# A Machine Learning Approach To Detect DeepFake Videos

By

Md. Mahedi Hassan
17301098
Nafisha Nawrin
20241064

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
June 2021

# Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

**Student's Full Name & Signature:**

| | |
|---|---|
| Mahedi | Nafisha Nawrin |
| Md. Mahedi Hassan<br>17301098 | Nafisha Nawrin<br>20241064 |

# Approval

The thesis named “A Machine Learning Approach To Detect DeepFake Videos” submitted by

1. Md. Mahedi Hassan (17301098)
2. Nafisha Nawrin (20241064)

Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 02, 2021.

**Examining Committee:**

Supervisor:
(Member)

Md. Khalilur Rhaman

Dr. Md. Khalilur Rhaman
Associate Professor
Department of Computer Science and Engineering
BRAC University

Co Supervisor:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi
Chairperson and Associate Professor (CSE)
Department of Computer Science and Engineering
BRAC University

# Ethics Statement

In every level of research, this research paper is free of any plagiarism. Total security of the informants or any relevant people have been ensured.

# Abstract

DeepFake detection is important as the internet is a big part of our lives. DeepFake photos and videos can easily mislead us into thinking something that probably did not happen. It can also reduce trust in the media. As these manipulations become more convincing, celebrities are usually the victim of these kinds of misleading photos and videos. To detect fake videos, we will focus on existing methods and build our model to be more accurate as images of small imperceptible perturbations are sufficient to fool the most powerful neural network. In our Machine Learning approach, we first take the sample videos for training. Then, using open CV2, we have generated images from those videos. After that, we have passed these images to PCA for extracting principal component features. Then we applied VGG-16 and finally we have compared the train-test accuracy using different classifiers like SVC, RFC, GNB, CNN etc. After analyzing through our model we will be able to infer whether the input video is real or fake.

**Keywords:** Machine Learning; Neural Networks; DeepFake

# Dedication

This research is dedicated to the individuals who want to know if the videos they are seeing on the internet is real or fake. The authors, the teachers and the supporters all have been dedicated to improve the quality of this paper throughout the journey.

# Acknowledgement

We want to thank Omnipotent Allah, our honorable Supervisor Dr. Md. Khalilur Rahman, our Co-supervisor Dr. Md. Golam Rabiul Alam for their advice and feedback. Moreover, our parents, friends and faculty members supported us whenever we needed any help.

# Table of Contents

# List of Figures

# List of Tables

# Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

| | |
|---|---|
| $CNN$ | Convolutional Neural Network |
| $conv.$ | Convolutional |
| $FN$ | False Negative |
| $FP$ | False Positive |
| $GNB$ | Gaussian Naive Bayes |
| $ML$ | Machine Learning |
| $RF$ | Random Forest |
| $SVM$ | Support Vector Machine |
| $TN$ | True Negative |
| $TP$ | True Positive |

# Chapter 1

# Introduction

## 1.1 Overview

As the popularity of social media and various types of apps is growing, there is also an increasing number of manipulative visual contents on the internet. The new generation of deep neural networks is capable of incorporating videos from large volume of dataset which requires minimum manual editing. After the emergence of DeepFakes, face counterfeit techniques are reduced at a great level. DeepFake replaces or manipulates the face in a video with another person's face by using generative adversarial networks (GANs).[15] In a GAN model, there are millions of images to help to build a realistic facial structure. This realistic nature of the fake videos is highly targeted for fake news, pornographic material, malicious files and fake surveillance videos. The fake videos are also used to create political instability. A GPU and training datasets are only required to make these kinds of highly advanced face-swap videos. In this era of the information age, gaining public trust is really challenging. So, the importance of identifying DeepFake is beyond description. Through machine learning, we will use predictive models to classify Deep Fake and also predict the source that is made on those photos or videos. This is a very important topic since it will be helpful to avoid such occurrences if they are predicted and identified. It is high time we got to detect and combat DeepFake contents that can include fake images, videos, audios etc. Again, tracing the history of this content is also significant. In general, if a person is given access to a content, he/she should be able to track back to trace the history of that content.

## 1.2 Motivation

As we know early in the year of 2019, Facebook banned deepfakes that might mislead users into thinking a subject had said something they had not. Computer-generated photos of people's faces have become a common factor of sophisticated foreign interference campaigns, it feels more authentic in those fake photos and videos. DeepFake videos are growing on the internet everyday and a lot of people are being fooled by it. By performing extensive experiments to demonstrate the robustness of our model, we will be able to show that our method is able to detect high- and medium resolution deepfakes from various GANs with good accuracy. We plan to further enhance our face detection algorithms we will train our models with different levels of technologies which may potentially strengthen our result for the long run.

## 1.3 Problem Statement

Previously, people have been using manual methods like photoshop to modify images. DeepFakes were only accessible to big-budget movies, known as ‘computer-generated imagery’ (CGI). But over the decades, faster processors, good graphics and better algorithms have made it more available to general people. With Generative Adversarial Networks, this process is being automated and the results have become significantly better. Deepfake is more significant than other video manipulation techniques because it is photorealistic. Apps like FakeApp or OpenFaceSwap, which is built based on the DeepFake algorithm, by this millions of people who have limited knowledge of programming and machine learning, can create fake videos. These kinds of videos may harm the intellectual property of many internet users. To prevent this, we must track this sort of attack to keep the users safe. In this paper, we will be developing such a model that will detect and predict DeepFakes.

## 1.4 Research Objectives

Our research has two major objectives:
• To identify deepfake photos and videos from distinct sources: As there are millions of deepfake videos on the internet, we will try to predict which photos, videos are fake.
• To compare existing methods and see which one works : We will be able to find out the best method to find out the DeepFake videos

## 1.5 Thesis Structure

This thesis work is organized as following structure:
Chapter 1 shows the introduction which consists thesis overview, our motivation behind this work, problem statement, and research objectives
Chapter 2 presents the literature review which includes the background study we did for this paper
Chapter 3 presents the system work flow and our overall dataset analysis which includes collection of data, extracting their feature and processing them
Chapter 4 presents the implementation of algorithms and different classifiers
Chapter 5 shows the analysis of our result
Chapter 6 concludes the paper with summary and future possibilities

# Chapter 2

# Background and Literature Review

## 2.1 Literature Review

In the research paper [16], an investigation has been made about the prediction of DeepFakes in cyber-based on face-swap techniques. Open Source Intelligence has been used for getting the dataset. The tools for creating DeepFakes are more open to people now. In 2015, Google released its own AI tool known as TensorFlow which is used for machine learning and image processing. It compares image-to-image and predicts medical-related issues. But as a matter of fact, it has been used in a malicious way to create DeepFakes. It can affect negatively to the targeted people as they remain on the internet for millions to be accessed.

By the face-swapping technology, researchers created dataset which included half a million of edited images which showed that, the same machine learning algorithm can be both used for distinguishing face swaps as well as making the detection more difficult. There are also various techniques and models to identify it, but after training the models, it was found that SVM had the maximum accuracy in predicting these attacks. Another important discovery by this paper is the variable importance in predicting fake videos.
In the research paper [10], photo response non-uniformity (PRNU) has been applied to detect it to get access to the method's accuracy. It detects the noise pattern in the light-sensitive sensors which is used in image forensics. After that research, only a cut-off value of 0.05 result has come off a 3.8% false positive rate with 0% false negative rate which is a nearly effective method to detect DeepFakes. Gyfcat is another attempt to use artificial intelligence and facial recognition software to detect the inconsistent part of the rendered face area. Inconsistent head poses can also be helpful here to detect it [18]. SVM algorithm has been applied here to estimate and classify head poses and inconsistent poses. As it has made a tremendous change in social life, journalists are also concerned about it and it is time we combined technical knowledge as well as findings of the study by biological signals and Pixel level irregularities [24].
Paper [17] acknowledges the problems that arise while trying to detect DeepFakes. The authors have proposed a way to develop a deep learning pipeline to detect fake audio. By using signal distortion, the data augmentation techniques where explored.

The different levels of learning pipeline as such: high resource language training, low resource language training on augmented data and fine tune on un-augmented data. These different kinds of augmentation methods were used to transfer learning schemes. As a result, it was found that, it helped to enhance the performance of the acoustic model as well as provided improvement of ASR performance. The fully convolutional neural networks (CNN) and a convolution recurrent neural network was applied here. An approach to distinguish manipulated videos can be error level analysis (ELA) which is based on deep learning. The cross-entropy of the loss function in the last soft-max layer were calculated here by which we can detect different image compression ratios. The convolutional neural network (CNN) can extract fake features and can significantly improve the ELA method [26].

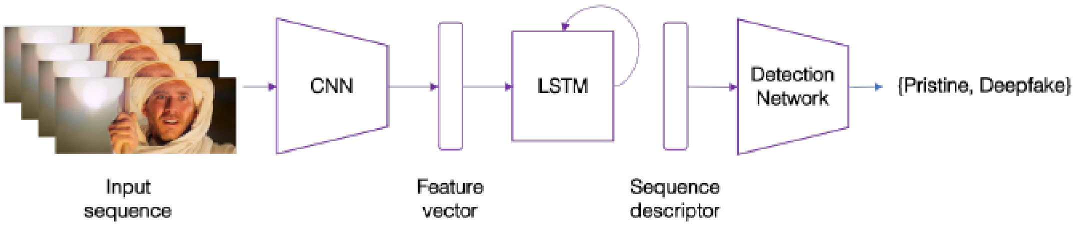


Figure 2.1: Detection System Using Convolutional LSTM Subnetwork

In this picture, Hochreiter and Schmidhuber [1] applied LSTM networks and it is then combined with a CNN, which led to massive improvements to this study. [9] However, in paper [8] the authors evaluated fast networks on an existing dataset, which demonstrated 98% accurate detection. They used their own dataset by downloading the videos available on the internet. To review the existing techniques, paper [25] reviewed four types of facial manipulation are reviewed. They provided details regarding manipulation techniques, key benchmarks and then a summary of results from those evaluations. They also discussed improvements in the latest DeepFake technologies. Attribute manipulation, entire face synthesis, identity swap and expression swap are the key types of manipulation techniques that are discussed.
About the machine learning approach on DeepFake detection creating DeepFake videos to autoencoding, using GAN, analyzing key indicators, model experiments with celeb-DF dataset and after comparing the AUC performance, the authors of paper [23] made a summary of 3 best methods to detect it.

### 2.1.1 Background

There are many features like surreal Background, Asymmetrical face, Non-stereotypical gender presentation, Semi-regular noise, Iridescent color bleed which can be implied to create these types of pictures.
DeepFake is basically based on Generative Adversarial Networks (GANs) where a generator creates fake samples - in this special cases, fake faces - and a discriminator tries to distinguish whether a given image is real or not. They train each other in a min-max fashion and become better until the generator converges to some hopefully good performance.

