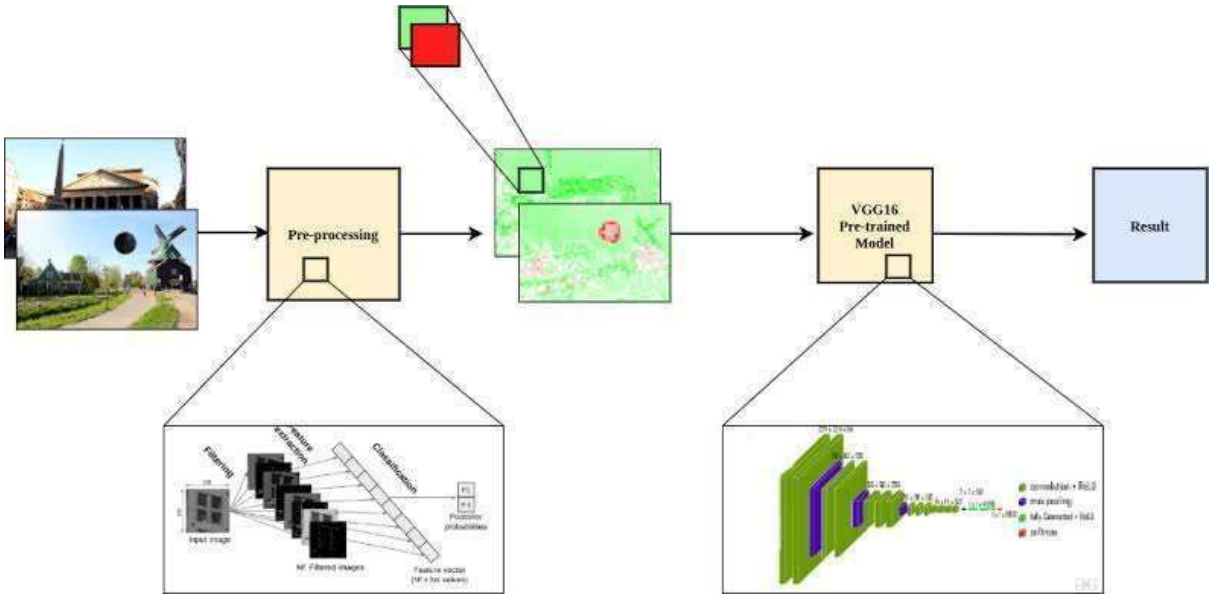


Fig 4.1.1 Block Diagram of Proposed System

The above diagram illustrates an image detection process using a pre-trained VGG16 model. It begins with an input image that undergoes pre-processing to enhance or filter regions of interest. The pre-processed image is then passed through the VGG16 model, which extracts features using convolutional layers. The output is a result that identifies or classifies objects or anomalies in the image. This approach leverages deep learning for accurate detection and classification.

5.2 Implementation

1. The prediction is carried out by using a dataset consisting of both authentic and forged images.
2. The dataset includes categories such as splicing, copy-move, and tampered images, along with original (authentic) samples.
3. The dataset is loaded, preprocessed, and balanced to ensure fair training and evaluation of the model.
4. Initially, all images are resized to 128x128 pixels and normalized for consistency and compatibility with the CNN model architecture.
5. In the earlier phase of experimentation, we also apply Error Level Analysis (ELA) to introduce compression-based pixel-level inconsistencies.
6. The ELA images help the model detect forgeries more effectively by enhancing subtle visual artifacts introduced during image manipulation.
7. For performance evaluation, we experiment with multiple CNN-based architectures including traditional CNN, CNN with varying ELA percentages (10%, 50%, 90%, and 100%), and standard deep learning models like VGG-16, VGG-19, DenseNet-121, ResNet-50, and Xception.
8. The dataset is divided into training, validation, and testing sets in a 76:19:5 ratio.
9. We build both the CNN and ELA-enhanced CNN models using TensorFlow and train them using the Adam optimizer with early stopping based on validation accuracy.
10. The model evaluation is based on accuracy, precision, recall, and F1-score.
11. After comparing across all models, the CNN with 100% ELA (our proposed system) achieves the best overall performance.
12. The system is designed to work in real time and is tested on unseen forged samples to simulate real-world usage.
13. The results clearly show that the inclusion of ELA provides a significant advantage in highlighting forgery regions that are otherwise hard to detect by the human eye or standard deep learning models.
14. We also maintain two versions of the dataset—one containing raw image data and the other containing preprocessed image vectors—to allow flexibility in experimentation and deployment.
15. Since this is a supervised problem with clearly labeled authentic and forged images, we use binary classification.
16. However, in future work, the system can be extended for unsupervised clustering of forged content in cases where labeled data is unavailable.

