FORGERYLENS: Unmarking Image and Video Forgeries

Thejaswini C S[1], Abhivarsha[2], Apoorva G C[3], Darshan H A[4], Rohith Y[5]

[1]Principal Researcher, Computing Sciences, ISE, VVIET, VTU Belagavi

email: thejanaveen.vviet@gmail.com

[2]Department of Information Science and Engineering, VVIET, VTU Belagavi

email: abhivarsha03@gmail.com

[3] Department of Information Science and Engineering, VVIET, VTU Belagavi

email: apoorvagc4@gmail.com

[4] Department of Information Science and Engineering, VVIET, VTU Belagavi

email: darshdarshan079@gmail.com

[5] Department of Information Science and Engineering, VVIET, VTU Belagavi

email: rohithbabu297@gmail.com

Abstract *—FORGERYLENS: Unmarking Image and video forgeries" In today's digital world, seeing is no longer believing. With the rapid advancement of AI and editing tools, it's easier than ever to manipulate images and videos, spreading misinformation, eroding trust, and even rewriting reality. FORGERYLENS is a powerful step forward in the fight against digital deception. This project aims to uncover the hidden fingerprints of tampering in both photos and videos, offering a way to detect and expose forgeries that often go unnoticed by the human eye. Rather than just spotting obvious edits, FORGERYLENS dives deep analyzing subtle inconsistencies, unnatural lighting, digital noise patterns, and other clues left behind during manipulation. Whether it's a deep-fake video or a subtly altered image, the system is designed to peel back the layers and reveal the truth beneath. Our approach combines advanced machine learning with a user-friendly lens—bringing forensic-level analysis into the hands of journalists, researchers, and everyday people who simply want to know what's real. In a time where digital trust is under siege, FORGERYLENS offers clarity, accountability, and the hope of restoring confidence in the visuals we see every day.*

KEYWORDS — FORGERYLWNS, Deepfake detection, Forgery detection, Splicing, Copy-Move Forgery, *digital deception*

1. PROBLEM DEFINITION

With the rise of powerful editing tools and AI-generated content like deepfakes, it has become increasingly difficult to trust the authenticity of images and videos shared online. Fake visual content can be used to spread misinformation, defame individuals, or manipulate public opinion, often with serious consequences. This project aims to address the growing concern by building an intelligent system that can automatically detect whether an image or video has been tampered with. By combining digital forensic techniques and deep learning models, the system will help users identify forgeries and highlight the manipulated areas, making it easier to verify the truth in a world full of digital deception.

2. INTRODUCTION

In today's digital age, the spread of manipulated videos, especially deepfakes, poses a significant challenge to discerning truth from fiction. These forgeries can be highly realistic, making it difficult to detect them with the naked eye. To combat this, researchers have developed automated systems for moving visual forgery detection. One such system leverages user-uploaded videos and analyses them frame by frame. Features are extracted from each frame, essentially capturing fingerprints of the visual content. These features are then compared against a pre-trained deep learning model,

like VGG16, which has been trained to recognize patterns indicative of manipulation. This approach offers a powerful tool for identifying potential forgeries and ensuring the authenticity of moving visuals.

This project aims to develop an intelligent system for detecting image and video forgeries using deep learning and digital forensic techniques. It focuses on identifying common manipulations such as copy-move, splicing, frame insertion/deletion, and deepfakes by leveraging datasets like CASIA and FaceForensics++. The system uses convolutional neural networks (CNNs), recurrent models (LSTM), and advanced architectures like XceptionNet to analyze spatial and temporal inconsistencies. Traditional forensic methods such as Error Level Analysis and metadata inspection are also integrated to enhance reliability. The end goal is to create an accurate, automated tool that can verify the authenticity of visual media and highlight tampered regions, with potential deployment through a simple GUI or web interface.

Enhance trust in digital media: By automatically detecting forgeries, the system aims to make users more confident in the authenticity of the videos they encounter online. Combat the spread of misinformation: Deepfakes and other manipulated videos can be used to spread false information. This project aims to identify such forgeries and prevent their misuse. Provide a tool for content verification: The system can be used by journalists, social media platforms, and individuals to verify the authenticity of videos before sharing them. Advance the field of digital forensics: This project contributes to the development of more sophisticated methods for detecting forgeries and staying ahead of evolving manipulation techniques.

3. LITERATURE SURVEY

1. AltFreezing for More General Video Face Forgery Detection

This paper introduces "AltFreezing," a novel training strategy aimed at enhancing the generalization capabilities of face forgery detection models. Traditional models often focus on either spatial artifacts (like blending) or temporal artifacts (such as flickering), leading to performance degradation when encountering unseen manipulations. AltFreezing addresses this by alternately freezing spatial and temporal weights during training, encouraging the model to learn both types of features effectively. The approach utilizes a spatiotemporal network divided into spatial-related and temporal-related weight groups. By alternately freezing these groups, the model avoids overfitting to one type of artifact. Additionally, various video-level data augmentation methods are introduced ton further improve generalization. Experiments demonstrate that this framework outperforms existing methods in detecting unseen manipulations across different datasets.