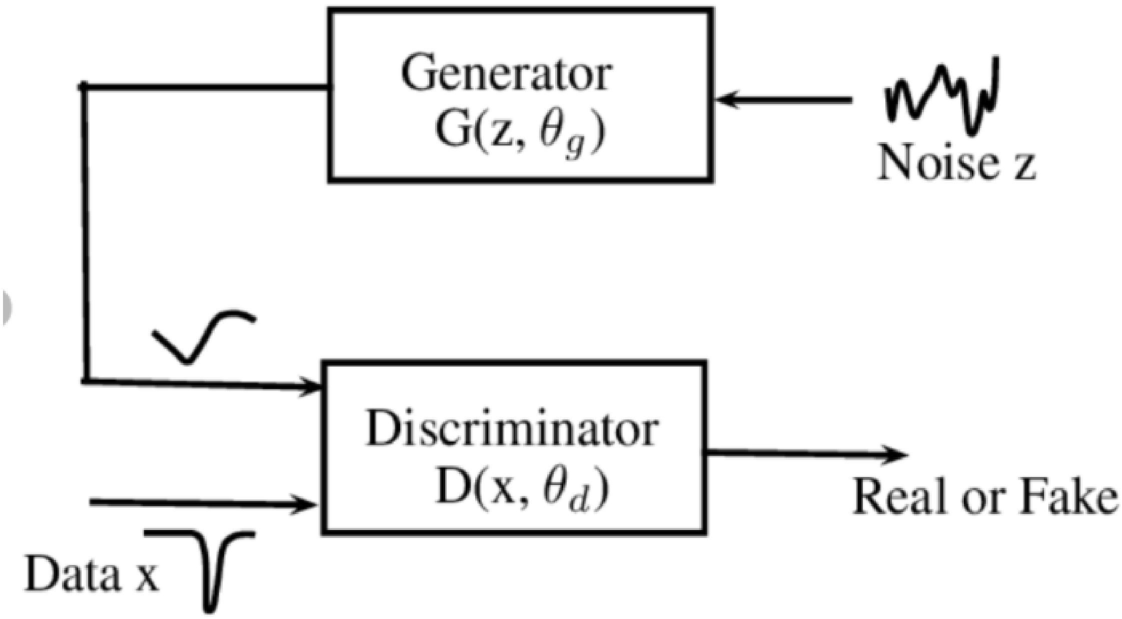


Figure 2.2: How GAN works

So the approach does basically already include a detection model, namely the discriminator that should be about as good at detecting fake images as the generator is in generating them. In practice, you could probably figure out detection models that might become better than the discriminator in a DeepFake's original GAN model or further improve it with more data but in the end, if there is a possibility to make such a detector better, it might be also applied to improve the generator so such a detection would most probably only work out for a limited time before it would need to be upgraded, similar as the security mechanics in bank notes have to be upgraded regularly as counterfeit becomes more and more advanced.
Two networks influence each other and iteratively update themselves in GAN. In the picture, the generator incrementally updates to improve itself, we can learn and visualize that. The generator does that to generate fake samples that are realistic. Moreover, the generator tries to trick the discriminator. The loss value of the generator decreases when the discriminator classifies fake and real samples. It visualizes gradients for the fake samples such that the generator would achieve its success
So, if the algorithm has the ability to evade an AI-based detection method, then this will improve detection systems in a significant amount. Facebook announced DeepFake Detection challenge on Kaggle in September 2019, by inviting researchers to improve or invent models [14]. 3,500 actors were hired to record videos for this purpose, which were then manipulated using various deepfake creation techniques. After about 35,000+ detection algorithm submission, it was found that 35% of deefakes identified were false positives. It is still hard to detect such videos with more accuracy. So we can conclude that, to overcome the flaws, it requires AI expertise and expensive GPU resources.[13] The following figure shows how GAN works.
Basically, GAN consists of two networks called Generator and Discriminator. The role of the generator is to generate signals or images and try to fool the discrimi-

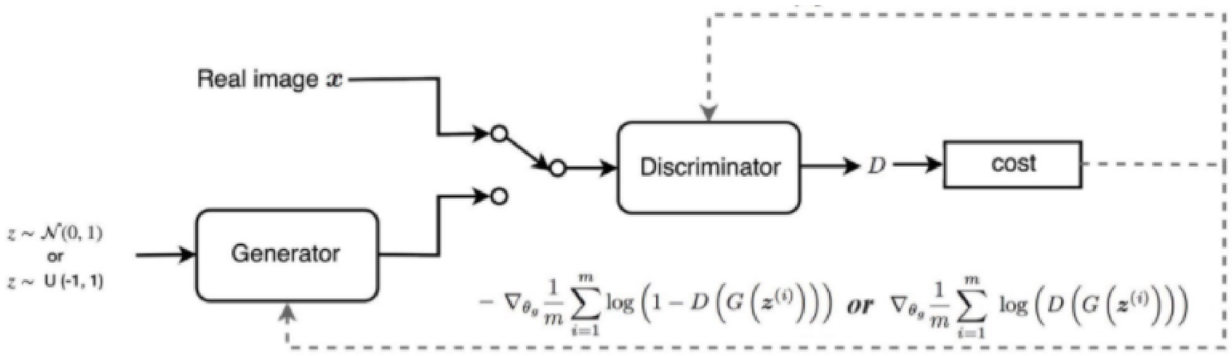

Real image $x$
$z \sim \mathcal{N}(0,1)$
or
$z \sim$ U (-1, 1)
Generator
Discriminator
$D$
cost
$-\nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^{m}\log\left(1-D\left(G\left(z^{(i)}\right)\right)\right)$ or $\nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^{m}\log\left(D\left(G\left(z^{(i)}\right)\right)\right)$

Figure 2.3: How GAN functions

nator. Through the training generators learn to generate fake signals tricking the discriminator. Discriminator tries to detect if the signal is fake or real. If the generator succeeds in fooling the discriminator it achieves the goal of generating deepFake. These deepFake images and videos become a great threat for society. This deepFake can manipulate people and create fake digital evidence which is the violation of laws. Using deepFake many fraudulent activity could occur like generating non-consensual fake pornography, fake audios may for money extortion and lot more fraudsters may target politicians and celebraties manipulating with such fake images, audio , videos. To detect this deepFake here in this paper we proposed a machine learning based model.

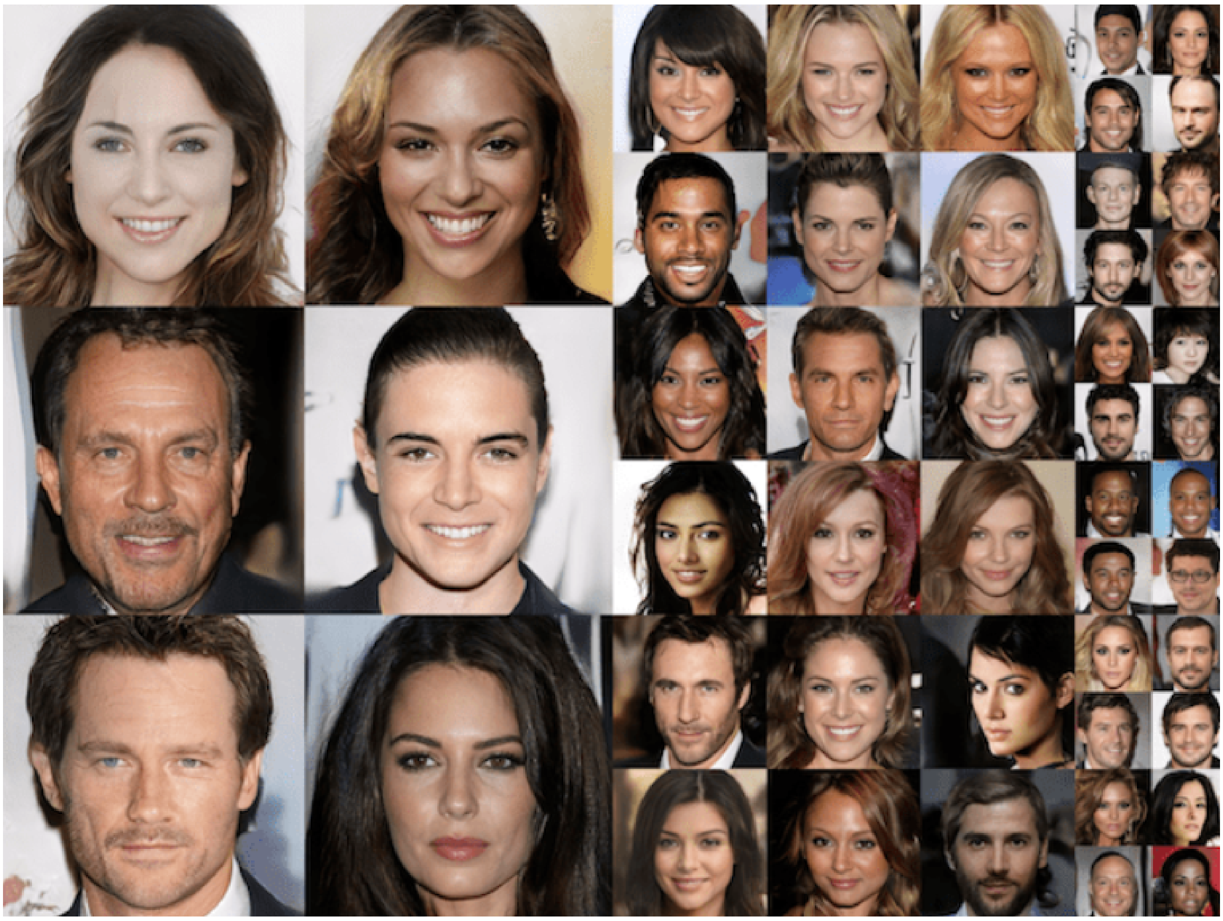


Figure 2.4: Photorealistic GAN-Generated Faces
[12]

# Chapter 3

# Dataset and Workflow Analysis

## 3.1 Data Collection

There are many datasets available on the internet for this purpose as the deepfake videos are constanty rising. Here we have taken our dataset from Kaggle deepfake challenge[45]. Deepfake strategies, which present practical AI-produced recordings of individuals doing and expressing anecdotal things, can essentially affect how individuals decide the authenticity of data introduced on the web. This substance age and adjustment advances may influence the nature of public talk and the protection of common liberties—particularly given that deepfakes might be utilized malevolently as a wellspring of deception, control, provocation, and influence. Distinguishing controlled media is an actual requesting and quickly advancing test that requires coordinated efforts across the whole tech industry and past. Here, the ultimate objective of the challenge is to give space to the researcher to come up with innovative approaches to deal with this deepfake threat of this generation and contribute to society by detecting fraudulent activity.

| Train Videos | Fake Videos | Real Videos |
|---|---|---|
| 400 | 323 | 77 |

Table 3.1: Training Fake and Real Videos

In our data set, we have found bias. Though both train and test data contain 400 videos each, in the train data set we have found fake and real labeled videos ratio 80.75: 19.25. Hence, we have found clear biases in the data set. So it may cause overfitting the dataset. We have to deal with this biased data and with our machine learning approach, we will try to maximize our accuracy. At first, we have to take each video in loop and capture pictures from those videos in 128*128 resolution. Keeping those pictures under the dataset folder in two separate folders named fake and real based on their label. During train and test, we have split our deepfake data 80% for training and 20% for testing. We will be using this for our further processes.

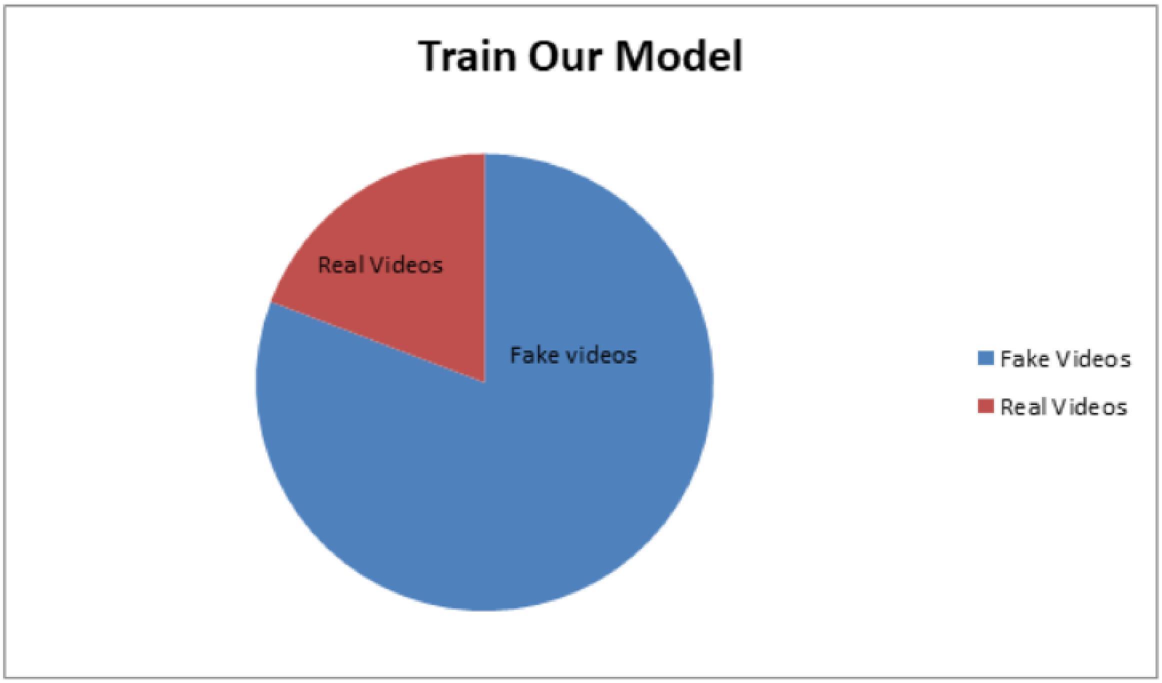


Figure 3.1: Training Out Model

## 3.2 Workflow

The general pattern of our paper:

