



NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

School of Electrical Engineering and Computer Sciences

DISCERNING DEEPPAKE VIDEOS

Final Year Project Report

Muneeb ur Rehman
Muhammad Anas Tahir
Zaryab Muhammad Akram

In Partial Fulfillment of the Requirements for the degree
Bachelor of Science in Computer Science (BSCS)
School of Electrical Engineering and Computer Science National University of
Sciences and Technology
Islamabad, Pakistan

DECLARATION

We hereby declare that this project report entitled “**Discerning Deepfake Videos**” submitted to the *School of Electrical Engineering and Computer Science* is a record of an original work done by us under the guidance of Supervisor *Dr. Muhammad Imran Malik* and that no part has been plagiarized without citations. Also, this project work is submitted in the partial fulfillment of the requirements for the degree of *Bachelor of Computer Science*.

Team Members:

Muneeb ur Rehman – BSCS-7C

Muhammad Anas Tahir – BSCS-7C

Zaryab Muhammad Akram – BSCS-7C

Advisors:

Advisor Name: Dr Muhammad Imran Malik

Co-Advisor Name: Dr. Muhammad Shehzad

DEDICATION

This thesis is dedicated to our parents, siblings, friends, professors and mentors who helped us learn and grow throughout the course of our degree, without whom, we would not have been here, and who have shown support and trust in us in all endeavors of our life. We also dedicate this thesis to our seniors for their encouragement that has enabled us to achieve everything in our lives.

ACKNOWLEDGEMENTS

Foremost, we are extremely grateful to our advisor *Dr. Muhammad Imran Malik* and our co-advisor *Dr. Muhammad Shehzad* for their continued support throughout the project. Their suggestion and feedback have allowed us to deliver on our work and this thesis would not have been possible had it not been for their mentorship and expert knowledge.

Besides our advisors, we would also like to thank *Dr. Faisal Shafait* for inspiring us and enabling us to get started with Machine Learning and Deep Learning research through his mentorship.

We would also like to mention *TUKL Research and Development Lab (NUST)* for providing us with the required computational machines for this project. Their help enabled us to run our computationally expensive experiments with ease.

TABLE OF CONTENTS

DECLARATION	2
Dedication	3
ACKNOWLEDGEMENTS	4
TABLE OF CONTENTS	5
LIST OF FIGURES	7
LIST OF TABLES	9
ABSTRACT	10
Introduction	11
What are deepfakes:	12
Why deepfake detection is important:	12
Malicious usages:	13
Politics:	13
Businesses:	13
Other malicious usages:	14
Deepfake identification and problems	14
Motivation	15
Summary	16
Background on deepfake generation	17
Video manipulation and deepfake generation	18
How are deepfakes made?	19
Deepfake datasets	25

Celeb df:	26
Faceforensic++	27
Deeper Forensic	28
Wild deepfake:	29
Previous work in deepfake detection	30
Classical computer vision approaches for video alteration:	31
Deep learning-based approaches	32
Methodology	34
Data Preprocessing	35
Facial masks	36
Using face detection	37
Classification through Neural Network	38
Convolutional-LSTM (Baseline)	39
Y-shaped Architecture	40
Vision Transformer based Architecture	42
Inception-ConvRNN Architecture	43
Web Application	53
User Interface (Frontend):	54
Backend	57
Application Structure:	58
Conclusion	59
References:	61

LIST OF FIGURES

Figure 1: Neural Network.....	18
Figure 2: Autoencoder [43].....	20
Figure 3: Training	20
Figure 4: Deepfake generation.....	21
Figure 5: X.hou [45] shows the image reconstruction using VAE	22
Figure 6: Generative adversarial network	23
Figure 7: Images provided by karras [48]	24
Figure 8: Images from celebdf	26
Figure 9: User accuracy as presented in [51]	27
Figure 10: Deepfake variational autoencoder presented in [49]	28
Figure 11: Feature comparison of different dataset as presented in [50]	29
Figure 12: A sample frame from face forensic data set	36
Figure 13: Mask for image in fig 12.....	36
Figure 14: After preprocessing	37
Figure 15: Convolutional-LSTM	39
Figure 16: Y-shaped architecture	41
Figure 17: Vision transformer-based architecture	42

Figure 18: Proposed ConvRNN cell	44
Figure 19: Unrolled ConvRNN layer.....	44
Figure 20: Proposed Inception-ConvRNN Architecture	45
Figure 21: Meso Inception encoding module	47
Figure 22: Micro Inception encoding module	48
Figure 23: Comparison of AUC for meso and micro module	49
Figure 24: Train and validation AUC scores for Celeb-DF.....	51
Figure 25: Train and validation AUC scores for FaceForensics++ HQ.....	51
Figure 26: Train and validation AUC scores for Deeper Forensics	52
Figure 27: User interface for portal.....	54
Figure 28: Upload video.....	55
Figure 29: Threshold adjustment.....	56
Figure 30: Uploaded video	56
Figure 31: result	57
Figure 32: Application structure	58

LIST OF TABLES

Table 1: Convolutional-LSTM results	40
Table 2: Y-shaped architecture results	41
Table 3: Vision transformer-based architecture result	43
Table 4: Number of filters for each convolution in Inception modules used in meso-ConvRNN	46
Table 5: Number of filters for each convolution in Inception module used in micro-ConvRNN	48
Table 6: Inception-ConvRNN results	49
Table 7: Comparison of detection performance of different methods using AUC score.....	50
Table 8: Application dependencies	55

ABSTRACT

Deep learning has been successfully applied to solve various complex problems ranging from computer vision to human-level intelligence. Recent deep learning advances, however, have also been employed to create tools that have made easier than ever to create synthetic media that is indistinguishable to naked eye.

Such tools are now being used negatively in the form of **deepfake**, which has raised severe societal concerns. Deepfake can create fake images and videos that humans cannot distinguish from authentic ones. With easy access to such technology malicious usages are increasing, and deepfakes are becoming more and more common on social media platforms.

Lately, several detection models have been proposed to discern a manipulated video or imagery from an original one. Some deep learning models have shown good performance and robustness in this task. However, most of these are mammoth models with millions and millions of parameters. Such models have huge computational costs and cannot be used for fast inference in real-world scenarios. Additionally, most of these methods work with images and discard important temporal information in the videos.

We present a novel **Recurrent Convolutional layer** and proposes two lightweight deepfake detection models:

1. **Micro-Inception CRNN** with 9,247 parameters
2. **Meso-Inception CRNN** with 17,237 parameters.

Experiments show that these models can efficiently use the Spatial-temporal information from videos and achieve state of the-art results on recent deepfake datasets.

INTRODUCTION

WHAT ARE DEEPPAKES:

Deepfakes are computer synthesized images and videos where a person could be speaking words or performing actions which he or she may have never spoken or done.

Fake images and videos are not something new; however, deepfakes rely on powerful neural networks to synthesize images, videos, and even voice in a way that a normal human being can not differentiate from real. This artificial intelligence based fake videos now commonly known as deepfakes are created using deep learning algorithms such as autoencoders or generative adversarial networks (GANs), and by using a few other video processing tools the product of these algorithms is almost indistinguishable to the naked eye.

WHY DEEPPAKE DETECTION IS IMPORTANT:

As the world is rapidly digitizing and the amount of digital data is increasing exponentially, therefore there is a growing need for fact checking and identification of malicious actors on the internet. With the advent of social media platforms spreading misinformation via fake news, documents, images, and videos has become very easy, and it is becoming very difficult for a novice user to identify misinformation.

Fake news, disinformation and propaganda news are spreading like wildfire on the internet these days. The latest tool and the one which can wreak havoc in near future is deepfake. Videos and images are considered to be the most trustworthy and irrefutable form of evidence, as any alteration or change in it could be easily identified previously with just an eye. However, deepfake, a new technology which has leveraged the power of deep learning and is helping in creation of synthetic images and videos which a naked eye cannot discern from real images and videos.

While there are some useful usages of this technology, the harmful and malicious usages far outweigh the positive aspects of this technology. Recently, many negative usages of this technology have emerged, and bad actors have used it for blackmailing, misinformation, and etc.

MALICIOUS USAGES:

Over the past 2-3 years the quality of deepfakes have improved exponentially and as a result this has prompted responses from the governments and companies across the world. Even though the technology is relatively new, these fake videos have been used as a means of spreading disinformation and extortion on numerous occasions.

POLITICS:

Deepfake technology can become a powerful tool in spreading politically motivated lies. Fake videos can be quickly created using this technology and using the power of social media such videos can be spread worldwide almost instantly.

In May 2019, a deepfake video of Donald trump surfaced in which he offered advice to Belgium people on climate change and this sparked an outrage in Belgium and people viewed this as an intervention of the United States of America in Belgium's climate policy [78]. President Donald J. Trump also shared a video of his political rival, Nancy Pelosi in which she appeared to be intoxicated was widely shared on Facebook and YouTube [74].

In recent elections in India, Bharatiya Janta party (BJP) also used deepfakes in its election campaign [75]. In one such fake video which emerged one day before elections in India, Mr. Tiwari is urging citizens to vote for Bharatiya Janta party, and this video was sent to about 15 million voters via 5800 WhatsApp groups [76]. This just shows how a well-crafted video at the right time can spread like a wildfire.

Deepfakes are expected to be the next big problem in the future elections, and according to Paul Barrett, professor at NYU, “a skillfully made deepfake video could persuade voters that a particular candidate said or did something she didn't say or do”

BUSINESSES:

Businesses are constantly under a threat of hackers or corporate espionage. Deepfakes in combination with other social engineering techniques can be used by malicious actors to gain access to sensitive business information or can be used to create miscommunication within the company. Other than this such videos can also be used to

spread fake news regarding a company or its executive to harm the share price of company or its credibility in market.

In June 2019, a deepfake of CEO of Facebook, Mark Zuckerberg, went viral in which he talked about controlling the data and paid tribute to a fictional society [79]. In another instance, scammers defrauded a UK based firm for \$220,000 by impersonating the CEO of the company [77].

OTHER MALICIOUS USAGES:

Videos are considered to be a very reliable source of evidence, however, deepfake technology changes that as well. Currently, our judicial systems are not well equipped to handle such fake videos

Deepfakes have also been used in creation of fake adult movies or pornographic content, for example Scarlett Johansson's face was grafted onto several graphic scenes by anonymous creators [80].

As it is evident from the above stated examples that the threat posed by deepfakes is real and, it is only going to get worse in future. Therefore, deepfake identification is a very important topic.

DEEPFAKE IDENTIFICATION AND PROBLEMS

Deepfake identification problem can be considered as binary as any image or video is either fake or not. Most of the current research in the domain of deepfake detection has relied heavily on identification of several details or characteristics of deepfakes, however, these signs have rather been short-lived. For example, research at the University of Albany pointed out that deepfakes could be detected using eye blinking as the deepfake clones did not blink frequently enough like a normal human [1].

However, shortly afterwards this finding was made public, the next generation of deepfakes started to incorporate blinking thus making this technique of identification

obsolete. Most of the current research papers, such as [2], point out that GAN generated deepfakes are the hardest to detect.

Mike Schroepfer, Facebook's chief technology officer, has termed catching deepfakes using Artificial Intelligence as a cat and mouse game [82]. The argument by Facebook's CTO is true and the primary reason that GAN based deepfakes are getting harder to detect is that detector algorithms can be trained to detect the deepfakes based on the latest research and as a result the generator algorithm learns to evade detection thus rendering the latest advancement in detection almost obsolete.

MOTIVATION

Video and image are considered to be the most trustworthy and irrefutable form of evidence, as any alteration or change in it could be easily identified previously with just an eye. However, with the latest technology, leveraging the power of deep learning, hyper-realistic digital synthetic images and videos can seamlessly be created such that the product of these algorithms is almost indistinguishable to the naked eye. Deepfake is a technique which aims to replace the face of a targeted person by the face of someone else in a video.

With the progress in Generative Adversarial Networks (GANs) [1] and Variational Autoencoders (VAEs) [2], an improvement in the quality of deepfake generation methods [3, 4, 5] has also been observed. In response to these continuously evolving deepfake generation methods, several deepfake detection approaches [6, 7, 8] have been proposed to detect such forgery content.