2. On Learning Multi-Modal Forgery Representation for Diffusion Generated Video Detection

The authors propose MM-Det, an innovative algorithm designed to detect diffusion-generated videos, which pose significant challenges due to their diverse semantics. MM-Det leverages Large Multi-modal Models (LMMs) to generate a Multi-Modal Forgery Representation (MMFR), enhancing the model's ability to detect unseen forgery content. The approach incorporates an In-and-Across Frame Attention (IAFA) mechanism for spatio-temporal feature augmentation and employs a dynamic fusion strategy to refine forgery representations. To support this research, the authors construct the Diffusion Video Forensics (DVF) dataset, encompassing a wide range of forgery videos. MM-Det achieves state-of-the-art performance on DVF, demonstrating its effectiveness in detecting complex, diffusion-generated video forgeries.

3. Hierarchical Fine-Grained Image Forgery Detection and Localization

This work presents a hierarchical fine-grained approach to image forgery detection and localization (IFDL). Recognizing the significant differences in forgery attributes between CNN-synthesized and image-editing domains, the authors propose representing forgery attributes with multiple labels at different levels. The model performs fine-grained classification at these levels, leveraging their hierarchical dependencies to learn comprehensive features. The proposed IFDL framework comprises a multi-branch feature extractor, along with localization and classification modules. Each branch focuses on classifying forgery attributes at a specific level, while the localization module segments pixel-level forgery regions. The authors also construct a hierarchical fine-grained dataset to facilitate their study. Evaluations on seven different benchmarks demonstrate the method's effectiveness in both detection and localization tasks.

4. Multiple Forgery Detection in Digital Video Based on Inconsistency in Video Quality Assessment Attributes
This study addresses the detection of multiple forgeries in digital videos by analyzing inconsistencies in video quality assessment (VQA) attributes. The authors focus on identifying tampered regions resulting from frame deletions or insertions. The methodology involves extracting key frames and evaluating quality metrics to detect anomalies indicative of tampering. Unlike deep learning approaches, this method offers a lightweight alternative by relying on VQA attributes, making it suitable for scenarios with limited computational resources. The approach demonstrates effectiveness in detecting various types of video forgeries, including multiple frame deletions and insertions.

5. A Hybrid Approach of Traditional Block-Based Deep Learning for Video Forgery
The authors propose a two-stage hybrid method that combines traditional block-based analysis with convolutional neural networks (CNNs) to detect video forgeries. This approach capitalizes on the strengths of both techniques: block-based methods excel at examining local artifacts, while CNNs are adept at learning high-level features. The hybrid model first performs meticulous spatial artifact examination and then applies deep learning for feature extraction and classification. The method achieves an accuracy of 79.31% and an F1-Score of 65.87%, outperforming existing methods. It proves robust against various types of video forgeries, such as face swapping, face reenactment, and splicing, offering a comprehensive solution for video forgery detection.

