National College of Ireland

Unmasking Deception: Deepfake detection using Shallow CNN

MSc Research Project
Data Analytics

Ashwyn Singh
Student ID: x22128310

School of Computing
National College of Ireland

Supervisor: Christian Horn

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ashwyn Singh
Student ID:	x22128310
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Christian Horn
Submission Due Date:	14/12/2023
Project Title:	Unmasking Deception: Deepfake detection using Shallow CNN
Word Count:	6352
Page Count:	17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	31st January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Unmasking Deception: Deepfake detection using Shallow CNN

Ashwyn Singh
x22128310

Abstract

Deepfakes are made using machine learning algorithms to create picture alteration and face swapping to present individuals speaking or doing things that never occurred. GANs have been used to generate fake pictures and videos because of its multipurpose approach. It utilizes "landmark" points on the face to create such realistic fake videos which appear real to the naked eye until looked very carefully. Examples of such characteristics include the edges of an individuals eyes and lips, nostrils, and the contour of the jawline. These new advancements can negatively affect society, breaching privacy and security of individuals and organizations alike. The study utilizes shallow CNN along with post processing to identify deepfake videos efficiently. As deep learning is a field that is constantly developing, researching new techniques to detect these kind of videos is a must. According to the study done in this paper, the model performs well on an untrained dataset, with an accuracy of 86% percent and a logloss of 0.4084.

1 Introduction

The advancement of technology in recent years, especially in the domain of machine learning and artificial intelligence has been staggering. These technologies are becoming integral parts of our daily lives, particularly in social media and content sharing applications. It is important now to safeguard and protect data integrity in these applications, given the increasing trend of fake and manipulated media. One significant issue of manipulated media which causes a major challenge is the use of deepfakes in manipulating visual and audio media. Deepfakes are made by employing powerful deep learning algorithms to make very realistic and often convincing fake images, videos, or audio. Deepfakes were first developed due to the advancement of GANs (Generative Adversarial Networks), which was developed in a study done by Goodfellow et al. (2020). The word "Deepfake" originates from the phrases "deep learning" and "fake". Most of the deepfakes that are created are done by substituting faces in video and or photos to create realistic-looking looking fakes. However, there are a lot of positive use cases of creating content like that. For example, deepfakes have proven useful in the entertainment business, such as making films starring Will Smith in 2019 picture "Gemini Man" where his 20 year old version was created using deepfake technology. They also help in enhancing the quality of film dubbing for international audiences. They are an integral part of creating historical reenactments in museums and schools, making history more fascinating for enthusiasts. Deepfakes are also now being tried to enhance virtual shopping

experience for the customers. However in 2017, a Reddit user named 'Deepfake' superimposed pictures of celebrities onto pornographic videos, drawing the world's attention to the potential misuse of this technology. After that. there was a rise in the utilization of more open-source and accessible software and algorithms for their creation. Since deepfakes pose a significant threat to privacy and the spread of misinformation, social media companies and researchers are consistently working together to devise methods for their detection.

1.1 Background

There have been numerous instances since 2017 where deepfake technology has been utilized for malicious purposes. As discussed by Pawelec (2022) in their study, that the misuse of fake audiovisual media to promote misinformation and hate speech poses a grave risk to core democratic norms. In March 2022, a cyber attack hit a Ukrainian news agency's website, where a deepfake video of President Zelenskyy[1] encouraging Ukrainians to surrender was purposefully uploaded. Another study done by Wang et al. (2022) validated the problems and its uses in making falsified photos of scientific nature utilized in publications, which is another concerning example of deepfake technology's potential danger. GANs (Generative Adversarial Networks) were employed to create images of the western blot as well as more complicated ones relating to esophageal cancer. A panel comprising three highly qualified biomedical professionals was then presented with these false images. Surprisingly, two of the three professionals only detected the fake photos with 10% and 30% accuracy, respectively. The third specialist performed better, attaining 60% accuracy by distinguishing between actual and false blot pictures, indicating the importance and difficulty of the situation.

1.2 Importance

Detecting deepfakes has become an urgent necessity owing to their escalating sophistication and how easily they can be created using open source software. An example can be seen on the subreddit page[2] where people share their created deepfakes. Popular celebrities are specially at a very high risk. This is because the process for creating any deepfake using GANs requires a lot of training data. Since a lot of pictures of celebrities are available from almost every angle and in any light environment, creating their deepfakes is quite easy. Also using deepfakes to defame or abuse celebrities for financial gain is a powerful motivator for making such material. As for political figures, Deepfakes of them can sow seeds of misinformation, eroding trust in authentic media sources. They might manipulate public opinion, leading to widespread confusion and destabilizing the democratic process. According to the research done by Verdoliva (2020) , with the rapid progress of deepfake technology and the existence of skilled perpetrators, no method or technology will offer permanent protection, requiring a constant demand for new solutions to deal with unforeseen dangers.

[1] `https://mixed.de/selenskyj-deepfake-miserabel-und-dennoch-historisch/`

[2] `https://www.reddit.com/r/SFWdeepfakes/`

1.3 Research Question

RQ: How does the integration of post-processing techniques in conjunction with a shallow Convolutional Neural Network (CNN) like Mesonet impact the accuracy and reliability of deepfake detection?

Sub RQ: How do alterations in both model training parameters (including learning rate, batch size, and network architecture) and post-processing parameters (such as prediction value cutoff threshold and the quantity of images per class for video classification) collectively influence the performance and accuracy of a shallow Convolutional Neural Network (CNN) like Mesonet in detecting manipulated or deepfake videos?