In addition, several datasets have been made publicly available, such as Celeb-DF [9], FaceForensics++ [10] and UADFV [11]. Recently, Facebook also released a dataset and launched the DeepFake Detection Challenge (DFDC) to improve DeepFake detection performance. All these datasets contain both the real and manipulated videos, with the real videos mostly collected from online video streaming sites like YouTube. A thorough investigation of the current reported results on these datasets reveals that present deepfake detection approaches can extract useful manipulation features, but their

performance is unsatisfactory while they are unreasonably large networks which make them inefficient to be deployed at scale.

The current proposed methods are also utilizing dataset-dependent heuristics which need to be incorporated in the detection process to obtain substantial results. This results in poor generalizability of a detection method across different datasets.

We base our work on this weakness of current deepfake detection approaches and propose two novel architectures which are lean while providing similar results to that of state-of-the-art solutions.

Our proposed approaches are computationally inexpensive due to lower number of parameters thus allowing them to be deployed at scale.

SUMMARY

This chapter provides us with motivation behind our approach. It also helps us in understanding the problem that we ought to solve. The chapter provides a detail background of the current malicious usages of this technology and explains how it can be used by bad actors in various domains of our life. Finally, the chapter closes with the motivation behind our solution and the unique aspect of our contribution in this area.

BACKGROUND ON DEEPPFAKE GENERATION

Deepfakes emerged relatively recently on reddit and since then it has become a hot topic within the artificial intelligence research society. Considerable work is being done in this field as deepfake detection is considered a very hot topic these days due to the impact deepfakes can have on society.

To understand the ongoing research in deepfake detection, it is pertinent that we understand how these fake videos are generated in the first place using neural networks.

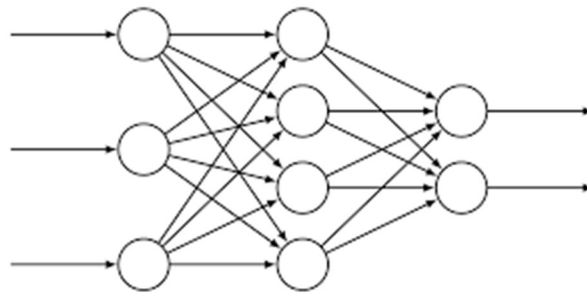


Figure 1: Neural Network

VIDEO MANIPULATION AND DEEPFAKE GENERATION

Facial manipulation techniques are not new, they have been around since 1990s. Bregler et al. [52] presented a technique to generate mouth movements so a person can be seen saying the words that he/she did not say. Dale et al [53] presented an automatic Face Replacement technique. They tracked the 3D facial movement in both source and target videos and exploit the corresponding 3D geometry to warp the source face to the target face. Garrido et al. [54] presented a system that replaces the face of an actor with a target while preserving the original expressions. This technique was able to work with limited data. Thies et al. [55] demonstrated the first real-time expression transfer for facial reenactment.

Video manipulation is the act of altering digital videos using various video editing and processing techniques. These manipulations can be categorized into three main sets:

- Fully synthesized videos
- Altering in the semantics or style of videos
- Altering the video content via cut-paste clips

In the context of this report, we will be purely discussing techniques related to deepfake generation.

The first deepfakes emerged on reddit back in 2017. Fakeapp, was developed by the reddit user using an autoencoder-decoder combination [34]. Since then, various other open source deepfake generators have been published. These include the likes of Faceswap [35], Faceswap-GAN[36], Dfaker [37], DeepfaceLab [38], and Faceshifter [39].

HOW ARE DEEPPAKES MADE?

Initially, deepfakes were made using autoencoder and decoder pair; however, recently by adding adversarial loss and perceptual loss to encoder-decoder architecture new and improved quality of deepfakes have emerged based on GANS or generative adversarial networks.

We discuss both of these networks in detail below.

Auto Encoders

Auto encoders is a type of neural network that can efficiently learn the data features in an unsupervised manner [40]. According to Vincet et al [42] autoencoders are useful for learning useful properties which can be used for classification problems.

An auto encoder consists of two parts: encoder and decoder. Basically, the main functionality these two units perform is that they learn to copy inputs to outputs. Encoder learns to map the input to a vector output while the decoder tries to use the vector produced by encoder and translate it back into the original input.

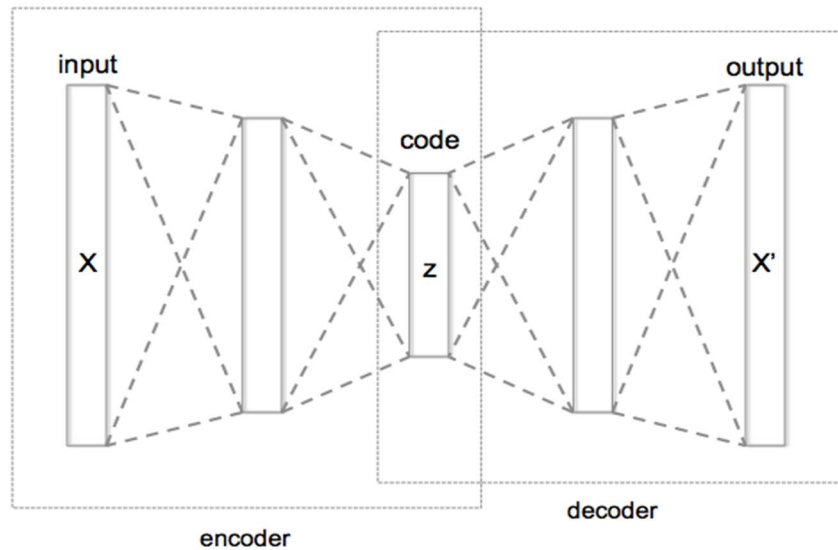


Figure 2: Autoencoder [43]

The compressed representation, or z as pointed in the figure above, learned by the auto-encoder is the basic element or step that allows face swapping capability [29]. For deep fake generation two sets of encoder-decoders are used. Two networks are designed in a way that they share the weights among encoders, however, they have separate decoders.

Training

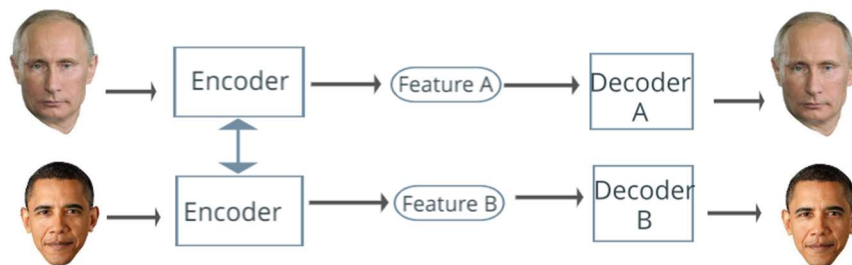


Figure 3: Training

During the training phase as shown in fig 10 the two decoders are forced to train together on two sets of images. The first set of images contains the target face that will be replaced

whereas the second set of images contain the face of the person that will be swapped into the images.

The two auto-encoders are forced to train together because if each autoencoder is trained separately on a set of images then each decoder will only be able to decode a certain latent space. By forcing the two autoencoders to share the weights of the encoder during the training phase, the encoder is forced to identify the common features between the two faces.

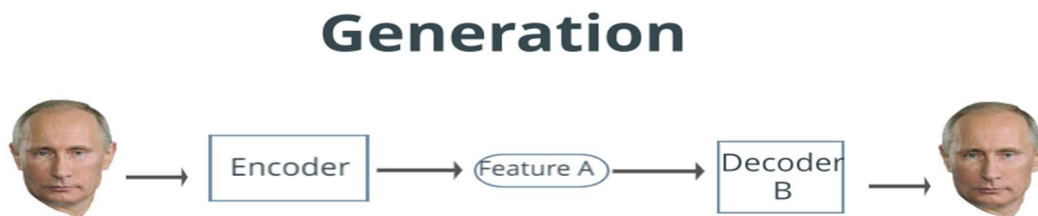


Figure 4: Deepfake generation

To swap faces in a video, firstly a face detector algorithm is used to identify the face to be swapped in video. The facial region is passed to the encoder to identify the latent features of the original subject present in the video. These features are then passed into the decoder trained on the images of the person we want to swap into the video [29]. The decoder in this way tries to reconstruct the new face into the video by utilizing the common features between the original face and the other face.

By using such an operation on each frame in a video a deepfake is generated using face swap operation. However, such deepfakes have several limitations and due to the way encoder tries to reconstruct the face several inconsistencies arise in the generated videos. Due to the inconsistencies in the images and videos the network is trained on, for example there could be inconsistency in the illumination, shadows, camera angle, face

view, or differences in the quality of videos, as a result the generated deepfakes are not of such high quality.

Variational autoencoders have also been used in deepfake generation. Variational autoencoders were introduced in 2014 by Diedrik [44]. A major difference between the two is that simple autoencoders are deterministic whereas variational autoencoders or VAEs are stochastic in nature. Variational autoencoders can produce an image from the random noise they receive as input.

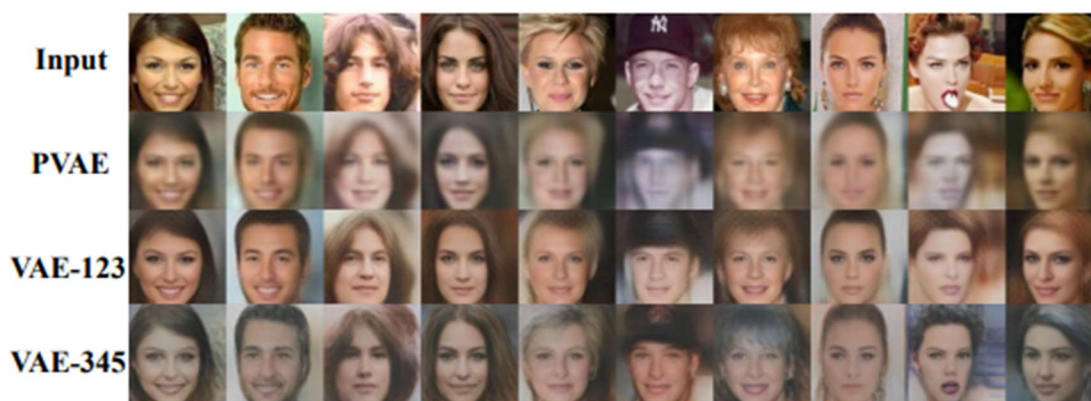


Figure 5: X.hou [45] shows the image reconstruction using VAE

In simple auto encoder, input x is encoded into latent representation of $e(x)$ which is later decoded back into the input or $d(z)$ by decoder. However, in variational autoencoder the input is initially encoded to latent distribution $p(z|x)$ and then sampled into latent representation $z \sim p(z|x)$ which is decoded to input or $d(z)$.

Generative Networks

As pointed out previously, the deepfakes generated using only autoencoder pairs are not of such high quality due to several limitations. To improve the quality of deepfakes and make them as realistic to original videos as possible, recently the power of generative adversarial networks to produce highly realistic deepfakes has been leveraged.

Generative adversarial networks were introduced by Ian good fellow in 2014 [46]. Ian proposed a framework for evaluating generative models using an adversarial network [46]. The two neural networks are meant to compete against each other in a zero-sum game.

In GANs there are two neural networks that compete with each other:

- Generator
- Discriminator

Generator networks are designed to fool discriminators whereas discriminator is tasked to identify whether the input belongs to the distribution or not.

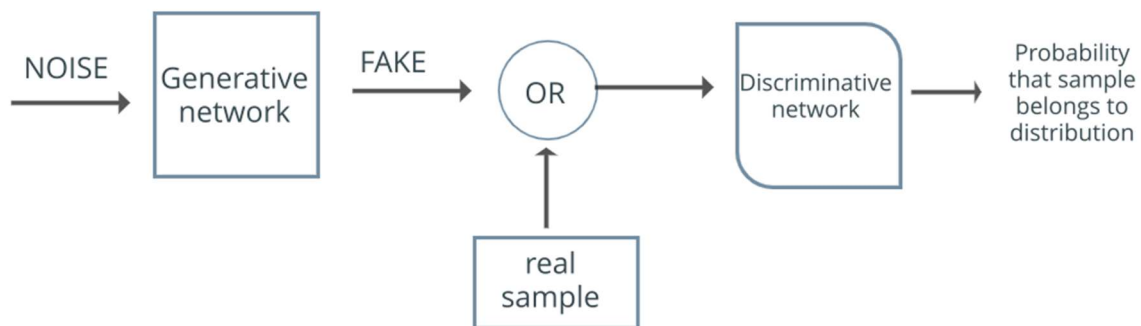


Figure 6: Generative adversarial network

Generative networks receive random noise, and their goal is to transform this noise to a specific form. The discriminative network receives input either from the generative model or the intended distribution, and it outputs the probability that the given input belongs to the distribution or not.

