Design and Development of Framework for DeepFake Video Detection using CNN and LSTM

A DISSERTATION

A MAJOR PROJECT REPORT
SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY
IN
INFORMATION SYSTEMS

Submitted by:

NIKITA DAGAR
2K20/ISY/12

Under the supervision of
Prof. Dinesh K. Vishwakarma

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi college of Engineering)
Bawana Road, Delhi-110042, India

MAY, 2024

CANDIDATE'S DECLARATION

I, Nikita Dagar, Roll No. 2K20/ISY/12 student of M.Tech., Information Systems, hereby declare that the Research Problem Formulation titled "Design and Development of Framework for DeepFake Video Detection Using CNN and LSTM" which is submitted by me to the Department of Information Technology, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi **NIKITA DAGAR**

Date: May 31, 2024 **(2K20/ISY/12)**

CERTIFICATE

I hereby certify that the Research Problem Formulation titled "Design and Development of Algorithm for DeepFake Video Detection Using CNN and LSTM" which is submitted by Nikita Dagar, Roll No. 2K20/ISY/12 Information Technology, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi.

Date: May 31, 2024

Prof. Dinesh K. Vishwakarma

SUPERVISOR

DEPARTMENT OF INFORMATION TECHNOLOGY

ABSTRACT

The increasing worldliness of mobile and camera based techniques which have been seen to have expanding scope of multi-media content and the internet based facilities have made it more convenient than ever before to produce and distribute digital videos. Although the manipulations of digital images andvideos have been seen by the use of visual effects for several decades, recent developments in deep learning have caused a drastic increase in the real content and accessibility of the manipulated content that can be produced. Recent advancements in automated video and audio editing tools, **Generative Adversarial Networks** (**GAN)** and a good and manipulated video content obtained from internet or social media can be produced and easily disseminated.

In todays scenario, it has become difficult for people to distinguish the fact from fiction because of the technological advancements made in automated approaches for creating and sharing content online. A survey of videos, often indecent, the face of one person in the source video is swapped with face of other person in the resultant video using deep neural network that automatically maps the face expressions of the person in the original video to the expression of other person in manipulated video, so called deepfakes which are grabbing a lot of widespread alarm.

Deepfakes are named so because they use artificial neural network and deep learning to create fake content. The deep learning based models have been widely used for creating manipulated content and some of the models are **Autoencoders** and **Generative Adversarial Networks (GAN)**. These models monitors face expressions and different kinds of movement of the person in the original video and then synthesize face images of the another person in the manipulated video to make similar face expressions and other movements.

Nowadays creating deepfake is quite easier because of the developments of applications like faceapp and fakeapp, nowdays anyone can use these application in order to create their own manipulated content. So, it has become very important to detect deepfakes in order to avoid the spread of the fake content. This report presents the survey of algorithms used to create deepfake and deepfake video content detection methods.

ACKNOWLEDGEMENT

I am very thankful to **Prof. Dinesh K. Vishwakarma** (Professor, Department of Information Technology) and all the faculty members of the Department of Information Technology at DTU. They all provided me with immense support and guidance for the project.

I would also like to express my gratitude to the University for providing us with the laboratories, infrastructure, testing facilities and environment which allowed us to work without any obstructions.

I would also like to appreciate the support provided to us by our lab assistants, seniors and our peer group who aided us with all the knowledge they had regarding various topics.

Nikita Dagar
Roll No. 2K20/ISY/12
M.TECH (Information Systems)

CONTENT

TABLE OF FIGURES

TABLE OF ABBREVIATIONS

ABBREVIATIONS	FULL FORM
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
DNN	Deep Neural Network
ANN	Artificial Neural Network
ResNet	Residual Neural Network
VGG	Visual Geometry Group
LSTM	Long Short-Term Memory
ReLU	Rectified Linear Unit

CHAPTER 1

INTRODUCTION

The term Deepfake comes deep learning and fakee, is a technology with the help of which we can replace the face of a person in a source video with the face of another person in the target video to create the video of target person doing and saying things that the source person does. The access to the open source and applications like FaceApp and FakeApp for such face swapping leads to the generation of large amount of manipulated video content from internet and television, all these leads to design and develop algorithms in order to detect manipulate video content that is being widely spread through internet.

The first manipulated video have been came in 2017 in which have been manipulated by swapping the face of a celebrity with the face of a porn actor. Deepfakes therefore are a serious threat to attack the reputation of public subjects and can cause humiliation. Moreover, the impact of deep fakes is magnified by the social networks that delivers the information quickly and worldwide.

These AI-synthesized media commonly termed as deepfakes comes in one of the three categories:

- **Face-swap**, the face of a person in source video is being swapped with the face of another person in multi-media i.e the video content. The above technique has been used to replace the face of porn actor with theface of a celebrity or to insert faces of actors into video clips in which they never appeared.
- **Lip-sync**, in this the source video have been modified in such a way that we try to keep the mouth region consistent during performing manipulations. One the most shocking example is of President Obama whose video has been manipulated by the altering the speech content that is in the manipulated video he was saying things which he never said.
- **Puppet Master**, in this the manipulated video content is created by manipulating various features like movement of head moved, movement of eyes, the expressions of face, etc.

In order to create manipulated content, we need a model that needs to be trained with a large dataset which consists of good number of images and video content. In the first place the movie actors and actor or the actresses of cinema world are being target and even the political

leaders have been targeted in order to create the manipulated content because their images and videos are freely and widely available on the internet. The availability of their images and their videos in such a large number over the internet have made it easier to create their manipulated content. So now can conclude that Deepfakes have become a great threat to the privacy of popularly known celebrities and even to their democracy and it is also great threat to the security of nation. Therefore, deepfakes leads political or religious conflicts across countries.

The development, advancement of deep learning methods and the availability of data in abundant amount over the internet have made it difficult for humans and even computer algorithms to distinguish between fake and original content. Nowadays creating deepfakes as shown in **Figure *1*** has become so easy that it just need a passport sized photo or a short video clip of the target.

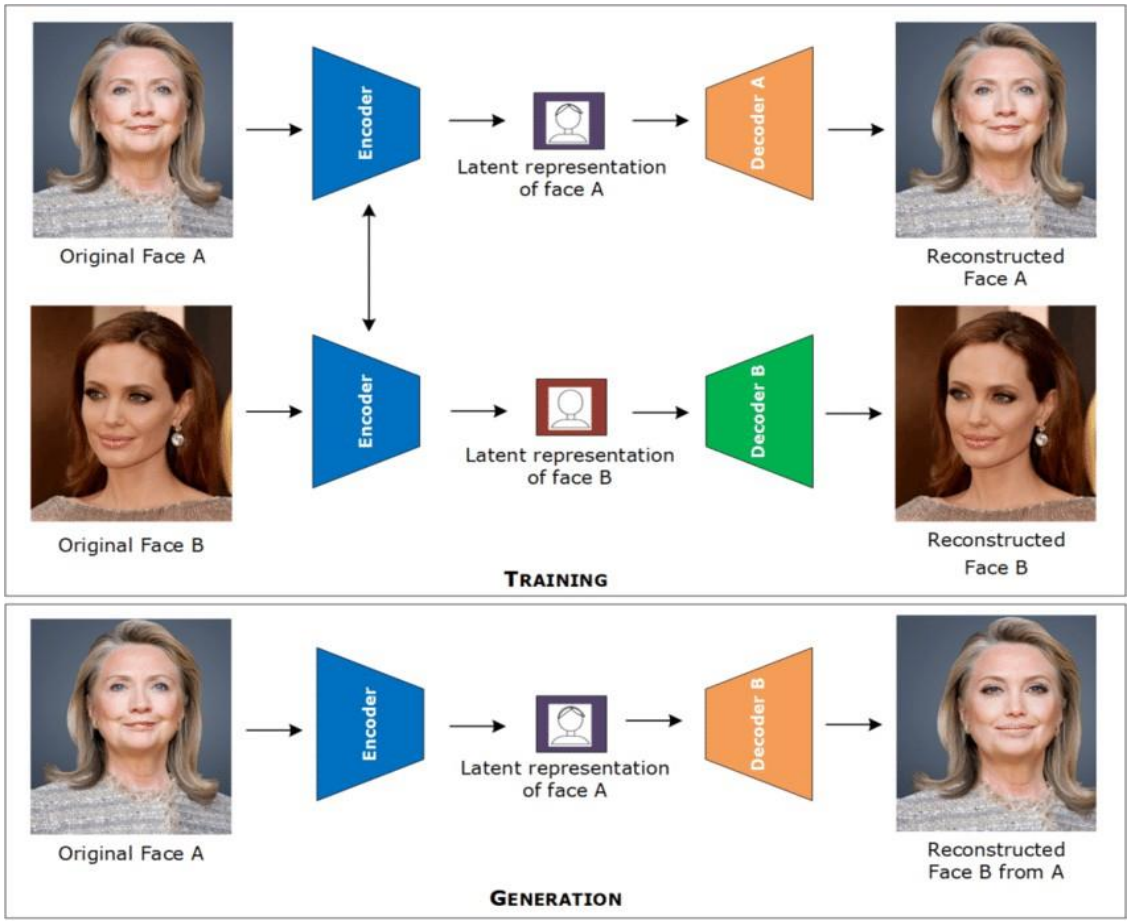


Figure 1: Demonstration how deepfakes are created using autoencoder and decoder

1.1 CONVOLUTION NEURAL NETWORKS:

CNN's as shown in **Figure** *2* is a kind of Deep-Neural Network which is capable of recognizing and categorizing particular features from any visual image data. The major applications in which CNN's are widely used are computer vision, video recognition, medical image analysis, image classification. The word Convolution represent an operation in which mathematical computations are being performed on two functions in order to get the resultant function, the mathematical operation that is performed is multiplication. The multiplication represents how the shape of one function is dependent on another. In CNN images are being represented in the form of matrices and the mathematical operation that is being performed is the matrix multiplication in order to fetch the features from images. A convolutional neural network is a kind of deep neural network, which is widely being used in lot of recent computer vision tasks. The various CNN architectures such as VGG-Net, AlexNet, ResNet, DenseNet, etc are being designed in such a way that they can themselves learn and adapt spatial features by backpropagating through various building blocks.

The various building blocks that constitute a CNN are:

- Convolution layers: Feature extraction
- Pooling layers: Downsampling
- Fully Connected Layers: Classification

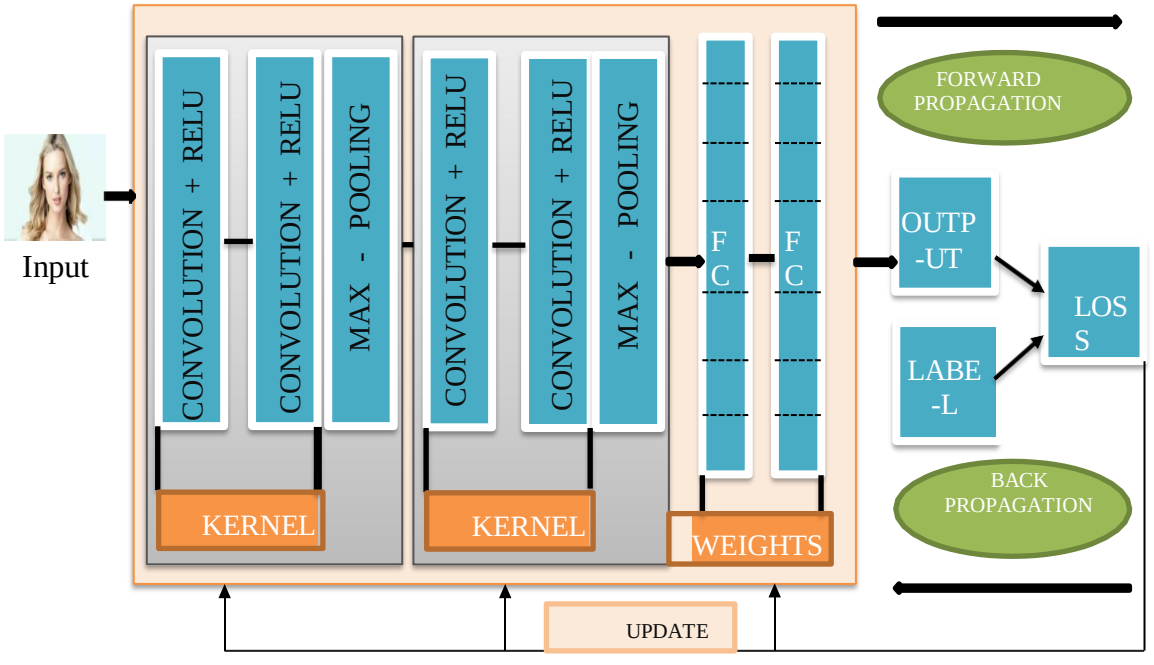


Figure 2: Convolutional Neural Network

1.1.1 CONVOLUTION LAYER:

The convolution layer is basically used for extracting features from the visual data. In this layer the mathematical operation as shown in **Figure *3*** is performed in which in which image represented by the matrix is multiplied with the filter which is also a matrix of size M*M. The output of this operation is the feature map. The feature map provides us various vital information of an image like corners, edges, etc. For further operation this feature map is used as input to other layers in order to learn more minute features of an image. It is a combination of linear (Convolution) and non-linear (Activation) operations.

