MIM-VIT: DEEPFAKE DETECTION USING MASKED IMAGE MODELLING AND VISION TRANSFORMER

B. TECH. PROJECT REPORT

SUBMITTED BY

SAMEER KAVTHEKAR 111903153

SHREYA VAIDYA 111903156

VISHWESH PUJARI 111910127

UNDER THE GUIDANCE OF

PROF. SUNIL B. MANE

COLLEGE OF ENGINEERING, PUNE

DEPARTMENT OF COMPUTER ENGINEERING AND INFORMATION TECHNOLOGY,

COLLEGE OF ENGINEERING, PUNE-5

MAY 2023

SBM_03

ORIGINALITY REPORT

11%	7%	8%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	aniketbhadane.github.io Internet Source	1%
2	Yang Yu, Xiaohui Zhao, Rongrong Ni, Siyuan Yang, Yao Zhao, Alex C. Kot. "Augmented Multi-Scale Spatiotemporal Inconsistency Magnifier for Generalized DeepFake Detection", IEEE Transactions on Multimedia, 2023 Publication	<1%
3	mattdeitke.com Internet Source	<1%
4	arxiv.org Internet Source	<1%
5	"ECAI 2020", IOS Press, 2020 Publication	<1%
6	"Image Analysis and Processing – ICIAP 2022", Springer Science and Business Media LLC, 2022 Publication	<1%
7	www.researchgate.net	

Dr. Sunil B. Mane
Associate Professor
Dept. of Computer Engg & IT
College of Engineering,
Pune - 411 005.

Abstract

Over the last decade, deep learning has become one of the fastest-growing fields in computer science. It finds applications in several sectors, such as Healthcare, Agriculture, Financial Services, and Crime Investigation. However, with the rapid development of General Adversarial Networks in 2017, the concept of deepfakes emerged. The term deepfake refers to an artificially synthesized image or video generated using techniques like Face Swapping and Face Expression Re-enactment. Spreading misleading news via politicians or celebrities is one such example. Such ill-intended videos can exacerbate the increasingly prevalent problem of false information online. Therefore, being able to differentiate between real and fake videos is crucial. This work proposes a solution based on Masked Image Modelling using Auto-Encoders and Vision Transformers to tackle the problem of Deepfake Detection. The solution consists of two sub-models working in parallel, namely the Multi-scale Vision Transformer and the Convolutional Neural Network framework, ConvNeXt V2. A novel facial quality detection algorithm is developed, which helps improve the data quality by overcoming the challenge of misrepresented facial data. The proposed model, MIM-ViT, achieves competitive results on popular datasets like the Deepfake Detection Challenge Preview and Face Forensics++ with a test accuracy of 80.22% and 98.1% along with an Area Under the Receiver Operating Characteristics score of 84.48% and 99.8% respectively. The proposed model generalizes well and performs competitively on unseen data, achieving an AUCROC score of 68.21%. Additionally, the model obtains an AUCROC score of 68.71% on the multi-face dataset, Face Forensics in the Wild 10K. The proposed model is utilized in two real life applications - a video streaming platform and a Discord bot.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Deepfakes are synthetic media that use neural networks, to manipulate or replace the original content of an image, audio or video file, in order to create a new version that appears to be authentic but is actually fake. The term "deepfake" is derived from the words "deep learning" and "fake". Deepfakes can be used to superimpose a person's face onto another person's body or to alter their voice, making it seem like they are saying or doing something they never actually did. They can also be used to create entirely new digital personas, complete with fake images and videos of a person who does not exist in reality.

While deepfakes can be used for entertainment purposes, such as creating funny videos or memes, they also have the potential to be used maliciously, such as in online scams, disinformation campaigns, or to manipulate public opinion. Deepfakes have been a growing concern in recent years due to their potential for misuse, and efforts are underway to develop methods for detecting and preventing the spread of deepfakes.

Deepfake detection is the process of identifying whether an image, video or audio file has been manipulated using deep learning algorithms. Following are the underlying concepts used for deepfake detection.

Forensic analysis & Anomaly detection: Forensic analysis is a scientific method of examining physical or digital evidence in order to identify the origin, authenticity, and integrity of the evidence. In the context of deepfakes, forensic analysis can be used to detect any inconsistencies or anomalies in the video or image that suggest tampering. This involves examining a wide range of features, including metadata, pixel patterns, and facial movements. Pixel patterns are important features that forensic analysis can examine. In a deepfake, the pixels of the manipulated image or video may be of a different quality or resolution than the original, which can be detected through careful analysis of the pixel patterns. This can include examining the sharpness and clarity of the image, as well as any irregularities in the pattern. For example, a deepfake image may have noticeable blurriness or distortion around the edges of the face, or a deepfake video may have inconsistent lighting or shadows.

Facial and body movement analysis: Facial and body movement analysis involves the detailed examination of physical movements, expressions, and gestures of an individual in a video or image. This analysis is based on the fact that deepfakes often involve subtle inconsistencies or unnatural movements that are difficult to replicate using artificial intelligence. By analyzing these movements, it can be determined whether a video or image has been manipulated.

Examining the timing and duration of facial expressions is a key aspect of facial and body movement analysis. Deepfakes often involve facial expressions that are not aligned with the audio, or expressions that do not match the context of the situation. For example, a deepfake may show a person smiling during a solemn moment, or exhibiting an expression of fear during a mundane activity. These inconsistencies can be detected through the analysis of the timing and duration of facial expressions, which can reveal whether the movements are genuine or have been artificially created.

Audio analysis: Deepfake audio files are created by using machine learning algorithms to generate a voice that sounds like a real person. By analyzing the frequency patterns in the audio, it is possible to detect whether the voice has been artificially generated. Audio analysis includes examining the spectral characteristics of the recording. Spectral analysis involves breaking down an audio recording into its constituent frequency components. This analysis can reveal inconsistencies in the spectral content of the recording, which can be used to determine whether an audio clip has been manipulated. For example, a deepfake may have a voice that sounds unnaturally high or low, or may have a background noise that is inconsistent with the surrounding sounds. Speech analysis is another important aspect of deepfake detection using audio. In deepfakes, the speech may be distorted or misaligned, or may have unnatural pauses or hesitations.

Using a combination of the above, a deep learning based solution can be implemented to provide an effective solution to the problem of identifying deepfakes.

Deepfake Detection using Deep Learning

Deep learning involves the use of artificial neural networks that can learn from large amounts of data to recognize patterns and make predictions. These networks are designed to mimic the structure and function of the human brain, with layers of neurons that process and analyze data. By training a deep neural network on a large dataset of both real and manipulated media, the network can learn to recognize the subtle differences between them, and use this knowledge to detect deepfakes.

Deepfake detection using deep learning involves analyzing the visual characteristics of a media file, such as images or videos. This involves training a neural network to recognize the unique features of a person's face or body, such as facial expressions, movements, and other visual cues. A deepfake may have inconsistencies in the way a person's face or body moves, or may have unnatural lighting or shading that is inconsistent with the surrounding environment. A deep neural network is trained to recognize these patterns allowing it to identify fake media with a high degree of accuracy.

Deepfake Detection is an ongoing challenge, as the technology used to create deepfakes is constantly evolving, the methods used to detect them must also evolve to keep up with these changes. In this work, a Masked Image Modelling-Vision Tranformer (MIM-ViT) model along with a novel Face Quality testing algorithm is proposed to overcome the problem of deepfake detection. A series of experiments are carried out to test the performance of the proposed solution.

Chapter 2

Literature Review

At the onset of the twenty-first century, deep learning was introduced as a new sub-field of machine learning. Not long after, the term deepfake was coined by a Reddit user of the same name. Owing to the advancements made in the domain of face forgery and deepfake generation, the spread of false information via deepfakes has become a topic of concern. A substantial amount of research has been carried out in this domain and is discussed in the following subsections.

2.1 Deepfake Generation

Face manipulation methods are broadly classified into two categories - Face Swap and Facial Expression Reenactment. Korshunova et al. [1] propose a real-time face swapping mechanism using Convolutional Neural Networks which works on the principle of simple face replacement. The quality of the generated images were enhanced and made to look more realistic using style transfer on neural networks. Li et al. [2] demonstrate the generation of synthetic faces given a pair of real faces without subject specific training. The proposed novel framework addresses the problem in existing low fidelity face generation techniques and achieves superior results.

Kim et al. [3] take a minimalistic approach towards face swapping by using a barebones U-net architecture coupled with smooth identity embedding. In comparison with existing techniques, Smooth-Swap had the least complexity and fastest convergence time. Zhang et al. [4] present a model FusionNet for one shot facial reenactment. Using an auto encoder-adaptive decoder architecture and disentangle learning, improved performance on textured images was obtained.

2.2 Deepfake Detection using CNNs

Convolutional Neural Networks are one of the most popular models used in many image / video processing applications. A variety of CNN frameworks have been used to provide a solution for Deepfake Detection. Guera et al. [5] deployed a linear model comprising of two components - CNN for feature extraction and LSTM for sequence processing. A self-collected dataset was used and a testing accuracy of 97.1% was achieved. de Lima et al. [6] performed deepfake classification using the network architectures - RCN, R3D, ResNet Mixed 3D-2D Convolution, ResNet (2+1)D, and I3D. A comparison between all the spatiotemporal models is made on the basis of accuracy and ROC-AUC scores. Chang et al. [7] use an SRM filter layer to highlight the noise information of a given input image. The image noise is weakened using various data augmentation techniques and the model is tested against the Celeb-DF dataset. Zhao et al. [8] postulate the problem of identifying deepfakes to be a fine grained classification task. Attention is decomposed into multiple regions allowing local features to be collected leading to improved results.

2.3 Deepfake Detection using ViTs

In recent years, vision transformers (ViT) have emerged as a promising technique for deepfake detection due to their ability to analyze images and detect patterns in them.

Wodajo et al. [9] generated a large dataset of real and deepfake videos, which they use to train their model. They then developed a hybrid architecture that uses both CNNs and vision transformers to analyze the spatial and temporal features of the video frames. The authors use a self-attention mechanism to help the vision transformers focus on the most relevant frames. Coccomini et al. [10] propose a method for detecting deepfake videos using a combination of EfficientNet and vision transformers. A temporal aggregation method is used to capture the temporal relationships between frames and generate a video-level feature representation. State-of-the-art performance is achieved on the Deepfake Detection Challenge (DFDC) dataset. Ganguly et al. [11] extract features from the input images and videos using an Xception network. These features are passed through a vision transformer. A multi-scale approach is incorporated, where the input images and videos are resized to different scales to capture information at different resolutions. Wang et al. [12] propose a multi-modal approach, where both visual and audio information are extracted from the input images and videos. Convolutional neural network (CNN) is used to extract visual features and a recurrent neural network (RNN) is used to extract audio features. The combination of multi-modal and multi-scale transformers allows for accurate feature extraction and captures both spatial and temporal information, while the different types of transformers capture information at different scales.

