REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technology

DETECTING DEEP FAKES INSIDE SOCIAL MEDIA

Osman Tarig Osman AHMED

Master's Thesis

Supervisor

Asst. Prof. Dr. Hakan KOYUNCU

Istanbul, 2022

DETECTING DEEP FAKES INSIDE SOCIAL MEDIA

Osman Tarig Osman AHMED

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2022

The thesis titled DETECTING DEEP FAKES INSIDE SOCIAL MEDIA prepared by OSMAN TARIG OSMAN AHMED and submitted on 06 /12 /2022 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

Asst. Prof. Dr. Hakan KOYUNCU
Supervisor

Thesis Defense Jury Members:

Asst. Prof. Dr. Hakan KOYUNCU	Department of Computer Engineering, Altınbaş University	_____
Asst. Prof. Dr. Abdulahi Abdu IBRAHIM	Department of Computer Engineering, Altınbaş University	_____
Asst. Prof. Dr. Hakan AYDIN	Department of Computer Engineering, İstanbul Topkapı	_____

I hereby declare that this thesis meets all format and submission requirements of a Master`s thesis.

Submission date of the thesis to the Institute Graduate Studies: ____/____/____

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Osman Tarig Osman/AHMED

Signature

DEDICATION

Thanks to everyone who has helped me reach this stage of my studies, especially my dedicated father, my wonderful mother, my kind grandfather, my dear uncle, and my entire family.

ACKNOWLEDGEMENTS

I gratefully acknowledge Asst.Prof. Hakan Koyuncu from Alltinbas University in Istanbul, Turkey for his professional care about this work.

ABSTRACT

DETECTING DEEP FAKES INSIDE SOCIAL MEDIA

Ahmed, Osman Tarig Osman,

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Asst.Prof. Dr. Hakan Koyuncu

Date: December /2022

Pages: 72

Deep learning is the root of the phrase (deep fake). The deep fake is one field that has recently heavily used deep learning. Due to how simple it is for any user to make fake data using apps and data that are both readily available on the internet, then the deep fake poses a danger to privacy and security. Given this, a new technology has emerged recently to deal with deep fakes and reduce their severity known as (detecting deep fakes). In this paper, we will review a few of the deep fake detection algorithms that other researchers have developed, but the main approach used in this study is the combination of LSTM and resnetv2 technology with OpenCV. Resnetv2 works to extract features and transfer them to LSTM, which trains those features, and then OpenCV enlarges the face and produces an animated version of it that makes it easier for the naked eye to distinguish between the original and manipulated versions of data.

Keywords: Social Media, Deep Learning, Machine Learning, Deepfake Generation, Deepfake Detection, Data Sets, LSTM, ResNetV2, Open CV

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

AI	:	Artificial Intelligence
CASIA	:	Chinese Academy of Sciences' Institute of Automation
Celeb-DF	:	Celebrities Deep Fake
CNN	:	Convolutional Neural Network
CPU	:	Central Processing Unit
CPU	:	Graphics Processing Unit
CSV	:	Comma Separated Values
DFDC	:	Deep Fake Detection Challenge
DFFD	:	Diverse Fake Face Dataset
DF-TIMIT	:	Deep Fake -Texas Instruments Massachusetts Institute of Technology
Enc-DecS	:	Encoder-Decoder Source
Enc-DecT	:	Encoder-Decoder Target
FFHQ	:	Flickr Faces High Quality
GAN	:	Generative Adversarial Network
LSTM	:	Long Short-Term Memory
Open-CV	:	Open Computer Vision
RNN	:	recurrent neural network
VGG	:	Visual Geometry Group

1. INTRODUCTION

1.1 GENRAL VIEW

The deep learning subfield of artificial intelligence focuses on modeling human brain neurons. To offer machines the capacity to learn on their own. Machine learning has a subclass called deep learning. It is well-recognized that deep learning can assist in the generation and detection of deep fakes. The Deep fake first appeared in 2017 and soon became well-liked by Reddit users. Face reenactment and face swapping [1] are the foundations of this technology that let users change real data into phony data. The impact of this technology on our life can be either beneficial or detrimental, depending on the user's intention. Deep fakes can serve in education sides as history studies or in other different sectors, as an example, this technology was used in arts and cinema during the pandemic, especially in the series "Bleu Belle La Vie" to fake the face of one actor infected with covid19 viewers watched that series without the audience noticing the differences between actors as shown below in (Figure 1.1).

Harmful uses of the deep fake can include (using technology to threaten people by filming them doing awful things and then blackmailing them or disseminating false propaganda to sway public opinion). As an example, witnessed in the last US elections, a video of Joe Biden attempting to sway popular opinion. However, with the development of crime, it was necessary to develop methods of combating it, although crime is always proactive - thus appeared modern technology that is responsible for detecting fake data. The last few years have seen significant investment in research by numerous technology institutions looking into ways to reduce the risks posed by deep fakes. To demonstrate to the public the capabilities of this technology, Jordan Peele produced a fake Obama video (Figure 1.2).

This study will give information about deep faking and some of the strategies used to create it, as well as a general perspective on detecting deep fake methods using publicly available datasets. It will include information regarding past tactics, which include a combination of Long Short-Term Memory and resentv2 [2], [3], as well as a strategy that is based on open Computer vision approaches [4].

The proposed method will combine them to provide us with a new strategy for distinguishing authentic data from bogus data, which will leverage Open computer vision with the assistance of Long short-term memory and resentv2. The results will be separated into four sections and will offer us high precision in the end.

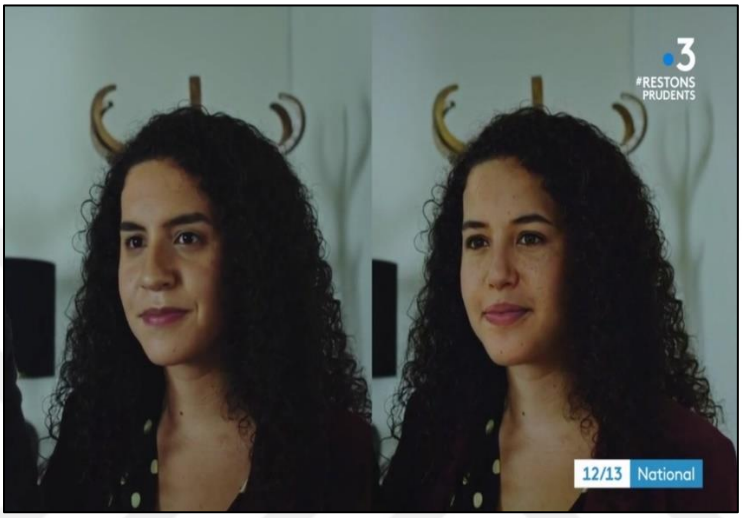

Figure 1.1: Deep fake in Bleu Belle La Vie [5]

Figure 1.2: Deep fake for President Obama [6]

1.2 RESEARCH BASICS

Collecting videos coming from social media like “Facebook “, “Twitter “, ""“YouTube “to check if they real or fake.

1.2.1 Definition of Main Concepts

The main concepts are improved software system to detect the deepfake videos in the social media to reduce the dangerous of deepfakes.

1.2.2 Tasks

There are a lot of software systems related with detecting deepfakes each one has their own characteristics, weakness, and strength point, in this research we try to avoid the weakness in previous works and improve the accuracy as we will show later in another part of the thesis.

1.2.3 Techniques

The majority of deep fake detection systems has its training, validation, accuracy, and language style. in this work we will try to deal with all these characteristics.

1.3 RESEARCH PROBLEMS

i. How do we make deep fakes?

ii. How can we improve deep fake detection?

1.4 RESEARCH OBJECTIVES

i. To generate deep fake, we must go through two main steps (encoder and decoder), with the encoder defining the fragments of root data. The decoder will receive the fragments extracted from the root data to generate deep fake data.

The amount of data available in recent times has made the process of producing deep fakes so simple with the help of simple applications available in any user's hand.

ii. To improve deep fake detection, we should focus on making the model robust and scalable so that it can handle any large amount of data. especially as deep fake generation improves rapidly.

1.5 LITERATURE REVIEW

1.5.1 Deep Fake Generation

Many approaches for creating deep fake (pictures, videos) have grown in recent years, and all these strategies are simple to apply by any type of user with accessible internet.

Generative adversarial networks (GANs) This technique has two components: an encoder, which converts data received from real data to fake data, and a decoder, which helps to classify data as original or fake [7].

Another technique is the (fake app) which looks like GAN because it also relies on an encoder and decoder this technique was developed by Reddit users. (Face2Face) is another technique used to generate deep fakes, and it is based on a three-component pipeline. First, check the identities of both real and fake data, then transform from real to fake, and finally, the render step reduces the inconsistencies found in the data we fake [8].

VGG is another deep fake creation strategy that depends on a generative adversarial network. This technique relies on the addition of two layers, one called adversarial loss and the other perceptual loss. Both were added to the autoencoder-decoder to generate bogus images [9].

Cycle GAN is a deep fake technique that uses GAN to replicate the lineaments of one image. This method is considered unsupervised because it learns self by self [10].

1.5.2 Deep Fake Detection

The process of detecting deep fakes has been going through various levels for years, and there are many deep fake methods, but in this research, we provide to classify them into two addresses, which are:

1.5.2.1 Image detection

Detecting deep fake image have a lot of techniques which had made by the researcher. And in this section, we will discuss some of them.

Shahroz Tariq and his staff [11] propose a method that relies solely on analyzing image content, this technique improves the chances of detecting deep fake images created automatically or with human assistance by 94% and 74.9%, respectively.

Haodong Li with help of his group [12] advocate a strategy using image colors; they discovered differences in colors between real and fake images, particularly (red, green, and blue); this aids in detecting manipulated images.

According to Xinsheng and other researchers [13] they believe that the generation of deep fakes does not yet focus on forensics generalization capabilities, so they attempt to solve this problem by generating two images. one of Gaussian Blur and the other of Gaussian Noise They also claim that this will allow them to ignore frequency clues and focus on frequency pixels. In this way, we can distinguish between real and fake images.

Peng Zhou with his group [14] propose a novel approach that uses a two-stream network to patch a triple stream using two face-swapping applications. This method first employs a two-stream network to manipulate detection using GoogleNet, followed by the mission of patching triplet stream steganalysis noise residuals and low-level captured cameras. When this is done, the two face-swapping applications will generate a new dataset; this method is referred to as a hybrid because it can learn both fake and real images.

Also, we have an approach that comes as a hybrid approach, and it uses a pairwise-learning at first creating a fake image by GANs then pairwise learning comes to know which image is real and which is not real [15].

1.5.2.2 Video detection

Last year's fake videos improved so much that we could not detect them with normal eyes, so a lot of research was done to try to reduce this and distinguish fake from real videos; here we explain some of them.

Yuezun Li and his group [16] propose a method based on eye blanking detection that employs (CNN) with the assistance of (RNN) to learn the movement of the eye and blanking.

Ciftci and other people [17] proposed yet another approach. This method is based on a heartbeat. This approach starts with a layer that includes all real videos, then the next layer includes fake videos, which is called registration, and the final layer is used to discover manipulated videos; this approach succeeds by 97.3%.

Mittal with help of others [18] hypothesize that the emotional way to discover fake videos The method uses network architecture to sync audio and visuals then analyzes the face in the video, audio respectively., and then uses triplet loss to determine fake or real.

Bansal and his group [19] discovered Recycle Gan It is based on the collection of spatial and temporal data, which aids in the detection of real and fake.

Sabir and his staff [20] discovered a new method with two levels used to analyze temporal and spatial features by going through face processing, which used (spatial transformer networks).

And then face manipulation, which used convolutional networks. Finally, it will reveal which videos are genuine and which are not.

1.6 CONTRIBUTION

The attempt of this research is for developing the detecting deepfakes software program with the perfect accuracy to help users distinguishable between original and manipulated videos.

1.7 THESIS OUTLINE

The remaining thesis chapters are written as follows.

Chapter 2: This chapter provides information on social media, including how it can be used to both create and identify deep fakes.

Chapter 3: This chapter provides information on the various learning styles and their distinctions, as well as certain difficulties.

Chapter 4: The core idea of producing deep fakes utilizing the sim swap method will be covered in detail in this chapter.

Chapter 5: The purpose of this chapter is to provide information on several well-known data sets that will be used in both the creation and detection of deep fakes.

Chapter 6: In this chapter, we'll go into more detail about our model, how it functions, and the methods we applied to it.

Chapter 7: This chapter will discuss our findings and results while also including some examples from our work.

Chapter 8: This chapter will provide the conclusions and offer some suggestions for scholars looking to complete their work in this area.

2. SOCIAL MEADIA

2.1 INTRODUCTION

Social media is a type of technology that allows people to share ideas, photos, and videos via the internet. It allows users to communicate quickly and easily. Nowadays, social media has such an impact on our lives that there isn't a single person on the planet who doesn't understand how to use it or share things on it. A large amount of data is shared daily around the world by various users of various social media platforms, according to a large number of users.

2.2 POPULAR TYPES OF SOCIAL MEDIA

2.2.1 Facebook

Is a popular free social media platform that allows users to communicate with each other and share their ideas, photos, and so on (Figure 2.1). Students from Harvard University Chris Hughes, Andrew McCollum, Dustin Moskovitz, Eduardo Saverin, and Mark Zuckerberg started it in 2004.

The Menlo Park, California-based company has recently changed its name from Facebook to (Meta). There were 2.80 billion Facebook users worldwide in 2021.

