TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING

LALITPUR ENGINEERING COLLEGE
CHAKUPAT, LALITPUR

MAJOR PROJECT REPORT
ON

DefaceLab: DeepFake Detection using ViT

SUBMITTED BY
ABHISHEK NEUPANE [LEC076BCT002]
RABINDRA ADHIKARI [LEC076BCT025]
SANJISH MAHARJAN [LEC076BCT032]
SUSHIL KAFLE [LEC076BCT045]

SUBMITTED TO
DEPARTMENT OF COMPUTER ENGINERING

AUGUST 2023

DefaceLab: DeepFake Detection using ViT

Submitted by
ABHISHEK NEUPANE [LEC076BCT002]
RABINDRA ADHIKARI [LEC076BCT025]
SANJISH MAHARJAN [LEC076BCT032]
SUSHIL KAFLE [LEC076BCT045]

Project Supervisor
Er. Bisikha Subedi

A project submitted in partial fulfillment of the requirements for the degree of
Bachelor of Computer Engineering

DEPARTMENT OF COMPUTER ENGINEERING
Lalitpur Engineering College
Tribhuvan University

AUGUST 2023

ABSTRACT

Deepfakes are realistic-looking fake media generated by deep-learning algorithms that iterate through large datasets until they have learned how to solve the given problem (i.e., swap faces or objects in video and digital content). The massive generation of such content and modification technologies is rapidly affecting the quality of public discourse and the safeguarding of human rights. Deepfakes are being widely used as a malicious source of misinformation in court that seek to sway a court's decision. Because digital evidence is critical to the outcome of many legal cases, detecting deepfake media is extremely important and in high demand in digital forensics. As such, it is important to identify and build a classifier that can accurately distinguish between authentic and disguised media, especially in facial-recognition systems as it can be used in identity protection too. This is what we tend to do. In this work, we will be using recent Vision Transformer Model, as it uses the attention based mechanism. Since, the Vision transformer has its own unique attributes and is very much reliable and provides a lot of significant merits in development compared to others, we plan to implement it for our work, for the detection of the AI generated images.

Keywords: *deepfake detection; digital forensics; media forensics; deep learning; Vision Transformer; AI generated images*

Contents

List of Figures

Abbreviations

AI	Artificial Intelligence
CASIA	Chinese Academy of Sciences Institute of Automation
DL	Deep Learning
FPS	Frames Per Second
GANs	Generative Adversarial Networks
ML	Machine Learning
MSA	Multihead Self-Attention
SA	Self-Attention
ViT	Vision Transformer

1 INTRODUCTION

Deepfake technology has revolutionized the world of digital media manipulation. By combining artificial intelligence and image/video processing, deepfakes have garnered widespread attention. Deepfakes involve the creation of realistic media portraying individuals in situations they never experienced or saying things they never said. As this technology becomes more sophisticated, concerns arise regarding its impact on politics, entertainment, and personal privacy. This report provides an overview of deepfakes, including their underlying processes, societal implications, ethical challenges and the way of detection. In navigating this landscape, it is crucial to find a balance between innovation and responsible use in our increasingly digitized society.

In the last few years, cybercrime, which accounts for a 67% increase in the incidents of security breaches, has been one of the most challenging problems that national security systems have had to deal with worldwide.[1] Deepfakes, at present time, are being widely used to swap faces or objects in video and digital content. This artificial intelligence-synthesized content can have a significant impact on the determination of legitimacy due to its wide variety of applications and formats that deepfakes present online (i.e., audio, image and video). Given the speed, simplicity, and effects of social media, convincing deepfakes can easily persuade millions of people, ruin the lives of their victims, and have a detrimental effect, persuasive deepfakes can rapidly influence millions of people, destroy the lives of its victims and have a negative impact on society in general [1]. Deepfake technology has been driven by various motivations, including individual attacks, political manipulation, and the spread of false information. Its impact extends beyond personal attacks to manipulating satellite images and using stock images for identity protection. Cyber attackers continuously adapt their strategies, making it challenging to identify deepfake media and stay ahead of evolving threats.

The societal implications of deepfake technology are profound. Misinformation and disinformation fueled by deepfakes erode public trust, damage reputations, and violate privacy at personal and professional levels. Deepfakes can also disrupt democratic processes and contribute to societal polarization. Addressing the legal and ethical concerns surrounding deepfakes requires technological advancements, policy development, media literacy, and careful consideration of privacy rights and the manipulation of visual evidence. To tackle these implications, it is essential to advance deepfake detection methods, bolster cybersecurity measures, promote media literacy for individuals to discern manipulated content, and establish clear legal frameworks governing the responsible use of deepfake technology. By taking a comprehensive approach, we can effectively navigate the ethical challenges and societal impacts posed by deepfakes.

The deepfake technology holds importance in several areas. It offers creative expression and entertainment possibilities, enhances research and development in fields like computer vision, and aids forensic analysis in legal investigations. Deepfakes also emphasize the need for media literacy and critical thinking skills, promoting education and awareness. Ethical considerations and policy development are crucial in addressing the responsible use of deepfakes and protecting individuals' rights. Understanding the significance of deepfake technology enables us to navigate its implications effectively and harness its potential while mitigating potential harm.

We cannot dispute the influence deep fakes will have in the next years given all the benefits and cons that have been presented. Therefore, keeping an eye on deepfake content is crucial. This paper will provide an overview of the fundamental organizational structure of our project on how deepfake detections can be done.

1.1 Background

At present context of time, the rapid advancements in mobile camera technology and the widespread use of social media platforms have made it easier than ever to create and share digital pictures. Deep learning has played a crucial role in developing technologies that were previously unimaginable. One notable example is modern generative models, which can produce highly realistic images, speech, music, and video. These models have been applied in various fields, such as enhancing accessibility through text-to-speech technology and generating training data for medical imaging.

The deepfakes are created using deep learning techniques like Generative Adversarial Networks (GANs). These techniques involve two main components: a generator and a discriminator. The generator produces fake content, such as faces or scenes, while the discriminator tries to distinguish between real and fake content. Through an iterative process, the generator improves its ability to create increasingly realistic output, aiming to deceive the discriminator. As training progresses, the generated content becomes more convincing, making it difficult to distinguish whether the content is genuine or artificially created. This technology has both creative potential and serious ethical implications, as it can be used for entertainment purposes but also for generating misleading or harmful content.

Deep fakes are now widely disseminated on social media platforms, which encourages spamming and the spread of false information. Just picture a deep fake image of Donald Trump getting arrested which was trending on twitter or a deep fake of a well-known celebrity assaulting their supporters. These types of misinformation can brainwash the audience and are awful and endanger and mislead the general public.

Deep fake detection plays a crucial part in overcoming such a circumstance. Therefore, we provide a novel deep learning-based method that can successfully separate artificial intelligence-generated fake contents from authentic digital materials. In order to identify deep fakes and stop them from spreading across the internet, it is crucial to develop technology that can detect deepfakes.

1.2 Problem Statement

With the help of visual effects, convincing modifications of digital photographs and videos have been proven for many years. However, new developments in deep learning have dramatically increased the realism of fake content and made it more widely available. These purportedly artificial intelligence-generated works of media are also known as ”deepfakes”. It is easy to create deep fakes utilizing artificial intelligence techniques. However, it is extremely difficult to identify these Deep Fakes. Globally, it is found out that about 71% of total population using internet do not know what a deepfake is. Just under a third of global consumers of internet say they are aware of deepfakes [8]. In the past, there have been numerous instances of deep fakes being used to effectively incite political unrest, stage terrorist attacks, blackmail individuals, etc. In North America alone, the proportion of deepfakes more than doubled from 2022 to 2023. This proportion jumped from 0.2% to 2.6% in the US. It is up from 0.1% to 4.6% in Canada [7] and is rapidly growing. Therefore, it becomes crucial to identify these deep fakes and montior their spread through social media. Hence, with the growing curiosity we have taken a step forward in detecting the deep fakes using different transformer based models.

1.3 Objectives

- Our project identifies real images and filters out deepfakes, preventing the spread of misinformation.

