DeepFake Detection

Sarthak Jain - 191IT145
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: sarthak94511@gmail.com

Yash Gupta - 191IT158
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: guptayash1104@gmail.com

Rishit - 191IT141
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: rishit.191it141@nitk.edu.in

***Abstract*—Deepfake is a technique to generate synthesized video in which the face of the person in an existing video is replaced with someone else's. With the advancement of Deep Learning and Artificial Intelligence techniques, distinguishing a deepfake from a real video is becoming an increasingly difficult task. This has led to various issues in recent times such as spreading fake news, communal violence, and revenge videos by making objectionable deepfakes. Thus detecting such videos is becoming more and more necessary. In this project, we aim to develop a novel system to detect if a video has been morphed, manipulated and deepfaked or not.**

I. Introduction

Deepfake is a technique of generating synthesized video using deep learning techniques to swap the face of a the person in the video with the face on the provided image in such a way that the generated video has the target person doing or saying things the source person does. While some deepfakes are generated using graphical image and video editing but with advancement in deep learning techniques, high quality deepfakes are easily generated using models such as autoencoders and generative adversarial networks (GANs). These models analyze facial movements and expressions of the source person in the input video and synthesize facial images of another person making similar expressions and movements. These kind of morphed videos are getting difficult to be detected day by day. This a great threat to privacy and world security. Deepfakes of famous personalities and politicians is made for unethical purposes, spreading misinformation and promoting violence. This can lead to ruining of political relations between countries and also may fool people into believing a statement that was never said by that particular individual.

Recent studies have shown that deepfake videos have been so heavily circulated because of easy availability of tools to create deepfakes in almost no time that detection of these have become increasingly important. Thus developing deepfake detection methods have witnessed a surge of interest recently. There are few good dataset available for deepfake detection such as DFD, DFDC, Celeb-DF, etc. However for certain datasets, the synthesized videos are clearly distinguishable as deep fake as they have been generated using not so good deepfake methodology. In present work, we have used Celeb-DF dataset for developing and evaluation of our deepfake detection model because the synthesized videos in this dataset are generated using advanced deepfake generation methodologies. The dataset consists total of 400 videos where 70 videos are original and 330 are synthesised videos generated by using deepfake generation models on the original videos. The original videos are taken from popular video sharing platform, YouTube.

II. Problem Statement

"Video Deepfake Detection"

A. Objectives

- Cluster the video data for better test train split.
- Perform inter-frame analysis to detect morphing directly.
- Build a model to predict deepfake or not based on the features extracted from the video.

III. Literature Survey

Through [1] we learnt about the extent of manipulation these deepfakes can lead to. We saw various examples where deepfakes were used against the popular world leaders to manipulate and misguide public against them. Through [2] we learnt about the feature extractions and creation of embedding which can be used to make predictions. We saw complex encoder architecture. More precisely autoencoders architecture was used in this paper.

In [3] we saw architecture similar to the one in the previous architecture. Variational Bayes technique was also incorporated in this methedology. We learnt about the embeddings generated and how they can be used further.

Through [4] we saw the structure of generative adversarial networks and how the can be used to generate data. Deep fake frames can be created by using generative adversarial networks as they can be used to do style transfer. GANs on very huge datasets can generate very hard to detect deepfakes. In [5] we saw the structure of adversarial auto-encoders. It is the

concept in which auto encoder architecture is fused effectively to adversarial loss as show In the [4]. They too can be used to generate good deep fakes.

Through [6] we learnt about the detection of the face if present in an image using the various techniques and models. We also saw various methods available for facial landmark detection. Several pre-trained models were used and the comparison was showed in this paper. Through [7] we learnt about the detection of the face if present in an image using the generative adversarial networks. We also saw methods for facial landmark detection. Hyper parameter tuning was performed and results were compared when various hyperparameters were used.

From [8] we saw method similar to [7] and [6] where image and video synthesis was shown using the generative adversarial networks. Various applications were also shown where they can be used with general idea of using them.

In [9] we saw various methods of deep fake detection and the performance of these models are compared. Various pretrained models were also used for face detection and facial landmarks recognition than embedding vectors are created which are further used for training. [10] was similar to [1] where maipulation through deepfakes are shown. It also shows how modern computing has made generation of deepfakes really easy. It also shows some real life examples where deep fakes were used.

IV. Methodology

A. Dataest

The dataset is comprised of mp4 files. A metadata.json accompanies each set of mp4 files, and contains filename, label (REAL/FAKE) and original video file if fake.The dataset contains 400 videos out of which some are fake and rest are real. In order to predict whether the videos are fake/real we have used two different methods – inter frame analysis of videos and deepfake detection using deep learning.