4. METHODOLOGY

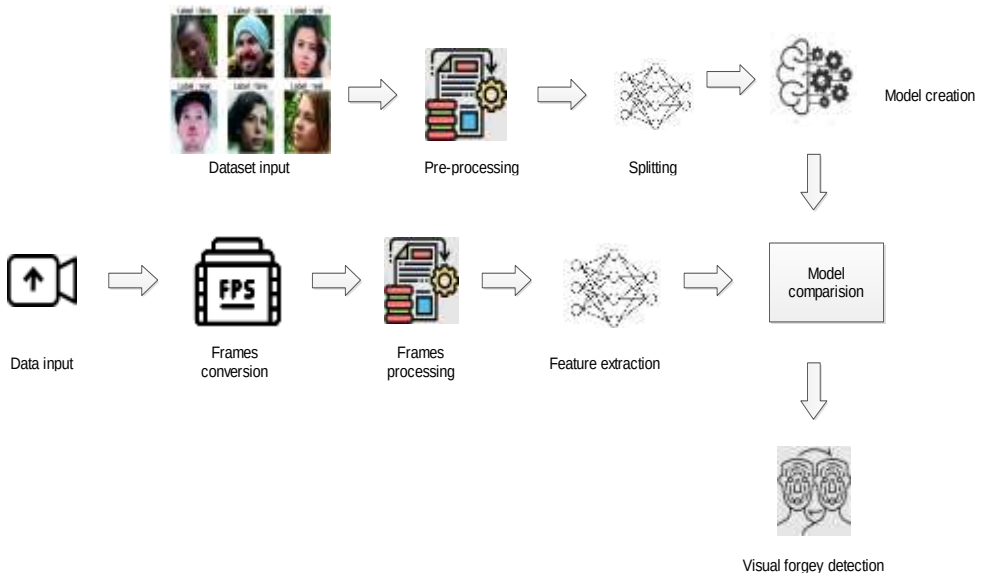

Fig.4.1. Architecture Diagram

Our proposed system tackles moving visual forgery detection, specifically deepfakes. Users upload videos through a user-friendly interface. The system then breaks down the video into individual frames. From each frame, it extracts a rich set of features that capture not only visual details but also how those details change between frames. These features might include colors, textures, motion patterns, and potentially how light interacts with objects in the scene. A deep learning model, like a Convolutional Neural Network (CNN), analyzes the features. This model has been trained on a massive dataset of real and manipulated videos, allowing it to recognize the subtle inconsistencies that often indicate forgery. We might start with a pre-trained model like VGG16, but the system could explore fine-tuning this model or even trying different architectures. Finally, the model generates a classification output, essentially saying whether the video is likely real or fake. Additionally, the system might provide confidence scores or highlight suspicious regions within the video for further investigation.

User Interface (UI): The UI acts as the entry point, allowing users to register, login, and upload videos for analysis. It should be user-friendly and guide users through the process.

Video Pre-processing: This component handles uploaded videos, segmenting them into individual frames for further analysis.

Feature Extraction: From each frame, this component extracts a rich set of features that capture visual details (colors, textures), temporal information (motion patterns), and potentially even how light interacts with objects (optical flow).

Deep Learning Model: This core component is a pre-trained Convolutional Neural Network (CNN) like VGG16, potentially fine-tuned for this specific task. The model analyses the extracted features and learns to identify patterns indicative of forgeries, especially deepfakes.

Detection and Classification: The model generates a classification output (real/fake) based on its analysis of the features. Additionally, it might provide a confidence score or highlight suspicious regions within the video for further investigation.

Database: The system would likely utilize a database to store user information, potentially anonymized video thumbnails for reference, and the deep learning model itself. Security measures are crucial for this component.

Output Interface: This component presents the results to the user. It should clearly display the classification (real/fake), any confidence scores, and potential visualizations of suspicious regions.

4.1. Data-Flow Diagram

DFD graphically represents the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expanding it to a hierarchy of detailed diagrams. DFD has often been used due to the following reasons:

- Logical information flow of the system
- Determination of physical system construction requirements
- Simplicity of notation
- Establishment of manual and automated systems requirements.