1.4 Limitations

The dataset on Kaggle is divided into 50 subsections due to it's huge size, with each file containing approximately 2500-3000 videos. Dolhansky et al. (2020) offer a concise overview of the Deepfakes Detection Challenge (DFDC) dataset. It has 119,154 10 second videos incorporating four facial modification techniques to create Deepfakes. Since the whole dataset is of 471GB, processing all videos would be a very computationally demanding and time consuming task. The creation of dataset for the project discussed in this paper in itself took eight to ten hours for each sub-folder. Also, the model itself took six to eight hours to train on these many images. Because of these limitations, only the preview dataset and the first seven datasets were used in the project. The total number of videos processed in this project is 3175 videos, from which 32894 images were generated. Also, the logloss which was used to decide the winner in the competition was calculated on a private dataset which is now not available. So the logloss and other evaluation metrics for this project were calculated on the test set, which was created using the train_test_split library of scikitlearn.

The sections that follow this text discuss a variety of topics. Section 2 assesses existing Deepfake detection techniques. Sections 3 and 4 detail the strategies and methodology, which includes dataset collecting, pre-processing strategies, and machine learning models implemented. Sections 5 and 6 discuss how the project was executed, the methods used for evaluation, and the outcomes achieved. Finally, the final section puts this study project to a close by providing a general overview while suggesting prospective possibilities for further investigation.

2 Related Work

2.1 Detecting Deepfakes by detecting biological signals

Identifying false images using unusual traits like misaligned noses and eyes is an area of study that received close attention since the introduction of tools such as Adobe Photoshop. But because technology continues to develop to create fake images and videos with very small shortcomings especially in the eyes, nose and lips area so the methods for detecting them have evolved as well and utilize increasingly complicated biological signals. One example is the approach used by Li et al. (2018) for identifying AI-generated fake videos by monitoring eye blinking patterns. The authors detected these fraudulent videos by exploiting humans involuntary and subtle eye blinking activity, which is frequently difficult for AI algorithms to accurately mimic. The method consists of two

major phases, first being that the video undergoes pre-processing to precisely extract face regions to identify eye positions. The photos are subsequently fed into the LRCN (Long-term Recurrent Convolutional Networks) model to be trained. The LRCN model is composed of three parts: feature extraction, sequence learning, and state prediction. This approach, however, has a drawback. The identification approach primarily relied on recognizing the individual's absence of blinking in the AI-generated videos. Therefore, it may not be useful against capable forgers who are able to create deepfakes with convincing blinking patterns utilizing models that are more complicated. As a result, the suggested strategy might also be unable to identify forgeries in which blinking is too rapid or too frequent, which is biologically impossible.

Yu et al. (2019) in their study discussed about small variation in the skin color of subjects which is recurring in nature. This phenomena occurs due to the blood pumping through the faces of the subjects in the videos. Qi et al. (2020) used this finding to get some insights on the difference between real and fake videos. The heartbeat rhythms were totally different for the real face videos as compared to the fake videos. This difference in heartbeat rhythms was then used to develop a technique to detect real videos from fake videos using motion-magnified spatial-temporal representation (MMSTR). This approach helped to obtain an accuracy of 64% on the DFDC preview dataset.

Yang et al. (2019) devised a methodology that exploited the mistakes that occur when an individual's face has mismatched facial landmarks to detect fake images. These are the areas on human faces that have essential features such as eye and mouth points. As a consequence, fake photos with irregular 3D head positions between central and whole face landmarks deviate greatly from actual ones. The results demonstrated that cosine distances between head orientation vectors are less and concentrate in a narrower range (up to 0.02) for actual photos, but for deepfakes, the bulk of values are between 0.02 and 0.08. This disparity in distribution was then utilized to distinguish deepfakes from genuine photos. The proposed method detects fake photos with major modifications with great precision but falls short in recognizing fake images with minor distortions which are now viable to construct with sophisticated GANs.

2.2 Detecting Deepfakes using irregularities between consecutive video frames

Researchers have delved in detection procedures that look for changes in optical flow or the presence of artifacts in between frames. Amerini et al. (2019) proposed utilizing optical flow fields to identify deepfakes from real videos. In principle, optical flow can disclose motion differences between synthetically generated frames and naturally generated frames taken by a video camera in face features such as the lips and eyes. The PWC-Net CNN model for optical flow was used to compute the forward flow between consecutive frames. The calculated flow was then sent into Flow-CNN, a semi-trainable CNN based on pre-trained networks such as VGG16 by Simonyan and Zisserman (2014) and ResNet50 by He et al. (2016). On the Faceforensics++ dataset, the model performed admirably. Although the model is straightforward to construct and has a large false positive rate which was discovered in the research done byAlnaim et al. (2023).

The approach presented by Saikia et al. (2022) entails taking frames from the video, capturing the face from each one, producing optical flow fields across subsequent frames, and modeling time data with a hybrid CNN-RNN architecture. The optical flow fields give insight into movement patterns, and the LSTM layers aid in capturing temporal

dependencies between frames for increased classification accuracy. Several cutting-edge pre-trained CNN models were investigated. For fine-tuning the models on the deepfake dataset, the final completely connected layers related to the job (dense layers) were eliminated. At the conclusion of the design, a softmax layer is introduced to compute the odds of the frame sequence being false or real.

