SIAMESE NETWORK-BASED MULTI-MODAL DEEPFAKE DETECTION

A THESIS IN
Computer Science

Presented to the Faculty of the University
Of Missouri-Kansas City in partial fulfillment
Of the requirements for the degree

MASTER OF SCIENCE

By
RAJU NEKADI

B.E., RGPV University Institute of Technology, Bhopal 462036, India

Kansas City, Missouri
2020

©2020

RAJU NEKADI
ALL RIGHTS RESERVED

SIAMESE NETWORK-BASED MULTI-MODAL DEEPFAKE DETECTION

Raju Nekadi, Candidate for the Master of Science Degree

University of Missouri-Kansas City, 2020

ABSTRACT

Deep learning widely applies to solve various problems in healthcare, robotics, and computer vision. Presently, an emerging deep learning application called "deepfake" has raised concerns about the multiple types of security threats that may pose severe harm to personal privacy and public safety. Deep convolutional neural networks like VGGNet and InceptionNet have recently set a proposal for detecting deepfake. The main challenge of these CNN-based algorithms is that they require extensive training datasets and high-end GPU resources. Furthermore, these studies mainly focus on identifying patterns in facial expressions in deepfake, and there are only very few studies on detecting audio fakeness.

In this thesis, we propose a novel method for uni-modal or multi-modal deepfake detection with minimum resources. The proposed solution was designed with a Siamese network-based deepfake model with invariant of constructive loss and triplet loss. Contrastive loss uses the trained network's output for a positive example and calculates its distance to an instance of the same class and contrasts it with the range to negative samples. The triplet loss was computed by positioning the baseline that minimizes the distance to positive samples but maximizes the distance to negative samples. To test and validate our proposed model, we report our metrics like similarity score, loss, and

accuracy on large-scale DFDC, Faceforensic++, and CelebDF datasets. We compared our method with state-of-the-art algorithms and confirmed that our overall accuracy is improved by 2-3% for deepfake detection.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Computing and Engineering, have examined a thesis titled "Siamese Network-Based Multi-Modal Deepfake Detection" presented by Raju Nekadi, candidate for the Master of Science degree, and hereby certify that in their opinion, it is worthy of acceptance.

Supervisory Committee

Yugyung Lee, Ph.D. (Committee Chair)
Department of Computer Science Electrical Engineering

Farid Nait-Abesselem, Ph.D.
Department of Computer Science Electrical Engineering

Md Yusuf Sarwar Uddin, Ph.D.
Department of Computer Science Electrical Engineering

Contents

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

ACKNOWLEDGMENTS

I feel very fortunate to work under the supervision of Dr. Yugyung Lee. I would like to thank her for the valuable guidance and immense support throughout the research work as my advisor. Her vast experience, unparalleled knowledge, agile and prompt feedback coupled with smart ideas have helped me immensely in putting up the whole job. She is very patient in listening to all the new ideas, being pragmatic in giving suggestions, and always helping me with a reality check. Her fantastic energy and enthusiasm always motivate me to go the extra mile. It has been an honor to work with her on many projects besides the thesis.

I would like to thank the thesis committee Dr. Yusuf Sarwar Uddin and Dr. Farid Nait-Abesselem, for providing valuable feedback.

I would also like to thank the University of Missouri-Kansas City for providing me the perfect environment to research. It provided me with many opportunities to support myself and world-class facilities to research with the most exceptional machines available without which the thesis work could not have accomplished.

Finally, I would like to thank my family and friends who always encourage me, gave me valuable suggestions throughout the research, and made sure that I pursue my dream without any problems.

CHAPTER 1. INTRODUCTION

The increasing popularity of social media (YouTube, Vimeo, Facebook, Snapchat, Twitter, and Instagram) and the full availability of high-end camera cell phones have made creation, editing, and distribution of digital media more accessible than ever in history. Every day, millions of videos/images/audio are shared through social media sites. A lot of fake video and sound generated by the digital manipulation method called "DeepFake" [1] and shared on this social media have become significant public privacy concerns recently [2], [3].

The popular term "DeepFake" is a deep learning-based technique that can create fake images/video/audio of the target person from the source person doing or saying. The first deepfake video emerged in 2017 after a Reddit user claimed to have developed a machine-learning algorithm that helped him to transpose celebrity faces into porn videos [4]. It is menacing to world security when deepfake algorithms can be employed to create videos of w leaders with fake speeches for falsification purposes [5, 6]. Deepfakes, therefore, can be misused to create chaos in financial markets by creating fake news.

There are also positive cases of deepfake usages, such as creating voices who have lost theirs or updating episodes of movies without reshooting those [7]. However, the number of misuse of deepfakes largely dominates that of the positive ones. Minimal effort is needed to produce a stunningly convincing manipulated media. A recent high-end algorithm can even create a deepfake with just a still image [8]. Deepfakes, therefore, can be a threat affect not only to politicians or celebrities but also

to the common man. For example, recently, an audio deepfake was used to scam a CEO out of $243,000 [9].

A recent release of Deep Nude software shows more disturbing threats as it can transform a person into non-consensual porn [10]. As a result, the area of research conventional devoted to the general image and fake video detection [11] is being invigorated and is now dedicating growing efforts for detecting facial manipulation in image and video [12], [13]. These revived efforts in fake face detection are developed around past research in biometric anti-spoofing [14]–[16] and modern data-driven deep learning [17]. The growing interest in fake face and audio detection is demonstrated through the increasing number of workshops in top conferences [18]–[20] and competitions such as the recent DFDC2 [21] and ASVSpoof2019 [22] by Facebook and Google respectively.

1.1 Problem Statement

To detect deepfake at first glance, it looks like a straightforward classification problem. Still, when we dive into it, we will find out that many challenging factors are involved in identifying the manipulation. A lot of research and work has been done to find the manipulation in the video, such as those produced by prevailing state-of-the-art face and audio algorithms, including Deepfakes [1], Fac2Face [24], FaceSwap [23] and NeuralTextures [25]. However, all this work is focused on finding the manipulation in the visual frame in the video; no importance has been given to the audio aspect of the video. Face and Audio forgery

detection is a challenging problem in the real world. We need to detect the forgery without knowing the underlying manipulation methods. Most existing works [28, 29, 30] recognize the face and audio manipulation in a supervised fashion, and their methods are trained for now manipulation techniques. For such manipulation, these detection methods work quite well and achieve 98% accuracy. However, these detection methods tend to exploit only a single modality that is the facial clue from these videos either by employing temporal feature or by exploring visual artifacts within frames.

1.2 Proposed Solution

In this thesis, we proposed a multi-modal deepfake detection using siamese network-based architecture. The main idea used for deepfake detection is to exploit the relationship between visual and audio modalities from the same video. Earlier studies in both psychology literature and Multi-modal machine learning feature have shown a robust correlation between different modalities of the same subject [11,43].In our work, we introduce a new Siamese based network architecture for detecting deepfake videos. In the training phase, we feed the real and fake video through our network and obtain the modality embedding vector for face and audio of the same subject. We use these embedding vectors to calculate the constructive loss, triplet loss, and euclidean distance function to minimize the modality from fake video and maximize the similarity between real video.

CHAPTER 2. BACKGROUND AND RELATED WORK

This chapter gives background information on the various components used in the thesis and provides an overview of related work that will help understand this work better.

2.1 Background

2.1.1 Deepfake

Deepfakes are artificial media in which a source image or video is replaced with another person's data. While the act of pseudo content is not new, deepfakes leverage powerful and impactful techniques from machine learning and artificial intelligence to operate or generate visual and audio content with a high potential to defraud. Machine learning for deepfakes is based on autoencoder or generative adversarial networks (GANs). These methods have been adapted by deepfake algorithms to examine facial expressions and gestures of a person as well as synthesize facial images of another person making similar or parallel expressions and gestures. Deepfake algorithms typically require a massive amount of image and video data to train models to create photo-realistic images and videos.

2.1.2 Deepfake Techniques

Deepfakes have become well-liked due to the quality of tampered videos and also the easy-to-use capability of its applications to a broad set of users with diverse computer skills from professionals to the neophyte. These applications are mainly developed based on deep learning techniques. Deep learning is well known for its capacity to represent complex and high-dimensional data. Deep autoencoders, which have been widely applied for image compression and dimensionality reduction. The initial attempt of deepfake creation was

FakeApp, which was created by a Reddit user using an auto encoder-decoder pairing structure [23, 24]. In that algorithm, the autoencoder extracts latent features of face images, and the decoder is used to reconstruct the face images. In order to switch faces between target images and source images, there are required pairs of two encoder-decoder where each pair is used to train the encoder's parameters are shared between two network pairs and an image set. Hence, two pairs have the same encoder network. This plan enables the standard encoder to learn the similarity between two sets of face images. These are relatively unchallenging because faces usually have similar features such as eyes, nose, mouth positions and find the same Figure. One shows a deepfake development where the feature set of face A is connected with the decoder B to rebuild face B from the original face A. This approach is applicable in several works like DeepFaceLab [25], DFaker [26], DeepFake-tf[27].