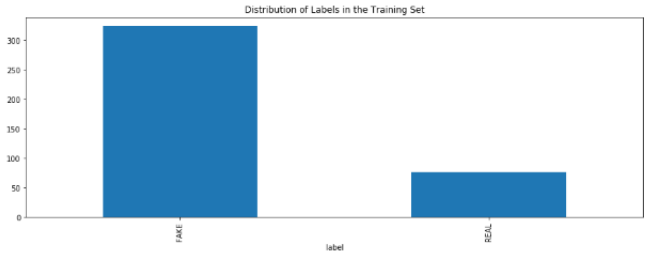

Fig. 1: Dataset

B. Feature Extraction

The ultimate goal of this work was to determine whether a video was genuine or generated using deepfake technology. As a result, the system's input must unquestionably be video. However, because deep learning models use images as input, the system (video) input must be converted to model input. Each frame in the video contains more than just a face. Indeed, the person's body parts and the image's background area take up the majority of the video frame. These irrelevant features can have a negative impact on the model's training. The image's focus is on the face, and the pre-processing module must capture the image's face as model input. Thus faces were extracted and the deep-learning model was trained on these datasets.

Fig. 2: Facial Landmarks

Fig. 3: Interframe Analysis

C. Inter-frame analysis

In order to perform the interframe analysis of the videos ,firstly the video was split into frames using the OpenCV module of python.Each video was divided into approximately 400 frames thus producing approximately 18000 frames for the entire dataset.On obtaining these frames, the faces were located in these frames using the face recognition module of python.Once the faces were detected the faces were zoomed and padding removed in order to filter out unnecessary data.On obtaining the faces the facial landmarks were marked using the face recognition module of python.These landmarks were then stored into a numpy array where each landmark was described by two co-ordinates the x co-ordinate and the y co-ordinate.

On obtainig the landmarks array each frame was compared with it's previous frame. If the numpy arrays were very

different that is if the arrays differed in a certain number of positions then the frame was said to be deepfaked. In order to compare the frames we have used mean squared error and similarity where in if the scores fall above a certain threshold (120 for mean squared error and 96 for s-similarity) then the frame is said to be deepfaked and the corresponding frame number is also obatined. Thus by analysing the frame number and the timestamp the purity of the video can be retained by removing the respective frame. However this method posed a problem of reporting a large number of false negatives. In order to overcome this problem we adopted the deep learning approach to detect deepfakes in video files.

D. Classification

In order to overcome the problem of false negatives, we have adopted the deep-learning approach in which the data files are split into train and test data following which the model is trained on the trained data and is evaluated based on the results obtained from test data. However,the train-test split method becomes biased if certain similar videos (of the same person) are placed in the train set and the test set contains videos of another person. Because images extracted from the same video can have some similarity, it is preferable that images extracted from the same video be in the same data set. If the model was trained using images from a single video, it will have a better chance of correctly predicting other images from that video. The goal, however, is to identify common features shared by different videos.

1) Clustering for efficient test-train split: In order to overcome this train test split problem we have clustered the data frames and picked certain frames from each cluster to place them in the training set and the remaining were placed in the test set thus removing the biasing of data points. In order to cluster the data points frames were generated from the videos and faces extracted from them using the same methodology mentioned in the previous section.These frames were then embedded using facenet embeddings which reduces the dimension of these frames to 512 dimensions.Thus we obtained arrays of size 512 for each frame. In order to cluster these frames the similarity scores between these 512 dimensional arrays were computed and the ones with highest similarity scores were clustered into the same group.

To achieve clustering of data frames we have incorporated 3 methods namely Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding(t-SNE) and Density Based Clustering (DBSCAN).The clusters obtained were then evaluated using spotify's annoy where in the a randomly selected frame was fed as the input and the module then computed the 8 closest/most similar frames.It was observed that these frames belonged to the same cluster,thus proving that the clusters obtained were highly accurate.The best results were obtained through Density Based Clustering (DBSCAN) where in 772 clusters were obtained.These clusters were then stored into a datframe from which a feather file was developed.The feather file thus contained the video,it's frame and the cluster number.

The feather file was then exported and the clustered data were stored in a set and then a 70-30 split was applied to each cluster and the corresponding videos were split into train and test files.Thus we could successfully achieve the train test split where in redundancy of data and inconsistency in the same were removed in order to make our deep-learning model more universal.

On obtaining the train and test data, the train data was broken down into frames using the procedure mentioned in the inter frame analysis following which faces were extracted from the same. The landmarks were then marked onto these faces and the facial feature vectors were used to train our classifier model where in the distance between the centroid and the facial features were calculated and stored in an array. The model was then trained on this array to make predictions. XGBoost gave the best results among the various ML algorithms. The model was then evaluated on the test data where in the accuracy recorded was 72.59 percent. Thus,we successfully developed a deepfake detector model using deep learning algorithms.