1.4 Scope

At present time there are numerous tools available for creating false videos in the current deepfake technology landscape, but there are few trustworthy tools available for spotting them. The idea creation of a deepfake detection software to solve this variance and stop the widespread dissemination of deepfakes is what our project is based upon. Users will be able to post images through our platform and segregate them as authentic or deepfake. This project can also be developed to include the development of a plugin for browsers that will automatically detect deep fakes. Notably, our idea can me implemented on different social sites as well as in various various governmental organizations. A synopsis of the program with the size of the input, bounds on the input, input validation, input dependency, the i/o state diagram, and the major inputs and outputs are explained in this report.

2 LITERATURE REVIEW

The facial manipulation is a techniques of altering someone's face in images or videos, often done using computer software. There are different types of facial manipulation techniques, each serving different purposes such as Face Morphing where two or more faces are blended together and create a seamless transition between them, Face Swapping, which involves replacing one person's face with another's while keeping the rest of the image intact, Attribute manipulation, where different attributes of face like nose, eyes, etc. are manipulated and Deepfakes, which uses advanced artificial intelligence and machine learning to create highly realistic digital contents by superimposing one person's face onto another person's body. These videos can convincingly mimic speech, expressions, and gestures, making them a source of concern for potential misinformation and privacy issues.

The generation of deepfakes has had a rapid growth. Generating deepfakes involves using advanced technology to create fake videos or images that look very realistic. The deepfakes are created using deep learning techniques like Generative Adversarial Networks (GANs). The GAN technique involve two main components: a generator and a discriminator. The generator takes random noise as input and produces data, like images. The discriminator evaluates this generated data alongside real data and tries to tell them apart. As training progresses, the generator refines its output to become more convincing, while the discriminator becomes better at distinguishing real from fake. This competitive process pushes the generator to create increasingly realistic data that can be mistaken for real. As training progresses, the generator refines its output to become more convincing, while the discriminator becomes better at distinguishing real from fake. This competitive process pushes the generator to create increasingly realistic data that can be mistaken for real.

This remarkable advancement in deepfake technology underlines the desperate need for effective deepfake detection mechanisms. As these deceptively authentic videos, images, and audio recordings becomes more widespread, the risk of misinformation, deception, and breaches of privacy escalates hugely. Detecting deepfakes is a difficult challenge. However, researchers and experts are actively exploring diverse strategies to differentiate between these genuine and manipulated content. These strategies often involve detecting facial features for anomalies, identifying patterns and residuals in lighting and shadows, and uncovering distortions introduced during the manipulation process such as iris and pupil deformation.

Since 2017, the Transformer has been very renowned as a new type of neural archi-

tecture for natural Language Processing which encodes the given input data into a powerful feature, with the help of attention mechanism. With the research article published in 2017, named "Attention Is All You Need" [6], by taking it as a framework, numerous researches has been done recently upon the visions task as well [3]. This is where the Visual Transformer comes in.

While the Transformer Architecture has been a new standard for Natural Language Processing tasks, its application to computer vision remains limited. From many recent studies and researches done, the common points mentioned was that in vision, either the attention is applied in conjunction with convolution's networks, or used to replace certain components of convolutional network while keeping their overall structure in place [4]. The Transformer-liked architecture has been employed in the computer vision field in present context in three fundamental computer vision tasks, namely classification, detection and segmentation [2]. Our work lies within one of these fundamental tasks i.e the detection of artificially generated images of a human face.

The Visual Transformer has been gaining many contributions in recent years such that its capabilities has increased beyond the old traditional models. As a matter of fact, the Visual Transformer model has clear advantages over traditional models such as CNNs and RNNs in specific situations. It can process whole images in sections, which helps it understand the bigger picture. Also, it is good with working on different image sizes and tasks. Training these models on big datasets first and then adjusting it for specific tasks makes it really flexible compared to the traditional model [5]. One of its key characteristics, self-attention, not only helps us understand how the model works but also reminds us to pick the right model based on the job, data, and what we have.

In the Vision Transformer, the use of multi-headed self attention is used which allows the model to associate each individual testing attributes of input to run parallelly. The query, key and values vectors are used as the calculation attributes to calculate scores, ultimately gaining the softmax score for obtaining probability values and scaled scores for attention weights. These multi-headed attention are the concatenated for further processes. Before feeding the inputs in the ViT model, the input datas are to be preprocessed first. Initially, the frame extraction process is done, followed by the features abstraction. The features such as facial structure can be extracted with the help of Face Alignment Library. Then the input is normalized followed by resizing and patching, which provides a representation of NLP for the input to Vision Transformer [4].

From recent researches done Vision Transformer (ViT) is deemed more suitable than Generative Adversarial Networks (GANs) for deepfake detection because Vision Trans-

former is specifically designed for image analysis tasks like classification. GANs are like artists that make fake pictures, but they're not as good at spotting fakes. Also, the tricks that make GANs create fakes can also fool themselves when looking for fakes. Vision transformer way of looking at pictures makes it a good choice for catching fake ones, because it's good at seeing small things that don't match up. Also, Vision Transformers (ViTs) have advantages over Recurrent Neural Networks (RNNs) in computer vision due to their parallel processing, better handling of long-range dependencies, efficient attention mechanisms, fewer assumptions about sequence structure, transfer learning capabilities, reduced overfitting, and interpretable representations. However, RNNs remain strong for sequential data tasks and short-term dependencies. Whereas against Convolutional Neural Networks (CNNs), ViT has advantages of efficient parallel processing, reduced invariance assumption, scalability, interpretable representations, fewer specialized architectures, and transfer learning. But CNNs is better for tasks requiring local features and spatial hierarchies.

In summary, with all these distinct attributes and merits the Vision Transformer imposes itself significantly compared to others at present time and with understanding of the factors more compatible for our project, we have selected the Vision Transformer as the major model for our project development.

2.1 Existing Systems

2.1.1 Deepware

Deepware.ai is an innovative company at the forefront of deepfake detection technology. They specialize in developing advanced AI-driven solutions to combat the spread of manipulated media content. With a team of expert researchers and engineers, Deepware.ai leverages state-of-the-art machine learning algorithms and deep neural networks to accurately identify deepfakes. Their cutting-edge technology, combined with a user-friendly approach, empowers individuals and organizations to protect themselves from the potentially harmful consequences of deepfakes. Deepware.ai's commitment to continuous improvement and staying ahead of evolving deepfake techniques positions them as a trusted leader in the field, offering reliable and scalable solutions that contribute to a safer digital landscape.

Figure 1: Deepware

2.1.2 DuckDuckGoose

DuckDuckGoose offers an open-source browser extension that keeps tabs on all websites you visit and alert you once manipulated media is detected. Users should also appreciate the transparency of the DeepFake detector, as DuckDuckGoose provides detailed explanations for why a video was flagged to give you some insight on what to look for in a DeepFake. The team behind the tool has been dedicated to sharing their research findings and encouraging participants from the community to contribute to building a more reliable model with higher accuracy.

Figure 2: DuckDuckGoose

2.2 Proposed Systems

Our Deepfake Detection System is an online tool designed to help people identify and deal with deepfake content. It focuses on providing strong detection capabilities for deepfakes. Users can use the system to analyze and detect potential deepfakes by submitting images. The system uses advanced algorithms and machine learning techniques to accurately identify manipulated or fake media. This helps users recognize and address the risks that come with deepfakes, such as spreading false information, committing fraud, or violating privacy. The deepfake detection system aims to empower users by giving them the tools they need to protect themselves and others from the harmful effects of encountering deepfakes. With the help of cutting-edge algorithms, the system assists users in detecting and raising awareness about deepfakes, making the digital world a safer and more informed place.

Figure 3: DefaceLab

3 FEASIBILITY STUDY

3.1 Economic feasibility

This is a low-budget project with no development costs. The total expenditure of the project is just computational power. The dataset and computational power required for the project are easily available. The computational power is easily provided by google collab. So, the project is economically feasible. The system will be simple to comprehend and use. As a result, there will be no need of trained personnel to use the system. This system will have the capacity to expand by adding more components.

3.2 Operational feasibility

The project is operationally feasible since after the completion of the project, it can be operated as intended by the user to solve the problems for what it has been developed.

3.3 Technical feasibility

The purpose of technical feasibility is to establish whether the project is possible in terms of software, hardware, manpower, and knowledge to complete. It will take into account determining resources in support of the suggested scheme. The system is platform independent because it is written in Python. Advanced machine learning libraries are available and the technology is cutting-edge. As a result, the system is technically possible.