5.2.1 Algorithm / Flowchart

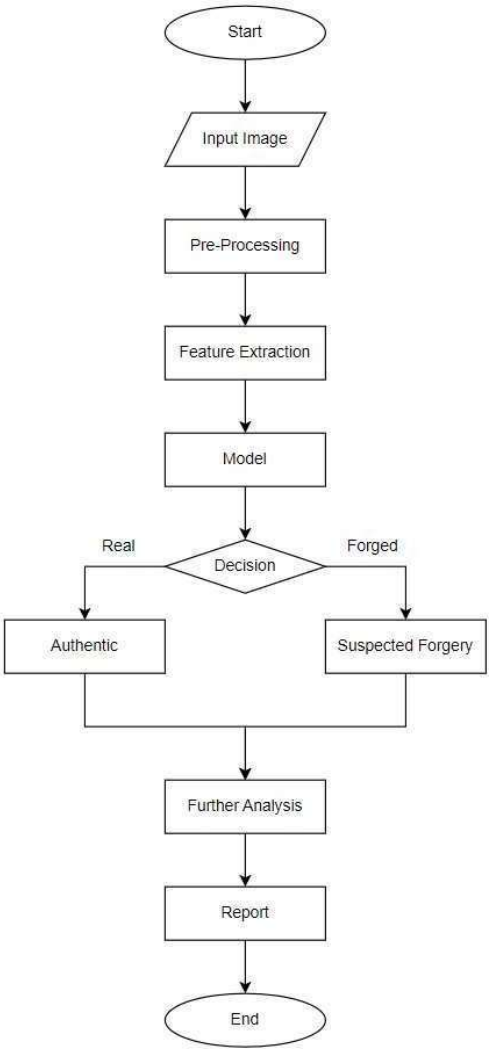


Fig 5.2.1 Flowchart

The flowchart illustrates the logical process of the image forgery detection system from the user's perspective. The flow begins with the input of an image by the user. The image then undergoes pre-processing, followed by feature extraction to identify relevant patterns. These features are then passed to the model for analysis. The system makes a decision based on the model's prediction, classifying the image as either authentic or suspected forgery. If the image is identified as forged, further analysis is conducted to gather more details. Finally, a report is generated, summarizing the results before the process concludes.

5.2.2 Dataset

Table 5.2.2 Dataset

Authentic	Forged
               	               

5.2.3 GUI

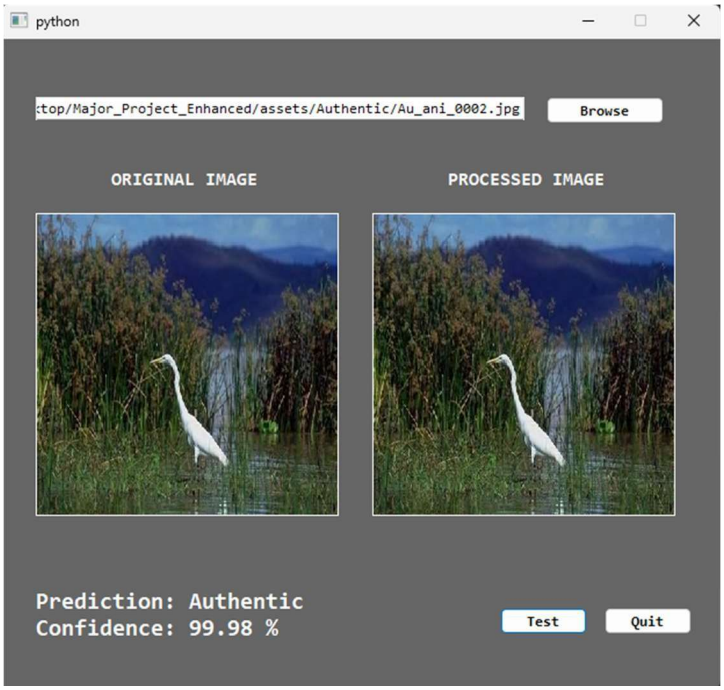


Fig 5.2.3.1 GUI for Image Forgery Detection using ELA (Authentic)

