DEEPFAKE DETECTION USING

DEEP LEARNING

A PROJECT REPORT

Submitted by

GURUNATHAN M

(2019202015)

submitted to the Faculty of

INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfillment for the award of the degree

of

MASTER OF COMPUTER APPLICATIONS

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITYCHENNAI

600 025

MAY 2022

ANNA UNIVERSITY

CHENNAI - 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled **“DEEPFAKE DETECTION USING DEEP LEARNING”** is the bonafide work of **GURUNATHAN M(2019202015)** who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE : CHENNAI

DATE : 30.05.22

DR.L.SAIRAMESH

PROJECT GUIDE

DEPARTMENT OF IST, CEG

ANNAUNIVERSITY 600025

COUNTERSIGNED

Dr.S. SRIDHAR

HEAD OF THE DEPARTMENT

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY

CHENNAI 600025

ABSTRACT

With the recent developments on the creation of deepfake videos using Generative Adversarial Network (GAN), which can produce realistic photos and videos, the reliability of digital images is becoming more challenging to identify. This research is an approach to develop a deep learning model which can efficiently distinguish between a deepfake and a real video. Deep learning is continuously evolving a lot in both areas of generating and detecting deepfakes. A model developed for detection of deepfake designed with older dataset may expire in time, and a need for new detection technique will always be there.

This system uses a Res-Next Convolution neural network to extract the frame-level features and these features and further used to train the Long Short Term Memory(LSTM) based Recurrent Neural Network(RNN) to classify whether the video is subject to any kind of manipulation or not, i.e whether the video is deep fake or real video. To emulate the real time scenarios and make the model perform better on real time data, we evaluate our method on large amount of balanced data-set from Celeb-DF. We also show how our system can achieve competitive result using very simple and robust approach.

Keywords : ResNext Convolutional Neural Network, Recurrent Neural Network (RNN), Long Short Term Memory (LSTM)

திட்டப்பணிச்சுருக்கம்

யதார்த்தமான புகைப்படங்கள் மற்றும் வீடியோக்களை உருவாக்கக்கூடிய ஜெனரேட்டிவ் அட்வர்ஸரியல் நெட்வொர்க் ஐப் பயன்படுத்தி டீப்ஃபேக் வீடியோக்களை உருவாக்குவதற்கான சமீபத்திய முன்னேற்றங்களுடன், டிஜிட்டல் படங்களின் நம்பகத்தன்மையை அடையாளம் காண்பது மிகவும் சவாலானதாகி வருகிறது. இந்த ஆராய்ச்சியானது ஒரு ஆழமான கற்றல் மாதிரியை உருவாக்குவதற்கான அணுகுமுறையாகும், இது டீப்ஃபேக் மற்றும் உண்மையான வீடியோவை திறமையாக வேறுபடுத்துகிறது. டீப்ஃபேக்குகளை உருவாக்குதல் மற்றும் கண்டறிதல் ஆகிய இரு பகுதிகளிலும் ஆழமான கற்றல் தொடர்ந்து நிறைய வளர்ச்சியடைந்து வருகிறது. பழைய தரவுத்தொகுப்புடன் வடிவமைக்கப்பட்ட டீப்ஃபேக்கைக் கண்டறிவதற்காக உருவாக்கப்பட்ட மாதிரியானது காலாவதியாகலாம், மேலும் புதிய கண்டறிதல் நுட்பத்தின் தேவை எப்போதும் இருக்கும்.

இந்த அமைப்பு, ஃபிரேம்-லெவல் அம்சங்கள் மற்றும் இந்த அம்சங்களைப் பிரித்தெடுக்க ரெஸ்-நெக்ஸ்ட் கன்வல்யூஷன் நியூரல் நெட்வொர்க்கைப் பயன்படுத்துகிறது, மேலும் வீடியோவை வகைப்படுத்த நீண்ட குறுகிய கால நினைவகம் (எல்எஸ்டிஎம்) அடிப்படையிலான ரீகரண்ட் நியூரல் நெட்வொர்க் (ஆர்என்என்) பயிற்சிக்கு மேலும் பயன்படுத்தப்படுகிறது. நிகழ்நேர காட்சிகளைப் பின்பற்றுவதற்கும், நிகழ்நேரத் தரவில் மாதிரி சிறப்பாகச் செயல்படுவதற்கும், பெரிய அளவிலான சமச்சீர் தரவு-தொகுப்பில் இந்த முறையை மதிப்பீடு செய்கிறோம். மிகவும் எளிமையான மற்றும் வலுவான அணுகுமுறையைப் பயன்படுத்தி எங்கள் கணினி எவ்வாறு போட்டித்தன்மையை அடைய முடியும் என்பதையும் நாங்கள் காட்டுகிறோம்.

முக்கிய வார்த்தைகள் : ரெஸ்-நெக்ஸ்ட் கன்வல்யூஷன் நியூரல் நெட்வொர்க், தொடர்ச்சியான நரம்பியல் வலைப்பின்னல், நீண்ட குறுகிய கால நினைவகம்

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
CSV	Comma Separated Value
ResNeXt-50	Residual Network

CHAPTER 1

INTRODUCTION

1.1 DEEPLEARNING TECHNIQUES

Deep learning is the subset of machine learning composed of algorithms that permits the machine to train itself and perform a task. These algorithms uses multiple layers to progressively extract the high level feature from the raw input(images, audio, video). In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. Most modern deep learning models are based on convolutional neural networks (CNNs).

A Convolutional Neural Network (CNN) is an algorithm which can take in an input image, assign importance (learnable weights and biases) tovarious aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a CNN is much lower as compared to other classification algorithms. It is able to successfully capture the Temporal dependencies in an image through the application of relevant filters. The role of the CNN is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction.

1.2 MOTIVATION AND OBJECTIVES

DeepFakes involves videos, often obscene, in which a face can be swapped with someone else's using neural networks. DeepFakes are a general public concern. As soon as fake videos go viral, people believe them initially and keep sharing them with others. This makes the targeted person embarrassed. Thus it's important to develop methods to detect them.

The creation of a synthesized video needs information to be examined in order to reveal the dishonest, particularly facial expression variables. It is defined as an in-depth video study to find minor imperfections such as boundary points, background incoherence, double eyebrows or irregular twitch of the eye. The impetus behind this research is to recognize these distorted media, which is technically demanding and which is rapidly evolving. Many engineering companies have come together to unite a great deal of dataset. Competitions and actively include data sets to counter deepfakes.

Deepfake videos are now so popular that multiple political parties utilize this tool to produce faked images of the leader of their opposing party to propagate hate against them. Fake political videos telling or doing things that have never happened is a threat to election campaigns. These images are the primary source of false media controversies and propagate misleading news. In order to expose the forgery in extremely detailed facial expression, such details to be investigated frame by frame in the production of a deepfake picture. The goal of this research is to create a deep learning model that is capable of recognizing deepfake images. The model will learn what features differentiate a real image from a deepfake.

1.3 SCOPE OF THE PROJECT

The scope of the project is to avoid false media controversies, propagate misleading news and fake political videos telling things that have never happened is a threat to election campaigns.

1.4 ORGANIZATION OF THE REPORT

The thesis is organized into 6 chapters, describing each part of the project with detailed illustration and system design diagrams. The chapter are as follows:

Chapter 1 : This module consists of Introduction, Problem statement, Motivation and Objectives etc.

Chapter 2 : This module consists of Literature survey details of the project alongside their detailed methodologies, advantages, disadvantages etc.

Chapter 3 : This module consist System design of the project with its preliminary design such as overall Architecture diagram and process flow diagram which tells about the modules integration in the project.

Chapter 4 : This module consists of Detailed system design or module description with their input and algorithmic steps involved in each module to derive the output as per the user requirement.

Chapter 5 : This module consists the details about the hardware and software requirement to the project and the experiments that has been performed along with their outcomes. The detailed result of the project is also portrayed in this chapter.

Chapter 6 : This module conclude the project report with all the results and implementation procedure that has been underwent during the project development. The future works and excellence of implemented project is detailed.

The above mentioned six modules are followed up with the Reference which deliberately explains and list all the reference documents used during the various phases of the project, which includes the journal papers, conference papers, white papers, articles and websites referred for tutorials.

CHAPTER 2

LITERATURE REVIEW

This Chapter explains about the literature survey made on the existing system, analyzing the problem statements and issues with the existing system and proposed objectives for the new system.

2.1 EYE BLINKING DETECTION

Exposing AI Created Fake Videos by Detecting Eye Blinking describes a new method to expose fake face videos generated with deep neural network models. The method is based on detection of eye blinking[1] in the videos, which is a physiological signal that is not well presented in the synthesized fake videos. The method is evaluated over benchmarks of eye-blinking detection datasets and shows Deepfake Video Detection using Neural Networks.