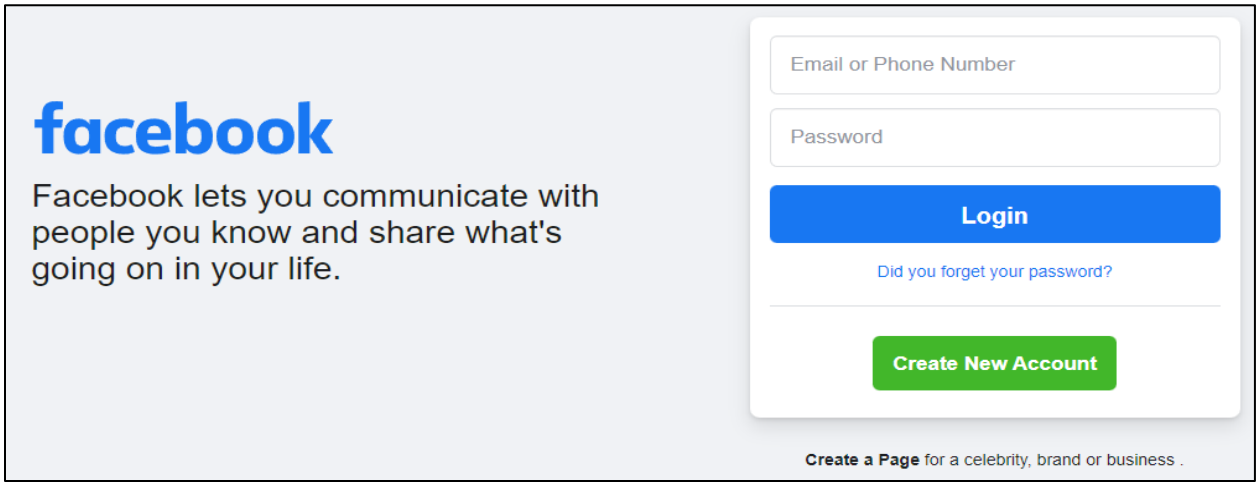

Figure 2.1: Facebook Login Page.

2.2.2 Twitter

It is a well-liked free social media site that enables users to share brief posts called (tweets) that can include photos, videos, and other media (Figure 2.2). It was created in March 2006 by Jack Dorsey.

Twitter is headquartered in San Francisco, California. Twitter had 69.3 million users in recent days.

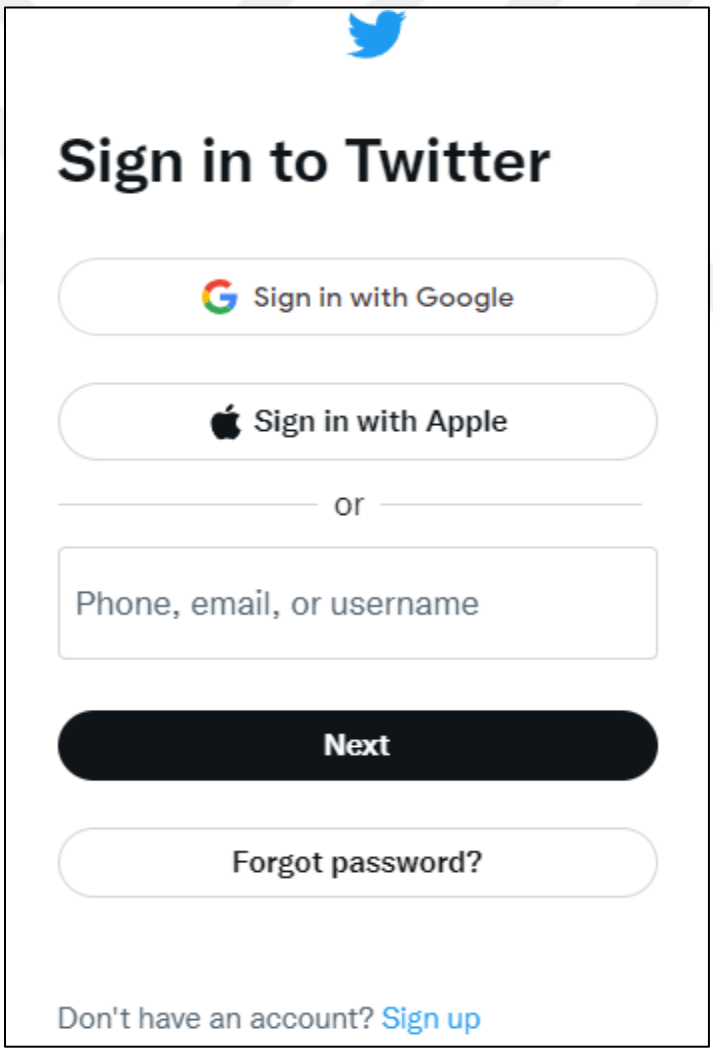

Figure 2.2: Twitter login page.

2.2.3 YouTube

Is a video-sharing service that allows people to share their videos or make likes and comments on other people's videos (Figure 2.3). It can be accessed via computer or mobile. Steve Chen, Chad Hurley, and Jawed Karim registered it on February 14, 2005.

The headquarters of the corporation are in San Bruno, California. two billion YouTube users globally in 2021.

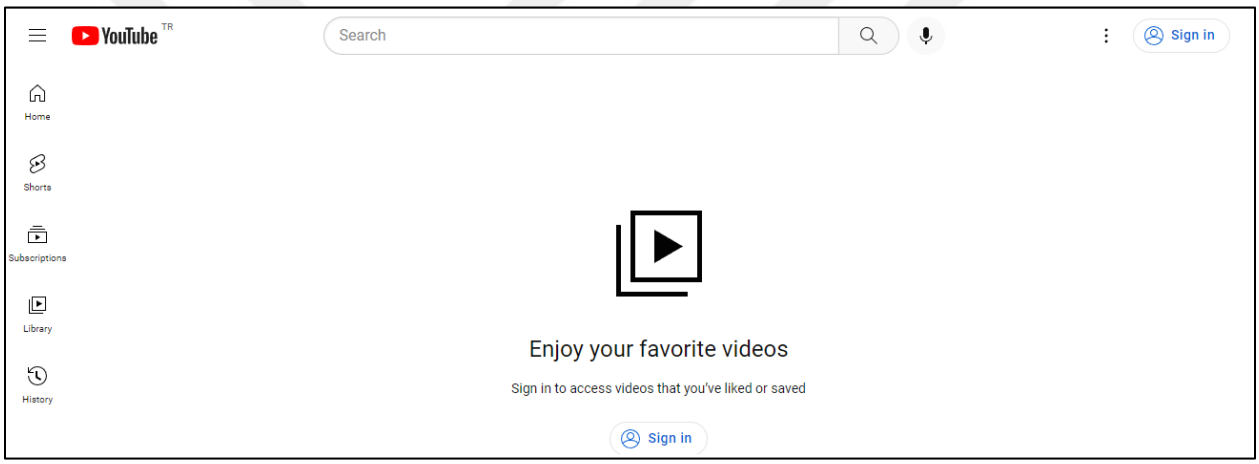

Figure 2.3: YouTube login page.

2.3 SAMMURY

As a result, this social media data with malicious intent may be more dangerous to users because the creator of the deep fake can easily access all pictures or videos that will aid him in creating deep fake data. However, it enables deep fake detectors to verify the authenticity of a video or image, allowing social media to be used in both cases (deep fake and detecting deep fake).

3. CONCEPT OF DEEP LEARNING AND MACHINE LEARNING

3.1 INTRODUCTION

First, it is worth noting that both machine and deep learning are regarded to be components of artificial intelligence. So, we need to define artificial intelligence after that we can go through the learning techniques. (Figure 3.1) shows the differences between them.

i. Artificial intelligence

It is the methods that can make the computer work like the human mind. Some famous programming languages that can be used in AI are (python, java…etc.).

ii. Machine learning

It is that part of artificial intelligence which can be defined as a method to create a system to do specific functions without any clear special instructions.

iii. Deep learning

It is that part of machine learning which can deal directly with data (images, texts, or sounds).

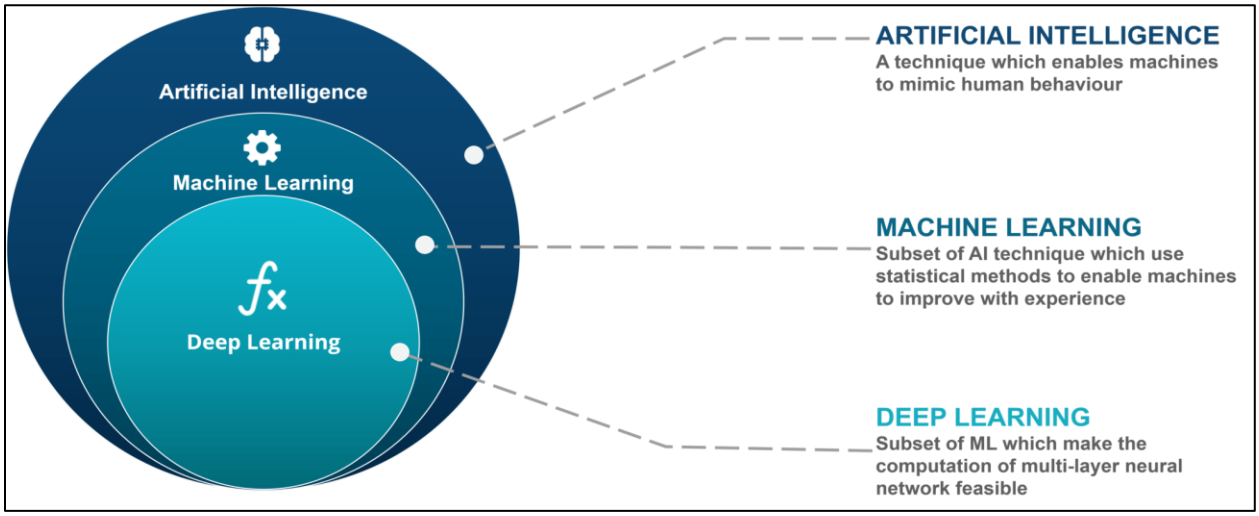

Figure 3.1: AI-vs-ML-vs-Deep-Learning [20].

3.2 MACHINE LEARNING AND DEEP LEARNING DIFFERENCES

3.2.1 Layers

3.2.1.1 Machine learning layers

i. Supervised Machine Learning

Supervised machine learning (Figure 3.2) This technique teaches the computer how to organize input and output in accordance with pairs of input and output. It includes (Regression, Classification).

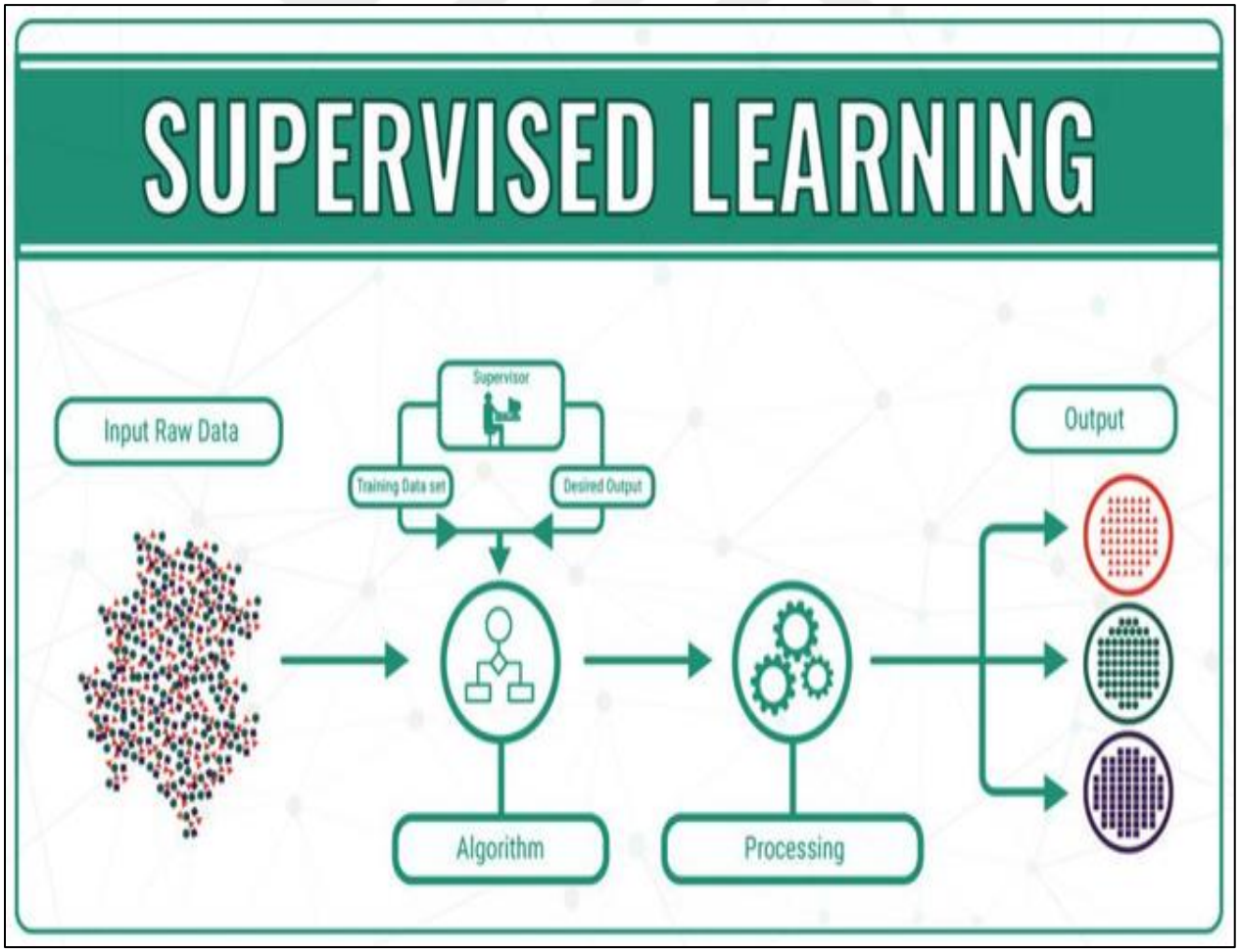

Figure 3.2: Stages of Supervised Machine Learning [21].

ii. Unsupervised machine learning

Unsupervised machine learning (Figure 3.3) It is a type used to look for unidentified, unaddressed patterns in datasets. The data can autonomously control itself after adhering to specific patterns. It includes (Clustering).

