Deepfake detection

by Hrushikesh Kedar

Submission date: 25-May-2022 12:25AM (UTC+0530)
Submission ID: 1843437577
File name: Untitled_document.docx (3.41M)
Word count: 6342
Character count: 34201

CHAPTER 01
INTRODUCTION

1.1 Overview

We've seen the rapid growth of Deepfake techniques in recent years, which allow attackers to change the facial region of photos and films, including facial data derived from digital manipulation, particularly with DeepFake approaches. Recently, these methods have piqued the public's interest.

Deepfakes are created by superimposing facial photographs onto a particular person, a video of a source person to create a video of the particular person doing or saying what the source person does. Videos are mostly obtained by putting the visage of one person on the face of another, with the goal of making the video look as real as possible. The effects are so accurate that it appears as if the person who is being replaced appears in the video, and the substituted Actors have been tricked to say things they never said. These kinds of videos endanger a person's privacy, as well as a country's peace and security. The authenticity of excellent DeepFake clips can be difficult to verify with the naked eye on occasion. As a result, researchers will need to create a few algorithms to verify the authenticity of movies. Here is a list of factors that can be used to determine the authenticity of videos.

Facial manipulation techniques can be categorized into different types based on the level of manipulation. Entire Face Synthesis this manipulation creates non-existing face images. These techniques usually archive facial photos of exceptional quality and realism. It will be used in 3D

modeling, gaming but could be used for the creation of fake profiles on social networks for misinformation. Face Swap, The appearance of one individual in a video, is replaced with that of another person in this manipulation. This consists of two categories based on techniques used face-swap, i.e., computer graphics-based technique and Using DeepFakes, which contains the recent mobile application based on deep learning modules to make images and video realistic. Attribute Manipulation contains face editing and retouching, and It consists of modifying some face attributes such as color, hair, eyes, gender, age, etc. A popular example of this is FaceApp mobile application. Expression Swap, this manipulation involves changing the person's face expression. In Figure I, President Lincoln's head and John Calhoun's body, a fake person was constructed (left). Modern tools, such as FakeApp, have made it simple for anyone to create "deepfakes," such as the one that swaps the heads of late-night TV anchors Jimmy Fallon and John Oliver (right).

Figure 1.1: Deepfake Example

Huge strides have been achieved in the domain of computer-assisted video editing in the last several years, and approaches for facial manipulation have sparked a lot of attention. For example, it is now quite simple to execute facial reenactment, or the transfer of facial expressions between persons. This allows a speaker's identity to be changed with little effort. These systems and tools for facial alteration have advanced to the point that even individuals with no prior training in digital arts may utilize them. Indeed, open-source programming and libraries that function almost automatically are becoming increasingly common. On the one hand, technical growth opens the door to previously unexplored territory. Some people, on the other hand, are exploiting these gifts

in the most terrible ways possible. In response to the more complex and realistic manipulated content, the research community is working hard to develop better algorithms for detecting face manipulation. There are many available state-of-the-art methods for deepfake detection like MesoNet, ResNet-50, VGG-19, XceptionNet.

1.2 Motivation

- Face-swapping videos created by artificial intelligence, termed the DeepFakes, causes a threat to security and this problem needs to be taken care of.
- As technology is growing day by day, we are seeing new trends day by day.
- DeepFake videos have grabbed a lot of attention. DeepFake videos can be created by swapping faces or expressions of two people.
- Our motivation is to prevent the most common and harmful DeepFakes that spread false information about a targeted person.

1.3 Problem Definition and Objective

- To survey and analyze the state-of-the-art of DeepFakes generation and detection method and tackle problems like personal attacks, fake terrorism or creating political distress, by video manipulation using an effective deep fake detection model to categorize a video as Real or DeepFake.
- To analyze and evaluate different DeepFakes detection algorithms, compare their performance on different benchmarks.
- To come up with an effective solution, comparing the results that we have got & solving the problem related to deepfakes.

1.4 Project Scope and limitations

Project Scope

- This project focuses on the detection of deepfake videos
- Pleasing UI where one can upload a fake video/ real video and our system will identify and give the response.
- Authentication System for the users

Limitations

- As a video from the internet has many unnecessary parts in it, if we consider all those parts our system accuracy will be decreased. So this system will take time to preprocess the video and finally give a response.
- The system doesn't consider audio manipulations.

1.5 Methodologies of Problem-Solving

- There are many tools are available which can solve given problem, the AI tools like Microsoft's Video Authenticator Tool Deepfake Detection Using Phoneme-Viseme Mismatches
- ExposingDF Videos by Detecting Face Warping Artifacts
- Exposing AI Created Fake Videos by Detecting Eye Blinking
- Exposing Deepfakes using an approach extracting biological signals from facial regions on authentic and fake portrait video pairs

CHAPTER 02

LITERATURE SURVEY

DeepFake is a face manipulation technology based on artificial technology that mainly uses deep learning, which changes the likeliness of one person's face to another in video. Deepfake generation has raised concerns in the field of digital forgery with easy to use ability. Deepfakes are highly realistic and difficult to detect the manipulations on videos. Though we are primarily concerned with the detection of deepfakes, here are some examples of deepfake generation.

2.1. Deepfake Generation

The process of deepfake generation began with traditional vision, and more recent works have included deep learning techniques such as generative adversarial networks. In general, types of deepfake generation are face swapping, face forgery, face reenactment that is puppeteering another face.

2.1.1. Face Swapping

a) Face swapping is the process of replacing a person's identification in a target image with another person's identity in a source image while retaining features such as head posture, facial expression, background, lighting. Face swaps are a graphical technique wherein the target's face is morphed into the source's using facial landmarks (nose, eyes, brows, lips, chin, and cheek regions), and the outcome is post-processed using edge polishing and color correction.

b) A Reddit user released the first face-swapping video created by the deepfake technology in December 2017, shocking the entire world. Deepfake algorithms are said to have been inspired by CNNs to produce face-swapped images.

c) Later, face-swapping videos grew popular worldwide for a number of reasons, both positive and negative. Proposed FaceswapGAN, is a more advanced version of the original deepfake algorithms. To make the autoencoder implemented by VGGFace perform better, perceptual loss and adversarial loss methods were introduced in order to generate more realistic facial expressions.

d) DeepFaceLab, is an open-source deepfake face swapping framework, and was presented as a way to achieve cinema-quality results with great realism using a flexible, easy-to-use pipeline for users without previous in-depth understanding.

e) Recently, proposed Face Shifter in order to achieve high fidelity and occlusion aware face swapping. In the first step, the AEI-Net adaptive combines identity and characteristics in order to synthesize high-fidelity outcomes. In the second step, the HEAR-Net recovers anomalous regions without the need for user annotations.

f) Black-box anti-forensics method suggested for DeepFake videos using Generative Adversarial Network (GAN) model to produce adversarial perturbations; these perturbations can cause substantial performance deterioration in deepfake forensic detectors, causing them to fail.

Figure 2.1.1: Very Famous Movie Interstellar's actor Matthew McConaughey face was replaced by Elon Musk

2.1.2. Face Reenactment

a) Face Reenactment is a face synthesis task that tries to accomplish two tasks at once: 1) transfer a source face form to a target face, and 2) retain the target face's look and identity, used to make the target act and speak in the same way as the source.

b) Face2Face is a face reenactment, GAN-based approach that transfers a source subject's facial expressions to a target while retaining the target's facial landmarks.

2.2 Deepfake Detection

The deepfake detection approach can be viewed as a binary classification process with two options: deepfake or original class. Several approaches for detecting altered videos have been proposed:

- The literature presented a temporal-aware pipeline that employs a convolutional neural network (CNN) to extract frame-level characteristics to automatically recognize deepfake videos. These features are being used to train a recurrent neural network (RNN) to detect whether a video picture has been tampered with. Researchers gathered 600 deepfake videos from a variety of video-hosting websites and picked them at random from the HOHA dataset. The system can reliably recognize whether a fragment is from a deepfake video or not with 97 percent accuracy with less than 2 seconds of video (40 frames for clips captured at 24 frames per second).
- The Literature. MesoNet This research takes a deep learning method and provides two networks MesoInception-4 and Meso-4, both with a few layers, that focus on image mesoscopic features. They tested such rapid networks on a dataset that already exists, as well as a dataset that they created from web videos. The tests show a high success rate of detection, with Deepfake more than 98 percent and Face2Face detecting 95 percent.
- A new multistream deep learning algorithm is developed, where three streams (VGG16, VGG19, ResNet18) merged at the feature level using the fusion layer then fully connected, Softmax, and classification layers are used to classify the deepfakes.

2.3 Algorithm Survey

Following are some state-of-the-art methods for detection of deepfakes

- MESONET NETWORK
- RESIDUAL NETWORK
- VGG NETWORK

- XCEPTION NETWORK

1. MESONET NETWORK

a. MesoNet is a network that combines layers of convolutions, pooling, and a dense classification network to create a high-performing classification network.

b. This neural network starts with four convolution layers and pooling, then moves on to a fully connected thick layer with one hidden layer in the center that is entirely linked.

c. The fully-connected layers use Dropout to regularize their output to improve robustness and take generalization to a new level, while the convolutional layers use ReLU as their activation function to introduce non-linearities and Batch Normalization to regularize their output to avoid the vanishing gradient problem.

