DETECTING DEEPFAKES USING HYBRID CNN-RNN MODEL

A Dissertation submitted in partial fulfilment of the requirement for the

Award of degree of

MASTER OF TECHNOLOGY

IN

INFORMATION SYSTEMS

Submitted By:

ANKIT SONI

(2K20/ISY/02)

Under the esteemed guidance of

Dr. Ritu Agarwal

Assistant Professor

Department of Information Technology

Delhi Technological University

Bawana Road, Delhi-110042

2020-2022

CERTIFICATE

This is to certify that Mr. Ankit Soni (2K20/ISY/02) has completed the major project "Detecting Deepfakes Using Hybrid CNN-RNN Model" as part of the Master of Technology degree in Information Systems at Delhi Technological University.

During the academic session 2020-2022, the major project is a genuine piece of work that was carried out and finished under my supervision and guidance. This report's content has not been submitted anywhere else for the granting of any other degree.

(Major Project Guide)
Dr. Ritu Agarwal
Assistant Professor
Department of Information Technology
Delhi Technological University
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I would like to thank my major project guide Dr. Ritu Agarwal, Assistant Professor, IT Department, Delhi Technological University, for her invaluable assistance and direction in completing this major project. It gives me great pleasure to express my heartfelt gratitude to my esteemed mentor for her constructive critique and understanding, without which the project would not have taken the shape it has.

I respectfully express my thanks to the other faculty members in this department for their invaluable assistance and time whenever it was needed.

Ankit Soni

Ankit Soni
Roll No. 2K20/ISY/02
M.Tech (Information Systems)
E-mail: soniankit009@gmail.com

ABSTRACT

We are living in the world of digital media and are connected to various types of digital media contents present in form of images and videos. Our lives are surrounded by digital contents and thus originality of content is very important. In the recent times, there is a huge emergence of deep learning-based tools that are used to create believable manipulated media known as Deepfakes. These are realistic fake media, that can cause threat to reputation, privacy and can even prove to be a serious threat to public security. These can even be used to create political distress, spread fake terrorism or for blackmailing anyone. As with growing technology, the tampered media getting generated are way more realistic that it can even bluff the human eyes. Hence, we need better deepfake detection algorithms for efficiently detect deepfakes. The proposed system that has been presented is based on a combination of CNN followed by RNN. The CNN model deployed here is SE-ResNeXt-101. The system proposed uses the CNN model SE-ResNeXt-101 model for extraction of feature vectors from the videos and further these feature vectors are utilized to train the RNN model which is LSTM model for classification of videos as Real or Deepfake. We evaluate our method on the dataset made by collecting huge number of videos from various distributed sources. We demonstrate how a simple architecture can be used to attain competitive results.

Table of Contents

List of Figures

1 Chapter 1- INTRODUCTION

1.1 INTRODUCTION

With the continuous development of multimedia content over the years, a huge quantity of pictures and videos are getting generated. Meanwhile, forgery of multimedia has also been prevalent leading to malicious crimes. The examples of forgery like swapping of faces can be drawn from 19th century. During those times, the photos were made by hand. Further, with the coming up of digital age, the methods like Photoshop and other image processing software came into existence. After the proposal of GANs, the quality of the forged videos got so much better that it can even fool naked human eyes. The technique "Deepfake" is one such powerful technique of the modern era which uses deep learning in order to forge multimedia.

The term "Deepfake" has been formed using a combination of the two words "Deep Learning" and "Fake"[1]. These are the images or videos generated using deep learning-based techniques such that the face of a person in the video or image being applied in place of the other person's face in a realistic way. The most alarming part is that it can be created by anyone who is having no knowledge of creating a deepfake media. Nowadays, we can run any such programs which can create deepfake media by swapping our faces with others. Applications such as Snapchat, FaceApp have that ability to face swap and can create such media. Also, they can change the age and appearance of a person. An app called as DeepNude which pose more disturbing threat as it can create nude image of a target person. Similarly, the Chinese app Zao went viral as people with no skills or knowledge of deepfakes can swap their faces onto bodies of movie stars or can even insert their images in movies or TV-clips.

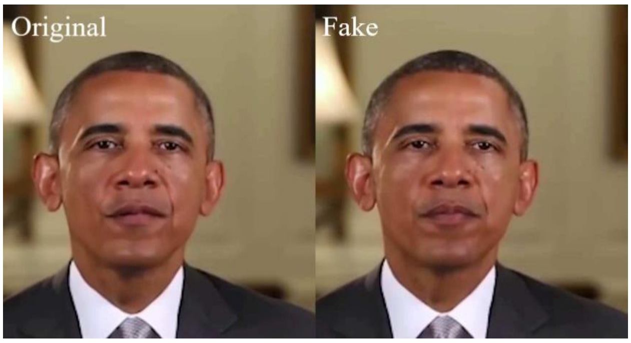


Figure 1.1 Left image is snapshot of original video and right image is a deepfake video[2].

The deepfakes was born way back in 2017, when a Reddit user having same username as that of the celebrities posted modified pornographic videos on that site using the deepfakes technology[3]. Those clips were containing face images of celebrities like Gal Gadot and Taylor Swift which were swapped onto the faces of porn performers. It is observed that a staggering 96% of the created deepfake content are pornographic and among them, 99% of them having face images mapped over porn stars from the female celebrities.

1.2 REAL WORLD IMPLEMENTATION

Deepfakes can be used in a constructive way. A very big example we can take from film industry. There are cases when an actor or actress died during the shooting tenure where the film was still incomplete. In those cases, we can use deepfake and do face swap of the person having similar and resembling face features than that of the dead actor or actress such that the actor or actress‟ role can be revived. Also, for cases when we dub movies hiring a popular actor but the actor does not know the language in which the movie is about to me made. In cases like this, we can use deepfakes to create a realistic feel so that the regional audience can also connect with the actor and thus the language barrier can also be removed.

But most often than not, the use of deepfakes is done in a negative manner like creating fake pornography content, fake news, maligning image of a person, extortion etc. There were instances of deepfake videos made where Obama is abusing Donald Trump[4].

Jon Snow apologized for the pathetic ending sequences to GOT Final Season[5] and Mark Zuckerberg declared having “total control of billions of people’s stolen data”[6].

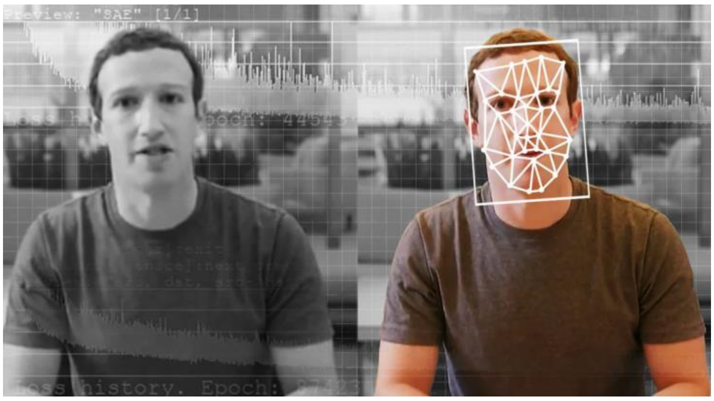


Figure 1.2 Mark Zuckerberg video manipulated using Deepfakes

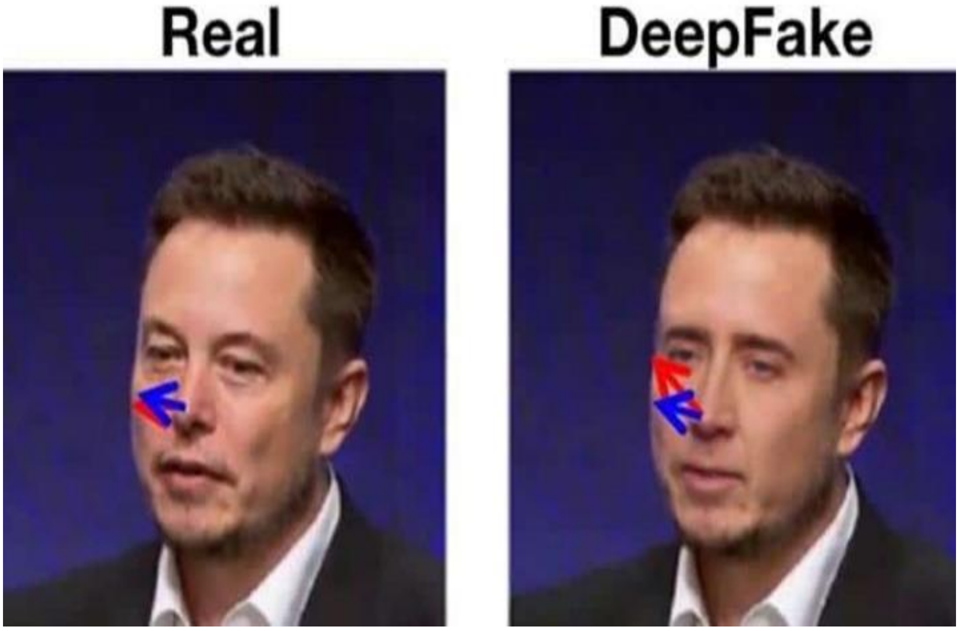


Figure 1.3 Real and Deepfake manipulation done on Elon Musk

It is observed that a staggering 96% of the created deepfake content are pornographic and among them, 99% among them having face images mapped over porn stars from the female celebrities.

An app called as DeepNude which pose more disturbing threat as it can create nude image of a target person. This is way more disturbing as it can easily tarnish image of an individual in a society as the created media seems very realistic and easy to believe. Similarly, the Chinese app Zao went viral as people with no skills or knowledge of deepfakes can swap their faces onto bodies of movie stars or can even insert themselves in movies or TV-clips[7].

The app like FaceApp can even create media content such that the age, maturity or even sex of a person can be changed drastically with such a quality that the morphed image looks very genuine and can fool anyone. Although, these applications are very creative but misuse of these applications can be a serious threat to the society. Teenagers can fake their age for any age-restricted access. Even people can use their opposite gender look for any inappropriate means.

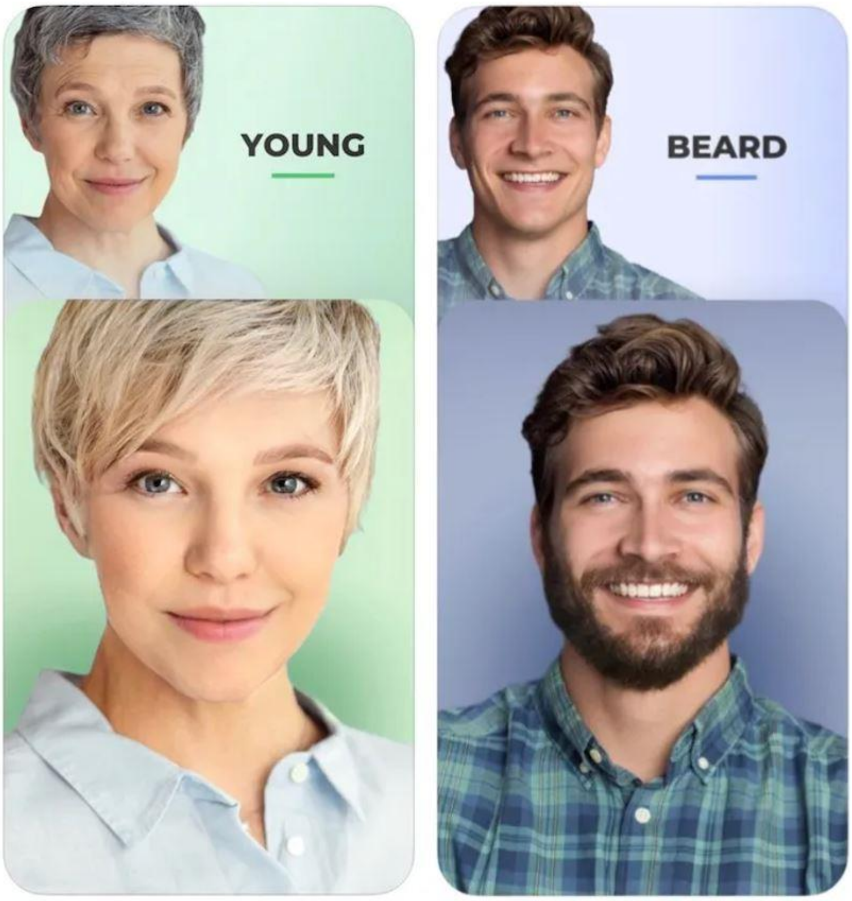


Figure 1.4 The snapshot of few of the different features of the FaceApp application[8].