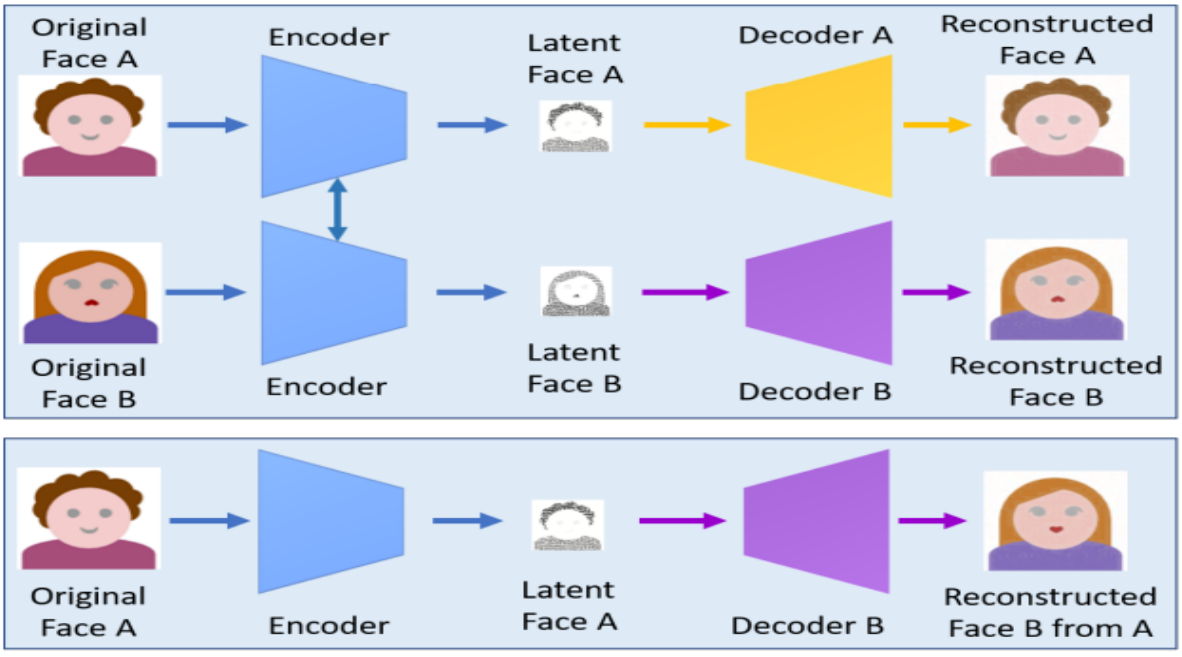

Figure. 1: Auto Encoder-Decoder

Table 1: Details Of Deepfake Tools

Tools	Key Features
Faceswap	- Parameters of the encoder are shared. - Using two encoder-decoder pair.
Faceswap-GAN	-Adversarial loss and perceptual loss (VGGFace Deepfake detection is typically deemed a binary classification problem where classifiers are used to classify) are added to the auto-encoder architecture.
DeepFaceLab	Expand from the faceswap model with a new model
DFaker	- Implemented based on Keras library.
DeepFake-tf	Similar to DFaker but implemented based on tensor flow.

Using perceptual loss and adversarial loss and used in VGGFace [28] to the encoder-decoder architecture, and refined version of deepfakes based on the generative adversarial network (GAN) [29] figure 2, i.e., faceswap-GAN, proposed in [30]. The VGGFace perceptual loss was added to make eye shift to be more consistent and realistic with input faces and help to flatten out artifacts in a segmentation mask, leading to higher quality output videos. This model creates outputs with 64x64, 128x128, and 256x256 resolutions. FaceNet [31] multi-task convolution network helps make face detection more stable and face alignment more reliable. The CycleGAN [32] is used for generative network implementation. In summary, important deepfake tools and their features are presented in Table 1.

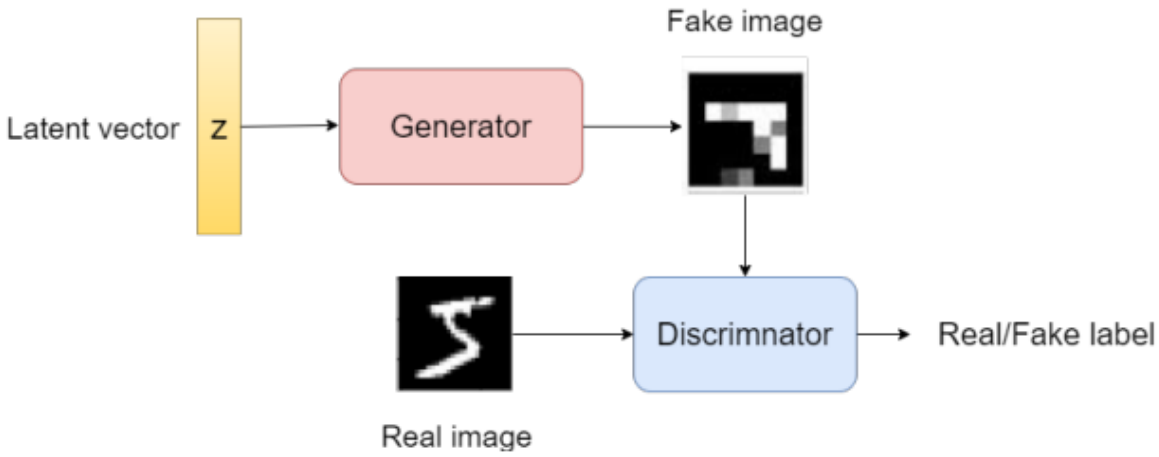

Figure 2: Generative Adversarial Network

2.1.3 Siamese Network

The Siamese network [14] shown in Figure 3, the architecture comprises of two neural networks that share the same weights and are trained in conjunction. Each network typically takes in an unlike pattern (e.g., two views of an image(s), two tones of a speech), and the final output is a value representing the resemblance between those two models. The overall network is trained using different forms of the triplet loss or the contrastive loss. These are formulated to maximize the distance between features learned from differing patterns and minimize the distance between features learned from non-differing patterns. With this training target, Siamese network-based architectures have been extensively used in applications such as face recognition [16], face verification [44], and speaker identification [15]. In our work, we proposed a Siamese network-based architecture and an alternative of the triplet loss to maximally unrelated features leamed from real and fake videos.

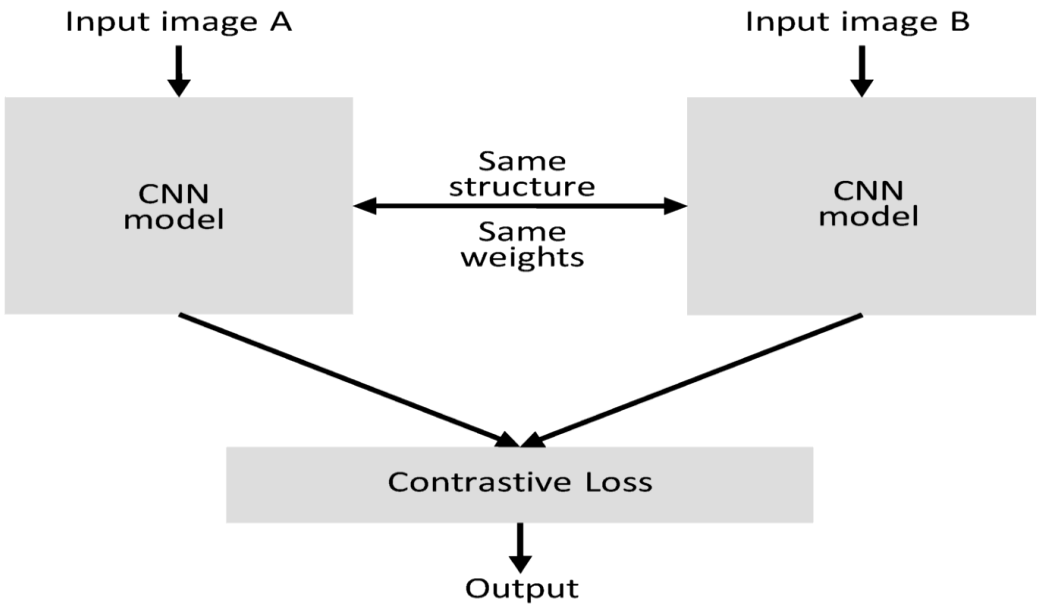

Figure 3: Siamese Network Architecture

2.1.4 Mel-Frequency Cepstrum Coefficients (MFCCs)

Speech signals can be represented digitally as an array of numbers with the same number of elements per second as the sampling rate. However, this representation does not contain information useful for speech recognition. In order to address this problem, the raw audio signal can be converted to the frequency domain using a fast Fourier transform (FFT) on a small audio window. While the FFT contains energy information at each frequency band in the audio window, it does not emphasize the band that is important for human hearing, which is below roughly 1000 Hz. To overcome this problem, Stevens et al. [53] suggested a scale, as shown in (1), to improve the emphasis on the frequency important to human hearing.

$$M(f) = 1125 \ln(1 + f/700) \quad (1)$$

The Mel-frequency cepstrum coefficient (MFCC) is a speech signal feature commonly used in ASR research as well as music classification research. Figure 2 shows the process of extracting MFCCs for a speech signal.

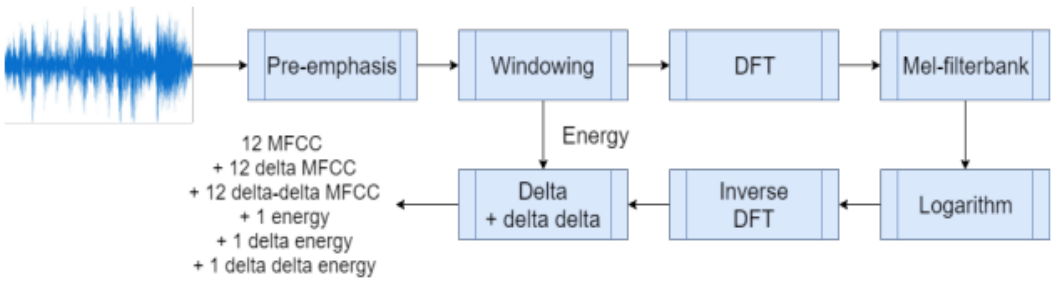

Figure 4 Mel-Frequency Cepstrum Coefficients

Since the characteristic of speech changes throughout an utterance, spectral features obtained over the entire statement would not convey useful information. Instead, features are typically extracted over a small window, typically 25 ms, stridden by 10 ms each window is then passed through the pipeline shown in Figure 5. The pre-emphasis step in Figure 4 aims to boost the signal's energy at high frequency. The high frequency of human speech typically has lower energy than low frequency but is also relevant to the speech recognition task. After the pre-emphasis step, the windowing step involves multiplying the signal with a predefined window. While a rectangular window (Figure 5a) is the most straightforward window to use, it causes the signal to be abruptly cut off at the edge. The cutoff causes problems when the discrete Fourier transform of the signal is obtained. A Hamming window approaches 0 at its edges, which shrinks the input signal value toward 0 at the boundaries [36].

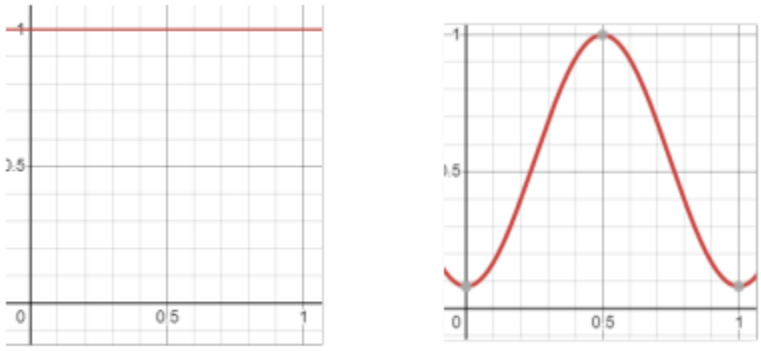

Figure 5: Mel-frequency Cepstrum Coefficients Wave

2.2 Related Work

2.2.1 DeepFake Detection

Deepfakes are increasingly harmful to privacy, society, and democracy. Methods for identifying deepfakes have been proposed as soon as this threat was introduced. Early work was based on handcrafted features obtained from inconsistencies and artifacts of the fake video synthesis process. On the other hand, the latest methods apply deep learning to extract salient and discriminative features to detect deepfakes automatically. Deepfake detection is typically deemed a classification problem where classifiers are used to classify between authentic videos and tampered ones. This kind of method requires a massive database of fake and real videos to train classification models. The number of manipulated videos is increasingly available online, but it is still limited in setting a benchmark for validating various detection methods.