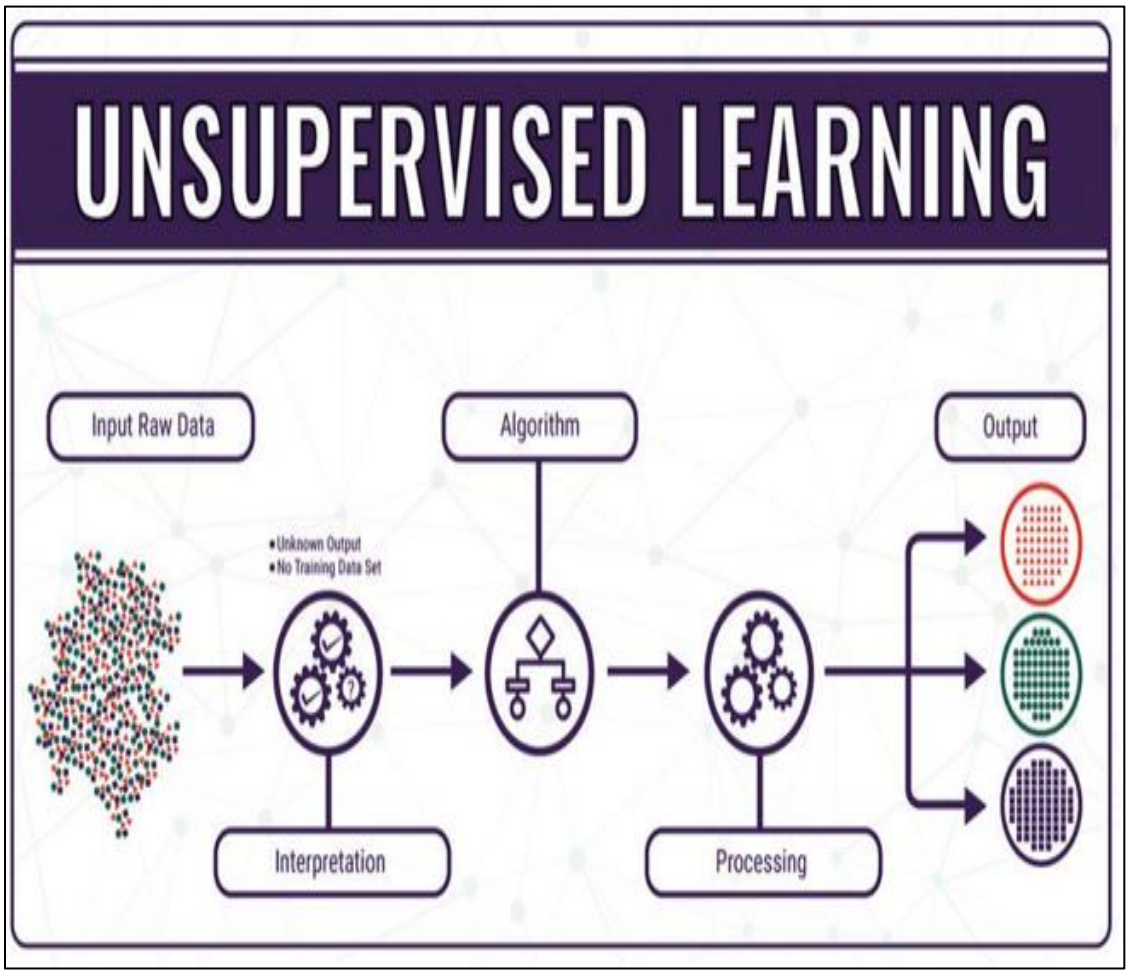

Figure 3.3: Stages of Un Supervised Machine Learning [21].

iii. Machine learning algorithms for reinforcement

Reinforcement learning (Figure 3.4) It focuses on the function of the shrewd operator in making a choice in a situation, which will help to strengthen the idea of expanding advantages. It includes (Indirect learning, Direct learning).

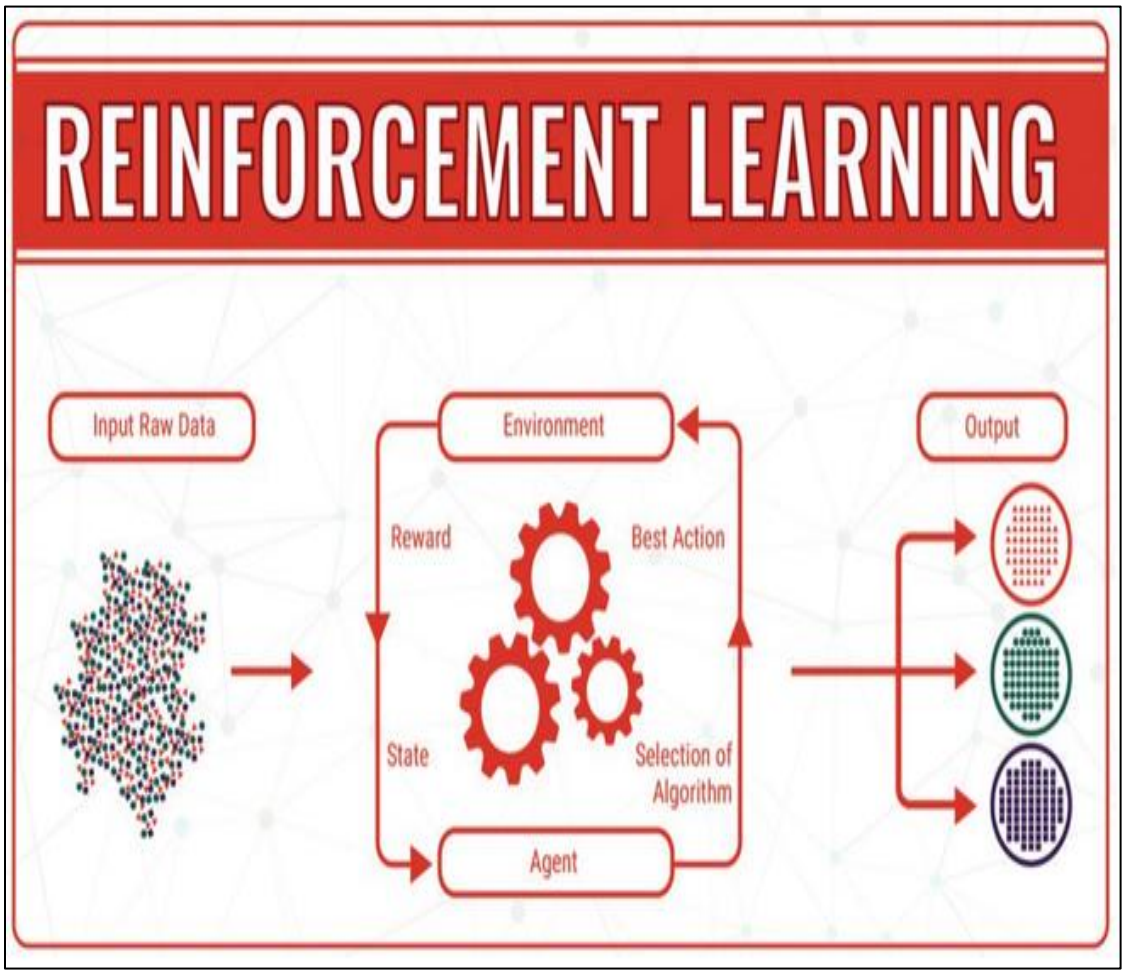

Figure 3.4: Stages of Reinforcement Machine Learning [21].

3.2.1.2 Deep learning layers

i. Input layer

This layer is in charge of receiving data.

ii. Hidden layer

It is in charge of operations on the come and exit layers. It can be more than one hidden layer.

iii. Output layer

It is in charge of displaying the end result.

The (Figure 3.5) shows the different layers of deep learning.

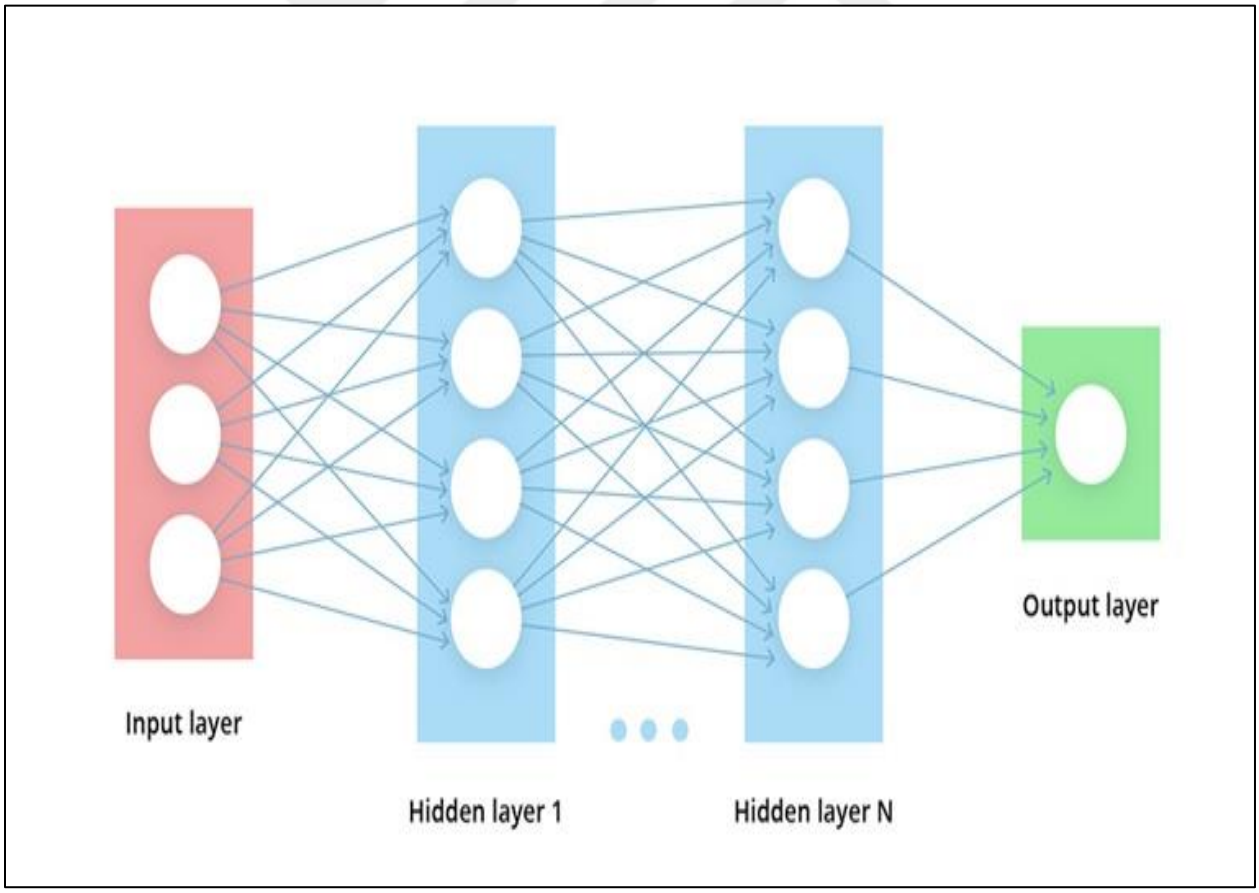

Figure 3.5: Deep Learning Layers.

3.2.2 Subset

Machine learning is a part of artificial intelligence, whereas deep learning is a part of machine learning.

3.2.3 Data Requirements

This is determined by the amount of data. Machine learning requires small or medium amounts of data to understand simple data. However, when the problem is complex, we must use deep learning because it requires a large amount of data to get it right and produce a perfect result.

3.2.4 Accuracy

The accuracy of machine learning is less than that of deep learning, which has infinite accuracy.

3.2.5 Training Time

Machine learning takes a brief period to manipulate, which could be a few minutes or hours. While deep learning manipulation time will be lengthy, it could take weeks.

3.2.6 Hardware Reliance

Because machine learning is performed on low-end machines, it requires a central processing unit (CPU) to train. When deep learning requires training in graphics processing units (GPU), it works on a high-end machine.

3.2.7 Tuning the Hyperparameters

Machine learning has limited tuning options. While deep learning can tune in any way available.

3.2.8 Algorithm Structure

Deep learning has a more complex algorithm structure than machine learning.

3.2.9 Learning

Machine learning requires Manuel input, whereas deep learning generates input automatically.

3.2.10 Problem Solving

Machine learning is thought to be easier to solve than deep learning because it divides each problem into simple different tasks and then gathers them all together to give us the final result. While deep learning employs a complex approach to problem-solving, it is known as (end to end approach).

3.2.11 Interpretation

In machine learning, interpretation is much easier than in deep learning because deep learning algorithms are difficult to understand and explain why the results are as they are.

3.2.12 Create a Feature

Machine learning requires human interaction to create new features, whereas deep learning can do so automatically.