1.3 RELATED WORK

1.3.1 Face Warping Artifacts Detection:

The authors Li & Lyu, described a new deep learning-based method in which the deepfake algorithms generate limited resolution images which are then warped to draw comparison with original faces present in the image/video. This leaves some distinctive artifacts which are easily detected by CNNs[9].

1.3.2 Eye Blinking Pattern Detection:

The authors Jung et al. proposed a method for detecting Deepfakes using a GAN model via an algorithm known as DeepVision to analyse changes in the blinking pattern of eyes. The blinking pattern of eyes changes from person to person since it is influenced by several factors like a person's gender or the person's age or the time of the day, degree of alertness, emotional state etc[10].

1.3.3 Capsule networks:

The authors Kim, Hyeongwoo et al. proposed an approach that makes use of a capsule network in order to identify tampered images and videos in different outlines, like detection of replay attack and detection for computer-generated videos. In this they made use of random noise for training the model which cannot be considered to a good option but in this case, the model still performed well enough in their dataset but it may fail for real-time data as the training is done using random noise. Their method was proposed to be trained on noiseless and real time datasets[11].

1.3.4 Mesoscopic features:

The Mesonet approach uses mesoscopic property of images in order to identify image manipulation. The two architectures achieved the best classification having a small number of parameters and a low degree of representation, namely Meso-4 and MesoInception-4[12].

1.3.5 Inconsistent head poses:

The authors Yang et al. suggested a new method on the basis of the observation that the deepfake contents are generated by splicing the source face image onto the target image, and by doing this, there are errors and inconsistency in the 3D head poses that are estimated from formed image/video[13].

1.3.6 Steganalysis features:

The classifier model, XceptionNet[14] is the implementation of traditional CNN model with pre-trained weights of ImageNet, which uses steganalysis features of images evaluated on FaceForensics++[15] as well as DFDC Preview datasets[16].

2 Chapter 2: BACKGROUND

2.1 NEURAL NETWORKS

Artificial Neural Networks (ANNs)[17] are a learning mechanism having a generalized structured that is inspired by the brain of a mammal. They are fundamentally utilized as ways to approximate a mapping function for a certain set of input domain to a certain set of output domain. The network consists largely of a high number of interconnected nodes, arranged layer by layer having weights designated to each connection between the nodes.

Every node calculates a linear combination when an input sample is sent through the network. All of the node inputs are used as coefficients, along with their appropriate weights. After that, the linear combination is usually put via a non-linear function which is known as an activation function. This results in the node's activation, which is its output. These computations are carried out in all nodes and layers to create the network's final output.

The basic purpose of a system like this is to discover appropriate weights in its connections such that the whole network approximates a function that translates the input domain to the output domain. This may be accomplished via supervised learning, in which we have some samples in both domains with known values. We see the outputs after allowing these samples to travel through the network. For each sample, we carry out the comparison of the network's outputs to the right values.

A loss function is then being used to determine how close the network's approximations were. The learning process is now essentially a non-convex optimization problem in which we aim to maximise the loss function in relation to the network's weights. This optimization can be done via gradient descent, in which the weights are updated by travelling in the negative direction of the loss function's gradient. Backpropagation is commonly used to compute the gradient in an efficient manner.

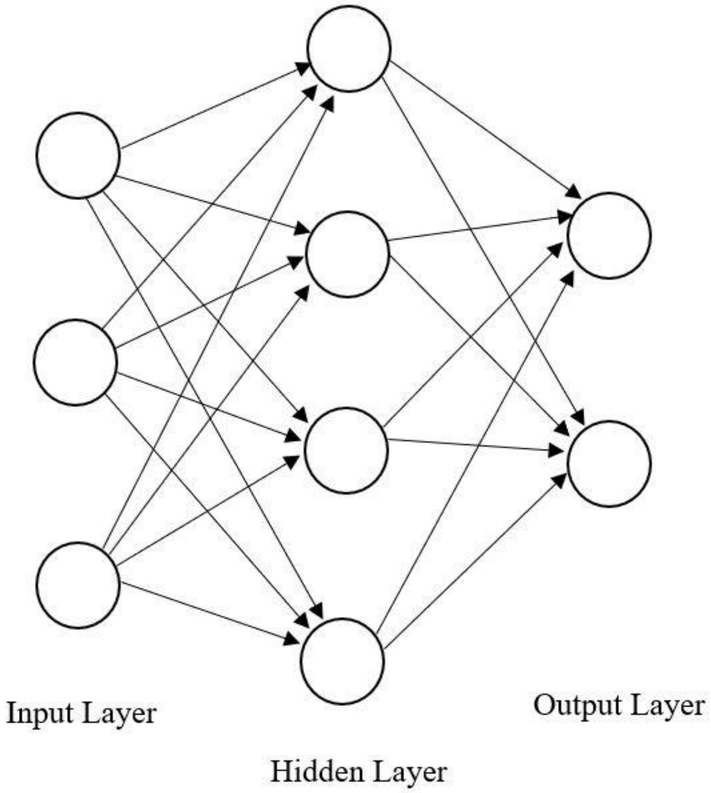


Figure 2.1 Neural Network

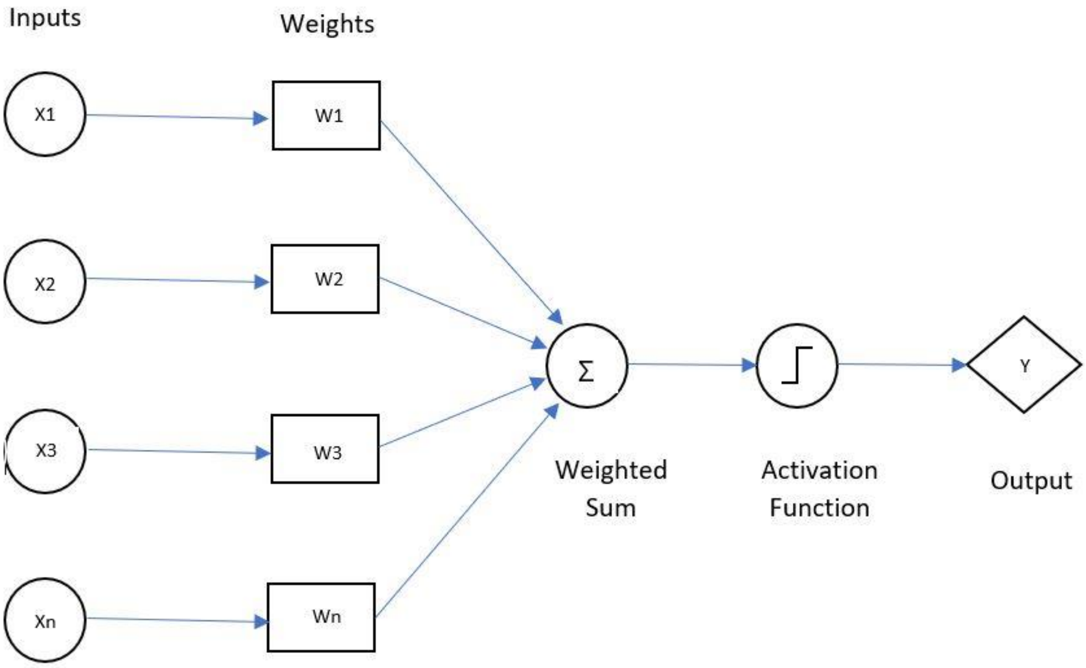


Figure 2.2 Functioning of a Neural Network Perceptron

2.2 CONVOLUTIONAL NEURAL NETWORKS (CNNs)

When the input consists of pictures or videos, convolutional neural networks (CNNs)[18] are most typically utilized. These are typically rather huge. If a network of completely linked layers were used, the number of weights in the network would skyrocket. Furthermore, neural networks have trouble translating pictures; merely shifting things around in the image might provide totally different results.

CNN solves both of these issues. To analyze the image, they utilize so-called filters, which are small windows that are moved over an image. The values of the pixels are multiplied with the learnt parameters to give an output value for each place the window is moved into, resulting in a 2D picture. When combining numerous filters in a given layer, you receive one 2D image per filter, that leads to creation of a 3D "data cube." A convolutional layer which was named after the mathematical technique required to generate the resulting value from the convolutions performed, is the layer that makes one among these cubes from a picture or another cube. These layers make up the CNN network with more layers in between them.

The different types of CNNs include LeNet-5, AlexNet, VGG-16, ResNet, InceptionNet etc.

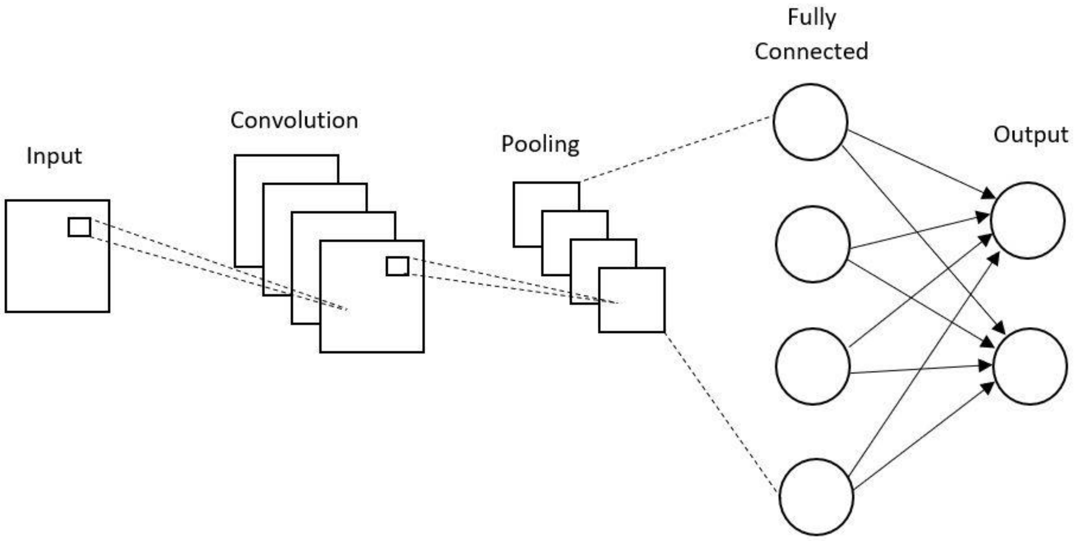


Figure 2.3 Convolutional Neural Network (CNN)

The different layers shown in the above figure:

1) Convolution Layer - A series of filters is applied to an input picture to create convolutional layers. The feature map that is created using the convolutional layer is a representation of that input image which was subjected to those filters. Convolutional

layers may be layered to construct increasingly complex models who can learn more complex characteristics from the image.

2) Pooling Layer – The pooling layers are one of a form of convolutional layer. The input's spatial size is decreased by the pooling layers, thus makes the processing of it easy and less memory consuming. Pooling also reduces the number of parameters and speeds up the training process. Pooling can be categorized into two different types: maximum pooling and average pooling. In max pooling, we use the maximum value from each feature maps which are generated, whereas average pooling makes use of the average value. After operating through convolutional layers, the pooling layers are often employed to minimize the size of the input before it is fed into a completely connected layer.
3) Fully Connected Layer - Fully-connected layers are among the most important forms of CNN layers. In this, each neuron in a completely linked layer is fully connected to every other neuron in the layer before it. Fully connected layers are usually employed at the conclusion of a CNN when the objective is to use the features learnt by the layers encountered earlier to make predictions. If we used a CNN to categorize photos, the final fully connected layer might use the information obtained by the earlier encountered layers to classify the image.

2.2.1 BENEFITS OF CNN

Convolutional Neural Networks has the following advantages:

1) They minimize computation as compared to a regular neural network.
2) Achieves commendable accuracy in problems related to image recognition.
3) Identifies the valuable features automatically without any supervision of human.
4) Uses the same knowledge across all image locations.
5) Weight Sharing.

2.2.2 LIMITATIONS OF CNN

Following are the limitations of Convolutional Neural Networks:

1) The CNN does not encode an object's position or orientation.
2) Inability in becoming spatially invariant while working with the incoming data.
3) Lots of training data is required.

2.3 RECURRENT NEURAL NETWORKS (RNNs)

Traditional neural networks are inefficient when dealing with sequential data because they include distinct input and output layers. As a result, the RNNs[19] was created to store the outcomes of the outputs generated at early stages by the internal memory. These findings are then used as inputs in the network. Pattern identification, voice and speech identification, natural language processing (NLP), and predictions related to time series can all benefit from this.