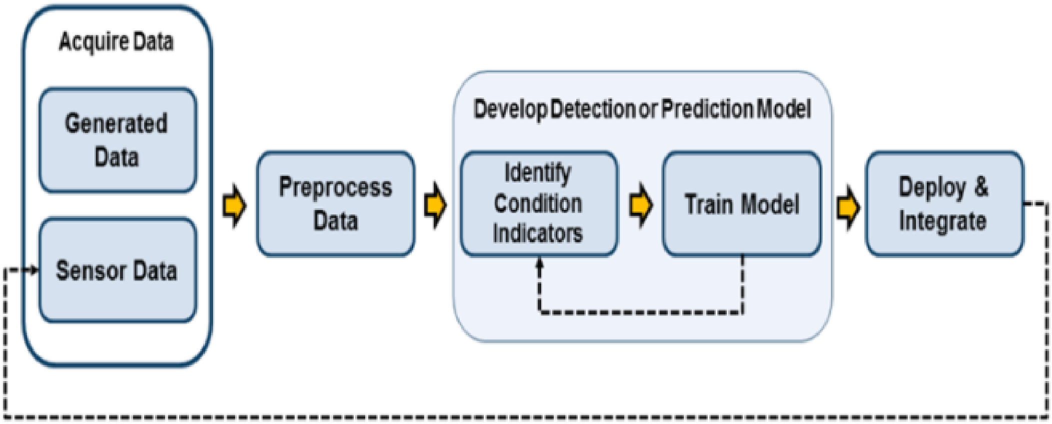


Figure 3.2: General Workflow Diagram

The dataset will be taken from available internet videos. As there are huge amount of fake videos, so collecting the data will not be hard. Many competitions are held to detect such videos to make people concerned about this issue and overcoming them.
As the gathered data will contain many irrelevant data, Data pre-processing will be required.
Following the 80/20 rule for machine learning we will spend most of the time of our study in this step.
Data pre-processing means cleaning raw data into clean data, i.e. keeping the data

that is most relevant for training. Basically, we will repopulate missing data, remove noisy data and remove data duplication.
After that, we will start Training Dataset and Testing it simultaneously.
We will evaluate the results and begin polishing our Algorithm meanwhile.
We will have to pick the best algorithm so that it performs properly with our chosen dataset.
We have distinguishable categories for our dataset. Training Data for training our algorithm, Testing Data for checking if the algorithm is providing the expected results.
Validation set will be used to predict the ability of the model to function on unseen data. During evaluation phase we will be able to pin point the perfect state for our model.
Finally, when our model is ready, we will provide it with data and it should be able to predict DeepFake photos and videos.

Here is the workflow diagram of our paper:

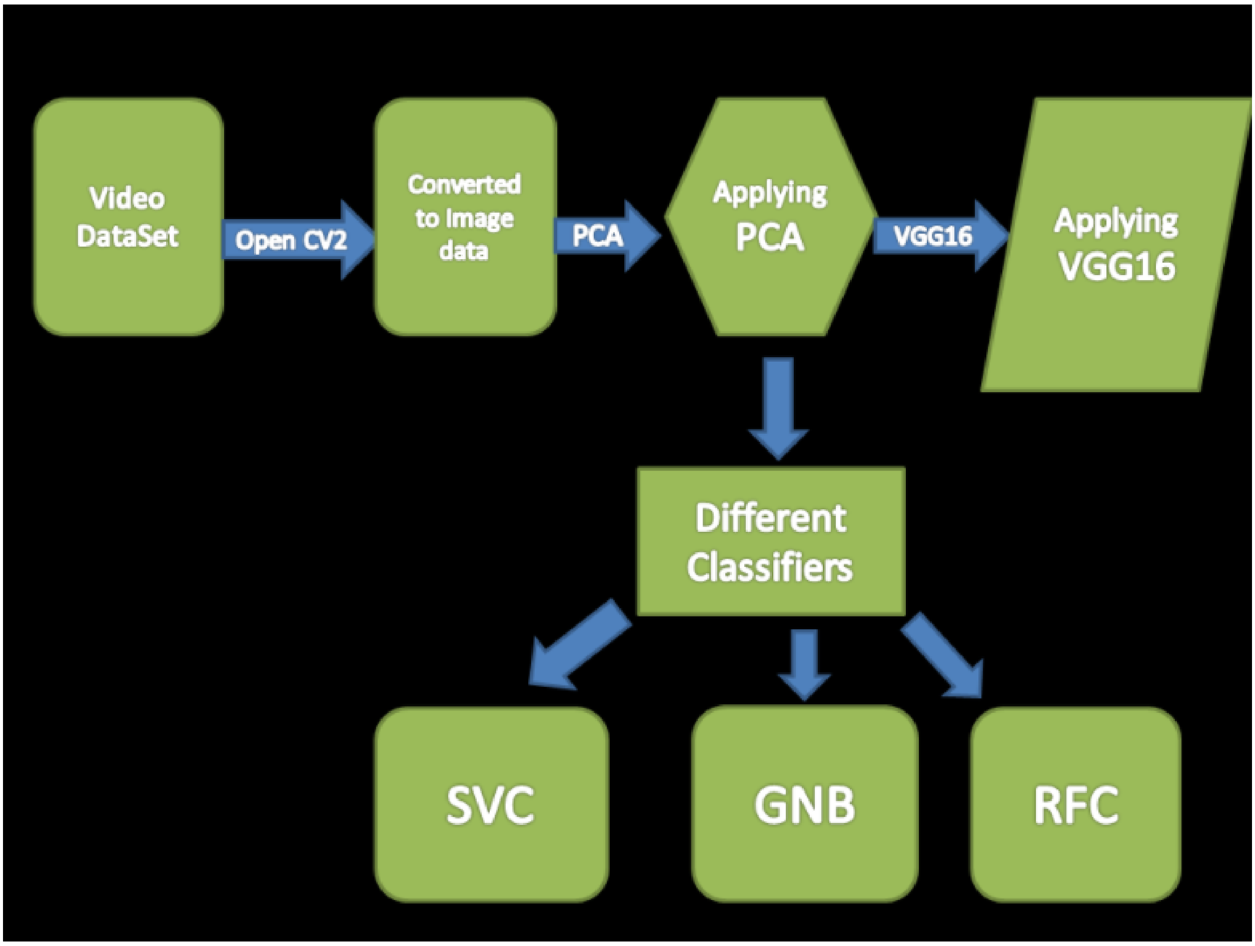


Figure 3.3: Working methodology of our machine learning based model

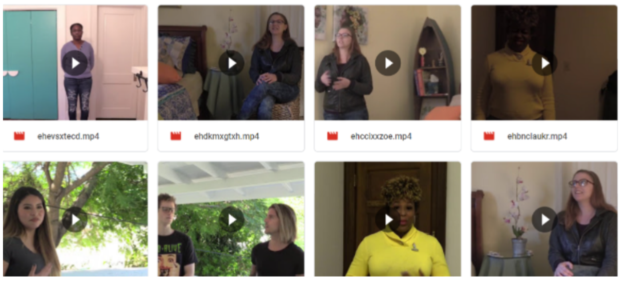


Figure 3.4: Kaggle Video Dataset

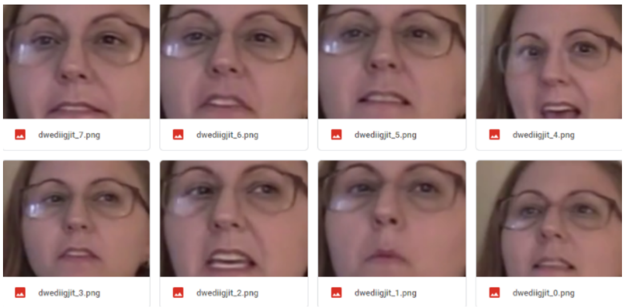


Figure 3.5: Extracting Images from Videos

### 3.2.1 Extracting images from video data set

OpenCV accompanies numerous amazing video-altering capacities. In the current situation; procedures, for example, picture examining, face acknowledgment can be refined utilizing OpenCV. Picture Analysis is an exceptionally normal field in the space of Computer Vision. It is the extraction of significant data from recordings or pictures. OpenCV library can be utilized to play out various procedures on recordings. Using openCV2 we have extracted images from the given training video dataset and then we have kept it to two separate folders named fake and real based on their label.

### 3.2.2 Applying PCA to extract features

Principal Component Analysis (PCA) is a technique that is generally used for the reduction of dimensionality in machine learning; this is also an unsupervised, non-parametric statistical technique. Here, Higher dimensionality refers to a large number of features of the dataset. For filtering noisy datasets, PCA can also be used; for example image compression.[4] Since we are reducing the number of variables there is a high chance that we might lose some accuracy, but there is a little accuracy for the sake of trade in the diminishing of dimensionality to make it more simple. Here, the reason is small-sized datasets can be explored and visualized very easily and analyzed data faster and easier for algorithms like machine learning and no need to process irrelevant variables.[3] First of all, we have to focus on standardization. For ensuring the equal contribution of each variable in the analysis, the successive primary variables range need to be standardized. Being more specific, performing prior standardization to PCA is difficult because the sequent is more sensitive than the primary variable variances. In case of the huge differences between the scopes of primary variables, larger range variables will dominate the smaller range variables like a variable that has a range between 0 and 1 will be dominated by the variable that has a range between 0 and 100, that will give us biased results.[5] Hence, this problem will be solved by the transformation of data to matchable scales. Secondly, we have to focus on covariance matrix computation. Here, we have to understand the variation of the input data set of the variables among each other; otherwise, we have to find out the relationship among those input data. And the reason is that variables sometimes contain unnecessary information because of their correlation. Hence, we are computing the covariance matrix for identifying the correlations.[2] Thirdly, we have to focus on computing the eigenvectors and eigenvalues. These are the terms from linear algebra and we have to compute them by using a covariance matrix for determining the main components of given data. So, for this, first of all, we have to understand what the principal component means. Now, in this step, we have to compute the eigenvectors and have to order them by descending order and this will help us to find out the significant principal components. So, here, we choose and keep the most significant components with higher eigenvalues or discard the less significant components with lower eigenvalues. Then by the remaining components, we form a vector matrix called Feature vector. In our former steps, we just find out the principal components and using the remaining components to form a feature vector but do not change any data without standardizing them but we will always get the input data set according to their original axes or primary variables.[41] Lastly, In our final step, we have to focus on the usage of the feature vector that we formed from the eigenvectors of the covariance matrix. Here, we will orientate the data again to the axes represented using principal components from the original axes. We will multiply the transpose of main datasets with the transpose of the feature vector and then this reorientation will be done by this process.[36]