Drutarovsky et al. [2] analyzed the variance of the vertical motions of eye region which is detected by a Viola-Jones type algorithm. Then a flock of KLT trackers are used on the eye region. Each eye region is divided into 3x3 cells and an average motion in each cell is calculated. This system proposed a scalar quantity that measures the aspect ratio of the rectangular bounding box of an eye corresponding to the eye openness degree in each frame. They then trained an SVM of EARs within a short time window to classify final eye state. Kim et al. [3] studied CNN-based classifiers to detect eye open and close state.

2.2 FACE BASED VIDEO MANIPULATION

Multiple approaches that target face manipulations in video sequences have been proposed since the 1990s [4]. This demonstrated the first real-time expression transfer for faces and later proposed Face2Face, a real-time facial reenactment system, capable of altering facial movements in different types of video streams.

Several face image synthesis techniques using deep learning have also been explored as surveyed by Generative adversarial networks (GANs) are used for aging alterations to faces [5], or to alter face attributes such as skin color. Deep feature interpolation shows remarkable results in altering face attributes such as age, facial hair or mouth expressions. Most of these deep learning based image synthesis techniques suffer from low image resolution. Karras et al. [6] show high quality synthesis of faces, improving the image quality using Generative Adversarial Network(GAN).

2.3 DETECTION USING MESONET

This section aims to discuss and analyze various techniques that have been used for deepfake detection. Various attempts have been made to detect deepfakes which use deep learning at their core. These approaches work either on detecting faults in video or in separate frames of the video. Approaches which involve image analysis target various parameters like face warping artifacts, eye blinking [2]. In 2018, “MesoNet” [7] was developed which used Inception model to detect faults at mesoscopic level. Convolutional Neural Networks (CNN) have shown excellent feature extraction properties that can be used by a model to detect deepfake videos.

Various other approaches have used CNN along with other learning models like Recurrent Neural Network (RNN), Long Short-Term Memory Networks (LSTM) and Capsule Network[8] to further improve the accuracy by detecting temporal discrepancies and have shown good results on dataset containing videos generated by FaceSwap and Deepfacelab.

2.4 PROPOSED SYSTEM

This approach for detecting the DF will be great contribution in avoiding the problems of the DF over the world wide web. It will be a web-based platform for the user to upload the video and classify it as fake or real. This project can be scaled up from developing a web-based platform to a browser plugin for automatic DF detections. Even big application like WhatsApp, Facebook can integrate this project with their application for easy pre detection of DF before sending to another user. The important objective is to evaluate its performance and acceptability in terms of security, user-friendliness, accuracy and reliability.

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

The proposed work of the system architecture is shown in Fig 3.1. The proposed system works on DeepFake detection. Celebrity dataset are labeled to the defined classes. Then they are loaded to the pre-defined model for training the dataset. In deepfake detection, ResNext50 is chosen to trained the dataset. Also use LSTM to enrich the accuracy gives the checkpoint file. At last the user video tested with the checkpoint file to detect fake or real.

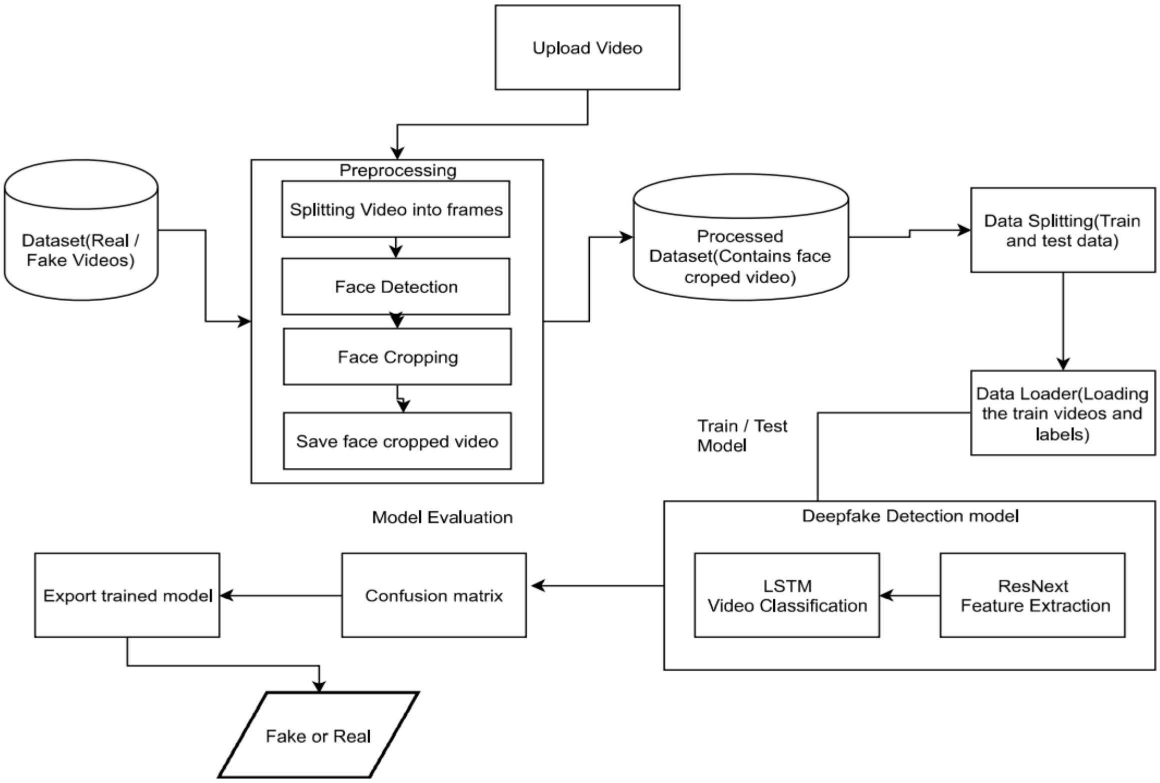

Upload Video
Preprocessing
Splitting Video into frames
Face Detection
Face Cropping
Save face cropped video
Dataset(Real / Fake Videos)
Processed Dataset(Contains face croped video)
Data Splitting(Train and test data)
Data Loader(Loading the train videos and labels)
Train / Test Model
Deepfake Detection model
LSTM Video Classification
ResNext Feature Extraction
Model Evaluation
Confusion matrix
Export trained model
Fake or Real

Fig 3.1 SYSTEM ARCHITECTURE

3.1.1 TRAINING DATASET

For DeepFake detection, Celebrity video datasets are collected from theGitHub. This contains 50% real and 50% fake videos.

3.1.2 DATASET PREPROCESSING

Dataset preprocessing includes the splitting the video into frames. Followed by the face detection and cropping the frame with detected face.

3.1.3 EXTRACT FEATURE

Instead of writing the code from scratch, we used the pre-trained model of ResNext for feature extraction. ResNext is Residual CNN network optimized for high performance on deeper neural networks. For the experimental purpose we have used resnext50_32x4d model. We have used a ResNext of 50 layers and 32 x 4 dimensions.

Following, we will be fine-tuning the network by adding extra required layers and selecting a proper learning rate to properly converge the gradient descent of the model. The 2048-dimensional feature vectors after the last pooling layers of ResNext is used as the sequential LSTM input.

3.1.4 LSTM FOR SEQUENCE PROCESSING

2048-dimensional feature vectors is fitted as the input to the LSTM. We are using 1 LSTM layer with 2048 latent dimensions and 2048 hidden layers along with 0.4 chance of dropout, which is capable to do achieve our objective. LSTM is used to process the frames in a sequential manner so that the temporal analysis of the video can be made, by comparing the frame at ‘t’ second with the frame of ‘t-n’ seconds. Where n can be any number of frames before t.

The model also consists of Leaky Relu activation function. A linear layer of 2048 input features and 2 output features are used to make the model capable of learning the average rate of correlation between eh input and output. An adaptive average polling layer with the output parameter 1 is used in the model. Which gives the the target output size of the image of the form H x W. For sequential processing of the frames a Sequential Layer is used. The batch size of 4 is used to perform the batch training. A SoftMax layer is used to get the confidence of the model during prediction.

3.1.5 HYPER-PARAMETER TUNING

It is the process of choosing the perfect hyper-parameters for achieving the maximum accuracy. After reiterating many times on the model. The best hyper-parameters for our dataset are chosen. To enable the adaptive learning rate Adam optimizer with the model parameters is used. The learning rate is tuned to 1e-5 (0.00001) to achieve a better global minimum of gradient descent. The weight decay used is 1e-3.

As this is a classification problem so to calculate the loss cross entropy approach is used. To use the available computation power properly the batch training is used. The batch size is taken of 4. Batch size of 4 is tested to be ideal size for training in our development environment.