RNN features hidden layers that function as memory locations for looped layer outputs.

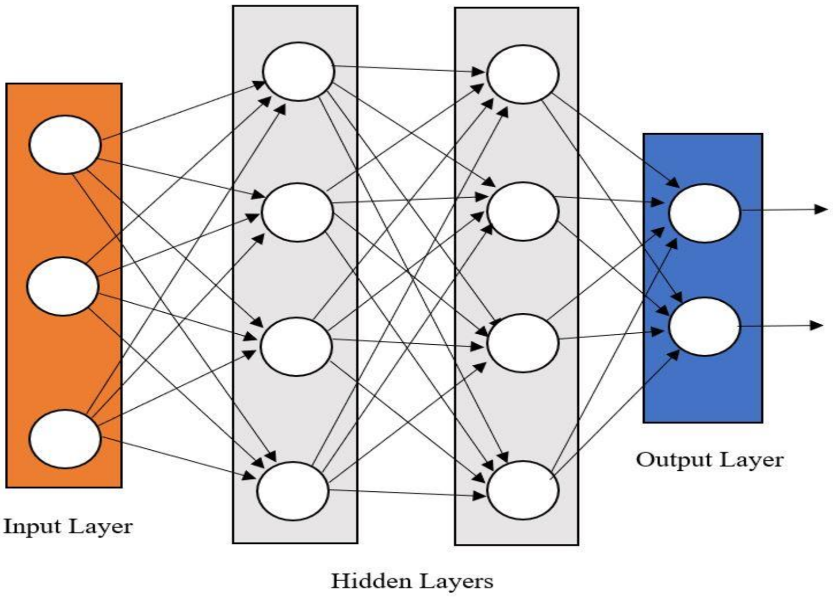


Figure 2.4 Recurrent Neural Network (RNN) Architecture

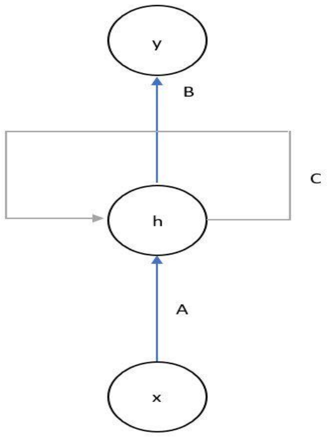


Figure 2.5 Recurrent Neural Network (RNN) Loopback Structure

In the mentioned figure,

y - Output Layer

h - Hidden Layer

x - Input Layer

A, B, C - The Network Parameters required to improvise the model’s output

2.3.1 TYPES OF RNN

The four different types of RNN that are most commonly used are:

1. One-To-One RNN:

The most basic form of RNN is the One-to-One, which has only one input and output. It is a standard neural network with set input and output sizes. Image Classification contains the One-to-One application.

2. One-to-Many RNN:

When given a single input, one-to-many RNN model produces several outputs. It accepts a set input size and produces a series of data. Music generation and image captioning are two of its uses.

3. Many-to-One RNN:

Many-to-One RNN model is used when we want only one output from a number of input units or a sequence. Sentiment Analysis is one of a example of this type of RNN. In this, for obtaining a fixed output, it takes sequential inputs.

4. Many-to-Many RNN:

Many-to-Many RNN model is a technique for creating output data series from input unit series.

This sort of RNN is further subdivided into the two groups below:

1) Equal Unit Size: Both the input and output units are the same size in this situation. The recognition of names and entities is a popular application.
2) Unequal Unit Size: In this scenario, the number of units in the inputs and outputs are unequal. Machine Translation is one of its applications.

2.3.2 BENEFITS OF RNN

Recurrent Neural Networks have the following advantages:

1) Performs sequential data processing.
2) Previous findings can be remembered and stored.
3) In the computation of new results, both the present and past outcomes are taken into consideration.
4) The model size remains constant regardless of the magnitude of the input.
5) It distributes weight to other components throughout time.

2.3.3 LIMITATIONS OF RNN

Few of the drawbacks of RNNs are listed below:

1) Because it is recurrent, the computation time is slow.
2) If you use the tanh or ReLU activation functions, you won't be able to process a long series of data.
3) In the calculation of present data, future data cannot be processed.
4) Training is difficult.
5) Exploding Gradient: An accumulation of huge gradient mistakes causes an exponential growth in model weights.
6) Vanishing Gradient: The gradients tends to become quite small so that they become inefficient to achieve noticeable changes in model weights, and the gradients vanish.

3 Chapter 3: FACIAL VIDEO MANIPULATION STRATEGIES

3.1 FACESWAP

This is an approach that makes use of computer graphics. Using the sparse detected landmarks from faces, the facial area is extracted from the media. These markers are used to ensure that the approach is compatible with a 3D template model. The target picture data is back projected using the specified model by difference minimization among the shape projected and the localized landmarks by using the input image texture data. Finally, the mentioned model which is rendered is then mixed with the image and correction for colour is done.

The algorithm does not require any prior knowledge of each actor's face or a large amount of computer capacity. When there are significant variances in illumination between the two movies, however, it suffers and relies heavily on a decent face identification algorithm.

3.2 DEEPFAKE

This is a machine learning based technique. In this, we first use facial recognition to detect the facial region, post that it uses two auto-encoders to train the model, one for the source and another for the destination.

The two autoencoders are each trained on videos of a single individual, the two persons whose identities you want to switch. These autoencoders must all use the same encoder, but the decoders can be different. In theory, this implies that the encoder concentrates on video-specific information like facial expressions and illumination while neglecting the two people's characteristics, which the decoders may restore. Typically, a face identification technique is employed ahead of time to ensure that only the faces of the people are input into the autoencoders.

The target video is supplied into the encoder after the auto-encoders have been taught, and a video is extracted using the decoder trained on the source actor rather than the target actor. As a consequence, the face of the target actor is replaced with that of the source actor, although facial emotions, lip movements, and other aspects of the video remain untouched.

3.3 FACE2FACE

One of the more well-known methods for face replication is Face2Face. The algorithm's main idea is to generate 3D models of the source and target actors' faces, then track deformations in the source actor's model and transfer them to the target actor's model.

Principal Component Analysis (PCA) is used to track deformations and parametrize the faces, rather than learning. The work is reduced to an optimisation problem using Iteratively Reweighted Least Squares when these parameters are fed into an objective function (IRLS).

After the emotions have been transferred, the mouth interior is found by scanning the target actor's video and warping the best match to the present frame. As a result, the target actor's video must be known ahead of time, but the source video may be used to change the video in real time.

3.4 NEURAL TEXTURES

This is also a machine learning based technique. It includes facial re-enactment based rendering approach using Generative adversarial networks (GANs)[20]. It uses the original media in form of videos in order to learn the neural textures of the target person in the media, together with a rendering network.

Neural textures use images of the object as input to replace the object's handcrafted characteristics with learnt ones. A viewpoint-specific texture may be created using this neural texture, a perspective, and a UV-map that relates components in the neural texture map to points in the object. It can create a picture of the item from the supplied viewpoint if this is fed into a delayed neural renderer and trained with the neural texture.

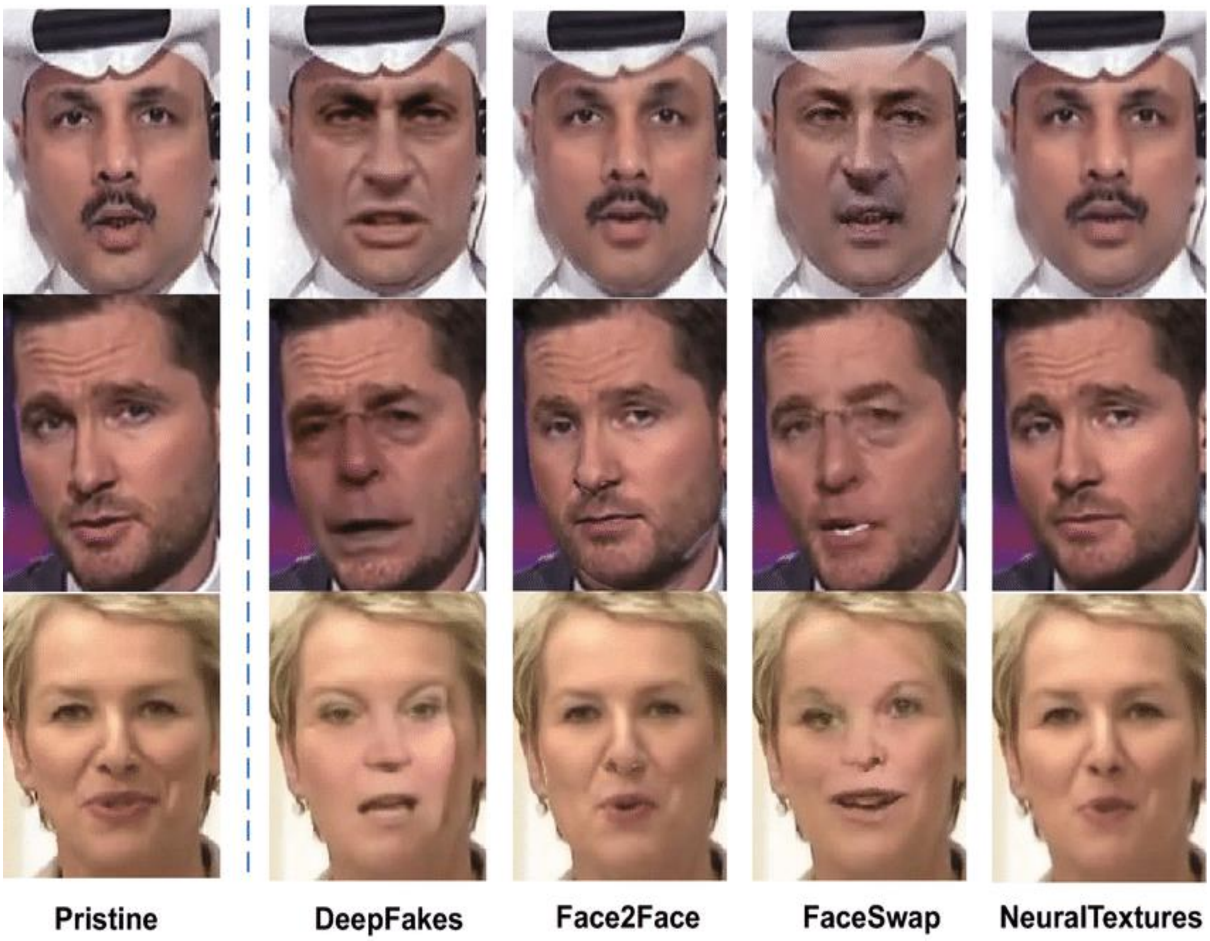


Figure 3.1 Manipulation using DeepFakes, Face2Face, FaceSwap and NeuralTextures[21]

4 Chapter 4: DEEPFAKE DETECTION METHODOLOGY

4.1 AUTOENCODERS

The Autoencoders basically contains two parts, encoder and decoder. Firstly, two encoders are trained such that they can reproduce the face images of the two persons, person A and person B with precision.

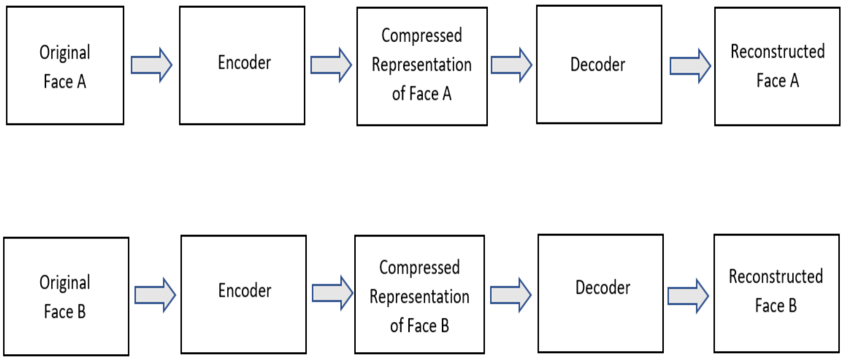


Figure 4.1 Deepfakes Auto-Encoding for training

Following this, when the two encoders are trained enough to reconstruct the face images of person A and person B, then the next step involved swapping the given two faces. This involves providing the decoder for face of person A with a compressed representation of face for person B. Decoder for person A converts this representation into the face of person A with the expressions and actions performed by person B[22].