2.3 Detecting Deepfakes using supervised Deep Learning

In 2017, advanced applications like Fakeapp and Face2Face emerged, employing auto encoders to generate convincing deepfake content. Addressing the challenge of detection, Afchar et al. (2018) introduced two cutting-edge methods for identifying deepfakes: Meso-4 and Mesoinception-4. The study focused on a mesoscopic analysis level, a departure from microscopic image analyses, as the latter proves impractical in compressed videos due to substantial image noise degradation. In Meso-4, the researchers utilized four layers of successive convolutions and pooling, followed by a dense network featuring one hidden layer, totaling 27,977 trainable parameters. Mesoinception-4 innovatively replaced the initial two convolutional layers of Meso-4 with an inception module variant. This module integrates outputs from convolutional layers with diverse kernel shapes, expanding the function space for optimization. The network comprises 28,615 trainable parameters. Notably, these approaches exhibit efficiency in terms of computational power, as they downsample video frames and employ shallow CNNs for training. After that, the model was put to the test on photos that had been altered with Face2Face and Fakeapp, and it did well on both datasets. But the accuracy decreases, particularly for low-quality videos. Salman and Shamsi (2023) evaluation of several deepfake detection methods put this to the test.

Researchers currently choose to use transfer learning to detect false images or videos since it makes it easier to create deepfake detection systems that are accurate and efficient by utilising pre-trained neural network models on huge datasets. Three cutting-edge 3D CNNs (I3D, 3D ResNet, and 3D ResNeXt) were employed by Wang and Dantcheva (2020) for this purpose. The Faceforensics++ dataset is used to further hone these networks once they have been pre-trained on the Kinetics-400 human activity dataset. The prediction layer in every network is swapped out for a single neuron layer in order to tackle the binary classification challenge. Videos involving (A) all manipulation techniques, (B) just one manipulation technique, and (C) cross-manipulation techniques to make fake videos were identified using the following methodology. These three categories made it easier to compare how well 3D CNNs performed versus image-based forgery detection systems, including XceptionNet developed by Rössler et al. (2019). The video-based algorithms performed on par with XceptionNet for both the first and second types. The third type involves training 3D CNNs with three different manipulation techniques in addition to the original one, and testing them on the remaining technique. Although the methodology is made simpler by using pre-trained models, the method's inability to detect deepfake approaches for which it was not specially trained is a downside.

At the University of California, Agarwal et al. (2020) achieved deepfake detection by using CNN to detect pixel-by-pixel differences in photos that resulted from face warping. A Siamese network was trained to detect differences in camera data, including ISO, focal length, aperture size, and so forth, in order to carry out the research. The primary objective of the first network is to detect any alterations made to the face, whilst the secondary network looks for subtle characteristics that signify the consistency of the

image. The outputs of these two networks are combined to predict the result. After testing the system on multiple data sets, they achieved an accuracy of 82.4% for the Deepfake Detection Challenge (DFDC) data set.

3 Methodology

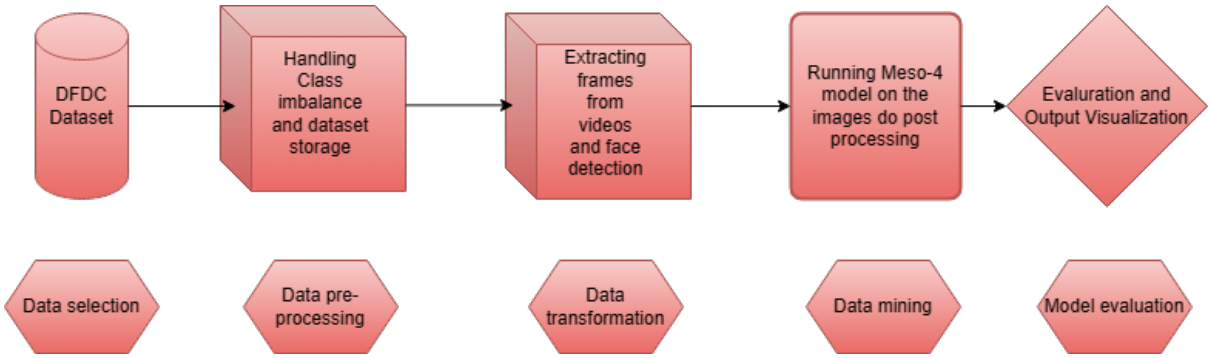

Figure 1: Methodology

Because this project is based on data mining and data science, Knowledge Discovery in Databases (KDD) was chosen as the approach for it. It is chosen mainly because it is not required in KDD that the project should be finished so it is appropriate for this project's particular requirements. Figure 1 provides a complete overview of the techniques important to the KDD method.

3.1 Data Selection

This project uses the Deepfake Detection Challenge (DFDC) dataset, accessible via Kaggle. AWS, Facebook, Microsoft, the Partnership on AI's Media Integrity Steering Committee, and academics initiated the Deepfake Detection Challenge (DFDC). Due to its massive size, the dataset on Kaggle is separated into 50 subsections, with each component comprising around 2500-3000 films. The Deepfakes Detection Challenge (DFDC) dataset is summarized by Dolhansky et al. (2020). Deepfakes are created using 119,154 10 second videos that use eight facial alteration techniques. The actors involved have consented to the utilization and manipulation of their images and videos for the dataset's development. This global initiative aims to inspire researchers worldwide to develop innovative technologies for identifying deepfakes and manipulated content. To form this dataset, 486 subjects were involved and were divided into two distinct groups—training and testing—to prevent facial swaps across sets. These videos have been taken from various angle with respect to the subject as well as have a wide variety of lightning conditions. A notable difference of this dataset from others is the active involvement of actors in its construction, involving the utilization and alteration of their appearances.