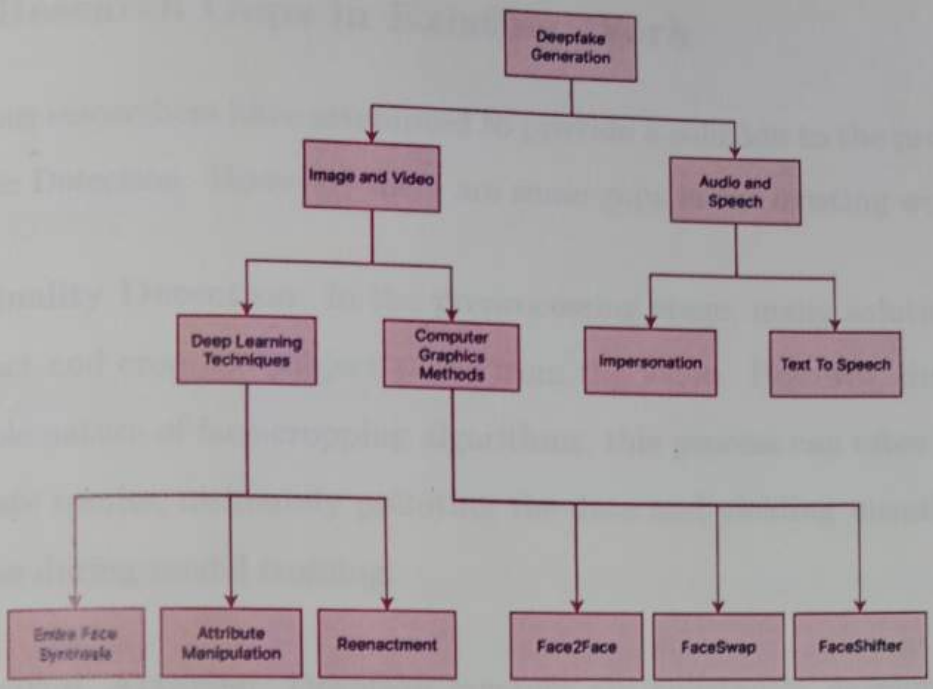

Figure 2.1: Taxonomy of Deepfake Generation

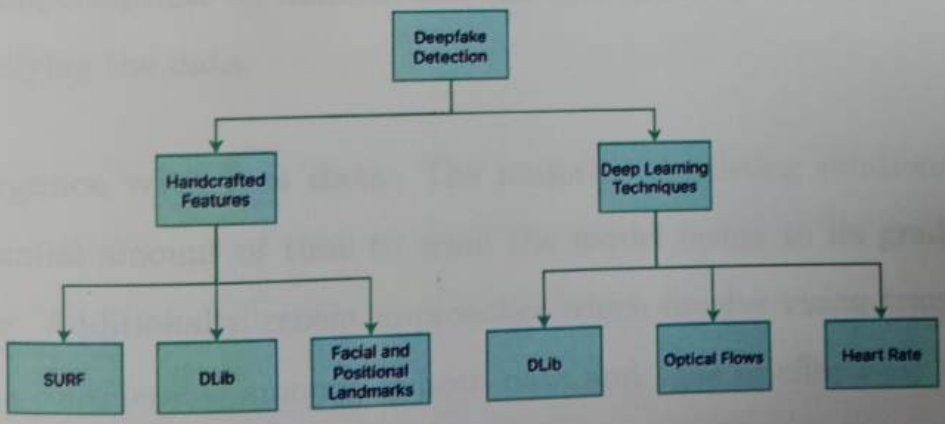

Figure 2.2: Taxonomy of Deepfake Detection

2.4 Research Gaps in Existing Work

Numerous researchers have attempted to provide a solution to the problem of Deepfake Detection. However, there are some gaps in the existing work.

Face Quality Detection: In the preprocessing stage, many solutions tend to extract and crop the subject's face from the video. However, due to the unreliable nature of face-cropping algorithms, this process can often produce inaccurate results, ultimately polluting the data and yielding unsatisfactory outcomes during model training.

Adversarial Attacks: Deepfake creators are constantly developing new ways to evade detection, such as using adversarial attacks to fool deep learning models. Adversarial attacks involve manipulating the input data in a way that is imperceptible to humans but can fool the deep learning model into misclassifying the data.

Convergence with less data: The majority of existing solutions require a substantial amount of time to train the model owing to its gradual convergence. Additionally, recent approaches which involve vision transformers require a considerable amount of both data and time to effectively train the model.

Multi-face Deepfake Detection: Most of the work carried out in the domain of deepfake detection has been with respect to single face datasets. More research is needed to develop a model that can accurately identify deepfakes with multiple faces.

Chapter 3

Problem Statement

The number of deepfake videos has been multiplying rapidly in the past 5 years. The alarming rate of growth of deepfakes is a major cause of concern. Deepfake detection is important for several reasons:

Preventing Misinformation: Deepfakes can be used to spread false information and manipulate public opinion. In the wrong hands, they can be used to create fake news, mislead people, and cause public unrest.

Protecting Personal Privacy: Deepfakes can be used to create fake images or videos of individuals, which can be used to damage their reputation, invade their privacy, or even be used for blackmail.

Ensuring Digital Trust: As deepfakes become more advanced and harder to detect, it can become more difficult to distinguish between real and fake content, making it harder to trust anything we see or hear online.

This work proposes a deep learning based approach to the above-stated problem statement which addresses three major research gaps, Multi-face Deepfake Detection, Face Quality Detection and Convergence with less data, mentioned in Section 2.4.

Chapter 4

Proposed Solution

This work proposes a novel architecture to allow accurate classification of Deepfake videos. The general flow of the solution is shown in Figure 4.1.

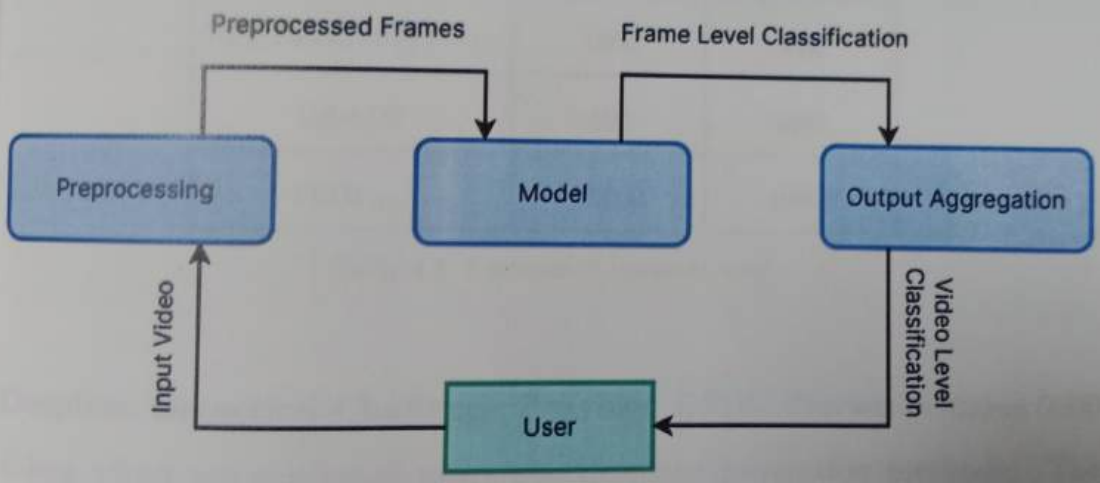

Figure 4.1: Data flow diagram for proposed solution

Each component is discussed in depth in the following subsections.

4.1 Dataset

Three datasets have been utilized in this work, namely, Deepfake Detection Challenge Preview (DFDC-P) [13], Face Forensics (FF++) [14], and CelebDF [15]. These datasets provide over 15,000 training, testing, and validation videos. The chosen datasets are preferred due to the different variations in the methods of generating the fake videos, the environments, the means of acquisition, the pose as well as the actors considered. The summary of the datasets is given in Table 4.1.

Dataset	Real Videos	Fake Videos
DFDC Preview	1460	3754
Face Forensics (FF++)	1000	5000
CelebDF	590	5639
$FFIW_{10K}$	10000	10000

Table 4.1: Features of datasets used

Deepfake Detection Challenge Preview: DFDC Preview contains 5000 videos which are generated using two deepfake generation methods. The original sequences contain videos from YouTube and collaboration with paid actors. All the videos contain a variety of gender, ages, skin tone, and chosen arbitrary backgrounds.

FaceForensics (FF++): Face Forensics contains 6000 videos. The original sequences are sourced from YouTube. 5 deepfake generation techniques are applied to each of the original videos. These techniques include Deepfake, FaceShifter, Face2Face, NeuralTextures, and FaceSwap. The authors provide the dataset in varied lossy compression levels, compressed using FFMPEG. The three levels are dubbed Raw (No compression), High Quality (Compression factor of 23%), and Low Quality (Compression factor of 40%).

CelebDF: CelebDF contains over 5000 videos of synthesized videos, specifically of celebrities. The authors propose an improvised synthesis process, which they initially apply to the sourced videos. This dataset represents the deepfakes that are popularly available freely on various Internet sources. The purpose of this dataset is to provide high-quality deepfakes compared to previous work.

Figure 4.2: Sample images from FaceForensics++, DFDC Preview and CelebDF

Face Forensics in the Wild (FFIW10K): FFIW10K [16] is a dataset comprising of 10,000 real and 10,000 fake high-quality videos, each with an average duration of 12 seconds, resulting in a total video length of 33 hours. Most of the deepfake detection datasets that are currently available, such as CelebDF, FaceForensics++, DFDC, and others, only feature one person in each video. In FFIW10K, each video involves multiple individuals, however,

only some of the faces are manipulated. As a result, FFIW10K provides a suitable representation of real-world videos featuring multiple individuals. Each video in the dataset contains an average of 3 individuals, with a minimum of 1 and a maximum of 15 individuals. To ensure the quality of forged videos, a Quality Assessment Network (Q-Net) is employed. FFIW10k offers both video-level and face-level annotations, enabling training and testing to be done in various ways. Video-level annotations indicate whether a video is real or fake, whereas face-level annotations indicate whether a face in a video is real or fake. Each source video in the dataset has a corresponding target video and a target-mask video. The source video is the real video, while the target video is the manipulated video in which only the faces with masks provided in the target-mask video are altered. Thus, the target-mask videos are providing face-level annotations.

Challenges posed by FFIW10K:

- Only a few faces are manipulated in forged videos, the majority of faces in these videos are genuine.
- Some frames in a video do not include any individuals or faces.
- In a video which includes a total of n individuals, not all n individuals to appear in every frame of the video.

Figure 4.3: Sample Source, Target and Target Mask from FFIW10K Dataset

4.2 Preprocessing

Datasets are preprocessed separately based on the nature of the images - single face or multi face. Both preprocessing techniques are discussed in the following subsections.

4.2.1 Single Face

Most deepfake syntheses only manipulate the faces of real people. Therefore, all the videos in each dataset are cropped around the face region. A deep learning-based cutting-edge face detector called RetinaFace [17] is used. RetinaFace's face detection yields good detection results even in crowd photos. Extensive experimental results show that RetinaFace can simultaneously achieve stable face detection, accurate 2D face alignment, and robust 3D face reconstruction while being efficient through single-shot inference.