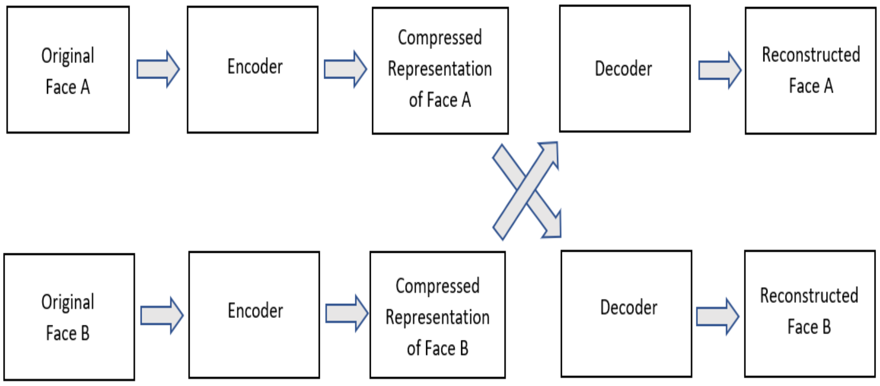


Figure 4.2 Deepfakes Auto-Encoding for actual face swapping

4.2 Generative Adversarial Networks (GANs)

GANs[20] uses the autoencoder technique to create deepfake media. The GANs contains two deep neural networks, namely a discriminator and a generator. The generator functions similar to autoencoder, but due to the fact that we also use discriminator, we achieve better results as discriminator discards bad deepfake creation while generator keeps on producing deepfake media such that it can successfully fool the discriminator. Due to this, the output content is so realistic that it can even fool naked human eyes. Some open-source projects use the same technique e.g., Faceswap-GAN.

4.2.1 GAN Architecture

The architecture of GAN model comprises of two sub-models present inside it, namely a generator model which is used for forming new instances and a discriminator model which will be making decision whether the produced instances are real or fake which were created by the generator model.

Generator Unit: A model for generating fresh believable instances from the issue domain.

Discriminator Unit: Model used for identification of samples as genuine (from the given domain) or fraudulent (generated or formed).

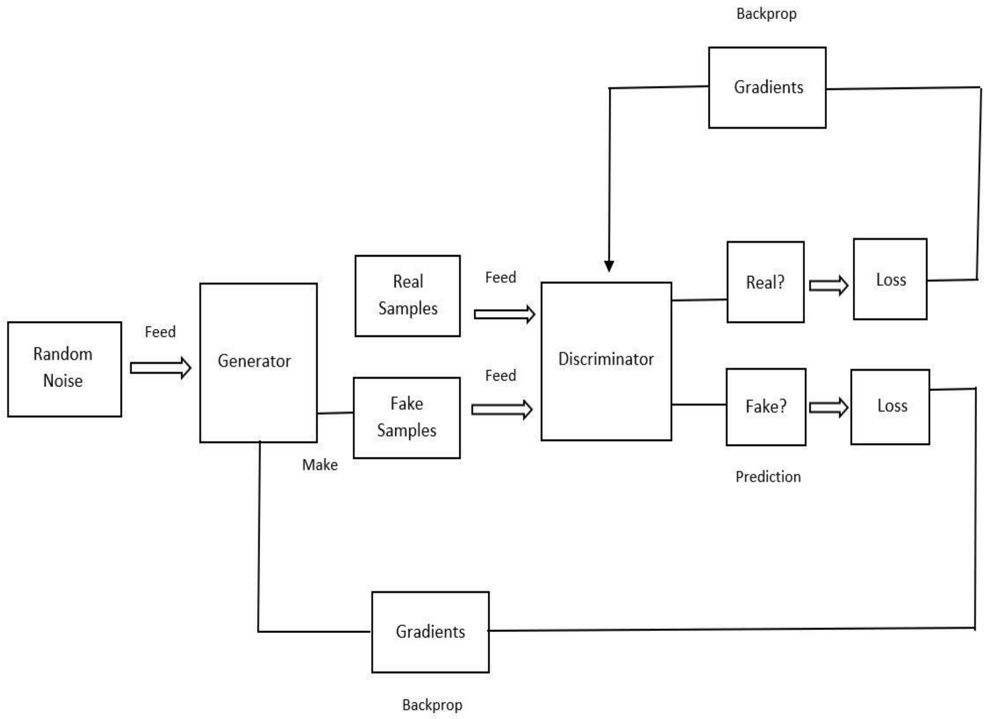


Figure 4.3 Generative Adversarial Network (GAN) framework

4.2.2 GAN as a two-player game

Although generative modelling is one of a unsupervised learning issue, it is presented as a supervised learning problem during training.

Both the discriminator and generator models are trained parallelly. The generator creates a chunk of generated samples which are then delivered to the discriminator model altogether with authentic domain occurrences for classification this as true or false. As a result, the two models are in rivalry, hostile in game theory, tries to achieve a zero-sum during this game.

The discriminator model when effectively differentiate between actual and false samples generated, it either gets rewarded for that or no changes to the model parameters are done, but the generator model is fined for that and then update to substantial model parameter is done.

When the generator model deceives the discriminator model, then it is either rewarded or the model parameters are not changed, while the discriminator is fined for that and then updating to its model parameters is done.

At a point in time, the generator always creates flawless clones of the input domain such that the discriminator becomes unable to detect the difference between real or fake, predicting "unsure" in every case. This is only an idealistic scenario; we do not need to reach to this stage in order to arrive at a functional generator model.

5 Chapter 5: DATASETS USED

5.1 FACEFORENSICS++

Faceforensics++[15] dataset contains a total of 1000 original youtube video sequences and 4000 manipulated video sequences from each manipulation technique, i.e., 1000 videos using Deepfakes, 1000 videos using Face2Face, 1000 videos using FaceSwap and 1000 videos using NeuralTextures.

5.2 CELEB-DF

Celeb-DF v2[23] dataset consists of 590 real and 5639 deepfake videos along with 300 additional real videos taken from Youtube. The average length of all the videos is around 13 seconds having 30 frame-per-second(fps) frame rate. The real videos are taken from publicly available youtube videos, consisting of 59 celebrity interviews with varied range of people hailing from different age, sex and ethnic groups.

5.3 FINAL DATASET

We have taken the following number of videos from the datasets-

- Celeb-DF v2: 539 Original Videos (239 real + 300 additional youtube real videos) and 1540 Deepfake Videos

- Faceforensics++: 250 Original Youtube Videos and 1000 Deepfake videos (250 deepfakes, 250 Face2Face, 250 Faceshifter, 250 FaceSwap, 250 NeuralTextures)

Our Dataset Consists of 789 Original Videos and 2790 Deepfake Videos.