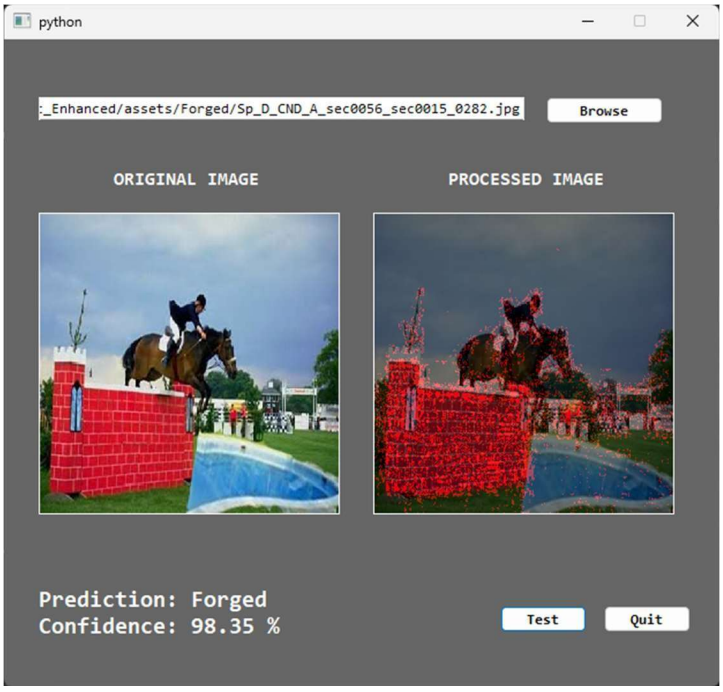


Fig 5.2.3.2 GUI for Image Forgery Detection using ELA (Forged)

This GUI allows users to upload an image and analyze it for tampering using Error Level Analysis (ELA). It displays the original and processed images side by side, along with a prediction of authenticity or forged and confidence level.

Chapter 6

Results and Discussion

6.1 Mean Pixel Intensity per Channel

Figure 4.2.3(a) Mean Pixel Intensity per Channel displays the mean pixel intensity per color channel (Red, Green, and Blue) in an Error Level Analysis (ELA). It highlights that the blue channel has the highest mean intensity, followed by red and then green. This analysis can be used in image forensics to detect inconsistencies in an image's compression or tampering.

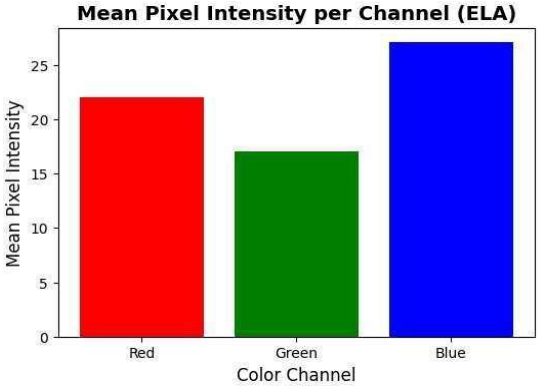


Figure 6.1 Mean Pixel Intensity per Channel

6.2 Authentic vs Forged (Bar Chart)

Figure 4.2.3(b) Authentic vs Forged (Bar Chart) shows a comparison between two categories: "Authentic" and "Forged." The chart indicates that there are more instances of forged images (over 800) compared to authentic ones (around 750). This visualizes the distribution of the two categories in a dataset, which can be relevant for tasks like image forgery detection.