2.2.2 Capsule Network

Nguyen et al. [26] proposed using capsule networks to identify manipulated videos and images. The capsule network was first introduced to address the drawbacks of CNNs when applied to reverse graphics tasks, which intent to find physical processes used to produce images of the world. The latest development of a capsule network based on a dynamic routing algorithm demonstrates its capability to describe the hierarchical pose relationships amid object parts. This creation is employed as a component in a pipeline for detecting fabricated images and videos, as illustrated in Figure. 6. A dynamic routing mechanism is used to route the outputs of all three capsules network through several iterations to distinguish between fake and real images. The proposed mechanism gave the best performance compared to its competing methods in all of these data sets.

This shows the effectiveness of the capsule network in building a comprehensive detection system that can work appropriately for various forged images and video attacks.

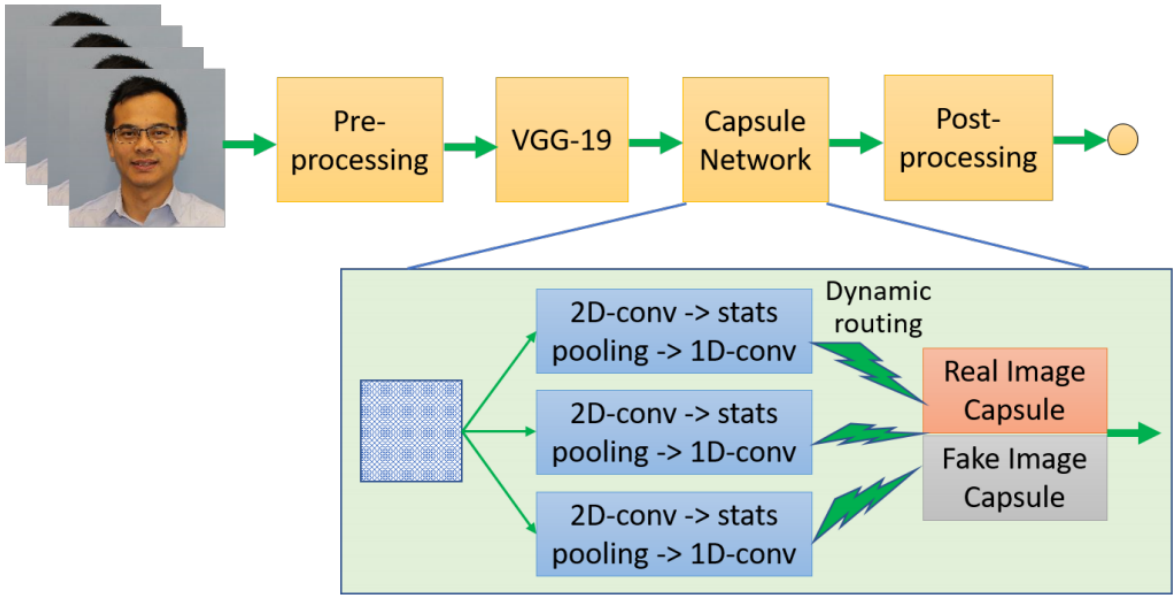

Figure 6: Capsule Network Architecture

In order to identify fake images or videos from the real ones, the capsule network takes features obtained from the VGG-19 network. In the pre-processing step, the face is detected and scaled to the size of 128x128 this detected face is given input to VGG-19 to extract latent features for the capsule network, which consists of three primary capsules and two output capsules, one for fake and one for real images. The analytical pooling plays an essential part in the capsule network that deals with forgery detection.

2.2.3 MesoNet

In MesoNet architecture [43] author presented an algorithm to automatically and efficiently detect face manipulation in videos and mainly focuses on two latest techniques used to generate hyper-realistic forged videos: Deepfake and Face2Face. Conventional image forensics techniques are usually not well suited to videos due to the compression that strongly tampers the data. Thus, this method follows a deep learning perspective and presents two networks, which are dual with fewer layers to aim at the mesoscopic attributes of images. They check for those fast networks on both an existing dataset and a dataset we have constituted from online videos. The tests signify a very successful detection rate with more than 98% for Deepfake and 95% for Face2Face. There is two network architecture proposed in this Meso-4 and MesoInception-4.

2.2.3.1 MesoNet-4

This network starts with a sequence of four layers of successive convolutions and pooling and is succeeded by a dense network with one hidden layer. In order to improve, the convolutional layers use ReLU activation functions that establish non-linearity and Batch Normalization [10] to standardize their output, avoid the vanishing gradient effect, and improve their robustness and regularize the fully-connected layers uses Dropout [24].

In general, there are 27,977 trainable parameters in this proposed model for details in Figure 7.

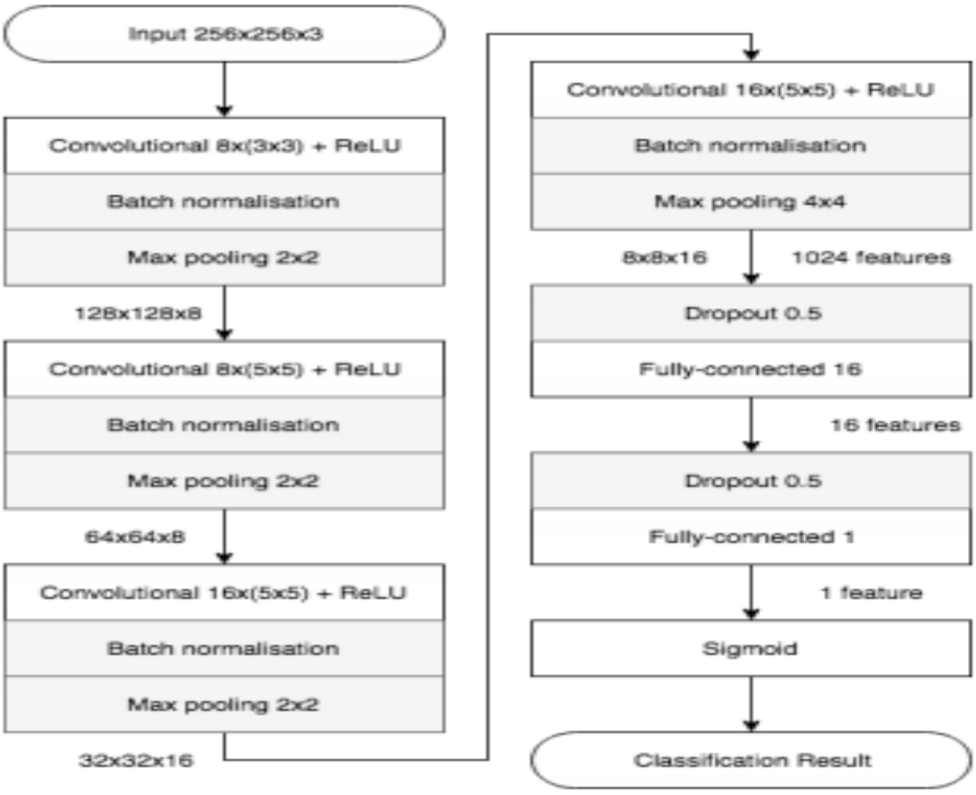

Figure 7. Meso-4 Network Architecture

2.2.3.2 MesoInception-4

Another method involves replacing the first two convolutional layers of Meso4 by a variant of the inception module. The goal of this module is to club the output of function space to optimize the model. In this model, 3 × 3 dilated convolutions [30] was used in order to avoid high semantic. This usage of the inception module, along with dilated convolution, can be found in [22] as a means to deal with multi-scale information. Still, we have added 1×1 convolutions before dilated convolutions for dimension reduction and an extra 1×1 convolution in parallel that acts as skip-connection between successive modules. Further details can be found in Figure 8.

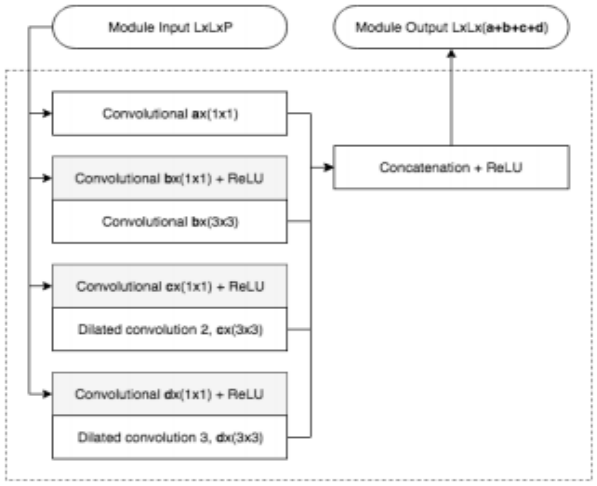

Figure 8. MesoInception-4 Network Architecture

2.2.4 Face Wrapping Artifacts

This method introduced by Yuezn Li [27] detects synthesized videos by exploiting the face warping artifacts resulted from the DeepFake production pipeline. For an adequate running time, the current DeepFake algorithms create synthesized face images of fixed sizes. These faces are then undergone an affine transform (i.e.,) scaling, rotation, and shearing) to match the poses of the target faces that they will replace (see Figure 9(a) – (d)).

As such, the facial region and surrounding regions in the original image/video frame will present artifacts. The resolution is inconsistent due to such transforms after the subsequent compression step to generate the final image or video frames. Therefore, we propose using a Convolutional Neural Network (CNN) model to detect artifacts from the detected face regions and surrounding areas.

First, detect faces in the original images and extract the face region using software package dlib. Second, we align faces into multiple scales and randomly pick one scale, which is then smoothed by a Gaussian blur with kernel size (5 × 5). This process aims to create more resolution cases in affine warped faces, which can better simulate different kinds of resolution inconsistency introduced in affine face warping.