After GANs were introduced by IAN et al [46] many important improvements have taken place in this area. CIGAN were introduced by Kurutach et al [47] that were capable of generating videos.

Karras et al [48] presented a major breakthrough in the size of image generative networks could generate. Figure 7 shows the images taken from [48] which were generated using proGAN provided by Karras in [48].



Figure 7: Images provided by karras [48]

DEEPPFAKE DATASETS

In the past years, many deepfakes datasets have been published. Each dataset relies on some specific techniques for deepfake generation and therefore has certain limitations.

We will only be discussing few of the recent datasets:

- Celeb DF
- FaceForensics++ (Raw, c23, c40)
- Deeper Forensics
- Wild Deepfakes

CELEB DF:

Celeb-DF consists of 5,639 deepfake videos which makes about 2 million frames. Real videos have been sourced from publicly available videos of 59 celebrities. The sample of celebrities consists of diverse genders, ethnic groups, and ages.

Deepfake videos for the datasets have been generated using a deepfake synthesis algorithm. By using the encoder decoder model with an increased number of layers the quality of synthesized faces has been improved to 256X256 pixels. By incorporating Kalman smoothing algorithm, temporal flickering has been reduced. Color mismatch between donor and target face has also been reduced significantly.

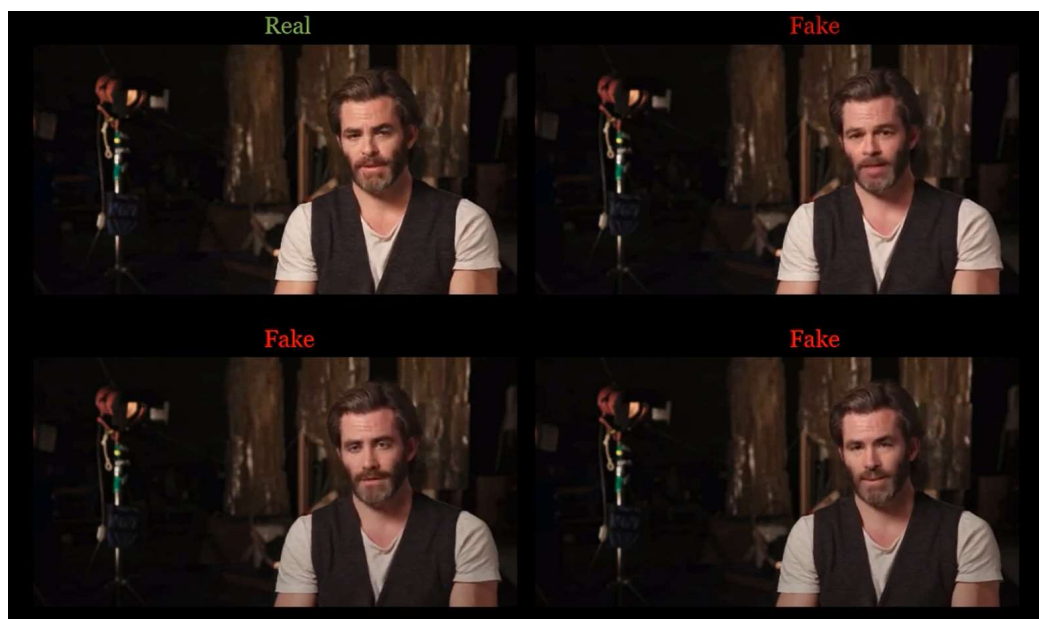


Figure 8: Images from celebdf

FACEFORENSIC++

Face Forensic++ is extension of faceforensic. It is one of the largest such datasets publicly available. Pristine or real videos in this dataset have been taken from the Internet. These videos imitate real world scenarios, and these videos make about 1000 in number or 509,914 images [51].

To generate fakes, two graphic based approaches and two learning-based approaches have been used.

- Graphic based approach
 - Face2Face
 - FaceSwap
- Learning based approach
 - DeepFakes
 - NeuralTexture

To read more about how the exact technique works refer to [51].

The videos in dataset in three different qualities: raw, HQ, and LQ.

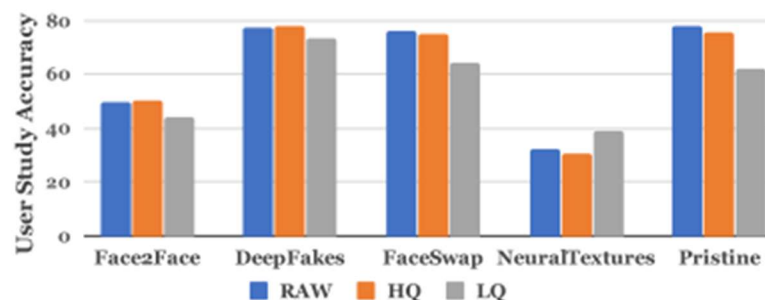


Figure 9: User accuracy as presented in [51]

Fig 9 summarizes human performance on these videos, and as it can be seen that human ability to discern these videos decreases as the quality of videos decreases.

DEEPER FORENSIC

Deeper forensic is the largest available dataset for deep fakes. It consists of 60,000 videos which altogether consist of 17.6 million frames. The dataset consists of 50,000 original videos while 10,000 are manipulated videos.

The source videos use 100 paid actors, 55 males and 45 females, from 26 different countries. Deeperforensic utilizes Deepfake Variational autoencoder, which consists of three parts:

- Structure Extraction Module
- Disentangled Module
- Fusion Module

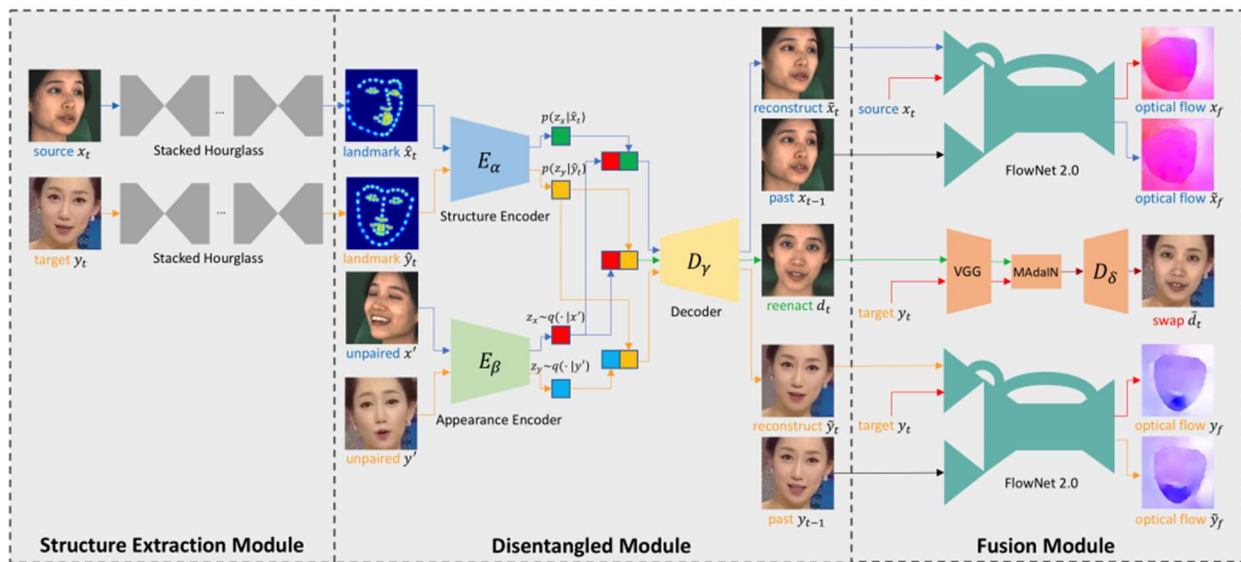


Figure 10: Deepfake variational autoencoder presented in [49]

Fig 10 explains the process used by deeperforensic for fake videos generation. Seven different types of distortions are also applied to videos to make them seem as close to real world videos as possible. These distortions are applied at random, and each variation has 5 different intensity levels which are applied at random.

Seven types of distortions are [49]:

1. Color saturation change
2. Local block-wise distortion
3. Color contrast change
4. Gaussian blur
5. White Gaussian noise in color components
6. JPEG compression
7. Video compression rate

WILD DEEPAKE:

WildDeepfake is a recently published dataset by Bojia Zi et al [50]. This dataset addresses a very important aspect which previous datasets were lacking. While previous datasets relied on self-produced deepfakes, which were based on popular deepfake generating software. As a result, datasets did not accurately present real world deepfakes.

WildDeepfake consists of videos entirely collected from the web and therefore is a much more accurate representation of distribution which a model deployed to identify videos on the web will face.

This dataset contains 707 well-made deepfake videos collected from the internet [50]. The dataset consists of 3,805 real face sequences and 3,509 fake face sequences.

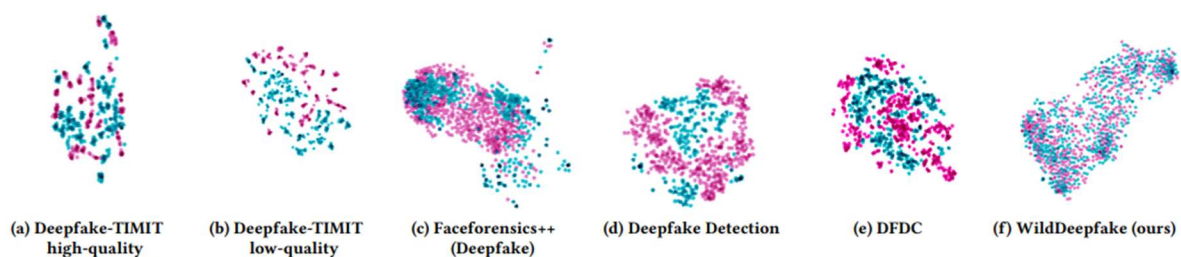


Figure 11: Feature comparison of different dataset as presented in [50]

PREVIOUS WORK IN DEEPPFAKE DETECTION

CLASSICAL COMPUTER VISION APPROACHES FOR VIDEO ALTERATION:

There are various techniques in the field of multimedia forensic for detection of forgeries and alterations in images and videos.

For video forensic, techniques include detection of duplicated frames or dropped frames [21], resampling [27] [19][23], CFA (Color filter array) interpolation [22], or copy-move manipulations [24]. For face-based manipulation detection numerous techniques have been proposed such as using differences in facial colors that emerge due to pulse [25].

Other than these, Ricard et al. [56] applied Discrete Fourier Transform and Azimuthal Average to create 1D representation of the FFT power spectrum and fed it to Logistic Regression (LR), SVM and K-Means Clustering to discriminate between altered and pristine images. Xin et al. [57] estimated the 3D head poses from images and trained an SVM classifier on them to distinguish real videos from fake images.

Classical computer vision techniques have played an important role in detection of any forgeries or alterations in images or videos. However, with recent advancement in technology and especially since malicious actors have started using deep learning-based approaches such as GANs for alteration in images and videos, these classical computer vision approaches have become obsolete as they have been unable to identify alterations made by these deep learning models [20]. GANs have the ability to preserve pose, facial expressions, and lightning of images [58]. Therefore, the detection of deepfakes using classical approaches has become very difficult [59].

DEEP LEARNING-BASED APPROACHES

Deep learning-based approaches have proven to be considerably better to detect alterations in videos being produced by deep learning algorithms as classical techniques are dependent on hand-crafted features to detect any alteration whereas deep learning models have been quick to learn to avoid these pitfalls. Deepfake algorithms produce content of limited resolution, as a result they need to be further warped to match original face. As a result, distinct artifacts can be found around the warped faces which can be exploited in detection.

Afchar et al [31] proposed a deep learning-based approach by proposing two networks Meso-4 and MesolInception-4 that utilized the mesoscopic property of images to classify fake videos. As naked eye finds it difficult to identify forged images especially if they contain human faces [32], and the noise in images at microscopic level is not helpful therefore by using a network with small number of layers authors in [31] were able to achieve high detection rate of 98% for their own Deepfake dataset and 95% for faceforensic [18] dataset generated using Face2Face approach.