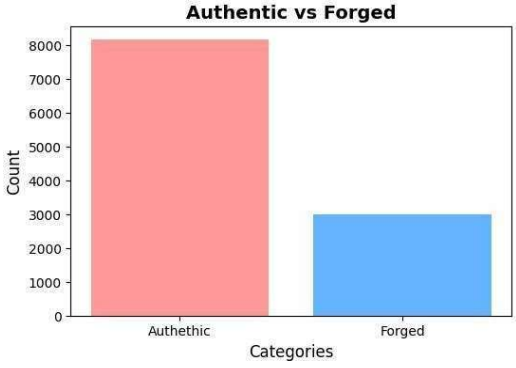


Figure 6.2 Authentic vs Forged (Bar Chart)

6.3 Authentic vs Forged (Pie Chart)

Figure 4.2.3(c) Authentic vs Forged (Pie Chart) displays the distribution between authentic and forged data. It shows that 53.8% of the data is categorized as forged, while 46.2% is categorized as authentic, giving a clear visual representation of the data distribution.

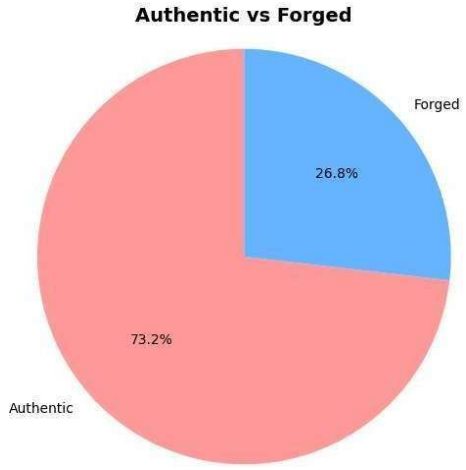


Figure 6.3 Authentic vs Forged (Pie Chart)

6.4 Training Loss vs Validation Loss

Figure 4.2.3(d) Training Loss vs Validation Loss shows the loss trend over 41 epochs for both training and validation. The training loss, represented by the blue line, steadily decreases as the epochs progress, indicating that the model is learning and improving. The validation loss, depicted by the red line, generally follows the same decreasing trend but has more fluctuations, suggesting occasional overfitting or instability in the model's performance on unseen data.

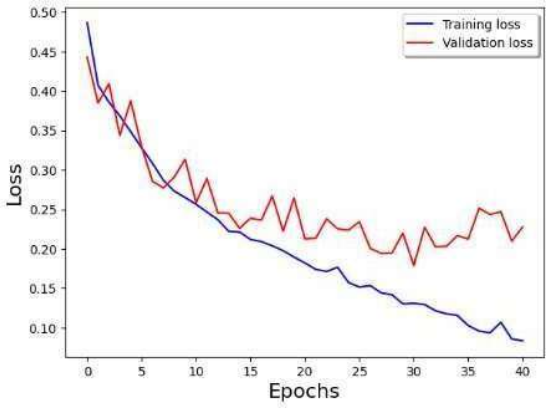


Figure 6.4 Training Loss vs Validation Loss

6.5 Training Loss vs Validation Loss

Figure 4.2.3(e) Training Accuracy vs Validation Accuracy represents the accuracy over the same 41 epochs for training and validation. The blue line (training accuracy) shows a gradual increase, meaning the model is progressively getting better at classifying the training data. The red line (validation accuracy) is more erratic but shows a general upward trend, indicating that the model is improving its performance on the validation set, although there are occasional drops. These fluctuations can be a sign of model variability when evaluated on new data.

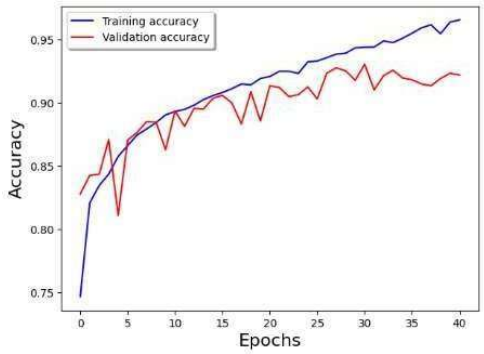


Figure 6.5 Training Accuracy vs Validation Accuracy

6.6 Confusion Matrix

Figure 4.2.3(f) Confusion Matrix shows the performance of an image classification model. It correctly identified 500 forged images and 1450 authentic ones. However, 80 authentic images were misclassified as forged, and 85 forged images were misclassified as authentic. The model's accuracy for each class is displayed as percentages, with more errors in classifying authentic images.