- **Convolution operation**: It involves convolving Kernel, which is an optimizable feature extractor. Features are being extracted in a hierarchical manner from low level to high level features. The size and the number of kernels are the two key hyper-parameters that we need to define for the convolution operations. The convolution operation generates the feature maps of reduced height and width as compared to input tensor.
- **Activation operation**: The output of the convolution operation is passed through the activation function in order to introduce non-linearity. Activation functions are the mathematical representation of biological neuron behavior. ReLU is the most

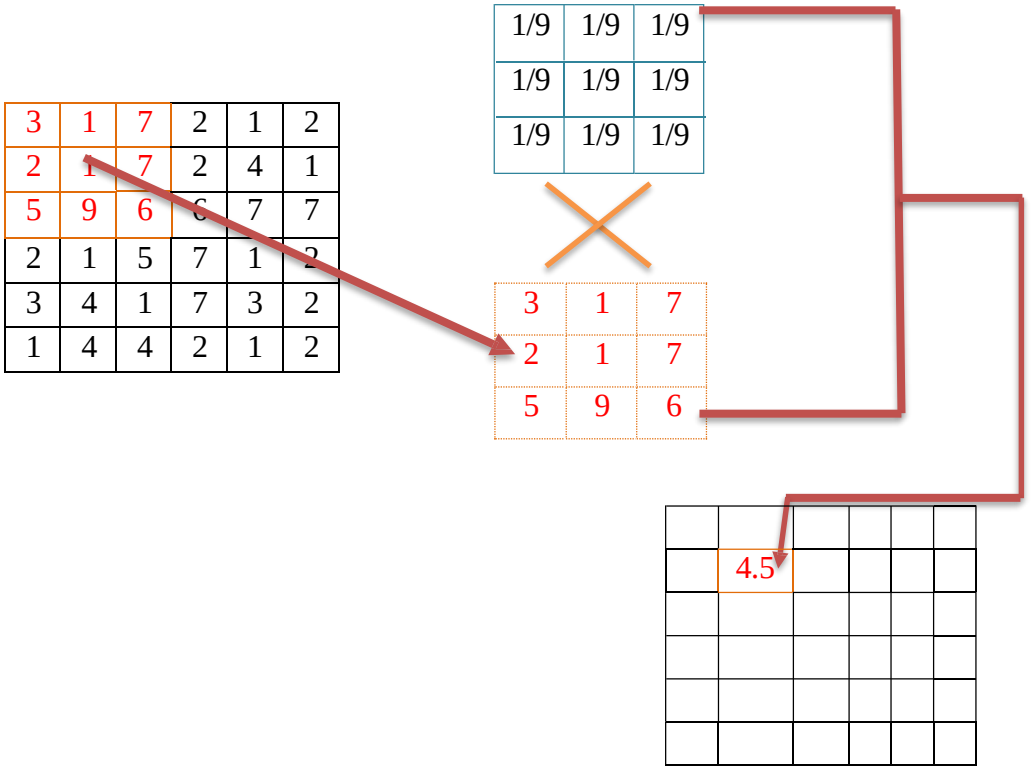


Figure 3: Demonstration of Convolutional Operation

commonly used activation function because it overcomes the vanishing gradient problem. ReLU is a simple linear function: f(x) = max(0,x). The use of Activation function in Deep Neural Network plays a vital role. The kind of Activation function to be used in hidden layers defines how the model will learn while training and the kind of Activation function to be used in the output layer will define what kind of predictions will be made by our trained model. It is very important to decide which activation function to be used in different deep neural networks. The use of activation function defines how we convert the weighted sum of an input from a node or nodes into an output in a neural network.

Activation functions are mostly used to introduce non-linearity in the deep neural network. Activation function have been given many names based on the functionality as demonstrated in **Figure *4*** it provides for example "Transfer function", "Squashing function", "Non-Linear function". The kind of activation function we use in the hidden layers also define the performance of deep neural networks. The kind of activation function to be used in the output layer depends on the type of the prediction to be made by neural network.

By using activation function we actually perform differentiation on input value mostly first-order derivative.

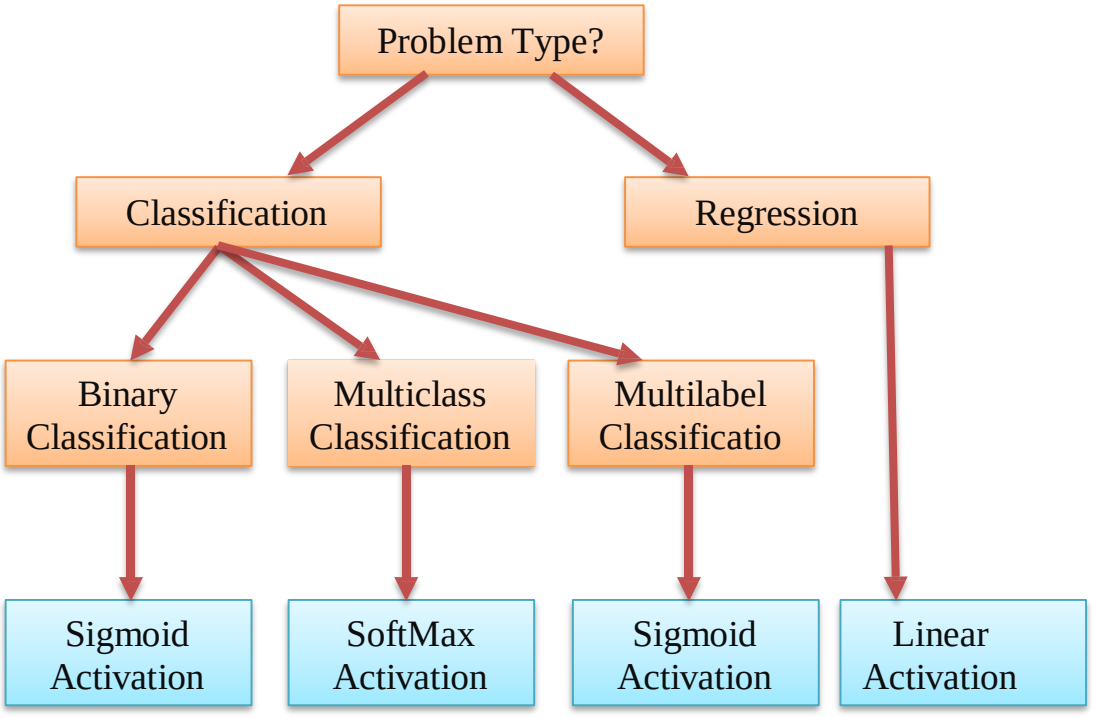


Figure 4: Types of Activation Functions Based on Problem Type

1.1.2 POOLING LAYER:

In allmost all cases the convolution layer is followed by pooling layer. The main objective of using this layer is reduce the size of the output of the convolution layer that is the feature-map. Pooling layer is able to reduce the size of the feature map by performing simple operation in which it reduces the number of connections between the consecutive layer and this operation is independently performed on each of the feature map. Pooling layer performs downsampling operation. While downsampling it simply reduces the dimensionality of the feature maps which in turn reduces the number of learnable parameters.There are two most common pooling operations:

- **Max Pooling**: It extracts the maximum value in each patch of input feature maps and discard all other values as shown in **Figure *5***. This layer reduces the size of the feature map that is in turn reduces the size of the image by small amount by adding translation invariance.

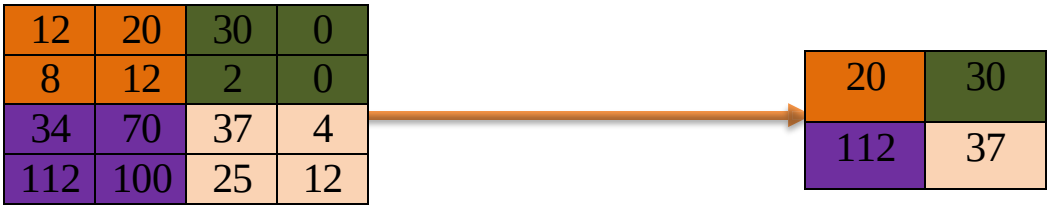


Figure 5: 2*2 MAX-POOL

- **Global Average Pooling**: It downsamples the height and width of the input feature maps to an 1X1 array by taking average of all elements in each feature map as shown in **Figure *6***. It not only reduces the number of learnable parameters but also allows CNN to accept inputs of different sizes.

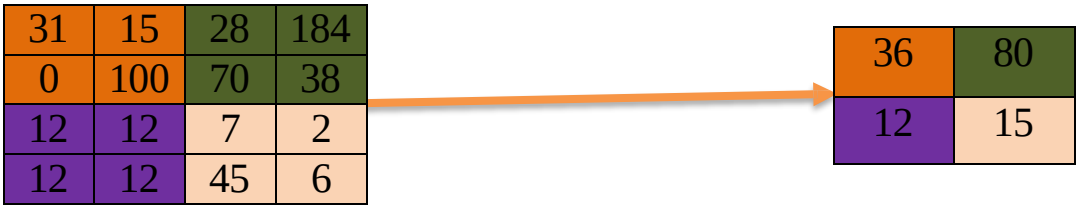


Figure 6: 2*2 AVG-POOL

1.1.3 DROPOUT LAYER:

Most of the cases while training the dataset leads to the overfitting problem. Overfitting problem arises because when all the features are connected to the fully connected layer. In the case of overfitting problem model gives better accuracy on training data than on validation data or new data. This overfitting problem is overcomed by using dropout layer. Using dropout layer small amount of neurons are dropped while training the neural network as shown in **Figure 7** which in turn reduces the size of the model. If in dropout layer 0.3 value is passed, then 30% neurons i.e nodes are dropped out randomly from the neural network.