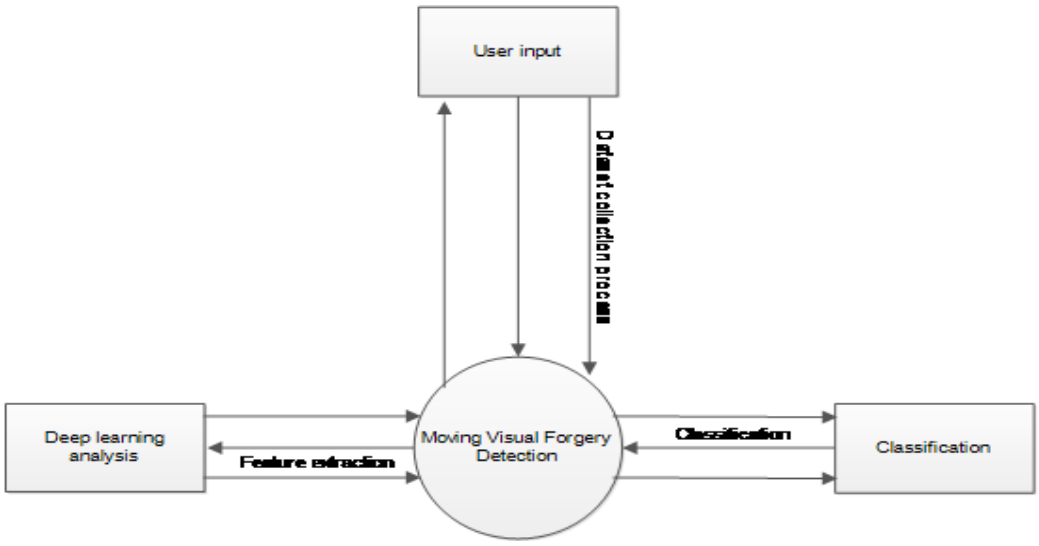

Fig.4.1.1 Level 0 Data Flow Diagram

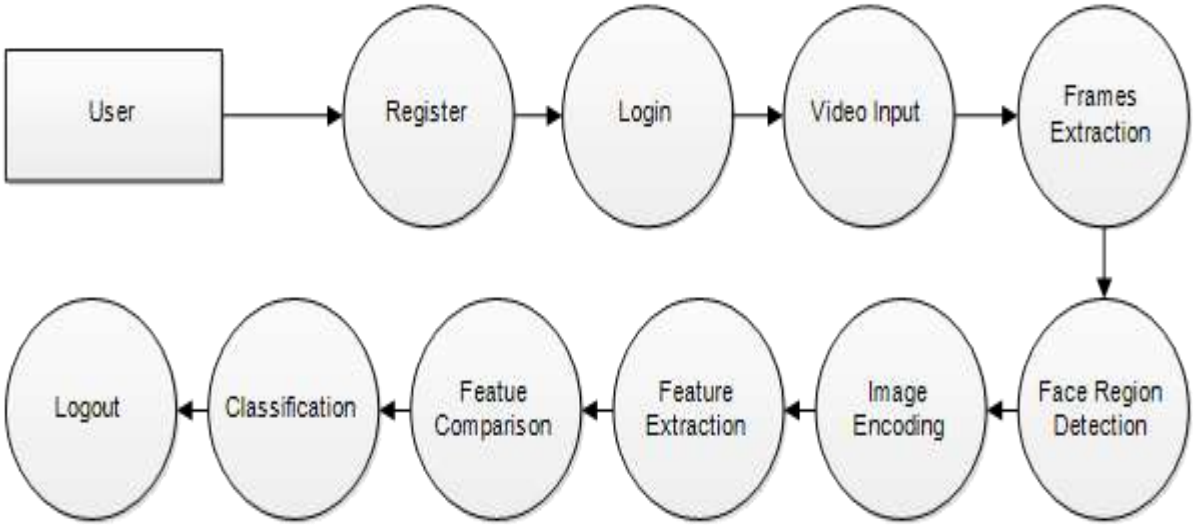

Fig.4.1.2 Level 1 Data Flow Diagram – User

4.2 Use Case Diagrams

Use case diagram is a graph of actors, a hard and fast of use instances enclosed by means of a device boundary, conversation associations among the actor and the use case. The use case diagram describes how a gadget interacts with out of doors actors; each use case represents a bit of functionality that a machine provides to its users. A use case is called an ellipse containing the call of the use case and an actor is shown as a stick figure with the call of the actor beneath the parent.

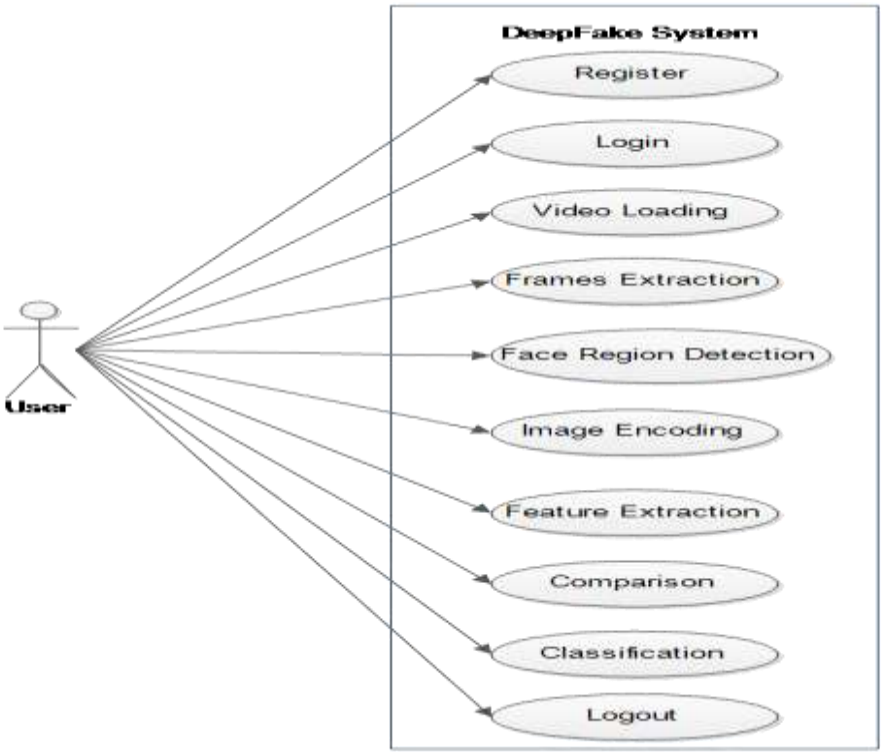

Fig 4.2.1 Use case diagram for User

4.3. Sequence Diagrams

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are sometimes called event diagrams, event scenarios.

UML sequence diagrams are used to represent or model the flow of messages, events and actions between the objects or components of a system. Time is represented in the vertical direction showing the sequence of interactions of the header elements, which are displayed horizontally at the top of the diagram Sequence Diagrams are used primarily to design, document and validate the architecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task or scenario. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior.