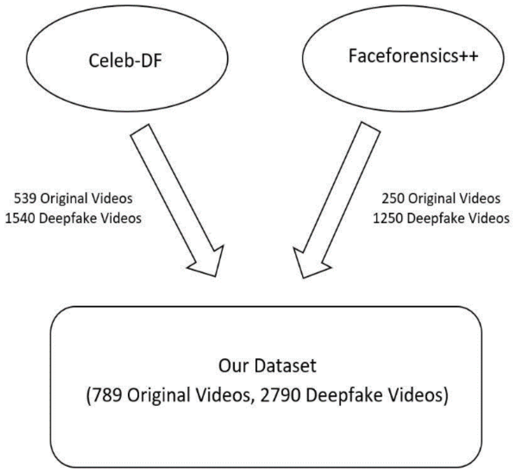


Figure 5.1 Collected Dataset from Sources

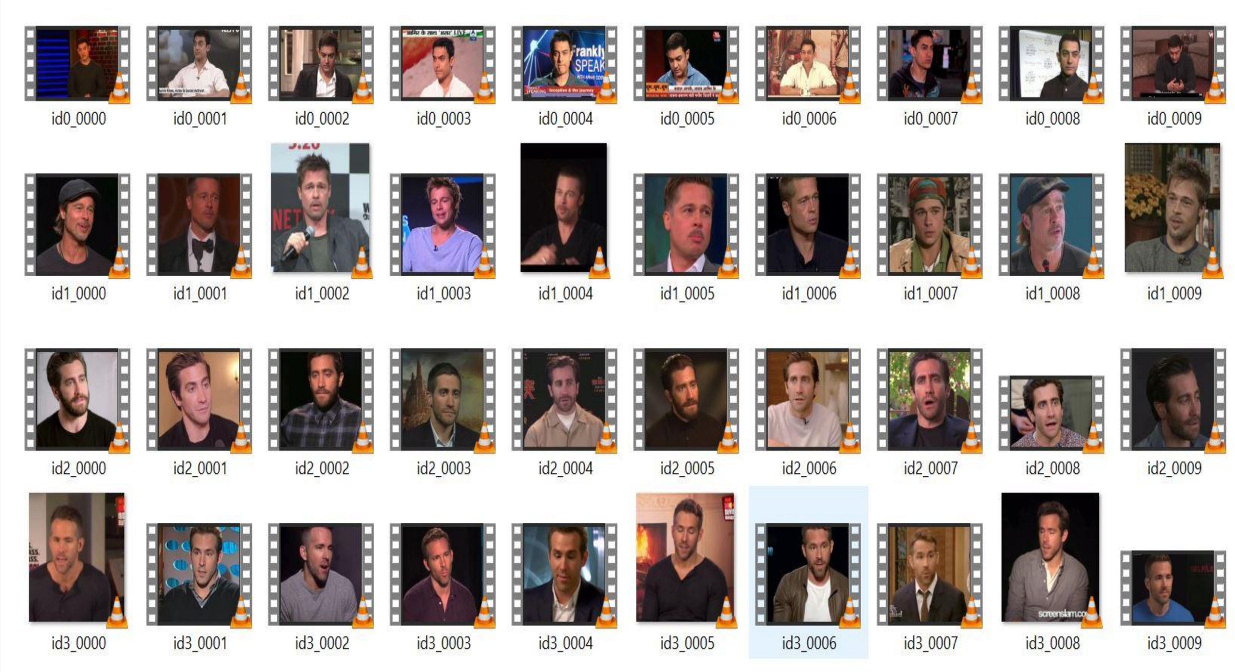


Figure 5.2 Final Dataset Snapshot-1

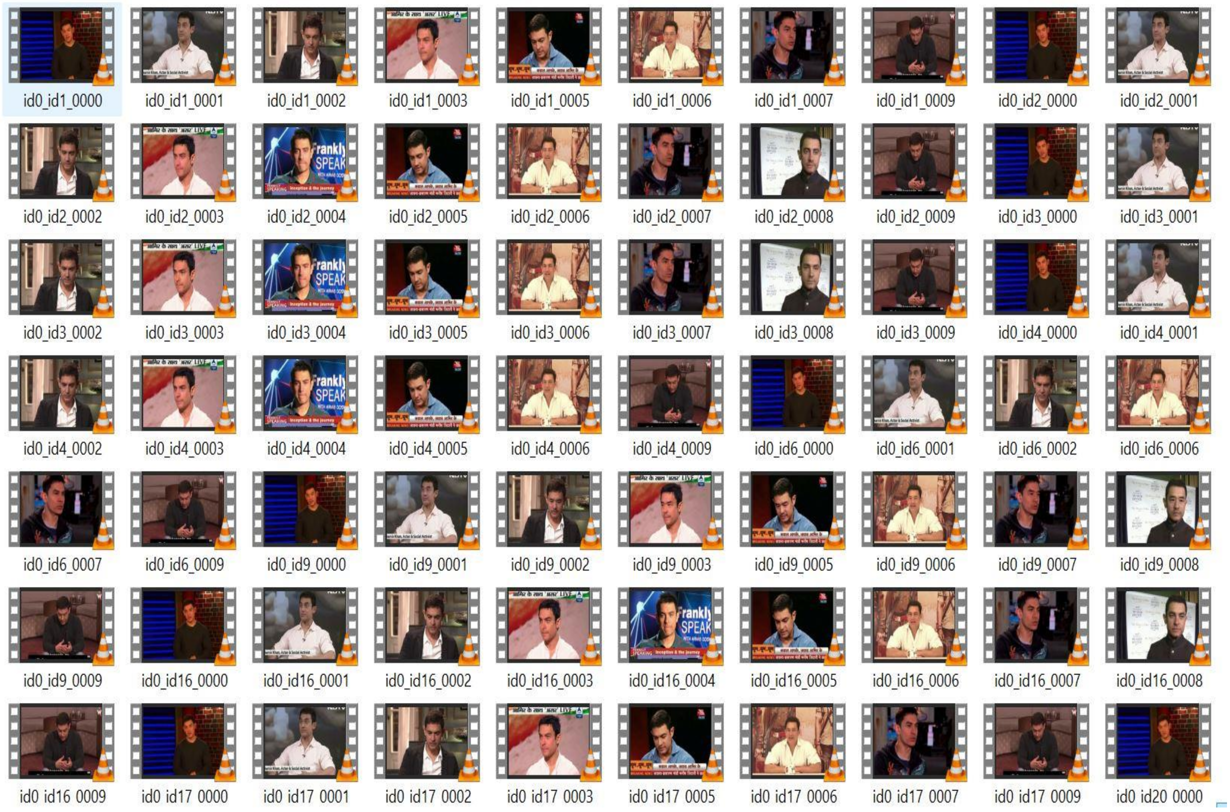


Figure 5.3 Final Dataset Snapshot-2

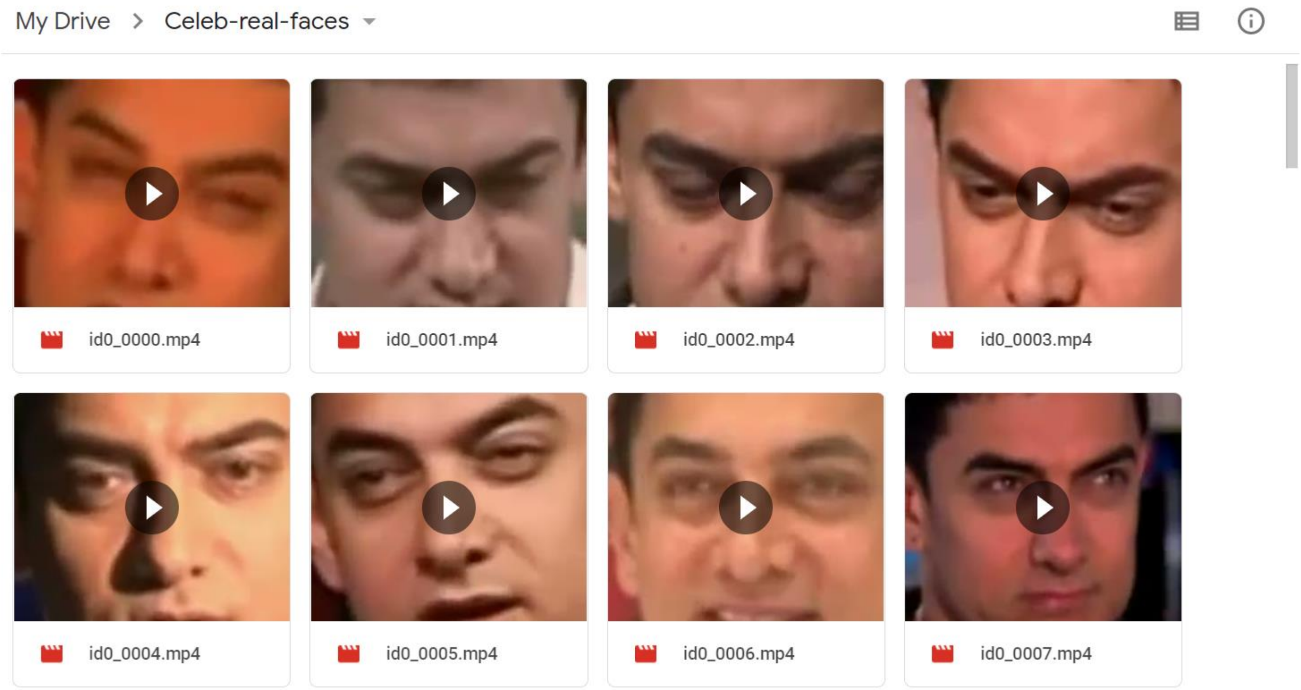


Figure 5.4 Face Cropped Videos from Dataset Snapshot-1

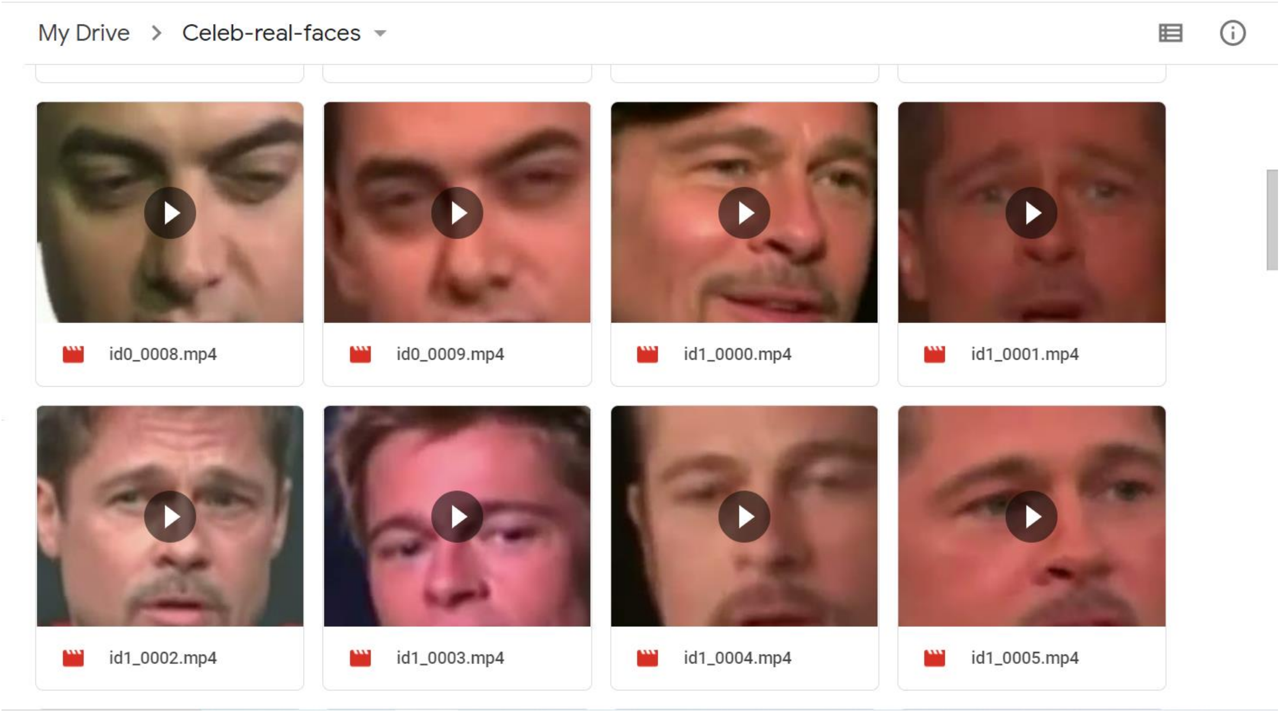


Figure 5.5 Face Cropped Videos from Dataset Snapshot-2

6 Chapter 6: PROPOSED MODEL

6.1 THE CNN MODEL

6.1.1 ResNeXt Model Block

ResNeXt[24] is a homogenous neural network that requires less number of hyperparameters than traditional Residual Networks(ResNet)[25]. This is accomplished through the use of "cardinality," a third dimension added to ResNet's width and depth. The size of the set of transformations is defined by its cardinality.

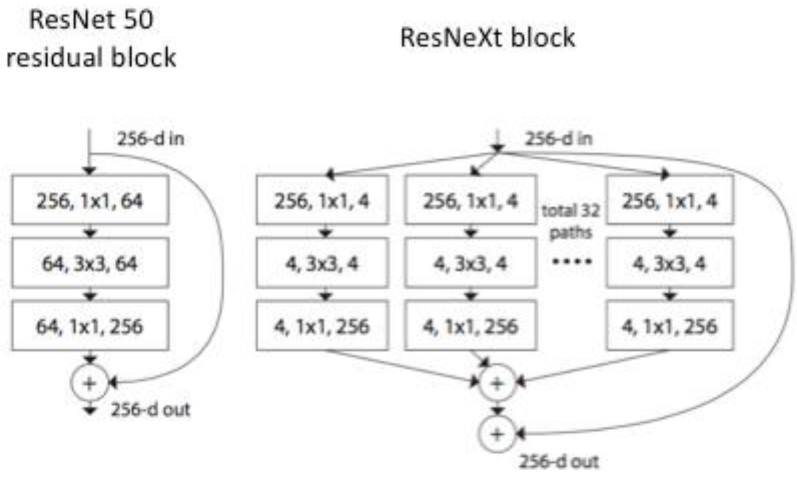
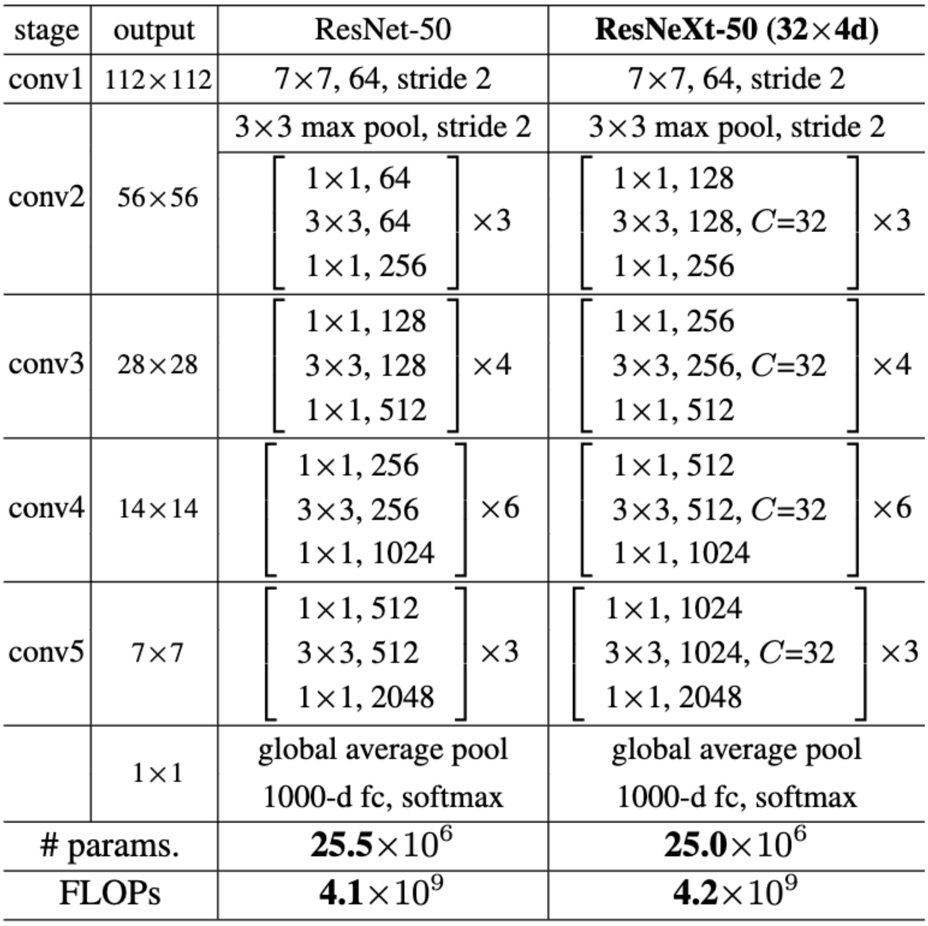


Figure 6.1 ResNeXt Block

A traditional ResNet block is shown on the left, while the cardinality of the ResNeXt block on the right is 32. The identical transformations are repeated over 32 times, with the final result being aggregated.

6.1.2 ResNeXt Architecture

ResNeXt's core architecture is determined by two rules. First and foremost, if the blocks create identical-dimensional spatial maps, then they share among each other the same set of hyperparameters, and second, if the spatial map is down sampled by a factor of 2, the block's width is increased by a factor of 2.



stage	output	ResNet-50	**ResNeXt-50 (32×4d)**
conv1	112×112	7×7, 64, stride 2	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2	3×3 max pool, stride 2
		$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 128\\3\times3, 128, C{=}32\\1\times1, 256\end{bmatrix}\times3$
conv3	28×28	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 256\\3\times3, 256, C{=}32\\1\times1, 512\end{bmatrix}\times4$
conv4	14×14	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 512\\3\times3, 512, C{=}32\\1\times1, 1024\end{bmatrix}\times6$
conv5	7×7	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 1024\\3\times3, 1024, C{=}32\\1\times1, 2048\end{bmatrix}\times3$
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params.		$\mathbf{25.5}\times10^6$	$\mathbf{25.0}\times10^6$
FLOPs		$\mathbf{4.1}\times10^9$	$\mathbf{4.2}\times10^9$

Figure 6.2 ResNeXt Architecture

6.1.3 Squeeze and Excitation Network (SENet)

The Squeeze-and-Excitation Block[26] is an architectural block that allows a network to execute dynamic channel-wise feature recalibration to increase its representational power. Here's how it works:

1) As an input, the block has a convolutional block.
2) Using average pooling, every channel is "squeezed" into a single numeric value.
3) Addition of non-linearity is done via a thick layer followed by a ReLU, and the complexity of output channel is lowered by a factor.
4) Each of the channels have a smooth gating function thanks to another thick layer followed by a sigmoid.
5) Finally, we use the side network to weight each feature map in the convolutional block; this is called "excitation."