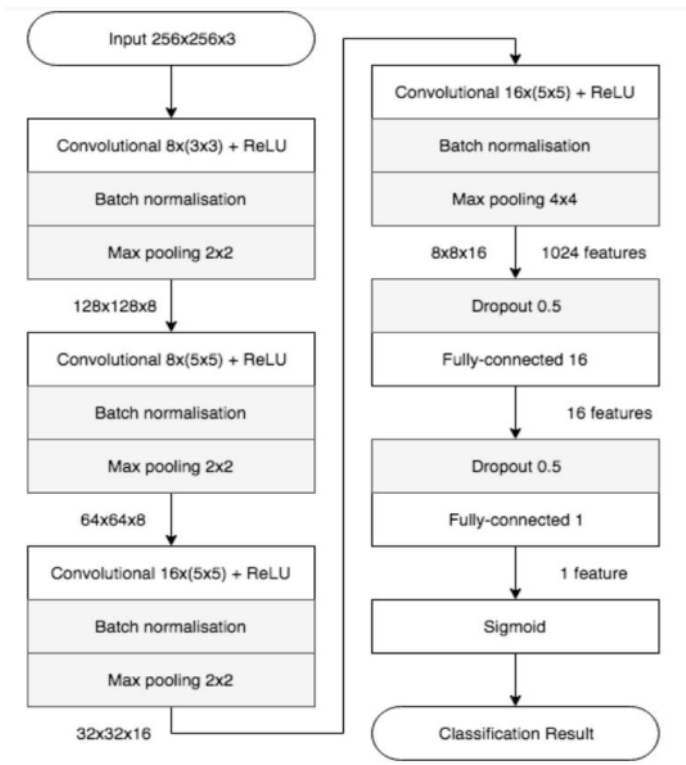


Figure 2.1: Mesonet Network

2. RESIDUAL NETWORK

a. Residual Network, commonly known as ResNet50, is a ResNet variation having 48 Convolution layers, 1 MaxPool, 1 Average Pool layer.

b. It can perform 3.8 billion floating-point operations per second. This is the most frequently used ResNet model.

c. It is feasible to train hyper DNN using this framework, as ResNet resolves the critical problem of vanishing/exploding gradient.

d. ResNet applies identity mapping that is input to some layer is passed directly or as a shortcut to some other layer. Consider the below image that shows basic residual block:

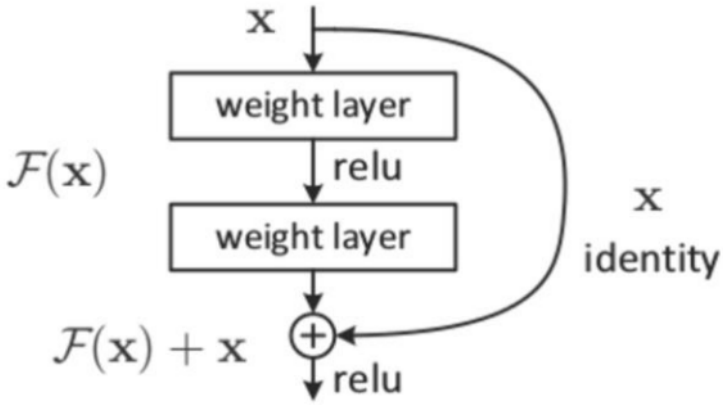
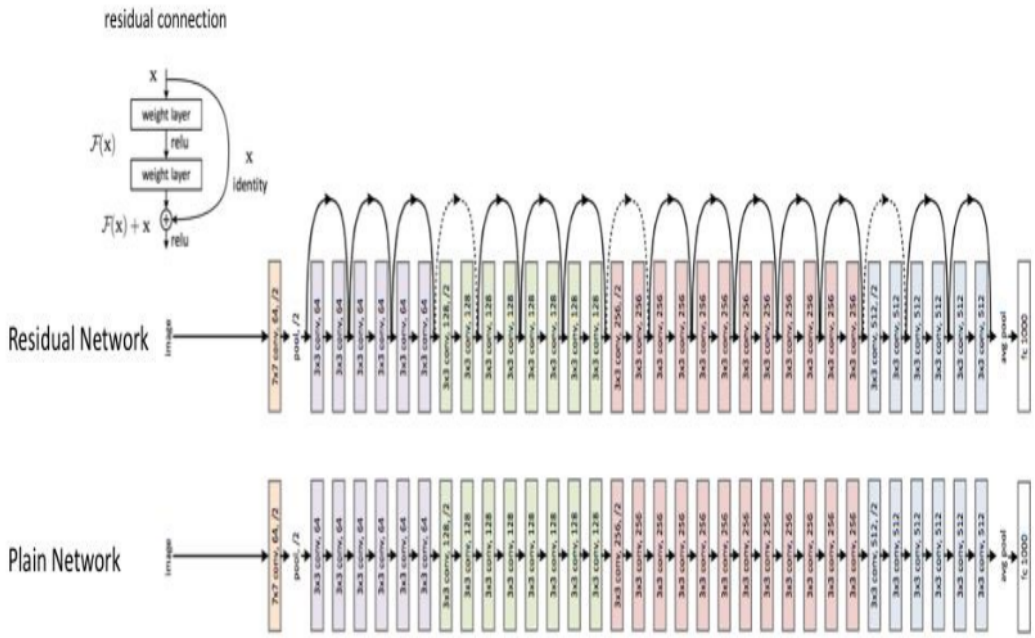


Figure 2.2: Residual Network



e. Skip connection An important concept involved in identity connection is basically the identity mapping where the input from the previous layer is added directly to the output of the other layer.

3. VGG NETWORK

a. Visual Geometry Group Network known as VGG Network has various variants VGG-11, VGG-16, and one of its latest variants VGG-19 consists of nineteen layers that consists of 16 convolutional layers, 3 fully connected layers, 5 MaxPool layers, and 1 SoftMax layer along with capability of 19.6 billion floating points operations.

b. VGG Network is a deep CNN for classifying images.

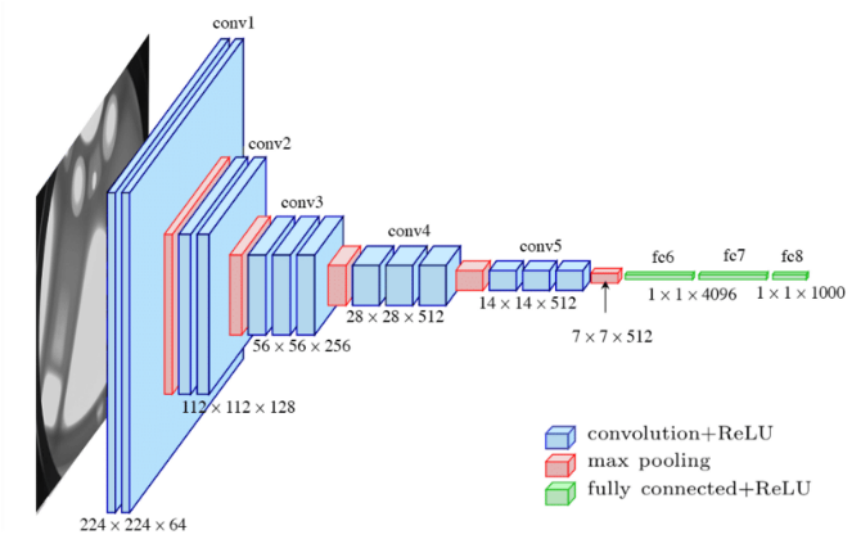


Figure 2.3: VGG19 Network

4. XCEPTION NETWORK

a. In Xception's modified depth wise separable convolution, the point wise convolution is followed by a depth wise convolution.

b. The inception module in Inception-v3 motivated this modification.

c. Xception has three flows: Entry flow, Middle flow, Exit flow. Entry flow includes a classical convolution block and 3 residual separableConv blocks; Middle flow includes 8 residual separableConv blocks; Exit flow includes 1 residual separableConv block and 1 separableConv block.

d. Except the first and last blocks, all blocks have linear residual connections surrounding them to prevent the gradient from vanishing throughout the network's training phase.

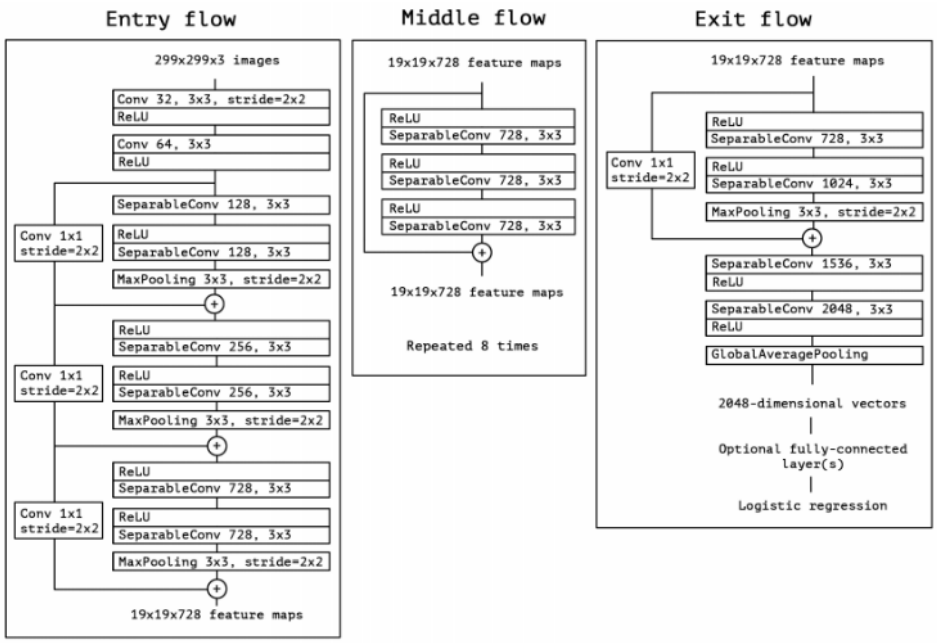


Figure 2.4: Xception Network

Although there are many available techniques due to the increase in the number of deepfakes, still more research needs to be done in this area

CHAPTER 03

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Assumptions and Dependencies

1. Input video must contain the person's face. Face should be properly visible under good lighting conditions
2. The system assumes that the video has no sound and only focuses on the video frames.