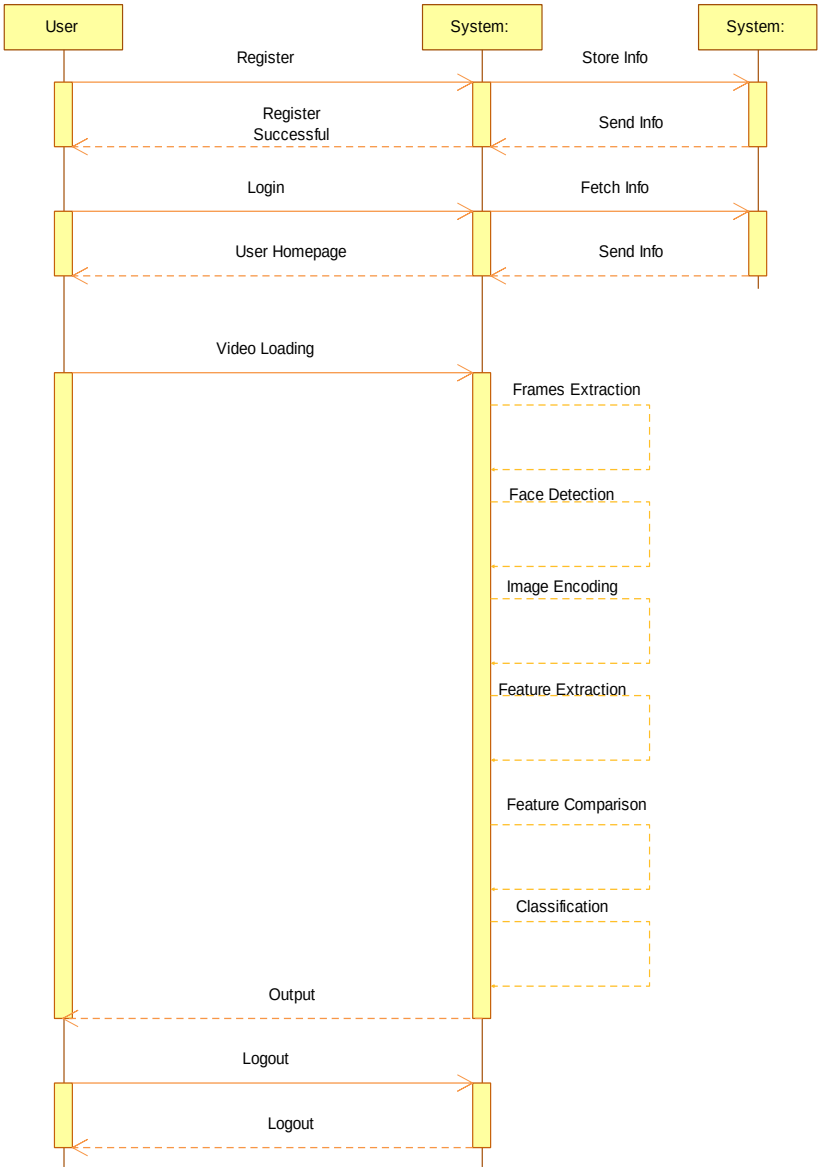

Fig.4.3.1 Sequence diagram for User

5. IMPLIMENTATION

5.1. System Implementation

There are three major types of implementations are there, but the following are proposed for the project.

5.1.1. Parallel Conversion type of Implementation:

In this type of implementation both the current system and the proposed system run in parallel. This happens till the user gets complete confidence on the proposed system and hence cuts of the current system.

5.1.2 Phase - in method of implementation

In this type of implementation, the proposed system is introduced phase-by-phase. This reduces the risk of uncertainty about the proposed system.

Each program is tested individually at the time of development using the data and has verified that this program linked together in the way specified in the programs specification, the computer system and its environment is tested to the satisfaction of the user. The system that has been developed is accepted and proved to be satisfactory for the user. And so the system is going to be implemented very soon. A simple operating procedure is included so that the user can understand the different functions clearly and quickly.

Initially as a first step the executable form of the application is to be created and loaded in the common server machine which is accessible to the entire user and the server is to be connected to a network. The final stage is to document the entire system which provides components and the operating procedures of the system.

Implementation is the stage of the project when theoretical design is turned into a working system. Thus it can be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing methods to achieve changeover and evaluation of changeover methods.

Implementation is the process of converting a new system design into operation. It is the phase that focuses on user training, site preparation and file conversion for installing a candidate system. The important factor that should be considered here is that the conversion should not disrupt the functioning of the organization.
This application is executed utilizing python programming language. Article situated in writing a stepwise program to implement the application. Each program needs to store inside the memory and the capacities can be in any formats.
Highlights of the object-oriented worldview:

Emphasis is based on the given information may opposes to methods. Computer programs are divided into different items. Product is partitioned according to the data structures. Methods will work according to the design of the system. Objects of the system will relate to one another techniques. Different strategies are used to improve the product or application. According to the plan, application or product will be generated or designed. Datasets are used within the given capacities.

5.2. VGG16

Input: The process starts with an input image. For VGG16, this image is typically resized to a standard size (e.g., 224x224 pixels) to ensure compatibility with the network architecture.