V. Results and analysis

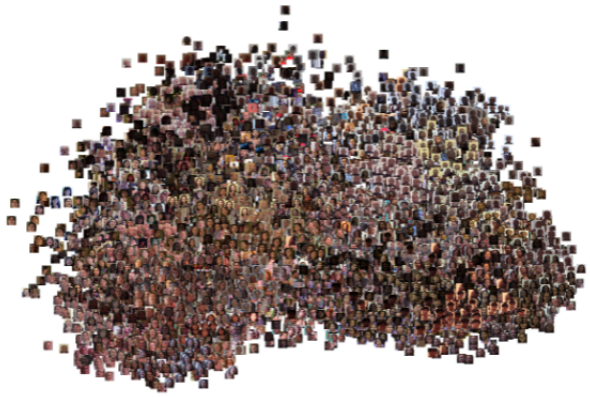

Fig. 4: PCA

Fig. 5: t-SNE

This section consists of the results generated while analyzing the data, clustering, and predicting the videos. Fig. 4, Fig.

5, and Fig. 6 shows the result of clusters formed after applying PCA, t-SNE and DBSACN algorithms respectively.

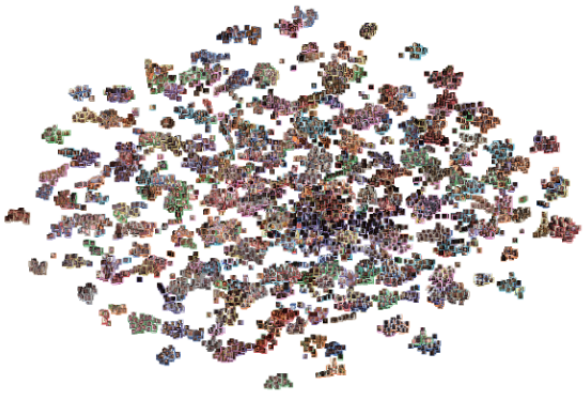

Fig. 6: DBSCAN

Fig. 7 shows the similar faces in the created cluseters.

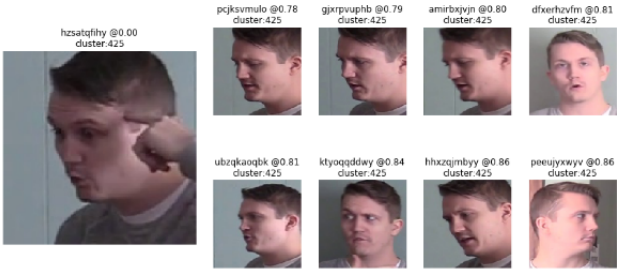

Fig. 7: Getting similar faces using ANNOY

Fig. 8 shows the bar plot of the predictions made.

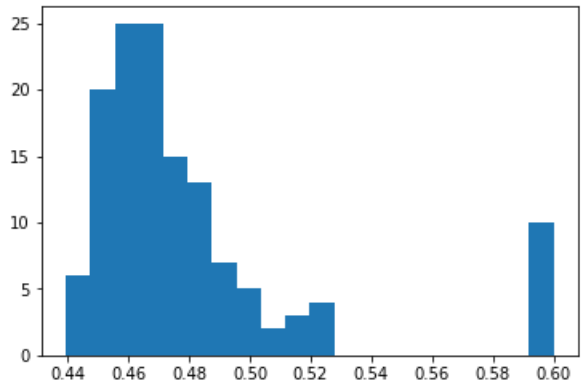

Fig. 8: Predictions

Fig. 9 and Fig. 10 shows the results of classification on test data.

Fig. 11, Fig. 12, Fig 13, and Fig. 14 shows the results of different evaluation metrics.