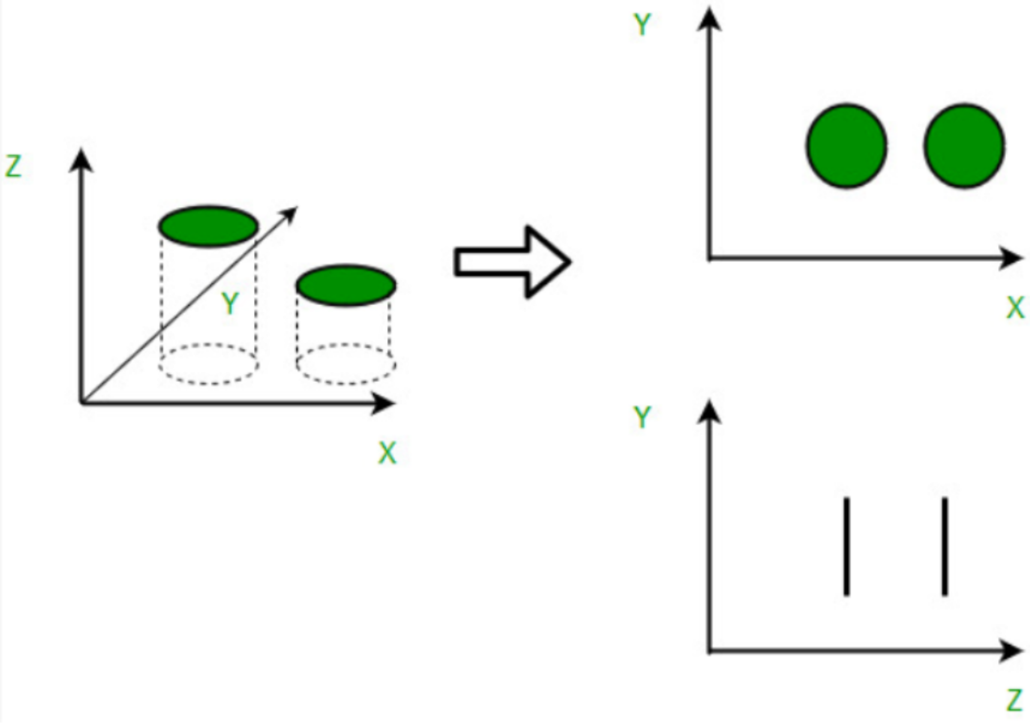


Figure 3.6: Dimension Reduction Using PCA

PCA gathers data from an enormous arrangement of factors into fewer factors by applying a type of change onto them. The change is applied so that straightly connected factors get changed into uncorrelated factors.[42] Relationships disclose to us that there is a repetition of data and assuming this excess can be diminished, data can be packed. For instance, if there are two factors in the variable set which are profoundly corresponded, at that point, we are not acquiring any additional data by holding both the factors since one can be almost communicated as the direct blend of the other. In such cases, PCA moves the change of the second factor onto the principal variable by interpretation and pivot of unique tomahawks and projecting information onto new tomahawks. The heading of projection is resolved to utilize eigenvalues and eigenvectors.[32] Thus, the initial not many changed highlights (named as Principal Components) are wealthy in data, while the last highlights contain for the most part commotion with insignificant data in them. This adaptability permits us to hold the initial not many head segments, consequently lessening the number of factors essentially with insignificant loss of data. Here, we have applied PCA to extract principal components from our image data.[39] Since Cov(x,y) is equivalent to Cov(y,x), the network is, as said, symmetric, and the fluctuations of the highlights lie on the central corner to corner.

$$\sum \begin{matrix} \mathrm{Var(x)} & \mathrm{Cov}(x,y) \\ \mathrm{Cov}(y,x) & \mathrm{Var}(y) \end{matrix}$$

The covariance network can expect various qualities relying upon the state of our information. At the point when the two highlights are decidedly connected, the covariance is more noteworthy than nothing, else, it has negative worth.[40] Besides, if there is no proof of a connection between them, subsequently the covariance is equivalent to zero. Some of them (all the more explicitly, as numerous as the number of highlights), however, have an exceptionally intriguing property: undoubtedly, when applied the change T, they shift length yet not the course. Those vectors are called eigenvectors, and the scalar which addresses the numerous of the eigenvector is called an eigenvalue. [37]The two sections of this new, changed space Y are the Principal Components we will use instead of our unique factors. Those, as referenced above, are developed in a way to such an extent that they store however much data as could be expected. PCA is broadly utilized in Machine Learning tasks: to be sure, the preparation method must be just about as quick as conceivable all together for our calculation to be effective,[29] however, it doesn't mean we can diminish the dimensionality without a particular measure with the danger of losing applicable data. This will allow us to work faster as now instead of computing numerous components we have to deal with only some principal components. Here we have taken 100 principal components.

### 3.2.3 Applying VGG-16 algorithm and several classifiers

After successfully extracting features, we have used VGG-16 algorithm and four classifiers on our data. Those will be discussed on the next section of our paper.

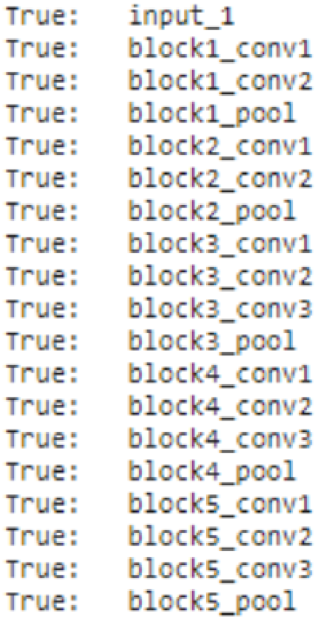
# Chapter 4

# Methodology

Machine learning approach is a data examination strategy that trains our system to do what effectively becomes alright for individuals and animals: acquire for a reality. Artificial intelligence estimations use computational procedures to "learn" information directly from data without relying upon a predestined condition as a model. The computations adaptively improve their show as the amount of tests open for learning augmentations. Deep learning is a specific kind of ML. Machine learning (ML) is a combination of different types of algorithms and it is a classification of artificial intelligence (AI). It allows us to generate more perfect results than our predicted outcomes and no need to do programming additionally for this. For predicting new values of output ML algorithms take input from historical data. It is very important in the business and development sector because it gives a trending view of behaviour of customers and provides operational business patterns. Development of new products also supported by ML. Nowadays ML has become the central part of many world's prominent companies like GOOGLE,UBER,FACEBOOK. Machine learning does the job of competitive differentiator for various companies in a significant manner.It has four basic approaches and they are unsupervised learning , semi-supervised learning, supervised learning and reinforcement learning. Based on data prediction, data scientists choose algorithms. In supervised machine learning, data scientists supplied algorithms along with training data that are labeled and defined that variable algorithms need to access for their correlations. They specified the input and output both for the algorithm. In unsupervised Machine learning, unlabeled data training algorithms are involved. They seek for suggestive connection scanning algorithms by data sets. Here, everything is predetermined like the algorithm's training data and output that was predicted and recommended. Semi-supervised ML is a mixture of two different predicted types. This approach is free for exploring data by itself and developed the understating for the dataset but here data scientists mainly use the algorithms that trained labeled data. When data scientists have specific rules for a multi-step process, then they will use reinforcement ML for teaching the machine how to complete this process. Here, data scientists gave negative and positive hints and for completing those tasks they programmed an algorithm. In most of the cases, the following steps are chosen by the algorithm itself.

## 4.1 VGG-16 Algorithm

VGG models are a kind of CNN Architecture proposed by Karen Simonyan and Andrew Zisserman of Visual Geometry Group (VGG), Oxford University, which brought surprising outcomes for the ImageNet Challenge. They explore different avenues regarding 6 models, with various quantities of teachable layers. In view of the quantity of models the two most famous models are VGG16 and VGG19. It is a CNN model which is an excellent vision model architecture.[31] It is a pyramid shaped model. The number is 16 because it has 16 weight layers and the top layers are deeper. We can say it is a big network as it has approximately 138 million parameters. It enhances classification accuracy as it increases CNN depth. To down sample the image, 5 max pooling filters are added here. There are three fully connected layers. The last layer is named soft-max layer.[30]



```
True:   input_1
True:   block1_conv1
True:   block1_conv2
True:   block1_pool
True:   block2_conv1
True:   block2_conv2
True:   block2_pool
True:   block3_conv1
True:   block3_conv2
True:   block3_conv3
True:   block3_pool
True:   block4_conv1
True:   block4_conv2
True:   block4_conv3
True:   block4_pool
True:   block5_conv1
True:   block5_conv2
True:   block5_conv3
True:   block5_pool
```

Figure 4.1: VGG-16 model

The contribution to the conv1 layer is of fixed size 224 x 224 RGB picture. The picture is gone through a pile of convolutional (conv.) layers, where the channels were utilized with a little open field: 3×3 (which is the littlest size to catch the idea of left/right, up/down, focus). In one of the designs, it likewise uses 1×1 convolution channels, which can be viewed as a straight change of the information channels (trailed by non-linearity). The convolution step is fixed to 1 pixel; the spatial cushioning of conv. layer input is with the end goal that the spatial goal is protected after convolution, for example the cushioning is 1-pixel for 3×3 conv. layers. Spatial pooling is completed by five max-pooling layers, which follow a portion of the conv. layers (not all the conv. layers are trailed by max-pooling). Max-pooling is performed over a 2×2 pixel window, with step 2. Three Fully-Connected (FC) layers follow a heap of convolutional layers (which has an alternate profundity in various designs): the initial two have 4096 channels each, the third performs 1000-way ILSVRC order and hence contains 1000 channels (one for each class). The last layer is the delicate max layer. The arrangement of the completely associated

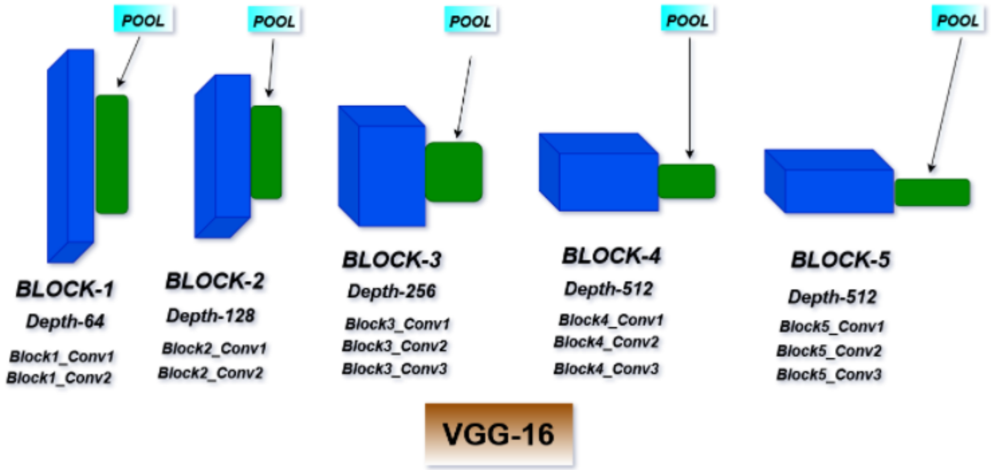


Figure 4.2: VGG-16 Architecture of Our Model

layers is something very similar in all networks.All covered up layers are furnished with the amendment (ReLU) non-linearity. It is additionally noticed that none of the organizations (aside from one) contain Local Response Normalization (LRN), such standardization doesn't improve the presentation on the ILSVRC dataset, yet prompts expanded memory utilization and calculation time. Here, we have used VGG16 that helps us to analyze our image data and push us to have a good classification.

## 4.2 Classifiers

A classifier in ML, AI is a calculation that naturally arranges or sorts information into one of a bunch of classes. It is a cycle of ordering a given arrangement of information into classes, It can be performed on both organized or unstructured information. [35]The cycle begins with anticipating the class of given information. The classes are frequently alluded to as target, mark or classifications. Here, we have used different classifiers. We have use SVC, GNB, RFC and CNN for the classification. Using these classifiers we get training and testing accuracy prediction which we will discuss in our result and analysis chapter.

### 4.2.1 Convolutional Neural Network Classifier

#### Convolutional Neural Network Classifier Concept

Neural Networks have been seeing a stupendous development in overcoming any issues between the abilities of people and machines. Specialists work on various parts of the field to get astonishing things going. One of numerous such regions is the area of Computer Vision. The plan for this field is to empower machines to see the world as people do, see it likewise and even utilize the information for a large number of Image and Video to understand and for the acknowledgment of mage Analysis and Classification, Media Recreation, Recommendation Systems, Natural Language Processing, and so on The headways in Computer Vision with Deep Learning has been built and idealized with time. The pre-preparing needed in a CNN is a lot lower when contrasted with other order calculations. Solitary neurons respond to redesigns simply in a limited region of the visual field known as the Receptive Field. The whole visual region is covered by such a field. The part of the CNN is to diminish the pictures into a structure which is simpler to measure, without losing highlights which are basic for getting a decent forecast. Image classifier CNN can be used in a lot of ways. It is mainly used to analyze visual images.

Firstly, convolution is a merge of multiple functions.[11] It is a fully connected neural network. The dimensionality is reduced in a CNN model by a sliding window which has a small size than the input matrix. A convolutional layer has Convolutional kernels which are defined by hyper-parameters. One layer's input channels have to be equal to the number of output channels. Then these layers pass their result to the next layer. The response of a human neuron is similar to this.[19][27] CNN may include pooling layers and they can reduce the dimensions of data by combining the outputs of neurons. After applying a specific function, each neuron computes an output value received from the receptive field in the previous layer.

CNN is susceptible to overfitting data. The usual ways of preventing this or ways of regularization are During training, penalizing parameters or trimming the connectivity. CNN takes advantage of the hierarchical pattern that appears and create patterns of complex nature, utilizing simpler and similar patterns that are embossed in the filters.