3.2 Functional Requirements

3.2.1. Authentication

a. If the user is not registered to our system, He/She has to via Email & Password.
b. Already registered user can log in to our system
c. Users information will be stored in SQLite database

3.2.2. Landing Page

a. Landing page will be the first page displayed to any user
b. All the key features of this system will be mentioned there

3.2.3. Detection Module

a. Detection module is the main feature of our system
b. User can upload a short video to our system
c. Our system will process the video and tell whether the video is Real or Fake, along with its confidence.

3.2.4. Feedback Module

a. The system will prompt the user to with the result, and the user will provide feedback based on the result.

3.2.5. Feedback video upload and re-train phase

a. Based on user feedback, face cropped video of input video is uploaded with the label to the Google's Firebase storage.

b. And uploaded video will go through the training phase.

3.3 Nonfunctional Requirements

3.3.1. Performance requirement

a. Performance of video uploading depends on network connection speed.
b. Once video uploaded, the system should give a result as early as possible
c. Model should be robust with correct accuracy.

3.3.2. Safety Requirement

a. New videos that are being added in the database, their detail should be safe.
b. The system should maintain data integrity by keeping backups of all updates to the database for every record transaction.

3.3.3. Security Requirement

a. Systems should have Hierarchical access to privileged information.
b. Only the users with the role "admin" can view the applicant's details.
c. User data should be secure

3.3.4. Software Quality Attributes

a. The website's interface has to be user-friendly and easy to use.
b. The system should be flexible and scalable.
c. Users should get the result of their input video accurately.
d. The system should maintain data integrity.
e. The system should be portable.
f. The system should be secure.
g. The system should be recoverable.

3.4 System Requirements

3.4.1. Database Requirements

a. User data: Users can register to our system. The user will be added to the user table. The table will have the following columns: UID, username, password (hashed), Last logged in, API secret key.
b. Detection Table: Whenever a user uploads a video to our system for detection. Reference to that video will also be added to the database along with UID.
c. We will store all the data in SQLite database
d. Total training images: 2,194 DFDC - (Real: 1097 | Fake: 1097) Total test images: 2,194 DFDC - (Real: 1097 | Fake: 1097)

3.4.2. Software Requirement

a. Python 3.9
b. IDE: VS Code, Pycharm, Jupyter Notebook
c. Node.js
d. Frontend Framework: React
e. Backend Framework: Django

3.4.3. Hardware Requirement

a. Pc / Laptop
b. CPU: Intel® Core™ i5-8250U or AMD Ryzen™ 5 plus
c. Ram: 8gb +
d. GPU: GeForce ® GTX 1050 (4 GB)

3.5 Analysis Model

- We have used the **Waterfall Model** to maintain track of the project.

- It is really easy to comprehend and use. Each phase must be finished before the next can begin in a waterfall model, and there is no phase overlap. The whole process of software development is divided into separate phases, and these are -

1. **Requirements Analysis**: We've gathered information on numerous datasets, detection APIs, and databases to utilize when determining whether a video is real or fake.

2. **Design**: This system design aids in designing the overall system architecture as well as describing hardware and system requirements

3. **Implementation**: Our solution will be a web application with user authentication, separate methods for retrieving videos, and the most accurate detection of actual and fake videos.

4. **Testing**: During the implementation phase, testing is carried out to ensure that the models employed are accurate.

5. **Deployment**: The web app will be deployed in the client environment once functional and non-functional testing is completed.

6. **Maintenance**: In the client environment, there are a few challenges that arise. In order to improve the existing, newer versions will be produced. Maintenance will be carried out in order to bring about these modifications for the best accuracy.

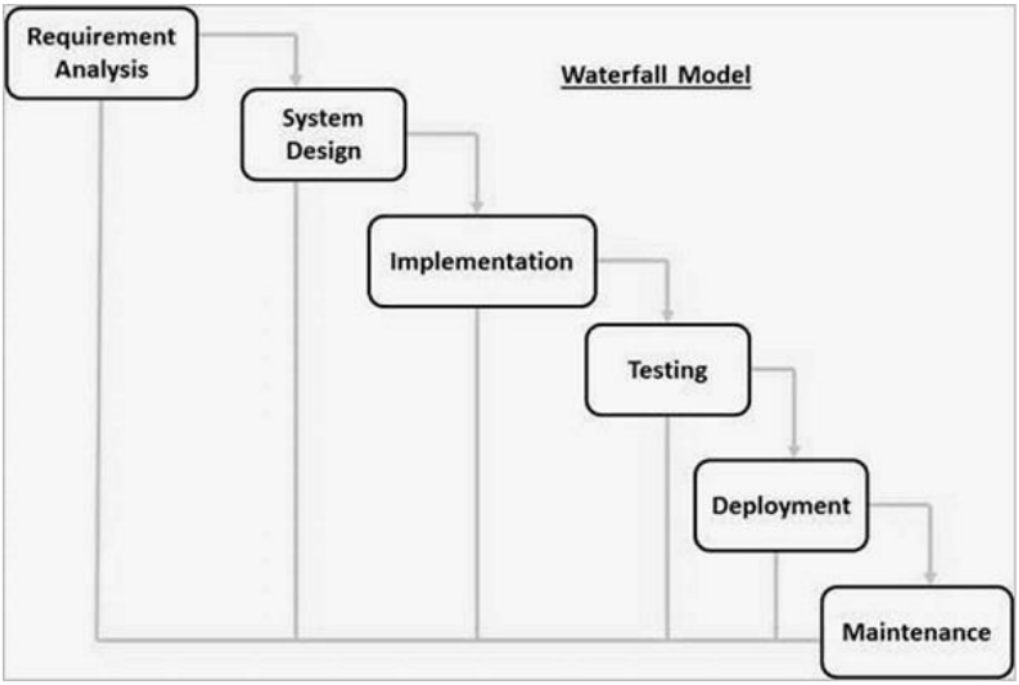


Figure 3.5: Waterfall Model

CHAPTER 04

SYSTEM DESIGN

4.1 System Architecture

There are several tools available for constructing the Deepfakes(DF), however there are few for detecting the DF. We will provide a web-based platform for users to post videos and determine if they are fake or real. Even large applications like Instagram and YouTube can integrate this project into their applications for quick deepfake detection. Users can also provide feedback, if they already knew about the video. This project can be further scaled to multiple platform as we are providing an API to detect videos. One key goal is to assess its performance and acceptability in terms of security, usability, correctness, and dependability. Our technique is designed to detect all kinds of DF, including replacement, retrenchment, and interpersonal DF.

Our model is divided into 6 major phases, which are preprocessing phase, training phase, feature extraction phase, testing phase, detection phase & feedback phase.

Let's understand this models in brief, First phase will be preprocessing phase, here we will extract face cropped images from the video file. In the training Phase, a Neural network will be trained on a set of face cropped images obtained from the preprocessing phase, the output of the training phase will be a trained model which will be used further in feature extraction phase. The third phase is a feature extraction phase, input to this phase is the trained model from training phase, output will be CSV file including feature vectors of the images, we will explain the module in detail in later part. The fourth phase will be testing phase. We will apply machine different machine learning algorithms on input dataset obtained from the feature extraction phase. The fifth phase will be the detection phase. Here we will apply the machine learning algorithm which is giving the better results on the user input. Last phase will be the feedback phase, if the user already knew about the video and our prediction is wrong, we will review that video and retrain the model.

So this system is an incremental system, meaning it will get better and better if more and more videos are fed to it. All these six phases are equally important for this system to work properly.