Due to the immense size of this dataset and the constraints of the machine utilized to implement the model for this project, only the first 7 attachments of the dataset were employed. This resulted in a total 3175 videos available for the project.

3.2 Data Pre-processing

Total number of videos are assessed and also the class imbalance issue is looked into as in the training dataset, approximately 83% of the videos are fake. To tackle this issue, the number of fake videos picked for further processing was the same as the number of real ones. The fake videos that were picked were chosen at random to avoid biased selection of videos which may result in a loss of information. Also, disk space is created to keep the frames from videos and to keep the information about whether the video is fake or real is kept for these images.

3.3 Data Transformation

In this project the individual frames are extracted after every 10 frames from videos. This step helps reduce the duplication of images as well as helps in minimizing the computational costs. Once these frames are extracted, an advanced MTCNN (Multi-Task Cascaded Convolutional Neural Network) face detection algorithm is employed, and the resulting images of the faces are saved in specific dimensions for the model to operate on.

3.4 Model Evaluation

The winner from the Deepfake detection challenge was decided through the score of Log loss. It measures the accuracy of a model by penalizing incorrect predictions and lower the Log Loss values indicate better performance. The formula for Log Loss is given as:

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Where:

n is the number of videos being predicted.

$\hat{y}_i$ is the predicted probability of the video being fake.

y_i is 1 if the video is fake and 0 if it's real.

log() represents the natural logarithm.

Log Loss heavily penalizes confident wrong predictions. This means that predicting with high confidence (close to 0 or 1) when the prediction is incorrect results in a significantly higher error contribution.

For deepfake detection, precision, recall, and accuracy are also essential metrics:-

Precision: Precision measures the accuracy of the positive predictions. It calculates the ratio of correctly predicted positive observations to the total predicted positives (both true positives and false positives). In deepfake detection, high precision indicates fewer real videos misclassified as fake.

Recall (Sensitivity): Recall calculates the ratio of correctly predicted positive observations to the actual positives in the dataset (true positives and false negatives). In this context, high recall means fewer fake videos are missed or falsely classified as real.

Accuracy: Accuracy measures the overall correctness of the model and is the ratio of correctly predicted observations to the total observations.

4 Design Specification

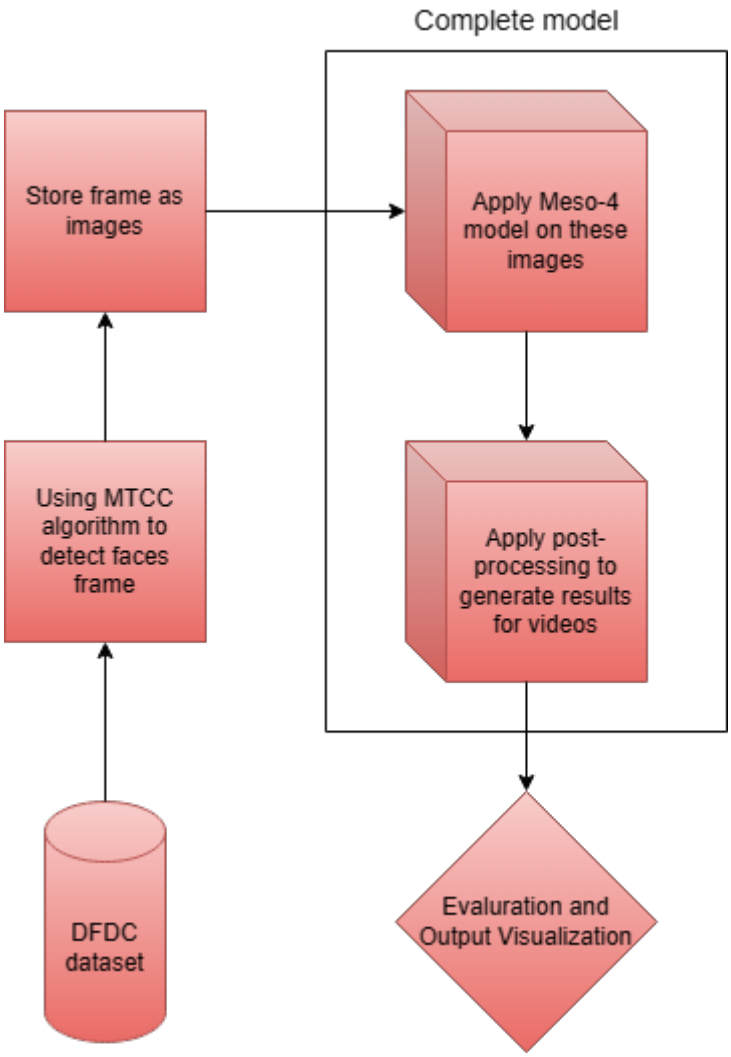

Figure 2: Design Overview

1. The initial task is to assess metadata files within each sub-folder of the dataset and to identify the distribution of real and fake videos. The next step is to take care of any class imbalance problem if it exists by choosing an equal amount of real and false videos to train the model on.