Similarly, Yuzezun et al [30] used convolutional Neural Network (CNN) to detect affine space warping artifacts introduced by deepfake generation pipeline. Yuzezun et al [30] proposed a CNN based detection mechanism to detect artifacts around facial region. Four different CNN were trained VGG16 [60], ResNet50, ResNet101, and ResNet152 [61].

Nguyen [3] proposed the use of capsule networks for detecting manipulated images and videos. A dynamic routing algorithm is deployed to route the outputs of the three capsules to the output capsules through several iterations to separate between fake and real images.

Rossler et al [62] did a detailed analysis of deepfake detection methods, and found that Xception net using ImageNet pretrained weights performed best when retrained for deepfake classification.

Most of these approaches rely on within frame artifacts only as they make use of frame wise aggregation to decide whether input video is deepfake or not. Thus, these methods

do not take into account intra-frame inconsistencies produced by deepfake generation pipeline. Intra-frame artifacts can be exploited by using the power of recurrent neural networks.

D. Güera et al [29] was first to propose that due to the way deepfakes are generated, intra-frame or temporal inconsistencies emerged between frames which could be used in detection by extracting frame level features from sequential frames using CNNs and passing these features into LSTMs to make prediction. By combining RNN network such as LSTM with CNN, the resulting networks are considered to be ‘deep in space’ and ‘deep in time’ and thus are well suited to exploit these intra frame inconsistencies.

E. Sabir [35] also used a combination of CNN and RNN for detection by using a landmark based face alignment with bidirectional recurrent densenet and were able to achieve accuracy of 96.9 on Deepfake dataset [16] and 96.3 on Faceswap[17].

Yuezun [34] proposed another method by exploiting the fact that deepfake faces at that time did not have the capability to generate fake faces that can blink normally. Long term recurrent convolutional network (LRCN) consists of a feature extractor based on CNN and a sequence learning based on long short-term memory (LSTM) [15], was used to predict probability of eye open and close state. The result showed that blinking rate in original videos and fake videos was 34.1 blink/min and 3.4 blinks/min, respectively.

Masi et al [36] proposed a two-branch recurrent network structure that detects fake videos by learning to amplify the artifacts while suppressing the high-level face content. One branch is subjected to a deep Laplacian of Gaussian and the second works in color domain, the features from both branches are fed into dense block and later a bi-directional LSTM. They were able to achieve an AUC score of 99.12 on FF++ (c23) and 91.10 on (c40) using this architecture. Javier [38] building upon the work of QI et al [37] used Convolutional attention network to extract spatial and temporal features was able to achieve a high AUC score of above 98% on Celeb-df[14] and DFDC [13]. Zhao [39] recently has proposed a patch-wise consistency learning (PCL), and have achieved AUC score of 99.79% of FF++ and 99.57% on DFR.

METHODOLOGY

Although the Deepfake detection technology has progressed significantly in the past few years there are some important gaps that have not been addressed. In this report we address some of the gaps we identified during our literature review. This section presents different models and experiments we ran to come up with our final model. In this section we cover:

- Preparation of datasets
- Overview of previous experiments
- Presents the analysis that microscopic spatiotemporal features are enough to detect most Deepfake videos
- Introduces ConvRNN layer that efficiently extracts spatial-temporal features
- Propose two very light weight Deepfake detectors that perform just as well as current state-of-the-art deep Learning models in the task of Deepfake detection

Our approach to tackle the problem of discerning deepfake videos can be divided into two steps:

- Data Preprocessing
- Classification through Neural Network

DATA PREPROCESSING

To prepare raw deepfake and their corresponding unfabricated real videos to feed to a neural network, we need to process them. Since we are tackling datasets in video format, one of the hurdles with them is that they occupy large sizes (typically in order of Gigabytes and Terabytes). Therefore, it became quite a challenge to store them. Similarly, large datasets would correspondingly require more training time, slowing down the training-loop iteration of a machine learning pipeline.

As discussed earlier, deepfake generation processes only involve swapping of faces to generate synthetic media. Only the facial region is replaced in the target video with someone else's likeness. Since the remaining part of the video frames (background) remains unchanged, it can be discarded in the video pre-processing step. Retaining only the facial region from a deepfake video will not only discard unnecessary information from

the frames but would also cause decrease in the size of the preprocessed video compared to the large original video. Thus, preprocessing of the deepfake videos allows decreasing the large dataset size.

In order to retain only facial regions in a deepfake video and discarding the rest of the scene, we need to detect faces in the videos. In this regard, there can be two cases deepening on the datasets:

- Facial masks included in the dataset (e.g., Faceforensic++)
- Facial masks NOT included in the dataset (e.g., CelebDF)

FACIAL MASKS

If the masks are already available with the dataset, this reduces the computations required and the facial regions can easily be extracted using these included masks. If not, an extra detection step is required to first detect facial regions in a video, processed frame-by-frame, and retain only that part of the frame, writing it back as a processed output video.



Figure 12: A sample frame from face forensic data set

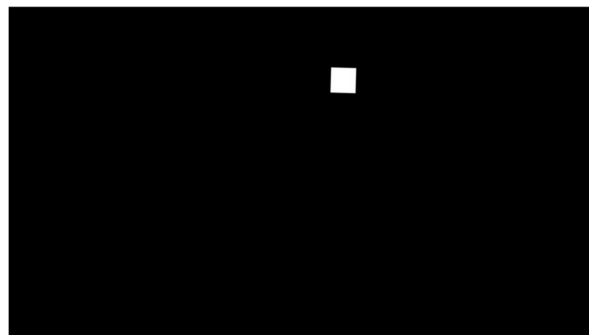


Figure 13: Mask for image in fig 12

USING FACE DETECTION

The videos are pre-processed to extract the facial region. Two different face detectors were used, namely:

1. Dlib [53]
2. MTCNN [54]

The Dlib face detector is fast but the detection accuracy is low, on the other hand the MTCNN face detector is slow but has a higher accuracy, so to speed up the pre-processing steps and get high detection rates Dlib face detector and MTCNN were arranged in a hierarchy - frames missed by Dlib were processed using MTCNN.

The images were then mean normalized, rescaled from 0-1 and sequences of consecutive frames are then formed in order to be fed to the neural network for detection before feeding them to the network.

In order to incorporate certain inconsistencies added in the frame, around the facial region, instead of retaining tight facial regions, padding was also applied on the output facial coordinates before cropping the frame. Thus, irregularities around the swapped face are not discarded and are also included in the processed video.



Figure 14: After preprocessing

CLASSIFICATION THROUGH NEURAL NETWORK

Following are the different architectures we experimented with. These series of approaches evolved, building upon each other, resulting in the final architecture which has also been discussed below.

The main goal behind our research was to explore the intra-frame inconsistencies that emerge due to the deepfake generation pipeline. Through detailed literature review on the generation of deepfakes as well as the current deepfake detection methods, we pinpointed a few anomalies in the deepfake generation process.

Deepfakes generated using encoder-decoder architecture have shared encoder weights. Ideally, the original face and target face (to be swapped with in the original video) should have similar conditions such as viewing angles and illumination etc. But since both original and target videos are recorded in different conditions, there is a mismatch in their viewing and illumination conditions. These differences leave their trails in the output video when the face is swapped with the target face. Thus, resulting in several inconsistencies of the swapped face with the rest of the scene. Not only the change in the viewing conditions, but even a simple change as a change in different codecs used for the video also result in several anomalies in the resultant deepfake video.

In addition to these spatial differences, several conditions also add temporal discrepancies in the deepfake videos. Since, only the detected face region is passed through the encoder-decoder architecture, it only swaps the facial region, remaining unaware of the rest of the scene (background). Similarly, the video is processed frame-by-frame, so the encoder-decoder architecture is also unaware of the previously generated frames with swapped faces and there is no link or connection in between them. This lack of temporal awareness in the deepfake generation process is also a source of multiple anomalies in the resultant video.

Although the above-mentioned anomalies are discrete to a naked eye, they can be captured by advance neural network architectures and can be used to accurately discern a deepfake video, whether a video has been fabricated or not.

We tried various different techniques to exploit these inconsistencies namely:

- Convolutional-LSTM
- Y-shaped Architecture
- Vision Transformer based Architecture
- Inception-ConvRNN Architecture

CONVOLUTIONAL-LSTM (BASELINE)

Our first baseline approach involved utilizing these anomalies and using them to classify an input video to either be real (pristine) or fake (deepfake). The architecture mainly consisted of two blocks:

- Convolutional block
- Sequence model block (LSTM)

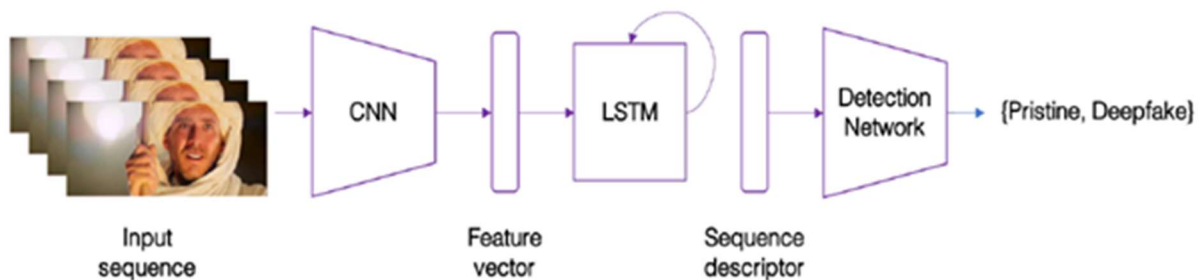


Figure 15: Convolutional-LSTM

In order to capture different frame-level inconsistencies (such as due to different viewing angles or illumination conditions), a pixel-level convolution block was added. Since convolution operation involves computations between frame pixels, identifying information such as edges (at lower level) or even object components (at high level), they can be used to catch the spatial inconsistencies.

For temporal inconsistencies in between consecutive frames, a sequence model block was added in the architecture. The sequences of frames, after being passed through the convolutional block, were then passed through Long Short-term Memory (LSTM) cells.

Finally, the feature vector extracted was then input to a detection network (sequence of fully connected dense layers) to output the probability of the input video being a deepfake.

Following results have been obtained using this approach:

Table 1: Convolutional-LSTM results

Convolutional-LSTM	Results on Celeb-DF Dataset	
	Accuracy	AUC
	73.67	80.14

Y-SHAPED ARCHITECTURE

On further experiments on our baseline model, it was observed that the model discarded several useful features.

Since the sequence of frames are first passed through a convolutional block before being fed to a sequence model block, with each convolution operation, several frame features are discarded, and they do not reach the sequence model. These discarded features might not be useful spatially, but they might be useful for the detection of deepfake video temporally.

Based on the following intuition, in our updated approach, we directly fed the sequence of consecutive input frames, both to the convolutional as well as the sequence model block. The output features from both these blocks were then concatenated before being fed to the detection network as a single feature vector.