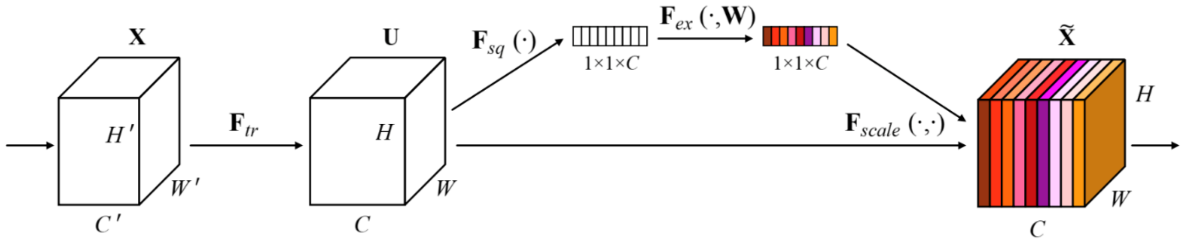


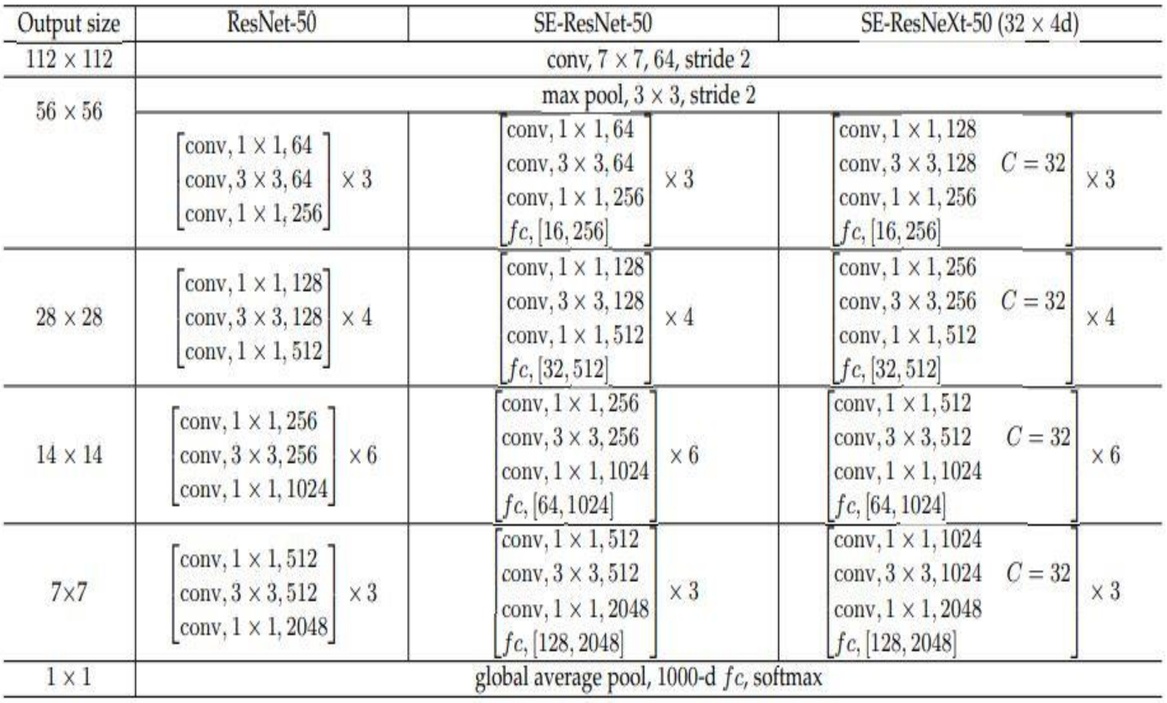
Figure 6.3 Squeeze and Excitation Block

To obtain features U, a feature transformation is first applied to the input picture X. Following that, we squeeze each channel of output U to obtain a single value. After that, we apply an excitation operation to the squeeze operation's output to get per-channel weights.

Finally, after we have the per-channel weights, we rescale the feature map U with these activations to get the block's final output.

6.1.4 SE-ResNeXt-101 Architecture

Squeeze and Excitation ResNeXt or SE-ResNeXt is one of a type of a ResNext model which makes use of the squeeze-and-excitation blocks to enable the network to work on dynamic feature recalibration channel-wise.



Output size	ResNet-50	SE-ResNet-50	SE-ResNeXt-50 (32 × 4d)
112 × 112	conv, 7 × 7, 64, stride 2		
56 × 56	max pool, 3 × 3, stride 2		
	[conv, 1 × 1, 64 conv, 3 × 3, 64 conv, 1 × 1, 256] × 3	[conv, 1 × 1, 64 conv, 3 × 3, 64 conv, 1 × 1, 256 *fc*, [16, 256]] × 3	[conv, 1 × 1, 128 conv, 3 × 3, 128 C = 32 conv, 1 × 1, 256 *fc*, [16, 256]] × 3
28 × 28	[conv, 1 × 1, 128 conv, 3 × 3, 128 conv, 1 × 1, 512] × 4	[conv, 1 × 1, 128 conv, 3 × 3, 128 conv, 1 × 1, 512 *fc*, [32, 512]] × 4	[conv, 1 × 1, 256 conv, 3 × 3, 256 C = 32 conv, 1 × 1, 512 *fc*, [32, 512]] × 4
14 × 14	[conv, 1 × 1, 256 conv, 3 × 3, 256 conv, 1 × 1, 1024] × 6	[conv, 1 × 1, 256 conv, 3 × 3, 256 conv, 1 × 1, 1024 *fc*, [64, 1024]] × 6	[conv, 1 × 1, 512 conv, 3 × 3, 512 C = 32 conv, 1 × 1, 1024 *fc*, [64, 1024]] × 6
7×7	[conv, 1 × 1, 512 conv, 3 × 3, 512 conv, 1 × 1, 2048] × 3	[conv, 1 × 1, 512 conv, 3 × 3, 512 conv, 1 × 1, 2048 *fc*, [128, 2048]] × 3	[conv, 1 × 1, 1024 conv, 3 × 3, 1024 C = 32 conv, 1 × 1, 2048 *fc*, [128, 2048]] × 3
1 × 1	global average pool, 1000-d *fc*, softmax		

Figure 6.4 SE-ResNeXt Architecture

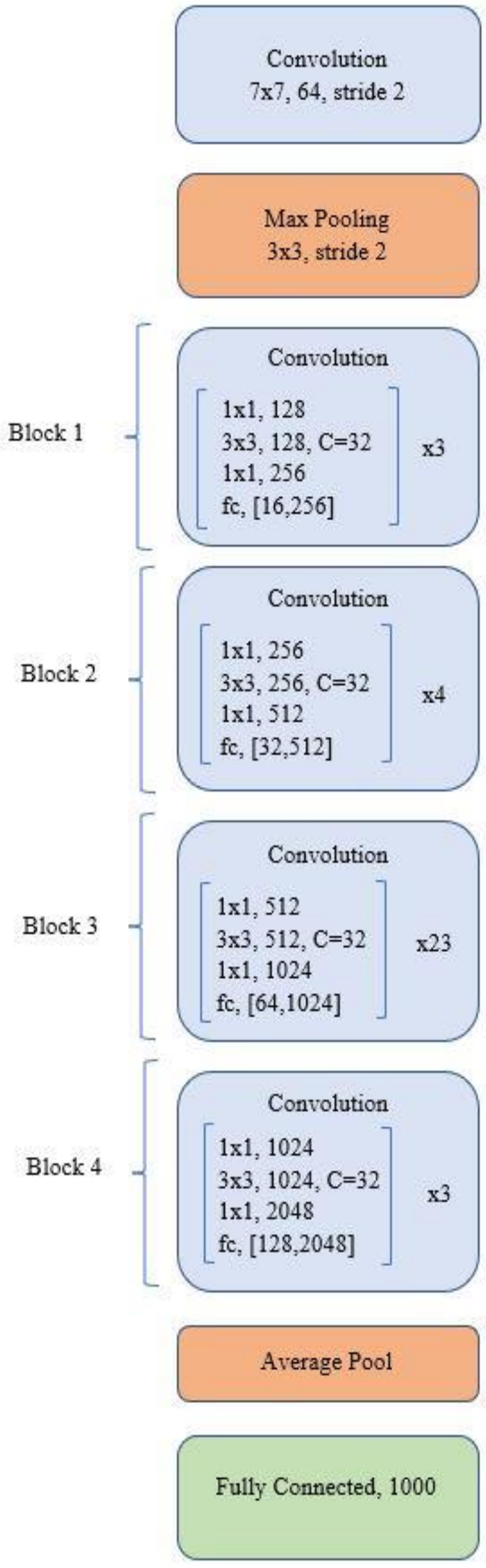


Figure 6.5 SE-ResNeXt-101 CNN Model

6.2 THE RNN MODEL

6.2.1 Long Short-Term Memory (LSTM)

LSTM[27] or Long Short-Term Memory is one of a type of artificial recurrent neural networks that is mostly used in the deep learning field. It processes single data points like images as well as entire sequences of data like video or speech. The LSTM has feedback connections.

A typical LSTM unit has a cell, a forget, an output and an input gate. The LSTM networks are quite efficient in problems related to classification, processing and prediction making given data for a time series, since there can be uneven duration gaps between crucial events in a time sequence. The values are retained by the cell over arbitrary time intervals and the three gates regulates the information flow in and out of the cell. LSTMs were primarily developed to handle the problem of vanishing gradient which is quite evident when training typical RNN models. Having lesser insensitivity to gap lengths is one of the advantages of LSTM networks over RNNs as well as other methods that involve sequence learning.

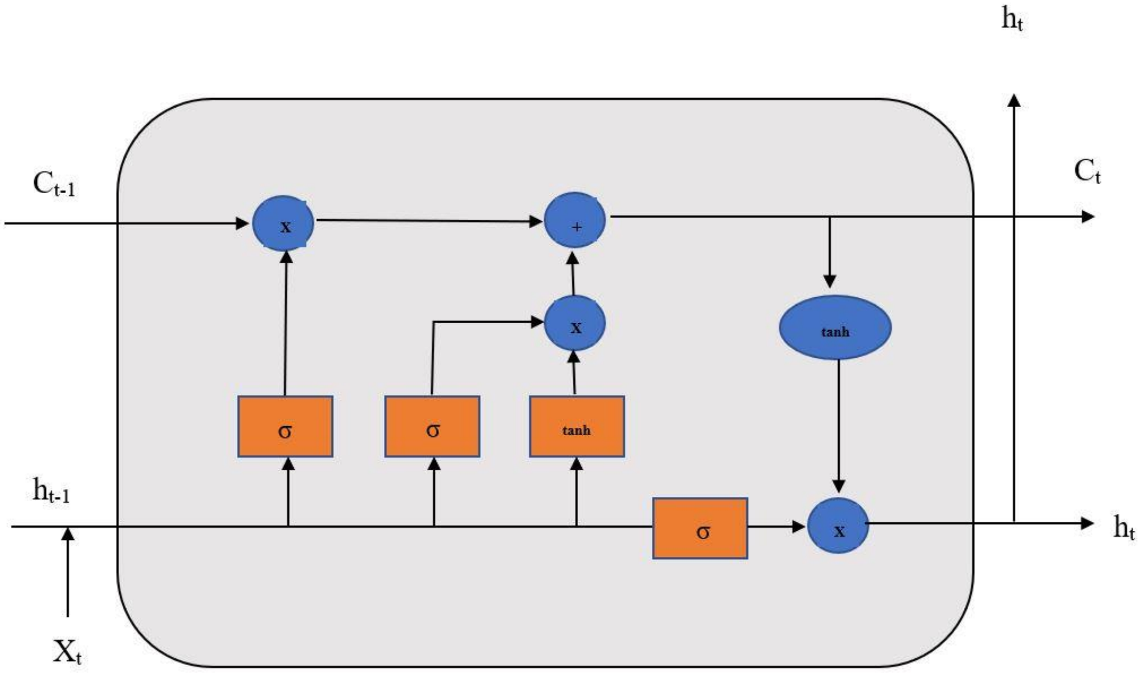


Figure 6.6 LSTM Block Architecture

6.3 PRE-PROCESSING

The process of dataset pre-processing involves-

- Data cleaning: This involves removing the videos with lesser number of frames so that we can have ample number of frames for further processing.
- Frame Extraction: Splitting the video into frames one by one.
- Cropping Facial Region: We will be detecting the facial region using the batch_face_locations function from face_recognition package.
- Stacking face cropped frames to create video: We will combine the face cropped frames extracted to form a video having only facial region.