The User Interface for the application is developed using Django framework. Django is used to enable the scalability of the application in the future.

The first page of the User interface i.e index.html contains a tab to browse and upload the video. The uploaded video is then passed to the model and prediction is made by the model. The model returns the output whether the video is real or fake along with the confidence of the model. The output is rendered in the predict.html on the face of the playing video.

3.2 FLOWCHART DESIGN

This is the Flowchart design of the proposed system. The Flowchart design of the prediction workflow is shown in Fig.3.2.

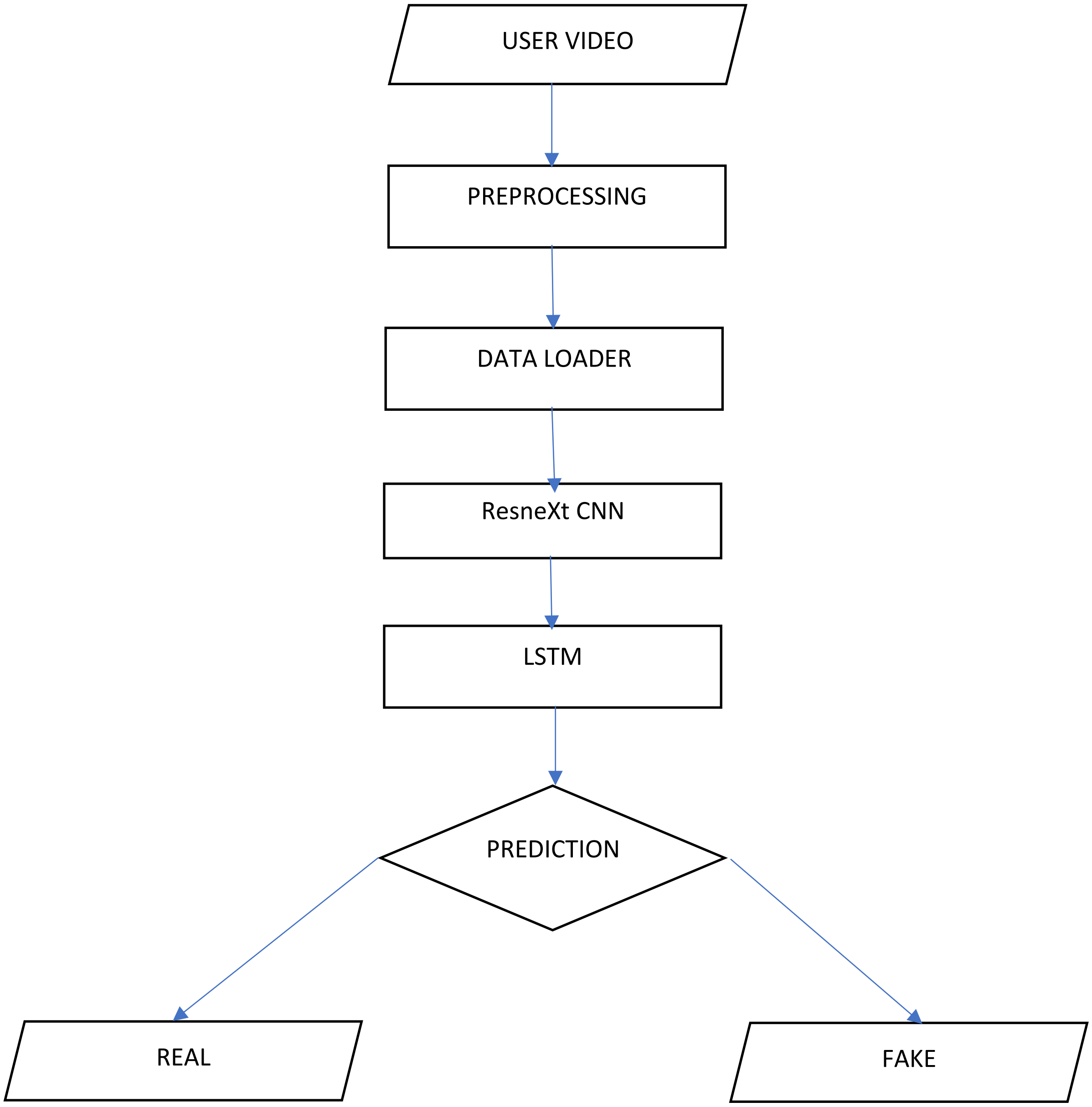


Fig:3.2 PREDICTION WORKFLOW

CHAPTER 4 – ALGORITHM IMPLEMENTATION

4.1 DATASET GATHERING

For making the model efficient for real time prediction. We have gathered the data from Celebrity DF. To avoid the training bias of the model we have considered 50% Real and 50% fake videos.

After Preprocessing of the Celebrity DF dataset, we have taken 800 Real and 800 Fake videos.

4.2 PRE-PROCESSING

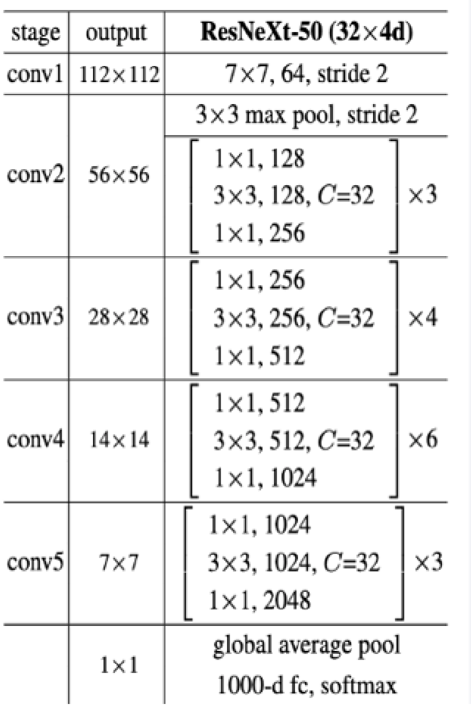
Dataset preprocessing includes the splitting the video into frames. Followed by the face detection and cropping the frame with detected face.

1. Using glob we can import all the videos from the directory
2. Cv2.VideoCapture is used to read the videos and get the mean number of frames in each video.
3. To maintain uniformity, based on mean a value 100 is selected as idea value for creating the new dataset.
4. The video is split into frames and the frames are cropped on face location.
5. The face cropped frames are again written to new video using VideoWriter.
6. The new video is written at 30 frames per second and with the resolution of 112 * 112 pixels in the mp4 format.
7. Instead of selecting the random videos, to make the proper use of LSTM for temporal sequence analysis the first 150 frames are written are written to new video.

4.3 MODEL DETAILS

4.3.1 ResNext CNN:

The pre-trained model of Residual Convolution Neural Network is used. The model name is resnext50_32x4d()[22]. This model consists of 50 layers and 32 x 4 dimensions. Figure 4.1 shows the detailed implementation of model.



stage	output	**ResNeXt-50 (32×4d)**
conv1	112×112	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2
		$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128, C{=}32 \\ 1\times1, 256 \end{bmatrix} \times3$
conv3	28×28	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256, C{=}32 \\ 1\times1, 512 \end{bmatrix} \times4$
conv4	14×14	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512, C{=}32 \\ 1\times1, 1024 \end{bmatrix} \times6$
conv5	7×7	$\begin{bmatrix} 1\times1, 1024 \\ 3\times3, 1024, C{=}32 \\ 1\times1, 2048 \end{bmatrix} \times3$
	1×1	global average pool 1000-d fc, softmax

Figure 4.1 : ResNext Architecture

4.3.2 SEQUENTIAL LAYER:

Sequential is a container of Modules that can be stacked together and run at the same time. Sequential layer is used to store feature vector returned by the ResNext model in a ordered way. So that it can be passed to the LSTM sequentially.

4.3.3. LSTM LAYER:

LSTM is used for sequence processing and spot the temporal change between the frames.2048-dimensional feature vectors is fitted as the input to the LSTM. We are using 1 LSTM layer with 2048 latent dimensions and 2048 hidden layers along with 0.4 chance of dropout, which is capable to do achieve our objective. LSTM is used to process the frames in a sequential manner so that the temporal analysis of the video can be made, by comparing the frame at 't' second with the frame of 't-n' seconds. Where n can be any number of frames before t. Figure 4.2 shows the Overview of LSTM Architecture.