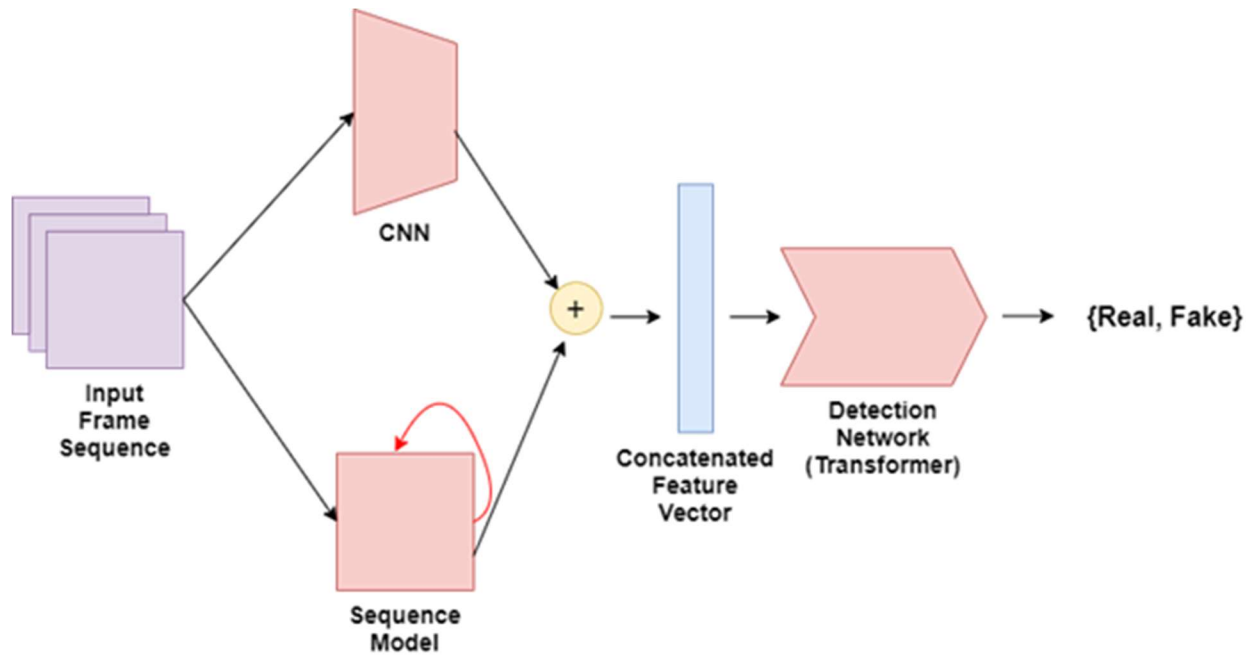


Figure 16: Y-shaped architecture

Table 2 summarizes the result we obtained using this architecture on celeb-df data set.

Table 2: Y-shaped architecture results

Y-shaped Architecture	Results on Celeb-DF Dataset	
	Accuracy	AUC
	45.81	50.92

VISION TRANSFORMER BASED ARCHITECTURE

In order to further improve our existing Convolutional-LSTM and Y-shaped architecture, we experimented with adding attention to our architecture.

Inspired by the Vision Transformer (ViT) by Alexey Dosovitskiy et al [65] from the paper “An Image is Worth 16x16 Words” we incorporated the Vision Transformer in our architecture as an attention module.

In this approach, the sequence of consecutive input frames was first passed through Conv-LSTM layers to extract features. These feature vectors were then fed to a Vision Transformer (ViT) and a probability was output whether the input sequence of frames is from a deepfake video or not.

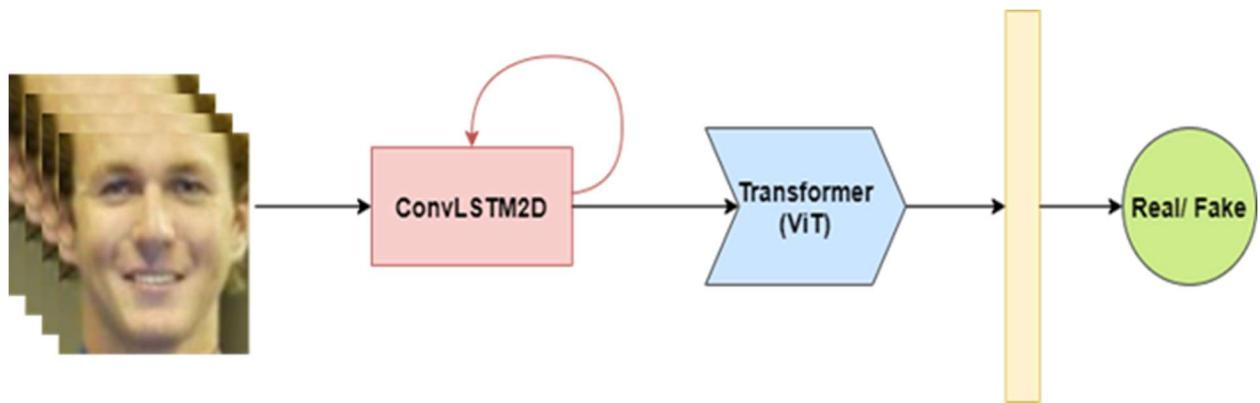


Figure 17: Vision transformer-based architecture

Table 4 summarizes the results below. However, during our experimentation with this architecture we also tried incorporating attention module in CNN layer, however, the results on celeb-df dataset did not show any improvement in any metric my incorporating attention in CNN.

Table 3: Vision transformer-based architecture result

Vision Transformer based Architecture	Results on Celeb-DF Dataset	
	Accuracy	AUC
	77.18	84.71

INCEPTION-CONVRNN ARCHITECTURE

This is the final model in our report. Below we present a detailed explanation for the intuition behind this architecture and how this architecture works.

Fchar et al [20] proposed light weight Meso4 and Mesoinception-4 networks to detect Deepfakes however, their performances degrade when tested against the generation-II deepfake datasets [66]. Guera et al [24] and Sabir et al [25] showed that using the temporal information from successive frames can increase the detection accuracy; however, most of the temporal information is lost during successive convolution and pooling operations of CNN. Sharoz et al [67] showed the use of ConvLSTM module to efficiently extract the spatial-temporal features from the videos, however CLRNet is a giant model with millions of parameters.

ConvRNN

ConvRNN is a type of recurrent neural network for spatial-temporal prediction that has convolutional operation in both the input-to-state and state-to-state transitions. The ConvRNN determines the future state of a certain cell in the grid by the current inputs and the past states of its local neighbors. This can easily be achieved by using a convolution operator in the state-to-state and input-to-state transitions, See fig 18.

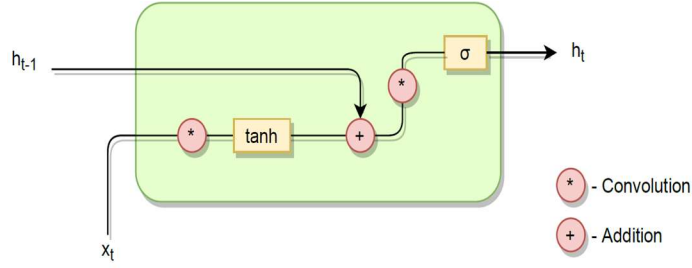


Figure 18: Proposed ConvRNN cell

The key equations of ConvRNN are shown below, where “*” denotes the convolution operator and ‘o’, denotes the Hadamard product:

$$i_t = \tanh (W_{xi} * X_t + b_i)$$

$$f_t = (i_t + \mathcal{H}_{t-1})$$

$$\mathcal{H}_t = \sigma (W_{fh} * f_t + b_h)$$

ConvRNN layer uses just two convolution operations and has 4x less parameters than ConvLSTM while performing better than ConvLSTM on the task of Deepfake detection.

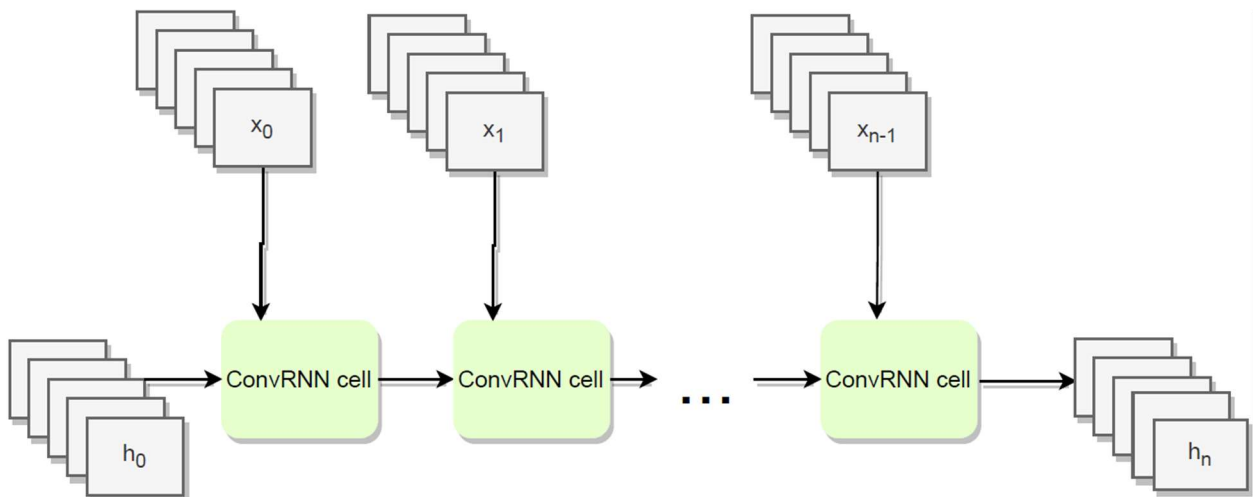


Figure 19: Unrolled ConvRNN layer

InceptionConvRNN Model

The proposed model consists of three parts:

1. An encoding module consisting of Inception sub-modules and convolution layers
2. A sequential module consisting of ConvRNN layer(s)
3. A classifier module consisting of Fully Connected layers.

Figure 20 below represents the general architecture of the proposed model visually.

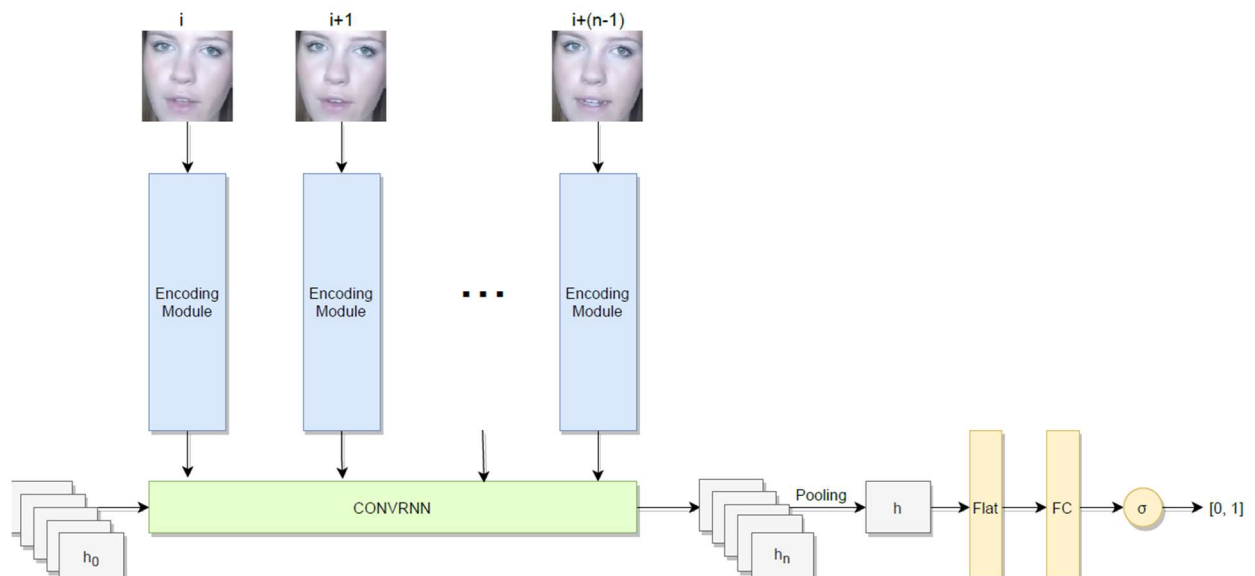


Figure 20: Proposed Inception-ConvRNN Architecture

We propose two different variants of the model in this report:

1. **Meso-ConvRNN**
2. **Micro-ConvRNN.**

Inception Module

For the Inception sub-modules, the variant presented by in [20] is used.

Inception module stacks the output of several convolutional layers with different kernel shapes increasing the function space in which the model is optimized. The original Inception module contains 5x5 convolutions [67], but the proposed variant has 3x3 dilated convolutions [68]. 1x1 convolutions were added before dilated convolutions for dimension reduction and an extra 1x1 convolution in parallel acts as skip-connection between successive modules.

Meso-ConvRNN

This variant consists of two Inception modules with batch normalization and pooling, two Convolution layers with Batch Normalization followed by a ConvRNN layer with 16 filters and a Fully Connected layer with 16 units.

The chosen parameters (a_i, b_i, c_i, d_i) for the Inception module 'i' can be found in Table 4. The total number of parameters for this model is 17K. Figure 21 represents the encoding module for this variant.