3.2.13 How it Works

Machine learning works by running algorithms that learn to perform a task and provide a result. While deep learning spans many layers to explain connection and data quality.

3.2.14 How it Manages

An algorithm that is determined by test data to examine specific variables in raw data. Deep learning, on the other hand, differs in that algorithms test data automatically when the operation begins.

3.2.15 Design Opportunities

Machine learning requires information from human on how to present data. Deep learning, on the other hand, requires just data implementation knowledge as shown in (Figure 3.6).

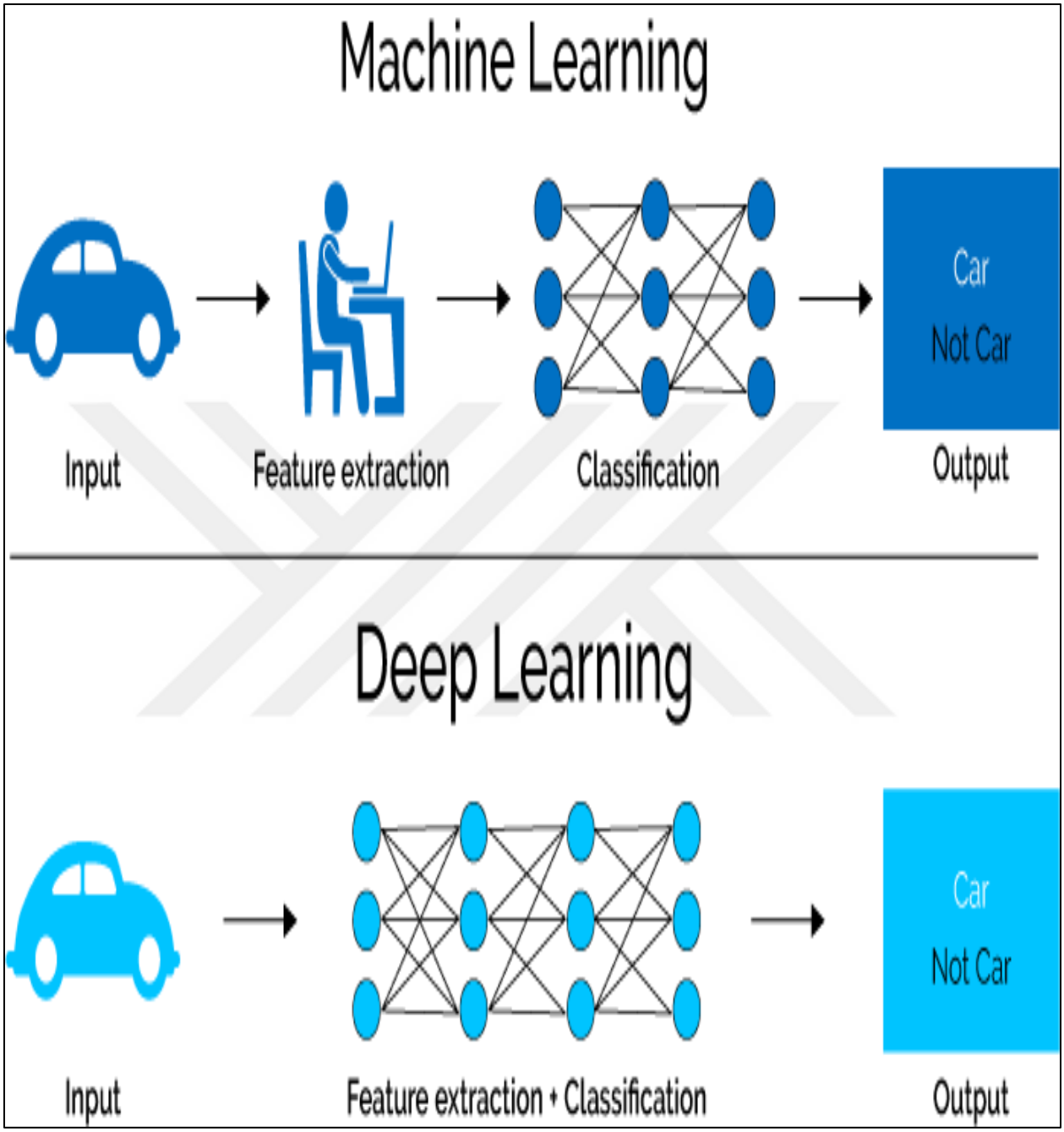

Figure 3.6: Differences in Design Opportunities in Learning Types [22].

3.3 CHALLENGES OF MACHINE LEARNING

3.3.1 Gathering Data

It’s important to mention that when gathering data, we need to make contact with professional persons in certain regions to make sure that we are gathering data with good quality.

3.3.2 Minimum Size of Coaching Data

After checking data quality, we need to select a certain algorithm and train it by feeding our machine a suitable size of data to avoid errors.

3.3.3 Unknown Coaching Data

It’s better to predict future vision to feed machines with data that may not need it now but in the future to avoid feeding it again when the world change.

3.3.4 Less Data Quality

To avoid errors, we must check data quality before feeding it to the machine.

3.3.5 Un Useful Peculiarities

Some data are un useful so we need to avoid using them to reduce them as much as we can to get the result that we want.

3.3.6 Excessive Coaching Data

It’s better to be careful when giving data to a machine to avoid excessive machine opinion.

We can do that by using the following steps:

i. Gathering a huge number of coaching data.

ii. Choose methods with fewer peculiarities.

iii. Delete extreme values.

iv. Decrease similarity data.

3.3.7 Unsuitable Coaching Data

It's the opposite of excessive coaching data and it happens when the machine didn’t get enough data.

To avoid this, we can use the following steps:

i. Increase learning algorithms quality.

ii. Delete un useful data.

iii. Choose a good quality model.

3.3.8 Noise Learning and Deployment of Model

When we establish an application, we should follow the below steps:

i. Gather data.

ii. Check data quality

iii. Manipulate data.

iv. Check the manner.

v. Exercise model.

vi. Deployment.

3.4 CHALLENGES OF DEEP LEARNING

i. Check if the model is complex or not.

ii. Evaluate the performance of the model.

iii. Improve better plans to do efficient tasks.

iv. Reduce the time that we need to learn and get the last result.

3.5 SUMMARY

We attempted to focus on the differences between learning methods in this chapter because it will help us better understand the process of feeding machines with data. First, we should care about the data quality that will feed our system, so an approach of filtering data with high quality and deleting unnecessary data guides us to good results when supplying the system.

4. CONCEPT OF CREATING DEEP FAKE

4.1 INTRODUCTION

The process of creating deep fake consists of two steps (Encoder, Decoder)

i. Encoder

The encoder is the step responsible for earning the lineaments of root data.

ii. Decoder

The decoder obtains the lineaments taken from root data to create deep fake data.

While the encoder is identical between source and target, The decoder contains two identity specific decoders (decoder source and decoder target).

4.2 WORKING MECHANISM

Both the encoder and decoder go through two levels (training and testing).

i. Training level

In this level, the Enc-DecS takes the distorted image and return it to the original image.

ii. Testing level

At this level, the Enc-DecT will take the target image and send it to Enc-DecS to make it distorted with identity and attribute.

During this process, the encoder takes target linemates that contain attributes and identities, while the decoder is responsible for changing target linemates to an image with the source's identity, which means the identity of the source image should be inserted with Decs weights. To avoid this, we must find a way to separate the identity information from the decoder, which will aid in the generalized identity of the entire architecture.

This model enhances the architecture by including an additional ID Injection Module between the Encoder and the Decoder. This ID injection is used to replace the identity of a target with the identity of the source without changing the attribute.

Since it is difficult to distinguish between identity and attribute in linemates due to their high coupling, we need to train the network to learn which parts of linemates should change and which should not.

4.2.1 ID Injection Module parts

ID Injection Module contains two parts (Identity extraction part).

i. Identity extraction part

This part directly manipulates the input source which has the identity and attribute of the source face. And because we just need the former, we ask the help of the face recognition network to extract the identity vector from the input source.

ii. Embedding part

In this part, the ID-Blocks are used to inject the identity information into the features with help of Adaptive Instance Normalization (AdaIN) [23], The main formulation of AdaIN given as

$$\text{AdaIN}\,(\text{Fea}, v_s\,) = \sigma_S\,\frac{\text{Fea}\;-\;\mu(\text{Fea})}{\sigma(\text{Fea})}\;+\;\mu_s \tag{4.1}$$

4.3 RESULT OF CREATING DEEP FAKE

The result of the sim swap [24] is displayed in (Figure 4.1), and it can be divided into three parts (Input picture, Input Video, and Final result)

i. Input picture

The input picture this is the image of the target person's face that we will use to generate the modified video of the subject.

ii. Input video

The input video is the original video that we want to alter. To do this, we will replace an original source person with a fictitious target person by superimposing a photograph on the input video.

iii. Final result

The final result will enable us to achieve our findings and objectives. By replacing the target person's real face with a fake one and producing a manipulated video of them.

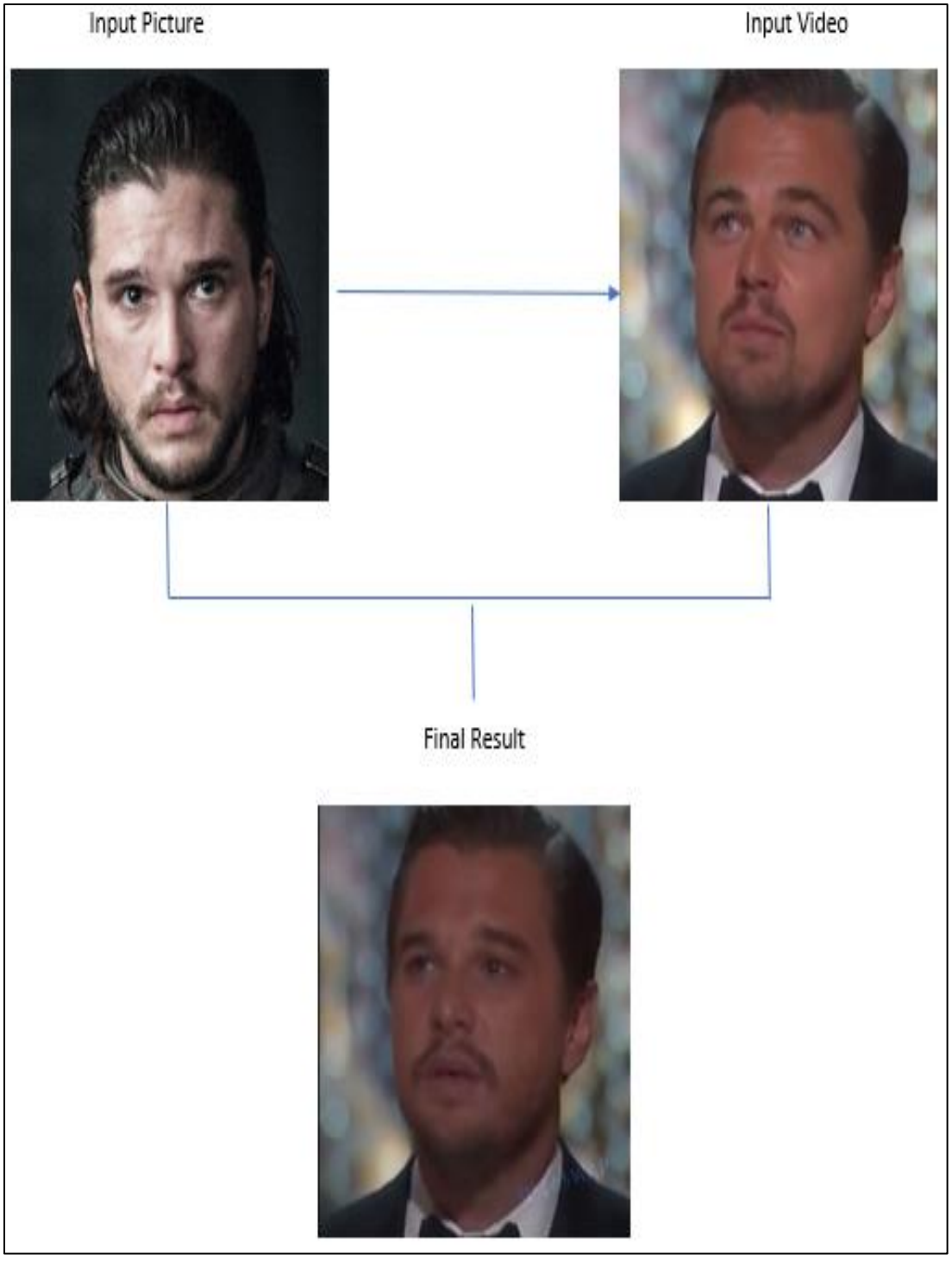

Figure 4.1: Deep fake creation by using Sim Swap technique.

4.4 SUMMARY

In the future, deep fakes will become increasingly valuable as the number of applications that can access by anyone in the world with Internet access grows. As a result, the danger of deep fake creation grows by the day. In this section, we attempt to provide information on how deep fake can create using a deep learning system called sim swap which involves swapping the original face with a fake face using an ID Injection Module.

5. DATA SETS

5.1 INTRODUCTION

The number of deep fake data sets has increased since 2018. To begin, we should define data sets as groups of data that are logically linked together. Deep fake videos can be created using a variety of data sets. the most readily accessible data sets for usage in the disciplines of deep fakes production and detection are presented in this section includes Table 5.1 and Table 5.2 that will serve as the link [25] for each dataset that will discuss in this chapter.

5.2 PUBLIC DATA SET

5.2.1 Image Datasets

5.2.1.1 FFHQ dataset

FFHQ stands for Flickr-Faces-High-Quality developed by Karras and his group [26] it contains 70.000 high-quality face images created by using a Generative Adversarial Network (GAN).