1. Preprocessing
2. Training phase
3. Feature extraction phase
4. Testing phase
5. Detection phase
6. Feedback phase

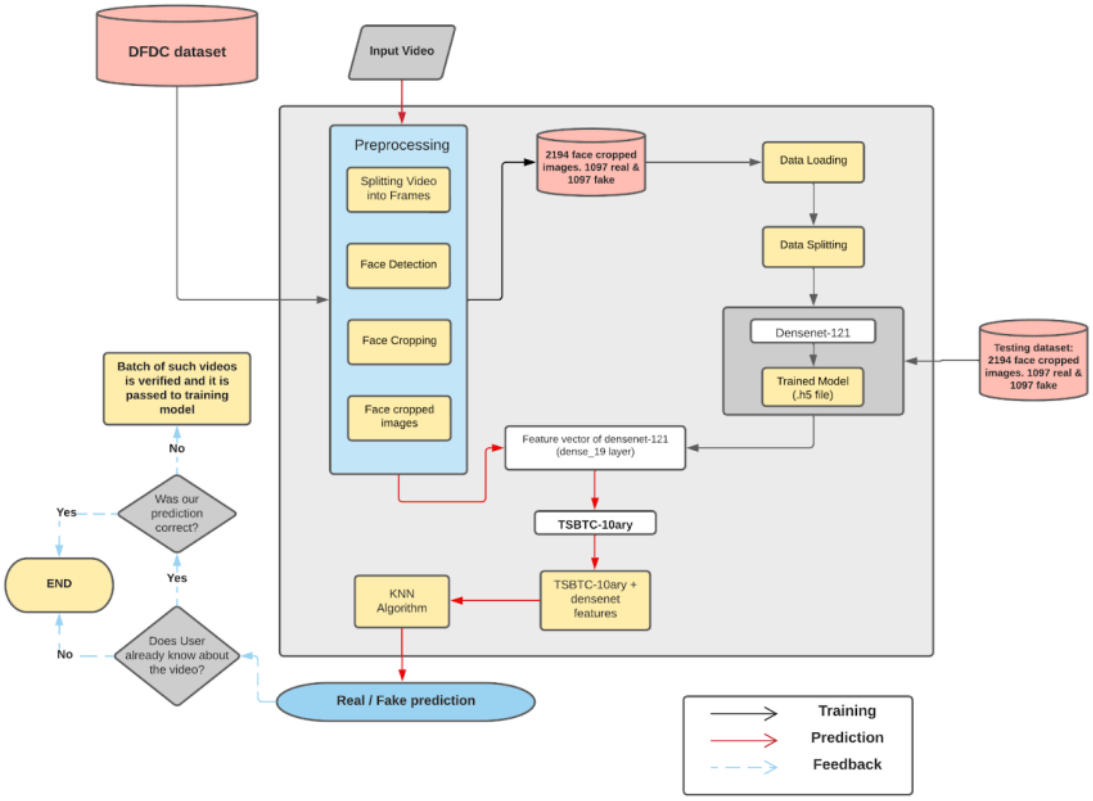


Figure 4.1: System Architecture

4.2 Flow Diagram

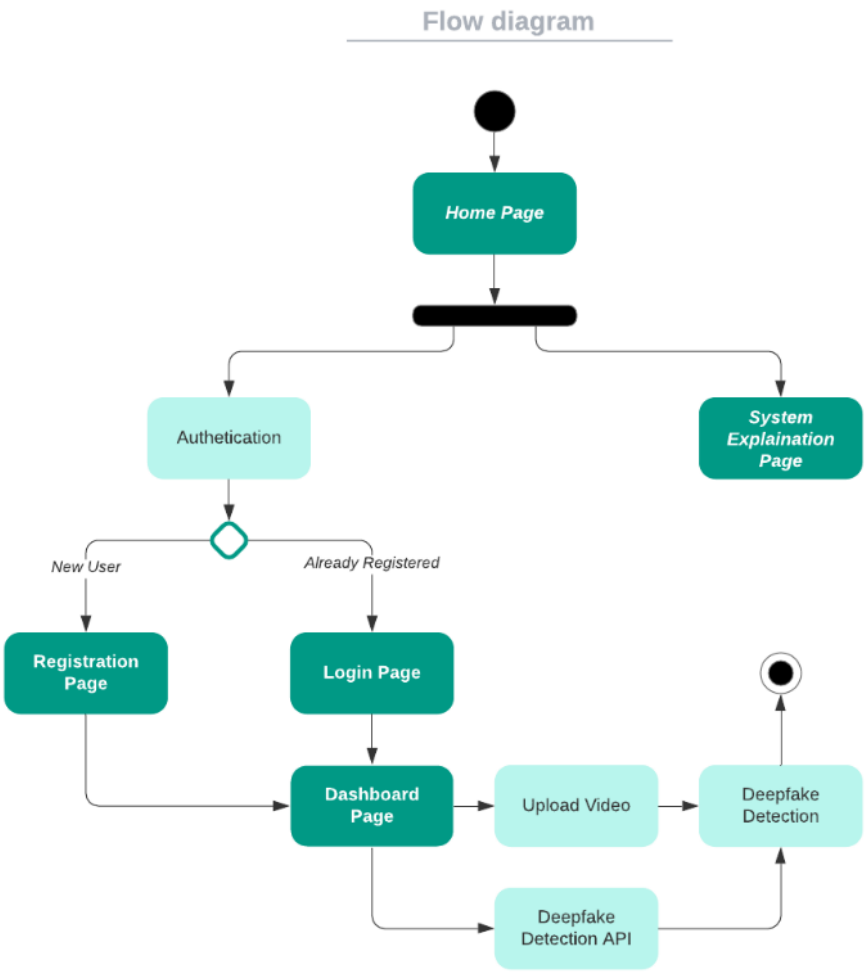


Figure 4.2: Flow Diagram

4.3 Activity Diagram

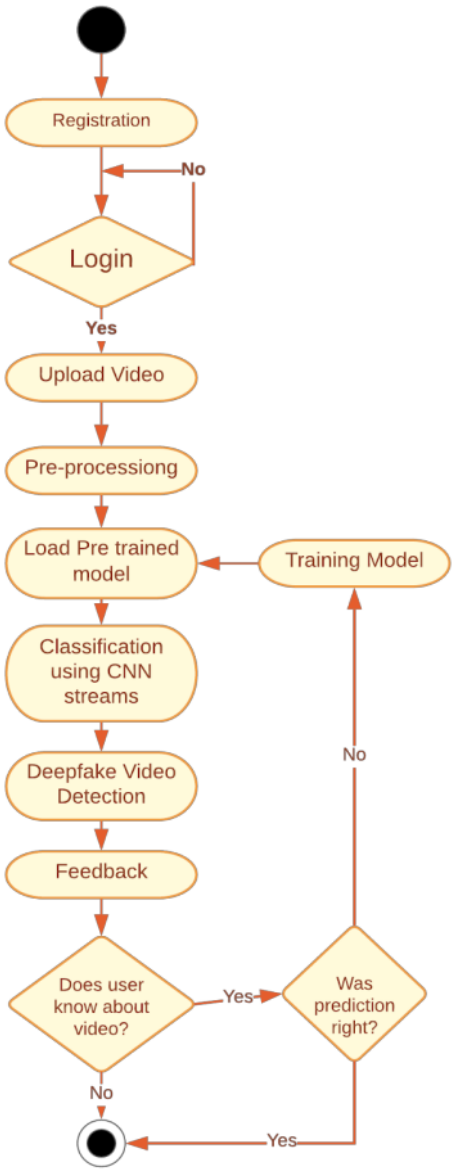


Figure 4.3: Activity Diagram

4.4 Use Case Diagram

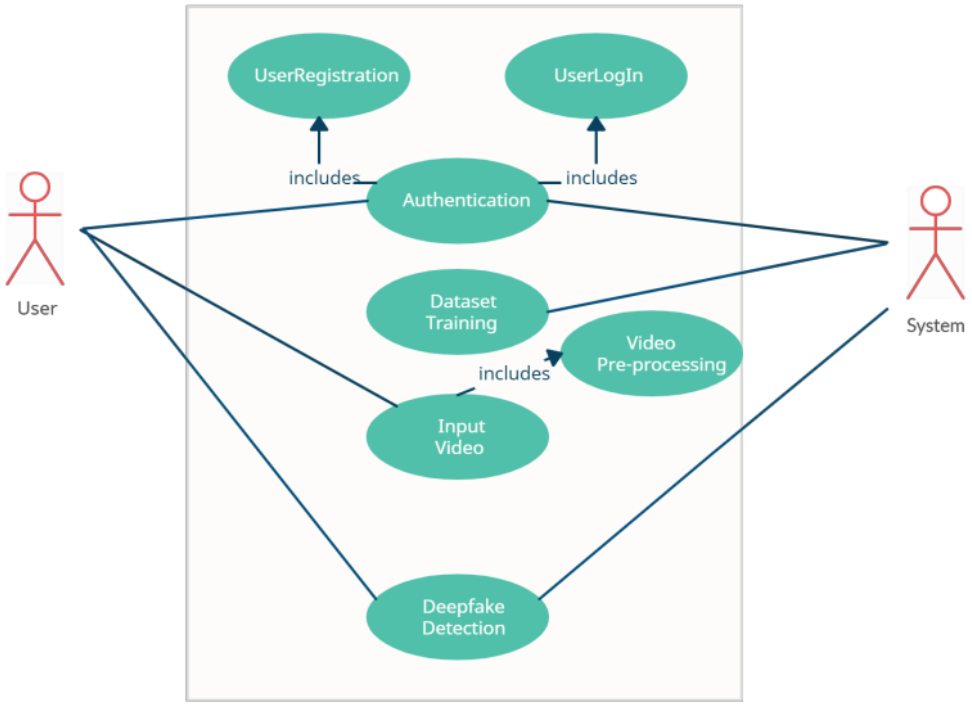


Figure 4.4: Use Case Diagram

4.5 Class Diagram

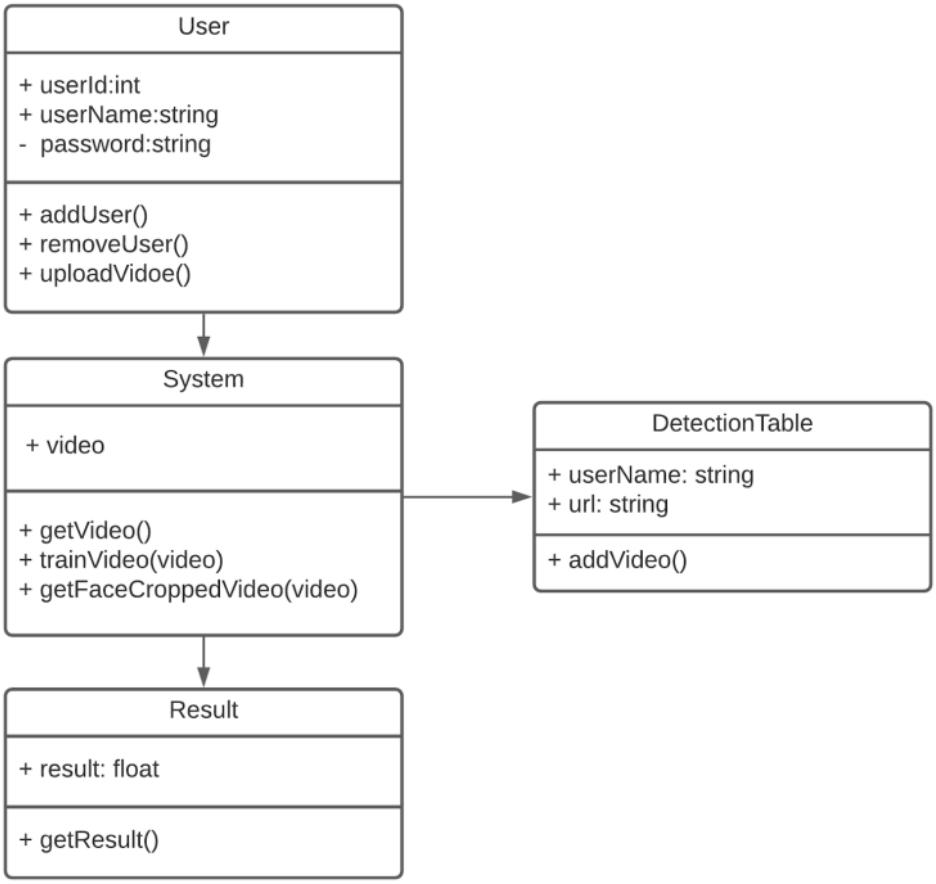


Figure 4.5: Class Diagram

CHAPTER 05

PROJECT PLAN

5.1 Project Estimate

5.1.1 Reconciled Estimates

In this chapter we are going to have an overview about how much time does it took to complete each task like- Preliminary Survey Introduction and Problem Statement, Literature Survey, Project Statement, Software Requirement and Specification, System Design, Partial Report Submission, Architecture Design, Implementation, Deployment, Testing, Paper Publish, Report Submission and etc

5.1.1.1 Cost Estimate

Sr No	Hardware/Software	Quantity	Cost
1.	Python IDE/ Google Colab, VSCode	1	Open Source
2.	Windows 10 OS	1	-
3	Firebase Cloud	1	
		Total	0.0/-

Table 5.1.1.1 Cost Estimates

5.1.1.2 Time Estimates

Month	Activity
July 2021	Searching and selection of domain and topic
August 2021	Literature survey, problem definition
September 2021	Requirement gathering & planning for the project
October 2021	Planning of the project
November 2021	Analysis and design the GUI of the project

December 2021	Construction/implementation of first project module
January 2022	Implementation of second project module
February 2022	Construction/implementation of third project module
March 2022	Testing modules to meet the requirement of project
April 2022	Project report making in Google Docs
May 2022	Submission of project report

Table 5.1.1.2 Time Estimates

5.2 Risk Management

5.2.1. Risk Identification

Risk is an inflexible and unavoidable aspect of the product development process, which develops over time and influences a project, programming, or both. As a result, the need to deal with and handle these threats in a productive and compelling manner emerges. When something goes wrong with a deep learning application, the challenges and opportunities created are frequently more bothersome than the reason. Gambles must also be identified and managed as soon as possible during the task. Risk identification takes place throughout the project life cycle, with a special emphasis on the critical milestones. One of the most important aspects of the normal undertaking status and revealing gatherings is risk identification.

5.2.2. Risk Analysis

Risk examination entails looking at how project outcomes and goals might change as a result of the risk event. When risks are identified, they are examined to determine the risk's subjective and quantitative impact on the work, so that appropriate steps can be taken to mitigate them.