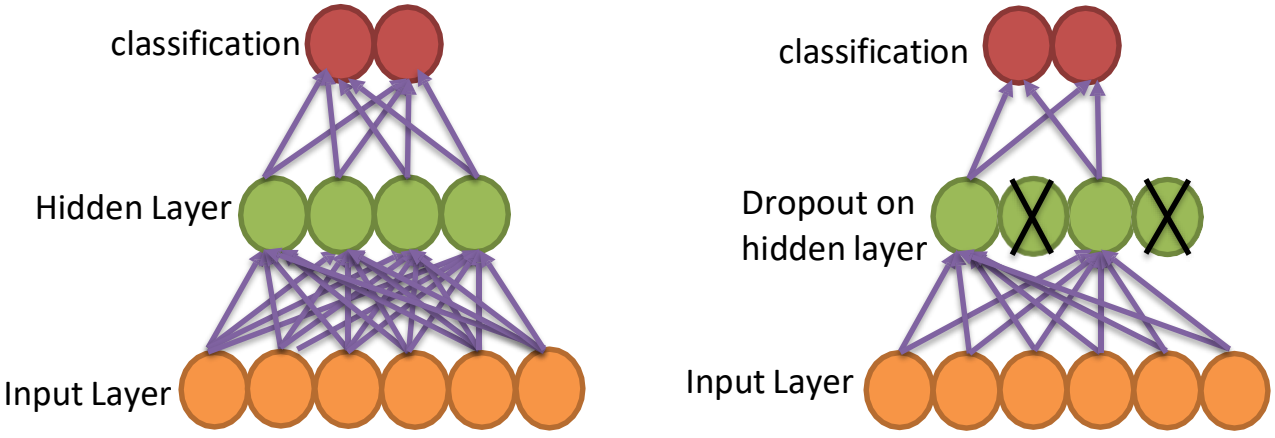


Figure 7: Figure demonstrating effect of using Dropout layer

1.1.4 FULLY CONNECTED LAYER:

The fully connected layer simply maps the extracted features to output. Firstly the output of last convolution or pooling layer is flattened i.e it transform the output into a 1-D array then pass it through sequence of fully connected layers in which every neuron is connected to every other neuron. The last fully connected layers have number of neurons equivalent to number of classes. It is mainly used for classification where it returns the probabilities corresponding to each class using Softmax activation function.

1.2 RECURRENT NEURAL NETWORKS:

Recurrent Neural Networks have overcome the problems faced by the traditional CNN architecture by allowing the previous outputs to be used as inputs while having the hidden layers. RNN are neural networks with loops in them. With the help of loops RNN are capable of passing information from one part of the network to the next part. RNN can be visualized

as multiple copies of the same network, where each copy pass information to the successor. Because of the chain like structure Recurrent Neural Networks are mostly related to sequencies and lists.

RNN **Figure *8*** are widely used in various applications such as translation, image captioning, speech recognition, language modeling, etc. The key feature of RNN is its capability to process inputof any length as depicted in **Figure *9***. Using RNN we need to increase the size of the model with the increase in the size of the input. LSTM is a special kind of Recurrent Neural Network, which enables to restore long term dependencies. LSTM's also overcomes the vanishing gradient problem in backpropagation through time algorithm during the training phase.

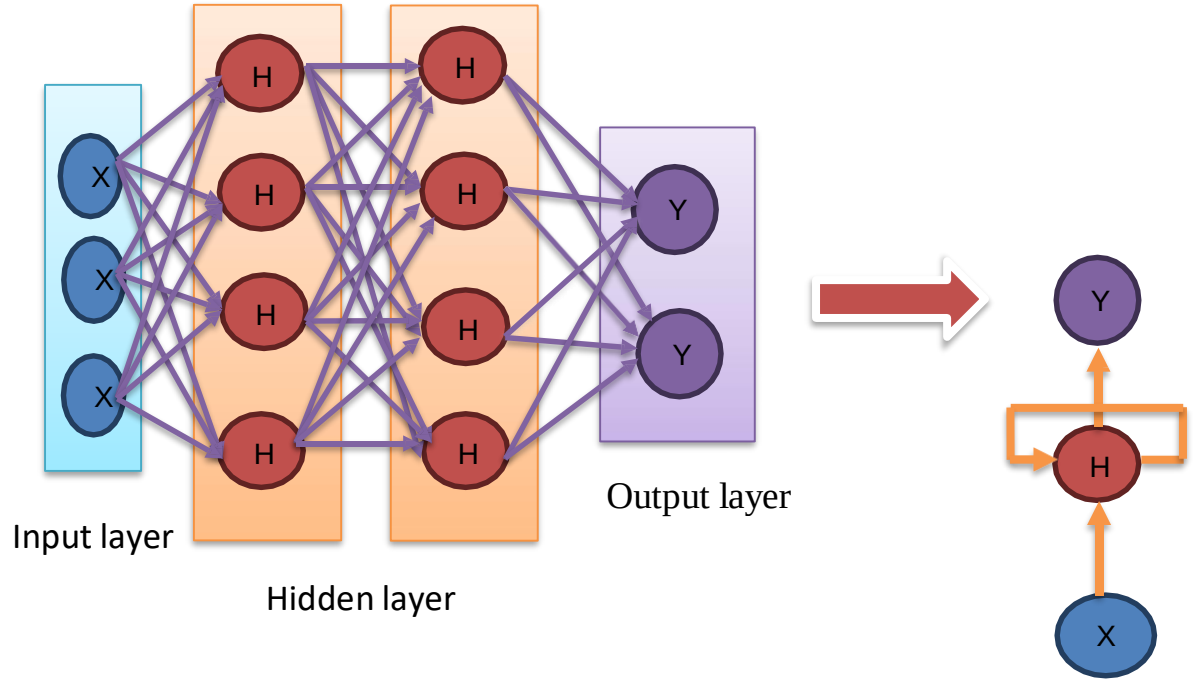


Figure 8: Simple Recurrent Neural Network

- **<u>Loss Function</u>**: The loss function in case of Recurrent Neural Network is defined based on the loss at each time step as depicted in **Equation *1***.

$$f(\hat{y}, y) = \sum_{t=1}^{T_y} f(\hat{y}<t>, y<t>) \quad (1.1)$$

Equation 1: Loss function of RNN

- **Backpropagation Through Time**: In Recurrent Neural Network, backpropagation is done at each point in time as depicted by **Equation *2***. During backpropagation, at every timestamp, we calculate the derivative of loss with respect to the weight matrix.

$$\frac{df(T)}{dW} = \sum_{t=1}^{T} \frac{df(T)}{dW} \quad (1.2)$$

Equation 2 : Back Propagation of RNN

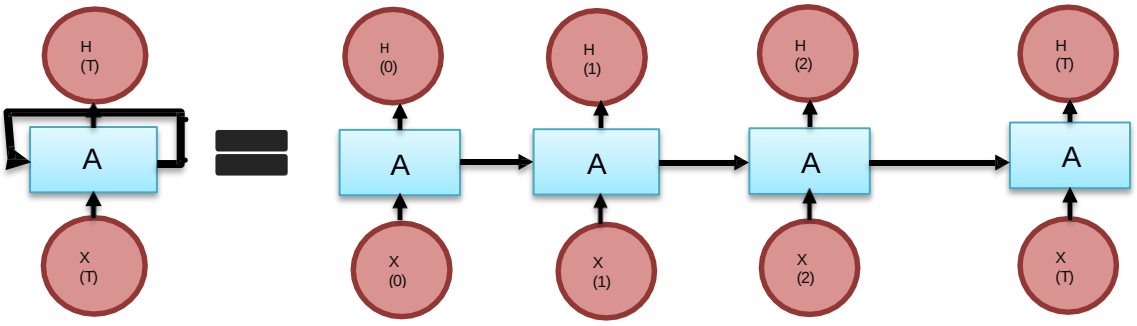


Figure 9: Unrolled Recurrent Neural Network.

Data obtained from https://app.dimensions.ai at the end of July 2020 shows that the number of deepfake research papers has significantly increased in recent years as shown in **Figure *10***.

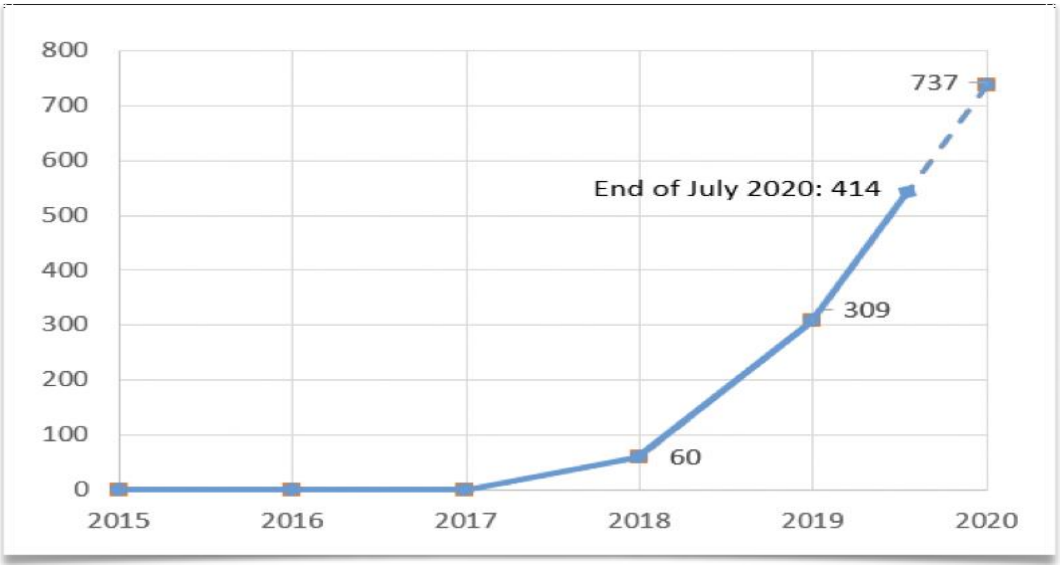


Figure 10: Number of papers related to Deepfake in years 2015-2020

CHAPTER 2

LITERATURE REVIEW

The field of DeepFake detection can be broadly classified into several approaches involving Image Recognition [1], 3D shape reconstruction [2], etc.

Donahue et al. [3] introduce a novel Long-term Recurrent Convolutional Network (LRCN) that combines conventional convolutional neural nets as feature extractor with recurrent architecture for the ask of visual recognition and description. Agarwal et al. [4] detect deepfake videos from the mouth shape (viseme) inconsistencies in deepfake videos introduced while pronouncing specific phonemes by utilizing a CNN to detect whether a video frame contains an open or closed mouth.

Fernandes et al. [5] utilize heart rate to differentiate between original and deepfake videos by training the state-of-the-art Neural Ordinary Differential Equations (Neural-ODE) model with the heart rate of original videos and using the same to predict the heart rate of deepfake videos created from commercial software. Sabir et al. [6] explore various recurrent convolutional models for facial manipulation detection in videos including face preprocessing steps to achieve state-of-the-art results on face manipulation video benchmarks. Fernandes et al. [7] utilize the novel Attribution Based Confidence (ABC) metric for deepfake video classification from a model trained on original videos only which ideally yield a confidence score greater than 0.94. Li et al. [8] trains CNNs to detect common artifacts introduced during generation of DeepFake videos by using affine face warping artifact as the discriminative feature to classify any video as original or fake.

Koopman et al. [9] utilized photo response non-uniformity (PRNU) analysis to detect DeepFake videos. Mean normalized cross correlation scores extracted from PRNU analysis demonstrate difference between original and manipulated videos. Soukupova et al. [10] proposed a real time algorithm, which estimates landmarks to extract eye-aspect-ratio (EAR). The proposed algorithm uses SVM classifier to differentiate original and fake videos and is robust to varying head orientation.

Agarwal et al. [11] prevent DeepFakes from maligning world leader reputation by using facial expressions and movements as depicted in **Figure *11*** to study their corelation in authentic or fake videos. Cozzolino et al.[12] proposed a novel approach ID-Revel that

extracts temporal facial features and uses metric learning with adversarial training strategy to overcome poor generalization of traditional methods trained to detect only specific facial manipulations in videos.

Other approaches include variation in eye-blinking [10], [13], [14], out of sync lip movement [15] have also been used for DeepFake video detection.