5.2.1.2 100K-faces dataset

100K-Faces as the name implies, it contains 100.000 human faces generated by a model called Style Generative Adversarial Network (Style GAN).

5.2.1.3 DFFD dataset

DFFD which stands for Diverse Fake Face Dataset, developed by Dang and others [27] containing 100.000 people and 200.000 photos made with models (Pro GAN and Style GAN). Image examples include both male and female photographs. Pro GAN is a generative adversarial network using a progressively increasing training approach.

The Style Generative Adversarial Network is an enhancement to the GAN architecture that provides management over the separated style aspects of image features.

5.2.1.4 CASIA-web face

Developed by Dong and others [28] which stands for Chinese Academy of Sciences' Institute of Automation Web Face.

It provides 10.000 subjects with 500.000 images of famous Internet Movie Database (IMDB) actors and actresses extracted using clustering methods.

Table 5.1: Image Datasets

Groups	Link
FFHQ	https://github.com/NVlabs/stylegan
1000K-Faces	https://generated.photos/
DFFD	https://github.com/NVlabs/ffhq-dataset
CASIA-Web Face	https://paperswithcode.com/dataset/casia-webface

5.2.2 Video Datasets

5.2.2.1 Deepfake-TIMIT dataset

Deepfake-TIMIT dataset, which was developed by Korshunov and his group [29] at Texas Instruments Massachusetts Institute of Technology.

It is considered the first dataset to combine videos using face swap-GAN, resulting in 640 videos ranging in quality from low to high depending on the image resolution (64-128).

5.2.2.2 Face forensics++ dataset

A new dataset emerged in 2019 called (face Forensics++ dataset) [30] which contains 1000 videos generated automatically from YouTube-8M. Four approaches to fake video are included in this dataset, two of which are based on computer graphics and two on deep learning. Additionally, it contains three different types of datasets (raw data, medium data, and high data), which are used to test one model (train, validation, and test).

5.2.2.3 DFDC dataset

DFDC This acronym stands for Deep Fake Detection Challenge. It was released by Facebook and can train 119,154 fake videos and 19.154 original videos.

5.2.2.4 Celeb-DF dataset

Celeb-DF is a type of dataset that stands for Celebrities Deep Fake. This data set was recently released and contains 560 real movies and 5639 fake ones.

5.2.2.5 VGG face2 dataset

A sizeable face dataset was contributed by Cao and his staff [31] (VGGFace2). There are more than three million face shots in it, with over 9,000 different people represented by an average of over 300 images per subject

5.2.2.6 Eye-blinking dataset

The (Eye-Blinking datasets) were published by Li and his group [12] This data collection is unique and is concerned with blinking eyes, which is not present in the other datasets discussed previously.

It also has more than 50 interviews. This data set's strategy is to place marks on the left and right eyeballs during around 30-second video runs.

Table 5.2: Video Datasets

Groups	Link
Deepfake-TIMIT	https://www.idiap.ch/en/dataset/deepfaketimit
Face Forensics++	https://github.com/ondyari/FaceForensics/tree/master/dataset
DFDC	https://www.kaggle.com/c/deepfake-detection-challenge/data
Celeb-DF	https://github.com/yuezunli/celeb-deepfakeforensics
VGGFace2	https://www.idiap.ch/en/dataset/deepfaketimit
Eye Blinking	http://www.cs.albany.edu/~lsw/downloads.html

5.3 SUMMARY

This chapter discusses data sets and their history, as they are widely available on the internet for all users. As previously demonstrated, data sets can both generate and detect deep fakes. We provided a link to the most popular data sets on the internet; however, anyone can now create their data set because the definition of data set is a collection of data in a related logically. That means there are other data sets we did not mention because they are not widely used or accessible to users.

6. THE PROPOSED SYSTEM

6.1 INTRODUCTION

The previous chapters provided an overview of artificial intelligence, deep fakes, and popular data sets. This chapter extends the discussion by providing detailed and in-depth discussions of the basic programming language used to build the model and construction mechanism.

The remaining three sections are as follows: Section 6.2 focuses on a detailed framework for the proposed research, whereas Section 6.3 tools are the programming language.

Section 6.4 discuss how system work. And last section 6.5 responsible for giving summary of this chapter.

6.2 THE SYSTEM FRAMEWORK

6.2.1 LSTM

The first technique which will guide us to create a system for detecting deep fake videos will depend on the Long short-term memory layer (LSTM) [32], [33] It is s a type of neural network used in artificial intelligence and deep learning. which is considered a set of Recurrent Neural Network (RNN) and can process all data sequences The Long Short-Term Memory (LSTM) differs from traditional feedforward neural networks in that it has feedback connections. After processing each frame in turn, LSTM compares its properties at various points in time. It is possible to tell whether a video is deep-faked or not by comparing the frames. Any video can be fed into the algorithm for prediction after training. Long Short-Term Memory (LSTM) has 3 gates, they are (forget gate, input gate, and output gate).

i. Forget gate

The first gate in (LSTM) Layer is known as the (Forget gate), This gate oversees categorizing prior material and determining whether it should be forgotten or pushed to the next phase.

ii. Input gate

The second gate in (LSTM) Layer is known as (Input gate), and it oversees classifying information and making decisions that will be added and updated from the current phase.

iii. Output gate

The third and final gate in the (LSTM) layer called (Output gate), it is responsible for giving a value for the next hidden state and passing updated information from the Input gate.

6.2.2 InceptionResnetV2

The second Technique for creating our system is a hybrid of inception and resnetv2 whereas inception focuses on learning and computational cost. Resnetv2 is a neural network architecture focused on computational accuracy. It has 164 layers that are used to detect the object photo, but the final layer is only responsible for analysing the outputs [2], [3].

Batch normalization is a method used by Resnet to expedite training and give some regularization. it is a method used to try to solve the internal covariate shift problem. When employing training a layer deep in a neural network, this issue occurs. The characteristics from the suggested region are extracted using the Inception V2 architectural style faster region-based convolutional neural network (Faster-RCNN).

The network outputs a list of estimated class probabilities after receiving a 299 by 299 image as input. The Residual connection and the Inception structure are combined to form it.

In the Inception-Resnet block, residual connections and convolutional filters of different sizes are combined. In addition to resolving the degradation problem brought on by deep structures, residual connections help save training time.

6.2.3 Open CV

A software library for computer vision and machine learning is called Open CV. It was launched in 1999 under the direction of Bradsky and a team of Intel engineers. It was first made available in the C programming language, but as time went on, it was also implemented in plenty of other computer languages. In this part, we'll outline a few different varieties of computer vision.

6.2.3.1 Face detection

This is a type of computer vision that extracts faces and objects from images or videos. Face detection eliminates background noise and focuses on individual facial features. During face detection, two types of false faces may be encountered.

i. False positive

The first type is a false positive which means there is no face to detect in our images.

ii. False negative

The second type of false occurs when the detector is unable to detect a face in a photograph because many faces have already been identified by humans.

6.2.3.2 Face recognition

One type of open CV is facial recognition, which determines the face based on its appearance. It can recognize faces based on their expressions. When face detection is used to identify faces and filter out the background, face recognition is used to determine if the current face belongs to someone (an animal, human, or other). According to this, we can use both face detection and face recognition in a variety of fields, particularly security and police investigations, to reduce the likelihood of dangerous events occurring.

6.2.3.3 Crop image in open cv

The term crop refers to the process of resizing a real image in specific areas that will bear a new image with a new height and width. This process is provided by open cv, which uses a sorted image in a 2D array to specify height and width.

6.2.3.4 Zoom image in open cv

This is a feature that allows you to change the size of an image while maintaining the quality and resolution of the original image. The original image size should be specified in an open cv (250,250).

6.2.3.5 Haar cascade

A machine-learning technique that detects objects by entering several images with negative and positive characteristics and using them in different frames.

6.3 TOOLS

This model is based on three key tools (python, google drive, google Colab)

i. Python language

As we all know, is the best language for implementing artificial intelligence. Python was created in 1991 by Guido van Rossum. (Figure 6.1) provide the page to download this programming language platform.

Figure 6.1: Python download page.

ii. Google Drive

Introduced on April 24, 2012, enables users to exchange files, synchronize data across devices, and store files in the cloud on Google's servers (Figure 6.2).

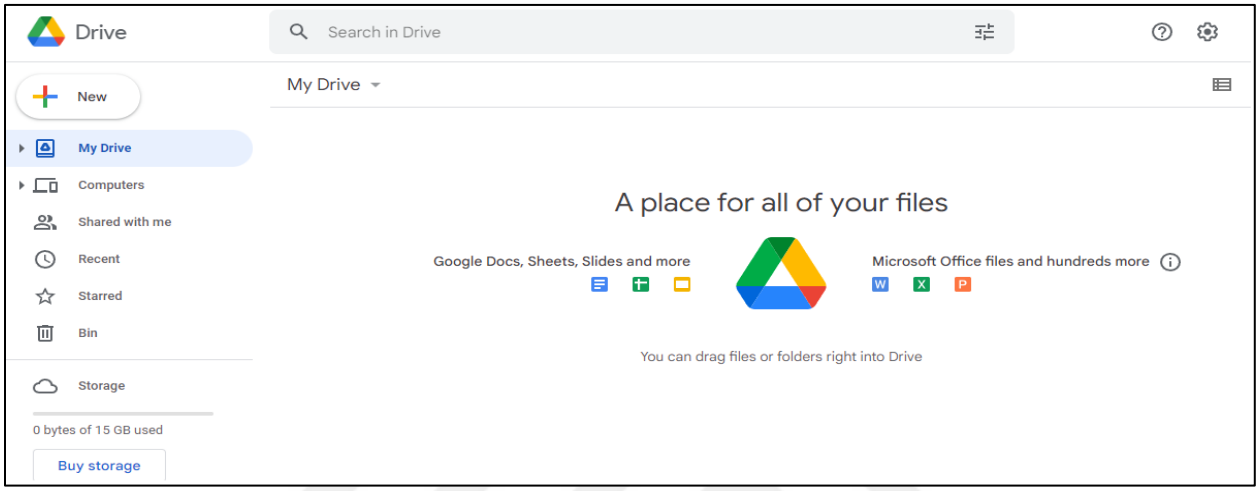

Figure 6.2: Google Drive login page.

iii. Google Colab

Colaboratory, also spelled "Colab," is a Google Research product. Since it allows anyone to create and run Python code through the internet, Colab is especially valuable for machine learning, data analysis, and education. (Figure 6.3)

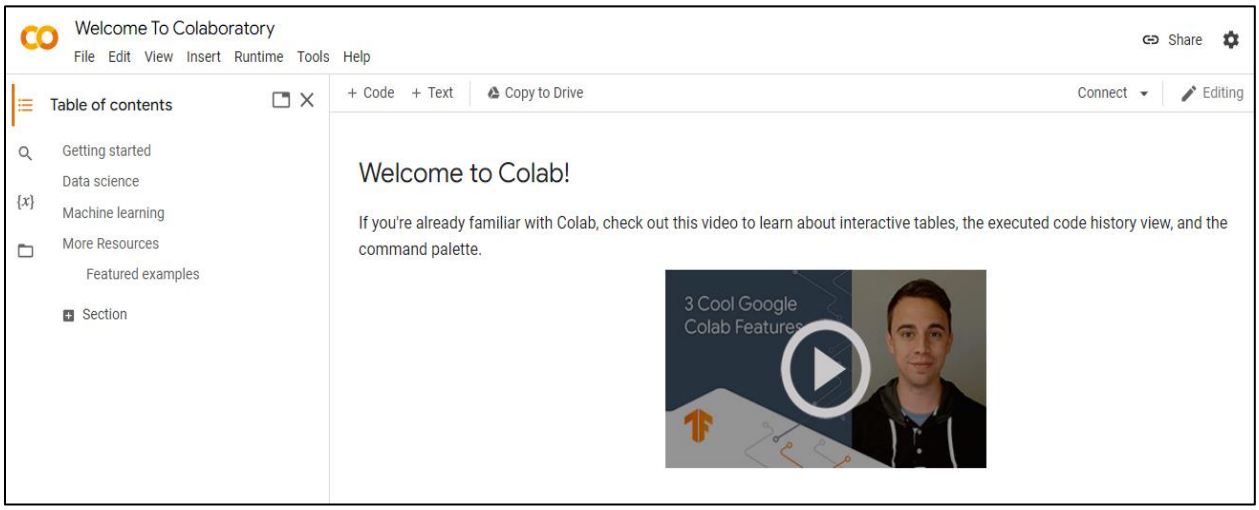

Figure 6.3: Google Colab page.

iv. CSV file

Stands for Comma-separated values and will help us categorize our data as phony or real to train our prediction model. (Figure 6.4) depicts an example of our file.

	A	B
1	file	label
2	000_003.m	FAKE
3	001.mp4	REAL
4	001_870.m	FAKE
5	002.mp4	REAL
6	002_006.m	FAKE
7	003.mp4	REAL
8	003_000.m	FAKE
9	004.mp4	REAL
10	004_982.m	FAKE
11	005.mp4	REAL
12	005_010.m	FAKE
13	006.mp4	REAL
14	006_002.m	FAKE
15	007.mp4	REAL
16	007_132.m	FAKE
17	008.mp4	REAL
18	008_990.m	FAKE
19	009.mp4	REAL
20	009_027.m	FAKE
21	010.mp4	REAL
22	010_005.m	FAKE
23	011.mp4	REAL
24	011_805.m	FAKE

Figure 6.4: Sample from our CSV file.