2. The chosen videos undergo processing to retrieve frames from each video. A face detection algorithm is next employed on each frame to detect faces. After that, the identified faces are taken out and saved as separate images on the local disk, with a standard resolution for model training. For purposes of ensuring traceability and ease of future referencing, these photos are labeled with names derived from the video's grouping (false or real) and the video's distinct identifier.

3. Following the face extraction procedure, the obtained images serve as data for training for a shallow CNN neural network architecture. The information that is extracted from the pictures is saved for additional examination and post processing

4. Finally the predictions produced from the model for each frame attributing to different videos are aggregated to produce final predictions for each of the videos. The model's accuracy in making predictions to assess the genuineness of the videos is then assessed using evaluation metrics, giving for an in-depth evaluation of the model's performance.

5 Implementation

This part thoroughly discusses the EDA process, processing of frames, model implementation as well post processing done in the project. The project is implemented in python and the environment used to run python is Jupyter notebook. The code for this project itself is divided into two parts, first being the dataset generation and second being the model implementation and model evaluation.

5.1 Preliminary EDA and Pre-Processing

The dataset available on Kaggle for DFDC challenge presents a significant volume of data, partitioned into 50 distinct sub-parts, each approximately 9GB in size. Each of these sub-parts or folders has a collection of about 3000 videos along with an associated metadata file. The first step of the project involves the analysis of metadata files within each folder. The metadata JSON file is accessed using JSON library and is utilized to count the number of real and fake videos in that sub-part. Since it was observed in each of the metadata files that the count of fake videos was significantly higher than real videos, the number of fake videos chosen for further processing was the same as the number of real ones. The decision to maintain a balance between real and fake video counts is essential for training models and avoiding the class imbalance problem. Given that the dataset comprises deepfake videos generated using eight diverse techniques, a deliberate choice is made to randomly select fake videos. This approach is intended to encompass a variety of deepfake creation methods without incurring any information loss that might arise from a biased selection process.

5.2 Dataset generation

5.2.1 Frame extraction

The exploration and frame extraction from videos was done using OpenCV. Every video in the selected dataset was parsed through to divide them into individual frames. The frames were converted into numpy arrays before they were utilized to extract faces using the MTCNN approach. A select number of frames were selected from each video, considering the fact that they contained 300 frames, for a pair of primary reasons: constraining the cost of processing and to prevent image replication. The choice to extract every 25th frame from each video was made to address the extended computation time needed for image extraction and running the MTCNN algorithm on each detected face.

5.2.2 Face Detection and Resizing

The code was developed in python to process video files, detect faces within the frames using MTCNN model, and extract faces by cropping them based on bounding box coordinates. The objective is to generate and save cropped faces from each frame of the

Figure 3: Data exploration for real and fake frames of a video

video into separate image files. The code iterates through each frame of the video file, reads the frame using OpenCV, and converts the color space to RGB. Then the MTCNN face detection algorithm is used to identify faces within the frame. For each detected face, if the confidence level exceeds 0.98, it extracts the bounding box coordinates, pads the face region based on the provided padding value, crops the face region from the image, resizes it to a specified size (224x224), and saves the resulting cropped face as a PNG image file in the specified output path. Since all the subjects in the videos were not equidistant from the camera, some of the faces of the subjects got distorted after resizing the image to the specific size. This happened because the MTCNN algorithm drew a bounding box around the face which was dependent of the pixels the face was covering on the image. If the subject is far from the camera then the pixel area which the MTCNN algorithm detected was small and resizing the images to 224x224 caused the images to be distorted. Also, the bounding boxes which are created with MTCNN algorithm are rectangular in shape due to the shape of natural shapes of human faces. Converting these images to 224x224 resolution will also result in a small amount of distortion on the horizontal section of the face.

5.3 Pre-processing before Model implementation

The information about the images generated using the previous process is first stored in a list that contains the video names and their corresponding labels, 'REAL' or 'FAKE'. Since each video comprises approximately 10-12 images and the final prediction is based on entire videos rather than individual images, a train-test-validation split is conducted using 'train_test_split' from scikit-learn, based on unique video names. This split produces three sets, 'Train_set', 'Test_set', and 'Val_set', divided in a ratio of 72% for training, 8%