The smoothed face undergoes an affine warp back to the same sizes of original faces to simulate the artifacts the same dimensions of original faces to reproduce the artifacts. In summary, popular deepfake detection and their features are presented in Table 2.

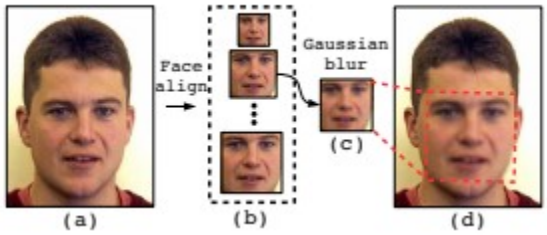

Figure 9 Face Wrapping Artifacts

Table 2: Deepfake Detection Methods

Methods	Key Features
Capsule Network	- VGG-19 Facial Feature Extraction - 3 Parallel CONV2D Network along with dynamic routing
MesoNet	- Meso-4 consists of 5x5 convolution, MesoIncpetion-4 3x3 convolution along with Inception layers - Faster training and good accuracy.
FaceWrapping Artifacts	- VGG-16 Facial Feature Extraction - Use the Gaussian Blur to detect face manipulation. - Require a large dataset.

CHAPTER 3. PROPOSED FRAMEWORK

The multi-modal deepfake detection is based on Siamese network-based architecture. In the proposed solution, we make use of visual and audio modality embeddings to identify given input video real or fake.

3.1 Framework Architecture

The architecture diagram shown in Figure 10 portrays how the entire multi-modal framework works. The framework is divided into two areas. First, of all, we take the real and fake videos of the same subject and extract the visual and audio features. Once the real and fake features are extracted successfully, they are passed on to the siamese convolution network for visual and audio learning. The visual and audio learning layer is made of four convolutional layers proceeded by ReLU function, maxpool, fully connected, and unit normalization layer. Once the visual and audio learning completes, it creates the 128 unit vector as output. The unit vector is then used to calculate the distance between the visual and audio modalities. The distance in the next step is used to calculate the triplet loss, which helps s identifying the video as real or fake.

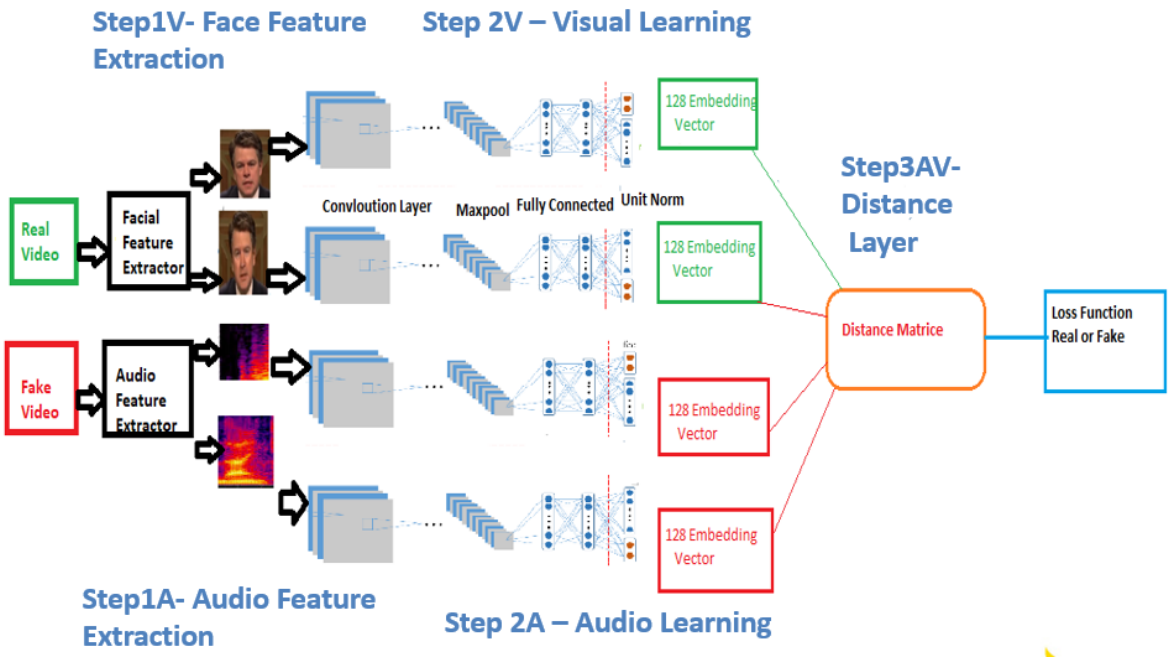

Figure 10:Multi-Modal DeepFake Detection

3.2 Facial Feature Extraction

The facial feature is extracted using the dlib and OpenCV library architecture, shown in Figure 11. The facial feature extraction involves multiple steps like face detect, landmark detection, transforming face for neural network, and crop the image. We see each step one by one below.

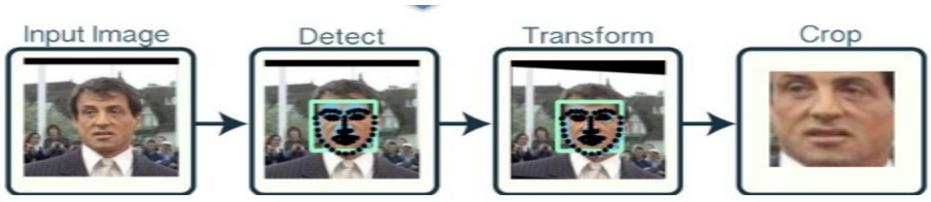

Figure 11: Facial Feature Extraction Steps

3.2.1 Detecting Face

The face is detected from the input image using OpenCV, which has inbuilt functionality based on Haar Cascade. Haar Cascade is an artificial intelligence object detection algorithm used to detect objects in a video or image based on the concept of features. The cascade function is trained using negative and positive images algorithm have four steps:

a. Haar Feature Selection: In Haar feature step adjacent rectangular regions at a specific location is considered in a detection window, then sums up the pixel intensities in each area and calculates the difference between these sums.

b. Creating Integral Images: An integral image enables you to calculate summations over image sub-regions rapidly. Sub-region summations can be calculated in constant time as a linear combination of only four pixels in the integral image, irrespective of the size of the sub-region. But most of these computed features are irrelevant.

c. Adaboost Training: The Adaboost training has two essential functions first to select the best features and train the classifiers that use them. Adaboost develops a "strong" classifier as a linear combination of weighted simple "weak" classifiers.

d. Cascade Classifier: The cascade classifier comprises a collection of steps, where each step is a collection of weak learners. They are simple classifiers, also called *decision stumps*. Each level is trained using a method called boosting. Boosting takes a weighted average of the weak learners' decisions and gives the ability to train the classifier with accuracy.

Every step of the classifier labels the region defined by the sliding window's current location as either negative or positive. *Negative* indicates no objects were found *positive* suggests that an object was found. For positive labels, the classifier passes the region to the next stage. For the negative label, the classification of this region is complete, and the detector slides the window to the following location. The detector informs about an object found at the current window location when the final steps classify the region as positive.

The steps are designed to reject negative samples as quickly as possible. The expectation is that the majority of windows do not contain the object of interest.

- A *true positive* appears when a positive sample is correctly classified.
- A *false positive* appears when a negative sample is mistakenly classified as positive.
- A *false negative* appears when a positive sample is mistakenly classified as negative.

To perform well, each step in the cascade must have a low false-negative rate. If a step incorrectly labels an object as negative, the classification stops, and you cannot correct the mistake. However, each step can have a high false-positive rate. Even if the detector incorrectly labels a no object as positive, you can fix it in subsequent steps. Adding more steps reduces the overall false-positive rate, but it also reduces the overall true positive rate.

Cascade classifier training needs a set of negative samples and a set of positive images. We must provide a collection of positive images with regions of interest specified to be used as positive samples. Image Labeler is used to label objects of interest with

bounding boxes. The output of the image labeler is a table to use for positive samples. We also must provide a set of negative images from which the function produces negative samples automatically. To achieve detector accuracy, set the number of steps, feature type, and other functional parameters.

3.2.2 Transforming Extracted Features

The transformation of the detected face is done using the affine transformation. It is any transformation performed using the matrix multiplication (linear transformation) followed by a vector addition (translation). Affine transformation is a relation between two images. The knowledge about this relation can come, in two ways:

a. Both X and *T*, and we also know that they are related. Then our job is to find M.

b. M and X. To obtain T, we only need to apply T = M. X Our information may be explicit (i.e., have the 2-by-3 matrix), or it can come as a geometric relation between points.

3.3 Audio Feature Extraction

The audio features are extracted from the wav file using the library based on LSTNet architecture. LSTNet architecture show below in Figure 12. Various components of LSTNet are described one by one below.

3.3.1 Convolution Component

LSTNet first layer is a convolutional network without pooling, whose goal is to extract short-term patterns in the audio wave as well as local dependencies between variables. The convolution layer consists of multiple filters of width w and height h, where the number of variables is the same. The k-th filter slides through the input matrix X and produces

$$h_k = \text{RELU}(W_k * X + b_k) \quad ()$$

The input is dynamically padded, depending on the height of the kernel. It helps each input slides over the input data. The convolution layer's output is of size d x T, where d denotes the number of filters.

3.3.2 Recurrent Component

The output of the convolution component is fed to the recurrent component in parallel. The recurrent component is the recurrent layer with RELU function hidden and with Gated recurrent unit. Each gates unit in the input of shape = (batch size, the total number of filters). The hidden state of each recurrent unit and any time t is calculated as

$$r_t = \sigma(x_t W_{xr} + h_{t-1} W_{hr} + b_r)$$

$$u_t = \sigma(x_t W_{xu} + ht-1 W_{hu} + b_u)$$

$$c_t = \text{RELU}\ (x_t W_{xc} + r_t \odot (h_{t-1} W_{hc}) + b_c) \quad ()$$

$$h_t = (1 - u_t) \odot h_{t-1} + u_t \odot c_t$$

Here $\odot$ is an element-wise product, σ is a sigmoid function, and X_t is the input layer at time t.

3.3.3 Temporal Attention Layer Component

The Recurrent-skip layer needed a predefined hyperparameter p, which is unfavorable for audio wave frequency series, or whose period length is dynamic over time. To resolve such an issue, we consider an alternative approach, attention method, which learns the weighted combination of the hidden input matrix. The attention weight at current time t is calculated as

$$a_t = AttnScore(H_t, h_{t-1})$$

H_t is a matrix hidden representation of recurrent network column-wise, and attnScore is a similarity function.