In the proposed method, 32 frames are extracted from each video by sampling one frame in every N/32 frames (where N is the total number of frames in each video). Facial landmark information from each frame is obtained using RetinaFace. The landmark information is used to crop the face from the frame. The frames are then merged back into a video.

Figure 4.4: Face extraction using RetinaFace

4.2.2 Multi Face

In this work, face-level labels have been taken into consideration for detecting multi-person deepfake videos. Hence, in the preprocessing stage, every multi-person video is converted into multiple single-person videos. If a video contains n different individuals, then n different videos are generated, with each video containing the face-cropped frames of only a single individual.

Preprocessing of single-person videos involved generating 32 face-cropped frames by processing 32 random frames from the video. However, for FFIW10K dataset, the challenges mentioned in the previous subsection necessitate processing all frames of the video to obtain faces of all individuals, unlike in single-person videos where processing 32 random frames was sufficient. Moreover, in a video that includes multiple individuals who may not be stationary in their position, it is crucial to track each individual's face across all frames to establish an association between faces in different frames. Both these requirements (processing all frames and tracking faces across frames) increase the time needed for preprocessing in comparison to the single-person scenario.

A multiple object tracker Deep SORT [18] has been used to track faces through the video. Deep SORT is an extension to the Simple online and realtime tracking (SORT) algorithm. Deep SORT is capable of reducing the number of identity switches by tracking objects over extended periods of occlusion. For tracking purposes, Deep SORT takes a bounding box around the object to be tracked in the form of co-ordinates, along with the features of that object. Deep SORT assigns a unique identifier to every tracked object.

All the frames of a video are sequentially given to RetinaFace which then gives facial landmark information of all the faces. That information in the form of bounding boxes and features is then passed to Deep SORT for tracking. Deep SORT assigns a unique identifier to every face, using which each individual's face-cropped frames are clubbed together. After processing all the frames of the video, from the face-cropped frames of each individual, 32 frames are randomly selected, resized to (224, 224, 3), and used to create a video for that individual. This process is repeated for all individuals, creating a separate video for each person.

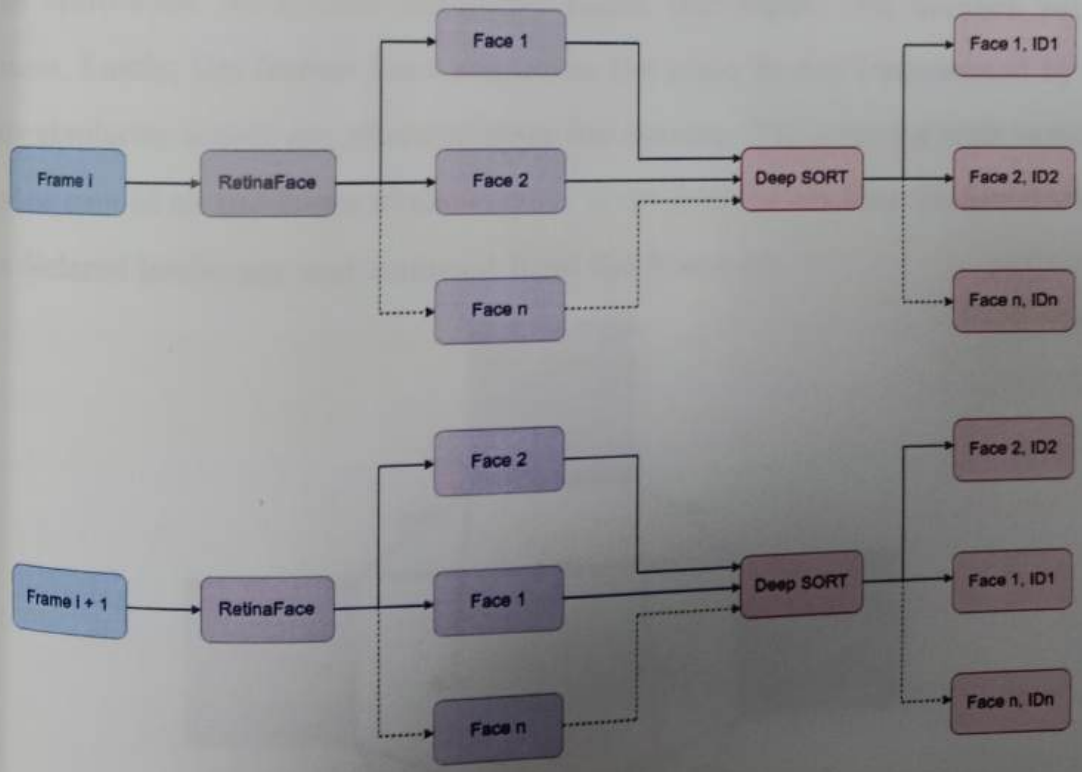

Figure 4.5: Preprocessing of multi-face dataset using Deep SORT

4.3 Face Quality Detection

Since RetinaFace's face detector yields landmark information in which no faces are present, we propose a novel strategy to detect frames in which no face is found. The false frames are removed from the dataset, and another frame is sampled.

In the proposed method, a complete undirected graph [V, E] is constructed, where each vertex is a sampled frame. Every edge contains the similarity score between the two respective frames. This score is calculated by using the Multi-scale Structural Similarity Index (MS-SSIM) [19] between each frame. Lastly, the frames least related to the other frames (represented by a low similarity score) are removed from the dataset. The score for each vertex is the sum of all the edges of the vertex. If the score is less than 20, the frame is declared irrelevant and removed from the frame set.

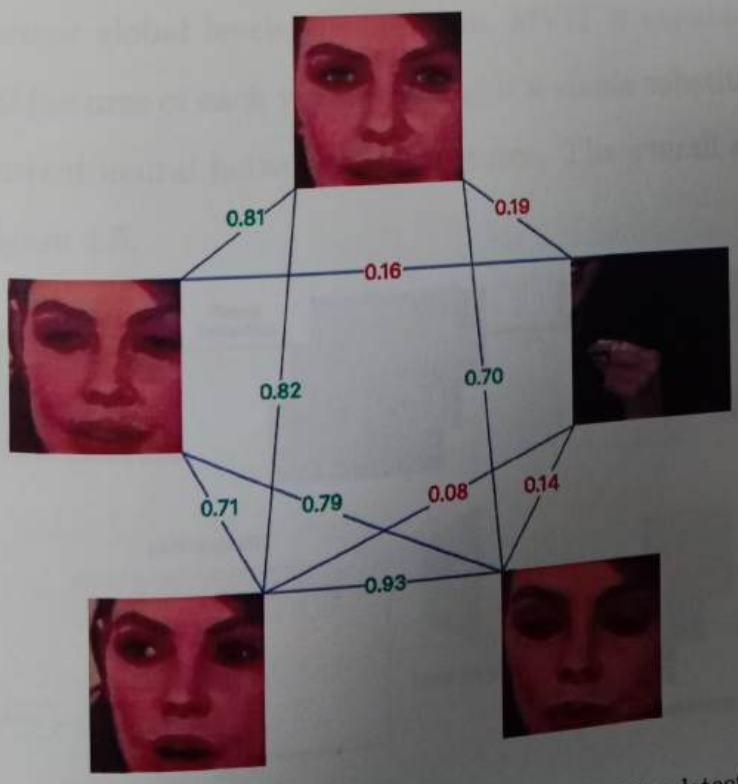

Figure ... directed graph for face quality detection

4.4 Model

The preprocessed and cropped faces obtained from a video are first given as an input to ConvNeXt v2 [20]. The output from the masked auto-encoder, ConvNeXt is a series of feature maps which are passed as inputs to the Multi-scale Vision Transformer (MViTv2) [21] model. The output embedding from the MViT is passed through a Multi Layer Perceptron network. Similarly, the feature maps from the ConvNeXt are passed to another Multi Layer Perceptron network. Finally, the outputs are aggregated, softmax is applied and the final output is used for classification. The two sub-models were taken from Pytorch Image Models [22].

Since each model detects different features, they are configured to operate in parallel. The Convolutional Neural Network (CNN) backbone is specialised in identifying local features, while the Vision Transformer (ViT) can recognize features on various global levels. In addition, MViT is capable of detecting spatiotemporal features of each video, making it a viable substitute for LSTM and GRN recurrent neural network architectures. The overall architecture is depicted in Figure 4.5.

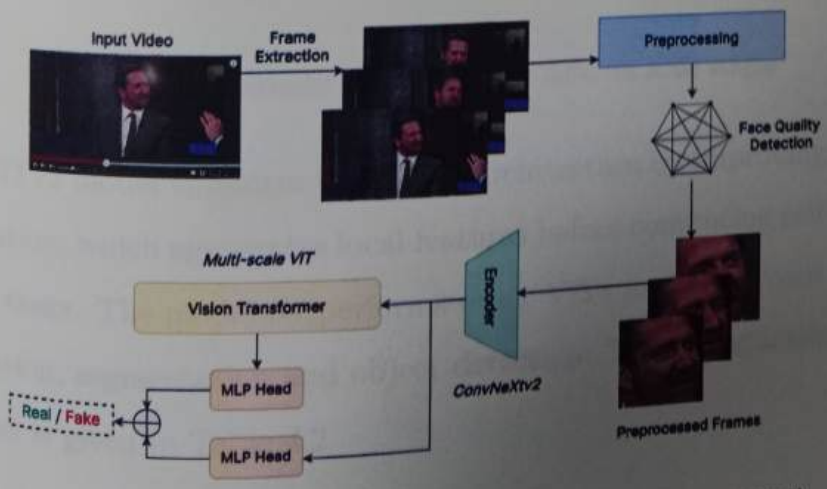

Figure 4.7: Architecture diagram for proposed model (MIM-ViT)

Multiscale Vision Transformer (MViT)

MViTv2, which is a Multiscale Vision Transformer, is particularly effective in image or video classification. MViTv2 can be used as a general-purpose vision backbone for spatial and spatiotemporal recognition tasks. Since both of these tasks are required for detecting deepfake videos, the feature maps obtained from the preprocessed videos are fed into MViTv2.

Unlike other transformer architectures, the MViT architecture works at many different scale stages as shown in Figure 4.6 [23]. At each scale stage, the feature resolution decreases while the feature channel depth increases. The basic transformer block arrangement is shown in Figure 4.7.

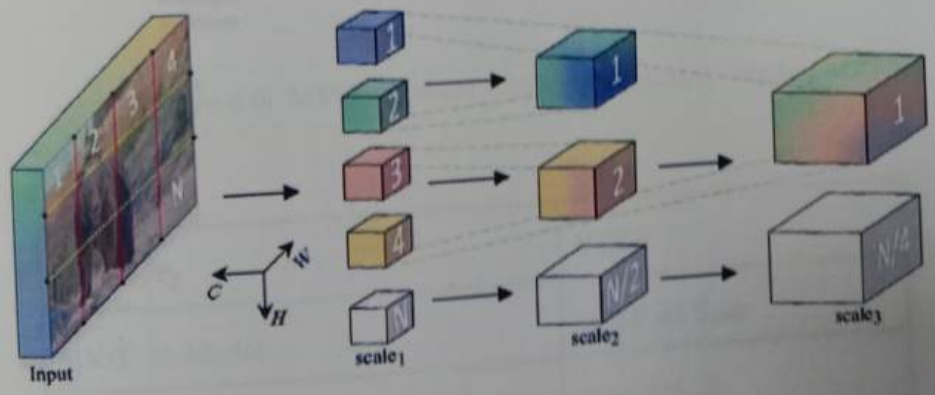

Figure 4.8: Hierarchy of learning over different scale stages

The MViTv2 model enhances the MViT architecture by improving the pooling attention, which aggregates local features before computing self-attention in video tasks. The model outperforms other ViTs in downstream tasks like classification, segmentation and object detection. The configuration used in our model is given in Table 4.2.