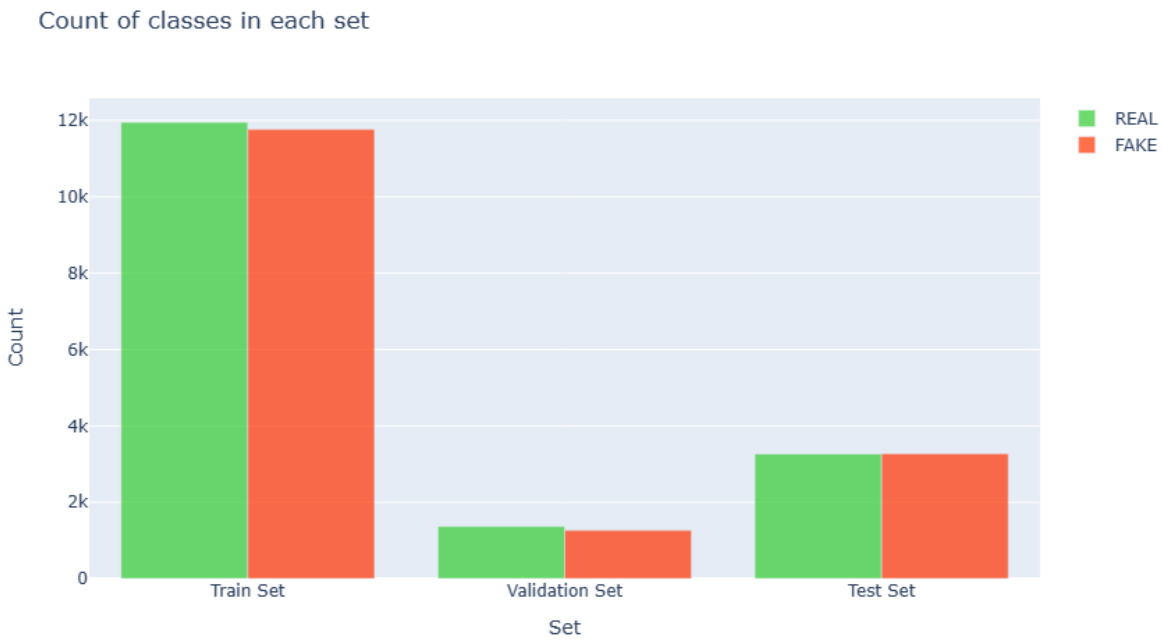

Figure 4: Number of images of each class after preliminary pre-processing

for validation, and 20% for testing.

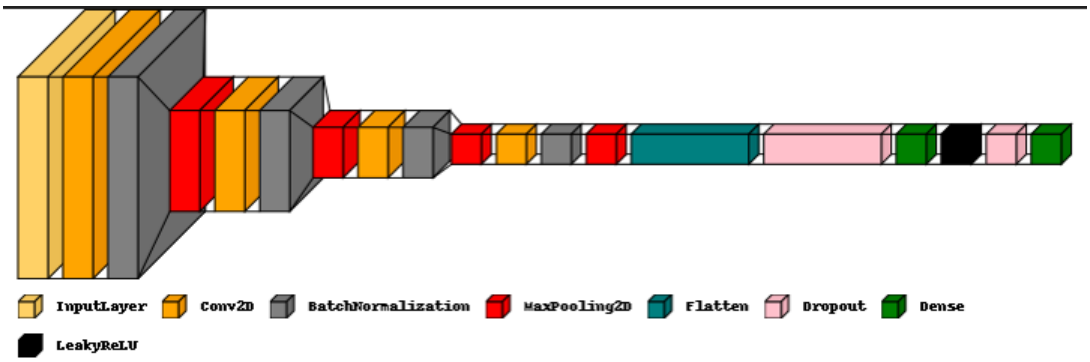

Figure 5: Original Meso-4 architecture proposed by Afchar et al. (2018)

5.4 Model Implementation

The design of the model employed in this project closely resembles the architecture proposed by Afchar et al. (2018) which was effective detecting deepfakes in Faceforensics++ developed by Rössler et al. (2019) and CelebDF dataset.The original model structure can be seen in the figure 6. The model has sequence of four successive layers of convolutions and pooling. Adjustments were made in the original model to take 224x224 images as input. There are 24,137 trainable parameters for the model. Figure 5 shows the model implemented in the project. The library of Keras is used to implement this model and it was trained on the current dataset without using any pre-trained weights. Finally the model is saved as an '.h5' extension file as the model itself takes 6 hours to train. This step is done so as to use saved results for figuring out the post-processing metrics.

5.5 Post-Processing

The model's predictions were based on frames extracted from videos, necessitating the collation of results for each video. To determine whether an image was real or fake, adjustments were made to the cutoff value, aiming for optimal results. The decision was made to set the confidence value of the model for a frame to be labeled as fake to be greater than 0.55. A dataframe was then created to store the count of images predicted as either fake or real. Finally to predict whether a video is fake or real, the number of fake and real frames from each video is counted. If the count of fake frames is greater than the 50% of the total frames, then the video is tagged as fake otherwise real. For instance, if 12 images were extracted from a video with 8 predicted as fake and 4 as real, then the number of fake frames predicted (8) are greater than 50% of total frame (6) so the video is tagged as fake. Multiple variations of both the image predicting cutoff and the video predicting cutoff were experimented with.

5.6 Evaluation metrics

Logloss is a metric which punishes the model for even one wrong prediction because of it's formula which tends to infinity. To cater to this issue, the scale of the prediction probability was shifted from 0-1 to 0.02 to 0.98, which was the way to go for participants in the DFDC challenge as well. Then precision, recall and accuracy are further calculated using their formulas from the results dataframe.

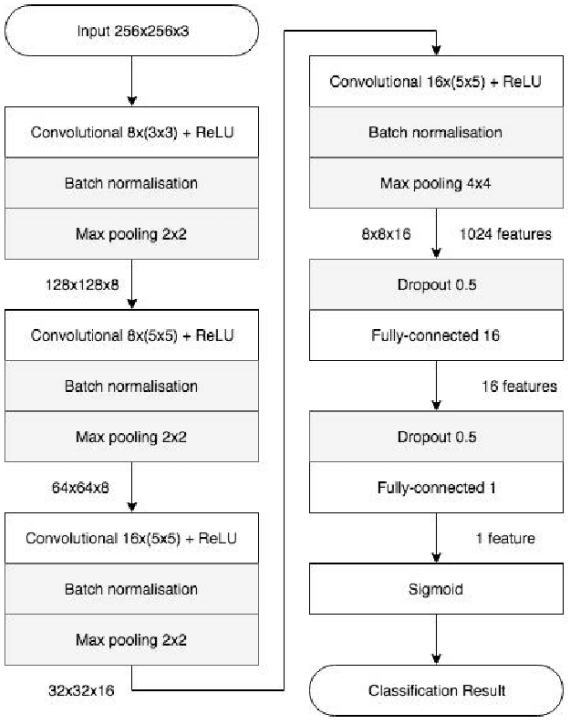

Figure 6: Original Meso-4 architecture proposed by Afchar et al. (2018)

6 Evaluation

6.1 Model Evaluation

Confusion matrix which for the train, validation and test set combination which produced best results can be seen Figure 7.