4 METHODOLOGY

4.1 Software Development Life Cycle

Agile method of Software Development uses iterative approach. Agile method cycles among Planning, Requirement Analysis, Designing, Development and Testing stages. These cycle is called sprints. Each sprints are considered as a miniature project on itself. Using this method allowed us to update various parts of project at any point of project development. In this model an iterative approach was taken where working software was delivered after each iteration some new features is added to main system. It works in incremental and iterative approach. Agile model mainly focuses on customer collaborations, on individuals and iterations and welcomes changes at anytime in SDLC process. We prefer to use agile model in this system as it helps in developing realistic systems and promotes teamwork during software development. Also system is easy to manage and it can accommodate new changes at any stages of software development phase.

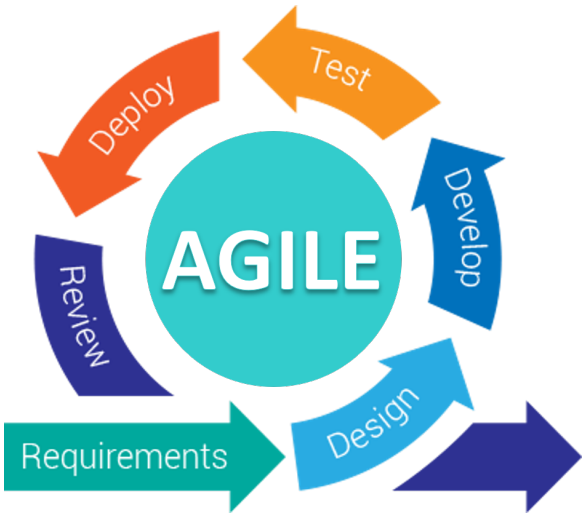

Figure 4: Agile Model

4.2 System Development Tools

Our static Deepfake detection System requires Python, OpenCV, Machine Learning which are listed below:

1. Python
2. Pytorch
3. NumPy
4. OpenCV

4.3 Functional Requirement

The functional requirements of the system are:

1. Accurately distinguish between real and manipulated images.
2. Provide a user-friendly interface for uploading and analyzing images.

4.4 Non Functional Requirement

These requirements are not needed by the system but are essential for the better performance of software. The points below focus on the non-functional requirement of the system.

- Reliability
- Usability
- Security
- Portability
- Speed and responsiveness
- Performance

4.5 Key Features Extraction

Effective deepfake detection relies on the extraction of specific features from videos or images that can reveal inconsistencies or anomalies introduced by the manipulation process. Various techniques are employed to capture these key features, enabling the development of robust detection models. The following are some of the key features commonly extracted for deepfake detection:

Figure 5: Facial landmarks extraction from video

4.5.1 Abnormal Patterns

Deepfake detection methods often look for abnormal patterns that deviate from the expected characteristics of genuine content. These patterns can include unnatural facial expressions, inconsistent facial movements. Detecting such abnormalities through pattern recognition can raise suspicion of manipulation.

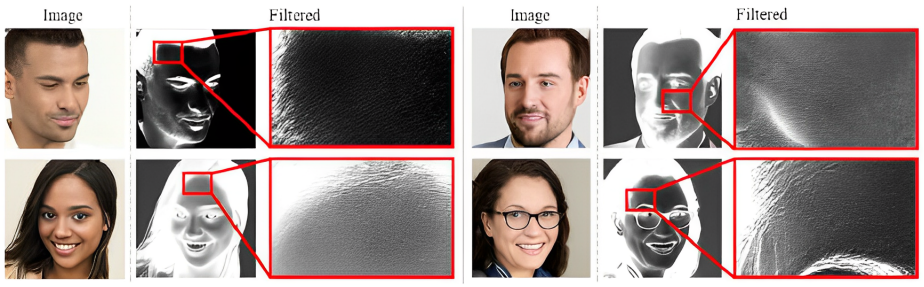

Figure 6: Abnormal Patterns

4.5.2 Iris Color and Pupil Boundary

Iris color and pupil boundary analysis involves examining the color distribution and boundary characteristics of the iris and pupil regions. Variances in iris color or irregularities in pupil shape can indicate potential manipulation. Comparing the extracted features of the iris and pupil between frames can help detect abnormal variations introduced by deepfake techniques.

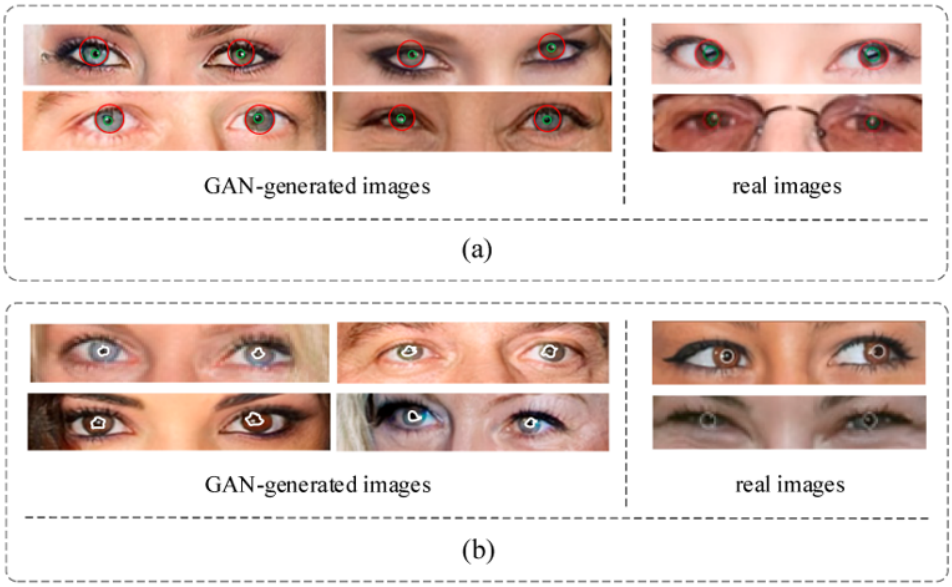

Figure 7: Iris Color and Pupil Boundary

4.5.3 Residual Comparison

Residual comparison involves analyzing the residuals obtained from subtracting the original image from its manipulated counterpart. By comparing the residuals, artifacts and inconsistencies introduced during the manipulation process can be detected. Unusual patterns or significant deviations between residuals may signify the presence of a deepfake.

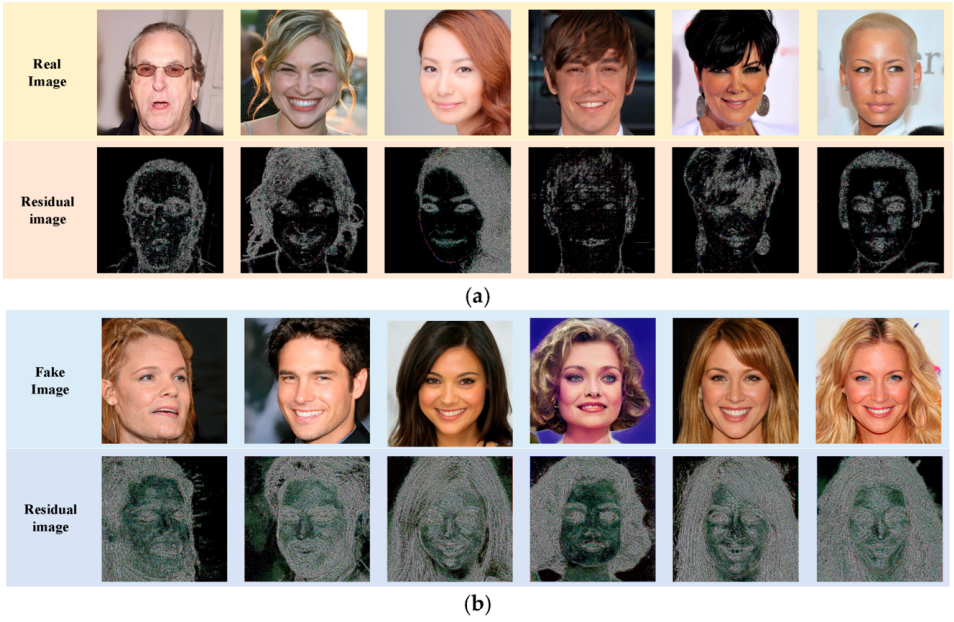

Figure 8: Residual

4.5.4 Kernel Density Estimation (KDE) of Color

Kernel Density Estimation (KDE) involves estimating the probability density function of color values in an image. By applying KDE to different regions of an image, it becomes possible to identify unusual color distributions introduced by deepfake manipulation. Sudden spikes or dips in color density can be indicative of tampering.