3.3.4 Autoregressive Component

The convolution and recurrent neural network can handle the constant change in real audio wave inputs in a non-periodic manner. In order to address this situation, the autoregressive model as a linear component was introduced in LSTNet. The autoregressive component primarily focused on scaling issues and adopted as a linear component.

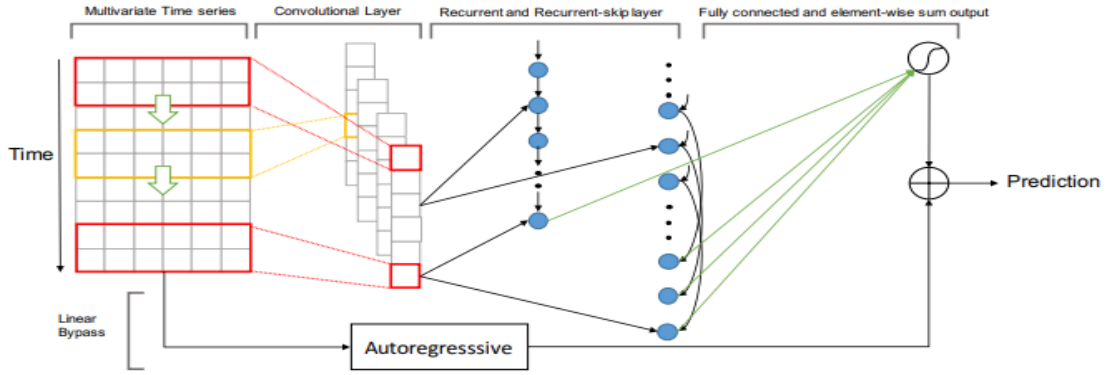

Figure 12: LSTNet Architecture

3.4 Visual and Audio Learning Network

The proposed architecture consists of a convolution network for audio and video learning. It is shown in Figure 13. The network is kept at the same weight as a model need to find the similarity rather than classifying. The input size to the network is of size 224 x 224.

There are four convolution network layers in proposed architecture with RELU to extract the meaning full feature from the input image. The architecture also contains maxpool layers, fully connected layers along with sigmoid function, and unit normalization layer. The convolution layer consists of multiple filters of width w and height h. The Filter sweeps through input matrix X and produce

$$h = \text{RELU}\ (W * x + b)$$

Where * denotes convolution operation h would be the vector and RLEU function is

$$\text{RELU}(x) = \max\ (0, x)$$

The unit vector produced by unit normalization layer used calculated the dissimilarity distance using distance function like euclidean or absolute distance.

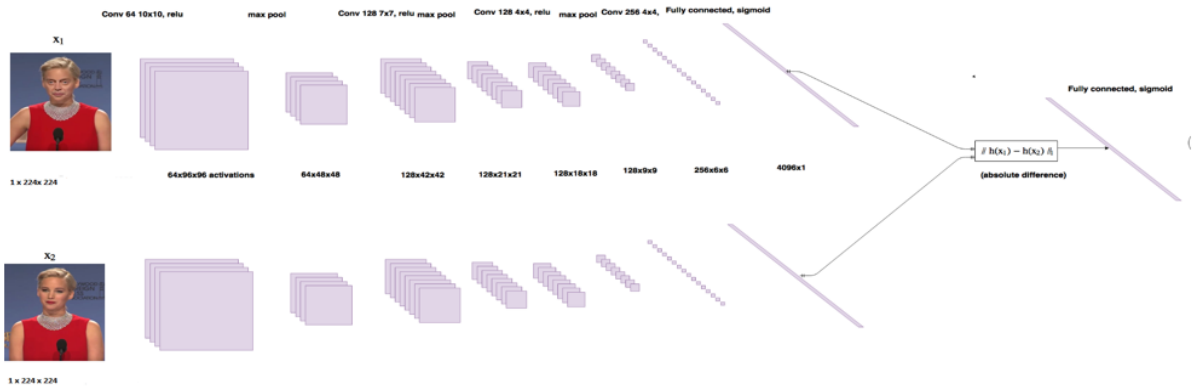

Figure 13: Visual-Audio Learning Network Architecture

3.5 Constructive Loss

The main goal of the siamese architecture is not to classify input images, but to find dissimilarity between them. So, a classification loss function (such as cross-entropy) would not be suitable for such a condition. Instead, siamese architecture is best suited to make use of a contrastive function. Intuitively, this function just evaluates how well the network is distinguishing a given pair of images.

The contrastive loss function:

$$(1-Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{max(0, m - D_W)\}^2 \quad (2)$$

Dw is denoted as the euclidean distance between the outputs of the sister siamese networks.

$$\sqrt{\{G_W(X_1) - G_W(X_2)\}^2} \quad (3)$$

3.6 Triplet Loss

Triplet loss is a loss function for a convolution neural network where baseline anchor input is compared to the truth(positive) and false(negative). The main aim of triplet loss is to minimize the distance between baseline(anchor) input to the positive (truth) input and to maximize the distance from the baseline(anchor) input to the negative (false) input

It often used for learning similarity for learning embeddings, like word embedding and even though vectors and metric learning.

Triplet Loss function can be described using Euclidean distance as follows.

$$L = \max(d(a,p) - d(a,n) + \text{margin}, 0) \quad (4)$$

The variable "a" represents the anchor image, "p" represents a positive image, and "n" represents a negative image. Another variable is called margin, which hyperparameter added to triplet loss. Margin defines how far dissimilarity should be i.e if margin = 0.3 and d(a,p) = 0.6 then d(a,n) should at least be equal to 0.9. Margin helps us distinguish the two images better.

As shown in Figure 14 input image of Obama is acting as the anchor(a), and again the image of Obama is acting as positive(p), and Emmanuel is acting as negative(n).

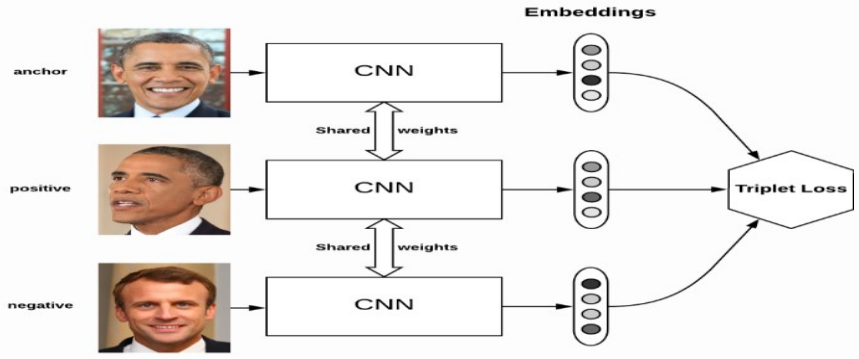

Figure 14 Triplet Loss

3.7 Video and Audio Modality Embedding

V_1 and A_1 are neural networks that we use to learn the unit-normalized embeddings for the face and speech modalities, respectively. In Figure 13, we depict V_1 and A_1 in both training and testing routines. They are composed of 2D convolutional layers, max-pooling layers, and fully connected layers. ReLU activation function is used between all layers. The last layer is a unit-normalization layer. For both face and speech modalities, V1 and A1 return 250-dimensional unit-normalized embedding.

Training: $m_{vreal} = V_1(f_{real}), m_{vfake} = V_1(f_{fake}), m_{areal} = A_1(a_{real}), m_{afake} = A_1(a_{fake})$

Testing: $m^f = V_1(f), m^s = A_1(a)$ (5)

3.8 Training Pipeline

In the training phase, we used a real and a fake video of the same subject as the input. After passing extracted visual and audio features from raw videos (f_{real}, f_{fake}, a_{real}, a_{fake}) through V_1 and A_1, we obtain the unit-normalized modality as described in Eq. 6. Considering an input fake and real video, we first compare f_{real} with f_{fake} and a_{real} with a_{fake} to understand what modality was manipulated more in the fake video. We recognize the face modality to be manipulated more in the fake video, inferring from these embeddings we can calculate the first similarity between the fake and real speech and face embeddings as follows:

Similarity Score: $SS = d(m_{vreal}, m_{areal}) - d(m_{afake}, m_{vfake})$ (6)

Where d denotes the Euclidean distance.

SS is computing the distance between two pairs d (m_{vreal}, m_{areal}) and d(m_{afake}, m_{vfake}). We expect m_{vreal}, m_{areal} to be closer to each other than m_{vreal}, m_{afake} as it contains a fake face modality. The triplet loss notation for training our model is given in equation 7.

Similarity Loss: $L = \max(SS + m, 0)$ (7)

where m is training convergence margin.

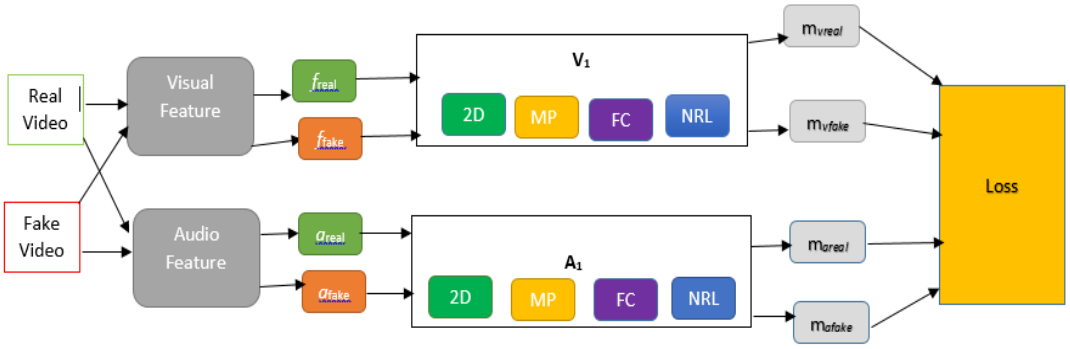

Figure 15: Training Pipleine

3.9 Testing Pipeline

In the testing phase, we only have a single input video that is to be labeled real or fake. After extracting the features, *f,* and *a* from the raw videos, we perform a forward pass through V_1 and A_1, as depicted in Figure 16 to obtain modality embedding.