In Table 1, we can see precision, recall and accuracy for the final model for three random tests of train, test and validation sets. This was done so as to randomize the training, testing and validation sets to test the results we are getting are consistent and reliable. Also, this ensured that the result achieved in one experiment is not a fluke.

As can be seen in the Table 1, the lowest logloss value that was achieved was 0.4084 and is a good indication that the model is performing well. Both recall and precision are above 80% in in two experiments out of three, which show that the approach effectively captures a large portion of the actual positive instances while also maintaining a relatively high level of accuracy among its positive predictions.

Table 1: Values of Metrics for 3 experiments

	Logloss	**Precision**	**Recall**	**Accuracy**
Exp. 1	0.4508	82.56%	88.20%	84.57%
Exp. 2	0.4084	86.85%	88.20%	87.24%
Exp. 3	0.4760	86.44%	79.19%	83.15%

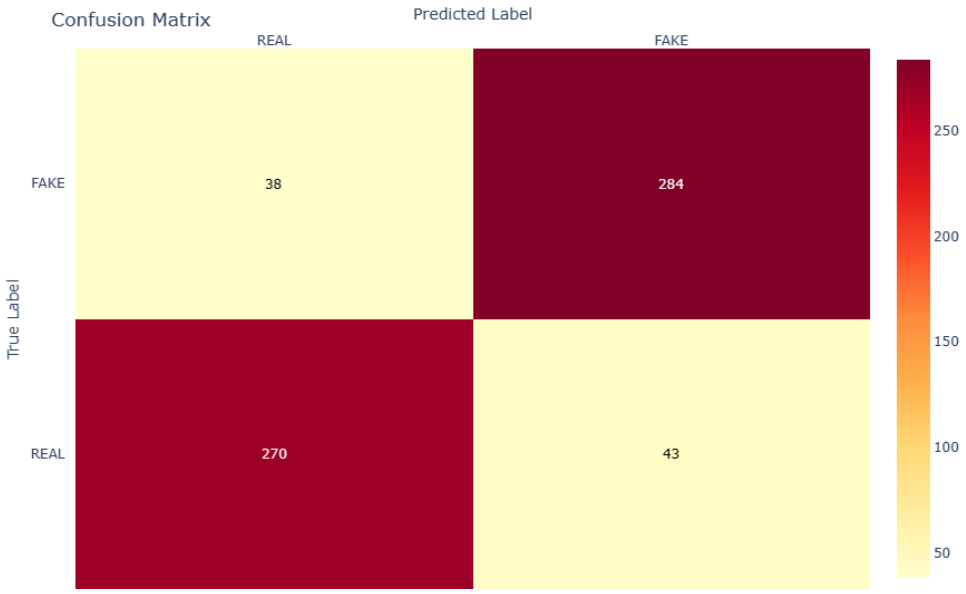

Figure 7: Confusion Matrix for Experiment 2

In Figure 8, it can be seen that the accuracy on the training set grows less rapidly after 20 epochs, which is why the value of 25 epochs was chosen. The model was tested using batch sizes of 1, 16, 32, and 64, and the highest accuracy was attained with a batch size of 64, leading to the decision to utilize it.

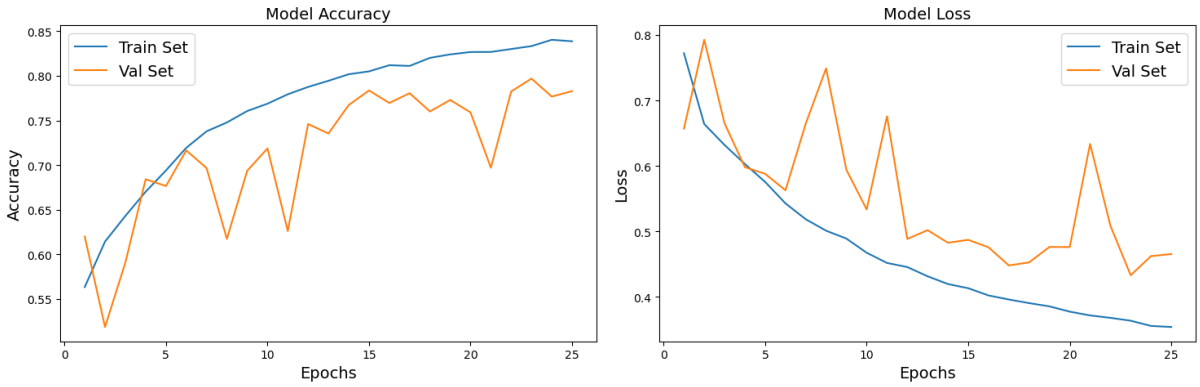

Figure 8: Model Loss and Model Accuracy vs epochs

The total time taken to train the model was approximately 100 minutes each time. Additionally, the pre-processing time required to detect faces, crop them, and save them as images averaged 6 hours for each sub-folder, resulting in a total of approximately 42 hours.