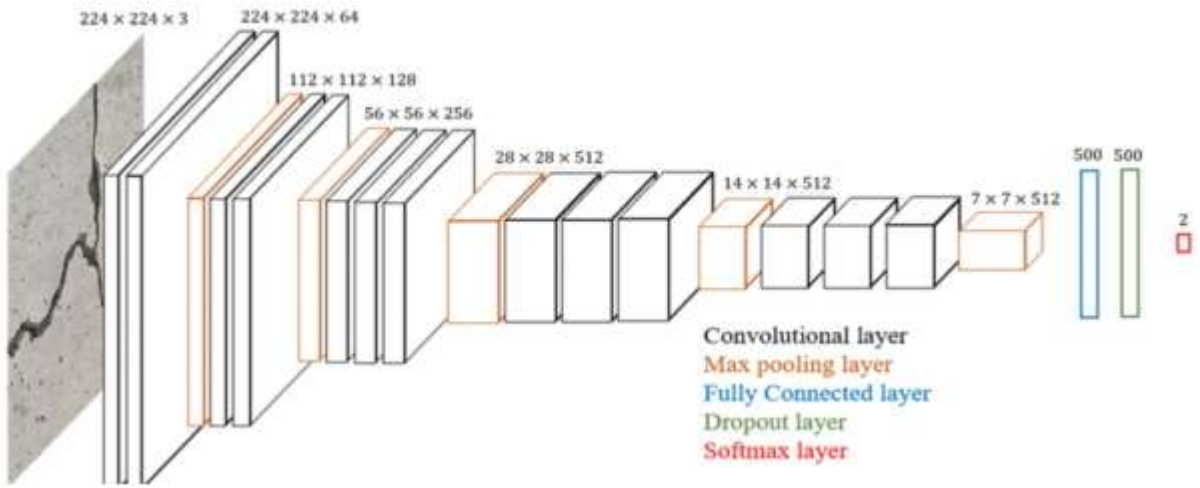

Fig.5.2.1. VGG16

Convolutional Layers: The core of VGG16 lies in its convolutional layers. These layers apply filters (small matrices) that scan the image, detecting patterns and features like edges, colors, and shapes. VGG16 uses multiple convolutional layers stacked together. Each layer learns its own set of filters, progressively extracting more complex features from the previous layer's outputs.

ReLU Activation: After each convolutional layer, a ReLU (Rectified Linear Unit) activation function is applied. This function essentially removes negative values from the output, introducing non-linearity and allowing the network to learn more complex relationships between features.

Pooling Layers: Pooling layers are inserted between convolutional layers to reduce the image's dimensionality and introduce some level of translation invariance (meaning the network can recognize the same feature even if it appears in a slightly different location). VGG16 uses max pooling, which takes the maximum value from a small grid of pixels in the previous layer's output.

Fully Connected Layers: After the convolutional and pooling stages, the network transitions to fully connected layers. These layers flatten the remaining activations from the previous layers into a single long vector. Fully connected layers act like perceptrons, where each neuron receives input from all neurons in the previous layer and applies a weighted sum with an activation function (often ReLU) to generate its output. There are typically multiple fully connected layers, allowing the network to learn increasingly complex combinations of features.

Output Layer: The final layer in VGG16 is the output layer, containing a number of neurons equal to the number of image classes the network is trained to classify (e.g., 1000 classes for ImageNet). Each neuron in the output layer represents a specific class. The network applies a SoftMax activation function to the output layer, which normalizes the outputs into probabilities between 0 and 1, indicating the likelihood of the input image belonging to each class. The class with the highest probability is chosen as the predicted classification.