Individual cortical neurons have a response to a stimulus that can only be found in the region of the visual field, known as the receptive field. These tend to overlap as they take cover of the whole visual fields. CNNS utilize very little pre-processing when compared with other image classification algorithms. The network utilizes automated learning to optimize filters where traditional algorithm filters are hand-engineer. This independence is a major advantage for CNN.

#### Result After Applying Convolutional Neural Network Classifier

We applied CNN classifier at first and then We calculated the following accuracy from this. The following figure shows the accuracy we found from applying CNN classifier. :

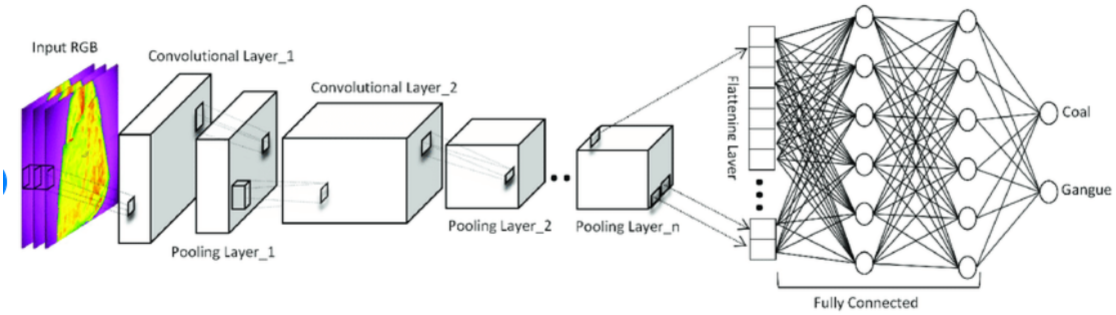


Figure 4.3: Construction of CNN Model

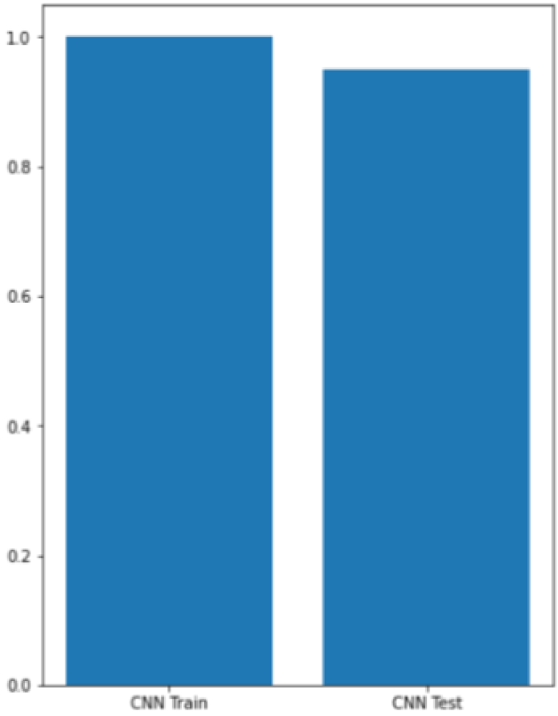


Figure 4.4: Accuracy of CNN classifier Train and Test Data

### 4.2.2 Gaussian Naïve Bayes Classifier

**Gaussian Naïve Bayes Classifier Concept**

To reduce the complexity of the Bayesian classifier, Naive Bayes Algorithm emerged. It does not guess any conditional independence on the training dataset. [20]So, the complexity gets reduced to only 2n. [22]Bayes Theorem acts as the basis for Naïve Bayes classifiers. A crucial assumption is between the features, there is strong independence. They assume the value of a feature that is independent of the value of any other. These Naïve Bayes classifiers require to be trained very efficiently. These classifiers require a small training data for classification to estimate the parameters.[21][34] Because of its simple design, it can be useful and implemented in many real life situations. A Gaussian distribution with no co-variance between dimensions is the approached needed to create a simple model.[7] [6]By finding the standard deviation and mean within each label, this is what it is required to define a distribution.

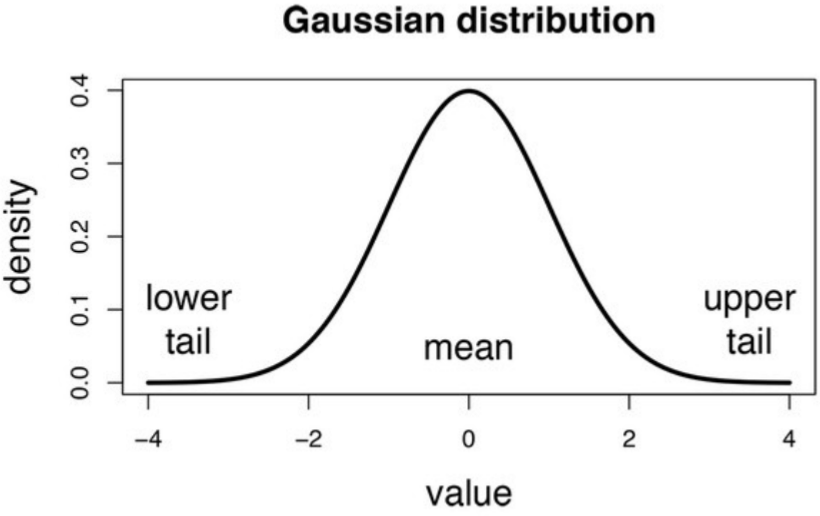


Figure 4.5: Gaussian Naive Bayes Distribution

Gaussian Naive Bayes algorithm is a variant of Naïve Bayes and follows a normal distribution. It follows a supervised machine learning classification. Even though it is simple, it has high functionality. The formula for the Bayes theorem is as followed:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Here,
P(A)= prior probability,
P(AjB) = posterior probability; (A)is true given some evidence (B)
P(B) = probability of the evidence
P(BjA) = probability of the evidence when hypothesis is true
In short, we can say that Bayes theorem:

$$\text{posterior } = \frac{\text{prior } \times \text{ likelihood}}{\text{evidence}}$$

For Gaussian Naïve Bayes, the likelihood of features is as followed:

$$P\left(x_i \mid y\right)=\frac{1}{\sqrt{2 \pi \sigma_y^2}} \exp \left(-\frac{\left(x_i-\mu_y\right)^2}{2 \sigma_y^2}\right)$$

Here,
$\mu_{i,j,}$ = mean
$\sigma_{i\ \mathrm{iv}}$ = standard deviation
So we can say that his model can be applied by finding the mean and standard deviation of the points within each label.
Mean:

$$\mu=\frac{\sum x}{n}$$

Standand Deviation:

$$\sigma=\sqrt{\frac{\sum(x-\mu)^2}{n}}$$

here

$$n= \text{ number of scores in sample.}$$

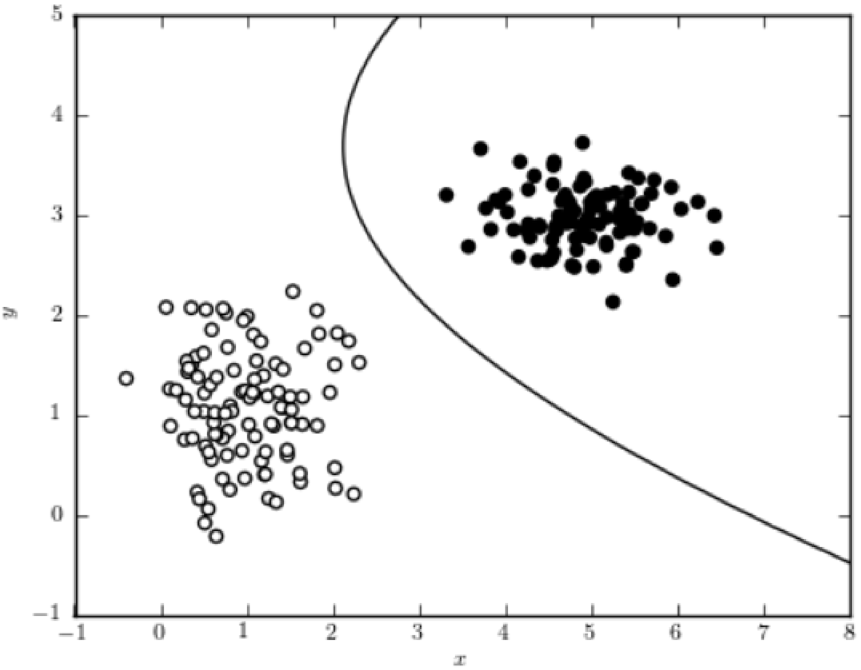


Figure 4.6: Simple Gaussian Decision Boundary

### Result After Applying Gaussian Naïve Bayes Classifier

We calculated the following accuracy from this:

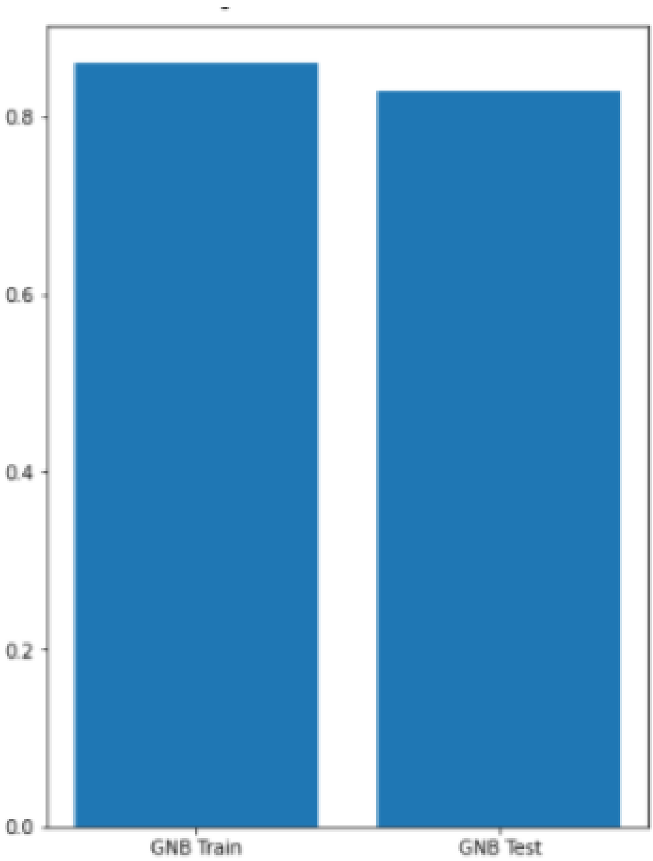


Figure 4.7: Accuracy of Gaussian Naïve Bayes Classifier Train and Test Data

### 4.2.3 Random Forest Classifier

#### Random Forest Classifier Concept

It can be used for classification and regression problems. Even without hyper-parameter tuning this classifier gives a good result.[36] It is also a supervised learning algorithm. To give a more accurate prediction, it builds multiple decision trees and ensembles them. During the tree construction phase, the training samples are chosen randomly with replacement. [38]. It is trained with the help of a bagging method and the forest is concurred with an arrangement of decision trees. Bagging method creates significant effects on overall result. In ML, it is an absolute privilege that we can use random forest in both the case of regression and classification. Similar to bagging classifiers and decision trees, random forest has almost similar parameters. The significance of Classification is that it is the base block of ML. By using random forest classifiers we got the privilege that further we don't need to merge bagging classifiers and decision trees, as we can use random forest classifier class directly. With the help of a regression algorithm, we can achieve the regression task in random forest.