To conclude about real and fake, we compute the following two distance values:

$$d = d\,(m^{f}, m^{s}) \tag{8}$$

To differentiate between real and fake, we compare d with a threshold, that is, τ empirically learned during training as follows:

$$d > \tau$$

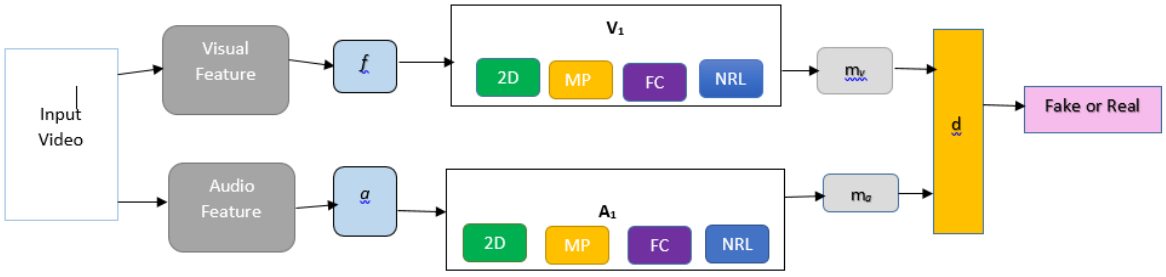

Figure 16: Testing Pipeline

CHAPTER 4. RESULTS AND EVALUATIONS

4.1 Introduction

In this section, we discuss the results and evaluation of the proposed framework. First, we describe the results of Unimodal Video and Audio in terms of accuracy. Second, we show the accuracy of the multi-modal invariant of video and audio as the anchor in terms of triplet loss and compare it with the start of the art deep fake detection methods.

4.2 Dataset

With the popularity of deepfake detection has increased considerable attention, and this research has been stimulated with many datasets. We summarize and analyze three benchmark deepfake video detection datasets in Table 2.

Table 3: Benchmark Dataset for DeepFake Video Detection

Dataset	**Released**	**Videos**			**Modes**	
		Real	**Fake**	**Total**	**Visual**	**Audio**
DFDC[22]	Oct 2019	19,154	99,992	119,146	Yes	Yes
CelebDF[17]	Nov 2019	408	795	1,203	Yes	No
Face Forensic++[13]	Jan 2019	1,000	4,000	5,000	Yes	No

4.2.1 DFDC

DFDC consists of videos from two facial modification algorithm. Also, DFDC does not disclose details of the AI model used to synthesize the fake videos from the real videos. Only the DFDC dataset contains a mix of videos with manipulated audio, faces, or both. All the other benchmark datasets contain only manipulated faces. Also, only DFDC includes both audio and video, allowing us to analyze both modalities. Show in Figure 17

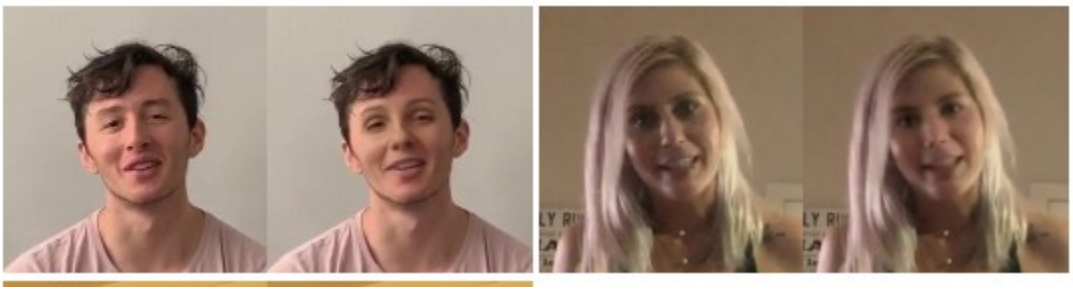

Figure 17 DFDC Face Swap Example

4.2.2 CelebDF

The celeb-DF dataset consists of real and DeepFake synthesized videos. The primary technique used for the creation of these videos is Face masking, Color mismatch, and low-resolution synthesis. To date, Celeb-DF includes 408 original videos collected from YouTube with subjects of different ages, ethnic groups, and genders, and 795 corresponding DeepFake videos. Shown in Figure 18

Figure 18 CelebDF Example

4.2.3 Face Forensic++

In Face Forensic++[18] author published the dataset consists of forged images from various techniques like FaceSwap, Face2Face Deepfake, and Neural Texture. The data has been gathered from 977 YouTube videos, and all videos contain a trackable mostly frontal face without occlusions, which enables automated tampering methods to generate realistic forgeries. They provide binary masks; the data can be used for image and video classification as well as segmentation. Also, they offer 1000 Deepfakes models to generate and augment new data. Shown in Figure 19

Figure 19 Face Forensic++ example.

4.3 Results

4.3.1 Unimodal Video Result

For Unimodal Visual modality, we received the accuracy of 90.4% in Table 1. As shown in Figure 20, the dissimilarity score for two real faces from the subject is 0.28, which results in a lower constructive loss of 0.0014. When the real and fake face of the same subject compared dissimilarity score increases, 2.37 and higher constructive loss vale are retrieved.

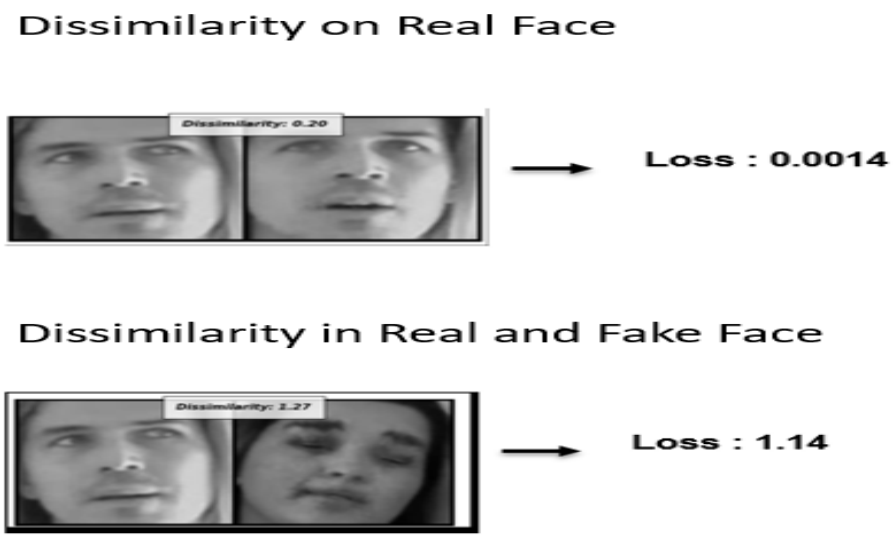

Figure 20: Visual Dissimilarity Score

Table 4: UniModal Audio and Visual Accuracy

Methods	Accuracy
Audio Modality	91.3
Visual Modality	90.4

As shown in Figure 21, it can be inferred that visual constructive loss during training decreases with an increase in the training epoch cycle.

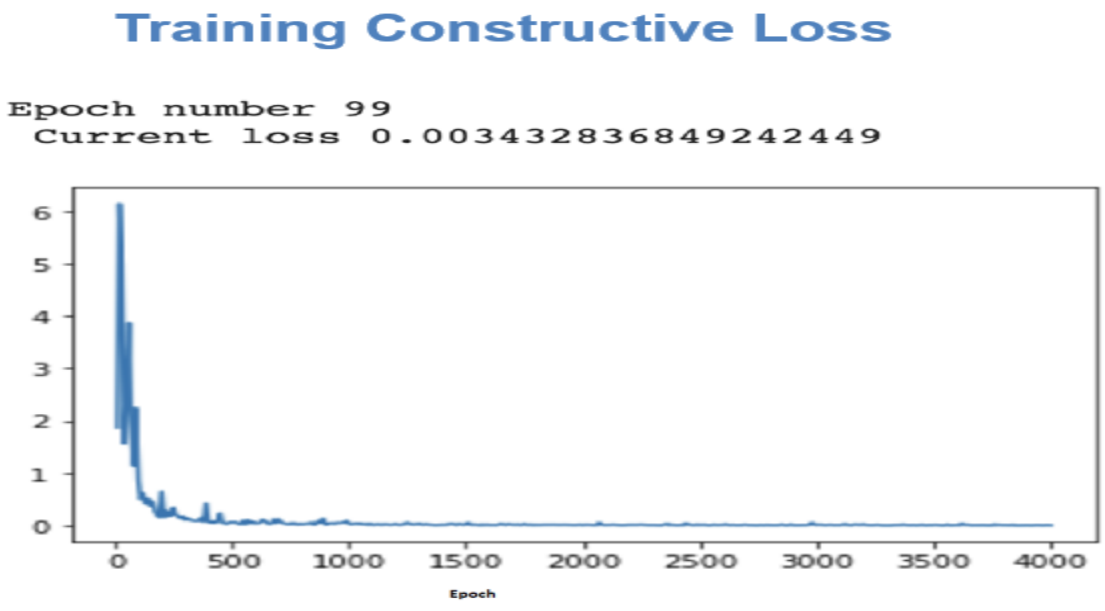

Figure 21: Visual Constructive Loss

4.3.2 Unimodal Audio Result

For Unimodal Visual modality, we received the accuracy of 91.3% in Table 1. As shown in Figure 22, when real and fake MFCC spectrograms of the same subject compared dissimilarity score increases, 2.62 and higher constructive loss vale are experienced.

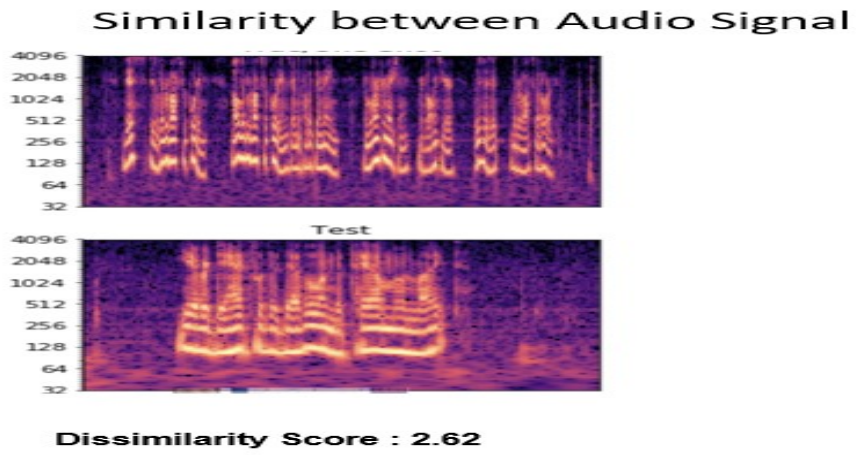

Figure 22: Audio Dissimilarity Score

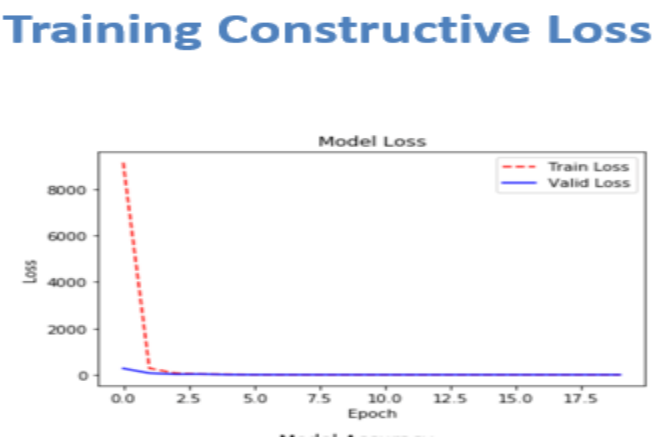

Figure 23: Audio Constructive Loss

4.3.3 Multi-modal Audio as Anchor

We report and compare per-video the accuracy of our method against three prior deepfake video detection methods on DFDC. We have summarized these results in Table 4. The following are the previous state-of-the-art methods used to compare the performance of our approach on the benchmark datasets.