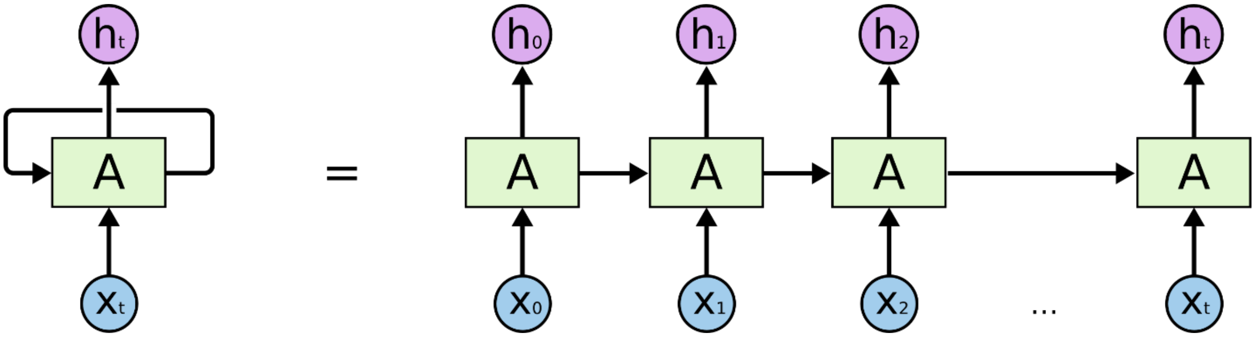


Figure 4.2 : Overview of LSTM Architecture

4.3.4 RELU:

A Rectified Linear Unit is activation function that has output 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input. The operation of ReLU is closer to the way our biological neurons work. ReLU is non-linear and has the advantage of not having any backpropagation errors unlike the sigmoid function, also for larger Neural Networks, the speed of building models based off on ReLU is very fast. Figure 4.3 shows the Relu activation function.

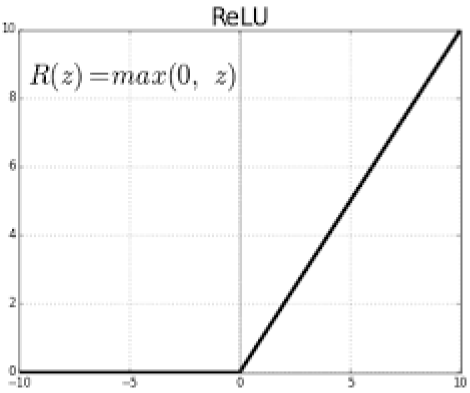


Figure 4.3 : ReLu Activation Function

4.3.5 DROPOUT LAYER :

Dropout layer with the value of 0.4 is used to avoid overfitting in the model and it can help a model generalize by randomly setting the output for a given neuron to 0. In setting the output to 0, the cost function becomes more sensitive to neighbouring neurons changing the way the weights will be updated during the process of back propagation.

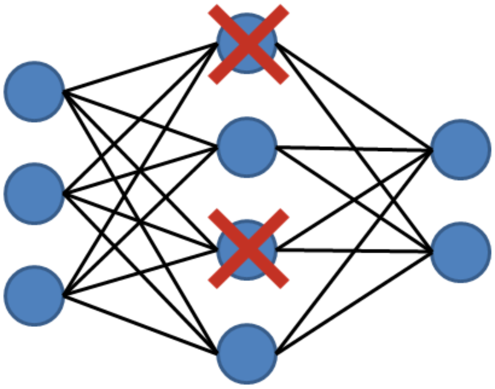


Figure 4.4 : Dropout layer

4.3.6 ADAPTIVE AVERAGE POOLING LAYER:

It is used to reduce variance, reduce computation complexity and extract low level features from neighbourhood. 2 dimensional Adaptive Average Pooling Layer is used in the model.

4.4 MODEL TRAINING

4.4.1 TRAIN TEST SPLIT

The dataset is split into train and test dataset with a ratio of 70% train videos (1,300) and 30% (300) test videos. The train and test split is a balanced split i.e 50% of the real and 50% of fake videos in each split.

4.4.2 DATA LOADER

A Dataset object loads training or test data into memory, and a Data Loader object fetches data from a Dataset and serves the data up in batches. It is used to load the videos and their labels with a batch size of 4.

4.4.3 TRAINING:

The training is done for 20 epochs with a learning rate of 1e-5 (0.00001),weight decay of 1e-3 (0.001) using the Adam optimizer.

4.4.4 ADAM OPTIMIZER:

Adam is an optimization solver for the Neural Network algorithm that is computationally efficient, requires little memory, and is well suited for problems that are large in terms of data or parameters or both. Adam is a popular extension to stochastic gradient descent. To enable the adaptive learning rate Adam optimizer with the model parameters is used.

4.4.5 CROSS ENTROPY:

To calculate the loss function Cross Entropy approach is used because we are training a classification problem.

4.4.6 SOFTMAX LAYER:

A Softmax function is a type of squashing function. Squashing functions limit the output of the function into the range 0 to 1. This allows the output to be interpreted directly as a probability. Similarly, softmax functions are multi-class sigmoids, meaning they are used in determining probability of multiple classes at once. Since the outputs of a softmax function can be interpreted as a probability (i.e.they must sum to 1), a softmax layer is typically the final layer used in neural network functions. It is important to note that a softmax layer must have the same number of nodes as the output later. In our case softmax layer has two output nodes i.e REAL or FAKE, also Softmax layer provide us the confidence(probability) of prediction.

4.4.7 CONFUSION MATRIX:

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made. Confusion matrix is used to evaluate our model and calculate the accuracy.

4.4.8 EXPORT MODEL:

After the model is trained, we have exported the model. So that it can be used for prediction on real time data.

4.5 MODEL PREDICTION

The model is loaded in the application. The new video for prediction is preprocessed and passed to the loaded model for prediction. The trained model performs the prediction and return if the video is a real or fake along with the confidence of the prediction.

CHAPTER 5 - EXPERIMENTAL RESULTS

In experimental results consists of the details about the hardware and software requirements to the project and the experiments that have been performed along with their outcomes. The detailed result of the project is also portrayed in this chapter.

5.1 HARDWARE REQUIREMENTS

In this project, a computer with sufficient processing power is needed. This project requires too much processing power, due to the image and video batch processing. So we need some minimum requirement of the system.

- Operating System - Windows
- RAM - Minimum 8 GB
- Hard Disk - Minimum 100 GB
- Graphic card - NVIDIA GeForce GTM Titan

5.2 SOFTWARE REQUIREMENTS

1. Operating System – Windows 8+

2. Programming Language – Python 3.10.4

3. PyTorch 1.11.0

3. Framework – Django 3.0

4. Libraries – OpenCV, Face Recognition

5.3 DATASET PREPROCESSING

Dataset preprocessing includes the splitting the video into frames. Followed by the face detection and cropping the frame with detected face. To maintain the uniformity in the number of frames the mean of the dataset video is calculated and the new processed face cropped dataset is created containing the frames equal to the mean. The frames that doesn’t have faces in it are ignored during preprocessing. The preprocessing workflow is shown in Figure.5.1.

To maintain the uniformity of number of frames, we have selected a threshold value based on the mean of total frames count of each video. Another reason for selecting a threshold value is limited computation power.

As processing the 10 second video at 30 frames per second (i.e) total 300 frames will require a lot of computational power. So for experimental purpose we are proposing to used only first 100 frames for training the model. The newly created video is saved at frame rate of 30 fps and resolution of 112 * 112.

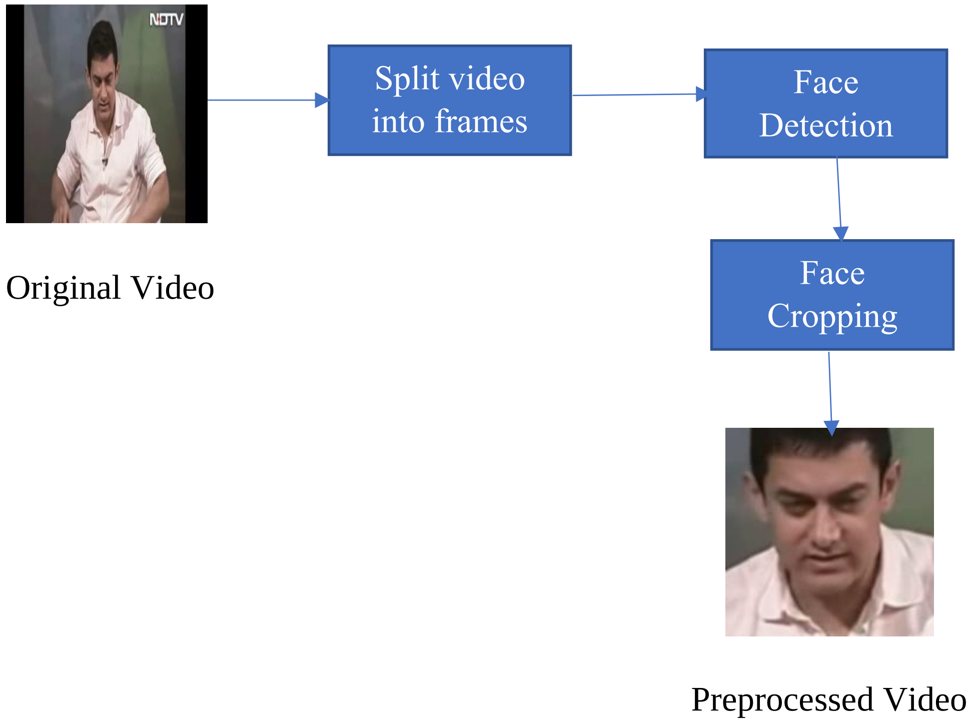


Fig 5.1 Preprocessing

5.4 TRAIN MODEL FOR DETECTION

The model will train using ResNext CNN and LSTM. Figure 5.2 shows the complete workflow of training process. ResNext is Residual CNN network optimized for high performance on deeper neural networks. For the experimental purpose we have used resnext50_32*4d model.

