Model to detect Deepfakes (Option D)

Group members: Rashmi Datta(rdatta2), Kriti Singh(ksingh23), Devadharshini Ayyappan(dayyapp)

Introduction:

In this report, we will discuss the progress of our machine learning project, which aims to classify real and fake faces in videos using the Multitask Cascaded CNNs (MTCNN) and XceptionNet architecture. The second phase of the project involved creating embeddings for the faces extracted from MTCNN through Facenet and then applying triplet loss to learn the features better and giving the embedded features to the train classifiers like SGD and Random Forest for classification. We were successful in achieving these milestones with the help of facenet embeddings and achieved results with improved performance metrics.

Milestones Achieved in Phase 2:

1. Created embeddings of extracted faces from frames of the dataset to learn features which can be used to train classifiers for deepfake classification.
2. Increased the training data to 20k and could see some improvement in training and testing accuracy of Xception Net.
3. Through the triplet loss method the accuracy improved with AUC score 0.95 and F1 score approximately 0.88.

Literature Review:

We reviewed several papers on face detection and video classification to improve our model. We found that using the MTCNN for face detection and the XceptionNet architecture for video classification is a popular and effective approach.While deep CNN models can very effectively detect local artifacts, modern deepfakes generation techniques can create a wide range of artifacts, from ones that are local to those that span the entire image. Moreover, there is a large diversity in the types of artifacts produced owing to the different types of generation techniques available.

Research [1] introducing the FaceForensics dataset, analyzes the performances of various detectors based on leared features i.e architectures like MesoNet, Inception and XceptionNet.Among the different architectures , XceptionNet performs best on various image corpus size compared to other architectures.

In [2] Zhu *et al.* propose to utilize 3D facial details to detect deepfakes. Authors find that merging the 3D identity texture and direct light is significantly helpful in detecting deepfakes. They employ the XceptionNet CNN model for feature extraction. A face cropped image and its 3D detail is used to train the detection model. They also perform a detailed analysis of a number of different feature fusion

strategies. The proposed technique was trained on FaceForensics++ dataset and evaluated on(1) FaceForensics++, (2) Google Deepfake Detection Dataset, and (3) DFDC datasets.

As analyzed in [3], XceptionNet was performing better on a huge corpus of data and many other methods like LSTM and RNN were used to improve the accuracy of the predictions.The accuracy didn't show much improvement.Metric learning was used on the image embeddings learned from the facenet model which showed huge improvement in the accuracy of predictions.

Vision transformers have been used with XceptionNet in [4] for deepfake detection and the method has shown promising results compared to the performance of just convolutional networks for the deep fake detection.

Architecture of Our Model:

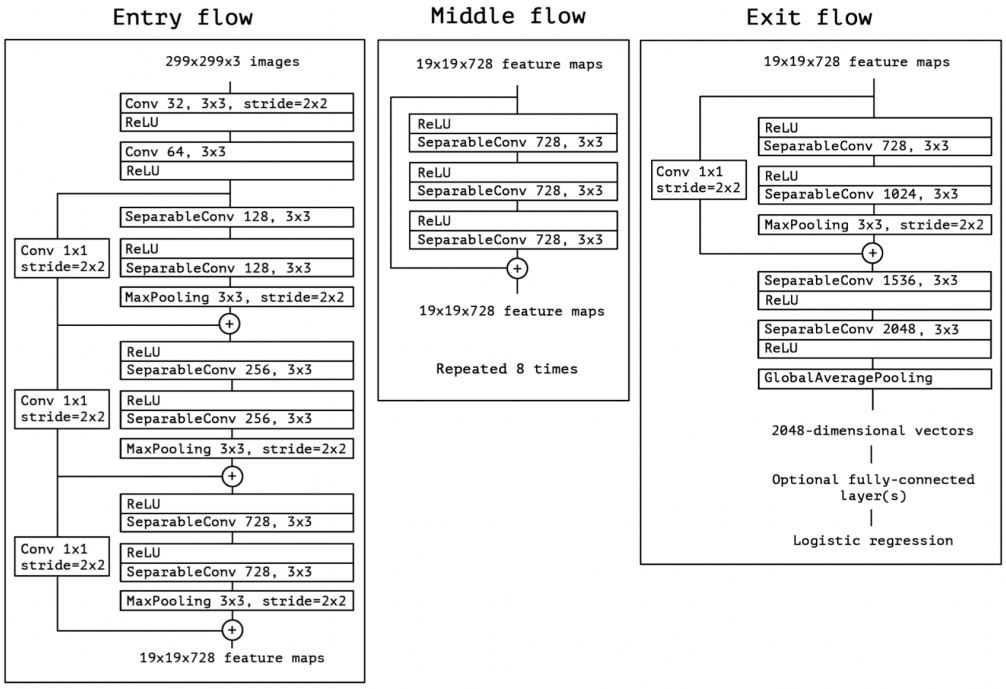

Fig1. Xception Network

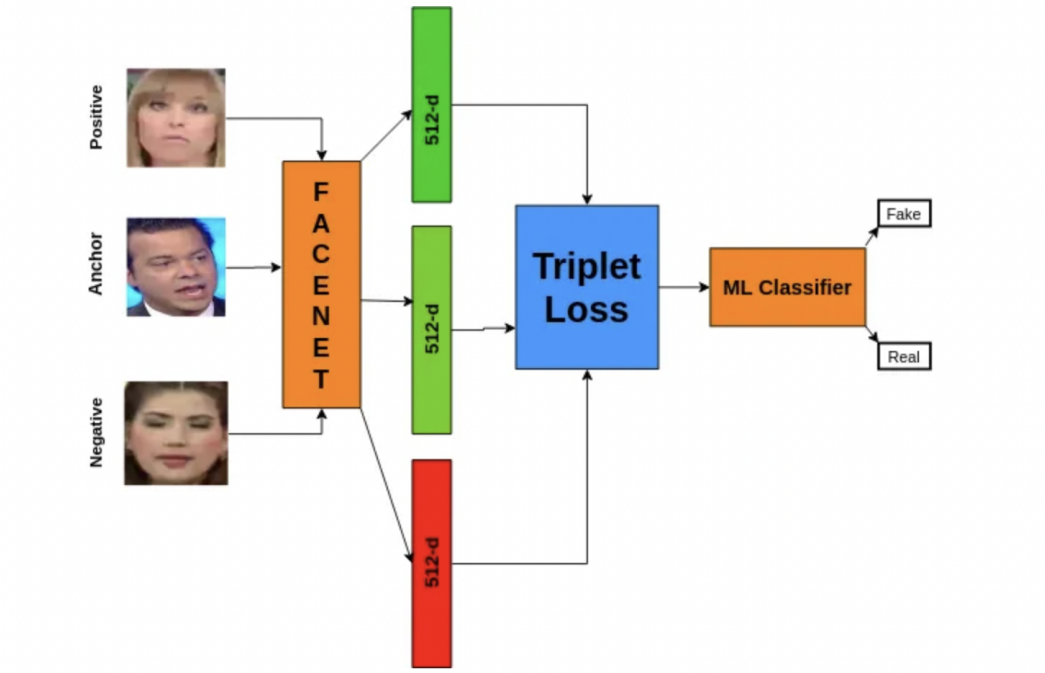

Fig2. Triplet loss method

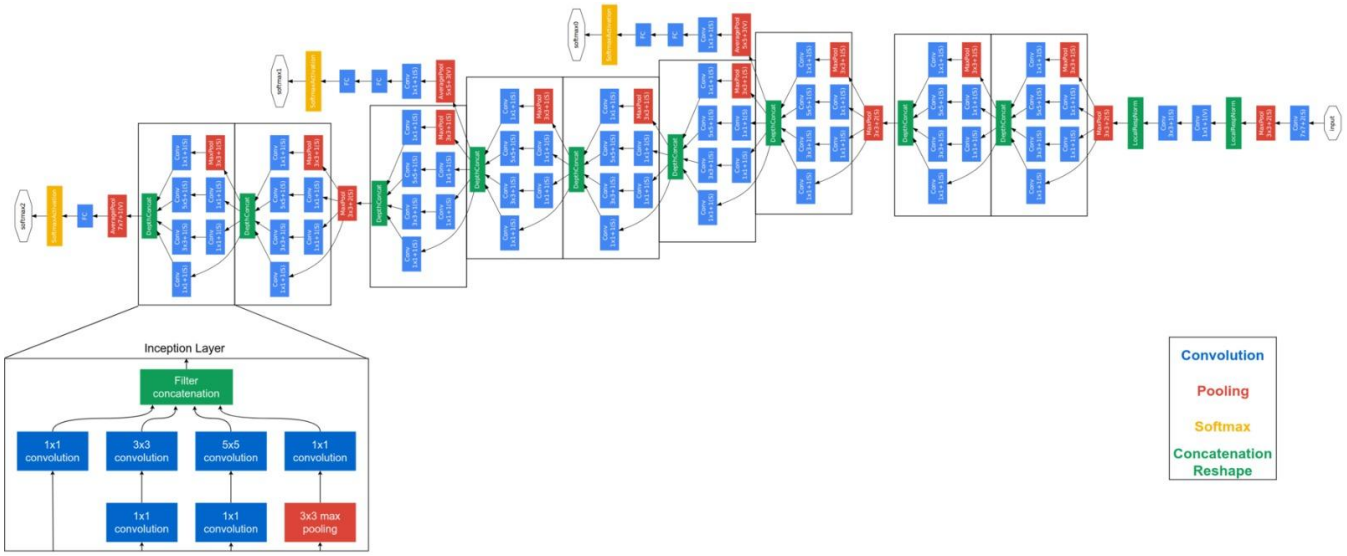

Fig3. Facenet Architecture

Triplet Loss Method:

It's a type of metric learning where the similar features are grouped together and dissimilar features are put farther apart in feature space. Let's take the anchor input sample A, a sample with the same label as P, and a sample with a different label as N. The loss function of triplet is defined as follows:

$$L(A,P,N)=\max((f(A)-f(P))^{**}2) - (f(A)-f(N))^{**}2) +\alpha,0)$$

There are three different type of triplets generation methods based upon the distance between anchor, positive and negative embedding vectors.

A. Easy Triplets: In this case, the distance between negative and anchor embedding is greater than the distance between anchor and positive embedding plus margin, i.e. d(a, p)+ margin < d(a, n). Hence, the loss propagated is zero and it does not help the network to learn anything.

B. Semi-hard Triplets: Distance between anchor and negative is between the distance between anchor and positive, and, distance between anchor and positive plus margin, i.e. d(a, p) < d(a, n) < d(a, p) + margin. The loss propagated is positive and zero in this scenario.

C. Hard Triplets: The distance between anchor and negative is less than the distance between anchor and positive plus margin, i.e. d(a, n) < d(a, p) + margin .Hence, the loss propagated backwards is always positive in this case.

Dataset Review:

We analyzed our video classification approaches using the datasets Celeb-DF and Celeb-DF-v2. This dataset comprises 52 celebrities whose interviews are available on YouTube. Celeb-DF dataset, released in the year 2019, contains 560 real videos and 5639 deepfake videos. They considered various factors such as gender, age and ethnic group bias to make the dataset more challenging. They created 5639 deepfake videos by swapping the faces amongst 59 celebrities. The frame size is arbitrary in these videos. Video format is MPEG4. We have randomly selected over 1500 videos each from Celeb-real and Celeb-synthesis to train and test our model. We have extracted 25 frames of each video into the train_frames folder where the subfolder '0' contains real video faces and '1' contains fake video faces for further processing. After face extraction, we have generated a csv datafile indicating which video is real and which is fake. Following this, we have segregated the data and labels(.npy) embeddings, which are passed through neural network layers with triplet loss.