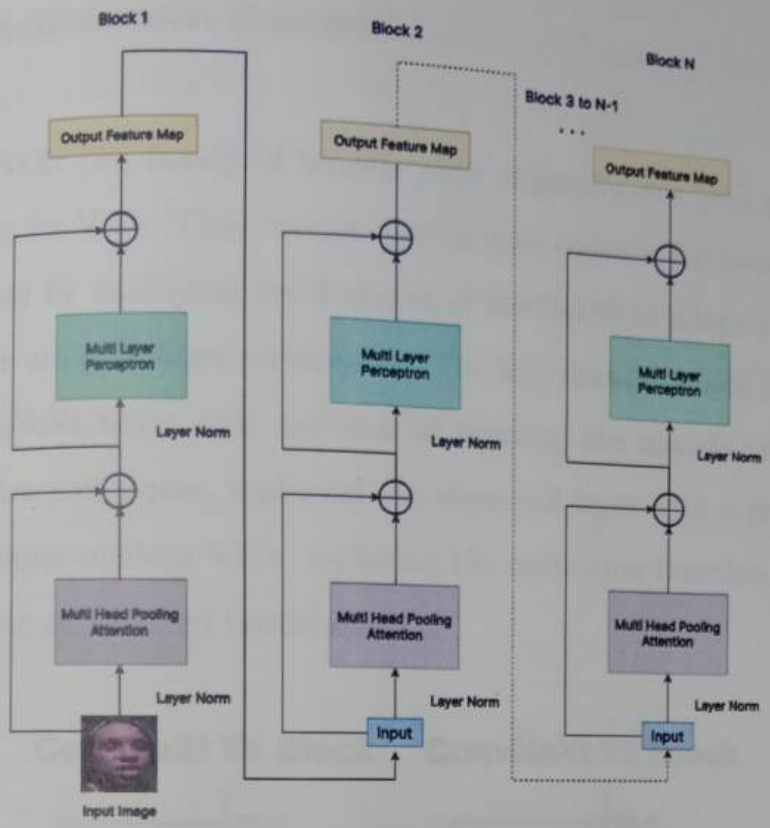

Figure 4.9: MViT Internal Block Arrangement diagram

Property	Value
MViT v2 Model	MViT v2 Base
Input Size	(224, 224, 3)
Patch Size	(56, 56)

Table 4.2: Configuration for MViT v2

Masked Autoencoder: ConvNeXt

The ConvNeXt [24] family of models were originally designed as a drop-in replacement for ViTs. They consist only of pure convolution based blocks. It was designed by modifying key features of ResNet50 to adapt them to perform better using modern techniques. The key changes made in the structure of ResNeXt were: new methods of training the models using modern augmentation techniques, replacing the stem cell layer with a patchify layer similar to other modern ViTs, replacing the activation function with GeLU and adopting an inverted bottleneck.

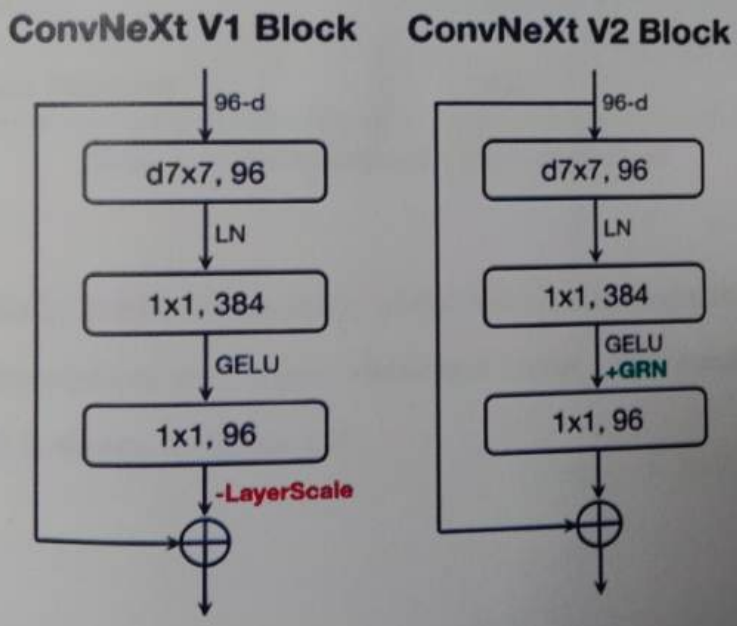

Figure 4.10: Differences between the building blocks of the two ConvNeXt families as illustrated by Woo et al.

ConvNeXt V2 is designed as an improvement to the original ConvNeXt. The major changes that contribute to the performance improvements are: training using self supervised learning using methods like Masked Auto Encoders (MAE) [25] and, introduction of the Global Response Normalization (GRN) layer as opposed to the Layer Normalization used by both ConvNeXts as well as all ViTs. The difference between the blocks of ConvNeXt and ConvNeXt V2 families is highlighted in Figure 4.8 [20].

Property	**Value**
ConvNeXt v2 Model	ConvNeXt v2 Base
Input Size	(224, 224, 3)
Output Dimension	1024

Table 4.3: Configuration for the ConvNeXt v2

The model family outperforms state of the art transformers in classification, semantic segmentation and object detection tasks. The configuration used by our model is shown in Table 4.3.

Chapter 5

Experimental Setup

The model (MIM-ViT) is trained on 4 Tesla V100 cards on the NVIDIA DGX workstation using the PyTorch 22.05 Docker container. Both the individual submodels are pre-trained on ImageNet21K dataset. Using transfer learning, the model is trained on the aforementioned datasets (FaceForensics++ and DFDC Preview) with an 80-20 train-test split.

To evaluate the model Binary Cross Entropy Loss was used. The optimization was carried out using Adam optimizer defined with the hyperparameters given in Table 5.1.

Name	**Value**
Epochs	15
Batch Size	8
Learning Rate	3×10^{-4}
Decay Rate	10^{-5}
Dropout Rate	0.4

Table 5.1: Hyperparameters used during training

Chapter 6

Results and Discussion

Performance Metrics

To measure the performance of the proposed model, three metrics of performance are used: Accuracy, AUC ROC and log loss.

Accuracy: Accuracy is defined as the total number of correct predictions mode by the model compared to the total number of samples in the dataset.

$$Accuracy = \frac{True_{positive} + True_{negative}}{True_{positive} + True_{negative} + False_{positive} + False_{negative}}$$

Area Under the Receiver Operating Characteristics: The ROC curve is a graphical representation of the probability distribution of a classifier's performance, while the AUC is a measure of the degree of separability between the classes. It indicates the model's ability to accurately differentiate between the classes, with a higher AUC suggesting better performance in predicting the correct class of the sample.

Experiments

Three types of experiments are conducted in this study: In-dataset experiments where the train, test and validation splits are taken from the same dataset, Cross-dataset experiments, where the train and validation splits are from one dataset and the testing from another dataset. This experiment helps to understand the generalizability of the model. Finally, an ablation study on the impact of the proposed face quality detection technique is tested.

For the in-dataset and the cross-dataset studies, a few baseline models are considered for a comparative study. MIM-ViT is compared with the baseline methods which are: Recurrent-network [5], Multi Attention [8], Xception [14], FWA [26], MesoInception4 [27], Multi-task [28], Capsule [29], EfficientNet-B4, SPSL [30], LTW [31], Two-branch [32], and F3-Net [33].

Methods	CelebDF		DFDC	
	Accuracy	**AUC**	**Accuracy**	**AUC**
Recurrent-network [5]	71.20	86.52	75.02	77.48
FWA [26]	64.73	60.16	73.25	72.97
Xception [14]	90.34	89.75	79.32	81.58
FT-two-stream	80.74	86.67	63.85	64.03
FInfer [34]	90.47	93.30	80.39	82.88
MIM-ViT (Proposed Method)	**92.82**	**98.80**	**80.22**	**84.48**

Table 6.1: Comparison with baseline method of in-dataset evaluation (in terms of Accuracy% and AUC%) on CelebDF and DFDC Preview

In-Dataset Experiment

(i) *Single Face*

In this experiment, two datasets namely CelebDF and DFDC Preview are considered. Train-test-validation splits are made from the preprocessed datasets. The model is trained and tested on splits from the same dataset. The results of the same are shown in Table 6.1.

The proposed model outperforms previous solutions by 5% on the CelebDF dataset and by 2% on the DFDC Preview dataset.

(ii) *Multi Face*

In the multi-face in-dataset experiment, the training and validation sets of the FFIW10K dataset are used for training the model, and the testing set is used to obtain the result highlighted in Table 6.2. All three splits are preprocessed using the technique mentioned in Section 4.2.2. The proposed model performs competitively and is shy of 1.19% from the state-of-the-art model.

Methods	FFIW10K
Xception	56.1
MesoNet	55.4
PatchForensics	61.6
FWA	63.1
FFIW	70.9
MIM-ViT (Proposed Method)	**68.71**

Table 6.2: Comparison with baseline method of in-dataset evaluation, trained and tested on FFIW10K (in terms of AUC%)

Cross-Dataset Experiment

In this experiment, two datasets namely FaceForensics++ and CelebDF are considered. Train-validation splits are made from the preprocessed FF++ dataset. The model is trained on FF++ and tested on FF++ and CelebDF. The results of the same are shown in Table 6.3.

The proposed model yields the same results as previous state of the art solutions on the FF++ dataset and performs competitively on the CelebDF dataset. SPSL [30] yields the best result on the CelebDF dataset.

Methods	FF++	CelebDF
Recurrent-network [5]	86.52	63.56
FWA [26]	80.10	56.90
Xception [14]	99.70	65.30
Multi-task [28]	76.30	54.30
Capsule [29]	96.60	57.50
SPSL [30]	96.91	76.88
Two-branch [32]	93.18	73.41
Multi Attention [8]	99.80	67.44
FT-two-stream	86.67	65.56
F3-Net [33]	98.10	65.17
FInfer [34]	95.67	70.60
MIM-ViT (Proposed Method)	**99.80**	**68.21**

Table 6.3: Comparison with baseline method of cross-dataset evaluation, trained on FaceForensics++ (in terms of AUC%) on CelebDF

Ablation Study

In this experiment, the model performance is tested with and without applying the proposed face quality algorithm during the preprocessing process. The results are recorded in Table 6.4. The results have been calculated using the FF++ dataset for training and testing.