6. RESULTS AND SNAPSHOTS

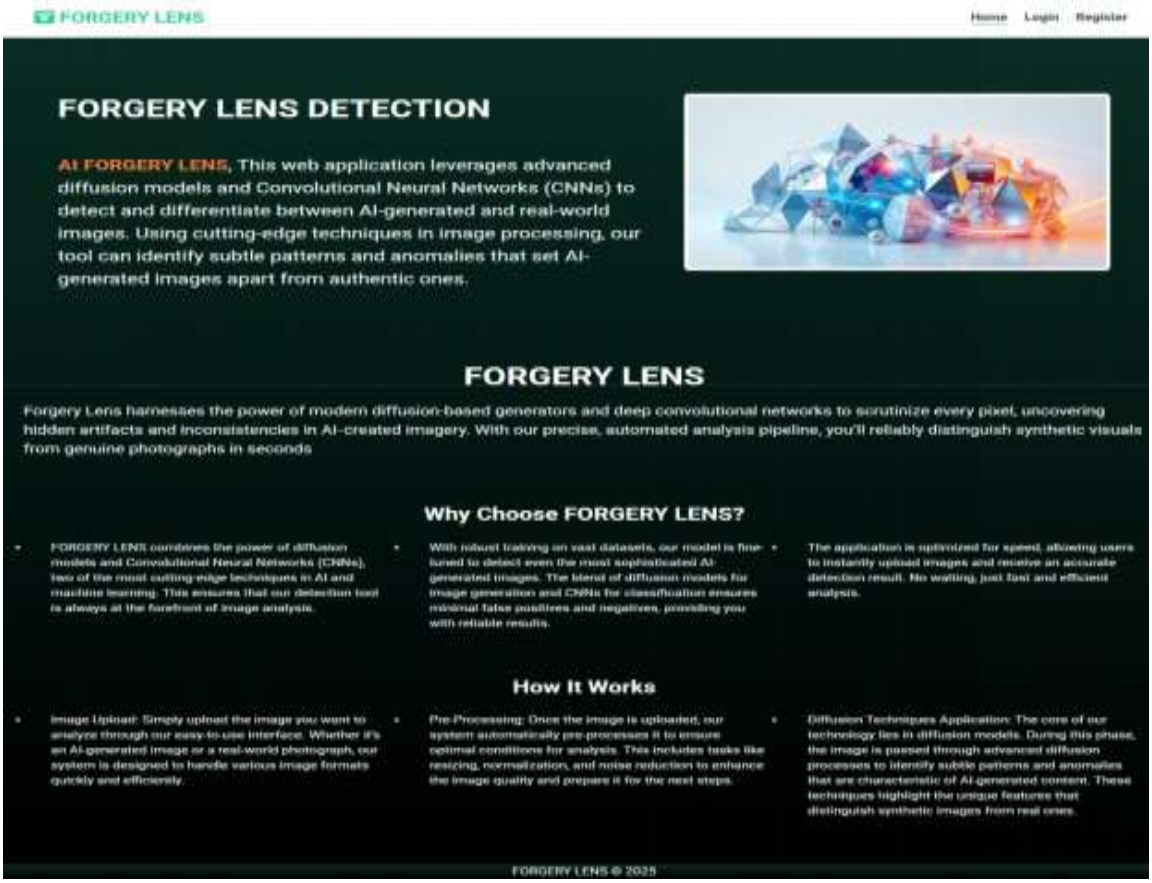

Fig: 6.1 Home Page

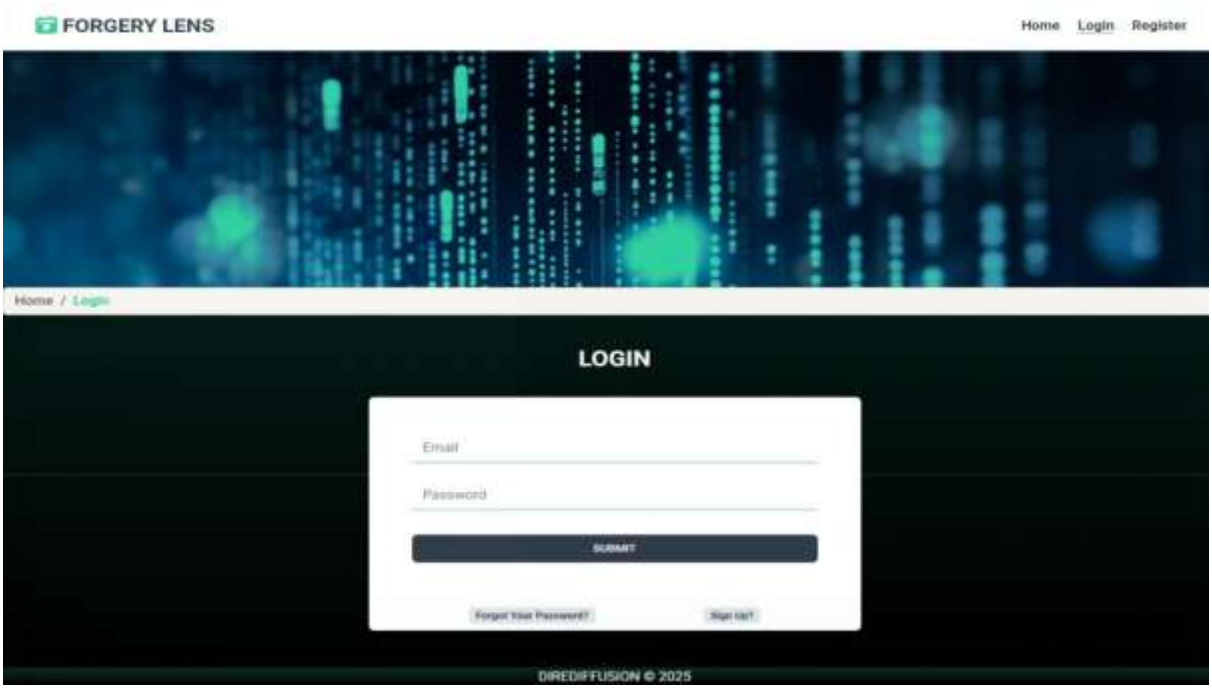

Fig: 6.2 Login page

Fig 6.3 Image and video upload page

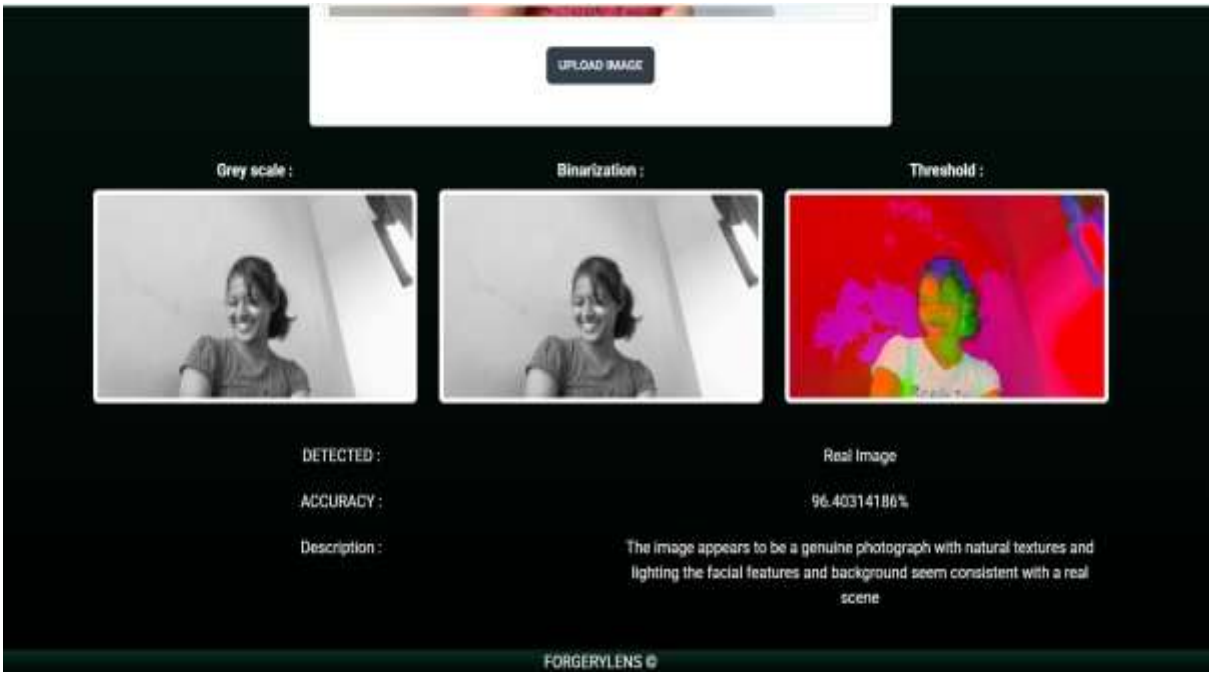

Fig 6.4 Result Pages

7. CONCLUSION

In conclusion, this project tackled the growing challenge of detecting forgeries in moving visuals, particularly deepfakes. We proposed a system leveraging the power of deep learning to analyse user-uploaded videos and identify potential manipulation attempts. The system extracts a comprehensive set of features from each video frame, capturing not only visual details but also temporal information between frames. A pre-trained deep learning model, likely a Convolutional Neural Network (CNN) like VGG16, analyses these features and is fine-tuned to recognize the subtle inconsistencies indicative of forgeries. This approach offers several advantages. Deep learning excels at recognizing complex patterns, making it well-suited for this task. Feature richness ensures the system can capture various manipulation techniques. Additionally, the system's design allows for continuous improvement through further training with more data. While technical, economic, and social considerations require careful planning, the project presents a feasible solution with the potential to significantly enhance trust in online videos. By combating misinformation and empowering users with a tool for verification, this project contributes to a more reliable digital media landscape.