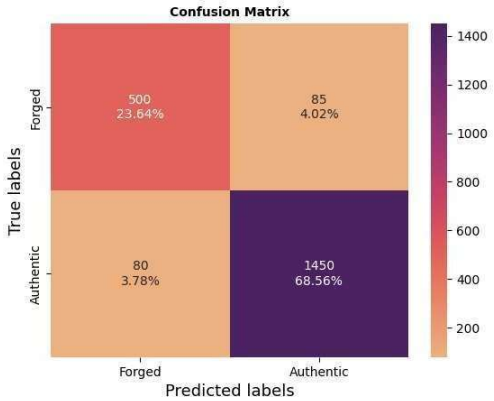


Figure 6.6 Confusion Matrix

Chapter 7

Conclusion and Future Work

This project developed an advanced image forgery detection system using Convolutional Neural Networks (CNN) and Error Level Analysis (ELA) to identify pixel-level inconsistencies and compression anomalies in manipulated images. With high accuracy and real-time processing capabilities, the system is scalable and suitable for sectors like journalism, law enforcement, social media, and digital forensics, where image authenticity is critical. Its user-friendly interface ensures accessibility to non-experts. The system provides an effective solution to combat digital misinformation and protect the integrity of visual content in today's digital landscape.

References

[1] N. Krishnaraj, B. Sivakumar, R. Kuppusamy, Y. Teekaraman, and A. R. Thelkar, ‘‘Design of automated deep learning-based fusion model for copy-move image forgery detection,’’ Comput. Intell. Neurosci., vol. 2022, pp. 1–13, Jan. 2022.

[2] Y. Abdalla, M. T. Iqbal, and M. Shehata, ‘‘Copy-move forgery detection and localization using a generative adversarial network and convolutional neural-network,’’ Information, vol. 10, no. 9, p. 286, Sep. 2019.

[3] Yaqi Liu, and Xianfeng Zhao, "Constrained Image Splicing Detection and Localization With Attention-Aware Encoder-Decoder and Atrous Convolution" IEEE Access2020.

[4] Yuan Rao, Jiangqun Ni, and Huimin Zhao, "Deep Learning Local Descriptor for Image Splicing Detection and Localization" IEEE Access2020.

[5] Zhang, Y.; Goh, J.; Win, L.L.; Thing, V. Image Region Forgery Detection: A Deep Learning Approach. In Proceedings of the Singapore Cyber-Security Conference (SG-CRC), Singapore, 14–15 January 2016; IOS Press: Singapore, 2016.

[6] Chen, J.; Kang, X.; Liu, Y.; Wang, Z.J. Median Filtering Forensics Based on Convolutional Neural Networks. IEEE Signal Process. Lett. 2015, 22, 1849–1853.

[7] W. Wang, J. Dong, and T. Tan. Exploring dct coefficient quantization effects for local tampering detection. Information Forensics and Security, IEEE Transactions on, 9(10):1653–1666, Oct 2014.

[8] F. Zach, C. Riess, and E. Angelopoulou. Automated image forgery detection through classification of jpeg ghosts. Pattern Recognition, 7476:185–194, January 2012.

[9] Doegar, A.; Dutta, M.; Gaurav, K. CNN Based Image Forgery Detection Using Pre-trained AlexNet Model. Int. J. Comput. Intell. IoT 2019, 2, 1.

[10] Wu, Y.; Abd-Almageed, W.; Natarajan, P. BusterNet: Detecting Copy-Move Image Forgery with Source/Target Localization. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.

[11] N. Krawetz, "A Picture's Worth... Digital Image Analysis and Forensics," 2007. Accessed: Sep. 28, 2020.

[12] H. Fahmi and W. P. Sari, "Effectiveness of Deep Learning Architecture for Pixel-Based Image Forgery Detection," Apr. 2021, pp. 302-307.

[13] W. Wang and J. Dong, CASIA v1.0, Tampered Image Evaluation Database, Available: http://forensics.idealtest.org/casiav1/, [Accessed: 29-May-2018]

[14] W. Wang and J. Dong, CASIA v2.0, Tampered Image Evaluation Database, Available: http://forensics.idealtest.org/casiav2/, [Accessed: 29-May-2018]

[15] A.C. Gallagher, "Detection of linear and cubic interpolation in jpeg compressed images," in Proc. 2nd Canadian Conf. Computer and Robot Vision., Victoria, British Columbia, Canada, vol. 171, 2005, pp. 65–72.