Table 4: Number of filters for each convolution in Inception modules used in meso-ConvRNN

Layer	a	b	c	d
1	1	2	2	1
2	2	4	4	2

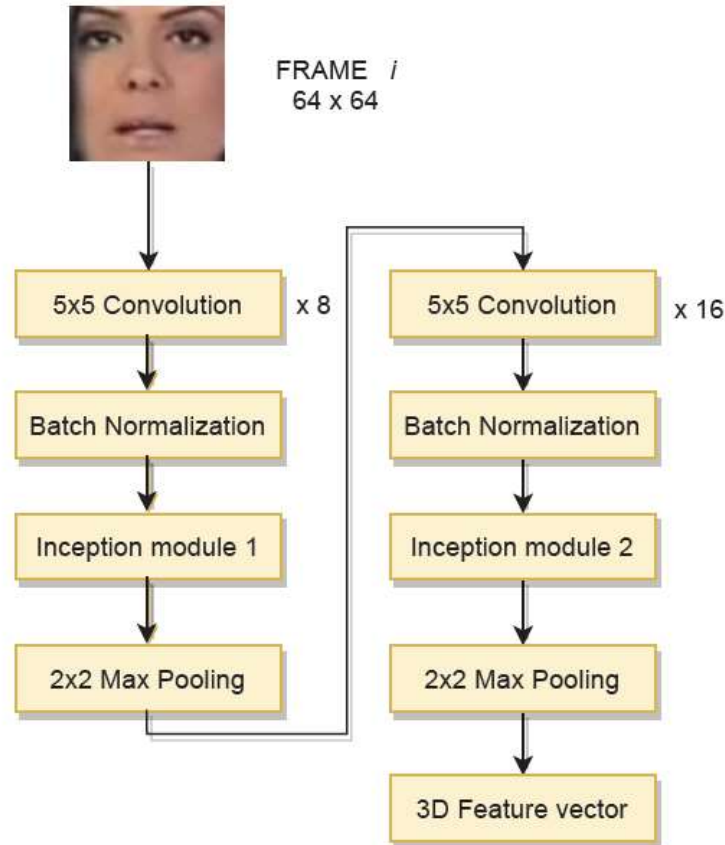


Figure 21: Meso Inception encoding module

Micro-ConvRNN

While the microscopic spatial features of the image are degraded by noise [20]. Deepfake videos have a strong microscopic spatial-temporal fingerprint that remains intact and can be used to easily detect Deepfakes.

The micro-ConvRNN is a shallow version of previous model consisting of one Inception module with batch normalization and pooling, one Convolution layer with Batch Normalization followed by a ConvRNN layer with 16 filters and a Fully Connected layer with 16 units.

The chosen parameters (a, b, c, d) for the Inception module can be found in Table 2. The total number of parameters for this model is 9K. Figure 22 represents the micro inception coding module.

Table 5: Number of filters for each convolution in Inception module used in micro-ConvRNN

Layer	a	b	c	d
1	2	4	4	2

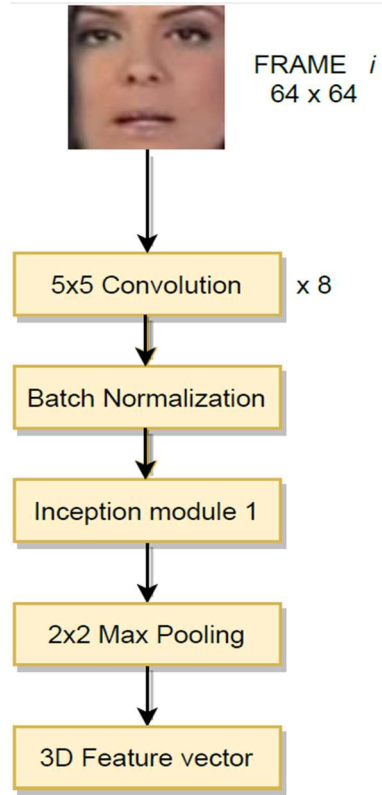


Figure 22: Micro Inception encoding module

Results

Extensive experiments were performed to judge the performance of the proposed methods on the Deepfake detection tasks. Three datasets were selected:

1. FaceForensic++ (FF++)
2. Celeb-df
3. DeeperForensic 1.0 (DFR v1)

Table 6: Inception-ConvRNN results

Dataset	Method	AUC
<i>FF++ (RAW)</i>	Micro	96.13
	Meso	99.29
<i>FF++ HQ (c23)</i>	Micro	95.75
	Meso	96.02
<i>FF++ LQ (c40)</i>	Micro	91.23
	Meso	93.08
<i>Celeb-df</i>	Micro	89.1
	Meso	93.51
<i>DFR v1</i>	Micro	96.72
	Meso	97.71

Table 6 summarizes the performance of both micro and meso architectures on different datasets. Proposed models show good performance on both generation I (FF++) and generation II (Celeb-DF, DFR) datasets.

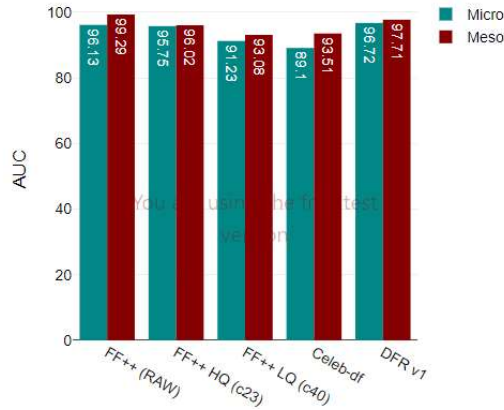


Figure 23: Comparison of AUC for meso and micro module

Meso architecture shows better performance than Micro architecture on all datasets as shown in figure 23. This was expected as Meso architecture has a greater number of parameters than Micro architecture. Different compression on FF++ videos cause loss of information and also affects the overall performance of both networks. The experiments

show that shallow light-weight models with limited parameters can detect Deepfakes efficiently.

Table 7: Comparison of detection performance of different methods using AUC score

	Method	Celeb-df	FF++/DF	Number of parameters
Zhou et al. (2017) [69]	InceptionV3	53.8	70.1	24M
Afchar et al. (2018) [20]	Meso4	54.8	84.7	27.9K
Afchar et al. (2018) [20]	MesoInception-4	53.6	83	28.6K
Li et al. (2018) [22]	FWA(ResNet-50)	56.9	80.1	23.8M
Yang et al. (2019) [57]	HeadPose (SVM)	54.6	47.3	-
Matern et al. (2019) [70]	VA-MLP	53.8	66.4	-
Rossler et al. (2019) [10]	Xception-raw	48.2	<u>99.7</u>	22.8M
Nguyen et al. (2019) [70]	Multi-task	54.3	76.3	-
Nguyen et al. (2019) [23]	CapsuleNet	57.5	96.6	3.9M
Sabir et al. (2019) [25]	DenseNet+RNN	79.1	99.6	25.6M
Li et al. (2020) [22]	DSP-FWA (SPPNet)	64.6	93	-
Tolosana et al.. (2020) [72]	Xception	83.6	99.4	22.8M
Hong et al. (2021) [73]	DeepfakeHop	90.56	97.45	42.8K
OURS	Micro-ConvRNN	89.1	95.75	<u>9.2K</u>
	Meso-ConvRNN	<u>95.9</u>	96.02	17K

The comparative analysis with the techniques presented in the literature is also done to show the effectiveness of the proposed approach in the table 7. The results show that all deep learning methods struggle on the Celeb-DF dataset. DeepfakeHop [73] is a non-DL method that shows good performance on Celeb-DF. The proposed Meso-ConvRNN outperforms all the other methods (DL and no-DL) on Celeb-DF and also achieves near state-of-the-art performance on FF++. Micro-ConvRNN does not lag behind much, despite having the lowest number of parameters it gives third best performance on Celeb-DF and rivals million parameter models on FF++ dataset.

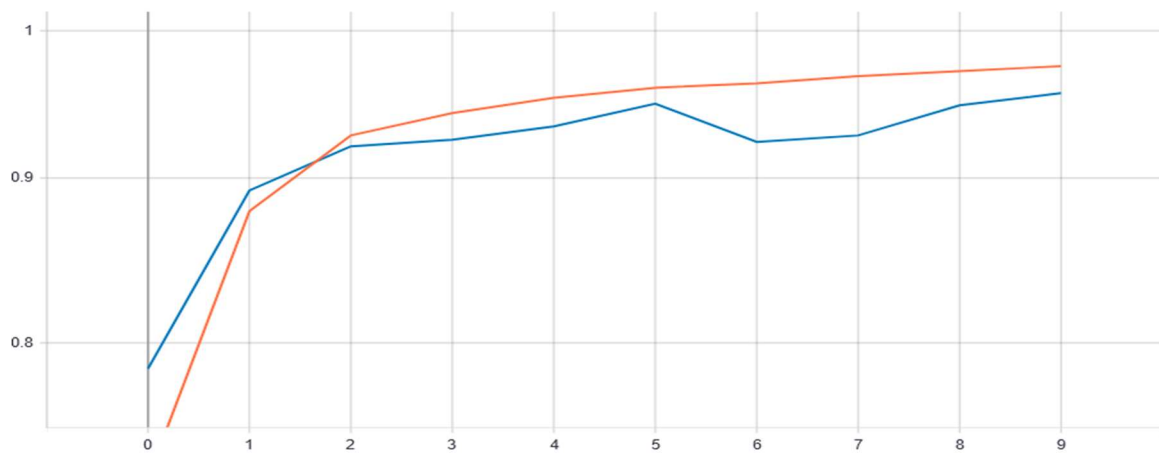


Figure 24: Train and validation AUC scores for Celeb-DF

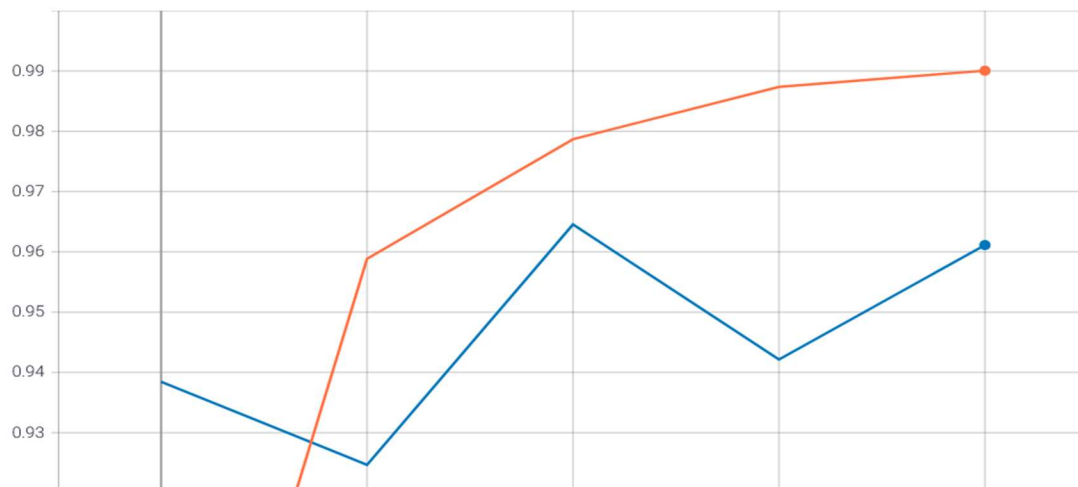


Figure 25: Train and validation AUC scores for FaceForensics++ HQ

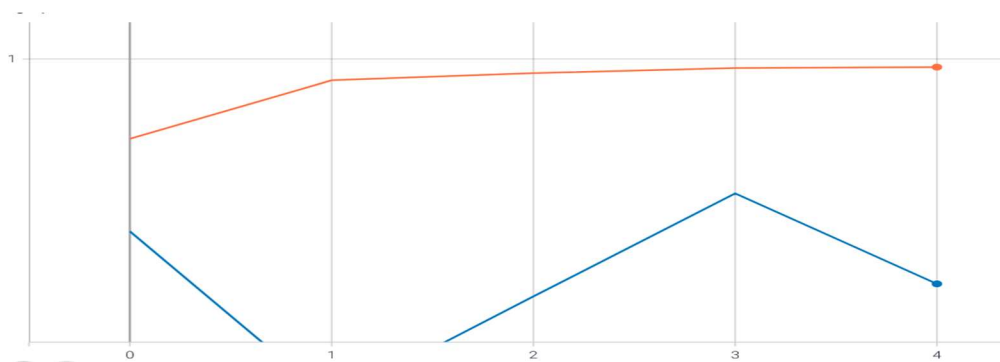


Figure 26: Train and validation AUC scores for Deeper Forensics

WEB APPLICATION

In order to interact with the designed machine learning model and to test on custom input videos provided by the user, a web application (portal) has been developed. Through this application, users can upload any video and get an accurate result whether the uploaded video is fabricated or not, obtaining results from our machine learning model.