Following, we will be fine-tuning the network by adding extra required layers and selecting a proper learning rate to properly converge the gradient descent of the model. The 2048-dimentiaonal feature vectors after the last pooling layers of ResNext is used as the sequential LSTM input. A linear layer of 2048 input features and 2 output features are used to make the model capable of learning the average rate of correlation between eh input and output. An adaptive average polling layer with the output parameter 1 is used in the model. Which gives the the target output size of the image of the form H x W. For sequential processing of the frames a Sequential Layer is used. The batch size of 4 is used to perform the batch training. A SoftMax layer is used to get the confidence of the model during predication.

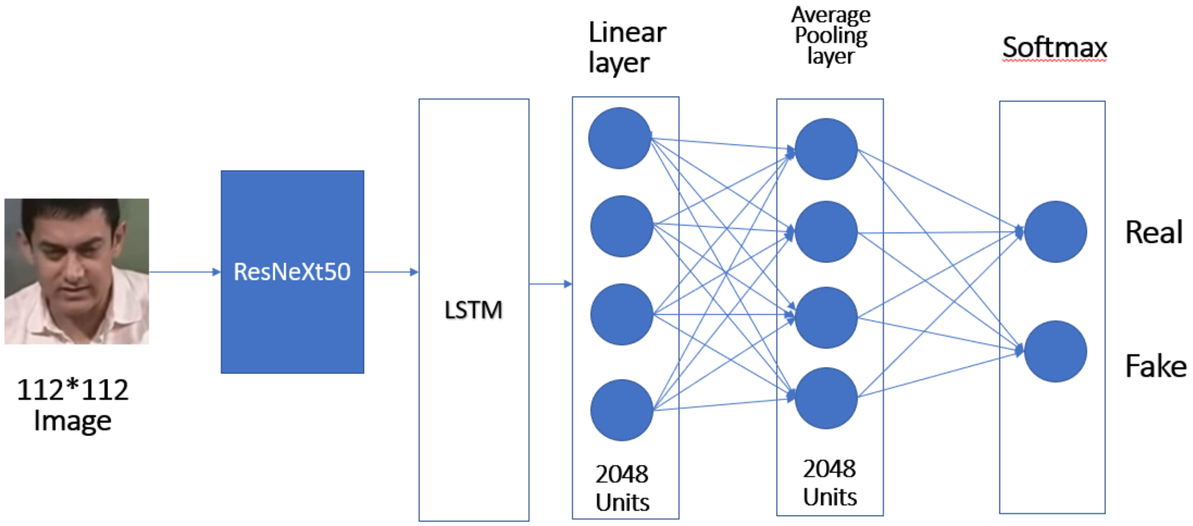


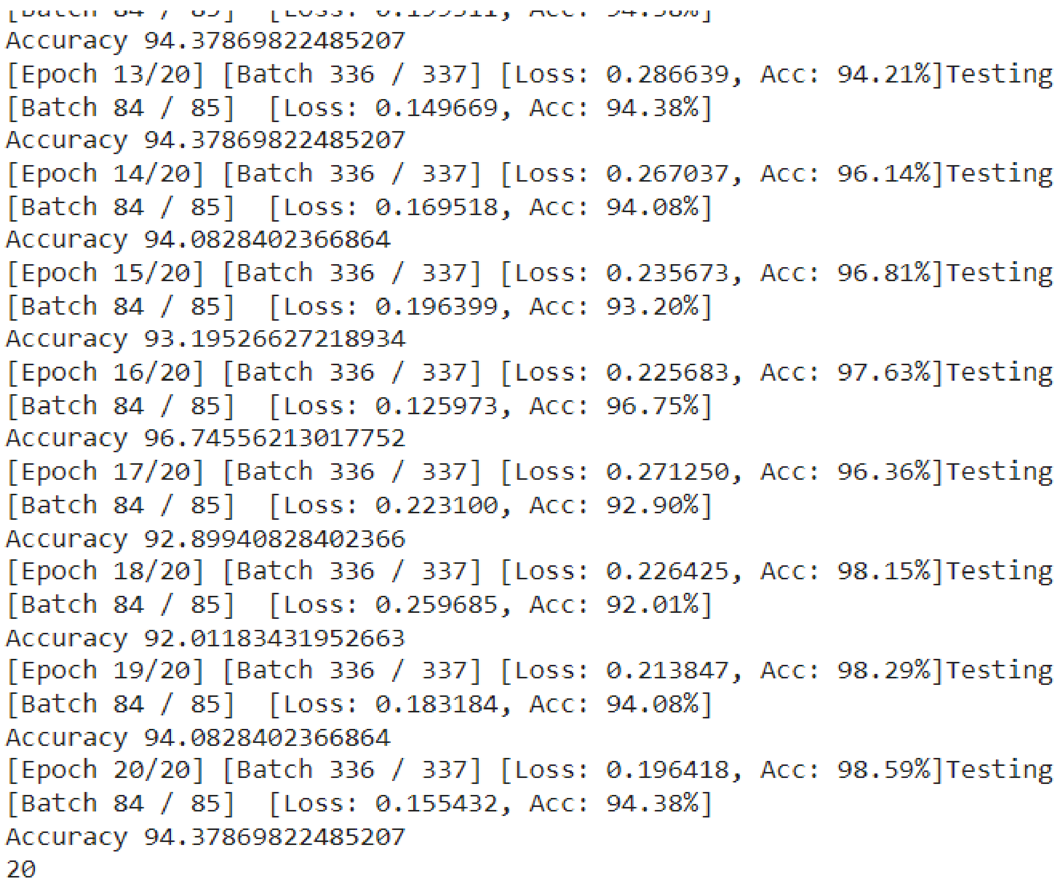
Figure 5.2: Complete workflow of training process

5.5 TRAINING AND TESTING ANALYSIS

The whole dataset contains 800 Real and 800 Fake Videos. The Dataset is divided as 80% for training and the remaining consider as the testing dataset. Training and Testing accuracy with 20 frames is 98% and 94%. Figure 5.3 shows the Accuracy and Loss of training and testing analysis.

Model Name	Total Frames	Training Time	Training Loss	Testing loss
ResNext 50	20	4 Hours	0.1964	0.1554

The above table brings it from the output of the resulting training Process. It consumes nearly 4 hours for the training process and results a checkpoint file.



```
[Batch 84 / 85]  [Loss: 0.159511, Acc: 94.38%]
Accuracy 94.37869822485207
[Epoch 13/20] [Batch 336 / 337] [Loss: 0.286639, Acc: 94.21%]Testing
[Batch 84 / 85]  [Loss: 0.149669, Acc: 94.38%]
Accuracy 94.37869822485207
[Epoch 14/20] [Batch 336 / 337] [Loss: 0.267037, Acc: 96.14%]Testing
[Batch 84 / 85]  [Loss: 0.169518, Acc: 94.08%]
Accuracy 94.0828402366864
[Epoch 15/20] [Batch 336 / 337] [Loss: 0.235673, Acc: 96.81%]Testing
[Batch 84 / 85]  [Loss: 0.196399, Acc: 93.20%]
Accuracy 93.19526627218934
[Epoch 16/20] [Batch 336 / 337] [Loss: 0.225683, Acc: 97.63%]Testing
[Batch 84 / 85]  [Loss: 0.125973, Acc: 96.75%]
Accuracy 96.74556213017752
[Epoch 17/20] [Batch 336 / 337] [Loss: 0.271250, Acc: 96.36%]Testing
[Batch 84 / 85]  [Loss: 0.223100, Acc: 92.90%]
Accuracy 92.89940828402366
[Epoch 18/20] [Batch 336 / 337] [Loss: 0.226425, Acc: 98.15%]Testing
[Batch 84 / 85]  [Loss: 0.259685, Acc: 92.01%]
Accuracy 92.01183431952663
[Epoch 19/20] [Batch 336 / 337] [Loss: 0.213847, Acc: 98.29%]Testing
[Batch 84 / 85]  [Loss: 0.183184, Acc: 94.08%]
Accuracy 94.0828402366864
[Epoch 20/20] [Batch 336 / 337] [Loss: 0.196418, Acc: 98.59%]Testing
[Batch 84 / 85]  [Loss: 0.155432, Acc: 94.38%]
Accuracy 94.37869822485207
20
```