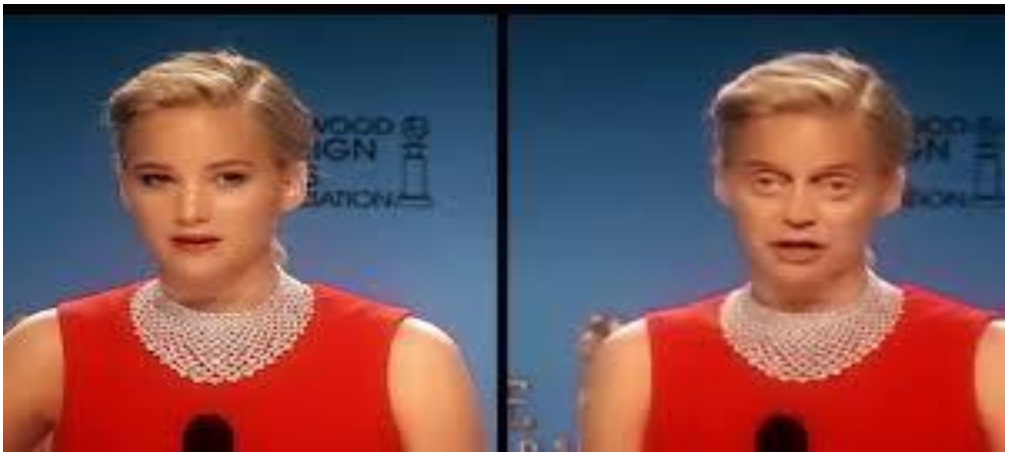


Figure 11: A clip of a video demonstrating the impact of deepfake video created

CHAPTER 3

PROPOSED MODEL

3.1. OVERVIEW OF PROPOSED APPROACH:

We have used a hybrid model (CNN + LSTM) to detect manipulated videos as depicted in **Figure *12***. Our model comprises of three parts namely feature extraction, sequence learning, state prediction. Firstly, we have trained ResNet50 based CNN architecture to perform binary classification to predict the state of eye, whether it is closed (0) or open (1). CNN-based architectures are commonly used for Image classification. Image classification is a process in which we segment the into different categories (closed eye and open eye state) based on the extracted features. Since eyeblinking is a temporal process, we have also used LSTM-RNN to fetch the temporal knowledge that is required to distinguish between the open and close eye state in a video. The feature extraction that we performed converts the input eye region into discriminative features, which are given as input to the LSTM for sequence learning. LSTM alsoFavoids the gradient vanishing problem during back propagation in the training phase. Finally, we have fully connected layers to predict the rate of eye blinking.

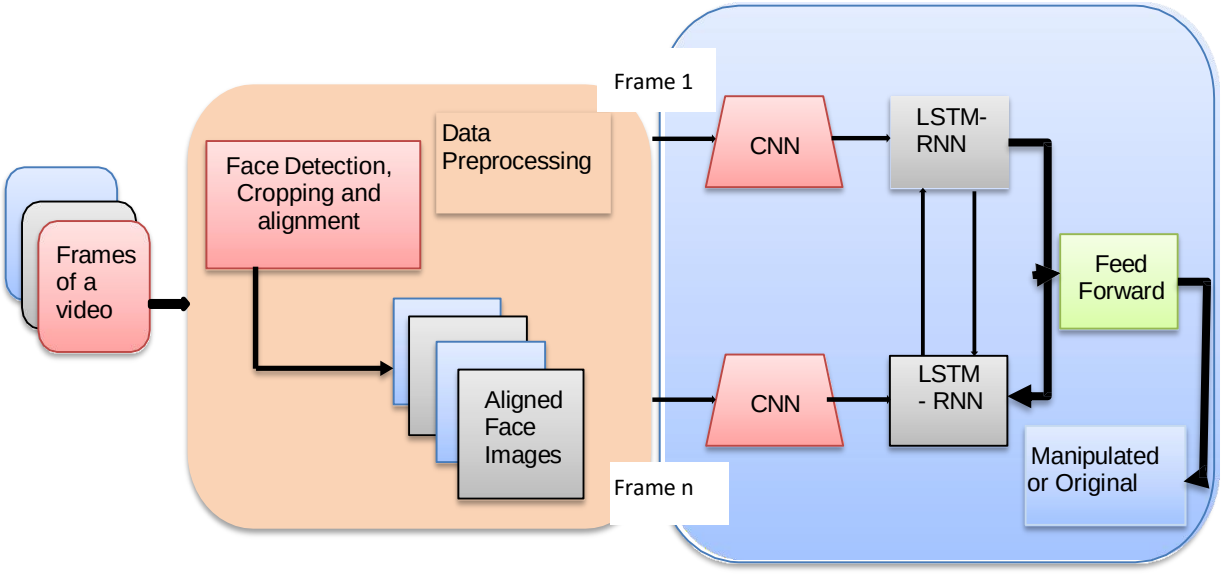


Figure 12: Architecture of Proposed Model

3.2. ARCHITECTURE USED:

3.2.1 RESNET50 – CNN:

ResNet-50 is one of the convolutional model which is widely used for classification applications. ResNet-50 is a convolutional model which has 50 layers as depicted in **Figure *13***. There are various other variants of ResNet which are available like ResNet-34, ResNet-101, etc. The ResNet architecture was firstly introduced by Kaiming He, Shaoquing Ren, and Xiangyu Zhang and Jain Sun in 2015. They have also published a research paper titled as "Deep Residual Learning for Image Recognition" in computer vision. While working with deep convolutional neural networks a lot many problems have been faced, one of the most common problem faced is the overfitting, degradation. As we increasing the number of layers in neural networks we might get good accuracy levels but its starts degrading slowly slowly which results in deteriorating the performance of the model on both training and testing data. This problem is occurs due to the exploding and vanishing gradients.

ResNet architecture are being basically created to tackle these problems. The use of the residual blocks results in improving the performance of the model. The main strength of this architecture are the skip connections which is vital part of residual blocks. These skip connections eliminate the problem of vanishing gradient by providing another shortcut connections for the gradient to pass. The use of skip connections makes sure that higher level of layers doesn't perform worse than lower level of layers because they are capable of learning identity functions.

The use of the residual blocks in ResNet architecture enables the layers to learn the identity function easily. The Resnet architecture reduces the percentage of errors which in turn results in improving the performance and efficiency of deep-neural-networks. The ResNet architecture have enabled to use as many layers as possible without worrying about problems like overfitting and degradation.

ResNet-50 architecture is very much similar to ResNet-34. Both the architecture follows the same bottleneck design, in ResNet-34 there is 3-layer and in ResNet-50 there is 3-layer bottleneck design.

Keras is one the most commonly used deep learning API because it provides various pre-trained models. ResNet-50 is one of the pre-trained model provided by the Keras. Residual

architecture is widely used for classification applications. When we use ResNet-50 with keras firstly we need to run the code in order to define the identity blocks which in turn transforms the convolutional network into deep residual-network and then finally build the convolutional block. In the second step, by combining the identity block and the convolution block a 50-layer ResNet architecture is builded. In the final step we only needs to train our model and by using keras library we easily generate the summary of the trained model. Residual architectures have brought a significant change in training process of deep convolutional network and it ensures better accuracy in most of computer vision tasks.

A 50 layer variant of the novel ResNet architecture is also used for binary image classification. A Residual block contains the novel 'shortcut connections' that skip one/more layers and help in combating the vanishing gradient problem by allowing gradients to flow

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2 $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 13: The Standard Architecture of ResNet50 Architecture

through shortcut connections during backpropagation. Two distinct kinds of shortcuts have been utilized in this implementation. The first is the 'identity block' in which the input output combination shares the same dimensions. The second kind uses convolutional layers and has different dimensions for its input and output. Such a shortcut connection is termed as 'convolutional block'.

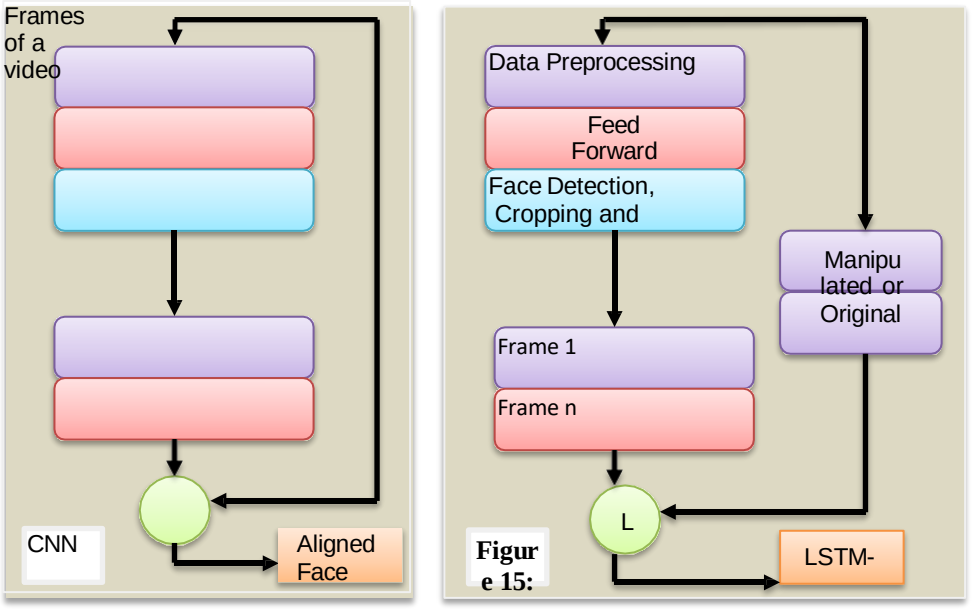


Figure 14: (a) Identity residual block (b) Convolutional residual block

The implemented ResNet50 architecture as depicted in **Figure *16*** works in five stages**.** As a preprocessing step, input images are padded with padding size of 3. The stage 1 includes 2D convolution, batch normalization and max- pooling layers. 64 filters of size 7 X 7 and stride (2, 2) are used. MaxPooling layer uses a (3, 3) window size with stride of (2, 2). Stage 2 contains a convolutional block and 2 distinct identity blocks. The convolutional block utilizes 3 filters of size 64, 64 and 256 respectively. Each of the two identity blocks also use 3 filters of size 64, 64 and 256 respectively. Similarly, stage 3, 4 and 5 also implement the convolution plus identity block as depicted in **Error! Reference source not found.** combination while varying the size of filters used. In stage 3 which contains 3 identity blocks, theconvolutional block and identity blocks both have 3 filters of size 128, 128 and 512 respectively.