5.2.3. Overview of Risk Mitigation, Monitoring, Management

Risk mitigation is a way for preparing for and reducing the effects of risks identified by the center. Risk mitigation, like risk reduction, does everything it takes to minimize the negative effects of risks and disasters on business continuity. Digital attacks, weather events, and other causes of physical or virtual harm to a server farm are all potential threats to a firm. Risks should be revisited at regular intervals for the group to examine each risk and determine when new circumstances cause its likelihood and effect on change. Proactive activities are possessed by people and observed as a key to good risk management.

5.3 Project Schedule

5.3.1. Project Task Set

Project Task Set Major Tasks in the Project stages are:

Task 1: Requirement Analysis.
Task 2: Project Specification (Paperwork).
Task 3: Technology Study and Design.
Task 4: Coding and Implementation (Module Development)

5.3.2. Task Network

Schedule		**Project Activity**
August	1st week	Requirement Analysis
	2nd week	Topic Finalization
	3rd week	Feasibility Assessment
September	1st week	Mid Sem Presentation
	2nd week	Submission of Literature Survey
	4th week	End Sem Presentation

October	1st week	Report Preparation and Submission
December	4th week	1st Module presentation
January	1st week	Discussion about further strategy of developing Prototype
	2nd week	2nd Module Implementation followed by presentation
February	3rd week	Design test cases for module
	4th week	Developing User Interface
March	2nd week	Integration all modules including interactive GUI

Table 5.3.2 Timeline Network

5.3.3. Timeline Chart

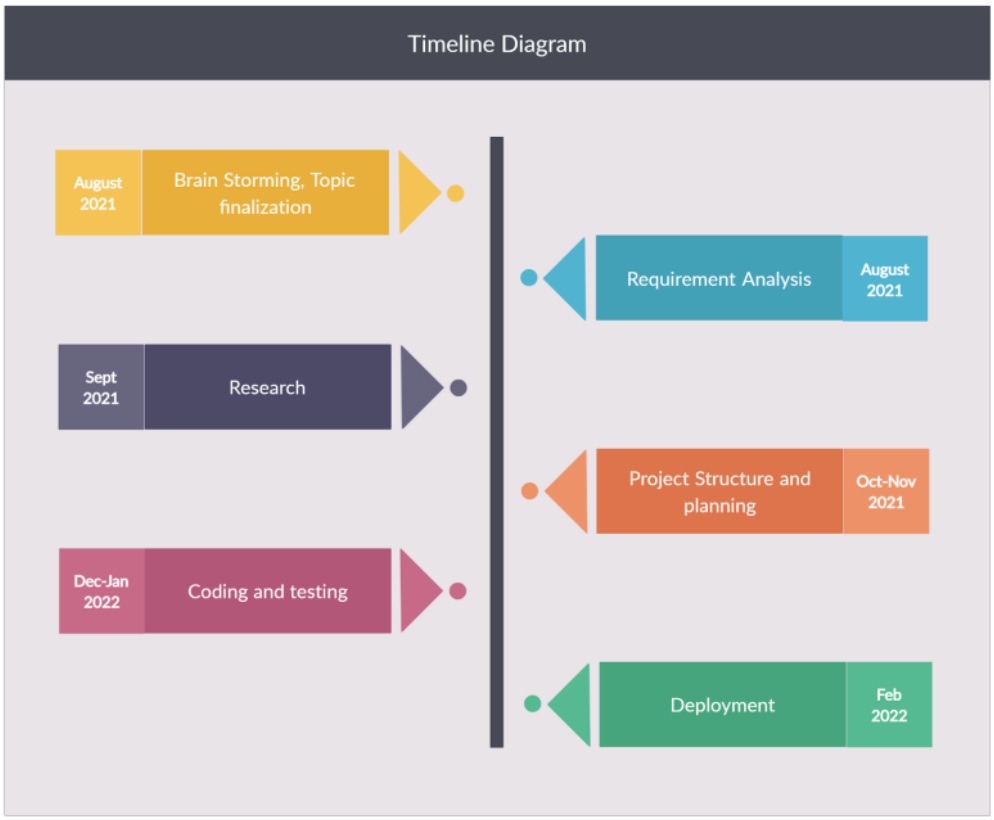


Figure 5.3: Timeline Diagram

5.4. Team Organization

5.4.1. Team Structure

Guide name: Prof. Smita Khairnar

Members: Sayyam Dhokane

Vaishnavi Kale

Sushil Kamble

Hrushikesh Kedar

5.4.2. Management reporting and communication

First four weeks were spent in searching and selection of domain and project topics followed by literature survey and formation of problem definition.Within the span of two to three weeks, we could identify the details that would be useful for our dataset creation and would be feasible to work. The Next five weeks were spent in analyzing and selecting the proper machine learning algorithm. Through this period we have analyzed various machine learning algorithms which will give proper accuracy and correct prediction. After the Analysis was done the next three weeks were spent on implementing and testing each module in python using Python IDE/ Google Colab Notebook and VSCode. Next seven weeks were spent on designing, implementing, and testing the prediction module in python using Python IDE/ Google Colab Notebook. Next four weeks were spent on analyzing the system and designing test cases to uncover defects in the system. Last two weeks were spent in documentation