REFERENCES

[1] Rana, M. S., Nobi, M. N., Murali, B., & Sung, A. H. (2022). Deepfake Detection: A Systematic Literature Review. IEEE Access, 10,pp.31578–31593.

[2] Aduwala, S. A., Arigala, M., Desai, S., Quan, H. J., & Eirinaki, M. (2021). Deepfake Detection using GAN Discriminators. In 2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService) (pp. 69-77). Oxford, United Kingdom. doi: 10.1109/BigDataService52369.2021.00014.

[3] H. Agarwal, A. Singh, and R. D, ”Deepfake Detection Using SVM,” 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2021, pp. 1245- 1249, doi: 10.1109/ICESC51422.2021.9532627.

[4] Khan, S. A., & Dang-Nguyen, D.-T. (2024). Dee Ac pfake Detection: An- alyzing Model Generalization ross

[5] Ramadhani, K. N., Munir, R., & Utama, N. P. (2024). Improving Video Vision Transformer for Deepfake Video Detection Using Facial Landmark, Depthwise Separable Convolution, and Self Attention. IEEE Access, 12,pp.8932–8939.

[6] Patel, Y. et al. (2023). Deepfake Generation and Detection: Case Study and Challenges. IEEE Access, 11, 143296–143323.

[7] Malik, A., Kuribayashi, M., Abdullahi, S. M., & Khan, A. N. (2022). DeepFake Detection for Human Face Images and Videos: A Survey. IEEE Access, 10,pp.18757–18775.