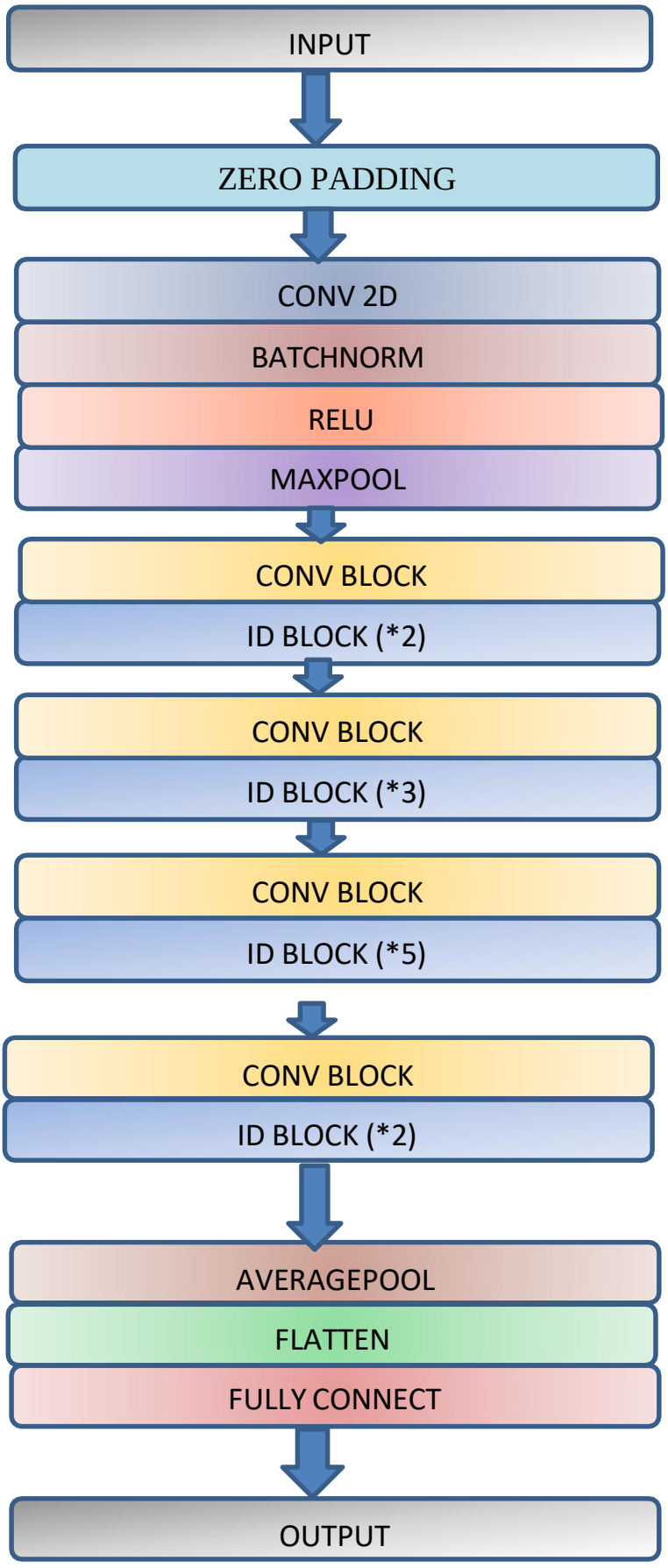


Figure 16: Detailed Overview of Implemented ResNet50 Architecture

In stage 4 having 5 identity blocks, the filter size is varied to 256, 256, 1024. Stage 5 containing 2 identity blocks have filter sizes of 512, 512, 2048 for its constituent blocks. After 5 stages, the features are processed with AveragePooling, flattening layer and finally a fully connected layer to reduce the number of features to classes i.e., 0 or 1.

3.2.2 : EFFICIENTNET – CNN:

Convolutional Neural Networks have become very important in the field of Computer Vision since its performance in classification applications. It has been found that the performance of efficientnet is much better than rest of state of art. To scale the model the conventional practice that is being opted is either by increasing the depth or the width of the model. ResNet architecture have scaled the model by increasing the number of layers from ResNet-18 to ResNet-200. But this way of model scaling in order to achieve better accuracy is very tedious because it needs to be done manually. EfficientNet is one of the new convolutional models which came into picture in 2019 through the research paper "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". This architecture has introduced a new way of scaling the model by using the compound coefficient. The Compound Coefficient scale the model in a more structured and simpler way. In this process the model is scaled uniformly by scaling each dimension in fixed proportion by using scaling coefficient.

A lot of studies have been done to understand the effect of scaling the model. It has been observed that by scaling each different dimensions independently improves the performance of the model.

3.2.2.1 : COMPOUND SCALING:

Compound Scaling as shown in **Figure *17*** is another approach opted to scale the model, this approach is yet simpler and efficient. In this approach the very first step is to find the relationship between different scaling dimensions of the baseline network by performing grid search and this search is performed under fixed resource constraint. This approach improves the performance of the model consistently and effectively. When we apply compound scaling on MobileNet it has improved the accuracy on imagenet dataset by +1.4% and when applied on the ResNet architecture the accuracy on the imagenet dataset has been improved by +0.7%.

In order to improve-the-accuracy of the model we need to scale the model either by increasing the number of layers i.e. the height of the neural network or by increasing the number of neurons in each layer i.e the width of the neural network and by increasing the height and width of the image i.e the resolution of the neural-network.

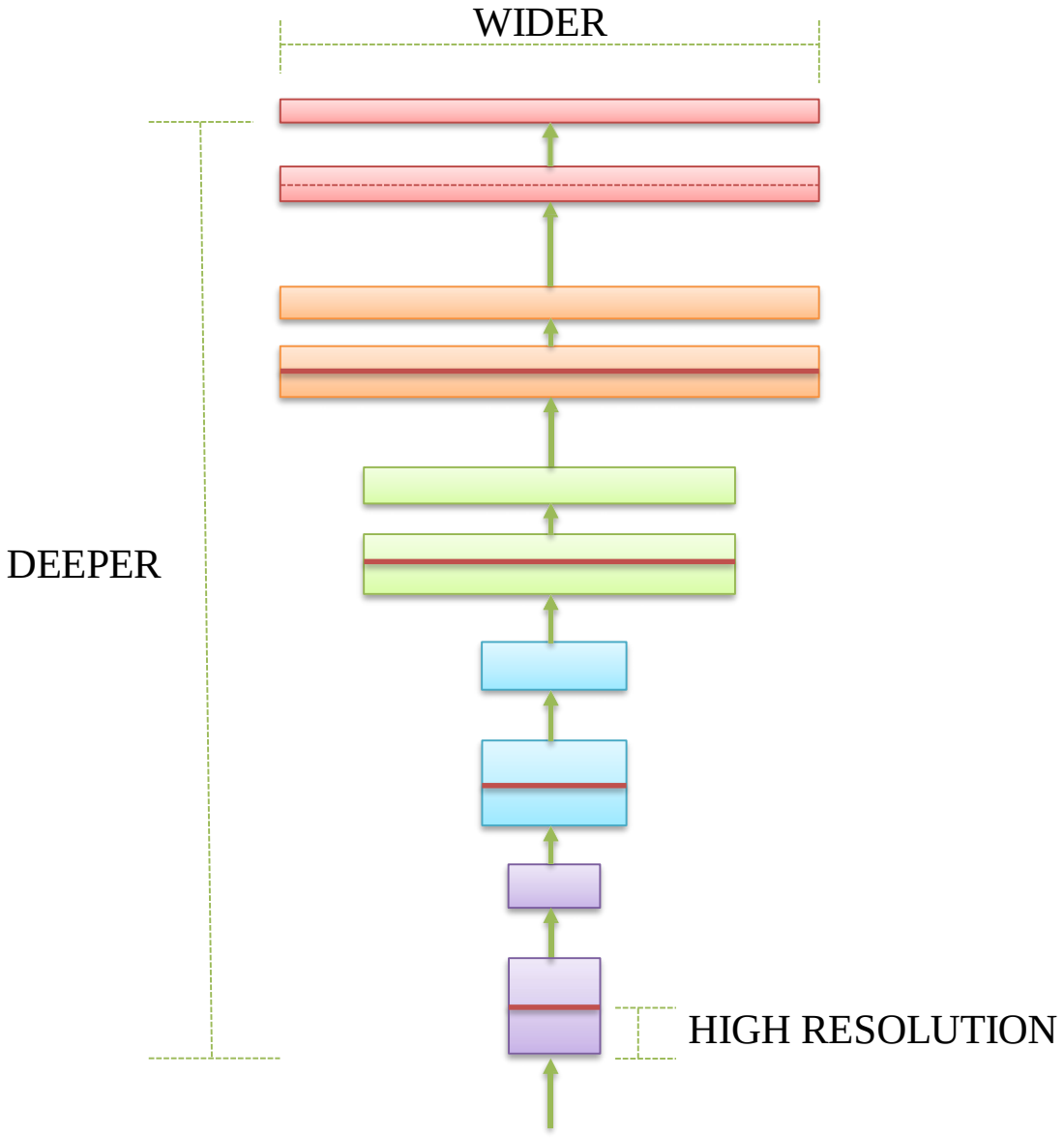


Figure 17: Compound Scaling in EfficientNet

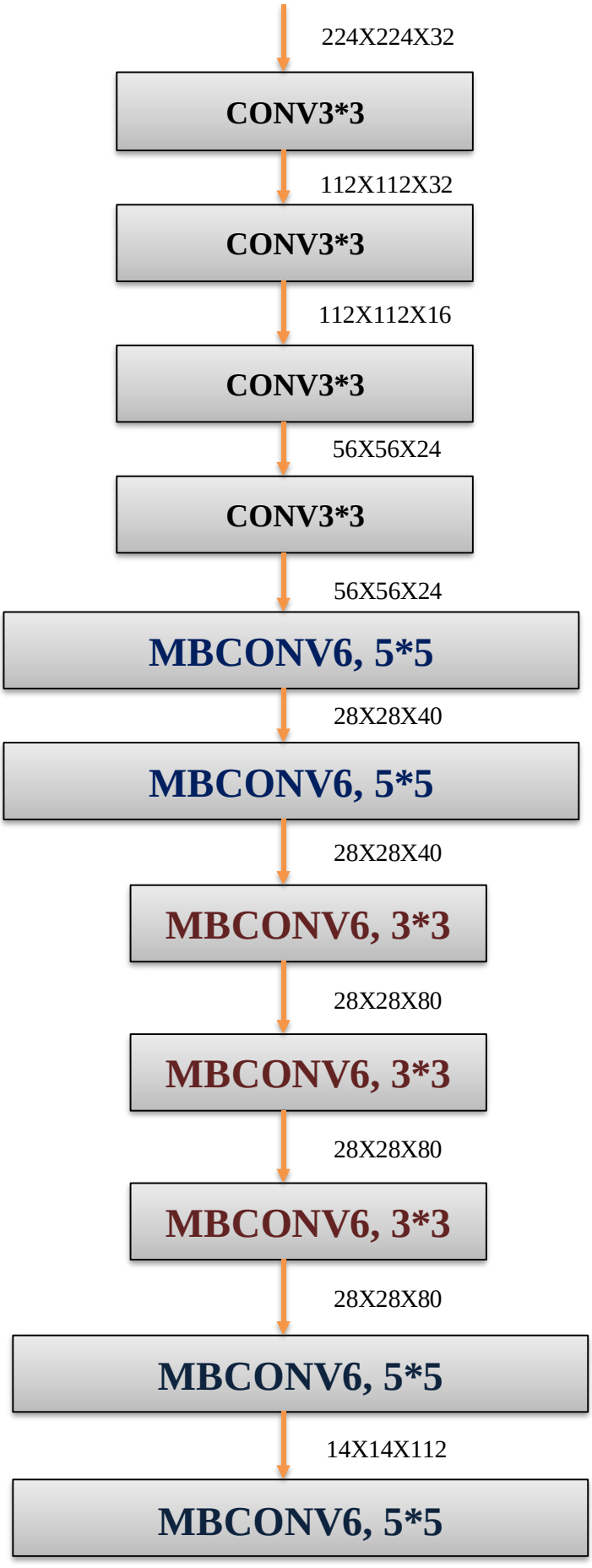
When we increase the height of deep neural network it helps to learn more and more complex features. But when we increase the number of layers, deep neural network suffers from vanishing gradient problem which in turn hinder the training process of the deep neural network. We can handle the problem of vanishing gradient by using various techniques such as batch normalization and skip connections.

When we increase the number neurons in each layer of deep neural network it helps to learn features which are more fine. When we increase the depth of the neural network it hinders the learning process of complex features which in turn decreases the accuracy of the deep neural networks. By incrementing the height and width of the image, we are providing more detailed information of an image and helps the model to model to more and more minute features and using these features tries to extract more and more finer and minute patterns.

3.2.2.2 : EFFICIENTNET ARCHITECTURE:

The baseline network of efficientNet as shown in **Figure *18*** is the most important factor on which the effective scaling of the model relies. Although the performance of the efficientnet is much better in terms of accuracy and efficiency as compared to other state of the art, changes have been made in the baseline network of the model by applying neural architecture search using AutoML MNAS framework which has resulted in better accuracy and efficiency. The newly developed baseline network of the efficientnet have used the mobile inverted bottleneck convolution (MBConv).

The performance of the efficientnet has been really good as compared to past all other state of art, especially in the field of computer vision. In classification application the performance of efficient is best in the current senerio as it provides better accuracy and efficiency over other past state of arts. The efficient have been tested not only on the Imagenet dataset but also other datasets like CIFAR-100 (have attained accuracy of about 91.7%), Flowers (have attained accuracy 98.8%), etc.