Removal of the face quality algorithm yields results that are 4% inferior to the model performance when the face quality algorithm is applied.

Face Quality applied	AUC% on FF++
No	94.12
Yes	99.80

Table 6.4: Ablation study on effect of application of proposed face quality detection algorithm

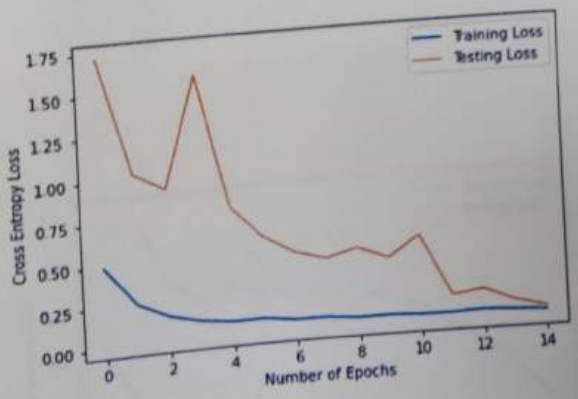

Figure 6.1: Training and testing loss on the DFDC Preview dataset

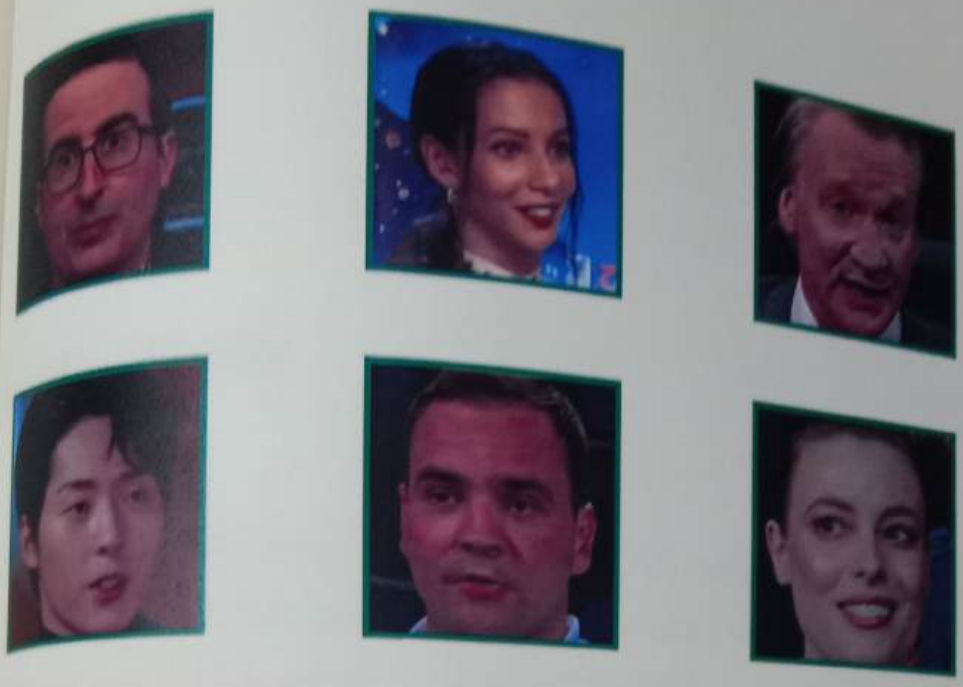

Figure 6.4: Extracted frames from videos classified as real by MIM-ViT

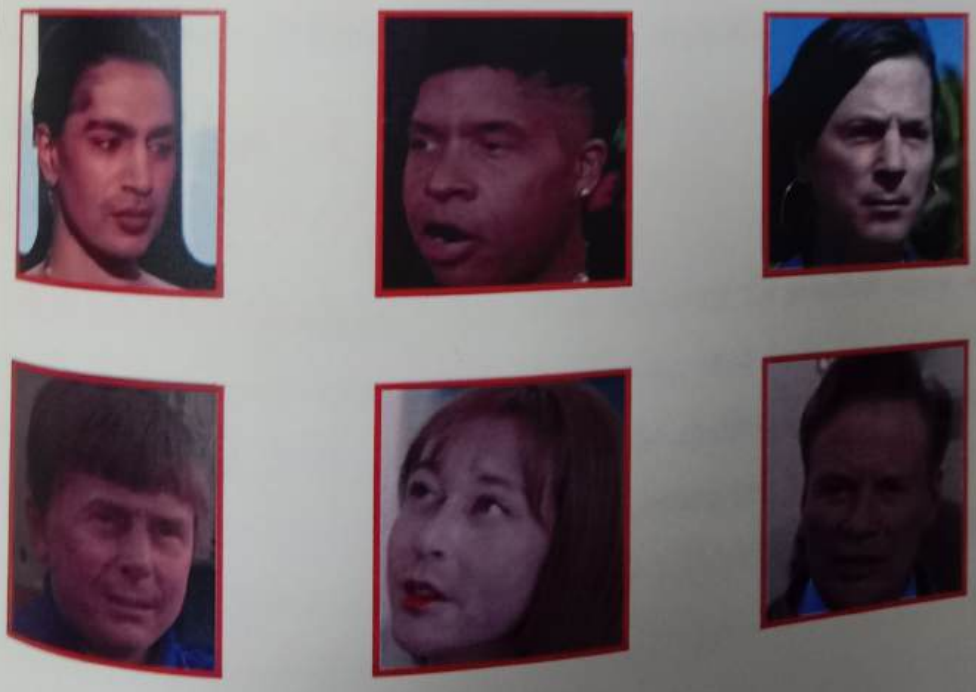

Figure 6.5: Extracted frames from videos classified as fake by MIM-ViT

Chapter 7

Applications

In order to make the proposed model easily accessible by users, two applications are developed, namely a full stack video streaming application (VStream) and a Discord bot (Deepfake Detector). A detailed Software Requirements Specification (SRS) report is given in the following subsections.

7.1 VStream - Video Streaming Application

7.1.1 Introduction

Title Introduction

VStream is a video streaming application designed to provide users with an authentic and secure experience by ensuring that only real videos are uploaded and displayed on the platform. The application allows users to upload and watch videos, with a focus on ensuring the authenticity of the content. VStream utilizes the proposed model MIM-ViT to detect deepfake videos and prevents them from being shared.

Project Domain

Video Streaming, Social Cybersecurity, Community Privacy, Computer Vision, Software Development

Need of Project

The need for a video streaming app with in-built deepfake detection functionality arises from the growing problem of deepfake videos. As the use of deepfake technology becomes more widespread, there is an urgency for solutions that can detect and prevent their spread. VStream addresses this need by providing a secure and trustworthy platform for users to upload and share videos. The application ensures that only authentic videos are shared on the platform, thereby preventing the spread of false or misleading information.

The application's deepfake detection functionality is also important for protecting the privacy and safety of individuals. Deepfakes can be used to manipulate images and videos of people without their consent, and can be used to create harmful content. By preventing the spread of deepfakes, the app helps to protect individuals from these types of abuses.

Objectives

- To provide a secure and trustworthy platform for users to share videos without the risk of deepfake videos being uploaded and shared.
- To use MIM-ViT, the proposed model, to analyze videos and determine if they are authentic or deepfake, ensuring that only real videos are displayed on the app.
- To protect the privacy and safety of individuals by preventing the spread

of deepfakes.

- To establish a user-friendly and seamless user experience, allowing for easy upload and viewing of videos on the app.
- To ensure scalability, allowing the app to handle a large number of concurrent users and videos without any technical issues.
- To raise awareness about the dangers of deepfakes and the importance of authenticity and security in digital media.
- To promote ethical and responsible use of technology, and to minimize the potential harm caused by deepfake videos.

7.1.2 Overall Description

Product Perspective

The product perspective of VStream is that it is a software product designed to meet the needs of users who want to share and view videos online while also protecting against the spread of deepfakes.

From a technical perspective, the app is built using modern software development techniques and architecture. The app is designed to be scalable, allowing it to handle a large number of concurrent users and videos without any performance issues. The deepfake detection functionality is implemented using advanced Masked Image Modelling techniques coupled with a Vision Transformer, which analyze videos for signs of manipulation and determine their authenticity.

From a user perspective, the app is designed to be intuitive and easy to use. Users can easily upload and view videos, with a simple and straightforward interface that makes it easy to navigate and find what they are looking for.

The deepfake detection functionality is integrated seamlessly into the app, providing users with peace of mind that the videos they are watching are authentic and trustworthy.

Product Functions

- **Video Upload**: The app allows users to upload videos to the platform. The upload function includes features such as the ability to add titles, descriptions, and tags to the video.
- **Video Viewing**: The app allows users to view videos uploaded by other users. Users can search for videos, browse categories, and view trending videos.
- **Deepfake Detection**: The app uses MIM-ViT model to analyze videos for signs of deepfakes, such as inconsistencies in facial expressions or voice patterns. If a video is flagged as a deepfake, it cannot be uploaded to the platform.
- **User Profiles**: The app allows users to create profiles, which include information such as their username, profile picture, and a list of videos they have uploaded.
- **Search Functionality**: The app allows users to search for videos using keywords, titles, or tags.

User Classes & Characteristics

- **Individual Users**: These are users who want to share and view videos online. They may be interested in uploading videos of themselves or others, sharing videos with friends and family, or discovering new and interesting videos. Characteristics of individual users may include being tech-savvy, social media users, and having an interest in video content.

- **Business Users**: These are users who want to use the app as a platform to promote their brand, products or services. Characteristics of business users may include being marketing professionals, entrepreneurs, and small business owners.
- **Moderators**: These are users who are responsible for monitoring and reviewing the content uploaded to the app. They are responsible for ensuring that the app's guidelines and policies are followed, and that deepfake videos are detected and removed. Characteristics of moderators may include being detail-oriented, tech-savvy, and experienced in content moderation.
- **Administrators**: These are users who are responsible for managing the app's overall operations, including technical support, customer service, and strategic planning. Characteristics of administrators may include being experienced in project management, software development, and business administration.