CHAPTER 06

PROJECT IMPLEMENTATION

6. PROJECT IMPLEMENTATION

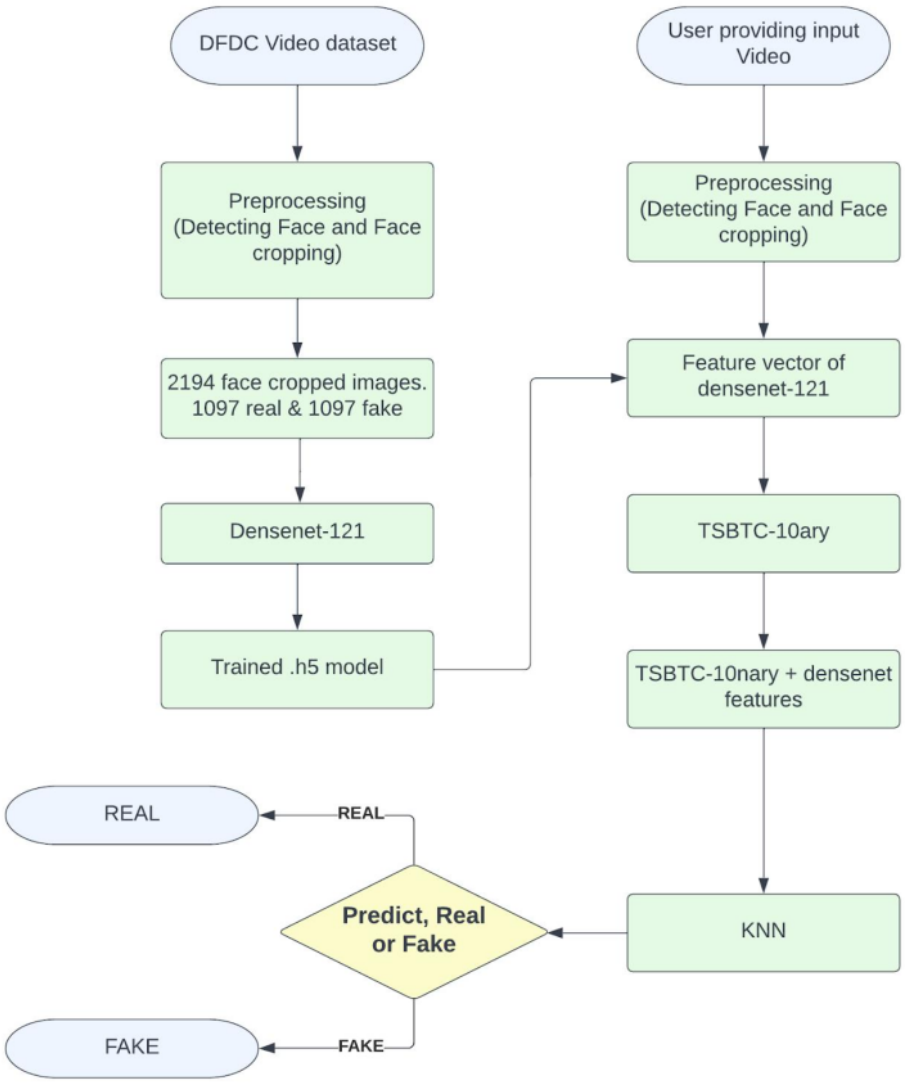


Figure 6.1.1: System flow

As shown in the above diagram, we take videos from the DFDC dataset and preprocess them by cropping faces. We have 2194 faces, 1097 of which are real and 1097 of which are fake videos,

which we feed to the Densenet121 model. We select feature vectors and add them with tsbtc n array features, after which we give the features to the KNN model to predict whether the video is real or fake.

6.1 Overview of Project Modules

6.1 Preprocessing

6.2 Training phase

6.3 Feature extraction phase

6.4 Testing phase

6.5 Detection phase

6.6 Feedback phase

6.1 Preprocessing

Preprocessing is the first stage of the project implementation. For preprocessing, videos are taken from the DeepFake Detection Challenge (DFDC) Dataset. Format of videos is .mp4. The length of the videos is 10 sec. Frames per second (fps) is 30, so in total we will have 300 frames per video. Approx 15 frames are extracted, keeping a gap of 20. Face cropped images are obtained using python's face_recognition module. Images where face is not detected are ignored. So we have approx 15 face cropped images per video at the end of this module.

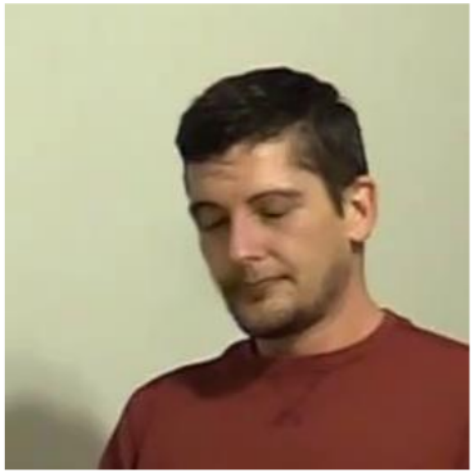


Figure 6.1.2: Preprocessed Face cropped images

6.2 Training phase

We have taken the equal number of real and fake images for training. Images are obtained from preprocessing phase. There are total of 2194 face cropped images out of those 1097 are Real and 1097 are Fake. Each image is having height and width as 300px. We have taken 80% of images for training and 20% for validation. Algorithms we applied on training dataset are Resnet, DenseNet-121, VGG-16, VGG-19 & Inceptionv3, ResNext. Top layer was not included in all these algorithms.

Following layer were added

```
Dense(512, activation='relu', input_dim=(IMG_HEIGHT,IMG_WIDTH,3)
Dropout(0.3)
Dense(512, activation='relu')
Dropout(0.3)
Dense(1, activation='sigmoid')
```

Input to this model is 2194 face cropped images & output will be the trained model. Trained model is a .h5 file which will be used in feature extraction phase

	Resnet	DenseNet-121	VGG-16	VGG-19	Inceptionv3	ResNext
Accuracy	48.125	82.326	71.560	69.464	78.123	50
Loss	70.001	43.872	55.764	59.901	49.503	55.58

Table 6.2: Testing accuracy & loss of different deep learning algorithms

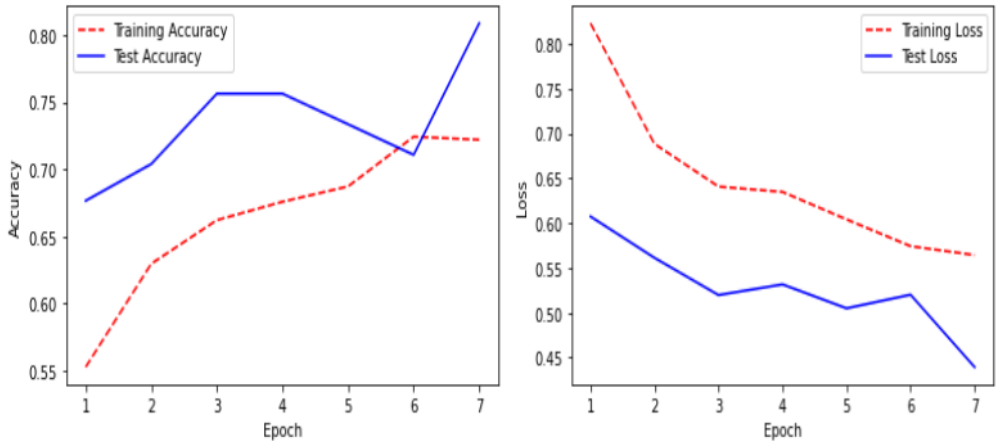


Figure 6.2.1 Densenet-121 Training and loss vs epochs

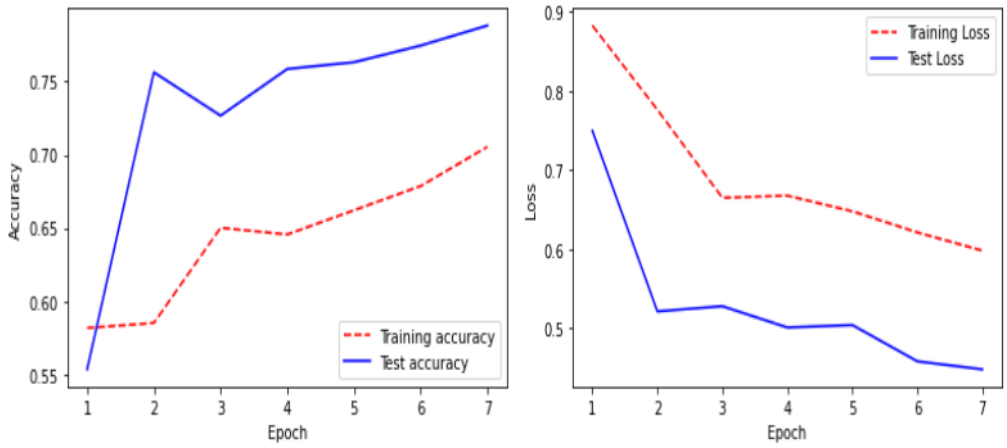


Figure 6.2.2 Inception Training and loss vs epochs

Densenet-121 gave the highest testing accuracy of 82.326 % on 7 epochs. Followed by Inception which gave testing accuracy of 78.123%.

6.3 Feature extraction phase

In feature extraction phase, we take the trained model & we take different set of images (training set). Training set also has total of 2194 images out of them 1097 are real and 1097 are fake.