224X224X32
CONV3*3
112X112X32
CONV3*3
112X112X16
CONV3*3
56X56X24
CONV3*3
56X56X24
MBCONV6, 5*5
28X28X40
MBCONV6, 5*5
28X28X40
MBCONV6, 3*3
28X28X80
MBCONV6, 3*3
28X28X80
MBCONV6, 3*3
28X28X80
MBCONV6, 5*5
14X14X112
MBCONV6, 5*5

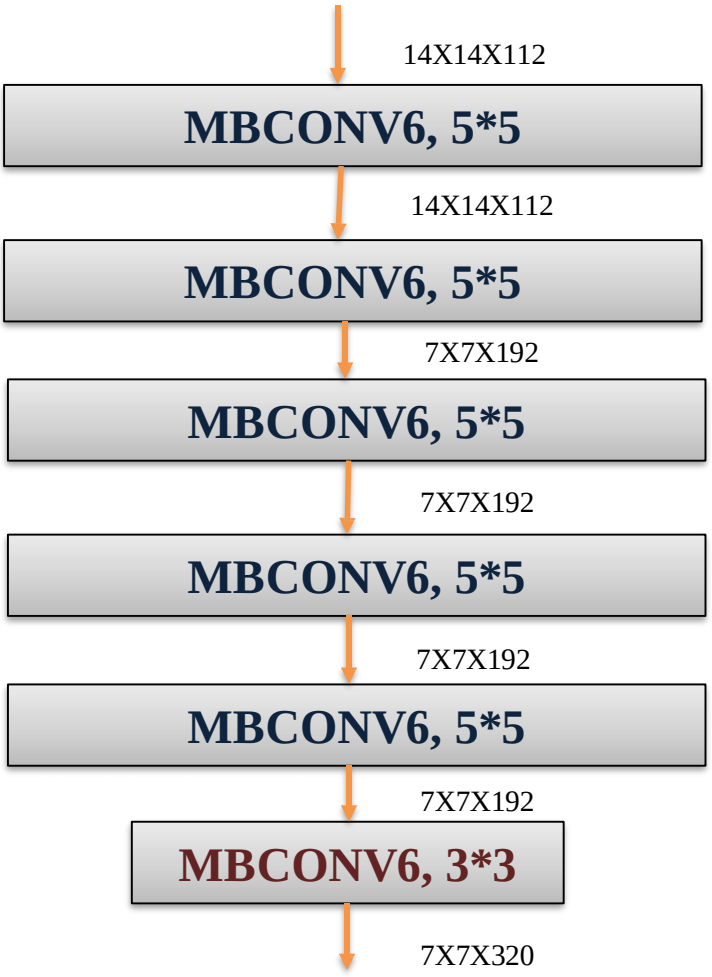


Figure 18: Architectural Design of EfficientNet

3.2.3. LSTM-RNN

Long Short Term Memory (LSTM) as demonstrated in **Figure *19*** is a kind of Recurrent Neural Network (RNN) introducedby Hochreiter and Schmidhuber (1997). They are capable of learning long term dependencies and is a widely used architecture. Like the traditional state-of-art RNN, LSTM also have chain like structure but in the chain the repeating modules are of different structures. It has four neural network layer that interacts in a very special way. The key feature of LSTM is its cell state which enables it to preserve long term dependencies depicted by the top most horizontal line in. In its cell state it can add information as well as it can discard information that it doesn't require any more.

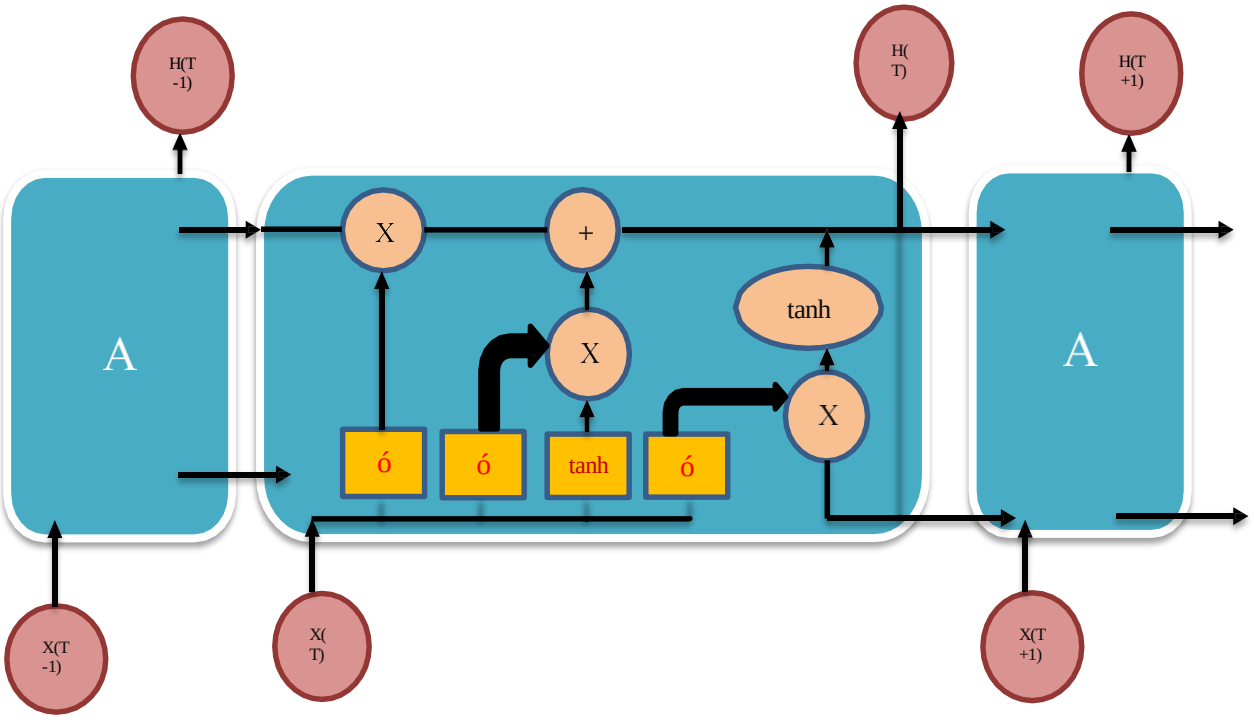


Figure 19: Long Short Term Memory Architecture

- **Sigmoid layer1**: This layer is often referred as the "forget gate layer", it decides what information we are going to remove from the cell state using the **Equation *3***. The output of this layer is a number between 0 which depicts "keep nothing" and 1 which depicts "keep everything".

$$\mathrm{F}(t) = \sigma\,(\,W(f)\ *\ [h(t-1), x(t)]\ + b(f)\,) \qquad (3.1)$$

Equation 3 : Sigmoid layer 1 of LSTM

- **Sigmoid layer2**: This layer is referred as the "input-gate-layer", it decides what data we want in the cell state using the **Equation *4***. This information is the one we want to update.

$$i(t) = \sigma\ (\ W(i) * [h(t-1), x(t)]\ + b(i)\,). \qquad (3.2)$$

Equation 4 : Sigmoid layer 2 of LSTM

- **Tanh layer:** This layer results in creating a vector of new candidate values C(t) as depicted in **Equation *5*** and **Equation *6*,** these values can than also be added to the cell state. After this we will add the output of sigmoid layer2 and tanh layer to update the cell state.

$$C(t) = \tanh(W(c) * [h(t-1), x(t)] + b(c)). \quad (3.3)$$

Equation 5 : Tanh layer of LSTM

$$C(T) = F(t) * C(t-1) + i(t) * C(t). \quad (3.4)$$

Equation 6 :Tanh layer of LSTM

- **Sigmoid layer 3**: This layer decides which part of the cell state so far we want to provide as output as depicted in **Equation *7*.** Then, the updated cell state is passed through tanh in order to pass or provide values between -1 and 1 and then we finally multiply the final outputs of this layer with the output of the sigmoid layer 3.

$$O(t) = \sigma(W(o) * [h(t-1), x(t)] +$$

$$b(o))\, h(t) = O(t) * \tanh\,(C(t)) \quad (3.5)$$

Equation 7 : Sigmoid layer 3 of LSTM

CHAPTER 4

IMPLEMENTATION

Social media networks, mobile phones, YouTube and other video portals have made creating, manipulating and propagating digital videos much easier. Moreover realistic fake videos can be created easily with minimum usage of manual editing tool with the help of Generative Adversarial Networks [16], [17]. We have implemented a hybrid model (CNN + LSTM) for fetching both spatial features and temporal knowledge to differentiate between the close and open eye state. We have trained our model in two steps:

- Firstly we have trained ResNet50-based CNN architecture on a labelled dataset of eye regions to make it learn the difference between the open and close state of eye. We have used ADAM optimizer and Sparse categorical entropy loss function with dropout in fully connected layers to avoid overfitting problem. We trained our ResNet50-CNN model on an open sourced RT-BENE dataset, which is widely used for gaze estimation and blink estimation. We have used input size of 64*64 and have taken batch size as 64 and finally trained our model for 50 epochs.

- Secondly using the weights of trained ResNet50-based CNN model, we extracted features from input eye regions images (obtained after detecting, cropping and aligning the face areas of video frames). As a result CNN model converts the input images into discriminative features which are then passed through LSTM-RNN. LSTM is mostly used for sequence learning, avoiding the gradient vanishing problem and preserving long term dependencies. We have removed the last fully connected layers from the ResNet50-CNN model and randomly selected a sequence of images of length between 15-20 images, which contains variety of temporal consecutive eye images with atleast one blink occurred. We have fixed the ResNet50 layer parameters and have only trained the LSTM cells and fully connected layers. For training LSTM cells and fully connected-layers we have used DeeperForensics-1.0 dataset which is one of the largest face forgery detection video dataset. We have used ADAM optimizer and have taken batch size as 32 and trained the trained the model for 100 epochs.

4.1. DATASET USED:

4.1.1. RT-BENE DATASET [18]:

It is one of the largest open-sourced dataset shown in **Figure *20***, which is widely used for gaze estimation and blink detection. This dataset is available with annotation of the eye-openness of more than 200,000 eye images, in which there are more than 10,000 images with closed eyes. This dataset is used by various baseline methods that allows blink detection using state-of-art Convolutional Neural Networks and the baseline methods have performed well in terms of precision and recall. This dataset is beneficial for both gaze estimation and blink detection methods.

Figure 20: RT-BENE Dataset.

4.1.2. DEEPERFORENCIS-1.0 [19]:

It is one of the largest face forgery detection dataset as shown in **Figure *21***. It consists of 60,000 videos, which in total have 17.6 million frames. It this dataset we have 48,475 source videos and 11,000 manipulated videos. The source videos have been collected on 100 number of paid actors from 26 different countries. Tampered videos have been generated

using recent proposed face swapping methods, DF-VAE. To ensure higher diversity seven types of real world perturbations at five intensity levels have been employed.

Figure 21: DeeperForensics-1.0 Dataset.