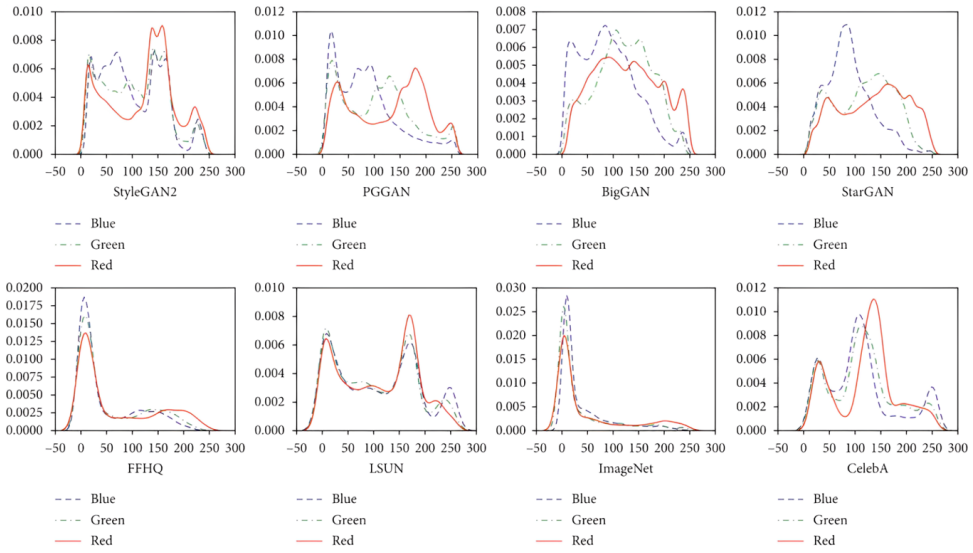

Figure 9: Kernel Density Estimation (KDE) of Color

4.6 Dataset Collection

Our dataset acquisition process involved the following steps:

4.6.1 Public Datasets

A significant part of our extensive dataset was thoughtfully put together from openly accessible collections of deepfake and authentic images. We carefully chose these datasets to make our deepfake detection model more diverse and applicable. This approach helps us ensure a strong and thorough training and testing process by making the most of these available resources.

Here's the breakdown of our dataset:

- **Trained:** 70,000 samples
- **Tested:** 5,000 samples
- **Validation:** 15,000 samples

Our dataset contains data from well-known public sources, and each of these sources has a specific role in making our deepfake detection system better :

1. **DFDC (DeepFake Detection Challenge) Dataset:** The DFDC dataset offers an invaluable benchmark for detecting deepfake manipulations across a wide spectrum of scenarios and visual contexts. Its meticulous curation and large-scale inclusion of deepfake and real videos enable us to train our model on highly realistic and challenging instances.

2. **CelebA Dataset:** By incorporating the CelebA dataset, a prominent repository of celebrity faces in diverse poses and expressions, we bolster our model's capacity to handle variations in lighting conditions, facial orientations, and natural facial expressions – all critical aspects in detecting nuanced manipulations.

3. **CASIA WebFace Dataset:** With the CASIA WebFace dataset, we tap into a wealth of real facial images sourced from the internet, exposing our model to a plethora of natural variations in appearance, pose, and lighting conditions. This exposure fortifies our model's ability to discern between real and manipulated facial features in an ever-evolving digital landscape.

4.6.2 Video Conversion

To include videos in our dataset, we first converted them into individual frames (images) to facilitate compatibility with the vision transformer architecture. This step involved extracting frames at a consistent frame rate from each video, resulting in a sequence of images for each video.

Figure 10: Video Sample

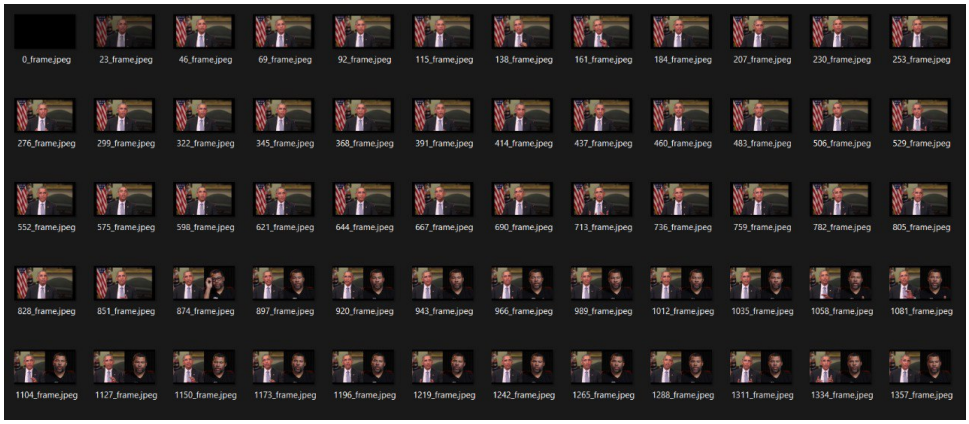

Figure 11: Frames Extracted from video

4.6.3 Frame Selection

To avoid redundancy and maintain dataset balance, we carefully selected frames from videos to represent various stages of manipulation, expressions, poses, and lighting conditions.

Figure 12: Seleting required frames

4.6.4 Annotation and Labeling

Each image was labeled as either ”real” or ”deepfake.” Annotations were done manually to ensure accurate labeling for training and evaluation.

4.7 Preprocessing

Before training the model, we preprocess the dataset as follows:

4.7.1 Resize

All images, whether sourced from videos or other datasets, were resized to a consistent resolution of 112x112 pixels. This resizing ensured a uniform input size for the vision transformer.

Figure 13: Cropped Image

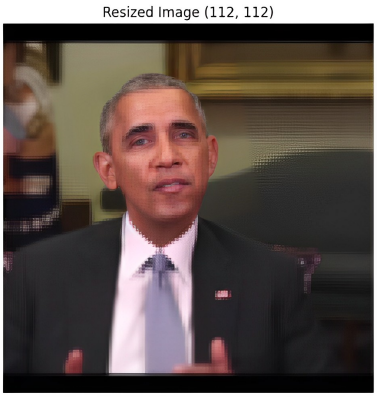

Figure 14: Resized Image

4.7.2 Augmentation

To increase the dataset's diversity and improve the model's ability to generalize, we applied various data augmentation techniques. These techniques included random rotations, horizontal flips, brightness adjustments, and minor deformations.

4.7.3 Normalization

Pixel values of the images were normalized to a specific range to ensure consistent input for the model during training and inference.

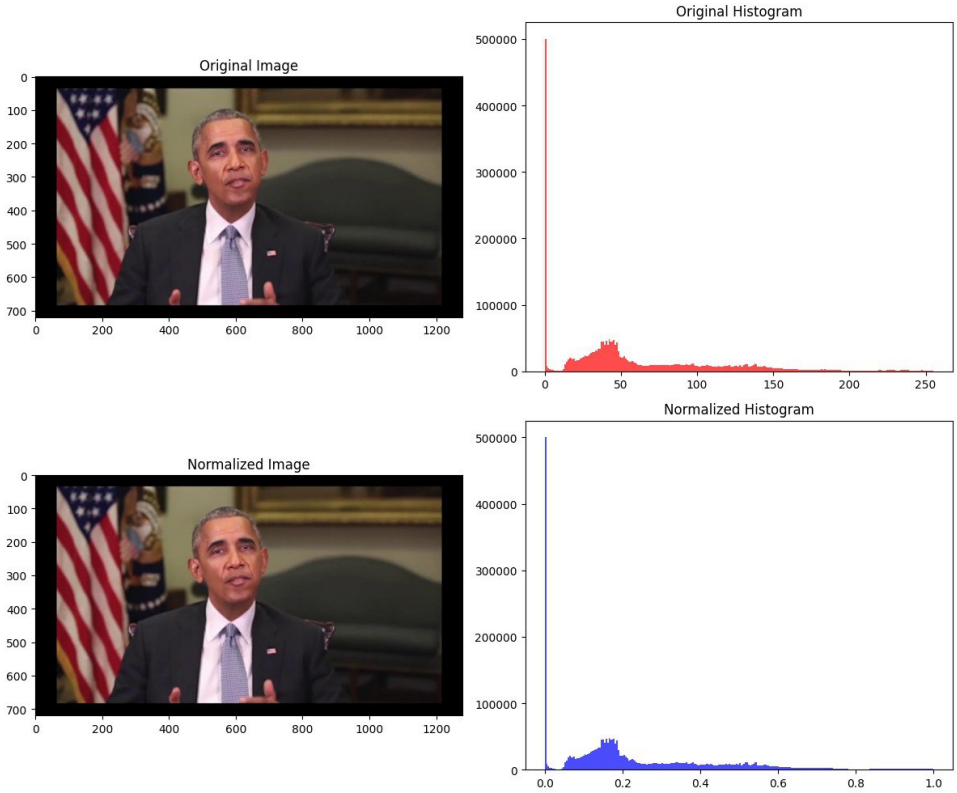

Figure 15: Normalization

By preprocessing the dataset, we enhanced the model's capacity to learn relevant features and intricate patterns required for accurate deepfake detection.