#### Result After Applying Random Forest Classifier

We calculated the following accuracy from this:

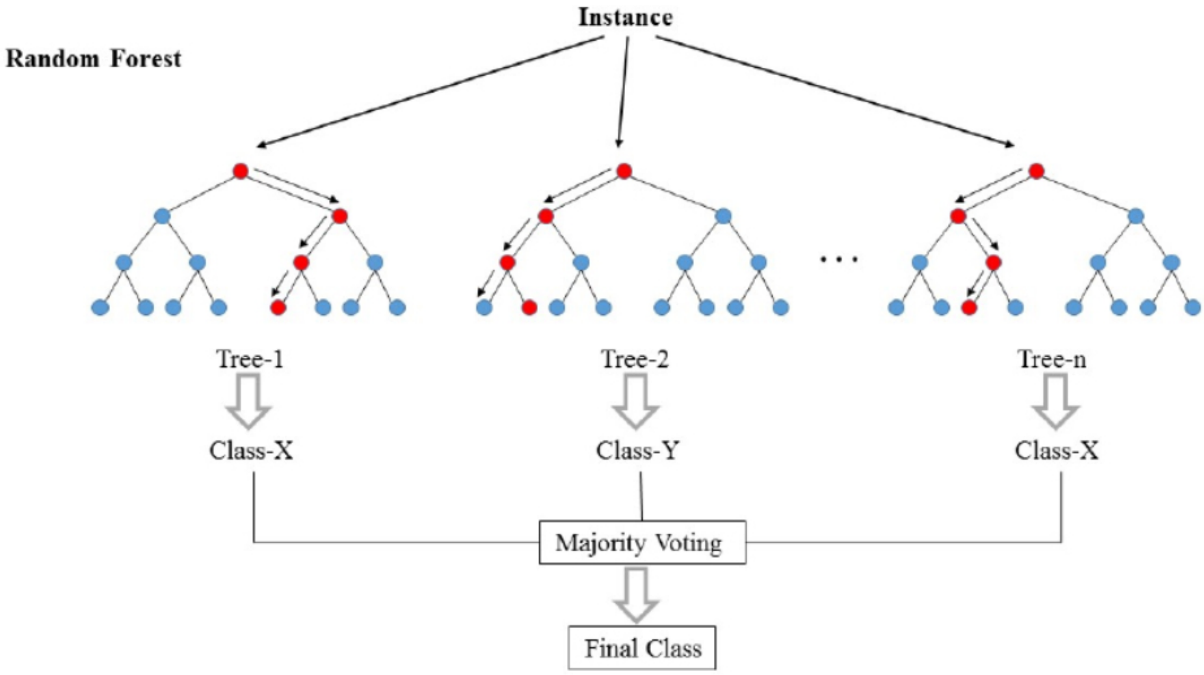


Figure 4.8: Random Forest Classifier

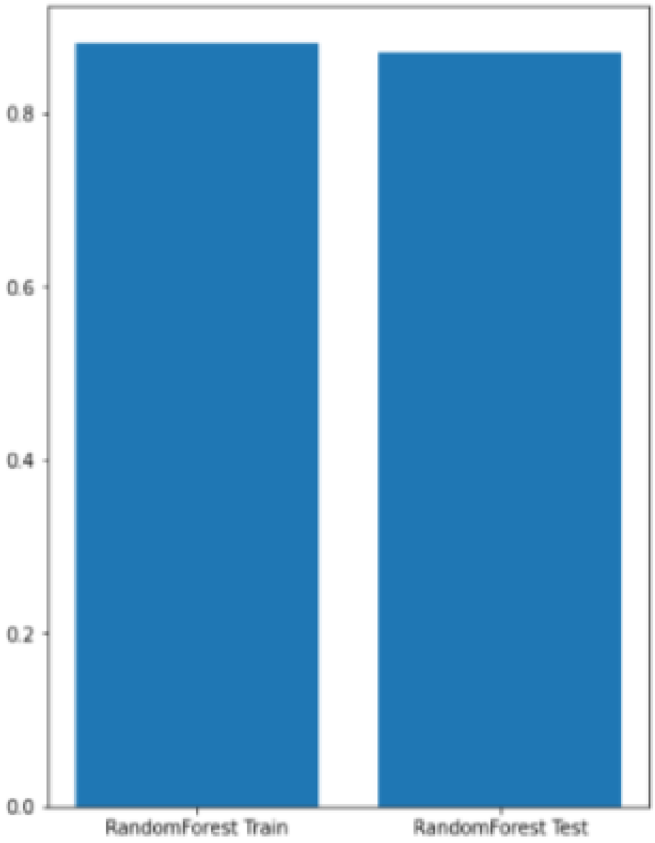


Figure 4.9: Accuracy of Random Forest classifier Train and Test Data

### 4.2.4 SVM Classifier

#### SVM Classifier Concept

It is a supervised machine learning approach to be used in classification. It was introduced in 1960. As they can handle multiple continuous variables, they got really popular in recent times. In a multidimensional space, there are different classes in a hyperplane.[28][34] It gives us maximum marginal hyperplane(MMH). After iteratively dividing the classes it generates the hyperplane in the best way. Here in SVM, support vectors are the data points that are closest to the hyperplane. A hyperplane is a decision space between a set of objects consisting of different classes. A margin is a gap between two lines that are situated on the closest data points. It reduces misclassification errors.[33]

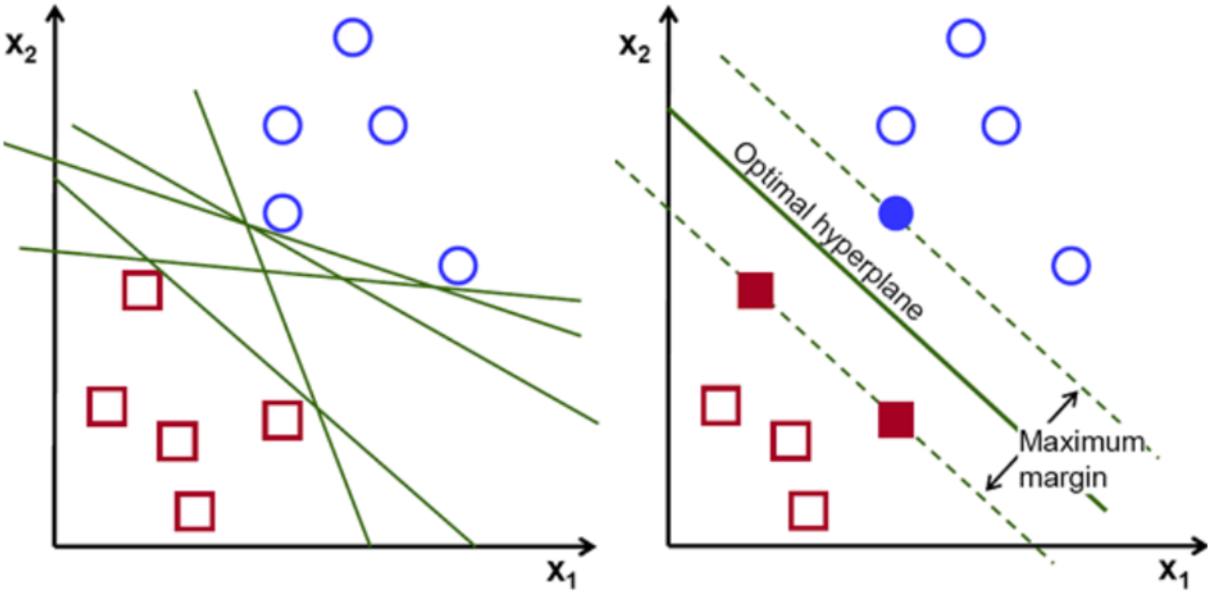


Figure 4.10: Hyperplane of Support Vector Machine

Positives of SVM classifiers are:
It gives great accuracy and can be worked in high dimensional space. A subset of training points is being used in SVM classifiers. And so, it needs very less memory.
Negatives of SVM classifiers are:
SVM classifiers take a lot of time for training. So, for a huge dataset, it is not feasible to use SVM classifier. Moreover, for overlapping classes SVM classifiers are not suitable.

#### SVC Concept

It is a clustering algorithm and it is not used to give assumptions for the shape of clusters. As it works best for low-dimensional data, we preprocessed out data using principal component analysis and then applied SVC. It organizes the data in a meaningful pattern. It belongs to kernel-based learning.
If A is an adjacency matrix between pairs of points then a given pair of points xi and xj the i,j element of A is given by

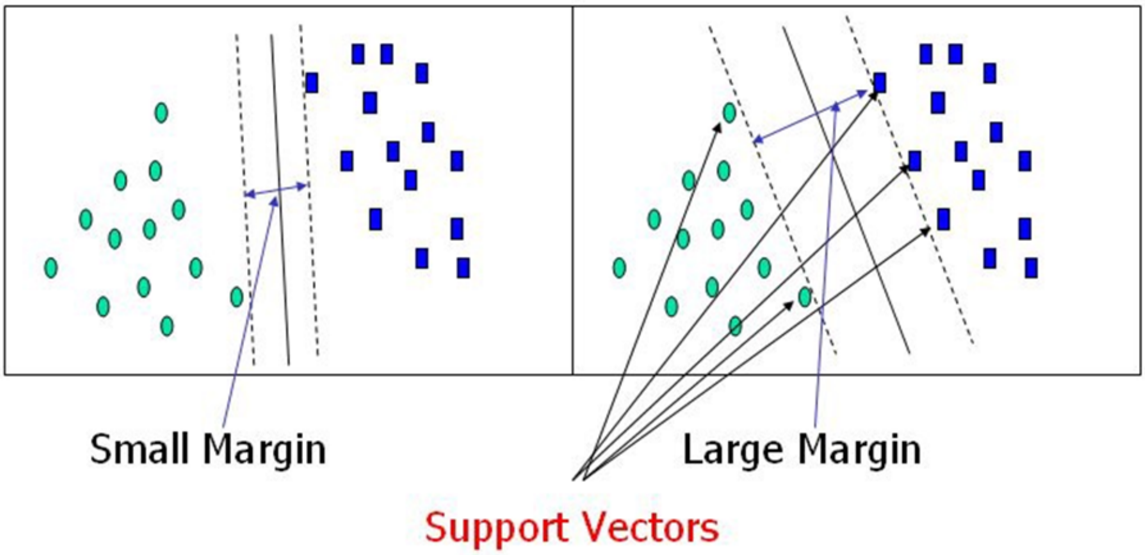


Figure 4.11: Selecting Maximum Marginal Hyperplane in SVM

$$A_{ij} = \begin{cases} 1, & \text{if } f(\mathbf{x}) > 0 \text{ for all } \mathbf{x} \text{ on the line segment connecting } \mathbf{x_i} \text{ and } \mathbf{x_j} \mid \\ 0 & \text{otherwise.} \end{cases}$$

**Result of our SVC**

we calculated the following accuracy from this

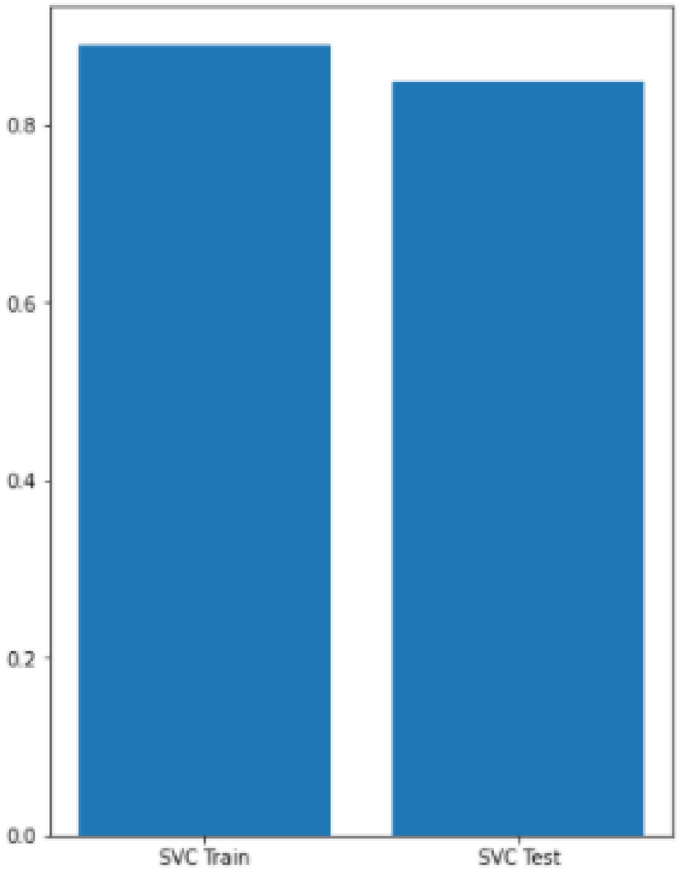


Figure 4.12: Accuracy of SVC classifier Train and Test Data

# Chapter 5

# Result Analysis

## 5.1 Accuracy Calculation

After building the model, performance was assessed to decide how accurately the model would have the option to anticipate weakness to enslavement. The actions that we have utilized for assessing execution depended on four boundaries of the disarray lattice. The boundaries are TP, TN, FP, and FN where True certain and Genuine negative demonstrated the number of perceptions was anticipated accurately.[43][44] Precision decides the proportion of the accurately anticipated examples to the all-out examples. The right expectation pace of the classifier is acquired from the following condition:

$$\text{Accuracy } = \frac{TP + TN}{TP + FP + TN + FP}$$

Sensitivity indicates how precisely the model could foresee results of tests contrasting with every one of the genuine results in the test set. In our trial, when the model arranged a perception as 'dependent' addressing the individual was powerless to fixation; affect ability perceived the example of expecting the right banner. The condition to process sensitivity:

$$\text{Sensitivity } = \frac{TP}{TP + FN}$$

Moreover, Precision shows the proportion of accurately anticipated positive perceptions to add up to positive perceptions of the test set.