Table 5: Multi-Modal Audio Anchor Accuracy

Methods	**Accuracy**
Meso4	87.8
Capsule	78.9
FWA	88.2
Our Method Audio Anchor	91.3

As shown in Figure 24, with audio being the anchor, the dissimilarity score of the real(positive) face is 0.34 and helps in achieving lower triplet loss. While on the other hand, the fake(negative) face has a dissimilarity score of 0.78 and results in substantial triplet loss value.

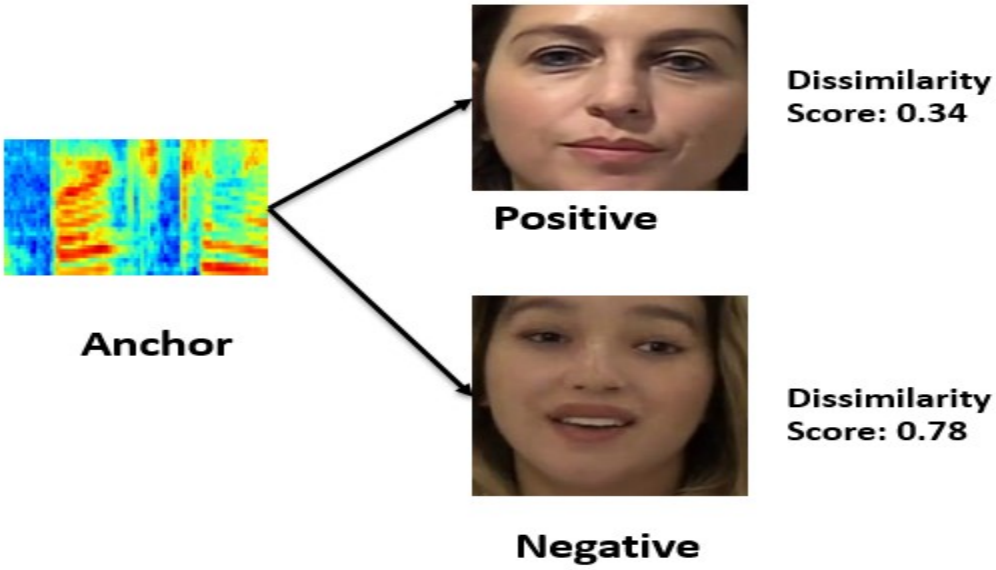

Figure 24: Audio Anchor Dissimilarity Score

As shown in Figure 25, the training and validation triplet loss value decreases with an increase in the epoch cycle.

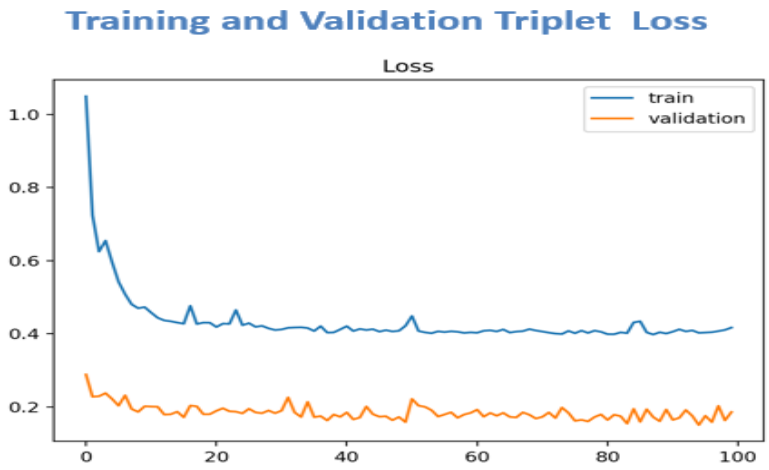

Figure 25: Audio Anchor Triplet Loss

4.3.3 Multi-modal Video as Anchor

We report and compare per-video the accuracy of our method against three prior deepfake video detection methods on DFDC. We have summarized these results in Table 5. The following are the former state-of-the-art methods used to compare the performance of our approach to the same datasets.

Table 6: Multi-Modal Video Anchor Accuracy

Methods	**Accuracy**
Meso4	87.8
Capsule	78.9
FWA	88.2
Our Method Video Anchor	90.2

As shown in Figure 24, with visual being the anchor, the dissimilarity score of real(positive) audio is 0.27 and helps in achieving lower triplet loss. While on the other hand, fake(negative) audio has a dissimilarity score of 0.62 and results in substantial triplet loss value.

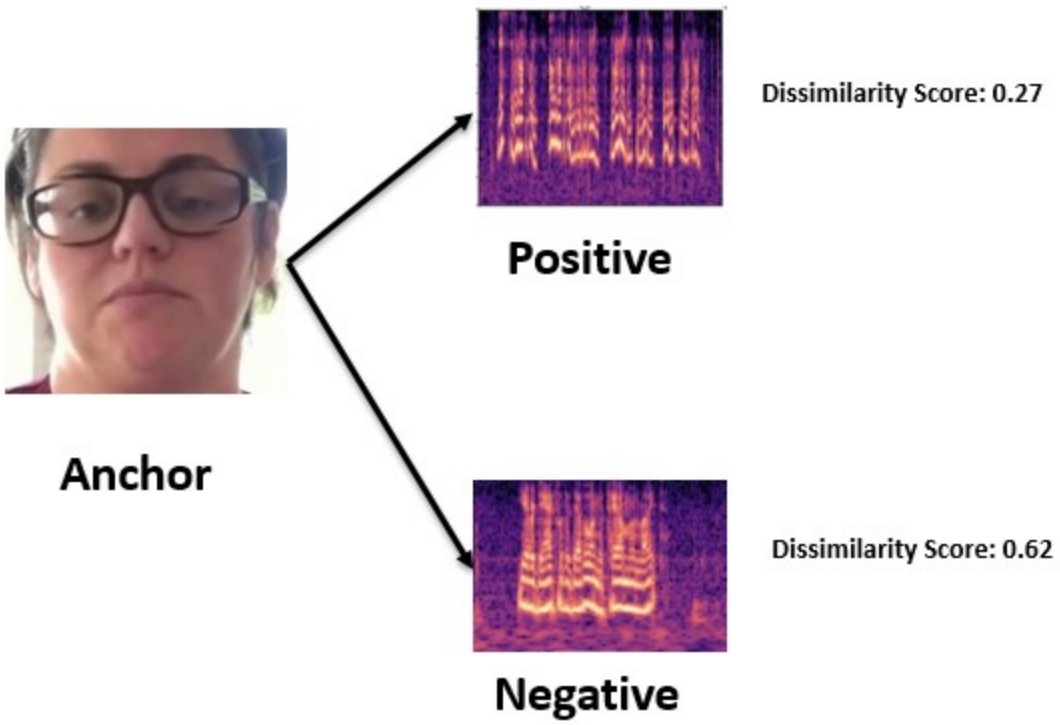

Figure 26: Video Anchor Dissimilarity Score

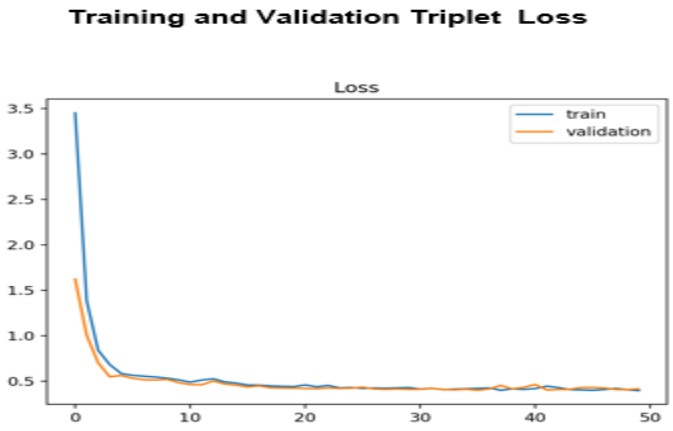

Figure 27: Video Anchor Triplet Loss

4.3.4 Interpreting the correlation of Visual and Audio Modality

We plot the distance between the speech embeddings and unit-normalized face learned from V_1 and A_1 1, 000 randomly chosen points from the benchmark train set. We plot $d(m_{vreal}, m_{areal})$ in blue and $d(m_{vreal}, m_{afake})$ in orange. The bulk of the subjects from real videos have a smaller distance, 0.2 between their embeddings as compared to the fake videos (0.5).

We plotted the percentage of subject videos versus the distance between the speech and face modality embeddings. Figure 28 shows that the fake videos (orange curve) are distributed around a higher distance center (0.5). In contrast, the distribution of real videos (blue curve) is centered around a lower modality embedding distance (0.2)

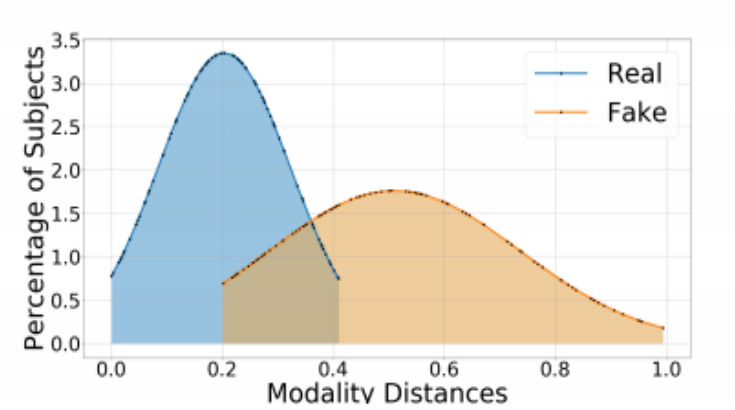

Figure 28: Modality Embedding Distance

We show that visual-audio modalities are more similar in real videos as compared to fake videos.