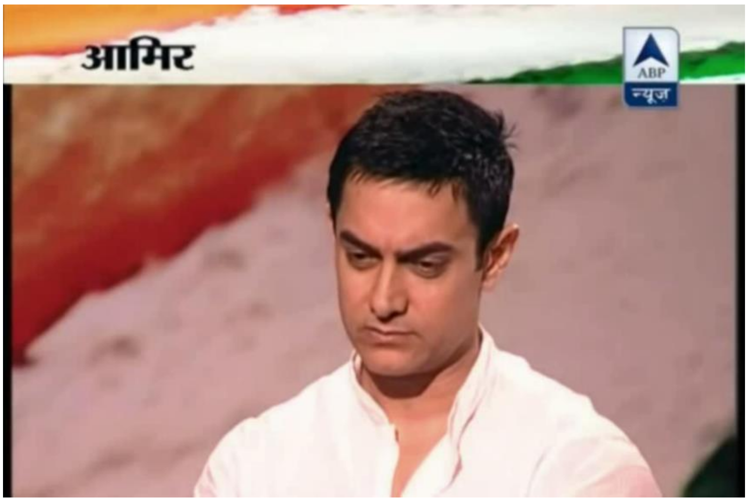


Figure 6.7 Video Snapshot before pre-processing

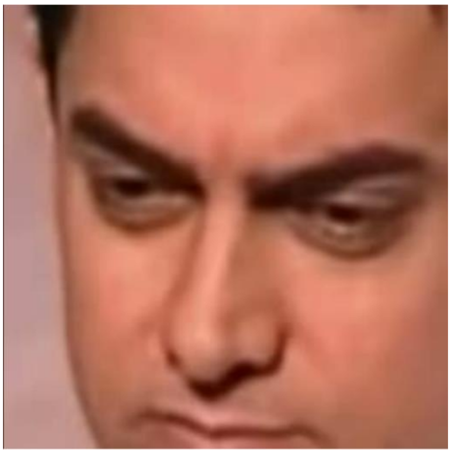


Figure 6.8 Video Snapshot after pre-processing

To maintain the uniformity among the videos, we have calculated the mean of the number of frames in each video in the dataset. Further, the new processed face cropped dataset which is formed will be having the number of frames equal to the mean we have calculated. The frames in which no face is present are simply ignored during the step.

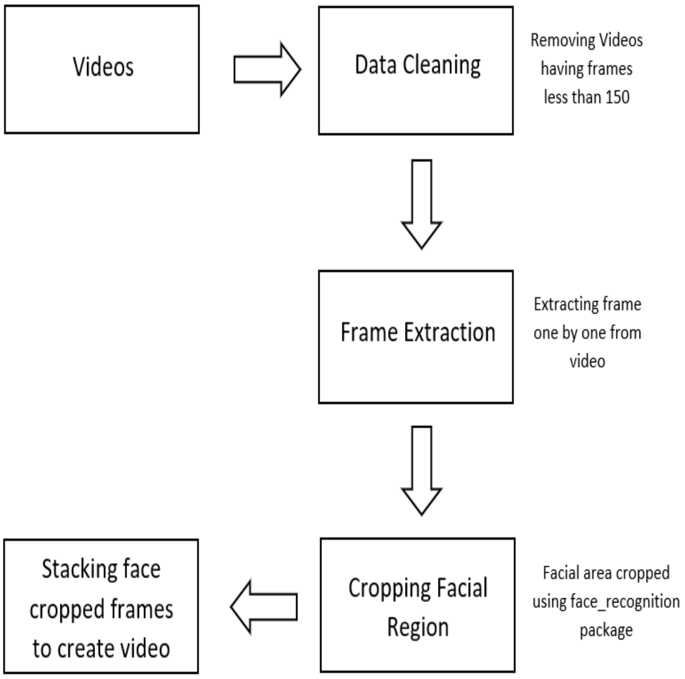

Videos
Data Cleaning
Removing Videos having frames less than 150
Frame Extraction
Extracting frame one by one from video
Cropping Facial Region
Facial area cropped using face_recognition package
Stacking face cropped frames to create video

Figure 6.9 Data Pre-Processing Process

6.4 FEATURE EXTRACTION

6.4.1 SE-ResNeXt-101

On top of the network is the fully connected layer that is deleted from the SE-ResNeXt-101, allowing the ImageNet model to immediately generate a detailed representation of each frame. Following the final pooling layers, the output which are the 2048-dimensional feature vectors are utilised as the LSTM's sequential input.

6.5 CLASSIFICATION

6.5.1 LSTM

Our LSTM model, in particular, takes a series of 2048-dimensional ImageNet feature vectors during training. A 512 completely-connected layer with a 0.4 risk of dropout follows the LSTM.

Finally, we compute the odds of the frame sequence being pristine or deepfake using a softmax layer. The LSTM module is an intermediary unit in our pipeline that is trained totally without the need of auxiliary loss functions from start to finish.

6.6 THE PROPOSED ARCHITECTURE

We have implemented the model consisting of CNN architecture combined with RNN model. For CNN, we have implemented SE-ResNeXt-101 followed by one LSTM layer.

We first loaded the pre-processed face cropped videos and then split the videos into training and testing set. The frames from the pre-processed videos are then passed to the proposed model for doing the training and testing in mini-batches.

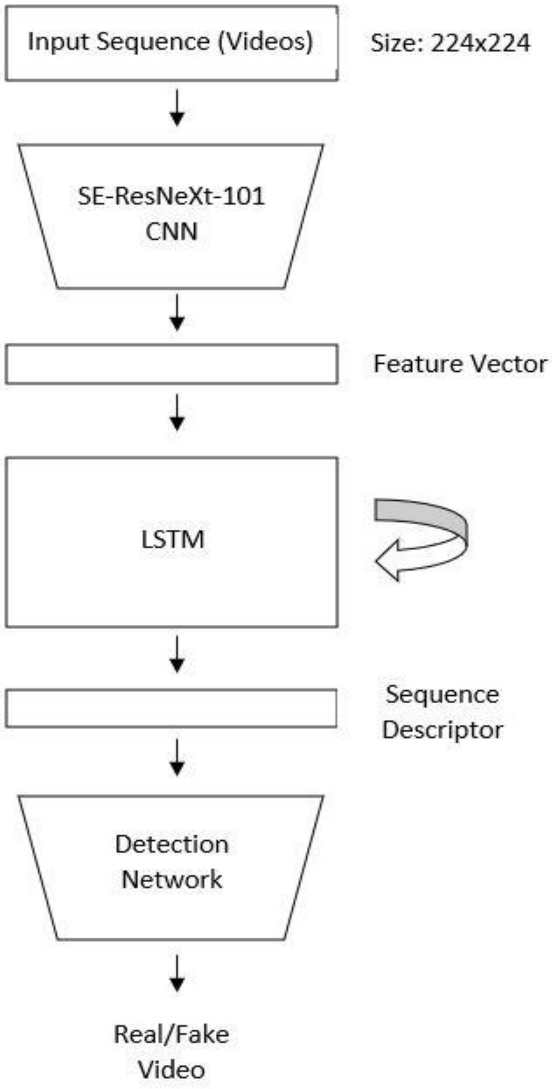


Figure 6.10 Proposed Model Architecture

We have implemented the SE-ResNext-101 CNN along with LSTM for tampered video detection. We compiled our dataset from various sources and then did pre-processing of the videos, such that we had extracted face cropped images from the videos and stacked them to form a video containing face images.

The pre-processed videos were then split for training as well as testing in the ratio 3:1 respectively. Then, we performed training of our model based on the pre-processed dataset.

We demonstrate the proposed system performance in terms of validation accuracy using 20-frame sub-sequences. These frame sequences are collected from each video in order.

In the validation set, the complete pipeline is trained end-to-end until we achieve a 20-epoch loss plateau.

7 Chapter 7: RESULTS AND CONCLUSION

A. Accuracy

In the mentioned research, we found out that the model with 20 frames sequence length obtained a training accuracy of 95.60% and a testing accuracy of 84.34% after 25 epochs. This states that the model correctly classifies the videos as either a real or a deepfake video with more than 84 percent accuracy.

B. Confusion Matrix

The Figure demonstrates the confusion matrix of the performed research. The confusion matrix mentioned such that the lighter colour shade corresponds to larger numbers whereas darker colour shade corresponds to lesser number for every class. The confusion matrix is darker for the classes False Positive and False Negative whereas lighter for the classes True Positive and True Negative, depicting that the model is making a fairly good prediction with less errors.