6.4 IMPLEMINTAION

Our model incorporates three distinct techniques (LSTM, Resnetv2, and open CV). Before it produces a result, it will go through four stages (Pre-Process, Pre-Trained, Training and Classification, and Prediction), as shown in (Figure 6.5) below.

Our model considers (Semi-supervised) which combines both supervised and unsupervised learning. It uses massive amounts of unclassified data and little classified data, combining the advantages of supervised and unsupervised learning without the problems of finding classified data.

6.4.1 Pre-Process

The model will extract images from each frame video and resize them in the first stage after we enter our target videos. The system will check if all inserted videos are suitable or not the codes which showed in (Figure 6.5) will care about detecting corrupted videos before we start our work to not face any problem in other steps. So, if there are corrupted video this group of codes will detect it and before we start, we will have choice to delete it from our dataset.

```
video_fil = sorted(glob.glob(DATA_PATH + '*.mp4'))

print("Total no of videos :" , len(video_fil))
print(video_fil)
count = 0;
for i in video_fil:
  try:
    count+=1
    validate_video(i,train_transforms)
  except:
    print("Number of video processed: " , count ," Remaining : " , (len(video_fil) - count))
    print("Corrupted video is : " , i)
    continue
print((len(video_fil) - count))
```

Figure 6.5: Codes for detecting corrupted video

After that we load labels and videos on the data loader, the model will split them according to our CSV file into (train, and test) it will be considered 80% for the training video and the rest 20% for the testing video.

(Figure 6.6) shows how the system divide inserted data which contain 64 videos to train which contain 38 videos and test which contains 26 as the figure show that in train group there are 21 real videos and 17 Fake videos. When in Test group there are 11 Real videos and 15 Fake videos .

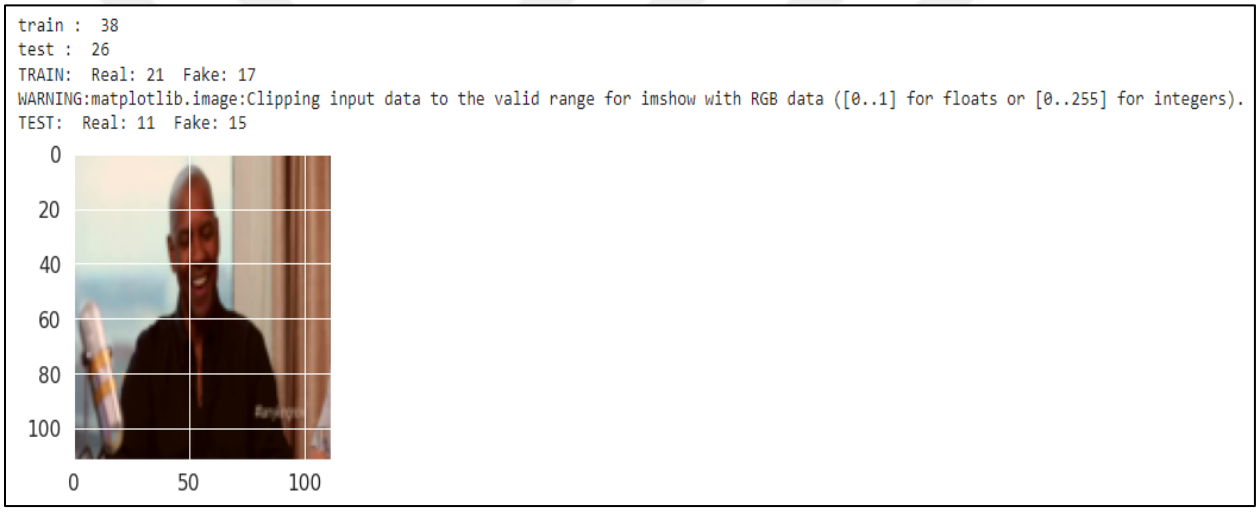

Figure 6.6: The result of split inserted data for train and test

This will provide binary classification by using sigmoid as translator if label is (Real = 1) if label is (Fake = 0).

$$\sigma(x) = \frac{1}{1 + exp\,(-x)} \tag{6.1}$$

The Binary Cross-Entropy loss (or log-loss) must be improving in this scenario, which implies

$$L(x, y) = -[y \log p_y + (1 - y) \log (1 - p_y)] \tag{6.2}$$

When Py = (sx), sx stands for the output of the Convolutional Neural Network for input picture x,

and y stands for the original label value for the input picture. will provide a detrimental impact by positioning a photo that has a high possibility of being on the wrong side. Equation 6.2 optimize the loss function using stochastic gradient descent (SGD) [34]. The data will be normalized so that it is consistent across all fields. After all those processes are completed, it will be passed on to the next stage, which is known as pre-trained.

6.4.2 Pre-Trained

This stage will use Inception resnetv2 for image classification, regression, and feature extraction. Inception Resnet V2 mission is to remove the loss layer and exchange it with the output layer called (the loss output layer) which was defined previously while pre-processing.

The results of our second stage will direct us to the third stage, which is referred to as (training and classification).

6.4.3 Training and Classification

This stage contains three distinct layers.

i. First layer

The First layer in our series called (LSTM layer) This strategy deals with data in various terms sequentially. Convolutional Neural Network outputs can be used as input for LSTM. After processing each frame in turn, LSTM compares its properties at various points in time.

It is possible to tell whether a video is deep-faked or not by comparing the frames. Any video can be fed into the algorithm for prediction after training.

ii. Second layer

The second layer in this stage called (Inner layer) Convolutional Neural Network (CNN) is used in this layer. It oversees taking data and transmitting it to the rest of the network. Will be directed to the pooling layer in our situation.

iii. Third Layer

The third layer in this stage is named (Pooling layer) This layer is considered a hidden layer. it oversees extracting sharp and smooth features, as well as reducing the spatial size of the input image, which reduces the number of computations in the network. Pooling down sampling already minimizes the size and sends only the necessary data to the next layer of the Convolutional Neural Network.

After all, the processes done in this stage will pass to the final stage which is called (prediction).

6.4.4 Prediction

The prediction oversees the detection of fake videos. At this level, we introduce open cv face recognition techniques, which are used to detect faces in video frames. Face aligning is a computer vision technique used to determine the shape of the face components. The model will zoom into the face.

Open CV library techniques are also utilized to extract images from a single video after focusing on the face, allowing a user to distinguish between original and phony face.

To be more effective there is an optional choice to create zoomed animation. (Figure 6.7) shows how the processes run from insert video till give final results.

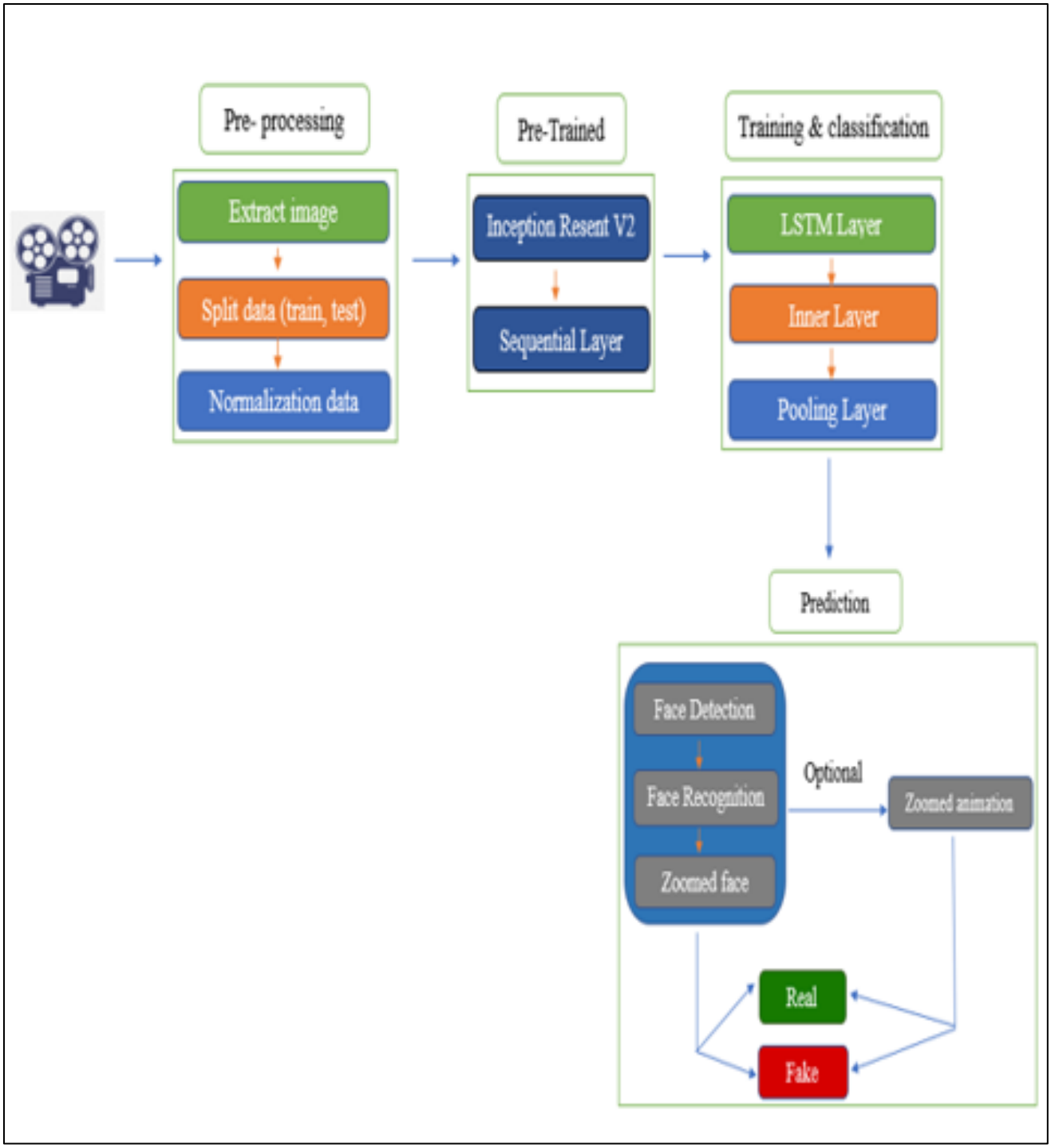

Figure 6.7: Overview of model stages.

6.5 SUMMARY

Our framework, described in this chapter, has been submitted to the system under development, which will decide whether the video entered into the system is authentic or froggy. Programming language like Python, which is rich in algorithms and techniques, allows our system to achieve the goal.

7. EXPERIMENTAL RESULTS AND DISCUSSION

7.1 INTRODUCTION

On this chapter, we will discuss our results and fundings and to be easier we can divide into three categories

Section 7.2 will discuss first category this category cares about checking and giving results (fake or real).

Section 7.3 will discuss our second category which is consider as an optional but will improve prediction because it will be visible to the naked eye.

Section 7.4 will discuss third and last category this category responsible for display the loss of training and validation also will display the accuracy of validation and training during specific epochs as graphical

7.2 FIRST CATEGORY RESULTS

In this category, we will provide inception resnetV2 and LSTM in our picture which we will be extracting a picture from a video, which will generate a color result (RGB) which is a shortcut for (Red, Blue, and Green).

Then we'll use an open cv technique to detect and recognize the face before cropping and zooming the extracted image. Three images will be created as a result.

A. First image

The first image in our series will be an extracted image from the embedded video.

B. Second image

The second image in the same series will detect the face and highlight it.

C. Third image

The third and final image in this series will crop and zoom in on the previously discovered and recognized face.

Open CV techniques are also utilized to extract many images from a single video after focusing on the face, allowing a user to distinguish between a real and a phony face.

This will be beneficial since taking only one image from a single scene in a video may not be sufficient but taking numerous images of the same person in various situations will enable the naked eye to recognize subtle alterations in a person's face.

(Figure 7.1) and (Figure 7.2) shows the results for one original video, (Figure 7.3) and (Figure 7.4) shows the results for one manipulated video.

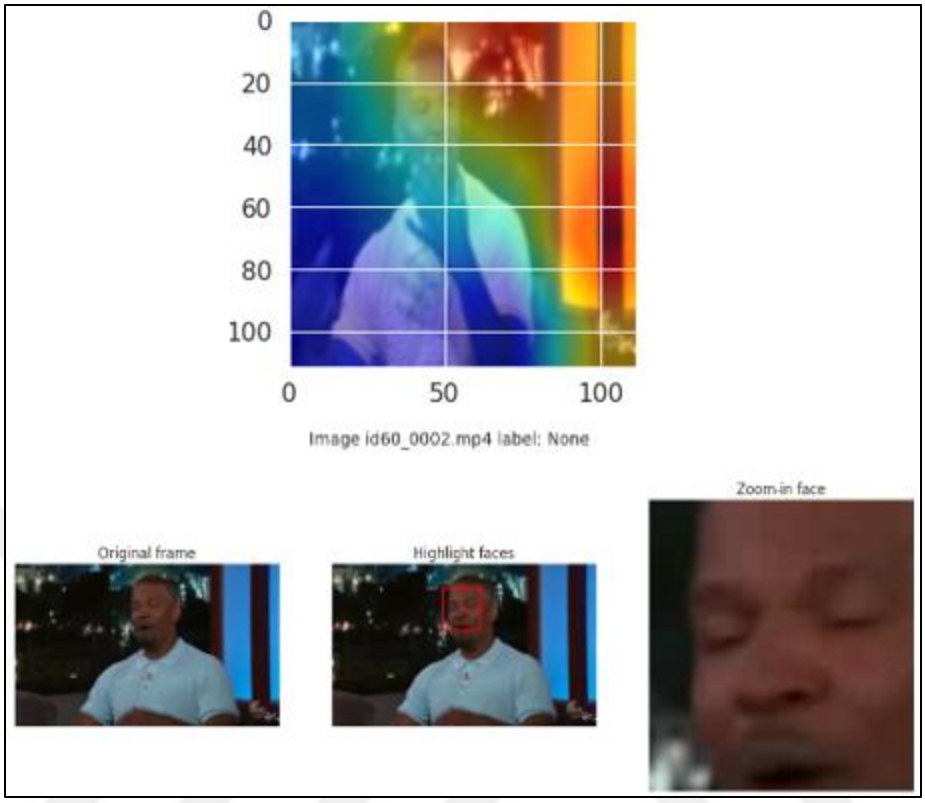

Figure 7.1: ResNetV2 with LSTM and OpenCV for Real video.