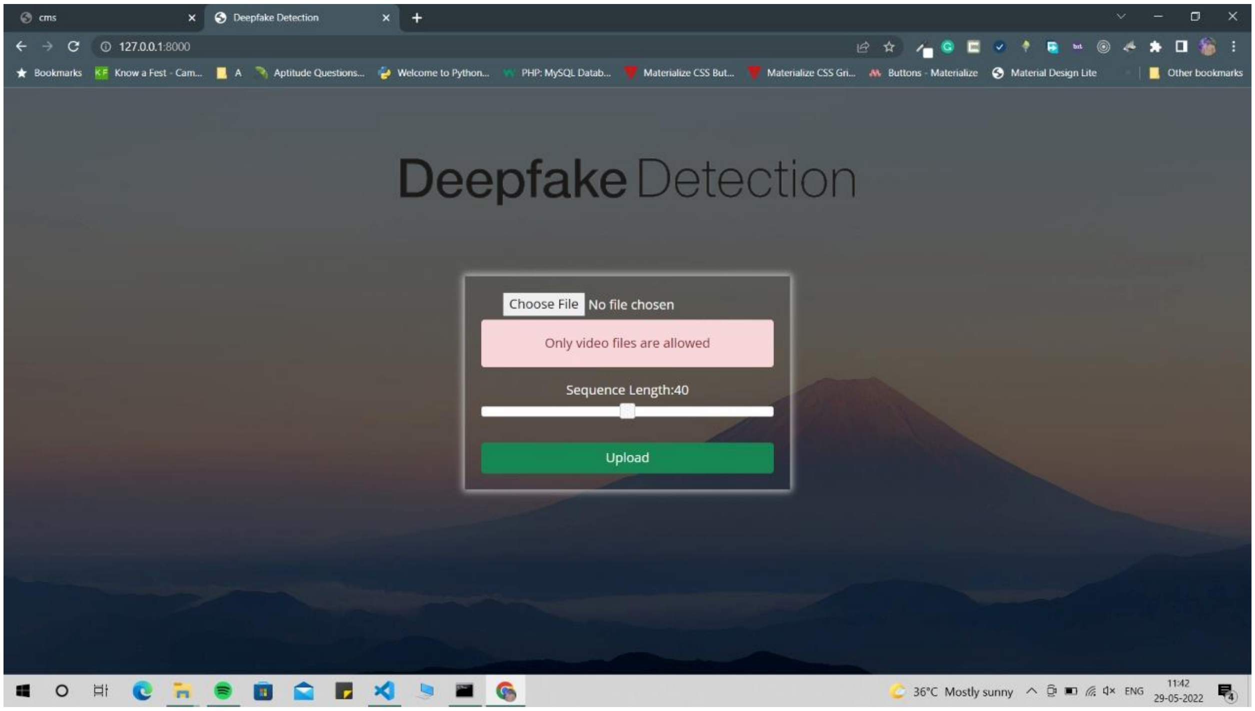
Figure 5.3 : Training and Testing Loss

5.6 TESTCASES:

To check the accuracy of the system we are using 6 sample test cases.

Following are the test cases:

1. Upload a PDF file instead of video



No file chosen

Only video files are allowed

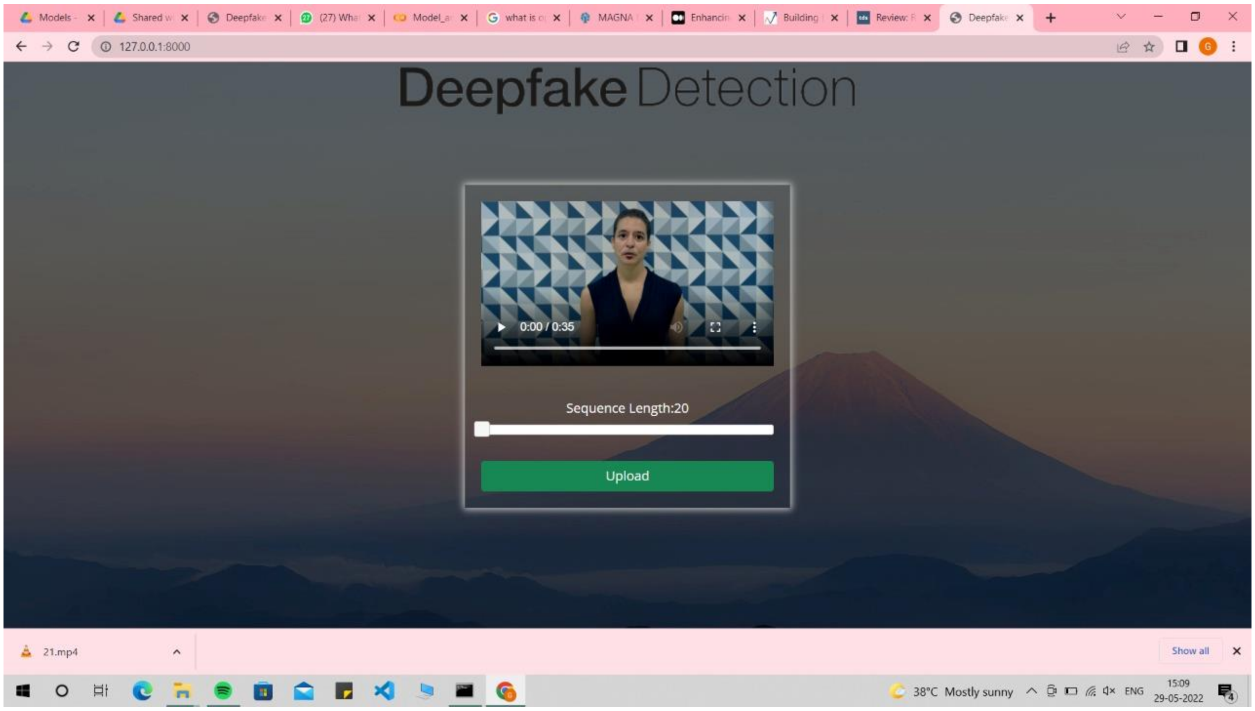
Sequence Length:40

Output : Only video files are allowed

Result : Passed

Figure 5.4 : Test Case 1

2. Upload more than 100 MB Video file

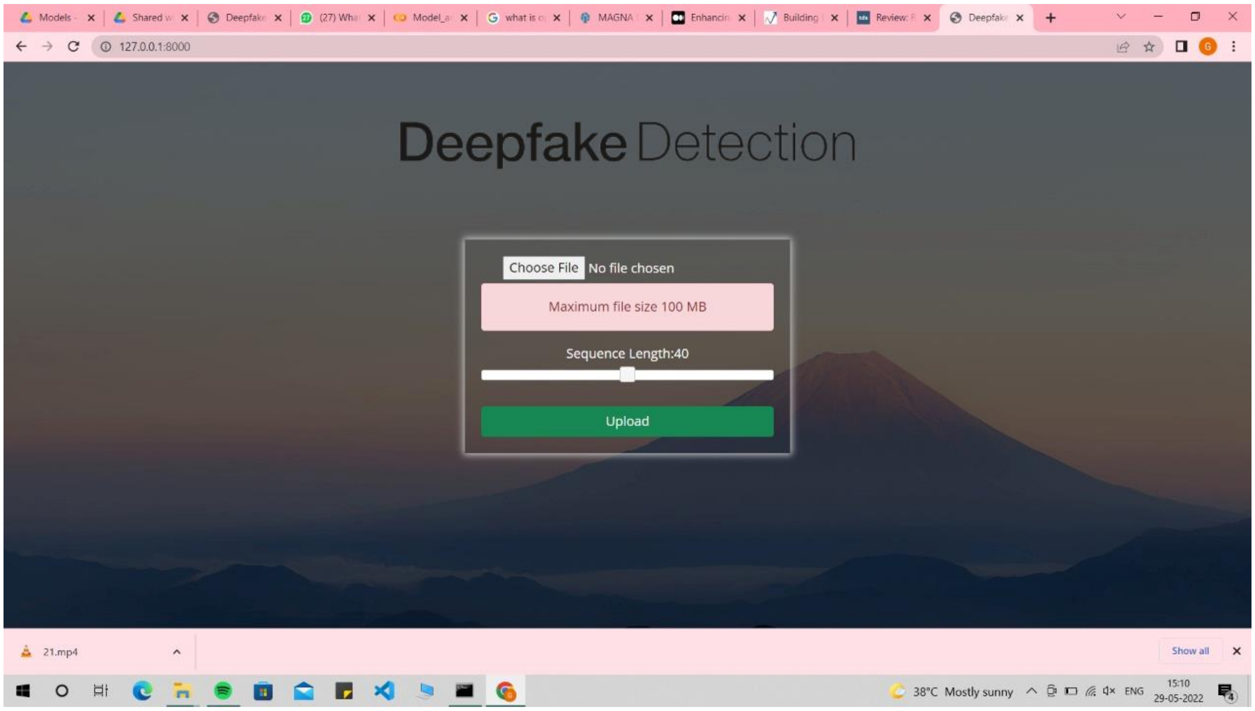


127.0.0.1:8000

Deepfake Detection

0:00 / 0:35

Sequence Length:20



127.0.0.1:8000

Deepfake Detection

Choose File No file chosen

Maximum file size 100 MB

Sequence Length:40

Figure 5.5 : Test Case 2

Output : Maximum file size is 100 MB.