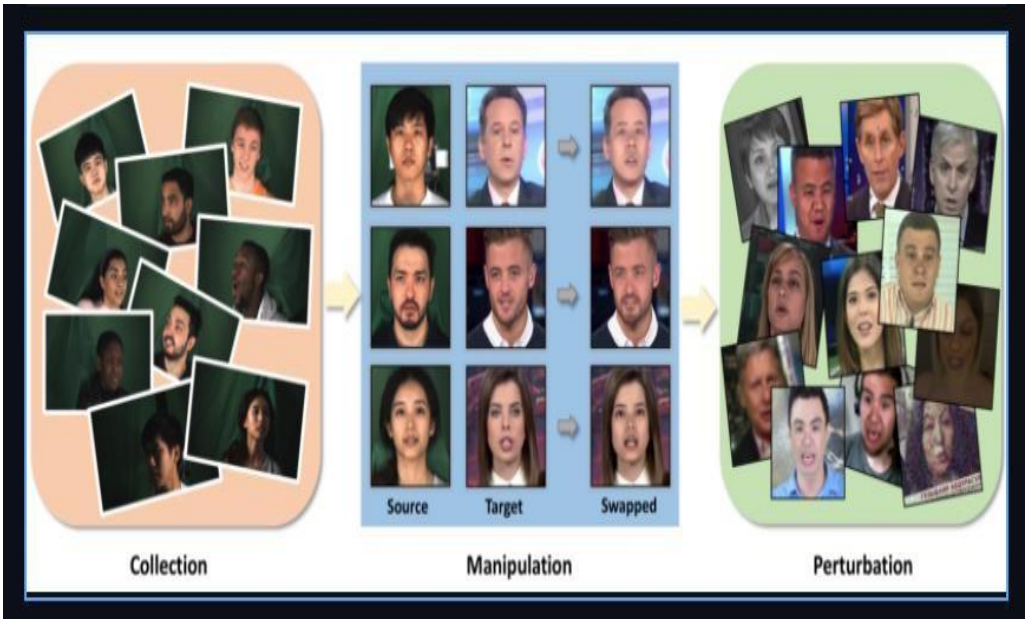


Figure 22: Collection , manipulation and perbutation of DeeperForensics-1.0

4.2. DATA PREPROCESSING:

- Firstly, we have extracted frames from video's.
- Then detected face area in each frame face detector of DLIB.
- Then fetched the facial landmarks using shape-predictor-68-face-landmark detector.
- Then performed face alignment to align facial regions to a uniform coordinate space. Performed 2D face alignment using landmark based face alignment algorithms for the transformed face to be

    1. In the center of image.
    2. Rotated to make eyes lie on the horizontal plane.
    3. Scaled to a similar size.

In the testing phase when we provide the source video as the input, after performing all pre-processing steps, further we have used state-of-art DeepFake algorithm to generate fake face videos.

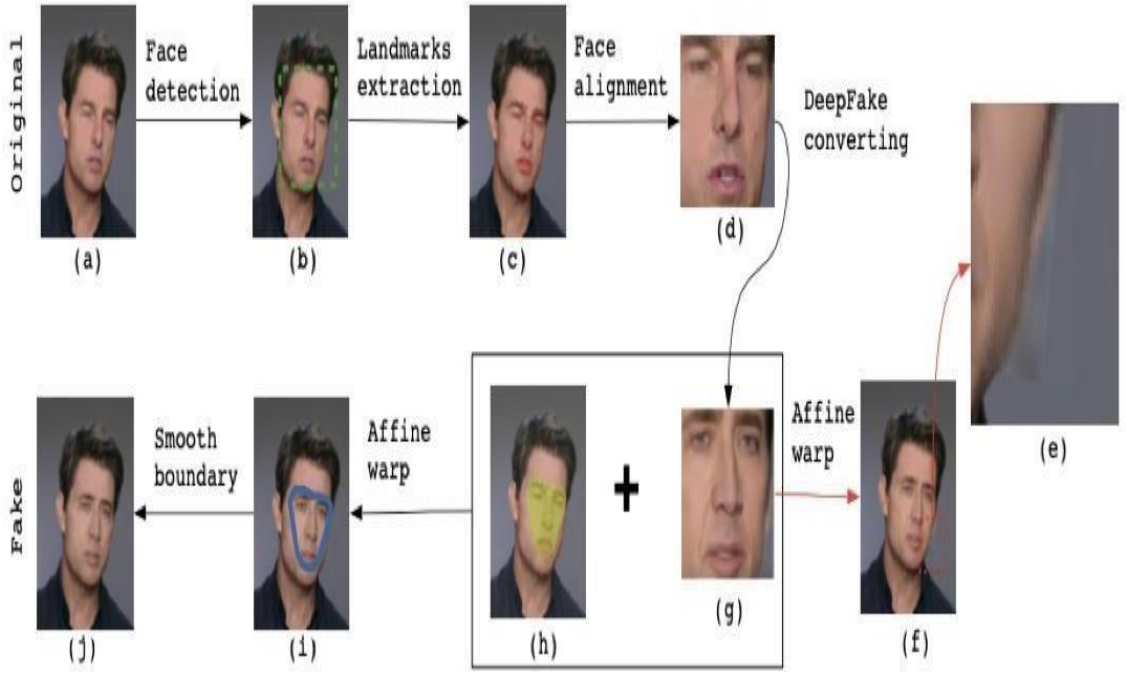


Figure 23: Overview of fake face generation pipeline

CHAPTER 5

EXPERIMENTAL RESULTS AND ANALYSIS:

5.1 TRAINING RESULT:

The dataset resulted in training accuracy of ~95% and validation accuracy of ~92.46% in ResNet50-CNN model due to its small size. But We have also used other models (each model trained for 50 epochs) or comparable results. The accuracy and loss evolution curves are shown in figure . EfficientNet Model performed best among all the algorithms. Table summarizes the results obtained.

- 85% training accuracy and 83% validation accuracy in VGG-16 Model

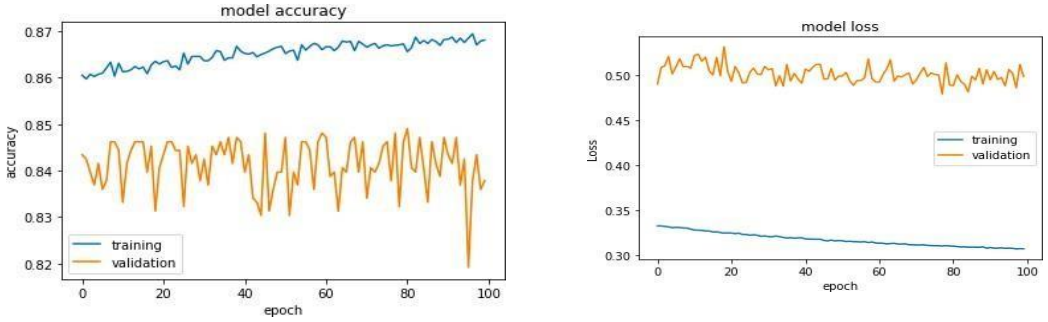


Figure 24: VGG-16 model accuracy and loss graphs.

- 86% training accuracy and 84% validation accuracy in VGG-19 Model

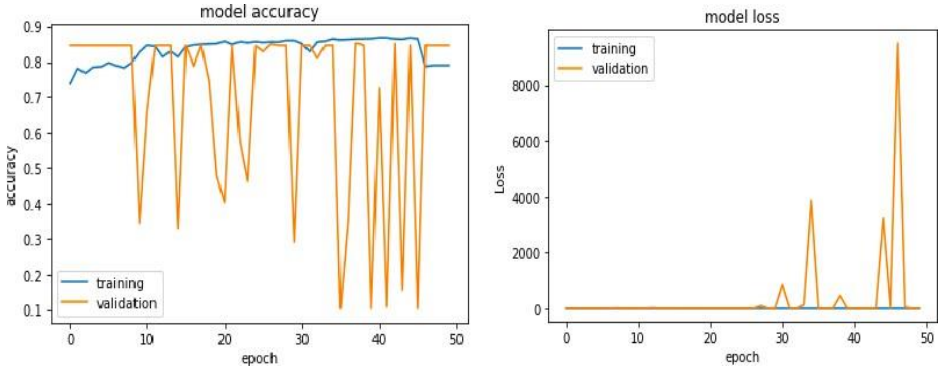


Figure 25: VGG-19 model accuracy and loss graphs.

- 83% training accuracy and 92% validation accuracy in ResNet50 Model

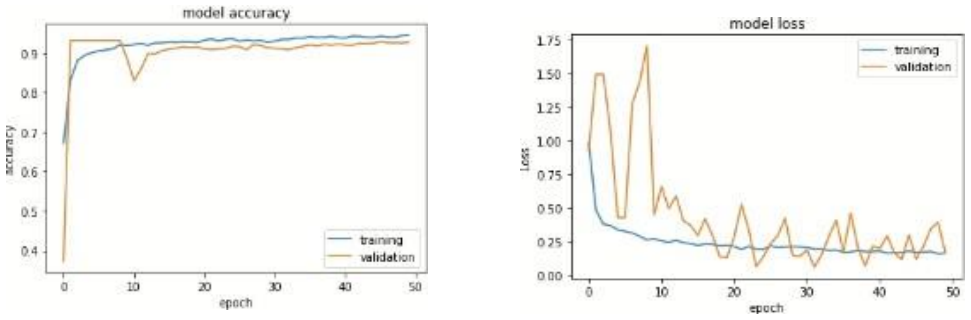


Figure 26: ResNet50 accuracy and loss graphs.

- 83% training accuracy and 93% validation accuracy in EfficientNet Model

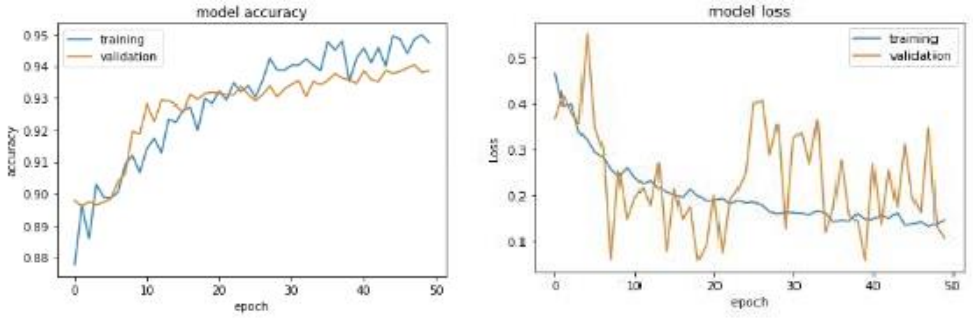


Figure 27: EfficientNet accuracy and loss graphs.

5.1. TESTING RESULT:

For the testing purpose, We are providing the input a video source then we are generating a fake video using deep fake algorithm. So for detection of fake video, we are taking rate of blinking into consideration. So in original video, the rate of blinking will be much less in comparison to in fake video. The same has been demonstrated in the figure 15.

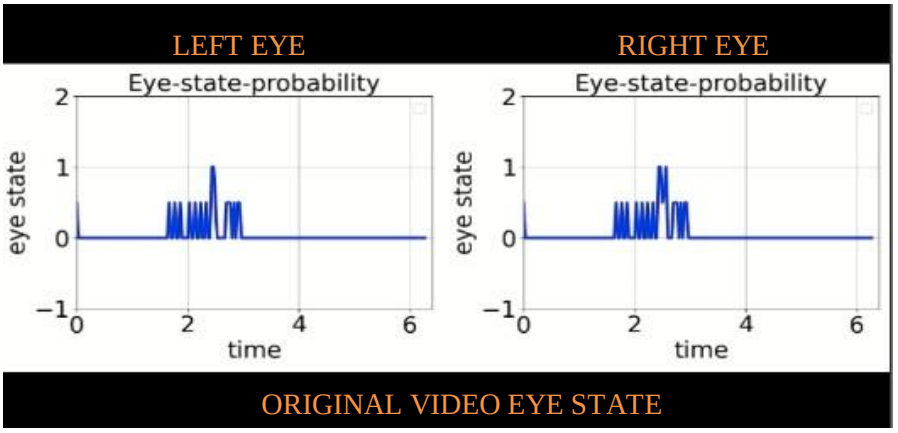


Figure 28: Left and right eye probability in original video

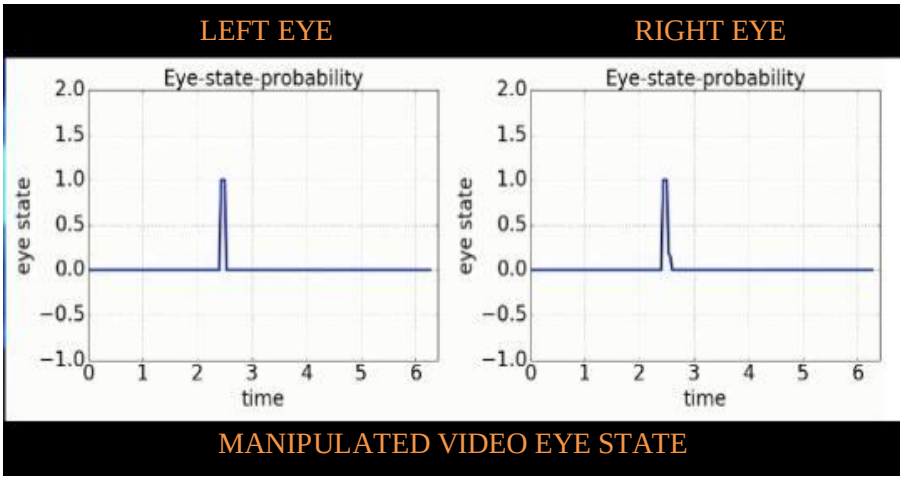


Figure 29: Left and right eye probability in manipulated video

COMPARISON BETWEEN DIFFERENT MODELS:

On Comparison the accuracy of all the models it was found that the performance of EfficientNet is better, which has attained an accuracy of about 93%. Further to test the performance of our model we have given a video as an input to our trained model which will first generate the manipulated video. Then both the original and manipulated video are being passed through the hybrid model which in turn will generate the graph representing rate of eye blinking of both left and right eyes separately described in below **Table *1***. In order to detect blinking we have passed the 30 frames consecutively, then LSTM will detect the blinking using temporal information. It has been observed that the rate of eye blinking helps a lot in detecting whether the video is fake or real.