Figure 7.2: OpenCV extracting multiple pictures from one Real video.

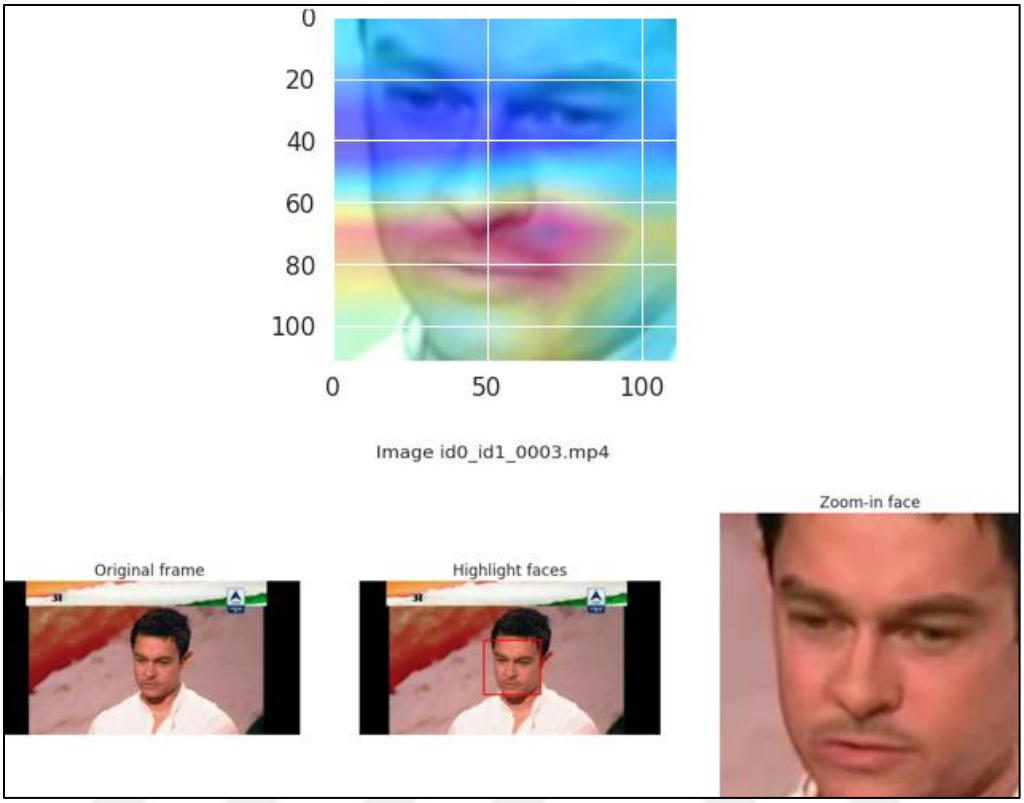

Figure 7.3: ResNetV2 with LSTM and OpenCV for Fake video

Figure 7.4: OpenCV extracting multiple pictures from one Fake video.

7.3 SECOND CATEGORY RESULTS

At this point, the system will display a zoomed animation created with open cv techniques. It also detects, recognizes, and generates a zoomed animation of the face. This step is optional for users. However, it improves prediction because it is visible to the naked eye. After running this optional step, the given of it will provide options that will be under the animation that was created. Those options will allow several facilities for users like run from the beginning (►), run from the end (◄), go little forward (►▏), go little back (▏◄), zoom out (-), zoom in (+), pause with a symbol like (II), go to the end by pressing the symbol (►►▏) or go to begin when press (▏◄◄) .it will also give choices like run once or loop or reflect.

The system will also allocate a big picture of the zoomed face that will extract from the zoomed animation. according to those results, it will be easy to discover changes in manipulated faces.

(Figure 7.5) depicts zoomed animation for the original video, while (Figure 7.6) depicts zoomed animation for the phony video.

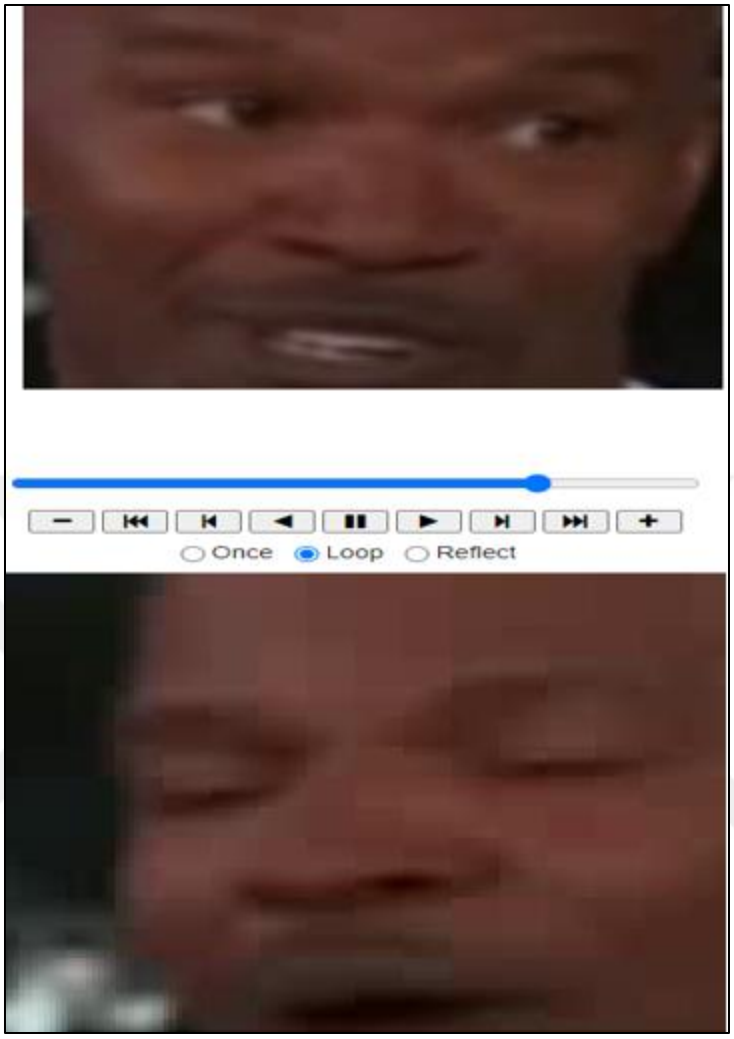

Figure7.5: OpenCV generated the zoomed animation for Real video.

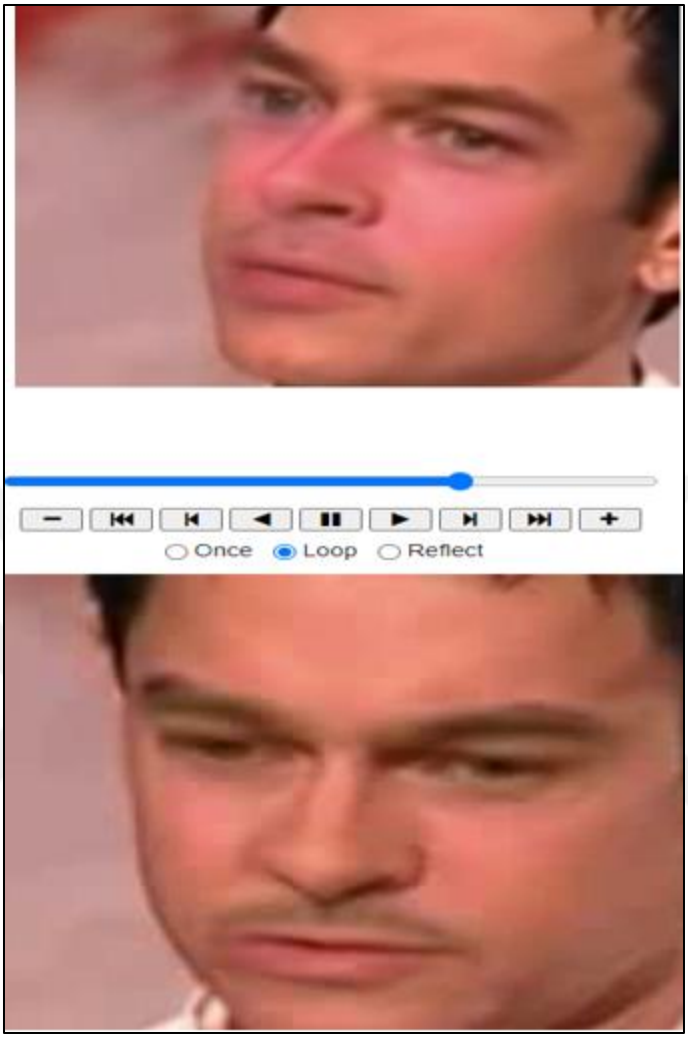

Figure7.6: OpenCV generated the zoomed animation for Fake video.

7.4 THIRD CATEGORY RESULTS

During this stage, we will be able to see the loss in training and validation, along with the accuracy of validation and training during specific epochs. So we will have two graphs in this stage and they are

i. First graph

This graph is concerned with loss since the green lines are the training loss and the blue lines are the validation loss The X-axis shows epochs, and the y-axis shows a loss.

ii. Second graph

In this graph accuracy is taken into account. Green lines represent the training accuracy, while blue lines represent the validation accuracy. The X-axis indicates epochs, and the y-axis indicates the calculated accuracy. This graph clearly indicates that the model was tested using epochs.

For 20 epochs (Figure 7.7), the accuracy is 87.5%. The accuracy is 91.66% for 40 epochs (Figure 7.8), and 93.75% for 60 epochs (Figure 7.9). By examining the above percentages, it is clear that the validation and training accuracy increase as the number of epochs increases.

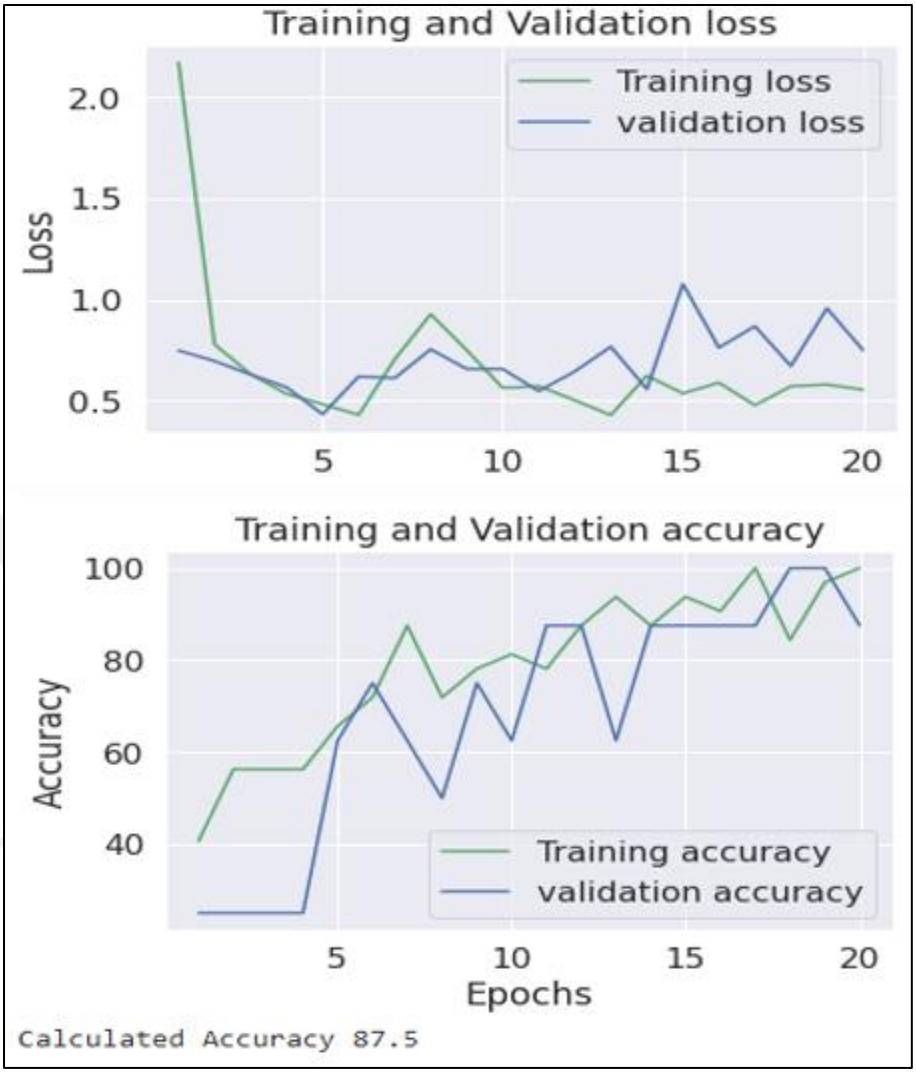

Figure7.7: Loss and Accuracy for 20 epochs.