Features are extracted from the intermediate level. We have again applied different deep learning algorithms to extract features of intermediate level & calculated accuracy using different machine learning model on the extracted features. Below is the table of accuracy for different machine learning algorithms vs deep learning algorithms.

	SVM	Decision Tree	Random Forest	Naive Bayes	Logistic Regression	KNN	Ada Boost
Densenet	51.06	52.12	50.6	50.75	50.91	52.73	48.17
Resnet	48.176	51.367	49.848	52.735	48.328	51.823	52.735
ResNext	48.176	49.240	49.392	50.151	51.063	48.480	47.872
Vgg-16	52.279	48.936	51.063	48.328	50.151	50.911	0.5
Vgg-19	49.088	48.328	48.328	47.112	50.911	48.784	48.632
Inceptionv3	50.151	51.823	49.696	50.303	49.240	52.127	49.848

Table 6.3: Feature vector training accuracy of deep learning algorithms

After extracting 512 features for each image. We have combined those features with TSBTC-nary features. We have tested for 2ary, 3ary, 6ary, 10ary. We have added results in result section of the report.

6.4 Testing phase

Input to this phase are features extracted from the deep learning model & TSBTC-nary features. We can see the table no. 6.2, densenet-121 was giving the highest accuracy (72%). So we have considered it and evaluated features on different machine learning algorithms and compared their accuracy. We have taken the best combination that we can find, comparing the accuracy.

6.5 Detection phase

Input to this phase will be video uploaded from our application frontend by the user. Face cropped images will be extracted from the uploaded video. We have decided that we will extract 5 frames from the input video. We will use the pretrained model on densenet-121 to extract features from 'dense-19' layer. Dense-19 layer has 512 features. So 512 features will be extracted for every image. We will then pass those images through TSBTC-10ary as it was giving the best accuracy. We will then concatenate densenet-121 features with tsbtc features. Furthermore, we have seen that KNN was giving the highest accuracy in the testing phase. So, we will pass those concatenated features through KNN to get the prediction. For every frame, we will get prediction whether the video is real or fake. We will consider the count which is high as our result.

6.6 Feedback phase

After the prediction phase is completed i.e. system has given output whether the video is real or fake, the User will be prompted whether he/she knows about the video. If the answer is yes and our prediction is wrong. Video will be marked, marked videos will be fed to model. Once the user gives the feedback, uploaded video will be again uploaded to Firebase storage with its correct label. Video files can be queued into Firebase storage. Model can be again trained on previous data and the new data from feedback phase. New can be queried from Firebase storage and appended to existing data. Do this will make the model better and better.

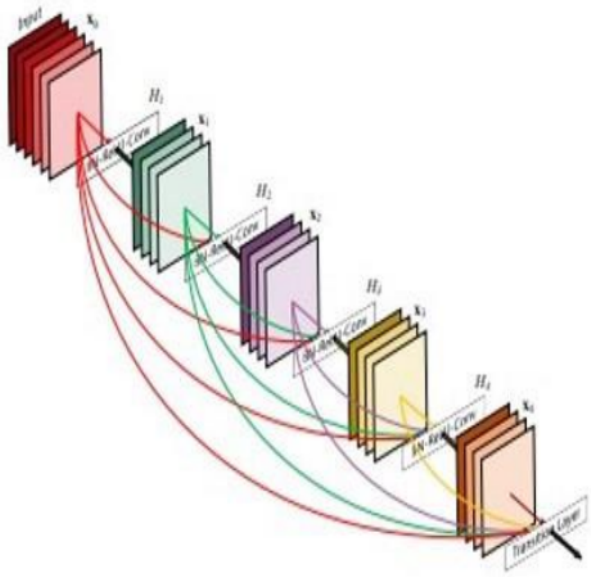
6.7 Algorithm Details

Almost all CNN designs use the same fundamental design concepts, which include applying convolutional layers to the input sequentially and occasionally down sampling the spatial dimensions while increasing the number of feature maps. While traditional network designs consisted solely of stacked convolutional layers, newer architectures look at new and novel ways of constructing convolutional layers in order to improve learning efficiency.

We extracted frames from videos taken from the DFDC dataset. The frames are sent to the densenet121 model, which extracts frame features before applying TSBTC-10ary features. The features are then fed into the KNN method, which is used to train the model.

6.7.1 DenseNet

DenseNet (Dense Convolutional Network) is a network architecture that focuses on deepening deep learning networks while also making them more effective to train by employing shorter connections between layers. DenseNet is a convolutional neural network in which each layer is connected to all other layers deeper in the network, so the first layer is connected to the 2nd, 3rd, 4th, and so on. This is done to allow maximal information flow between network tiers. To maintain the feed-forward nature, each layer receives input from all previous layers and sends its own feature maps to all subsequent layers.



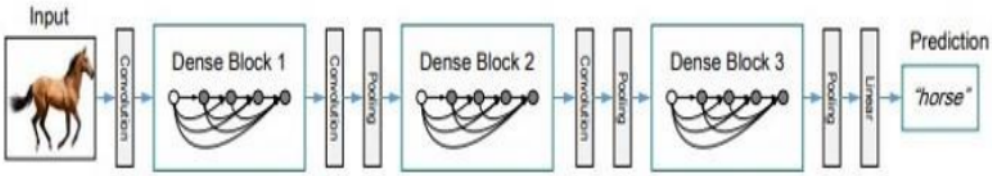


Figure 6.7.1: DenseNet

6.1.1. TSBTC Feature Extraction

In TSBTC-2 ary, given aerial image AE (v,w) of size 'rxc' pixels, global features are extracted using Thepade's Sorted Block Truncation Coding (Thepade SBTC). The colour planes R, G, and B are turned into a one-dimensional array and sorted as sortR, sortG, and sortB. The TSBTC-2ary feature vector is computed as [TR1, TR2, TG1, TG2, TB1, TB2] using these sorted one-dimensional color planes.

$$TR_1 = \frac{2}{rxc} \sum_{p=1}^{\frac{rxc}{2}} sortR(p) \tag{1}$$

$$TR_2 = \frac{2}{rxc} \sum_{p=1+\frac{rxc}{2}}^{rxc} sortR(p) \tag{2}$$

$$TG_1 = \frac{2}{rxc} \sum_{p=1}^{\frac{rxc}{2}} sortG(p) \tag{3}$$

$$TG_2 = \frac{2}{rxc} \sum_{p=1+\frac{rxc}{2}}^{rxc} sortG(p) \tag{4}$$

$$TB_1 = \frac{2}{rxc} \sum_{p=1}^{\frac{rxc}{2}} sortB(p) \tag{5}$$

$$TB_2 = \frac{2}{rxc} \sum_{p=1+\frac{rxc}{2}}^{rxc} sortB(p) \tag{6}$$

Figure 6.7.2: TSBTC-2ary feature vector

6.1.2. KNN Algorithm

K-Nearest Neighbour is a Supervised Learning-based Machine Learning algorithm that is one of the most basic. The K-NN algorithm assumes that the new case/data and existing cases are similar and places the new case in the category that is most similar to the existing categories. The K-NN method maintains all of the available data and classifies a new data point based on its similarity to the existing data. This means that new data can be quickly sorted into a suitable category using the K-NN method. The K-NN algorithm can be used

for both regression and classification, but it is more commonly used for classification tasks. The K-NN algorithm is a non-parametric algorithm, which means it makes no assumptions about the data.

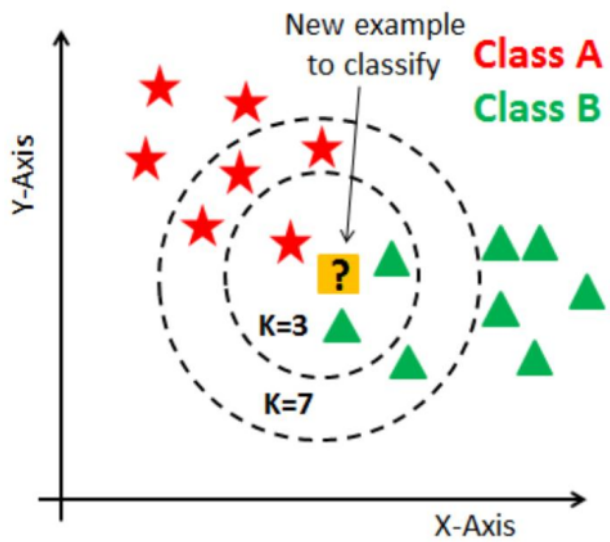


Figure 6.7.3: k-nearest neighbors (KNN) algorithm

CHAPTER 07

SOFTWARE TESTING

7. SOFTWARE TESTING

7.1 Manual Testing

We have used a Feedback Mechanism in our web app where the user can upload video and if our model predicts wrong output, the user can give feedback so that the same video can be used again to retrain the model.

7.2 Test case & Test Results

System Testing: The purpose of the test is to detect errors and bugs. Testing is the process of trying to identify every fault or error or weakness in a product. Provides a way to evaluate the performance of various components, subassemblies, subcomponents, and / or a finished product. There are different types of tests. Each type of test meets a specific testing requirement. UI was tested using Selenium Web driver & below are the results of the UI test.