Table 1: Training and Validation Accuracy Comparison

Model		Accuracy
VGG16-LSTM	TRAINING VALIDATION	85% 83%
VGG19-LSTM	TRAINING VALIDATION	86% 84%
RESNET50-LSTM	TRAINING VALIDATION	83% 92%
EFFICIENTNET-LSTM	TRAINING VALIDATION	83% 93%

It has been seen that normally people 34.1 times in a minute but when a manipulated video is created using the available deepfake algorithm, it was found the blinking rate was found much less of about only 3.4 times a minute. The below table ***Table 2*** shows that we have taken the input video of average length 10 seconds and have taken 30 frames per second to detect the blinking pattern in both original and manipulated video.

Table 2: Rate of Eye Blinking in source and fake video

VIDEO	**Average Video length**	**FPS**	**Rate Of Blinks**
Original	10 seconds	30	34.1/min
Fake	10 seconds	30	3.4/min

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

From the work done above we can conclude that we have successfully trained our ResNet50 -CNN model on a real-world RT-BENE dataset. We have used different CNN architectures (VGG-16, VGG-19, ResNet50) for comparable results. We have trained these models for 50 epochs for classification and obtained their accuracy scores. Our models do the binary classification to detect close and open state of eye in images/frames. We are currently focusing on tuning the parameters used in these models to increase their classification accuracies and get rid of overfitting that I am facing in VGG-16 and VGG-19 models. We will further merge the ResNet50-CNN and LSTM-RNN models to distinguish open and close eye states considering pervious temporal knowledge. We focus on coming up with an efficient classification model based on CNNs with fine-tuned hyperparameters providing greater accuracies and better classification.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015, [Online]. Available: http://arxiv.org/abs/1512.03385.

[2] W. Shen, Y. Jia, and Y. Wu, "3D Shape Reconstruction from Images in the Frequency Domain."

[3] J. Donahue *et al.*, "Long-term Recurrent Convolutional Networks for Visual Recognition and Description."

[4] S. Agarwal, H. Farid, O. Fried, and M. Agrawala, "Detecting Deep-Fake Videos from Phoneme-Viseme Mismatches." [Online]. Available: www.instagram.com/bill_posters_uk.

[5] S. Fernandes *et al.*, "Predicting Heart Rate Variations of Deepfake Videos using Neural ODE."

[6] E. Sabir, J. Cheng, A. Jaiswal, W. Abdalmageed, I. Masi, and P. Natarajan, "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos." [Online]. Available: www.adobe.com/products/photoshopfamily.html.

[7] S. Fernandes *et al.*, "Detecting Deepfake Videos using Attribution-Based Confidence Metric."

[8] Y. Li and S. Lyu, "Exposing DeepFake Videos By Detecting Face Warping Artifacts."

[9] A. M. Rodriguez, M. Koopman, A. Macarulla Rodriguez, and Z. Geradts, *Detection of Deepfake Video Manipulation*. .

[10] T. Soukupová, "Real-Time Eye Blink Detection using Facial Landmarks."

[11] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting World Leaders Against Deep Fakes."

[12] D. Cozzolino, A. Rössler, J. Thies, M. Nießner, and L. Verdoliva, "ID-Reveal: Identity-aware DeepFake Video Detection," Dec. 2020, [Online]. Available: http://arxiv.org/abs/2012.02512.

[13] Y. Li, M.-C. Chang, and S. Lyu, "In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking," Jun. 2018, [Online]. Available:http://arxiv.org/abs/1806.02877.

[14] G. Pan, L. Sun, Wu Zhaohui, and S. Lao, "Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcamera," in *Institute of Electrical and Electronics EngineersIEEE International Conference on Computer Vision 11 2007.10.14-21 Rio de JaneiroICCV 11 2007.10.14-21 Rio de Janeiro*, 2007, p. undefined.

[15] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: Learning lip sync from audio," in *ACM Transactions on Graphics*, 2017, vol. 36, no. 4, doi: 10.1145/3072959.3073640.

[16] R. Wu, G. Zhang, S. Lu, and T. Chen, "Cascade EF-GAN: Progressive Facial Expression Editing With Local Focuses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5021–5030, Accessed:May 03, 2021. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Wu_Cascade_EF-GAN_Progressive_Facial_Expression_Editing_With_Local_Focuses_CVPR_2020_paper.html.

[17] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation."

[18] K. Cortacero, T. Fischer, and Y. Demiris, "RT-BENE: A Dataset and Baselines forReal-Time Blink Estimation in Natural Environments." [Online]. Available:

www.imperial.ac.uk/Personal-Robotics/.
[19] L. Jiang, R. Li, W. Wu, C. Qian, and C. C. Loy, “DeeperForensics-1.0: A Large-Scale Dataset for Real-World Face Forgery Detection,” 2020. [Online]. Available: https://github.com/EndlessSora/DeeperForensics-1.0

ANNEXURE-IV

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis Design and Development of Framework for DeepFake Video Detection using CNN and LSTM Total Pages 34 Name of the Scholar Monish Kumar Sahu
Supervisor (s)

(1) Prof. Dinesh Kumar Vishwakarma

(2)_____

(3)_____

Department INFORMATION TECHNOLOGY, DELHI TECHNOLOGICAL UNIVERSITY

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: Turnitin Similarity Index: 11% , Total Word Count: 7725

Date: 30/05/2024

Candidate's Signature **Signature of Supervisor(s)**

11% Overall Similarity

Top sources found in the following databases:

- 7% Internet database
- Crossref database
- 6% Submitted Works database
- 7% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Helena Liz-López, Mamadou Keita, Abdelmalik Taleb-Ahmed, Abdenou... Crossref	<1%
2	arxiv.org Internet	<1%
3	Jitong Ma, Shih-Chun Lin, Hongjie Gao, Tianshuang Qiu. "Automatic M... Crossref	<1%
4	mdpi.com Internet	<1%
5	viso.ai Internet	<1%
6	cs.albany.edu Internet	<1%
7	atrium.lib.uoguelph.ca Internet	<1%
8	export.arxiv.org Internet	<1%

9	Fachhochschule Salzburg GmbH on 2020-12-03 Submitted works	<1%
10	University of Southern Mississippi on 2019-10-07 Submitted works	<1%
11	paperswithcode.com Internet	<1%
12	fastercapital.com Internet	<1%
13	downloads.hindawi.com Internet	<1%
14	techscience.com Internet	<1%
15	Kingston University on 2024-04-15 Submitted works	<1%
16	B. Shamna, C.P. Maheswaran. "Cardiac Affliction Detection Using Impr... Crossref	<1%
17	University of Cape Town on 2020-11-12 Submitted works	<1%
18	University of Edinburgh on 2017-08-18 Submitted works	<1%
19	Abdurrahman Pektaş, Tankut Acarman. "A deep learning method to det... Crossref	<1%
20	University of Sydney on 2023-11-03 Submitted works	<1%

21	University of Warwick on 2020-05-10 Submitted works	<1%
22	University of Warwick on 2020-09-17 Submitted works	<1%
23	Yongtai Pan, Yankun Bi, Chuan Zhang, Chao Yu, Zekui Li, Xi Chen. "Fee... Crossref	<1%
24	"Chinese Computational Linguistics and Natural Language Processing ... Crossref	<1%
25	Ankit Yadav, Dinesh Kumar Vishwakarma. "Datasets, clues and state-o... Crossref	<1%
26	Gonzalo de la Cruz, Madalena Lira, Oscar Luaces, Beatriz Remeseiro. "... Crossref	<1%
27	aircconline.com Internet	<1%
28	dokumen.pub Internet	<1%
29	peerj.com Internet	<1%
30	research.google Internet	<1%
31	uu.diva-portal.org Internet	<1%
32	Kevin Cortacero, Tobias Fischer, Yiannis Demiris. "RT-BENE: A Dataset ... Crossref	<1%

33	Mayur Bhargab Bora, Dinthisrang Daimary, Khwairakpam Amitab, Debd... Crossref	<1%
34	Samuel Henrique Silva, Mazal Bethany, Alexis Megan Votto, Ian Henry ... Crossref	<1%
35	Turun yliopisto on 2019-10-20 Submitted works	<1%
36	de Abreu, João Nuno Cardoso Gonçalves. "Development of DNA Seque... Publication	<1%
37	Liverpool John Moores University on 2020-04-18 Submitted works	<1%
38	Mostafa Mohammadian, Kyri Baker, My H. Dinh, Ferdinando Fioretto. "... Crossref	<1%
39	Sravya Yepuri, Ashapurna Marandi. "Classification of Blood Cell Data u... Crossref	<1%
40	Steven Fernandes, Sunny Raj, Eddy Ortiz, Iustina Vintila, Margaret Salte... Crossref	<1%
41	The University of Manchester on 2019-05-06 Submitted works	<1%
42	Universidad Carlos III de Madrid on 2021-09-09 Submitted works	<1%
43	University of Nottingham on 2023-04-17 Submitted works	<1%
44	University of Sheffield on 2018-05-07 Submitted works	<1%

45	Yashas Hariprasad, K. J. Latesh Kumar, L. Suraj, S. S. Iyengar. "Chapter... Crossref	<1%
46	openaccess.thecvf.com Internet	<1%

e-Receipt for State Bank Collect Payment

REGISTRAR, DTU (RECEIPT A/C)

BAWANA ROAD, SHAHABAD DAULATPUR, , DELHI-110042
Date: 31-May-2024

SBCollect Reference Number :	DUM6896966
Category :	Miscellaneous Fees from students
Amount :	₹2000
University Roll No :	2K20/ISY/12
Name of the student :	Nikita Dagar
Academic Year :	2023-2024
Branch Course :	MTech
Type/Name of fee :	Others if any
Remarks if any :	Examination Fee
Mobile No. of the student :	8800341650
Fee Amount :	2000
Transaction charge :	0.00
Total Amount (In Figures) :	2,000.00
Total Amount (In words) :	Rupees Two Thousand Only
Remarks :	Thesis Submission Fee
Notification 1:	Late Registration fee Rs.50 per day, Hostel Room Rent for internship Rs.1000 per month, Hostel Cooler Rent Rs.1000 per year, I card Rs.200, Character certificate Rs.200, Migration certificate Rs.200, Bonafide certificate Rs.200, Special certificate Rs.500, Provisional certificate Rs.200, Duplicate Mark sheet Rs.500, Training Diary Rs.70
Notification 2:	Fee Structure Rs.200, Admit Card Rs.50. Transcript fee and other fee rates has to be confirmed from the Academic Cell prior to remit the fees online by the student.