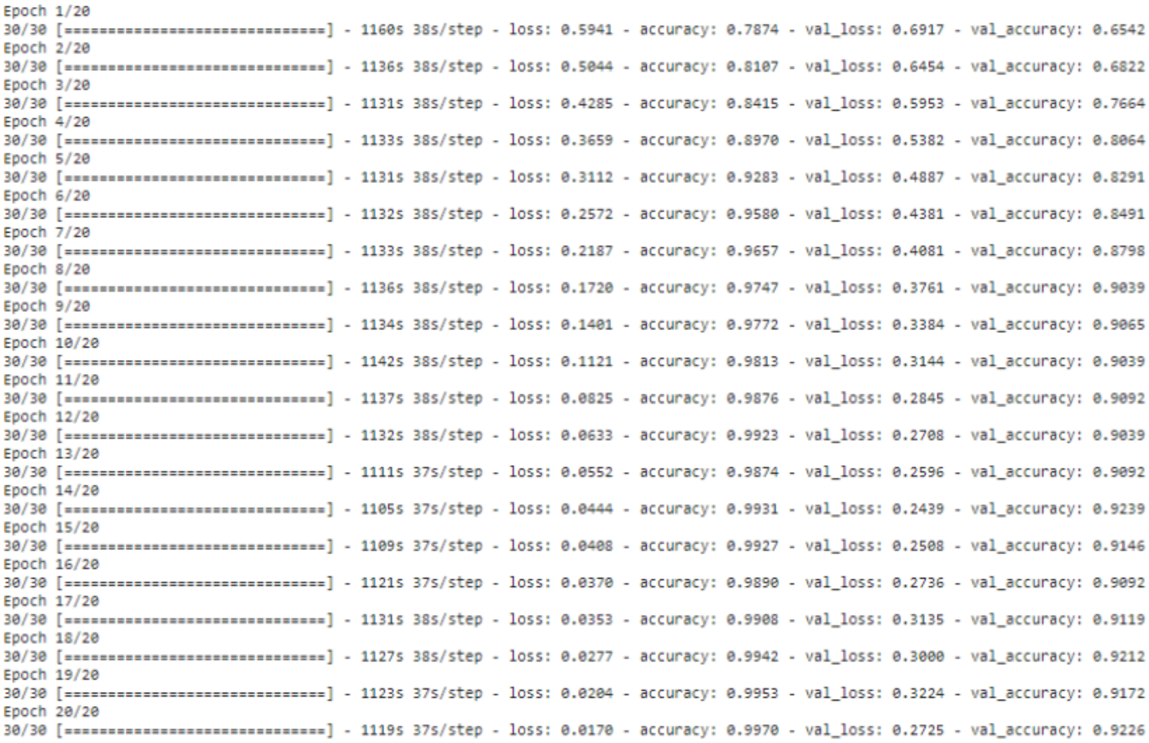
$$\text{Precision } = \frac{TP}{TP + FP}$$

Specificity is checked by how regularly the model could foresee false qualities among all real regrettable qualities. At the point when the model produced the result 'calm' demonstrating the individual is not inclined to compulsion, it projected how regularly the model could foresee the right adverse result.

$$\text{Specificity } = \frac{TN}{TN + FN}$$

### 5.1.1 Accuracy After applying VGG-16

Here, we have applied VGG-16 as our algorithm in out model. The arrangement of the completely associated layers is something very similar in all networks. All covered up layers are furnished with the amendment (ReLU) non-linearity. It is additionally noticed that none of the organizations (aside from one) contain Local Response Normalization (LRN), such standardization doesn't improve the presentation on the ILSVRC dataset, yet prompts expanded memory utilization and calculation time. Here, we have used VGG-16 that helps us to analyze our image data and push us to have a good classification. We have run this for Epoch=20. And find the accuracy and loss function shown in the figure below.



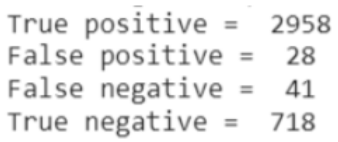
```
Epoch 1/20
30/30 [==============================] - 1160s 38s/step - loss: 0.5941 - accuracy: 0.7874 - val_loss: 0.6917 - val_accuracy: 0.6542
Epoch 2/20
30/30 [==============================] - 1136s 38s/step - loss: 0.5044 - accuracy: 0.8107 - val_loss: 0.6454 - val_accuracy: 0.6822
Epoch 3/20
30/30 [==============================] - 1131s 38s/step - loss: 0.4285 - accuracy: 0.8415 - val_loss: 0.5953 - val_accuracy: 0.7664
Epoch 4/20
30/30 [==============================] - 1133s 38s/step - loss: 0.3659 - accuracy: 0.8970 - val_loss: 0.5382 - val_accuracy: 0.8064
Epoch 5/20
30/30 [==============================] - 1131s 38s/step - loss: 0.3112 - accuracy: 0.9283 - val_loss: 0.4887 - val_accuracy: 0.8291
Epoch 6/20
30/30 [==============================] - 1132s 38s/step - loss: 0.2572 - accuracy: 0.9580 - val_loss: 0.4381 - val_accuracy: 0.8491
Epoch 7/20
30/30 [==============================] - 1133s 38s/step - loss: 0.2187 - accuracy: 0.9657 - val_loss: 0.4081 - val_accuracy: 0.8798
Epoch 8/20
30/30 [==============================] - 1136s 38s/step - loss: 0.1720 - accuracy: 0.9747 - val_loss: 0.3761 - val_accuracy: 0.9039
Epoch 9/20
30/30 [==============================] - 1134s 38s/step - loss: 0.1401 - accuracy: 0.9772 - val_loss: 0.3384 - val_accuracy: 0.9065
Epoch 10/20
30/30 [==============================] - 1142s 38s/step - loss: 0.1121 - accuracy: 0.9813 - val_loss: 0.3144 - val_accuracy: 0.9039
Epoch 11/20
30/30 [==============================] - 1137s 38s/step - loss: 0.0825 - accuracy: 0.9876 - val_loss: 0.2845 - val_accuracy: 0.9092
Epoch 12/20
30/30 [==============================] - 1132s 38s/step - loss: 0.0633 - accuracy: 0.9923 - val_loss: 0.2708 - val_accuracy: 0.9039
Epoch 13/20
30/30 [==============================] - 1111s 37s/step - loss: 0.0552 - accuracy: 0.9874 - val_loss: 0.2596 - val_accuracy: 0.9092
Epoch 14/20
30/30 [==============================] - 1105s 37s/step - loss: 0.0444 - accuracy: 0.9931 - val_loss: 0.2439 - val_accuracy: 0.9239
Epoch 15/20
30/30 [==============================] - 1109s 37s/step - loss: 0.0408 - accuracy: 0.9927 - val_loss: 0.2508 - val_accuracy: 0.9146
Epoch 16/20
30/30 [==============================] - 1121s 37s/step - loss: 0.0370 - accuracy: 0.9890 - val_loss: 0.2736 - val_accuracy: 0.9092
Epoch 17/20
30/30 [==============================] - 1131s 38s/step - loss: 0.0353 - accuracy: 0.9908 - val_loss: 0.3135 - val_accuracy: 0.9119
Epoch 18/20
30/30 [==============================] - 1127s 38s/step - loss: 0.0277 - accuracy: 0.9942 - val_loss: 0.3000 - val_accuracy: 0.9212
Epoch 19/20
30/30 [==============================] - 1123s 37s/step - loss: 0.0204 - accuracy: 0.9953 - val_loss: 0.3224 - val_accuracy: 0.9172
Epoch 20/20
30/30 [==============================] - 1119s 37s/step - loss: 0.0170 - accuracy: 0.9970 - val_loss: 0.2725 - val_accuracy: 0.9226
```

Figure 5.1: Accuracy after applying VGG-16

At the first epoch, we have got a Testing accuracy of 0.7874 and loss was 0.5941 and testing accuracy was 0.6917, value loss was 0.6542. After each epoch, we have seen accuracy was increasing and loss function decreasing. Surprisingly, at 20 epoch we have got training accuracy 0.9970 and testing accuracy was 0.9226 with very less loss function value.

### 5.1.2 Comparison of Training and Testing Accuracy with different Classifiers and VGG16

Here, we have seen we got the most TP and TN cases along with very few false cases. This indicates a good accuracy generated from the classification of our model.
We have got 2958 True positive indications and 28 false-positive classifications. The number of true negative cases found is 718 and false-negative cases are 41.
There are variation on training and testing accuracy of different classifiers.



```
True positive =  2958
False positive =  28
False negative =  41
True negative =  718
```

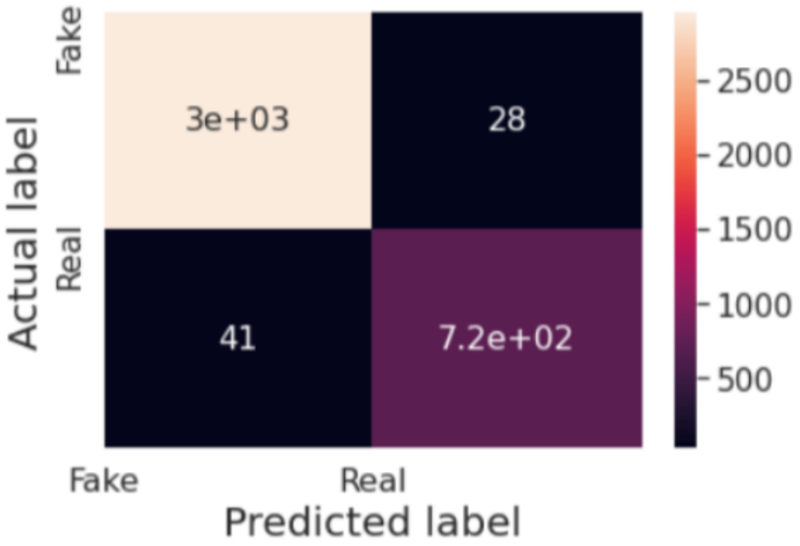


Figure 5.2: Confusion Matrix applying VGG16

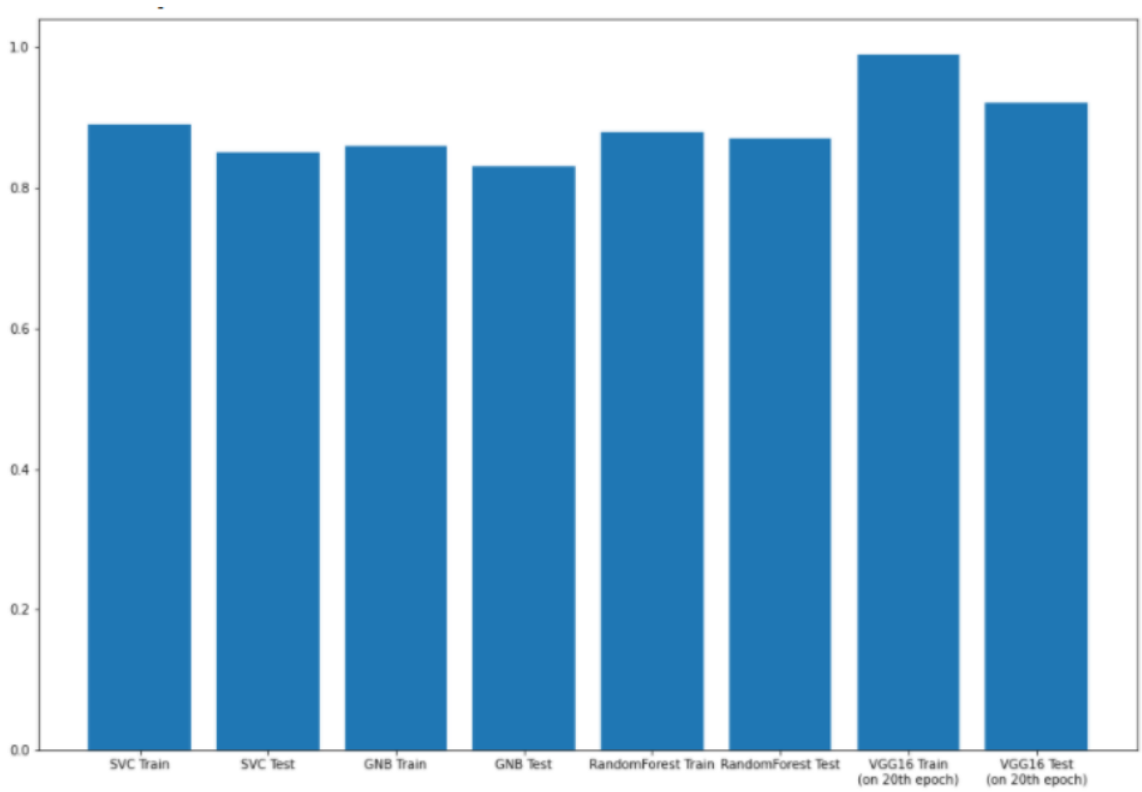


Figure 5.3: Comparison of Different Classifiers

| Name of the Classifier | Training Accuracy | Testing Accuracy |
|---|---|---|
| SVC | 0.89 | 0.85 |
| GNB | 0.86 | 0.83 |
| RF | 0.88 | 0.87 |
| VGG-16 (After 20 epoch) | 0.99 | 0.92 |