Result : Passed

3. Press Upload button without selecting Video

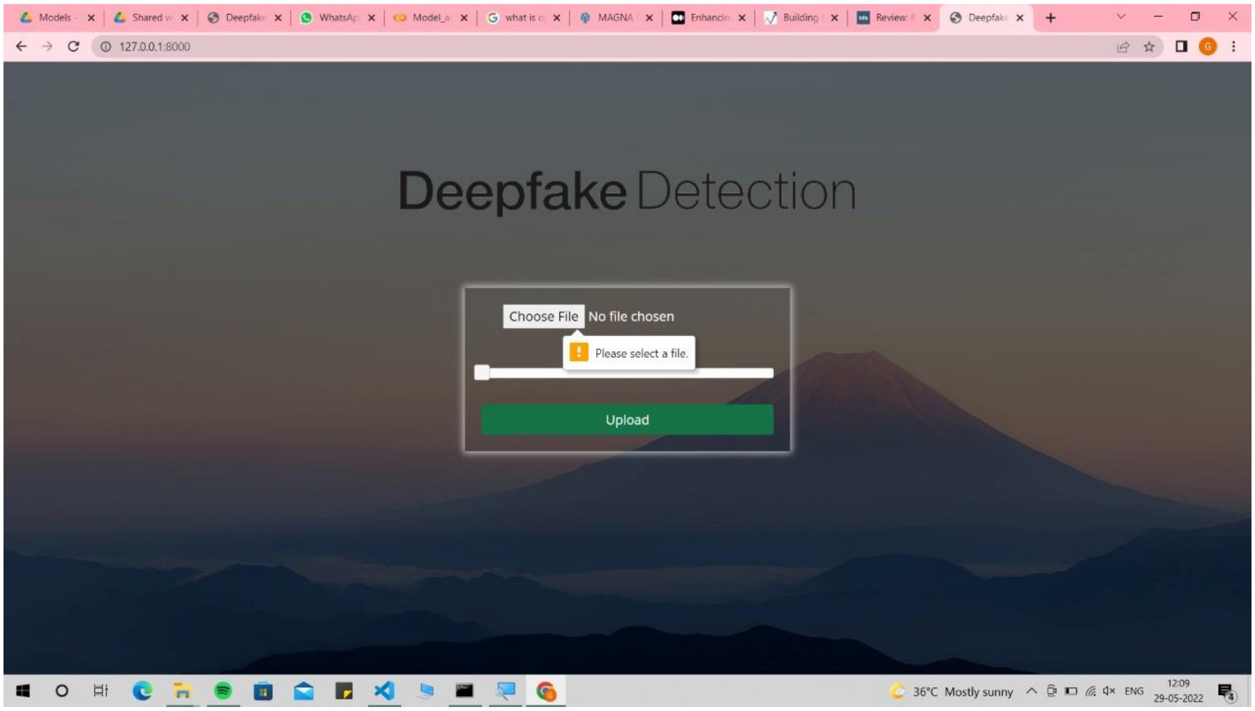
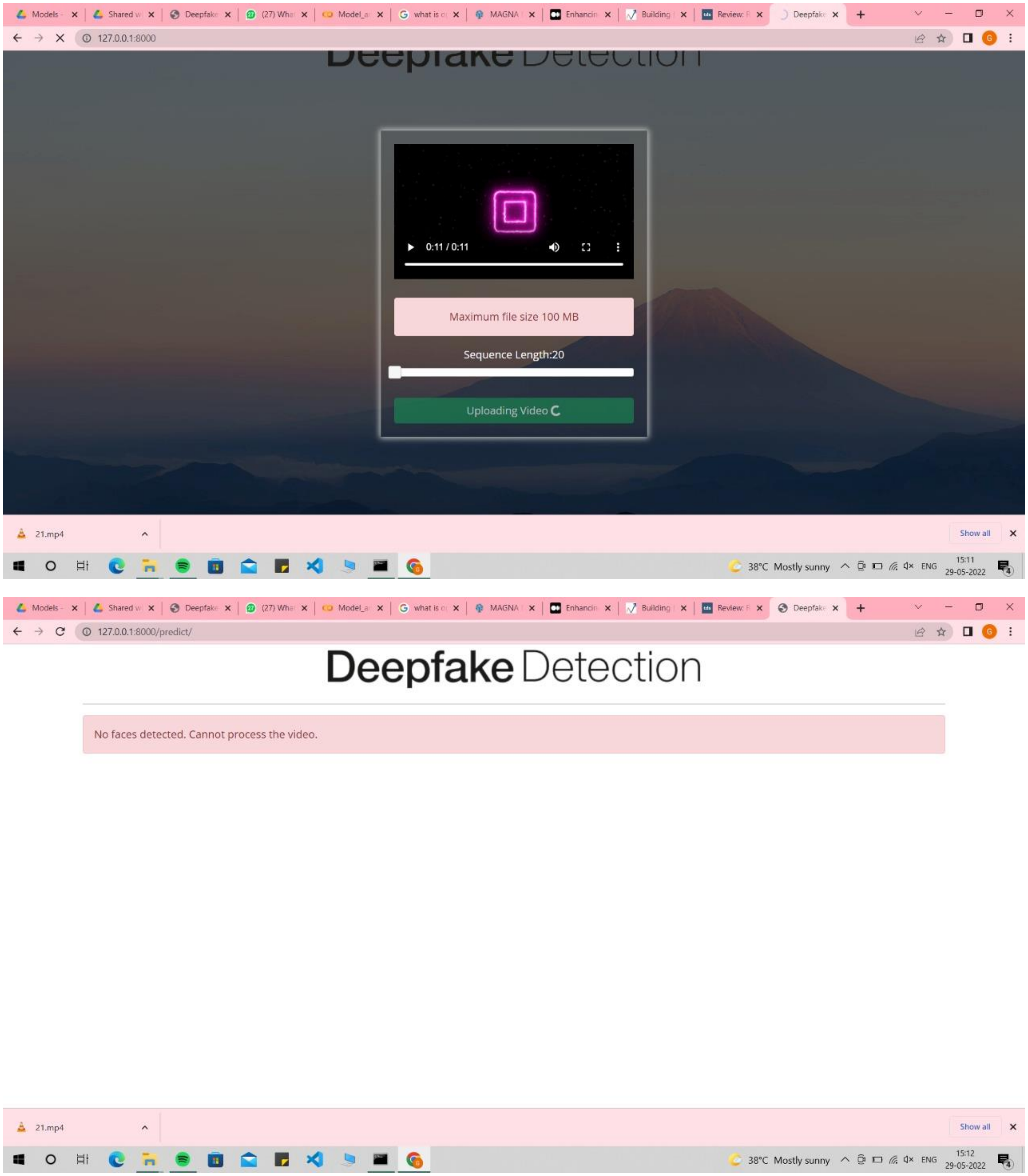


Figure 5.6 : Test Case 3

Output : Please Select a file

Result : Passed

4. Upload a file without any faces



No faces detected. Cannot process the video.