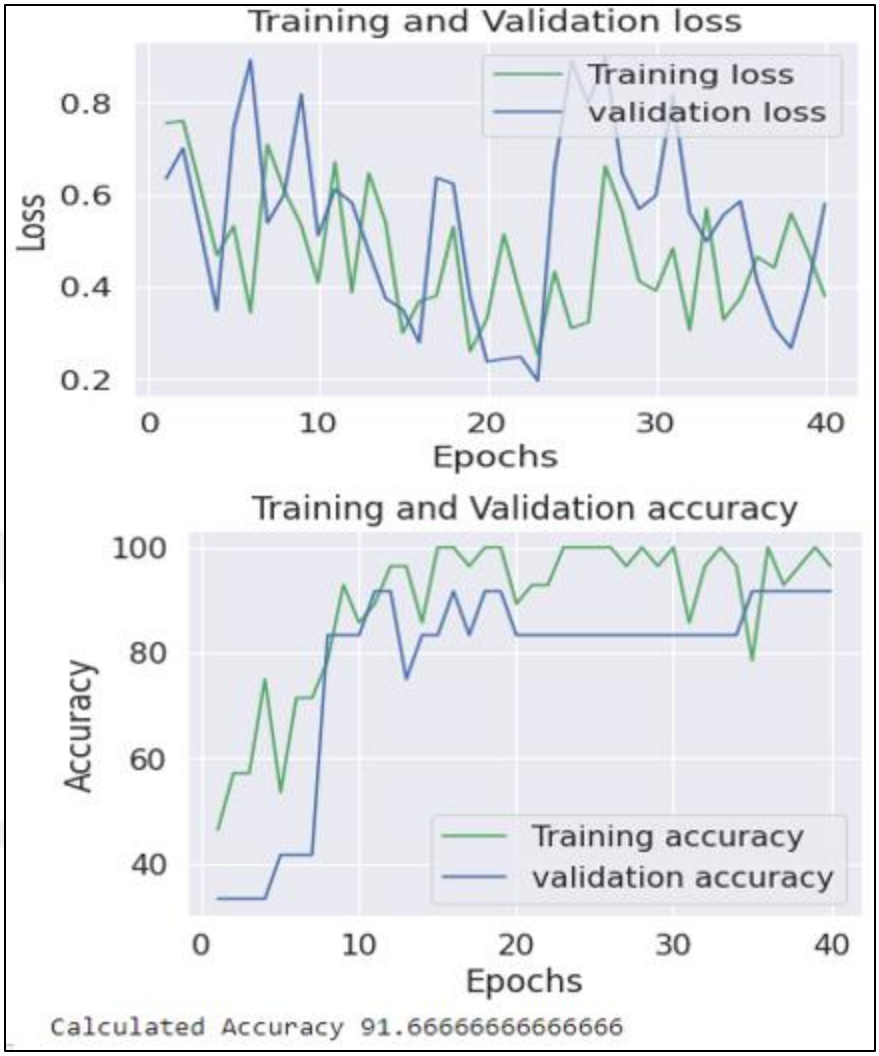

Figure 7.8: Loss and Accuracy for 40 epochs.

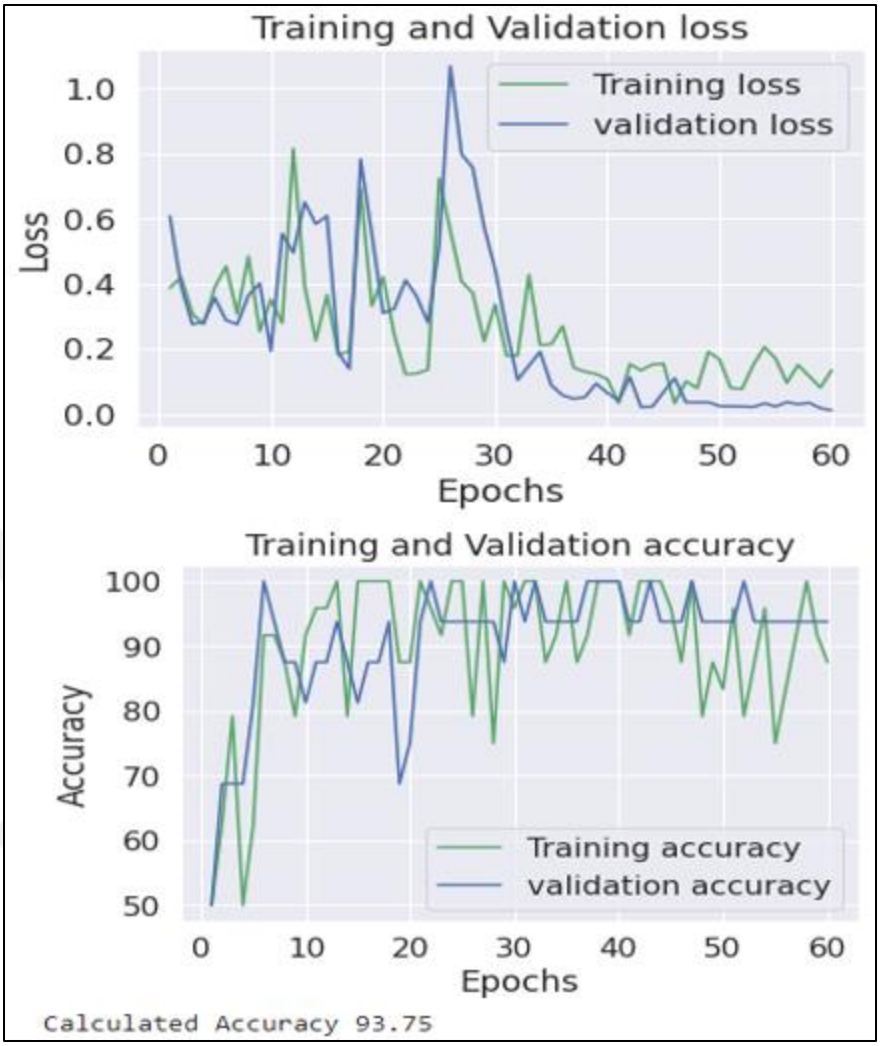

Figure7.9: Loss and Accuracy for 60 epochs.

7.5 SUMMARY

This chapter showed the outcome of our model, which went through several processes and was assisted by ResnetV2 and LSTM to detect the deep fake. Using open cv zoom and creating zoomed animation help the naked eye to determine whether the video is original or manipulated. The chapter also discussed how the accuracy increased as the number of epochs increased.

8. CONCLUSION AND FUTURE WORK

8.1 INTRODUCTION

The key findings, definitions, and outlook function of this research are collected in this chapter.

The study's conclusion is stated in part 8.2.

Recommendations for further work are shown in section 8.3.

8.2 RESEARCH CONCLUSION

Deep faking is a challenge for all of us in the modern world because it is improving so quickly and many users can use it in simple ways according to multiple mobile applications with tons of data that are available daily in social media if it is in bad hands, it may have negative effects in various sections such as (political, economic, or even in the personal side).

Deep fake detection should move quickly to reduce this risk by developing scalable and robust models. In this research, we attempt to present a method for detecting deep fake videos.

The proposed methodology is divided into two parts to detect deep fake videos using deep learning concepts (Inception ResNetV2 and the LSTM) with OpenCV library (face detection and recognition).

This collection of techniques will guide us after many processes to the result which will allow us to detect fake videos from real videos by extracting picture from inserted video then started going through our system until we reach our final goal.

8.3 RECOMMENDATIONS FOR FURTHER WORK

Future work can concentrate on improving accuracy by implementing the genetic algorithm, which will aid in system accuracy improvement by selecting the best accuracy and destroying others.

We can create multiple accuracies from our system discussed previously in the chapter 7 and repeat the process with the help of the genetic algorithm to make the same progress and select the best accuracy and destroy others, and the process is repeated automatically until we reach satisfactory accuracy.

On the other hand, we can improve the system in the future to detect deep fakes in multiple-face videos over time. We can add a technique to the system that will aid in determining whether the audio with inserted video is real or fake.

REFERENCES

[1] Nirkin, Y., Keller, Y., & Hassner, T. (2019). Fsgan: Subject agnostic face swapping and reenactment. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 7184-7193).

[2] Yadav, P., Jaswal, I., Maravi, J., Choudhary, V., & Khanna, G. DeepFake Detection using InceptionResNetV2 and LSTM.

[3] Abhijit Jadhav. (2022, July). Deepfake detection using deep learning. [Online]. Available: https://github.com/abhijitjadhav1998/Deepfake_detection_using_deep_learning.

[4] Aleksandra Deis. (2019, December). Deepfake detection. [Online]. Available: https://github.com/Lexie88rus/Deepfake_detection.

[5] Franceinfo. (2020, November). Plus belle la vie: un épisode utilise l'effet spécial du deepfake. [Online]. Available: https://www.francetvinfo.fr/culture/series/plus-belle-la-vie-un-episode-utilise-leffet-special-du-deepfake_4185041.html

[6] BuzzFeed Video. (2018, April). You won't believe what Obama says in this video! [Online]. Available: https://www.youtube.com/watch?v=cQ54GDm1eL0

[7] Nataraj, L., Mohammed, T. M., Manjunath, B. S., Chandrasekaran, S., Flenner, A., Bappy, J. H., & Roy-Chowdhury, A. K. (2019). Detecting GAN generated fake images using co-occurrence matrices. Electronic Imaging, 2019(5), 532-1.

[8] Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., & Nießner, M. (2016). Face2face: Real-time face capture and reenactment of rgb videos. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2387-2395).

[9] Refik Can Mali. (2020, March). keras vggface. [Online]. Available: https://github.com/rcmalli/keras-vggface.

[10] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).

[11] Tariq, S., Lee, S., Kim, H., Shin, Y., & Woo, S. S. (2018, January). Detecting both machine and human created fake face images in the wild. In Proceedings of the 2nd international workshop on multimedia privacy and security (pp. 81-87).

[12] Li, H., Li, B., Tan, S., & Huang, J. (2020). Identification of deep network generated images using disparities in color components. Signal Processing, 174, 107616.

[13] Xuan, X., Peng, B., Wang, W., & Dong, J. (2019, October). On the generalization of GAN image forensics. In Chinese conference on biometric recognition (pp. 134-141). Springer, Cham.

[14] Zhou, P., Han, X., Morariu, V. I., & Davis, L. S. (2017, July). Two-stream neural networks for tampered face detection. In 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW) (pp. 1831-1839). IEEE.

[15] Hsu, C. C., Zhuang, Y. X., & Lee, C. Y. (2020). Deep fake image detection based on pairwise learning. Applied Sciences, 10(1), 370.

[16] Li, Y., Chang, M. C., & Lyu, S. (2018, December). In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In 2018 IEEE International workshop on information forensics and security (WIFS) (pp. 1-7). IEEE.

[17] Ciftci, U. A., Demir, I., & Yin, L. (2020, September). How do the hearts of deep fakes beat? deep fake source detection via interpreting residuals with biological signals. In 2020 IEEE international joint conference on biometrics (IJCB) (pp. 1-10). IEEE.

[18] Mittal, T., Bhattacharya, U., Chandra, R., Bera, A., & Manocha, D. (2020, October). Emotions don't lie: An audio-visual deepfake detection method using affective cues. In Proceedings of the 28th ACM international conference on multimedia (pp. 2823-2832).

[19] Bansal, A., Ma, S., Ramanan, D., & Sheikh, Y. (2018). Recycle-gan: Unsupervised video retargeting. In Proceedings of the European conference on computer vision (ECCV) (pp. 119-135).

[20] Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., & Natarajan, P. (2019). Recurrent convolutional strategies for face manipulation detection in videos. Interfaces (GUI), 3(1), 80-87.

[21] Atul. (2020, January). AI vs Machine Learning vs Deep Learning. [Online]. Available: https://www.edureka.co/blog/ai-vs-machine-learning-vs-deep-learning/

[22] Ipfconline. (2019, April). MachineLearning Explained: Understanding Supervised, Unsupervised, and Reinforcement learning. [Online]. Available: https://twitter.com/ipfconline1/status/1117042321126764544/photo/1

[23] Narayana reddy. (2020, Novmber). Difference between Machine Learning and Deep Learning. [Online]. Available: https://morioh.com/p/ed56b4fdbf1c

[24] Huang, X., & Belongie, S. (2017). Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the IEEE international conference on computer vision (pp. 1501-1510).

[25] Chen, R., Chen, X., Ni, B., & Ge, Y. (2020, October). Simswap: An efficient framework for high fidelity face swapping. In Proceedings of the 28th ACM International Conference on Multimedia (pp. 2003-2011).

[26] Almars, A. M. (2021). Deepfakes detection techniques using deep learning: a survey. Journal of Computer and Communications, 9(5), 20-35.

[27] Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4401-4410).

[28] Dang, H., Liu, F., Stehouwer, J., Liu, X., & Jain, A. K. (2020). On the detection of digital face manipulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition (pp. 5781-5790).

[29] Yi, D., Lei, Z., Liao, S., & Li, S. Z. (2014). Learning face representation from scratch. arXiv preprint arXiv:1411.7923.

[30] Korshunov, P., & Marcel, S. (2018). Deepfakes: a new threat to face recognition? assessment and detection. arXiv preprint arXiv:1812.08685.

[31] Kumar, A., Bhavsar, A., & Verma, R. (2020, April). Detecting deepfakes with metric learning. In 2020 8th international workshop on biometrics and forensics (IWBF) (pp. 1-6). IEEE.

[32] Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018, May). Vggface2: A dataset for recognising faces across pose and age. In 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018) (pp. 67-74). IEEE.

[33] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[34] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE transactions on Signal Processing, 45(11), 2673-2681.

[35] Infante Molina, A. G. (2020). Learning to detect Deepfakes: benchmarks and algorithms.