6.2 Discussion

The winner of the competition achieved a logloss score of 0.4279 [3] on the private dataset on which meta tested the results. Although approach discussed in this paper achieved 0.4084

[3] `https://www.kaggle.com/competitions/deepfake-detection-challenge/leaderboard`

by creating a test dataset from the training dataset itself by using the train_test_split, the logloss is good when compared with the hardware that the competitors were using in the competition. The contestant who finished third in the competition mentions in his methodology [4] of using DGX-1 [5] to train the model. It is a very powerful GPU with 32gb dedicated memory and it took 5 days for the training time. This means that the amount of data to train is huge as well as using deep learning on this dataset takes a lot of time. The total time taken for pre-processing and model training, coupled with decent results on the test dataset, indicates that the approach is heading in the right direction. The training of the model and number of frames extracted were limited due to the shortage of ram in the current environment in which the project is developed.

7 Conclusion and Future Work

The main objective of this research project was to analyze how a shallow CNN model used along with post-processing techniques will perform on the best dataset available for the purpose and compare it to the competitors. The first seven folders of the DFDC dataset were used for the training, testing and validation of the datasets. Meso-4, which is a shallow CNN and which performed well in detecting deepfakes on other datasets was implemented. Post-processing techniques that utilized various parameters such as the model's confidence level on images and the number of fake images required to label a video as fake were also experimented with. These results were achieved in a lot less model training time as compared to others in the competition. Majority of time was utilized in the pre-processing done on the videos because of the number of frames which were generated as well as the time it took to run the MTCNN algorithm on each frame to detect faces. Logloss score of 0.4084 was achieved with an accuracy of 83%. However the results can only be partially compared as the logloss calculated in the competition is on a private dataset which is made private by Meta. Ways to manage RAM efficiently can be looked onto remove this bottleneck and to optimize the pre-processing step. To further improve on the project, an increase in the number of frames that are extracted from the videos can be done. This will improve the number of details captured in the video. Also, a deeper network like Resnet,Inception or VGG16 can be utilized instead of a shallow one after the bottleneck of RAM is fixed.

References

Afchar, D., Nozick, V., Yamagishi, J. and Echizen, I. (2018). Mesonet: a compact facial video forgery detection network, *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–7.

Agarwal, S., Farid, H., El-Gaaly, T. and Lim, S.-N. (2020). Detecting deep-fake videos from appearance and behavior, *2020 IEEE international workshop on information forensics and security (WIFS)*, IEEE, pp. 1–6.

Alnaim, N. M., Almutairi, Z. M., Alsuwat, M. S., Alalawi, H. H., Alshobaili, A. and Alenezi, F. S. (2023). Dffmd: A deepfake face mask dataset for infectious disease era with deepfake detection algorithms, *IEEE Access* **11**: 16711–16722.

[4] `https://www.kaggle.com/competitions/deepfake-detection-challenge/discussion/158158`
[5] `https://www.nvidia.com/en-in/data-center/dgx-1/`

Amerini, I., Galteri, L., Caldelli, R. and Del Bimbo, A. (2019). Deepfake video detection through optical flow based cnn, *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 1205–1207.

Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M. and Ferrer, C. C. (2020). The deepfake detection challenge dataset.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2020). Generative adversarial networks, *Commun. ACM* **63**(11): 139–144.
URL: *https://doi.org/10.1145/3422622*

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Y., Chang, M.-C. and Lyu, S. (2018). In ictu oculi: Exposing ai created fake videos by detecting eye blinking, *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–7.

Pawelec, M. (2022). Deepfakes and democracy (theory): how synthetic audio-visual media for disinformation and hate speech threaten core democratic functions, *Digital society* **1**(2): 19.

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.(2020), *Proceedings of the 28th ACM International Conference on Multimedia, MM*, pp. 12–16.

Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J. and Niessner, M. (2019). Faceforensics++: Learning to detect manipulated facial images, *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–11.

Saikia, P., Dholaria, D., Yadav, P., Patel, V. and Roy, M. (2022). A hybrid cnn-lstm model for video deepfake detection by leveraging optical flow features, *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7.

Salman, S. and Shamsi, J. A. (2023). Comparison of deepfakes detection techniques, *2023 3rd International Conference on Artificial Intelligence (ICAI)*, pp. 227–232.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition, *CoRR* **abs/1409.1556**.
URL: *https://api.semanticscholar.org/CorpusID:14124313*

Verdoliva, L. (2020). Media forensics and deepfakes: An overview, *IEEE Journal of Selected Topics in Signal Processing* **14**(5): 910–932.

Wang, L., Zhou, L., Yang, W. and Yu, R. (2022). Deepfakes: a new threat to image fabrication in scientific publications?, *Patterns* **3**(5).

Wang, Y. and Dantcheva, A. (2020). A video is worth more than 1000 lies. comparing 3dcnn approaches for detecting deepfakes, *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pp. 515–519.

Yang, X., Li, Y. and Lyu, S. (2019). Exposing deep fakes using inconsistent head poses, *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8261–8265.

Yu, Z., Peng, W., Li, X., Hong, X. and Zhao, G. (2019). Remote heart rate measurement from highly compressed facial videos: an end-to-end deep learning solution with video enhancement, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 151–160.