True positive = 623

False positive = 68

False negative = 72

True negative = 131

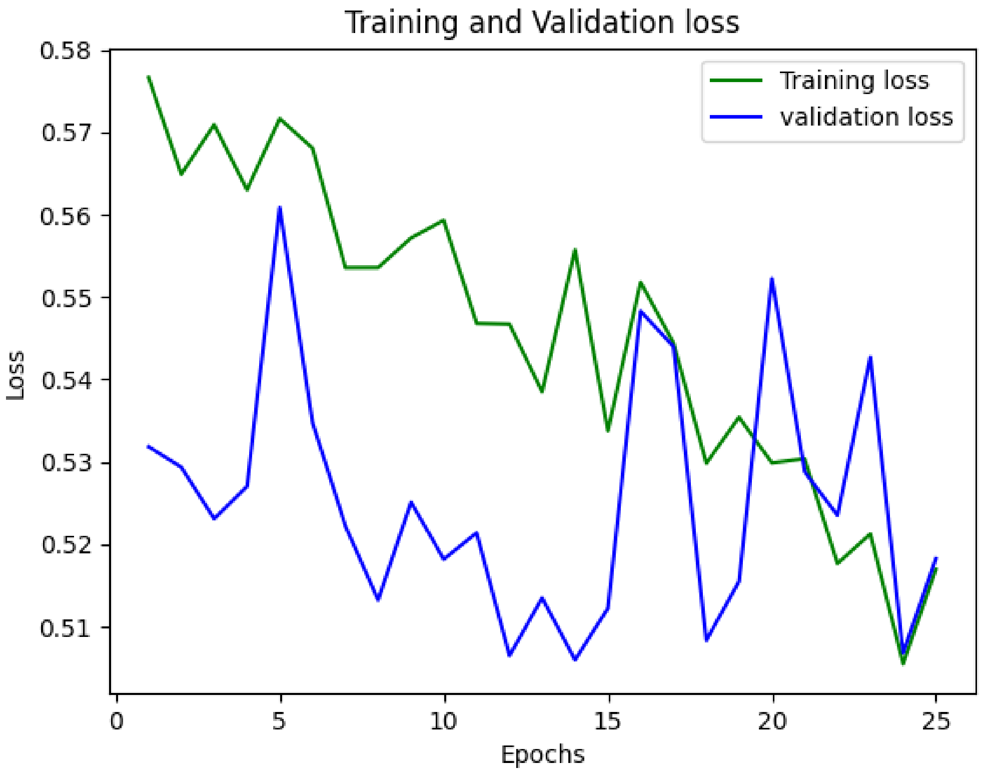


Figure 6.11 Training and Validation Loss Graph

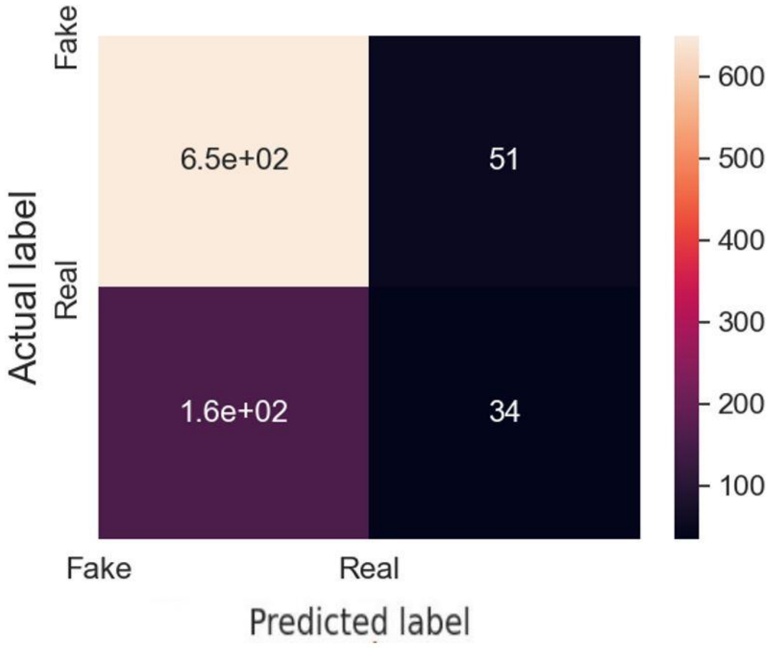
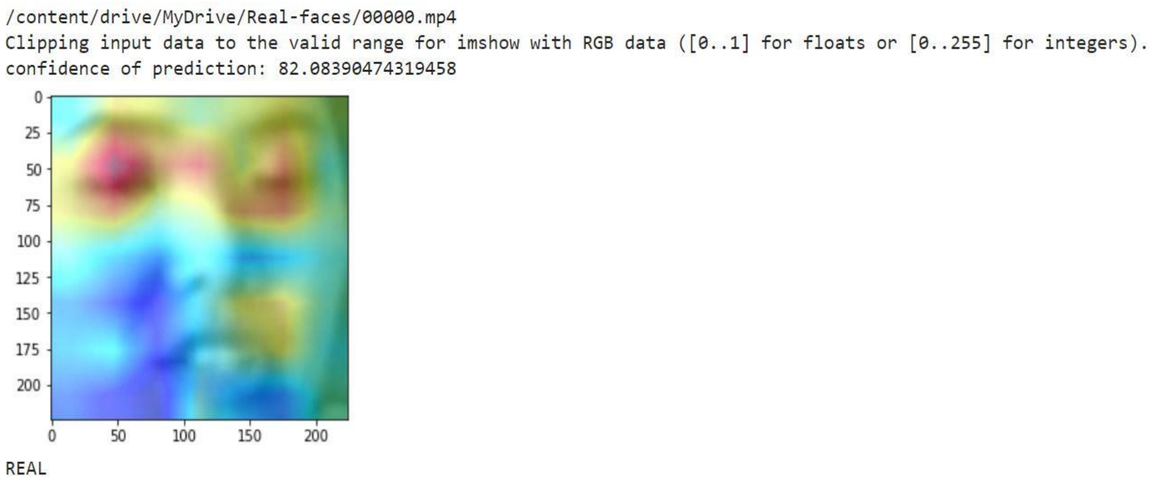


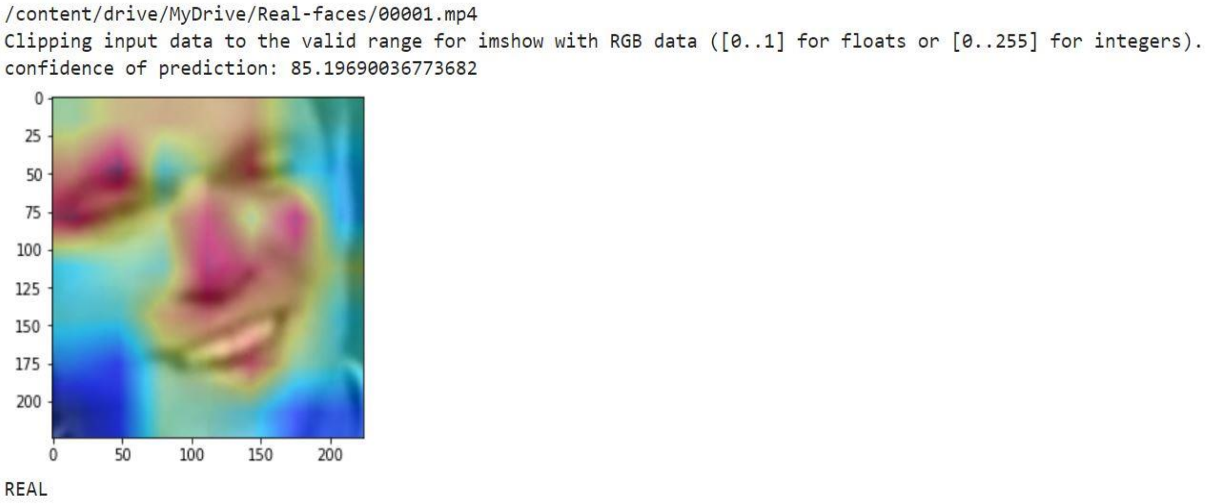
Figure 6.12 Confusion Matrix



```
/content/drive/MyDrive/Real-faces/00000.mp4
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
confidence of prediction: 82.08390474319458
```

```
REAL
```

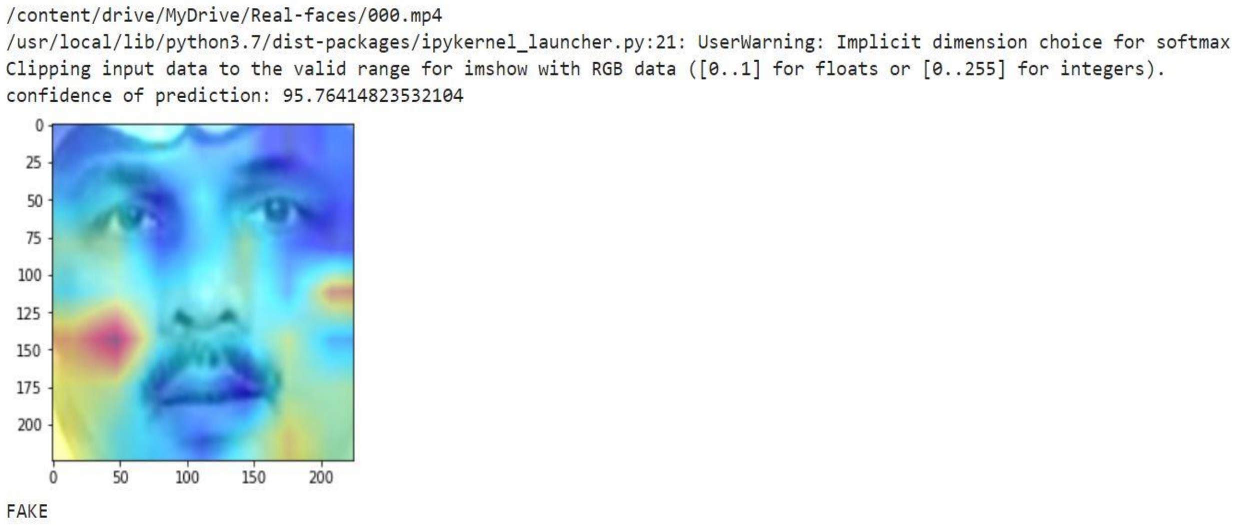
Figure 6.13 Classification and Confidence of prediction for video snapshot-1



```
/content/drive/MyDrive/Real-faces/00001.mp4
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
confidence of prediction: 85.19690036773682
```

```
REAL
```

Figure 6.14 Classification and Confidence of prediction for video snapshot-2



```
/content/drive/MyDrive/Real-faces/000.mp4
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:21: UserWarning: Implicit dimension choice for softmax
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
confidence of prediction: 95.76414823532104
```

```
FAKE
```

Figure 6.15 Classification and Confidence of prediction for video snapshot-3

The proposed model is a hybrid CNN-RNN model made using SE-ResNeXt-101 CNN and LSTM RNN model that can detect deepfake videos. The dataset collected area taken from various distributed sources, namely Faceforensics++ and Celeb-DF v2 datasets so that the model can be trained on distributed as well as effective data.

The experimental results obtained from the proposed model using a large and varied collection of tampered videos have shown that by implementing a convolutional LSTM structure, we can quite accurately predict the video as a real or a deepfake video that has been subjected to manipulation.

We believe that the research work presented presents a strong case to be considered for further research and development. We firmly believe that the work presented can be considered as a basis for further developments and improvements.

REFERENCES

[1] A. Koenig, "'Half the Truth is Often a Great Lie': Deep Fakes, Open Source Information, and International Criminal Law," *AJIL Unbound*, vol. 113, no. 2017, pp. 250–255, 2019, doi: 10.1017/aju.2019.47.

[2] "Obama Deepfake." https://www.abc.net.au/news/2018-09-27/fake-news-part-one/10308638?nw=0&r=HtmlFragment.

[3] "What are Deepfakes." https://www.theguardian.com/technology/2020/jan/13/what-are-deepfakes-and-how-can-you-spot-them.

[4] S. Agarwal, H. Farid, T. El-Gaaly, and S. N. Lim, "Detecting Deep-Fake Videos from Appearance and Behavior," *2020 IEEE Int. Work. Inf. Forensics Secur. WIFS 2020*, 2020, doi: 10.1109/WIFS49906.2020.9360904.

[5] "Jon Snow Apology Deepfake," [Online]. Available: https://knowtechie.com/jon-snow-game-of-thrones-deepfake/.

[6] "Mark Zuckerberg Declaration Deepfake," [Online]. Available: https://www.businessinsider.in/theres-a-fake-video-showing-mark-zuckerberg-saying-hes-in-control-of-billions-of-peoples-stolen-data-as-facebook-grapples-with-doctored-videos-that-spread-misinformation/articleshow/69748718.cms?utm_source=copy-link&utm_medium=referral&utm_campaign=Click_through_social_share.

[7] T. T. Nguyen *et al.*, "Deep Learning for Deepfakes Creation and Detection: A Survey," *SSRN Electron. J.*, no. July 2020, 2022, doi: 10.2139/ssrn.4030341.

[8] "FaceApp." https://www.rankred.com/best-deepfake-apps-tools/.

[9] Y. Li and S. Lyu, "Exposing DeepFake Videos by Detecting FaceWarping Artifacts," *arXiv*, 2018.

[10] T. Jung, S. Kim, and K. Kim, "DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern," *IEEE Access*, vol. 8, pp. 83144–83154, 2020, doi: 10.1109/ACCESS.2020.2988660.

[11] H. Kim *et al.*, "Deep video portraits," *ACM Trans. Graph.*, vol. 37, no. 4, 2018, doi: 10.1145/3197517.3201283.

[12] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial

video forgery detection network," *10th IEEE Int. Work. Inf. Forensics Secur. WIFS 2018*, 2019, doi: 10.1109/WIFS.2018.8630761.

[13] X. Yang, Y. Li, and S. Lyu, "Exposing Deep Fakes Using Inconsistent Head Poses," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2019-May, pp. 8261–8265, 2019, doi: 10.1109/ICASSP.2019.8683164.

[14] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.

[15] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-Octob, pp. 1–11, 2019, doi: 10.1109/ICCV.2019.00009.

[16] B. Dolhansky *et al.*, "The DeepFake Detection Challenge (DFDC) Dataset," 2020, [Online]. Available: http://arxiv.org/abs/2006.07397.

[17] J. Mao, "Why artificial neural networks - Mao - 1996.pdf," 1996.

[18] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," pp. 1–11, 2015, [Online]. Available: http://arxiv.org/abs/1511.08458.

[19] A. Graves, "Generating Sequences With Recurrent Neural Networks," pp. 1–43, 2013, [Online]. Available: http://arxiv.org/abs/1308.0850.

[20] J. Luo and J. Huang, "Generative adversarial network: An overview," *Yi Qi Yi Biao Xue Bao/Chinese J. Sci. Instrum.*, vol. 40, no. 3, pp. 74–84, 2019, doi: 10.19650/j.cnki.cjsi.J1804413.

[21] Z. Pan, Y. Ren, and X. Zhang, "Low-complexity fake face detection based on forensic similarity," *Multimed. Syst.*, vol. 27, no. 3, pp. 353–361, 2021, doi: 10.1007/s00530-021-00756-y.

[22] "Autoencoders & Decoders." https://www.alanzucconi.com/2018/03/14/understanding-the-technology-behind-deepfakes/.

[23] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics," *Proc. IEEE Comput. Soc. Conf. Comput. Vis.*

Pattern Recognit., pp. 3204–3213, 2020, doi: 10.1109/CVPR42600.2020.00327.

[24] S. Xie, R. Girshick, and P. Doll, “Aggregated Residual Transformations for Deep Neural Networks.”

[25] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[26] J. Hu, “Squeeze-and-Excitation_Networks_CVPR_2018_paper.pdf,” *Cvpr*, pp. 7132–7141, 2018, [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html.

[27] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.