4.8 Vision Transformer Architecture

The Vision Transformer (ViT) architecture comprises the following components:

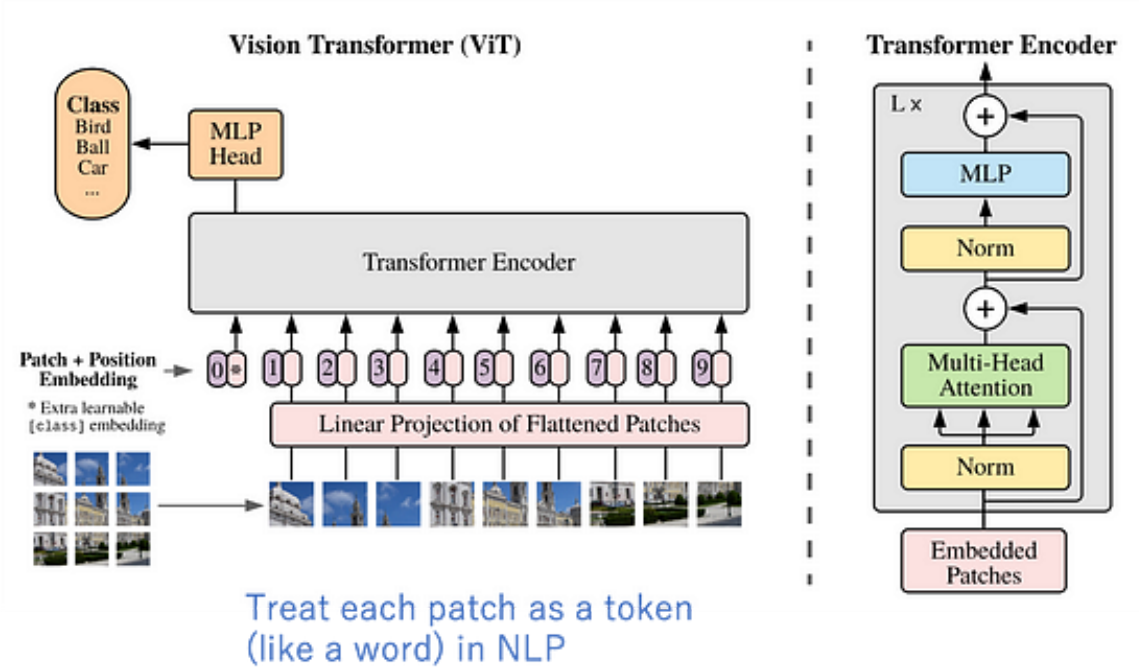

Figure 16: Vision Transformer Architecture

4.8.1 Patch Embedding

The Patch Embedding step in the Vision Transformer architecture involves the following detailed process:

a. **Image Patching:** The input image is divided into a grid of non-overlapping patches. Let $\mathbf{X} \in \mathbf{R}^{H \times W \times C}$ represent the original image, where H is the image height, W is the image width, and C is the number of channels (color depth). We partition $\mathbf{X}$ into patches of size $P \times P \times C$, resulting in a tensor $\mathbf{X}_{\text{patches}} \in \mathbf{R}^{N \times P \times P \times C}$, where N is the total number of patches.

b. **Flattening:** Each patch is reshaped into a vector using a flatten operation. The flattened patches are denoted as $\mathbf{X}_{\text{flat}} \in \mathbf{R}^{N \times (P \times P \times C)}$.

c. **Linear Projection:** The flattened patches are projected into a lower-dimensional space using a learnable linear transformation. Let $\mathbf{W}_{\text{proj}} \in \mathbf{R}^{(P \times P \times C) \times D_{\text{proj}}}$ be the projection matrix, where D_{proj} is the dimension of the projected space. The projected patch embeddings are computed as $\mathbf{E} = \mathbf{X}_{\text{flat}} \cdot \mathbf{W}_{\text{proj}} \in \mathbf{R}^{N \times D_{\text{proj}}}$.

d. **Positional Encoding:** To provide spatial information to the transformer model, positional encodings are added to the patch embeddings. Each patch embedding is enhanced with a positional encoding vector $\mathbf{P} \in \mathbf{R}^{N \times D_{\text{proj}}}$. The final patch embeddings with positional information are given by $\mathbf{E}_{\text{pos}} = \mathbf{E} + \mathbf{P}$.
Mathematically, the above steps can be summarized as follows:

$$\mathbf{X}_{\text{patches}} = \text{ImagePatching}(\mathbf{X}),$$

$$\mathbf{X}_{\text{flat}} = \text{Flatten}(\mathbf{X}_{\text{patches}}),$$

$$\mathbf{E} = \mathbf{X}_{\text{flat}} \cdot \mathbf{W}_{\text{proj}},$$

$$\mathbf{E}_{\text{pos}} = \mathbf{E} + \mathbf{P}.$$

The resulting patch embeddings with positional information, $\mathbf{E}_{\text{pos}}$, are then used as input for the subsequent stages of the Vision Transformer architecture.

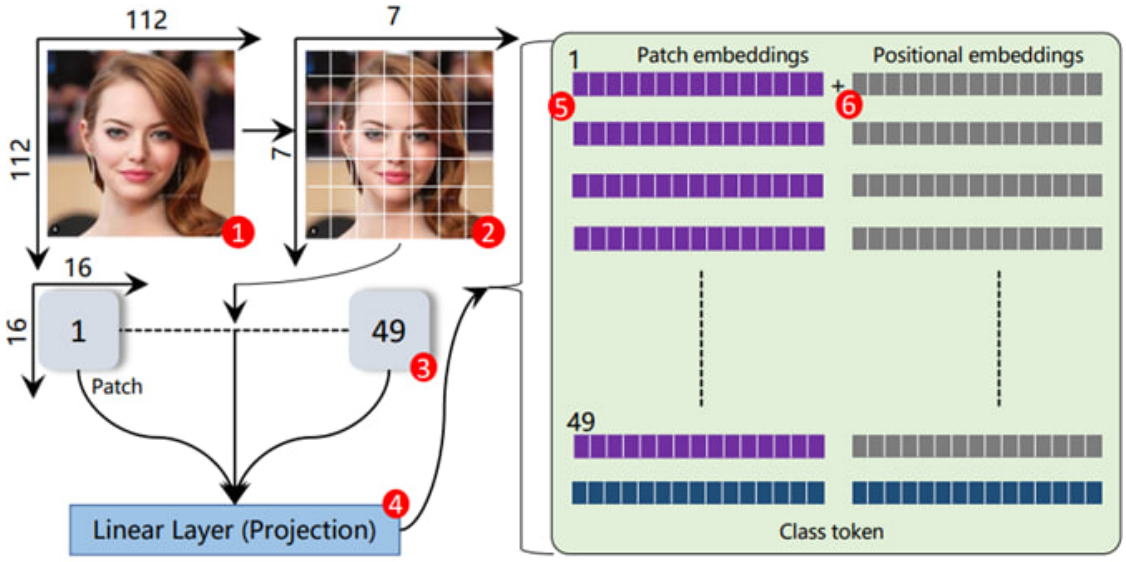

Figure 17: Patch Embedding

4.8.2 Transformer Encoder

The patch embeddings are then processed through a stack of transformer encoder layers. Each of these layers consists of two main components: multi-head self-attention and feedforward neural networks.

a. Multi-head self-attention: This mechanism allows the model to consider relationships between different patches, both locally and globally. It assigns different attention weights to different patches based on their relevance to each other, enabling the model

to capture long-range dependencies and relationships within the image.