Operating **Environment**

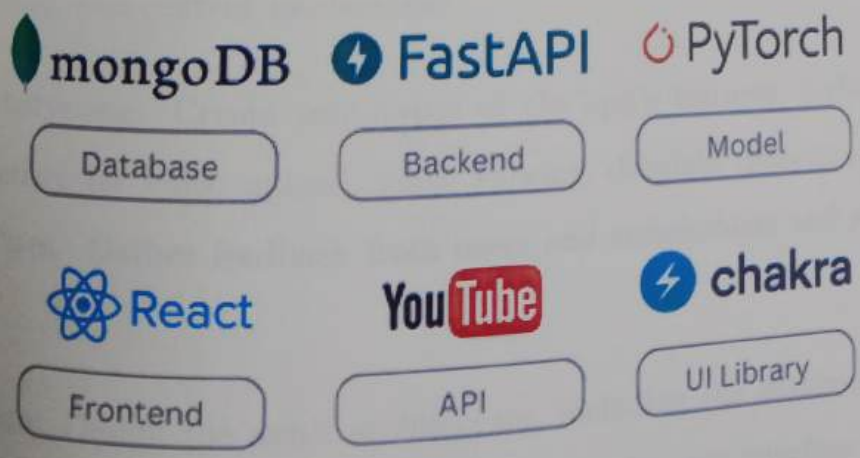

Figure 7.1: Tech stack of video streaming application

Software Development Lifecycle - Rapid Application Development Model

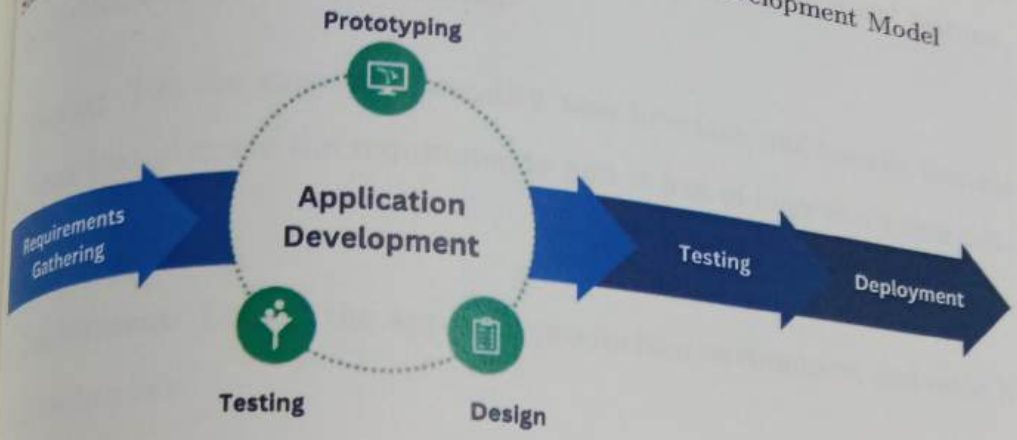

Figure 7.2: Rapid Application Development Lifecycle

During the development process of the application, the Rapid Application Development (RAD Lifecycle Method was used. Figure 7.2 depicts the stages of the RAD model. Following are the steps taken -

1. **Requirements Gathering**: Gather requirements from stakeholders and users, including the need for deepfake detection functionality, social features, user profiles, and content moderation.

2. **Prototyping**: Create prototypes of the app's features, including the user interface for video upload, video viewing, deepfake detection, and social features. Gather feedback from users and stakeholders and refine the design.

3. **Design**: Design the app's architecture, including the database schema and algorithms for deepfake detection. Develop the user interface based on the feedback from the prototyping phase.

4. **Development**: Write code and build the app's features and functionality, including video upload, video viewing, deepfake detection, social features, user profiles, and content moderation.

5. **Testing**: Test the app's functionality, user interface, and security features to ensure that it meets the requirements and is free of bugs and vulnerabilities.

6. **Deployment**: Deploy the app to a production environment and make it available to users.