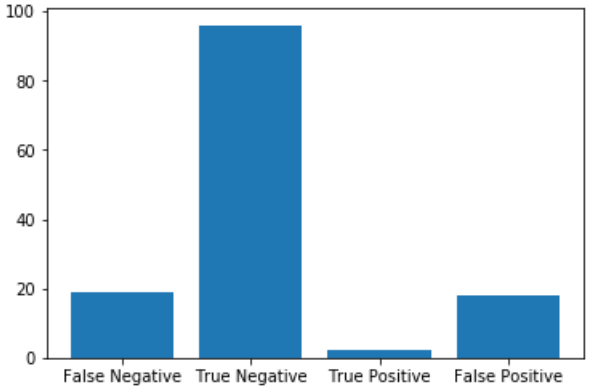

Fig. 9: Classification Results

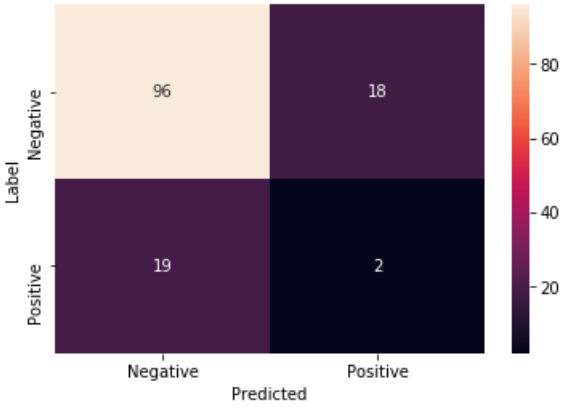

Fig. 10: Confusion Matrix

CONCLUSION

As processing power of modern hardware available is increasing it has become easy to create deep fakes usually to manipulate people on internet in order to fulfil one's desired goals.These deep fakes ca influence people's opinion about someone or something and may create ecosystem with undesired hatred.Through this project we try to automate process of deep fake detection using machine learning approaches.We also try detecting fake segment of video that is placed at some place between true video.This type of deepfakes are hard to detect ,so we performed statistical inter frame analysis in order to detect all such frames.In order to perform inter-frame analysis we located face in each frame ,if face in a frame is detected than we locate facial landmarks and perform structural similarity.If structural similarity is less than certain threshold we classify that frame to be fake.Threshold is obtained by observing fake data from our dataset. In order to detect deep fakes we need to have effective train test split. We tried forming clusters of high dimension data using PCA,T-SNE and DBSCAN. DBSCAN gave best clustering results in our case.Clusters formed are used to improve our training and testing split.Embedding vectors of all the images

```
accuracy = (TP + TN)/(TP+TN+FP+FN)
accuracy
```

```
0.725925925925926
```

Fig. 11: Accuracy

```
precision = TP/(TP + FP)
precision
```

```
0.1
```

Fig. 12: Precision

```
recall = TP/(TP + FN)
recall
```

```
0.09523809523809523
```

Fig. 13: Recall

```
f1_score = 2*((recall*precision)/(recall + precision))
f1_score
```

```
0.0975609756097561
```

Fig. 14: f1-score

are obtained using the facenet model.Faceet gives us 512 dimensions embeddings.We also tried verifying our clusters using method called Approximate Nearest Neighbour Oh Yeah (ANNOY).This helped us in successfully verifying our test-train split.After analysis we tried displaying appropiate results in form of images.In order to visualise clusters we printed clusters.PCA proved to be performing worse in our case.Now in order to detect fake frames we used two neural networks MTCNN and InceptionResnetV1.MTCNN is used as the face detector whereas InceptionResnetV1 which is pre-trained on vggface2 is used as facial recognition model.First we pas image batch to MTCNN and filter out frames without faces.Then we try generating facial feature vectors using our pretrained model.Centroid is calculated using the embedding vector.Prediction is made on embeddings obtained. Through our methodology we were able to achieve accuracy of around 72 percent , recall fo around 0.095 and precision of 0.1. Through our methedology we are successful in segregating deep fakes from the original videos.Through this we hope to remove content from internet which is meant to target certain group of people or to manipulate people.

ACKNOWLEDGEMENT

We would like to express our gratitude to Dr. Anand Kumar M for his assistance in making our project successful, along with the IT department, NITK, for providing us the opportunity to work on this project.

REFERENCES

[1] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In Computer Vision and Pattern Recognition Workshops, volume 1, pages 38–45, 2019.

[2] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine learning, pages 1096–1103,2008.

[3] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114, 2013.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in Neural Information Processing Systems, 27:2672–2680, 2014.

[5] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. arXiv preprint arXiv:1511.05644, 2015.

[6] Ayush Tewari, Michael Zollhoefer, Florian Bernard, Pablo Garrido, Hyeongwoo Kim, Patrick Perez, and Christian Theobalt. High-fidelity monocular face reconstruction based on an unsupervised model-based face autoencoder. IEEE Transactions on Pattern Analysis and Machine Intelligence,42 (2):357–370, 2018.

[7] Jiacheng Lin, Yang Li, and Guanci Yang. FPGAN: Face deidentification method with generative adversarial networks for social robots. Neural Networks, 133:132–147, 2021.14

[8] Ming-Yu Liu, Xun Huang, Jiahui Yu, Ting-Chun Wang, and Arun Mallya. Generative adversarial networks for image and video synthesis: Algorithms and applications. Proceedings of the IEEE, 109(5):839–862, 2021.

[9] Siwei Lyu. Detecting 'deepfake' videos in the blink of an eye.http://theconversation.com/detecting-deepfake videos - in - the - blink - of - an - eye - 101072, August 2018.

[10] Bloomberg. How faking videos became easy and why that'sso scary.https://fortune.com/2018/09/11/deepfakes-obama-video/, September 2018.