Standard qkv Self-Attention (SA): For an input sequence $z \in \mathbf{R}^{N \times D}$ (with N elements, each having a D-dimensional feature vector), we compute a weighted sum over all values v in the sequence. The attention weights A_{ij} are determined based on the similarity between elements of the sequence and their corresponding query q_i and key k_j representations.

$$[q, k, v] = zU_{qkv}, \quad U_{qkv} \in \mathbf{R}^{D \times 3Dh}$$

$$A = \text{softmax}\left(\frac{qk^T}{\sqrt{Dh}}\right), \quad A \in \mathbf{R}^{N \times N}$$

$$\text{SA}(z) = Av$$

MSA extends SA by running k self-attention operations (heads) in parallel and then concatenating their outputs. To ensure consistent computation and parameter complexity, the dimension Dh (from Eq. 5) is usually set to D/k.

$$\text{MSA}(z) = [SA_1(z); SA_2(z); \ldots; SA_k(z)], \quad U_{msa} \in \mathbf{R}^{k \cdot Dh \times D}$$

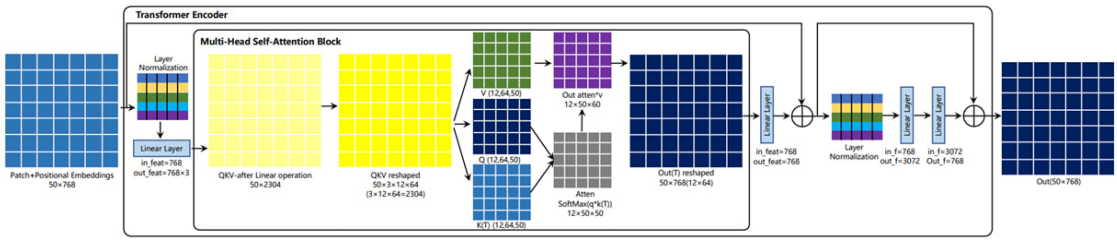

Figure 18: Transformer Encoder

b. Feedforward neural networks: After the attention mechanism, the data passes through feedforward neural networks, which process and transform the information further, helping the Vision Transformer model learn intricate features and patterns in the image data.

Feedforward neural networks consist of multiple layers, including fully connected (dense) layers followed by non-linear activation functions. Let's denote the input to the feedforward neural network as $\mathbf{x} \in \mathbf{R}^{D_{\text{FFN}}}$, where D_{FFN} is the dimension of the input features.

The transformation in each layer of the feedforward neural network can be represented as follows:

$$\mathbf{h}^{(l+1)} = \text{GELU}\left(\mathbf{W}^{(l)}\mathbf{h}^{(l)} + \mathbf{b}^{(l)}\right),$$

where $\mathbf{h}^{(l)}$ is the output of the l-th layer, $\mathbf{W}^{(l)}$ is the weight matrix, $\mathbf{b}^{(l)}$ is the bias vector, and GELU$(\cdot)$ is the Gaussian Error Linear Unit activation function.

The feedforward neural networks in the Vision Transformer's transformer encoder process each patch embedding independently through these layers, capturing complex patterns and relationships within the data.
After the final feedforward layer, the resulting representations are concatenated and used as the output of the transformer encoder for further processing or downstream tasks, making the Vision Transformer architecture a powerful tool for image analysis and understanding.

Activation Function: Gaussian Error Linear Unit (GELU) The Gaussian Error Linear Unit (GELU) is an activation function that provides a smooth approximation to the rectifier linear unit (ReLU) activation while maintaining differentiability, which can aid in the training process.
Two formulations of the GELU function are commonly used:

$$\text{GELU}(x) = 0.5x \left(1 + \tanh\left(\frac{\pi}{2} \cdot \left(x + 0.044715x^3\right)\right)\right)$$

and

$$\text{GELU}(x) = \frac{1}{2}x \left(1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$$

Where:

- x is the input to the GELU function.
- erf(z) is the error function.
- π represents the mathematical constant pi.
- $\sqrt{2}$ is the square root of 2.

The GELU activation function is often used in neural network architectures due to its smoothness and differentiability properties, making it suitable for gradient-based optimization during training.
Additionally, consider the mathematical expression:

$$P(X = c)$$

where X represents a random variable and c is a constant value. This expression is used to represent the probability that the random variable X takes on the value c.

4.8.3 Classification Head

The Classification Head is the final component of the Vision Transformer architecture, responsible for making predictions based on the learned features and patterns from the transformer encoder. It performs classification tasks, such as determining whether an image is real or manipulated.

a. Input Preparation: After going through the transformer layers, the image is divided into patches and transformed into a useful form for the classification head.

b. Pooling Operation: Think of this as collecting information from all the patches. The model adds up the important parts from each patch and takes the average. This gives a summary of the whole image.

c. Decision Making: The model uses this summary to decide what the image might be. For example, it could be deciding if an image is real or fake. It does this by comparing the features it has learned to what it has seen during training.

d. Class Prediction: The model assigns a score to each possible class (like ”real” or ”manipulated”). It does this by multiplying the summary with a set of numbers that it learned. Then, it uses the softmax function to turn these scores into probabilities. The class with the highest probability is the final prediction.

5 BLOCK DIAGRAMS

5.1 System Architecture

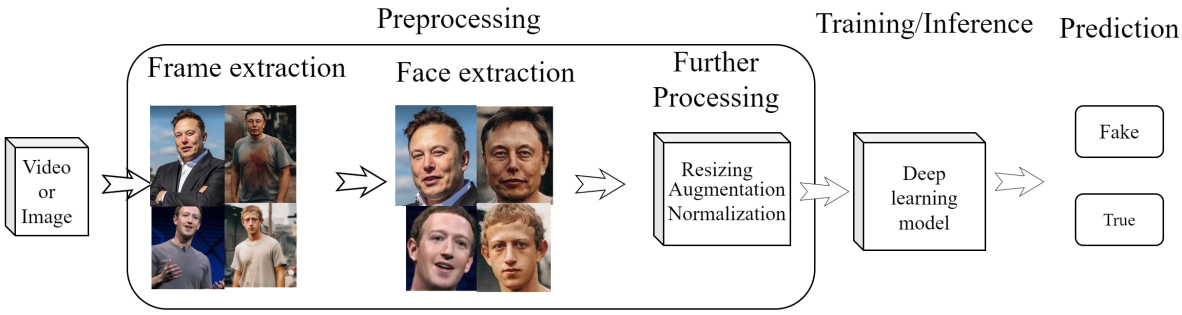

Figure 19: System Architecture

The system architecture of our project involves multiple steps. Initially, frames are extracted from the input video or obtained directly from an image. These frames then undergo face extraction, where faces are identified and cropped using face detection algorithms. The extracted faces are resized to a standardized size and undergo normalization to ensure consistent pixel values. The preprocessed face images are then fed into a deep learning model for classification. The model analyzes the features and patterns in the images to determine whether they are real or fake. Finally, the system produces the output, indicating the authenticity of the input video or image as either real or fake.