Table 5.1: Accuracy Table

In the case of SVC we can see the training accuracy is almost 0.89 which indicates strong training classifications. But in case of testing we have got 0.85 for SVC which shows a few false positives and false negative cases. However, in terms of GNB we have found training accuracy is 0.86 and testing accuracy slightly lower than other classifiers. It shows 83% testing accuracy. Compared with this we got slightly higher accuracy for Random forest classification where training accuracy was almost 0.88 and testing accuracy score was 0.87. Last but not least, we got very good prediction accuracy after extracting feature by our VGG-16 model. We got training accuracy score almost 1(0.99) and in terms of testing accuracy we got almost 0.92 score which indicates good prediction accuracy of our model.
The main objective of our research is to find an advanced and optimal Machine learning approach to detect deepfake videos as it has become a great threat to society.[29][44] In our model, we have shown from the video data extracting image and using PCA be we reduce the components to 100 and pass these data to deep learning algorithm VGG-16. Finally, we have used some classifiers like SVC, GNB,RFC etc. by which we showed the comparison between them with VGG16 and were able to prove the efficiency of our proposed model.

# Chapter 6

# Conclusion and Future Work

DeepFakes are affecting us in both active and passive ways in our lives. People who are usually not accustomed to technology much, are trusting every video and photos they are seeing on the internet. As technology is getting better and better, it is getting harder for everyone to detect which is real and which is a fake video. Previously, it was only limited to the fun purpose. But now, it is applied in different social and political aspects. Celebrities are constantly sufferers of DeepFake photos and videos. Making DeepFake videos with apps is easy, so anyone can do it anytime. When the purpose of it is to harm someone, DeepFakes can be dangerous. So, to get a better understanding of it and to be able to detect such videos, we have tried to do research on it. The main objective of our research is to find an advanced and optimal Machine learning approach to detect deepfake videos. In our model, we have shown from the video data extracting image and then using PCA be we have reduced the components to 100 and passed the data to deep learning algorithm VGG-16. Finally, we have used some classifiers like SVC, GNB, Random Forest, etc. by which we showed the comparison between them and have been able to prove the efficiency of our proposed model. As our data set is slightly biased, we are highly interested to make it balanced in our further work. Moreover, we want to extend our knowledge and are enthusiastic to research with different deep learning and neural network algorithms to improve our model. Last but not the least, we hope for a better world with privacy, security and we want to protect our world from any kind of misleading data and deepfakes.

# Bibliography

[1] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, “Kernel pca and de-noising in feature spaces.,” in *NIPS*, vol. 11, 1998, pp. 536–542.

[3] R. Rosipal, M. Girolami, L. J. Trejo, and A. Cichocki, “Kernel pca for feature extraction and de-noising in nonlinear regression,” *Neural Computing & Applications*, vol. 10, no. 3, pp. 231–243, 2001.

[4] A. Malhi and R. X. Gao, “Pca-based feature selection scheme for machine defect classification,” *IEEE transactions on instrumentation and measurement*, vol. 53, no. 6, pp. 1517–1525, 2004.

[5] L. I. Kuncheva and W. J. Faithfull, “Pca feature extraction for change detection in multidimensional unlabeled data,” *IEEE transactions on neural networks and learning systems*, vol. 25, no. 1, pp. 69–80, 2013.

[6] A. H. Jahromi and M. Taheri, “A non-parametric mixture of gaussian naive bayes classifiers based on local independent features,” in *2017 Artificial Intelligence and Signal Processing Conference (AISP)*, IEEE, 2017, pp. 209–212.

[7] M. Ontivero-Ortega, A. Lage-Castellanos, G. Valente, R. Goebel, and M. Valdes-Sosa, “Fast gaussian naıve bayes for searchlight classification analysis,” *Neuroimage*, vol. 163, pp. 471–479, 2017.

[8] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “Mesonet: A compact facial video forgery detection network,” in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, 2018, pp. 1–7.

[9] D. Güera and E. J. Delp, “Deepfake video detection using recurrent neural networks,” in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, 2018, pp. 1–6.

[10] M. Koopman, A. M. Rodriguez, and Z. Geradts, “Detection of deepfake video manipulation,” in *The 20th Irish machine vision and image processing conference (IMVIP)*, 2018, pp. 133–136.

[11] I. Amerini, L. Galteri, R. Caldelli, and A. Del Bimbo, “Deepfake video detection through optical flow based cnn,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[12] J. Brownlee, *18 impressive applications of generative adversarial networks (gans)*, Jun. 2019. [Online]. Available: https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/.

[13] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, “The deepfake detection challenge (dfdc) preview dataset,” *arXiv preprint arXiv:1910.08854*, 2019.

[14] H. R. Hasan and K. Salah, “Combating deepfake videos using blockchain and smart contracts,” *Ieee Access*, vol. 7, pp. 41 596–41 606, 2019.

[15] M.-H. Maras and A. Alexandrou, “Determining authenticity of video evidence in the age of artificial intelligence and in the wake of deepfake videos,” *The International Journal of Evidence & Proof*, vol. 23, no. 3, pp. 255–262, 2019.

[16] T. T. Nguyen, C. M. Nguyen, D. T. Nguyen, D. T. Nguyen, and S. Nahavandi, “Deep learning for deepfakes creation and detection,” *arXiv preprint arXiv:1909.11573*, vol. 1, 2019.

[17] B. Thai, “Deepfake detection and low-resource language speech recognition using deep learning,” 2019.

[18] X. Yang, Y. Li, and S. Lyu, “Exposing deep fakes using inconsistent head poses,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 8261–8265.

[19] L. Bondi, E. D. Cannas, P. Bestagini, and S. Tubaro, “Training strategies and data augmentations in cnn-based deepfake video detection,” *arXiv preprint arXiv:2011.07792*, 2020.

[20] L. Guarnera, O. Giudice, and S. Battiato, “Deepfake detection by analyzing convolutional traces,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 666–667.

[21] ——, “Fighting deepfake by exposing the convolutional traces on images,” *IEEE Access*, vol. 8, pp. 165 085–165 098, 2020.

[22] P. Gupta, K. Chugh, A. Dhall, and R. Subramanian, “The eyes know it: Fakeet-an eye-tracking database to understand deepfake perception,” in *Proceedings of the 2020 International Conference on Multimodal Interaction*, 2020, pp. 519–527.

[23] A. A. Maksutov, V. O. Morozov, A. A. Lavrenov, and A. S. Smirnov, “Methods of deepfake detection based on machine learning,” in *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, IEEE, 2020, pp. 408–411.

[24] S. J. Sohrawardi, S. Seng, A. Chintha, B. Thai, A. Hickerson, R. Ptucha, and M. Wright, “Defaking deepfakes: Understanding journalists’ needs for deepfake detection,” in *Proceedings of the Computation+ Journalism 2020 Conference. Northeastern University*, 2020.

[25] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, “Deepfakes and beyond: A survey of face manipulation and fake detection,” *Information Fusion*, vol. 64, pp. 131–148, 2020.

[26] W. Zhang, C. Zhao, and Y. Li, “A novel counterfeit feature extraction technique for exposing face-swap images based on deep learning and error level analysis,” *Entropy*, vol. 22, no. 2, p. 249, 2020.

[27] R. Caldelli, L. Galteri, I. Amerini, and A. Del Bimbo, "Optical flow based cnn for detection of unlearnt deepfake manipulations," *Pattern Recognition Letters*, vol. 146, pp. 31–37, 2021.

[28] ——, "Optical flow based cnn for detection of unlearnt deepfake manipulations," *Pattern Recognition Letters*, vol. 146, pp. 31–37, 2021.

[29] S. Das, A. Datta, M. Islam, M. Amin, *et al.*, "Improving deepfake detection using dynamic face augmentation," *arXiv preprint arXiv:2102.09603*, 2021.

[30] A. Deshmukh and S. Wankhade, "Deepfake detection by exposing ai-generated fake face video," in *Proceedings of Integrated Intelligence Enable Networks and Computing*, Springer, 2021, pp. 673–679.

[31] T. Fagni, F. Falchi, M. Gambini, A. Martella, and M. Tesconi, "Tweepfake: About detecting deepfake tweets," *Plos one*, vol. 16, no. 5, e0251415, 2021.

[32] M. Groh, Z. Epstein, C. Firestone, and R. Picard, "Comparing human and machine deepfake detection with affective and holistic processing," *arXiv preprint arXiv:2105.06496*, 2021.

[33] Y.-J. Heo, Y.-J. Choi, Y.-W. Lee, and B.-G. Kim, "Deepfake detection scheme based on vision transformer and distillation," *arXiv preprint arXiv:2104.01353*, 2021.

[34] J. Jiang, B. Li, B. Wei, G. Li, C. Liu, W. Huang, M. Li, and M. Yu, "Fakefilter: A cross-distribution deepfake detection system with domain adaptation," *Journal of Computer Security*, no. Preprint, pp. 1–19, 2021.

[35] R. K. Kaliyar, A. Goswami, and P. Narang, "Deepfake: Improving fake news detection using tensor decomposition-based deep neural network," *The Journal of Supercomputing*, vol. 77, no. 2, pp. 1015–1037, 2021.

[36] P. Korshunov and S. Marcel, "Subjective and objective evaluation of deepfake videos," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 2510–2514.

[37] P. Kwon, J. You, G. Nam, S. Park, and G. Chae, "Kodf: A large-scale korean deepfake detection dataset," *arXiv preprint arXiv:2103.10094*, 2021.

[38] Y.-C. Liu, C.-M. Chang, I.-H. Chen, Y.-R. Ku, and J.-C. Chen, "An experimental evaluation of recent face recognition losses for deepfake detection," in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 9827–9834.

[39] S. Pino, M. J. Carman, and P. Bestagini, "What's wrong with this video? comparing explainers for deepfake detection," *arXiv preprint arXiv:2105.05902*, 2021.

[40] L. Trinh and Y. Liu, "An examination of fairness of ai models for deepfake detection," *arXiv preprint arXiv:2105.00558*, 2021.

[41] L. Trinh, M. Tsang, S. Rambhatla, and Y. Liu, "Interpretable and trustworthy deepfake detection via dynamic prototypes," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1973–1983.

[42] J. Wang, Z. Wu, J. Chen, and Y.-G. Jiang, "M2tr: Multi-modal multi-scale transformers for deepfake detection," *arXiv preprint arXiv:2104.09770*, 2021.

[43] D. Wodajo and S. Atnafu, “Deepfake video detection using convolutional vision transformer,” *arXiv preprint arXiv:2102.11126*, 2021.

[44] P. Yu, Z. Xia, J. Fei, and Y. Lu, “A survey on deepfake video detection,” *IET Biometrics*, 2021.

[45] *Deepfake detection challenge.* [Online]. Available: https://www.kaggle.com/c/deepfake-detection-challenge/data.