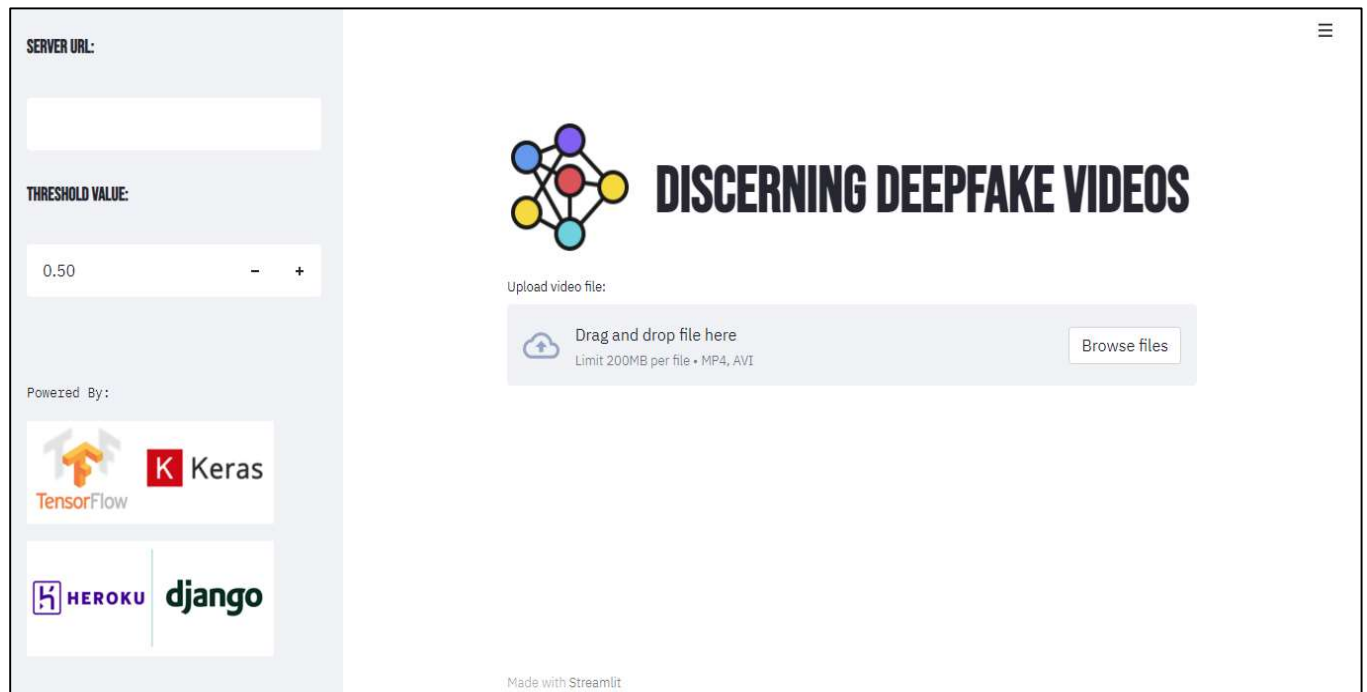


Figure 27: User interface for portal


Table 8 summarizes the dependencies of the complete web application.

USER INTERFACE (FRONTEND):

The user interface (frontend) of the application has mainly been developed in **stream lit**. Streamlit is a python-base open-source framework to create web applications, especially for machine learning and data science applications.

There are a number of input fields for the user to interact with in the application. The user can easily upload any video file using the file upload form. The supported video types include mp4 and AVI codecs, as shown in figure 28.

Upload video file:



Drag and drop file here
Limit 200MB per file • MP4, AVI

Browse files

Figure 28: Upload video

Dependency Name	Version
python	3.9.5
streamlit	0.82.0
plotly	4.14.3
Pillow	8.2.0
numpy	1.14.3
requests	2.18.4

Table 8: Application dependencies

Similarly, the user can also define the threshold value, used to compute whether the input video is a deepfake or a real video. The default threshold has been set to **0.5**.

The image shows a user interface for adjusting a threshold value. It consists of two main sections. The top section is labeled 'SERVER URL:' in bold black text, followed by a large, empty white rectangular input field. The bottom section is labeled 'THRESHOLD VALUE:' in bold black text, followed by a white rectangular input field containing the number '0.50'. To the right of the input field are two small, dark gray buttons: a minus sign '-' and a plus sign '+', used for incrementing and decrementing the value.

Figure 29: Threshold adjustment

Since we have trained separate machine learning models on different opensource datasets, in order to give user, the freedom to link the interface to any of the model backends, the backend URL can easily be specified by the user, for the interface to send video to, and receive the resultant output prediction from.

Once a video has been successfully uploaded by the user, the uploaded video is displayed to the user.



Figure 30: Uploaded video

When the resultant output is received by the frontend from backend, the results are displayed to the user as shown in figure 31.

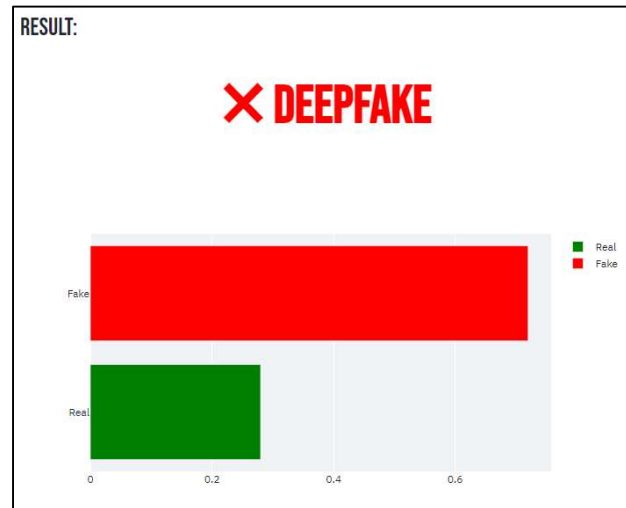


Figure 31: result

BACKEND

The backend logic and working of the web application has been implemented in **Django**. Django is a python-based web application framework, including all the components required for application development.

In addition to Django, in order to securely tunnel requests sent to Django from frontend of the web application, and its response back, **ngrok** has been used to expose Django server's address to a public URL. This public URL can then be input by the user in the specified input field in the frontend of the application, and all user requests would be directed to the entered URL.



Finally, the whole web application has been deployed using the **Heroku** platform. The application is currently live and can be accessed publicly through the following address:

<http://discern-deepfake.herokuapp.com/>

APPLICATION STRUCTURE:

The complete structure of the web application can be defined as follows:

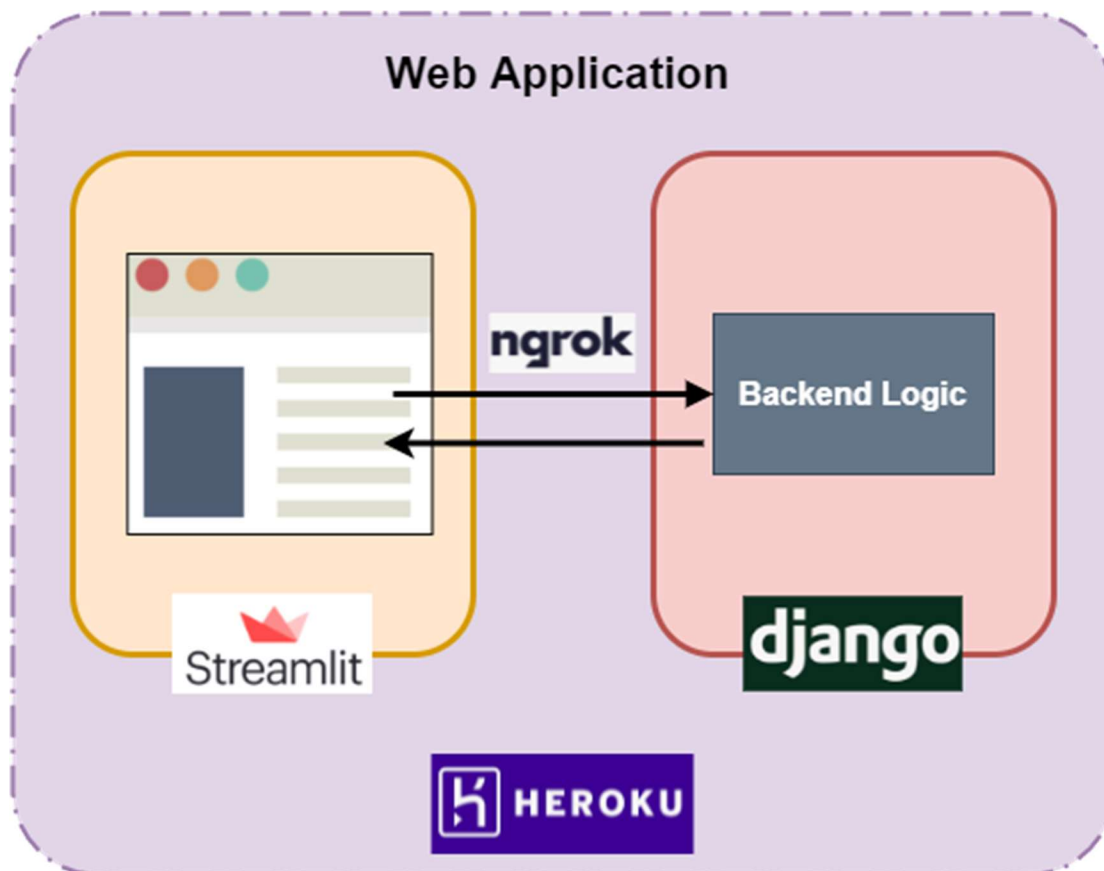


Figure 32: Application structure

CONCLUSION

It is no debate that Deepfakes are posing a serious threat to the society. In the fight against them, deep-learning based methods have shown some robustness and efficiency; however, all of deep-learning based models consist of millions of parameters.

In this regard this report presented two new deep learning based Deepfake detection models which exploit the micro and meso-scopic spatial-temporal inconsistencies in the Deepfake videos and gives state-of-the art performance on different Deepfake datasets.

Comparative analysis is also done to show that proposed the models are lightest deep learning models, in terms of parameters, until now and they rival and even outperform million parameter models in the Deepfake detection task. This makes them suitable candidates to be deployed in resource constraint environments for real time inference.

This report shows that Deepfake videos have strong spatial-temporal fingerprint, and it can easily be exploited. Further research needs to be done to increase the generalization capabilities of the proposed models so they can be on the forefront in the fight against Deepfakes.

REFERENCES:

1. Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.
2. Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.
3. Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment
4. Wayne Wu, Yunxuan Zhang, Cheng Li, Chen Qian, and Chen Change Loy. Reenactgan: Learning to reenact faces via boundary transfer
5. Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Morenó. Ganimation: Anatomically-aware facial animation from a single image
6. Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. In *IEEE WIFS*, 2018.
7. P. Korshunov and S. Marcel, “Vulnerability assessment and detection of deepfake videos,” in *International Conference on Biometrics (ICB)*, 2019.
8. Xi Wu, Zhen Xie, YuTao Gao, and Yu Xiao. Sstnet: Detecting manipulated faces through spatial, steganalysis and temporal features. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2952–2956. IEEE, 2020.
9. Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics, 2019.

10. Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. faceforensics++: Learning to detect manipulated facial images, 2019.
11. Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses, 2018.
12. W. Wang and H. Farid, "Exposing Digital Forgeries in Interlaced and Deinterlaced Video," in IEEE Transactions on Information Forensics and Security, vol. 2, no. 3, pp. 438-449, Sept. 2007, doi: 10.1109/TIFS.2007.902661.'
13. S. Prasad and K. R. Ramakrishnan, "On Resampling Detection and its Application to Detect Image Tampering," 2006 IEEE International Conference on Multimedia and Expo, Toronto, ON, Canada, 2006, pp. 1325-1328, doi: 10.1109/ICME.2006.262783.
14. M. Kirchner and R. Bohme, "Hiding Traces of Resampling in Digital Images," in IEEE Transactions on Information Forensics and Security, vol. 3, no. 4, pp. 582-592, Dec. 2008, doi: 10.1109/TIFS.2008.2008214.
15. "Exposing digital forgeries by detecting traces of re-sampling," IEEE Transactions on Signal Processing, vol. 53, no. 2, pp. 758–767, 2005
16. A. C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," in IEEE Transactions on Signal Processing, vol. 53, no. 10, pp. 3948-3959, Oct. 2005, doi: 10.1109/TSP.2005.855406.
17. J. Fridrich, D. Soukal, and J. Lukáš, "Detection of copy-move forgery in digital images," in ~ Proceedings of DFRWS, 2003
18. V. Conotter, E. Bodnari, G. Boato, and H. Farid. Physiologically-based detection of computer generated faces in video. Proceedings of the IEEE International Conference on Image Processing, pages 248–252, Oct. 2014. Paris, France

19. Luisa Verdoliva and Paolo Bestagini. 2019. Multimedia Forensics. In Proceedings of the 27th ACM International Conference on Multimedia (Nice, France) (MM '19). Association for Computing Machinery, New York, NY, USA, 2701–2702. <https://doi.org/10.1145/3343031.3350542>
20. D. Afchar, V. Nozick, J. Yamagishi and I. Echizen, "MesoNet: a Compact Facial Video Forgery Detection Network," 2018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, China, 2018, pp. 1-7, doi: 10.1109/WIFS.2018.8630761.
21. S. Fan, R. Wang, T.-T. Ng, C. Y.-C. Tan, J. S. Herberg, and B. L. Koenig. Human perception of visual realism for photo and computer-generated face images. ACM Transactions on Applied Perception (TAP), 11(2):7, 2014
22. Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. arXiv preprint arXiv:1811.00656, 2018
23. Nguyen, H. H., Yamagishi, J., and Echizen, I. (2019, May). Capsule-forensics: Using capsule networks to detect forged images and videos. In 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2307-2311). IEEE.
24. D. Güera and E. Delp. 2018. Deepfake Video Detection Using Recurrent Neural Networks. In AVSS.
25. E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.
26. Deepfake project - non official project based on original deepfakes thread. Available at github.com/deepfakes/faceswap, January 2018

27. Marek. 3d face swapping implemented in Python. Contribute to MarekKowalski/FaceSwap development by creating an account on GitHub, Apr. 2019. original-date: 2016- 06-19T00:09:07Z
28. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
29. Masi, Iacopo, Aditya Killekar, Royston Marian Mascarenhas, Shenoy Pratik Gurudatt and W. Abd-Almageed. "Two-Branch Recurrent Network for Isolating Deepfakes in Videos." *ECCV* (2020).
30. Hernandez-Ortega, J., R. Tolosana, Julian Fierrez and A. Morales. "DeepFakesON-Phys: DeepFakes Detection based on Heart Rate Estimation." *ArXiv abs/2010.00400* (2021): n. pag.
31. Qi, H.; Guo, Q.; Juefei-Xu, F.; Xie, X.; Ma, L.; Feng, W.; Liu, Y.; and Zhao, J. 2020. DeepRhythm: Exposing DeepFakes with Attentional Visual Heartbeat Rhythms. *arXiv preprint:2006.07634*
32. Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (DFDC) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019.
33. Zhao, Tianchen, X. Xu, Mingze Xu, Hui Ding, Yuanjun Xiong and W. Xia. "Learning to Recognize Patch-Wise Consistency for Deepfake Detection." *ArXiv abs/2012.09311* (2020)
34. FakeApp 2.2.0. Available at <https://www.malavida.com/en/soft/fakeapp/>
35. <https://github.com/deepfakes/faceswap>
36. <https://github.com/shaoanlu/faceswap-GAN>
37. <https://github.com/dfaker/df>
38. <https://github.com/iperov/DeepFaceLab>

39. <https://lingzhili.com/FaceShifterPage/>
40. Kramer, Mark A. (1991). "[Nonlinear principal component analysis using autoassociative neural networks](#)" (PDF). *AIChE Journal*. 37 (2): 233–243. doi:[10.1002/aic.690370209](#)
41. Y. Liao, Y. Wang, and Y. Liu. Graph regularized autoencoders for image representation. *IEEE Transactions on Image Processing*, 26(6):2839–2852, June 2017
42. Vincent, Pascal; Larochelle, Hugo (2010). "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". *Journal of Machine Learning Research*. 11: 3371–3408.
43. file from the Wikimedia commons. Commons is a freely licensed media file repository.
44. Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
45. X. Hou, L. Shen, K. Sun and G. Qiu, "Deep Feature Consistent Variational Autoencoder," 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 1133-1141, doi: 10.1109/WACV.2017.131.
46. Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014). [Generative Adversarial Networks](#) (PDF). *Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014)*
47. T. Kurutach, A. Tamar, G. Yang, S. Russell, and P. Abbeel, "Learning plannable representations with causal infogan," arXiv:1807.09341, 2018
48. T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017

49. DeeperForensics-1.0: A Large-Scale Dataset for Real-World Face Forgery Detection *Liming Jiang, Ren Li, Wayne Wu, Chen Qian, Chen Change Loy* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020
50. Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. 2020. WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection. In Proc. of ACM Multimedia
51. A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies and M. Niessner, "FaceForensics++: Learning to Detect Manipulated Facial Images," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1-11, doi: 10.1109/ICCV.2019.00009.
52. C. Bregler, M. Covell, and M. Slaney. Video rewrite: driving visual speech with audio. Proceedings of the 24th annual conference on Computer graphics and interactive techniques, 1997. 2
53. Kevin Dale Kalyan Sunkavalli, Micah Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. Video face replacement. ACM Trans. Graph., 30:130, 12 2011. 2
54. Pablo Garrido, Levi Valgaerts, Ole Rehmsen, Thorsten Thormaehlen, Patrick perez, and Christian Theobalt. Automatic face reenactment. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '14, page 4217–4224, USA, 2014. IEEE Computer Society. 2
55. Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. Real-time expression transfer for facial reenactment. ACM Trans. Graph., 34(6), October 2015. 2
56. Ricard Durall, Margret Keuper, Franz-Josef Pfreundt, and Janis Keuper. Unmasking deepfakes with simple features, 2020. 2
57. Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses, 2018
58. Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain

- image-to-image translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018
59. Pavel Korshunov and Sebastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection, 2018.
 60. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.
 61. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
 62. Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In International Conference on Computer Vision (ICCV), 2019.
 63. Davis E. King. dlib. <https://github.com/davisking/dlib>, 2011.
 64. Iván de Paz Centeno. mtcnn. <https://github.com/ipazc/mtcnn>, 2016. 6
 65. Dosovitskiy, Alexey et al. "An Image Is Worth 16X16 Words: Transformers For Image Recognition At Scale". Arxiv.Org, 2021, <https://arxiv.org/abs/2010.11929v1>.
 66. Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. Information Fusion, 64:131–148, 2020. 1, 2, 3, 7
 67. Shahroz Tariq, Sangyup Lee, and Simon S. Woo. A convolutional lstm based residual network for deepfake video detection, 2020
 68. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–9, 2015. 4
 69. Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Two-stream neural networks for tampered face detection, 2018.
 70. Falko Matern, Christian Riess, and Marc Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW), pages 83– 92, 2019. 1, 7

71. Huy H. Nguyen, Fuming Fang, Junichi Yamagishi, and Isao Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos, 2019
72. Ruben Tolosana, Sergio Romero-Tapiador, Julian Fierrez, and Ruben Vera-Rodriguez. Deepfakes evolution: Analysis of facial regions and fake detection performance, 2020
73. Hong-Shuo Chen, Mozhddeh Rouhsedaghat, Hamza Ghani, Shuowen Hu, Suyu You, and C. C. Jay Kuo. Defakehop: A light-weight high-performance deepfake detector, 2021
74. [99 Reuters , “Trump retweets doctored video of Pelosi to portray her as having 'lost it',” 2019. [Online]. Available: <https://www.reuters.com/article/us-usa-trump-pelosiidUSKCN1SU2CB>.
75. HindustanTimes, BJP’s deepfake videos trigger new worry over AI use in political campaigns. Available: <https://www.hindustantimes.com/india-news/bjp-s-deepfake-videos-trigger-new-worry-over-ai-use-in-political-campaigns/story-6WPIFtMAOaepkwdybm8b1O.html>
76. thehindu, <https://www.thehindu.com/news/national/deepfakes-enter-indian-election-campaigns/article30880638.ece>
77. The Wall Street Journal, “Fraudsters Used AI to Mimic CEO’s Voice in Unusual Cybercrime Case,” 2019. [Online]. Available: <https://www.wsj.com/articles/fraudstersuse-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11567157402>
78. [Oscar Schwartz](#) : You thought fake news was bad? Deep fakes are where truth goes to die
79. [Lex Haris](#): CBS News asks Facebook to remove "deepfake" video of Mark Zuckerberg with unauthorized CBSN trademark.
80. Drew Harwell: Scarlett Johansson on fake AI-generated sex videos: ‘Nothing can stop someone from cutting and pasting my image
81. Kate Fazzini: Trolls will use fake videos and other new tricks to try to sway the 2020 election, warns Alphabet researcher
82. Will knight: Deepfakes aren't very good|nor are the tools to detect them



DISCERNING DEEPFAKE VIDEOS

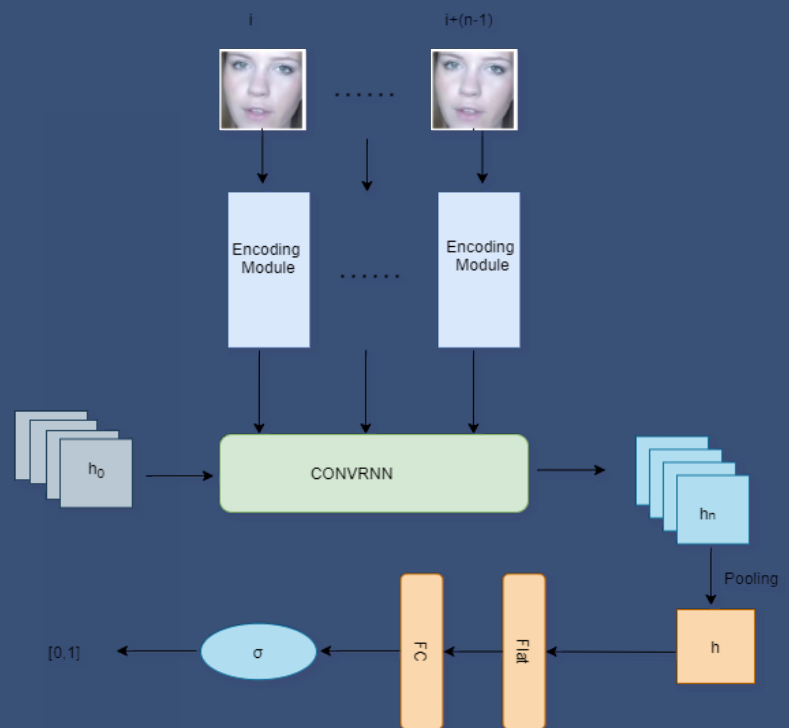


PROBLEM STATEMENT

Video and image are the most trustworthy form of evidence. However, deepfake a new technology which is helping in creation of synthetic images and videos which a naked eye cannot discern from real images and videos. In order to counter the threat of deepfake many detection algorithms have been proposed previously; however, these algorithms consist of large number of parameters which makes them unscalable.

METHODOLOGY

The Deep Learning models proposed for DeepFake detection are mammoth models with millions of parameters. We present a novel recurrent convolution layer and propose two light weight DeepFake detection models Micro-Inception CRNN and Meso-Inception CRNN with 9K and 17K parameters respectively. These lean models beat huge million parameter models at DeepFake detection.



Method Name	Number of Parameters	CelebDF (AUC)	FF++ (AUC)
Micro-ConvRNN	9,247	89.1	95.75
Meso-Inception	17,237	95.9	96.02



GROUP MEMBERS

Zaryab Muhammad Akram
Muneeb ur Rehman
Muhammad Anas Tahir

FACULTY ADVISORS

Dr. Muhammad Imran Malik (Advisor)
Dr. Muhammad Shehzad (Co-Advisor)