5.2 Use Case Diagram

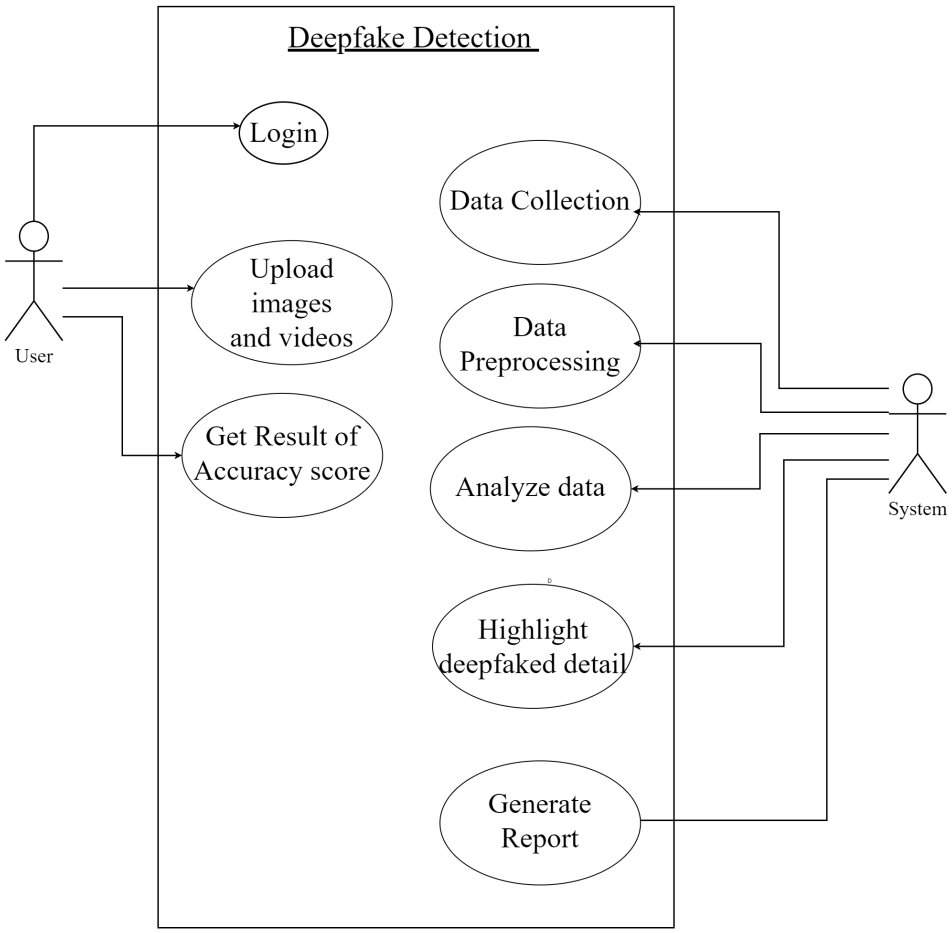

Figure 20: Use Case Diagram

The use case diagram for our system illustrates various interactions and roles of the system's users. The primary actors involved are the "User" and the "System." The User interacts with the system by initiating the deepfake detection process, either by uploading a video or an image. The User can also access the system to view the detection results. On the other hand, the System is responsible for managing the system, including user authentication, system configuration, and monitoring the overall functionality. The use case diagram shows the main use cases, such as "Upload Media," "Detect Deepfake," and "View Results," which represent the key functionalities of the system.

5.3 Sequence Diagram

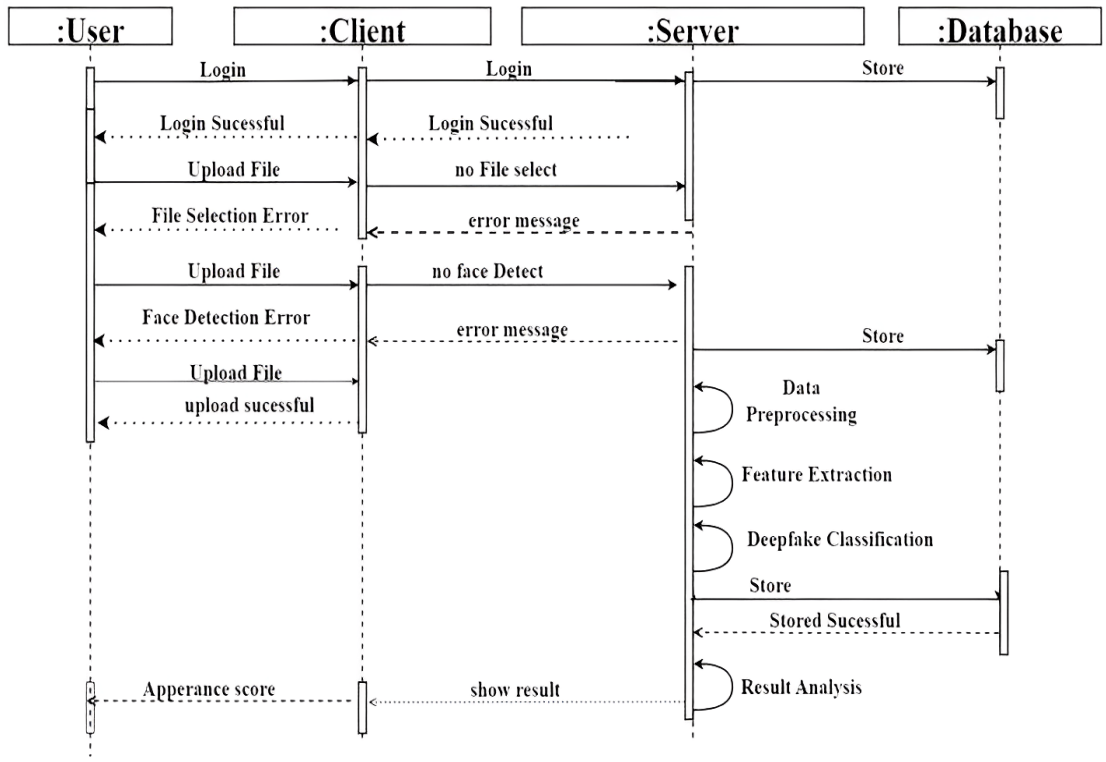

Figure 21: Sequence Diagram

The sequence diagram shows that the user initiates the process by accessing the system and providing their login credentials. The Server checks the login credentials and verifies the user's identity. Once authenticated, the user proceeds to upload a file containing the video or image to be analyzed for deepfakes. Then face detection algorithms are used to detect and extract faces from the uploaded media. This collected data undergoes further processing, including resizing and normalization, to prepare it for deep learning modeling. Finally, the processed data is fed into the deep learning model, which analyzes the features and patterns to classify the media as either real or fake.

5.4 Dataflow Diagram

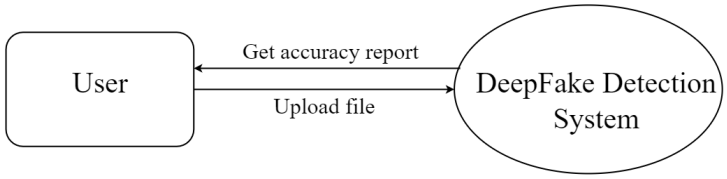

Figure 22: Level 0 DFD

DFD level – 0 indicates the basic flow of data in the system.

- User: User input to the system is uploading video.
- System: In system it shows all the details of the Video and output shows the fake video or not.
  and output flow

Hence, the data flow diagram indicates the visualization of system with its input feed to the system by User.

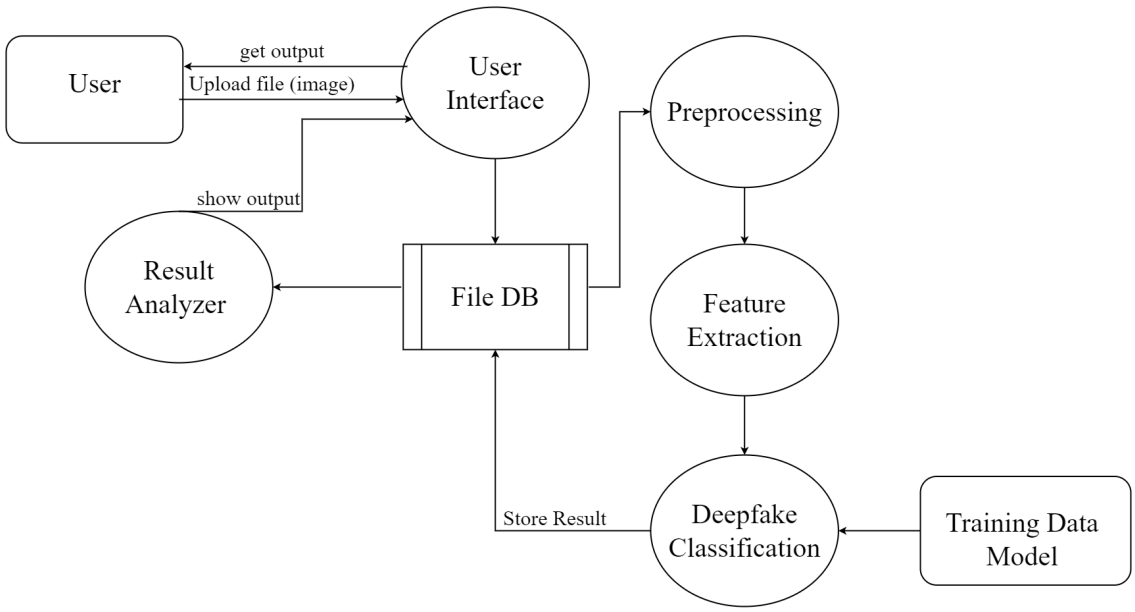

Figure 23: Level 1 DFD

DFD Level – 1 gives more in and out information of the system.Where system gives detailed information of the procedure taking place.