7.1.4 Screenshots of Working Application

Figure 7.3: Login Page

Figure 7.4: Signup Page

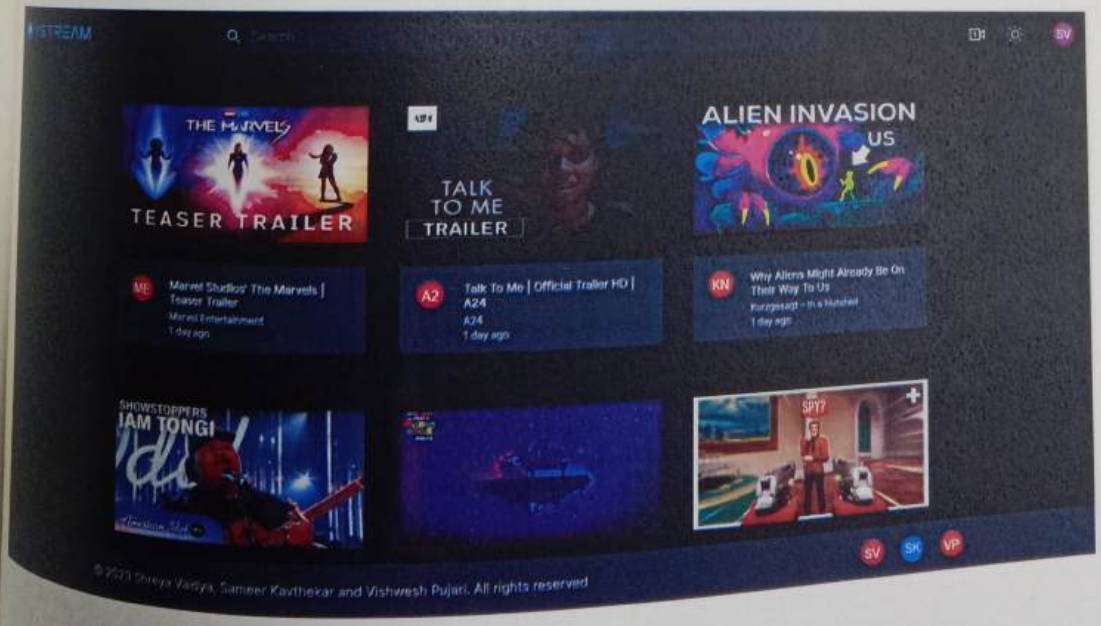

Figure 7.5: Home Page in Dark Mode

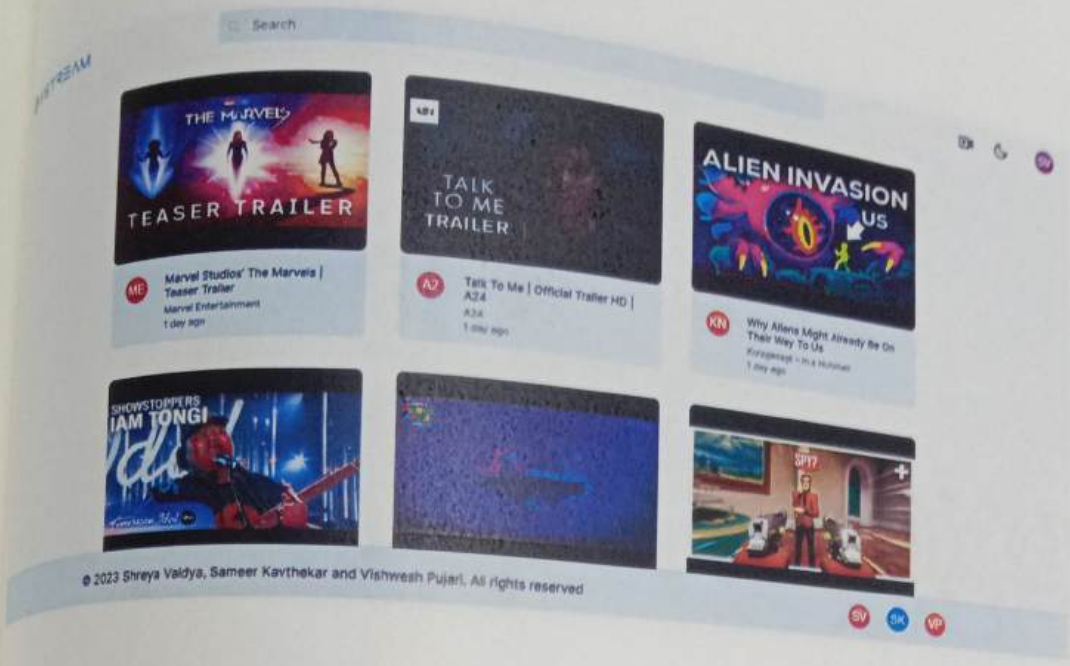

Figure 7.6: Home Page in Light Mode

Figure 7.7: Search Results for user query

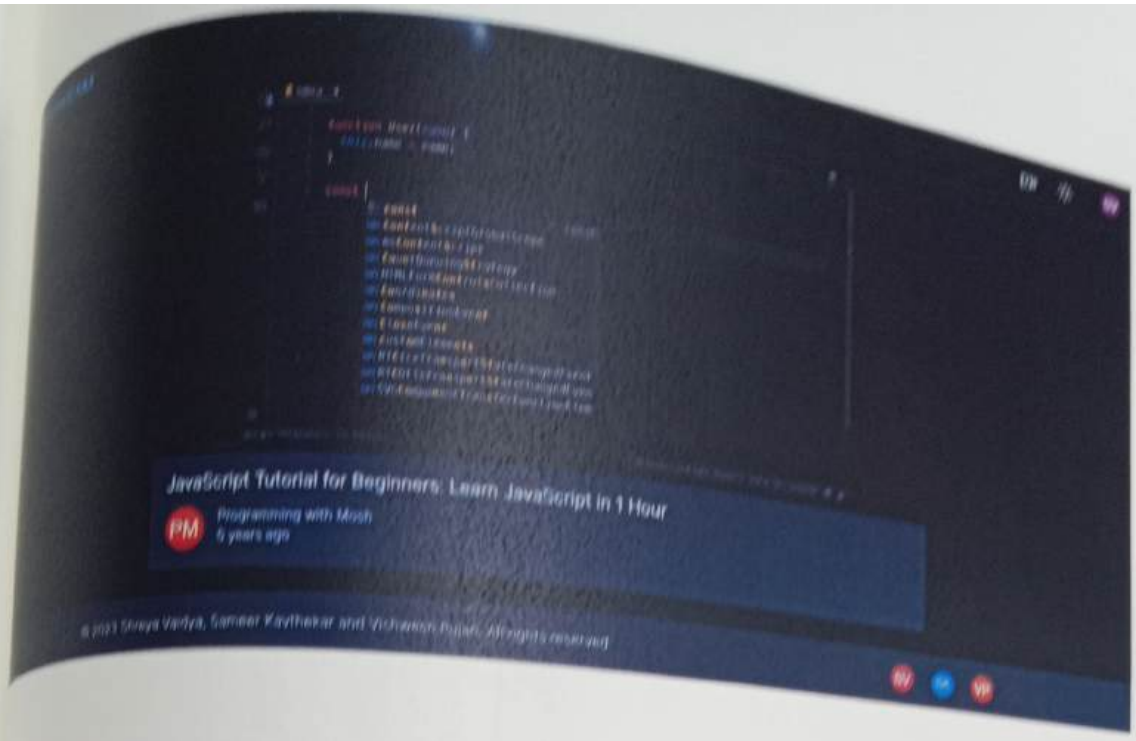

Figure 7.8: Video Player Page

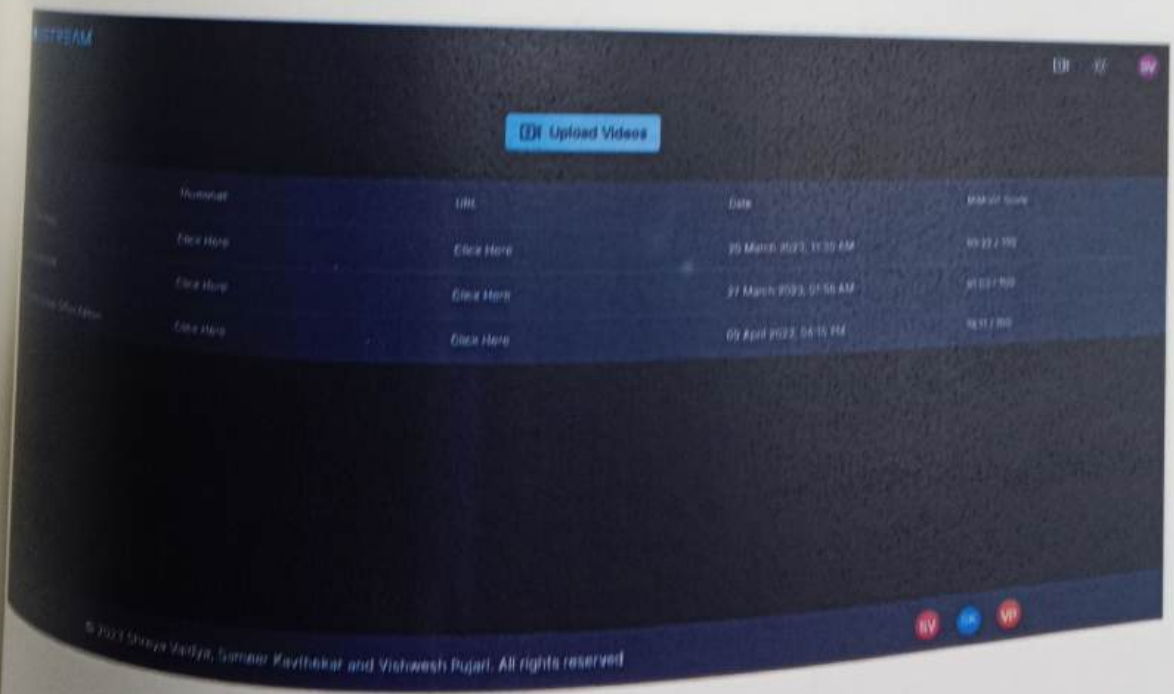

Figure 7.9: Creator Console for Video Uploads

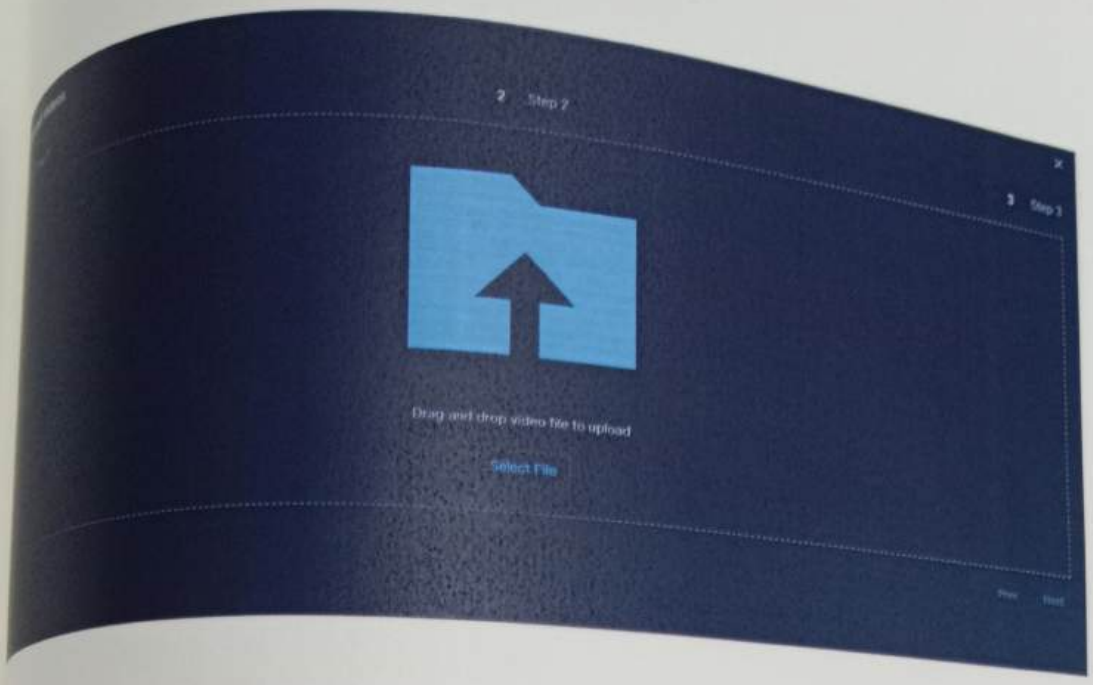

Figure 7.10: Upload Form - Step 1: Upload a file

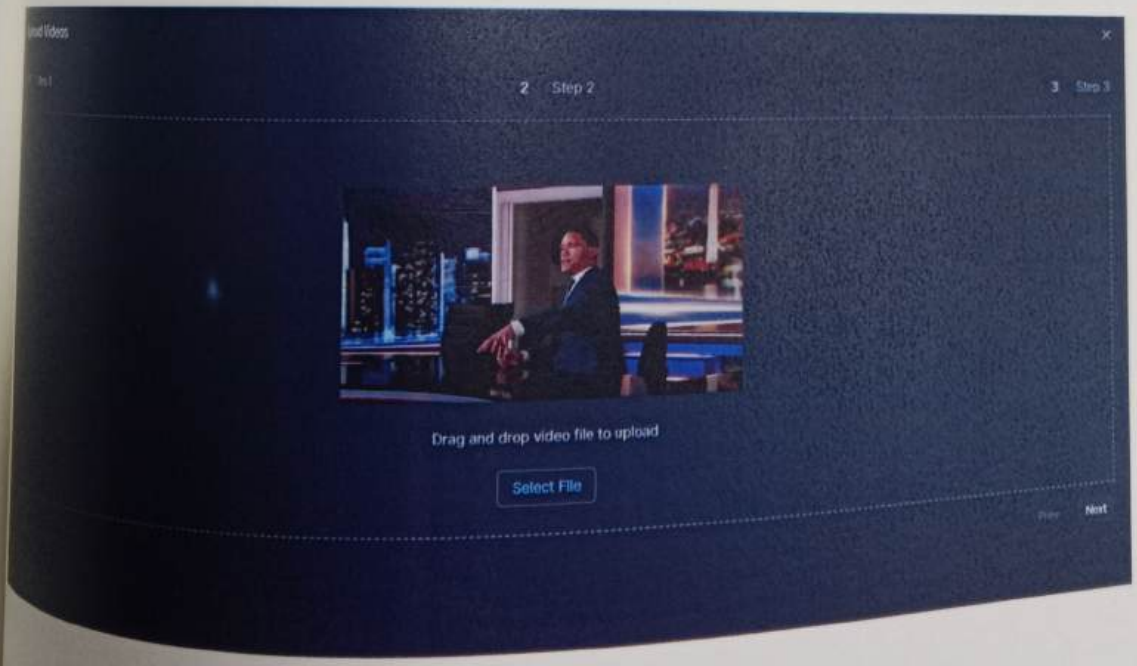

Figure 7.11: Upload Form - Step 1: Uploaded a fake video

Figure 7.12: Upload Form - Step 2: Add video title and description

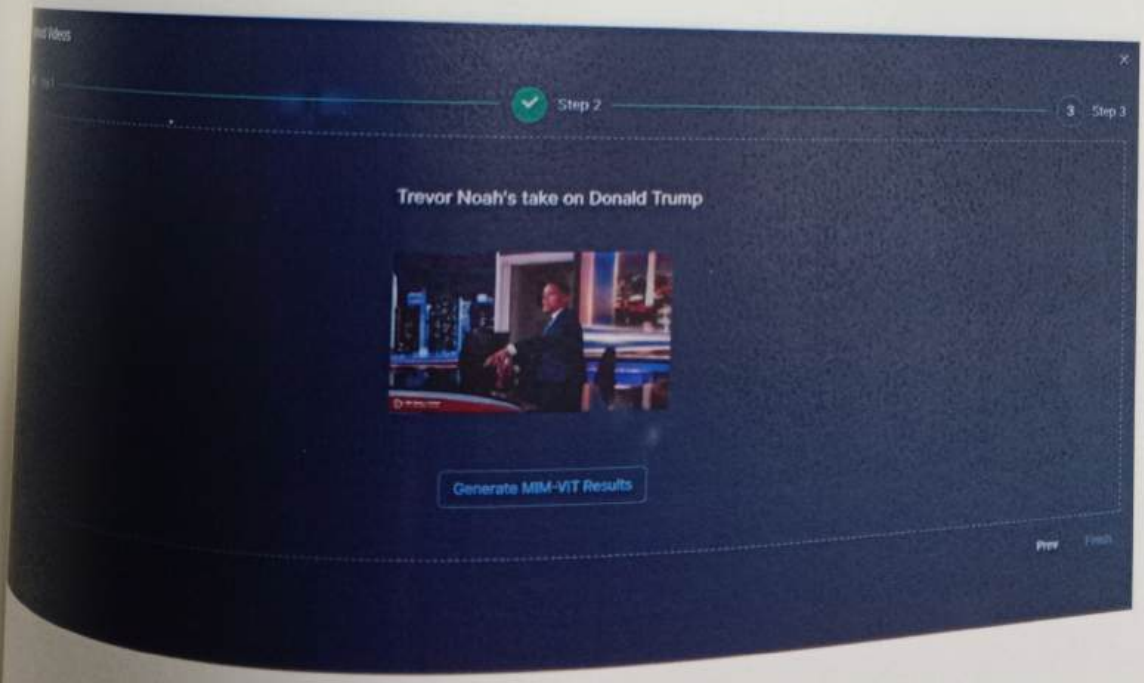

Figure 7.13: Upload Form - Step 3: Run MIM-ViT Deepfake Test

7.2 Deepfake Detector - Discord Bot

The following software requirement specification (SRS) outlines the functional and non-functional requirements of the Discord bot, Deepfake Detector, that detects and flags deepfake videos. The bot is designed to operate within a Discord server and monitor messages containing videos to identify and flag any videos that may contain deepfake content.

7.2.1 Functional Requirements

User Registration and Authentication

The bot should require users to register and authenticate themselves before allowing access to the deepfake detection service.

Monitoring Messages

The bot should monitor messages within the Discord server and identify messages that contain video content.

Deepfake Detection

The bot should use the proposed MIM-ViT model to analyze the video content and identify any evidence of deepfake manipulation.

Flagging Deepfakes

The bot should flag any videos that are identified as potentially containing deepfake content by adding a warning message to the Discord channel.

7.2.2 Non-functional Requirements

Performance

The bot should be able to handle a high volume of messages and provide quick and accurate deepfake detection results.

Security

The bot should ensure the confidentiality and integrity of user data by implementing secure registration and authentication processes.

Compatibility

The bot should be compatible with a variety of Discord server configurations and video formats.

7.2.3 Assumptions and Dependencies

Access to Video Content

The bot requires access to video content within the Discord server in order to perform deepfake detection.