[16] H. Farid, "Digital ballistics from jpeg quantization: A followup study," Dept. Comp. Sci., Dartmouth College, Tech. Rep. TR2008-638, 2008.

Appendix

Technologies Used

1. Python

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

2. NumPy

NumPy is a library for the Python programming language that supports large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on them. It is essential for numerical operations in deep learning and forms the core of many other scientific libraries.

3. Tensorflow

TensorFlow is an open-source deep learning framework developed by Google. It is widely used for building machine learning models, particularly neural networks. In your project, TensorFlow powers the backend for Keras and is used for defining the CNN architecture, compiling the model, and performing training and evaluation.

4. Keras

Keras is a high-level neural networks API that runs on top of TensorFlow. It simplifies the process of building and training deep learning models. Your project uses Keras to define

layers like Conv2D, MaxPool2D, Dense, and activation functions, and also uses its model training utilities and callbacks like EarlyStopping.

5. **Matplotlib**

NumPy is a library for the Python programming language that supports large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on them. It is essential for numerical operations in deep learning and forms the core of many other scientific libraries.

6. **Seaborn**

Seaborn is a statistical data visualization library built on top of Matplotlib. It provides attractive and informative plots. In this project, it may be used to enhance confusion matrices or correlation heatmaps for better model interpretability.

7. **h5py**

h5py is a Pythonic interface to the HDF5 binary data format. It allows storing large numerical datasets efficiently. In this project, it is used to save and load processed image datasets (X and Y) in HDF5 format to reduce RAM usage and enable batch processing.

8. **PIL (Pillow)**

Pillow is a fork of the original Python Imaging Library (PIL). It provides extensive file format support and image processing capabilities. Your project uses it for resizing images, converting image modes, applying Error Level Analysis (ELA), and saving manipulated images.

9. **tqdm**

tqdm is a Python library that displays smart progress bars in loops. It is used in this project to track image preprocessing and loading progress, which is particularly helpful when working with large datasets.

10. **Scikit-learn**

Scikit-learn is a powerful machine learning library in Python that provides tools for data splitting, classification, regression, clustering, and model evaluation. Your project uses it for splitting datasets (train_test_split) and evaluating performance (classification_report, confusion_matrix, roc_curve, and auc).

11. **gc (Garbage Collection)**

The gc module is part of Python's standard library and is used to manage memory by manually collecting garbage. It helps free up unused memory, especially when working with large datasets or models.

12. **os**

    The os module provides a way to interact with the operating system. In your project, it is used to navigate directories, read image paths, and perform file-based operations.

13. **json**

    The json module is used for parsing and writing JSON data. While not a major component of model training, it may be used for configuration or logging.

14. **itertools**

    itertools is a Python module that provides building blocks for efficient looping. It may be used for creating iterators or complex data pipeline components.

15. **collections (Counter)**

    Counter from the collections module is used to count hashable objects. In this project, it may help in tracking label distribution or class frequencies.

Publications

1. Shraddha S. More, Vivian Brian Lobo, Anita Chaudhari, Aditya Pandey, Bhavesh Kumavat, Yash Kamble. “Enhancing Image Forgery Detection with Convolutional Neural Networks and Error Level Analysis.” Journal of Information Systems Engineering and Management, vol. 10, no. 27s, Feb. 2025, pp. 980-993.

Acknowledgement

We owe our deepest gratitude and regards towards the ones who offered their valuable guidance in the hour of need. We thank our guide **Mr. Manthan Surti**, (Department of Information Technology, St. John College of Engineering and Management) for her guidance and precious insights. Her useful comments and feedbacks during the discussions we had and the encouragement to question every technical detail that we came across while partially completing this B.E. project helped us to a great extent.

We also take the opportunity to thank **Dr. Arun Saxena** (Head of Department) who have always rendered their support and assistance in the best possible way.

We would also like to thank **Dr. Kamal Shah** (Principal, St. John College of Engineering and Management) and the **members of Aldel Education Trust** who have given us the background to conduct this B.E project.