5.5 Activity Diagram

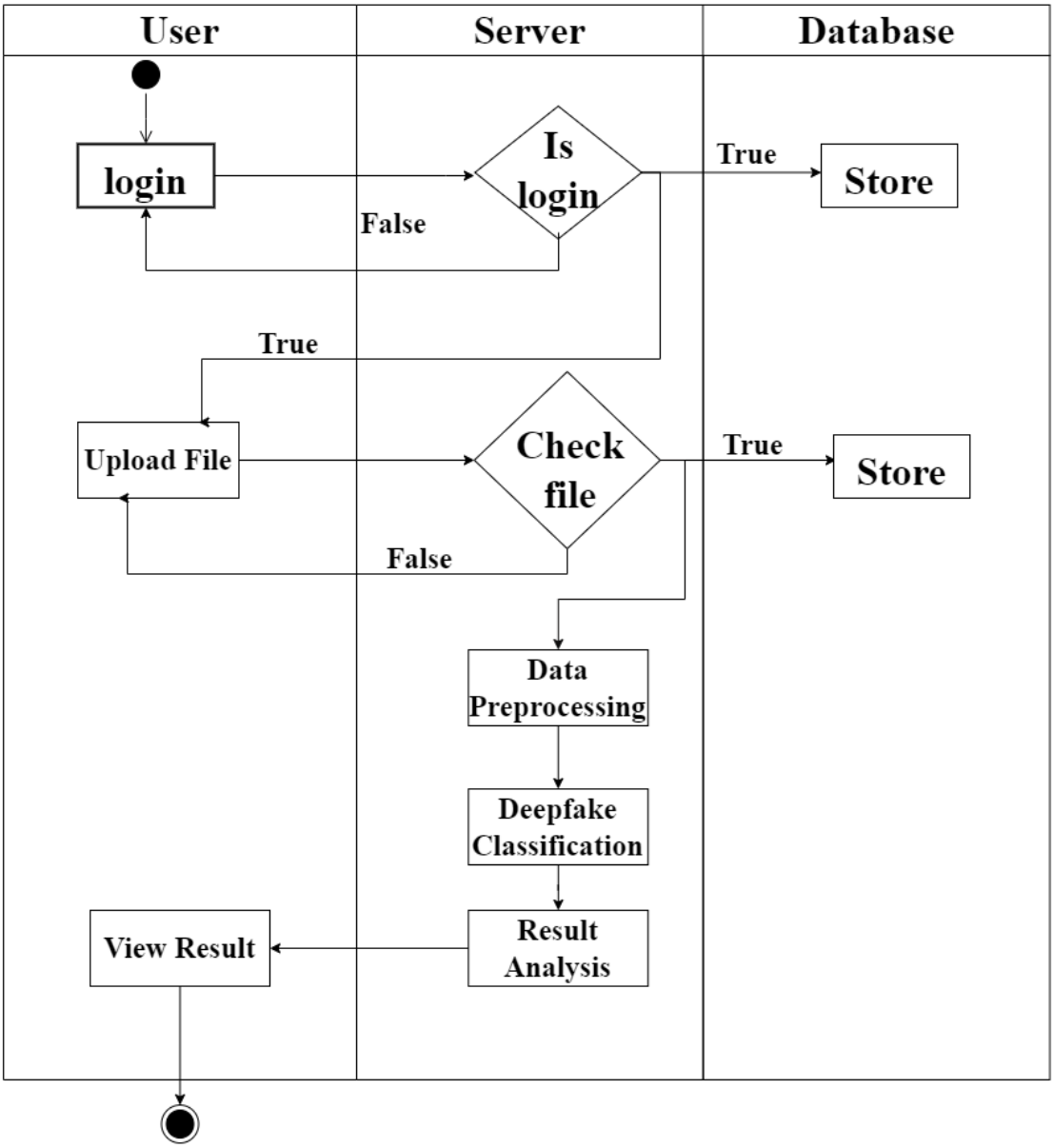

Figure 24: Activity Diagram

The activity diagram shows that the user initiates the process by accessing the system and providing their login credentials. The Server checks the login credentials and verifies the user's identity. Once authenticated, the user proceeds to upload a file containing the video or image to be analyzed for deepfakes.

6 Result and Analysis

6.1 UI of Project

We have used Flutter and Django to develop the mobile application in which we prepared a UI with user authentication, login page and home page where we can upload images to be classified as fake and real.

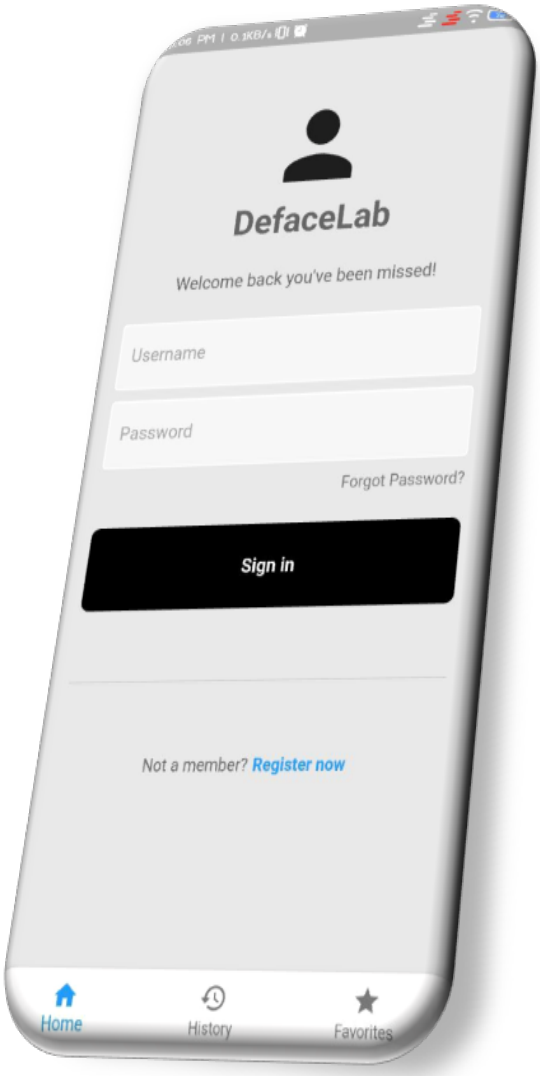

Figure 25: Login Page

Figure 26: Home Page

Figure 27: Upload image Menu

7 EXPECTED OUTCOMES

- User-friendly interface for easy upload and clear result presentation.
- Accurate identification of manipulated media content.
- Robust performance against different deepfake techniques and adversarial attacks.

References

[1] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, Matthias Nießner, ”FaceForensics++: Learning to Detect Manipulated Facial Images”

[2] Y. Liu et al., ”A Survey of Visual Transformers,” in IEEE Transactions on Neural Networks and Learning Systems.

[3] Han, K., Xiao, A., Wu, E., Guo, J., XU, C., and Wang, Y. (2021). ”Transformer in Transformer”.

[4] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). ”An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In International Conference on Learning Representations”

[5] Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, Zhiqiang He. (2021). ”A Survey of Visual Transformers”.

[6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N. and Polosukhin, I. (2017). ”Attention is all you need. In Advances in neural information processing systems (Vol. 30)”.

¡¡¡¡¡¡¡ HEAD

[7] Douglas Blakey, ”Forced verification and AI/deepfake cases multiply at alarming rates: Sumsub” (2023). https://www.electronicpaymentsinternational.com/news/forced-verification-and-ai-deepfake-caeses/ =======

[8] Douglas Blakey, ”Forced verification and AI/deepfake cases multiply at alarming rates: Sumsub” (2023). https://www.electronicpaymentsinternational.com/news/forced-verification-and-ai-deepfake-caeses-sumsub/ ¿¿¿¿¿¿¿ 877f7fb35d1be614bf0d8dd0d491ed47fbdc9c5e

[9] iproov, ”How To Protect Against Deepfakes – Statistics and Solutions” (2022). https://www.iproov.com/blog/deepfakes-statistics-solutions-biometric-protection