CHAPTER 5. CONCLUSION, LIMITATION AND FUTURE WORK

5.1 Conclusion

In this thesis, we present a siamese-based method for identifying fake videos. We use the likeliness between audio-visual modalities and the similarity between the useful clues of the two modalities to conclude whether a video is "real" or "fake." We evaluated our method on benchmark audio-visual deepfake datasets, called DFDC, CelebDF, and Face Forensic++.

5.2. Limitation

Our approach has some drawbacks. First, our approach could infer in misclassifications on both the datasets, as compared to the one in the real video. Furthermore, many of the deepfake datasets mainly contain one or more person per video. We may have to extend our approach to take into consideration the emotional state of one or more people in the video and come with a possible scheme for deepfake detection.

5.3 Future Work

In the future scope of this subject, we would like to focus on incorporating more modalities that are evenly contexted to conclude whether a video is a deepfake or not. We would like to amalgamate our approach with the existing ideas of detecting visual artifacts crossed over frames for better execution and performance. Additionally, we would like to approach more elegant methods for using audio cues.

BIBLIOGRAPHY

[1] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition? assessment and detection," *CoRR,* vol. abs/1812.08685, pp. 3-4, 2018.

[2] D. Citron, "How deepfakes undermine truth and threaten democracy," 13 09 2019. [Online]. Available: https://www.bu.edu/law/2019/09/13/how-deepfakes-undermine-truth-and-threaten-democracy/.

[3] R. C. Jones, "Deepfake videos double in nine months," 7 10 2019. [Online]. Available: https://www.bbc.com/news/technology-49961089.

[4] B. Zhang, "Deepfakes: what are they, and why would I make one," 2019. [Online]. Available: https://www.bbc.co.uk/bitesize/articles/zfkwcqt.

[5] E. Howcroft, "How faking videos became easy and why that's so scary," Bloomberg, 2018. [Online]. Available: https://www.bloomberg.com/news/articles/2018-09-10/how-faking-videos-became-easy-and-why-that-s-so-scary-quicktake.

[6] R. Chesney and D. Citron, "Deepfakes and the new disinformation war: the coming age of post-truth geopolitics," Foreign Affairs, [Online]. Available: https://www.foreignaffairs.com/articles/world/2018-12-11/deepfakes-and-new-disinformation-war.

[7] B. Marr, "The best (and scariest) examples of ai-enabled deepfakes," Forbes, 22 07 2019. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2019/07/22/the-best-and-scariest-examples-of-ai-enabled-deepfakes/.

[8] E. Zakharov, A. Shysheya, and E. Burkov, "Few-Shot adversarial learning of realistic neural talking head models," *CoRR,* vol. abs/1905.08233, 2019.

[9] J. Damini, "A voice deepfake was used to scam a CEO out of $243,000.," Forbes, 3 09 2019. [Online]. Available: https://www.forbes.com/sites/jessedamiani/2019/09/03/a-voice-deepfake-was-used-to-scam-a-ceo-out-of-243000/#d5221dc22416.

[10] S. Samuel, "Vox Media," Vox, 27 June 2019. [Online]. Available: https://www.vox.com/2019/6/27/18761639/ai-deepfake-deepnude-appnude-women-porn.

[12] M. Stamm, "Forensic detection of image manipulation using intrinsic statistical fingerprints," *IEEE Transactions o Information Forensics and Security,* pp. 492-506, 2010.

[13] A. Rosselar, "FaceForensics++: learning to detect manipulated facial images, " in Proc. International Conference on Computer Vision," 2019.

[14] J.Stehowuer and H. Dang, "On the detection of digital face manipulation," in *IEEE Conference Computer Vision and Pattern* 2019.

[15] J. Galbally and S. Marcel, "Biometric anti-spoofing methods: a survey in face recognition," *IEEE Acess,* pp. 1530-1552, 2014.

[16] A. Hadid and N. Evans, "Biometrics systems under spoofing attack: an evaluation methodology and lessons learned," *IEEE Signal Processing Magazine,* 2015.

[17] Y. Li, "Celeb-DF: a large-scale challenging dataset for deepfake forensics," [Online]. Available: https://arxiv.org/pdf/1909.12962.pdf, 2019.

[18] V. Vera-Rodriguez and R. Tolosana, "Real or Fake? Spoofing state-of-the-art face synthesis detection systems," *IEEE Vision,* 2019.

[19] C. Canton and L. Davis, "Applications of Computer Vision and Pattern Recognition to Media Forensics," in *Conference on Computer Vision and Pattern Recognition*, 2019.

[20] P. Korshunov and T. Mensink, "Deepfake using STOG methods," in *International Conference on Machine Learning*, 2019.

[21] P. Bestagini and L. Verdoliva, "Multimedia Forensics," ACM Multimedia, 2019. [Online]. Available: https://acmmm.org/tutorials/.

[22] "DFDC facebook deepfake dataset," Facebook, 2019. [Online]. Available: https://arxiv.org/pdf/1910.08854.

[23] "ASVSpoof 2019: Future Horizons in Spoofed and Fake Audio Detection [Online]," Google, 2019. [Online]. Available: https://arxiv.org/abs/1904.05441.

[24] M. Kowalski, "FaceSwap," 2019. [Online]. Available: www.github.com/MarekKowalski/FaceSwap.

[25] J. Thesis, M. Zollhofer, and M. Stamminger, "Face2face: Real-time face capture and reenactment of rgb videos," in *IEEE Conference on Computer Vision and Patterns Recognition*, 2016.

[26] Nugyen H, " Capsule-forensics: Using capsule networks to detect forged images and videos," in ICASSP 2019-2019 IEEE International Conference on Speech, Acoustics and Signal Processing (ICASSP) 2019 May 12 (pp. 2307-2311). IEEE.

[27] D.Afchar, " Mesonet: a compact facial video forgery detection network," 2018 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE, 2018.

[28] L. Sangyup and T. Shahroz, "Detecting both machine and human-created fake face images in the wild," in *2nd International Workshop on Multimedia Privacy and Security*, 2018.

[29] F. Marra and D. Gragnaniello, "Detection of gan-generated fake images over social networks," in *IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2018.

[30] M. Nießner and C. Riess, "A large-scale video dataset for forgery detection in human faces," *IEEE ComputerVision and Pattern,* 2018.

[31] L. Verdoliva and D. Cozzolino, "Faceforensics++: Learning to detect manipulated facial images," *IEEE ComputerVision and Pattern,* 2019.

[32] X. Xuan and W. Wang, "On the generalization of gan image forensics." *IEEE Conference on Computer Vision and Patterns Recognition*, 2016

[33] D. Maignan and L. Yuening, "Towards generalizable forgery detection with locality-aware autoencoder," *arXiv preprint arXiv:1909.05999,* 2019.

[34] Y. Bengio, and Y. LeCun, *Convolutional networks for images, speech, and time-series*, MIT Press, 1995.

[35] A. Dertat, "Applied deep learning - Part 4: convolutional neural networks," Toward DataScience, [Online]. Available: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2.

[36] P. Jain, "Complete guide of activation functions," [Online]. Available: https://towardsdatascience.com/complete-guide-of-activation-functions-34076e95d044.

[37] C. Francois, "Keras," 2015. [Online]. Available: https://github.com/fchollet/keras.

[38] T. Lin and M. Maire, "Microsoft-COCO: common objects in context," in *ECVV*, 2014.

[39] O. Russakovsky, "Imagenet large scale visual recognition challenge," [Online]. Available: http://www.image-net.org.

[40] S. Lui, "Very deep convolutional neural network-based image classification using small training sample size," in *IAPR Asian Conference on Pattern Recognition*, 2015.

[41] K. He and X. Zhang, "Deep residual learning for image recognition," *arXiv preprint arXiv: 1512.03385v1,* 2015.

[42] C. Szegedy, "Rethinking the inceptions architecture for computer vision," in *IEEE Conference Computer Vision Pattern Recognition*, 2016.

[43] D.Afchar, "Mesonet: a compact facial video forgery detection network," *in IEEE International Workshop on Information Forensics and Security*. 2019

[44] L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," *arXiv preprint arXiv: 1801.04381,* 2018.

[45] F. Jueves-Xu, "FakeSpotter: a simple baseline for spotting ai-synthesized fake faces," *arXiv preprint arXiv:1909.06122,* 2019.

[46] J. Kodovsky, "Rich model for steganalysis of digital images," in *IEEE Transactions on Information Forensics and Security*, 2012.

[47] M. Goljan, "CFA aware features for steganalysis of color images," in *International Society for Optics and Photonics*, 2015.

[48] S. Lu, "Exposing image splicing with inconsistent local noise variances," in *IEEE International Conference on Computational Photography*, 2012.

[49] . N. Rahmouni, "Distinguishing computer graphics from natural images using convolution neural networks," in *IEEE Workshop on Information Forensics and Security (WIFS)*, 2017.

[50] B. Bayar and M. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *4th ACM Workshop on Information Hiding and Multimedia Security*, 2016.

[51] L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection," in *5th ACM Workshop on Information Hiding and Multimedia Security*, 2017.

[52] X. Xie, "FakeSpotter: a simple baseline for spotting ai-synthesized fake faces," *arXiv preprint arXiv:1909.06122,* 2019.

[53] F. Marra, "Incremental learning for the detection and classification of gan-generated images," *arXiv preprint arXiv:1910.01568,* 2019.

VITA

Raju Nekadi completed his bachelor's degree in Computer Science from the University of Technology, Indore, India. After his bachelor's degree, he worked as a software engineer at Capgemini Pune, India, for three years and later moved to client MetLife Auto Home, Rhode Island USA. He started his master's program in Computer Science Electrical Engineering at the University of Missouri-Kansas City in fall 2018. During his master's program, along with maintaining a full-time job as a Software Engineer, he has worked on various research and theories and also won the third prize for RJI Student Innovation held at Reynold Journalism Institute, Columbia. He is currently working with client Sun Life in downtown Kansas City from the past four years. After completing his master's degree, he plans to pursue a Ph.D. degree in Computer Science.