Figure 5.7: Test Case 4

Output : No faces detected. Cannot process the video

Result : Passed

5. Upload a Fake Video

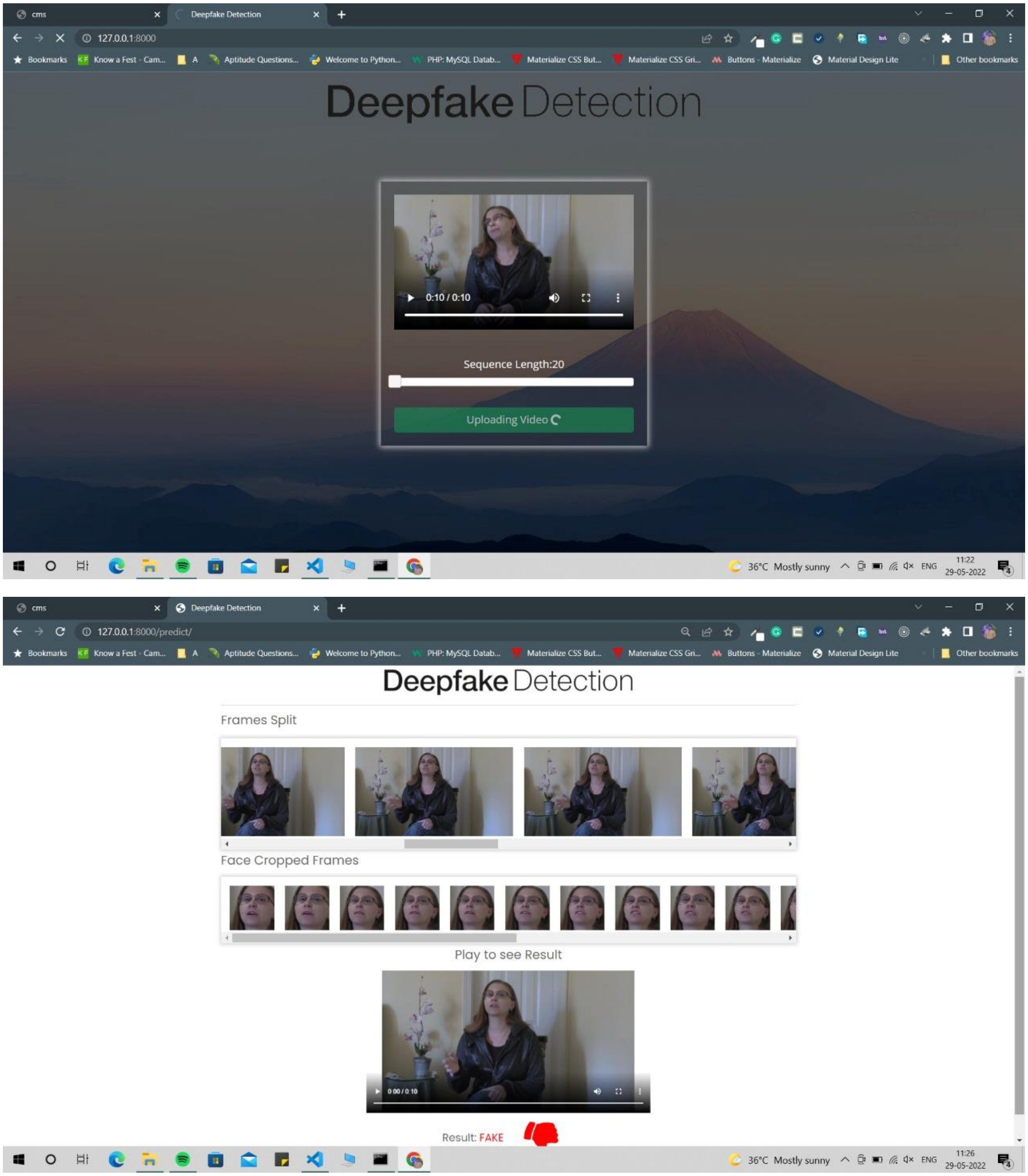
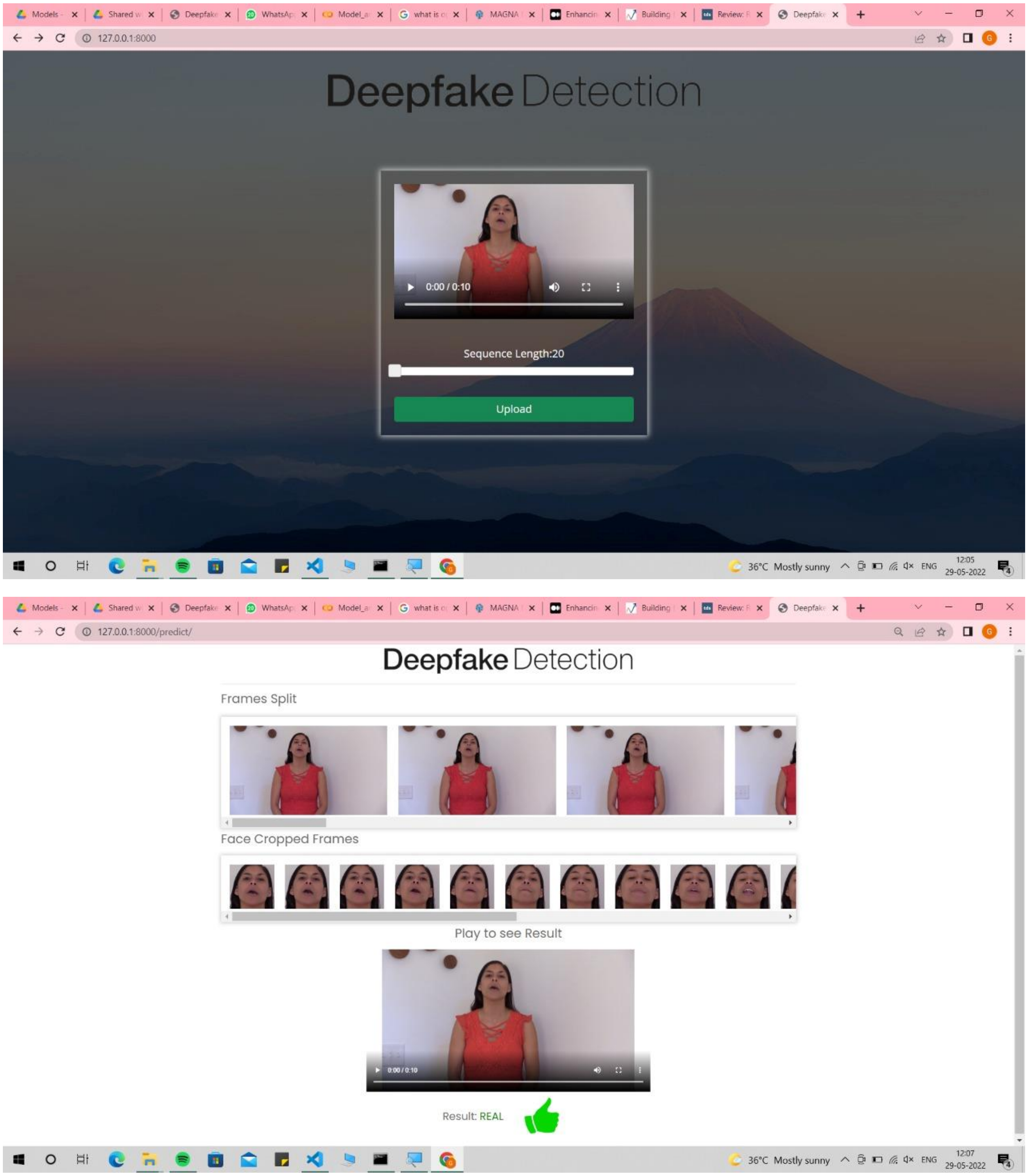


Figure 5.8 : Test Case 5

Output : Fake

Result : Passed

6. Upload a Real Video



Sequence Length:20

Result: REAL

Figure 5.9 Test Cases : 6

Output : Real

Result : Passed

REFERENCES

[1] Yuezun Li, Ming-Ching Chang and Siwei Lyu “Exposing AI Created Fake Videos by Detecting Eye Blinking” in arxiv.

[2] T. Drutarovsky and A. Fogelton. Eye blink detection using variance of motion vectors. In ECCV, pages 436–448, 2014.

[3] K. W. Kim, H. G. Hong, G. P. Nam, and K. R. Park. A study of deep cnn-based classification of open and closed eyes using a visible light camera sensor. Sensors, 17(7):1534, 2017

[4] David Guera and Edward J.Delp. Deepfake Video Detection Using Recurrent Neural Network. Video and Image processing Laboratory, Purdue University, In Ieeexplore, 2018

[5] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional generative adversarial networks. arXiv:1702.01983, Feb. 2017

[6] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. arXiv:1710.10196, Oct. 2017.

[7] DariusAfchar, Vincent Nozick, Junichi Yamagishi and Isao Echizen, “MesoNet: a Compact Facial Video Forgery Detection Network”, arXiv:1809.00888v1, 4 Sep 2018.

[8] Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen, “Use of a capsule network to detect fake images and videos”, arXiv:1910.12467v2 [cs.CV] 29 Oct 2019.