7.2.4 Operating Environment

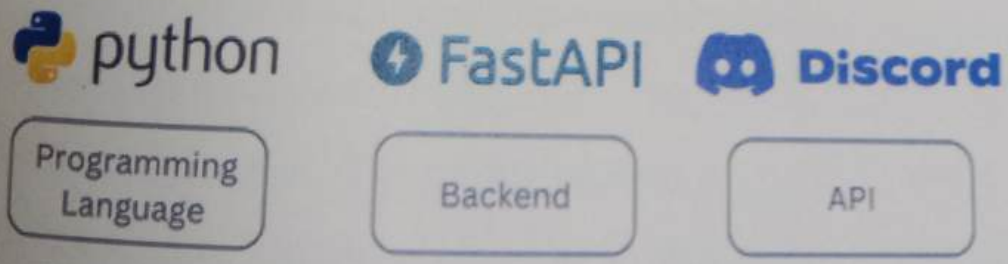

Figure 7.18: Tech stack of Discord bot

7.2.5 Screenshots of Working Bot

Figure 7.19: Deepfake Detector Discord Bot - Fake Video

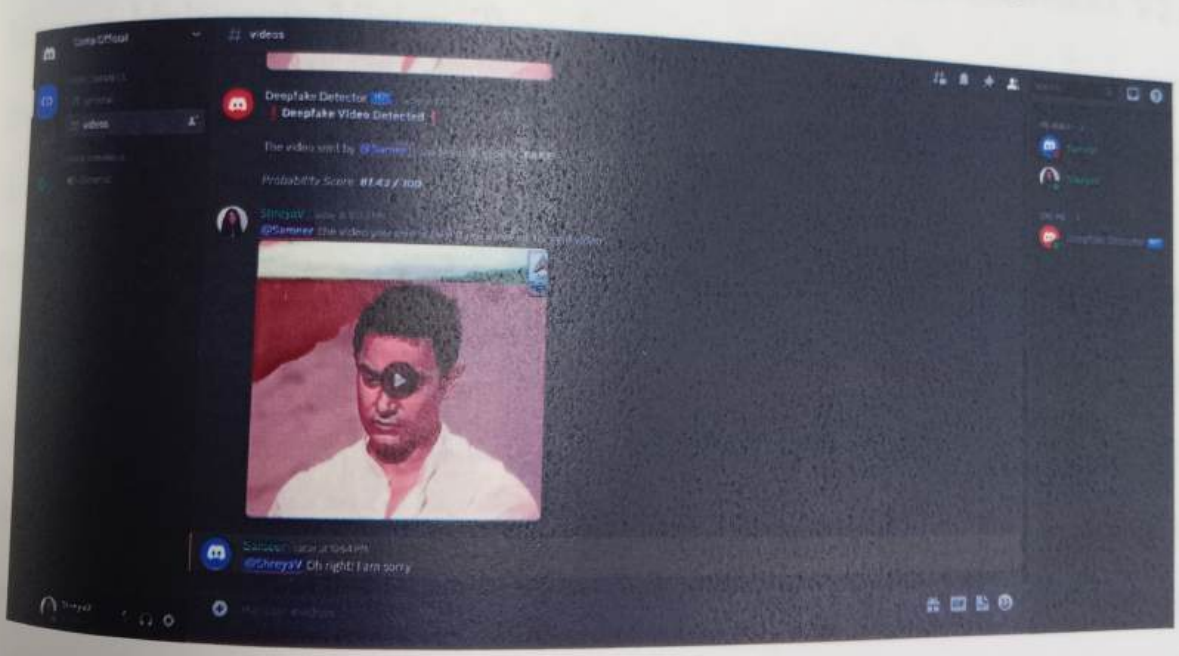

Figure 7.20: Deepfake Detector Discord Bot - Real Video

Chapter 8

Conclusion

This work attempts to solve the problem of deepfake detection in mainstream media by proposing a novel combination of CNN Encoder ConvNeXt V2 and Enhanced Multiscale Vision Transformer. Additionally, a new face quality detection algorithm is proposed, which prevents noisy data from being fed to the model. When trained on the Deepfake Detection Challenge Preview and CelebDF datasets, the model achieves a test accuracy of 80.22% and 92.82%, along with an AUCROC score of 84.48% and 98.82%. The model was trained on the Face Forensics++ dataset and tested on the CelebDF dataset to test the model's cross-dataset and generalization power performance. The model performs competitively by achieving a AUCROC score of 68.21%. The effect of the face quality algorithm is tested and a performance gain of 4% is observed. The preprocessing stage is modified in order to train and test the proposed model for multi-face deepfake detection on the Face Forensics in the Wild (FFIW10K) dataset. The model achieves an AUCROC of 68.71% for the same. In addition, VStream, a full-stack video streaming web application is developed to provide a platform for authentic and moderated content. Finally, Deepfake Detector, a Discord bot is developed to flag deepfake videos sent in a community to prevent spread of false information online.

Bibliography

[1] Korshunova, Iryna, et al. "Fast face-swap using convolutional neural networks." Proceedings of the IEEE international conference on computer vision. 2017.

[2] Li, Lingzhi, et al. "Advancing high fidelity identity swapping for forgery detection." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[3] Kim, Jiseob, Jihoon Lee, and Byoung-Tak Zhang. "Smooth-swap: a simple enhancement for face-swapping with smoothness." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[4] Zhang, Yunxuan, et al. "One-shot face reenactment." arXiv preprint arXiv:1908.03251 (2019).

[5] Güera, David, and Edward J. Delp. "Deepfake video detection using recurrent neural networks." 2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS). IEEE, 2018.

[6] de Lima, Oscar, et al. "Deepfake detection using spatiotemporal convolutional networks." arXiv preprint arXiv:2006.14749 (2020).

[7] Chang, Xu, et al. "Deepfake Face Image Detection based on Improved VGG Convolutional Neural Network." 2020 39th Chinese Control Conference (CCC). IEEE, 2020.

[8] Zhao, Hanqing, et al. "Multi-attentional deepfake detection." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021.

[9] Wodajo, Deressa, and Solomon Atnafu. "Deepfake video detection using convolutional vision transformer." arXiv preprint arXiv:2102.11126 (2021).

[10] Coccomini, Davide Alessandro, et al. "Combining efficientnet and vision transformers for video deepfake detection." Image Analysis and Processing–ICIAP 2022: 21st International Conference, Lecce, Italy, May 23-27, 2022, Proceedings, Part III. Cham: Springer International Publishing, 2022.

[11] Ganguly, Shreyan, et al. "ViXNet: Vision Transformer with Xception Network for deepfakes based video and image forgery detection." Expert Systems with Applications 210 (2022): 118423.

[12] Wang, Junke, et al. "M2tr: Multi-modal multi-scale transformers for deepfake detection." Proceedings of the 2022 International Conference on Multimedia Retrieval. 2022.

[13] Dolhansky, Brian, et al. "The Deepfake Detection Challenge (DFDC) Preview Dataset." arXiv preprint arXiv:1910.08854 (2019).

[14] Rössler, Andreas, et al. "FaceForensics: A large-scale video dataset for forgery detection in human faces." arXiv preprint arXiv:1803.09179 (2018).

[15] Li, Yuezun, et al. "Celeb-DF: A large-scale challenging dataset for deepfake forensics." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.

[16] Zhou, Tianfei, et al. "Face forensics in the wild." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[17] Deng, Jiankang, et al. "RetinaFace: Single-shot multi-level face localisation in the wild." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[18] Wojke, Nicolai, Alex Bewley, and Dietrich Paulus. "Simple online and realtime tracking with a deep association metric." 2017 IEEE international conference on image processing (ICIP). IEEE, 2017.

[19] Wang, Zhou, Eero P. Simoncelli, and Alan C. Bovik. "Multiscale structural similarity for image quality assessment." The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. Vol. 2. IEEE, 2003.

[20] Woo, Sanghyun, et al. "ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders." arXiv preprint arXiv:2301.00808 (2023).

[21] Li, Yanghao, et al. "MViTv2: Improved Multiscale Vision Transformers for Classification and Detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[22] Wightman, Ross. "PyTorch Image Models." GitHub, Ross Wightman, 2019, https://github.com/rwightman/pytorch-image-models.

[23] Fan, Haoqi, et al. "Multiscale vision transformers." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[24] Liu, Zhuang, et al. "A ConvNet for the 2020s." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[16] Zhou, Tianfei, et al. "Face forensics in the wild." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[17] Deng, Jiankang, et al. "RetinaFace: Single-shot multi-level face localisation in the wild." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[18] Wojke, Nicolai, Alex Bewley, and Dietrich Paulus. "Simple online and realtime tracking with a deep association metric." 2017 IEEE international conference on image processing (ICIP). IEEE, 2017.

[19] Wang, Zhou, Eero P. Simoncelli, and Alan C. Bovik. "Multiscale structural similarity for image quality assessment." The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. Vol. 2. IEEE, 2003.

[20] Woo, Sanghyun, et al. "ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders." arXiv preprint arXiv:2301.00808 (2023).

[21] Li, Yanghao, et al. "MViTv2: Improved Multiscale Vision Transformers for Classification and Detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[22] Wightman, Ross. "PyTorch Image Models." GitHub, Ross Wightman, 2019, https://github.com/rwightman/pytorch-image-models.

[23] Fan, Haoqi, et al. "Multiscale vision transformers." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[24] Liu, Zhuang, et al. "A ConvNet for the 2020s." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[25] He, Kaiming, et al. "Masked autoencoders are scalable vision learners." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[26] Li, Yuezun, and Siwei Lyu. "Dsp-fwa: Dual spatial pyramid for exposing face warp artifacts in deepfake videos." Retrieved December 18 (2019): 2019.

[27] Afchar, Darius, et al. "Mesonet: a compact facial video forgery detection network." 2018 IEEE international workshop on information forensics and security (WIFS). IEEE, 2018.

[28] Nguyen, Huy H., et al. "Multi-task learning for detecting and segmenting manipulated facial images and videos." 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS). IEEE, 2019.

[29] Nguyen, Huy H., Junichi Yamagishi, and Isao Echizen. "Capsule-forensics: Using capsule networks to detect forged images and videos." ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.

[30] Liu, Honggu, et al. "Spatial-phase shallow learning: rethinking face forgery detection in frequency domain." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[31] Sun, Ke, et al. "Domain general face forgery detection by learning to weight." Proceedings of the AAAI conference on artificial intelligence. Vol. 35. No. 3. 2021.

[32] Masi, Iacopo, et al. "Two-branch recurrent network for isolating deepfakes in videos." Computer Vision–ECCV 2020: 16th European Con-

ference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16. Springer International Publishing, 2020.

[33] Qian, Yuyang, et al. "Thinking in frequency: Face forgery detection by mining frequency-aware clues." Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII. Cham: Springer International Publishing, 2020.

[34] Hu, Juan, et al. "Finfer: Frame inference-based deepfake detection for high-visual-quality videos." Proceedings of the AAAI conference on artificial intelligence. Vol. 36. No. 1. 2022.

Appendix B

Publication Details

The research paper titled, "MIM-ViT: Deepfake Detection using Masked Image Modelling and Vision Transformer", based on this work is **accepted** by the *12th International Conference on Soft Computing for Problem Solving* hosted by **IIT Roorkee** with possible publication in the Scopus Indexed Series: Lecture Notes in Networks and Systems by **Springer**.