Test case Id	Test case	Expected Result	Actual Result	Status
1	User registration: username: sushil email: iamsushil303@gamil.com password: test@12345 confirm password: test@12345	User registered successfully	User registered successfully	PASS

2	User registration: username: sushil email: iamsushil303@ gamil.com password: test@12345 confirm password: test@123	Password & confirm password not same	Password & confirm password not same	PASS
3	User Login: username: sushil password: test@12345	User logged in successfully	User logged in successfully	PASS
4	User Login: username: sushil password: test@12	Username or password incorrect	Username or password incorrect	PASS
5	User Login: username: abc password: test@12345	User doesn't exist	User doesn't exist	PASS
6	Upload video: filename: video.mp4	Results dialog box appear	Results dialog box appear	PASS
7	Upload video: filename:	Select video to continue	Select video to continue	PASS

Figure 6.7.2: Test cases and results

CHAPTER 08

RESULTS

8.1 Outcomes

Dataset distribution with label:

Real Images: 1097

Fake Images: 1097

	SVM	Decision Tree	Random Forest	Naive Bayes	Logistic regression	KNN	Ada Boost
Densenet + TSBTC 2arr	85.106	60.182	84.954	51.367	78.115	97.872	89.665
Densenet + TSBTC 3arr	57.142	92.857	62.006	51.823	75.531	52.887	93.313
Densenet + TSBTC 6arr	98.936	99.10	94.224	58.966	98.632	93.768	98.89
Densenet + TSBTC 10arr	98.088	**99.15**	83.434	59.878	98.784	94.376	98.764

Table no 8.1: Accuracy matrices with Densenet-121 and TSBTC 2, 3, 6 and 10 array

Observations

The accuracies for each Machine learning method are displayed in the following table with densenet121 and TSBTC n-array and we discovered that the K-Nearest Neighbor model has the greatest accuracy 99.15 percent with the Densenet121 and TSBTC 10-array.

	SVM	Decision Tree	Random Forest	Naive Bayes	Logistic regression	KNN	Ada Boost
precision	98.443	99.151	85.616	61.572	98.432	93.750	98.136
recall	99.684	99.457	78.864	44.479	99.053	94.637	98.486
f1-score	99.059	99.786	82.101	51.648	98.742	94.191	98.756
support	317	317	317	317	317	317	317

Table no 8.2: Performance matrices with Densenet-121 and TSBTC 10 array

8.2 Screen Shots

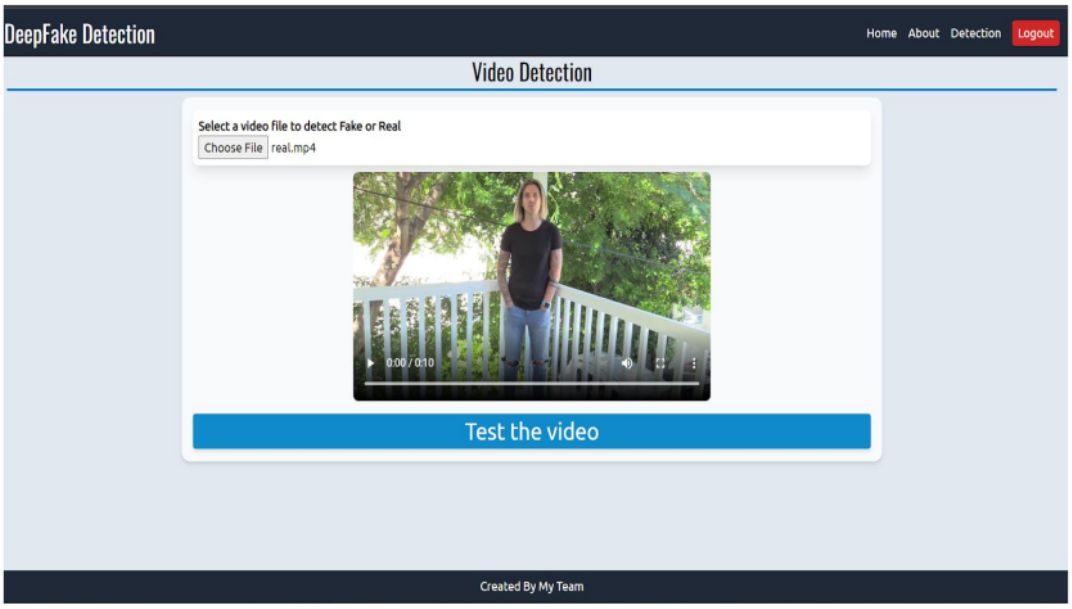


Figure 8.2.1: Detection page UI Screenshot

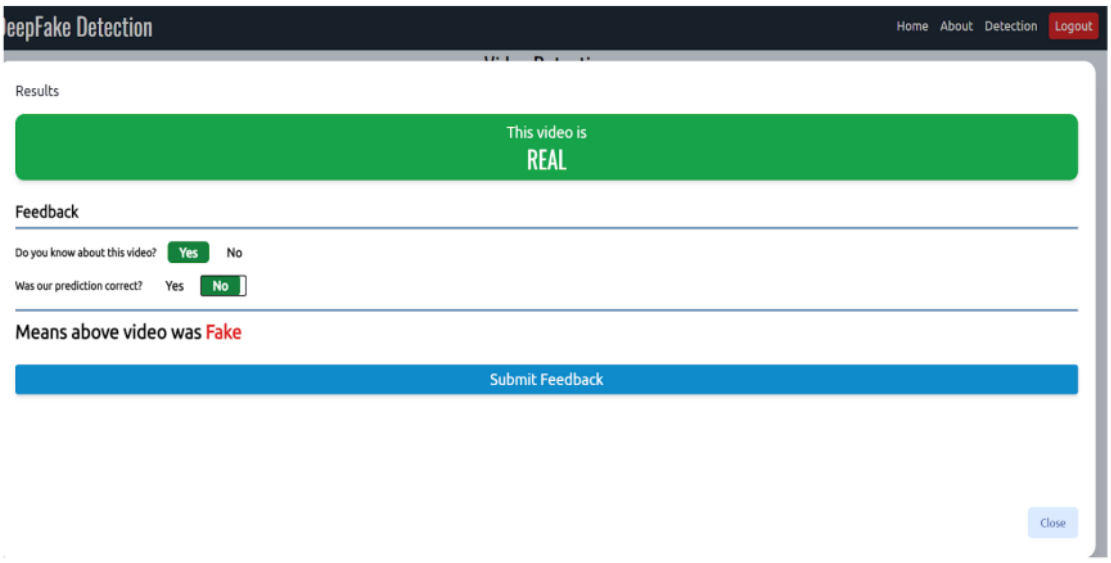


Figure 8.2.2: Result page UI Screenshot

CHAPTER 09

CONCLUSION

9.1 Conclusion

In this digital age, everyone should be extremely cautious about exposing personal information on any public forum, as this information could be misused, causing damage to one's reputation and image. Every technology has both a positive and negative aspect. We should keep this in mind while taking any action on social media because identifying the source has become quite difficult. The development of deepfake pictures has already caused significant difficulties and has the potential to become even more of a problem as technology advances. Image and video editing technology have advanced significantly over the years, and the development of deepfake pictures has already caused significant difficulties and has the potential to become even more of a problem as technology advances. It's currently impossible to distinguish between a real video and a computer-generated deepfake, and deepfakes are often undetectable to the naked eye.

Machine learning and artificial intelligence algorithms have been employed to help determine the authenticity of digital information and have produced accurate and dependable results. We investigated and discovered many strategies used to detect deepfakes. With diverse datasets, we compared some of the most cutting-edge deepfake detection algorithms in research now accessible. We've provided details on the most commonly used datasets, as well as approaches for generating and detecting deepfakes.

Because deepfake video generation techniques are constantly evolving, additional work is required to improve existing detection algorithms. For face detection, We plan to develop a more reliable deep-learning-based deepfake detection approach in order to stay up with developments in the deepfake generating process.

9.2 Future Work

We need to perform a few things in the future to improve the algorithm and predictions.

9.2.1 Enhancements to the dataset

Deepfake detection tools integrated into social media sites can help us to improve the quality and quantity of datasets. People utilize deepfake generation techniques on a daily basis to create deepfake videos on social media, and we can use such videos to improve the datasets.

9.2.1 Deepfake detection

Using the techniques provided to detect deepfakes, we can utilize the Ensemble methods methodology, which combines numerous base models to generate one ideal predictive model. Also, using social media to integrate deepfake detection techniques will increase the technology's ability to deal with the widespread proliferation of deepfakes on social media and limit their impacts.

9.3 Applications

9.3.1 Automation Industry

- Online video apps like Instagram, Facebook, YouTube can use our API when someone uploads a video to these Apps / Websites this video will be processed and if it is fake they can add a disclaimer that this video is fake, such that normal audience not be affected.
- The person who is uploading these fake videos can also be identified

9.3.2 Personal Use

- Users can process multiple videos, extract faces from them and store these videos.
- These videos can be used for training a model such that it can give a high accuracy.

9.3.3 Identity cyber crimes

- If a random video is circulated targeting a person, this system will give justice to that person.

Deepfake detection

ORIGINALITY REPORT

8%	8%	%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ijircce.com Internet Source	2%
2	deepai.org Internet Source	2%
3	arxiv.org Internet Source	1%
4	www.irjmets.com Internet Source	1%
5	iq.opengenus.org Internet Source	1%
6	Www.Ijeast.Com Internet Source	1%
7	abhishek-shaw.medium.com Internet Source	1%
8	winatalent.com Internet Source	1%
9	www.ijtsrd.com Internet Source	1%

Exclude quotes	Off	Exclude matches	< 1%
Exclude bibliography	Off		