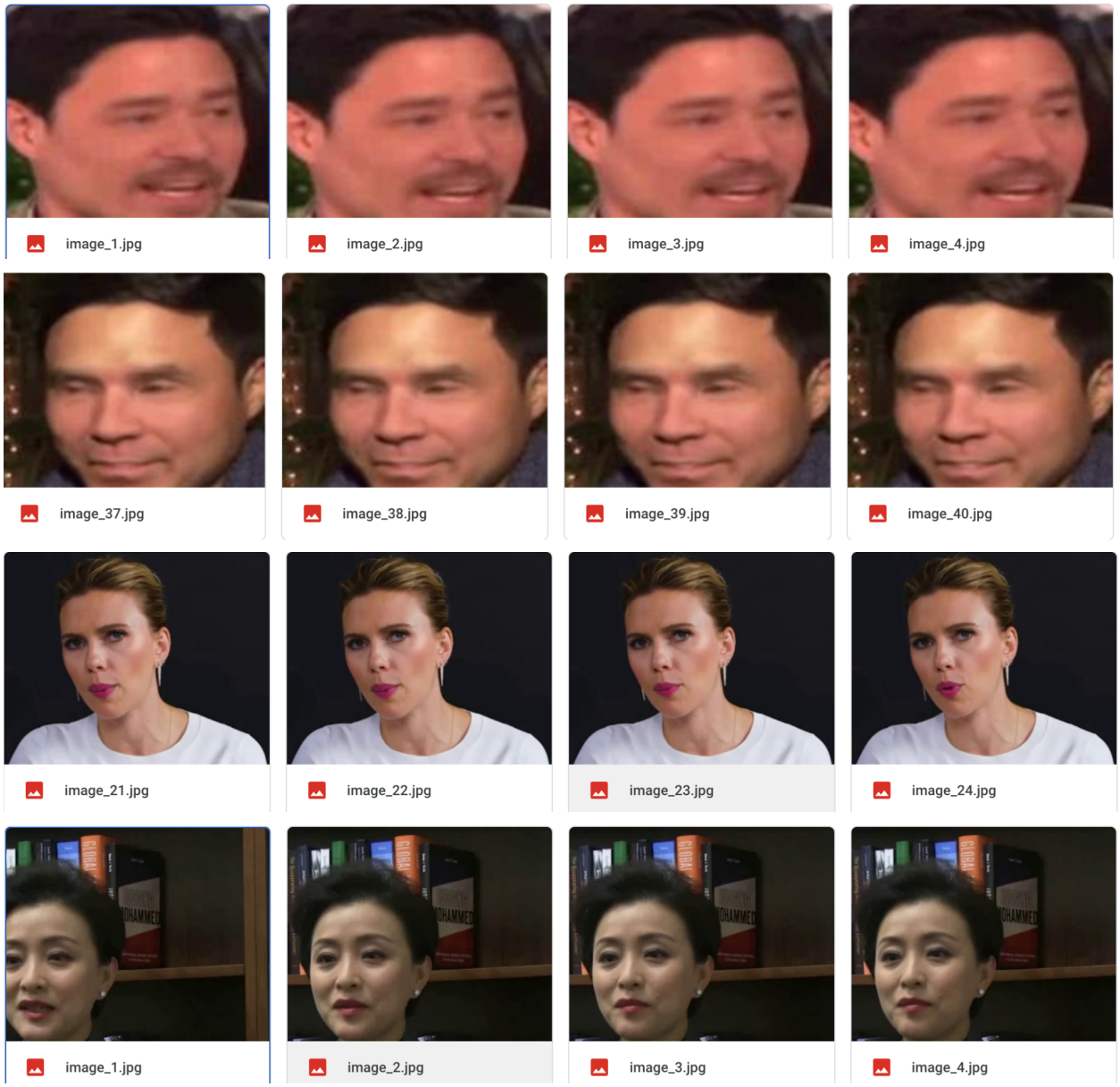

Fig: Above, the first two rows depict the deep-fake frames of two of the videos from the CelebDF-v2 dataset. The next two rows are examples of original sequences of two videos from the same dataset.

Challenges addressed from Milestone 1:

The accuracy we got by training Xception Net was not satisfactory with the amount of data we trained initially.We tried increasing the training data and the training and testing accuracy improved but the performance was not satisfactory.

Implementation:

Our training data consisted of approximately 30,000 images with equal distribution of manipulated and original images from both Celeb DF and Celeb DF v2 datasets. Our labels for the training data were categorical in nature with '0' representing the image is real and '1' representing that the image is manipulated. Tensorflow libraries were used for the implementation on google colab platform.

We have used semi-hard triplet loss.With 25 frames per video, we took 30,000 embedding into consideration. We generated 512 face embedding vectors using facenet. Then we trained a network with triplet loss function to get a better segregation of features and once we got the features we used it as a training dataset for different classifiers like KNN, SGD and Random Forest.

Results:

Our classifiers were able to classify real and fake faces with a high degree of accuracy. We achieved a testing accuracy of 87% on the dataset we used.We gave videos for testing where for each video we extracted frames and made predictions on the features of the 25 frames and then took the mean of the predictions to come up with a final classification output.

AUC Score: 0.9513888888888888

Accuracy: 0.875

Precision: 0.8

Recall: 1.0

F1 score: 0.888888888888889

Fig: Scatter plots comparison of data before and after training the model with Triplet Network

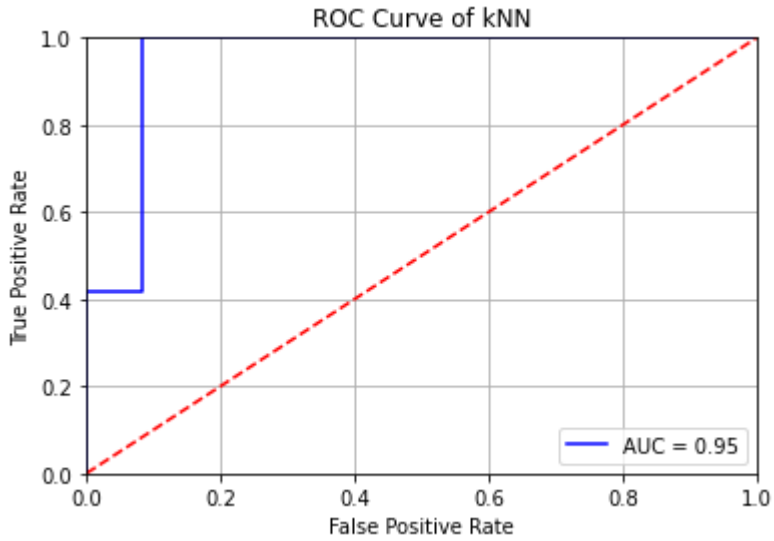

Fig: ROC curve of the K-Neighbors Classifier depicting False vs True positive rate

Conclusion:

In conclusion, we have achieved several milestones in phase 2 of our project. We were able to extract faces from frames using the Triplet network and classify real and fake faces in videos using the Facenet architecture. We have addressed the issue of spatio-temporal learning and less accuracy in the previous milestone. Our model is innovative and uses metric learning to learn crucial features from the triplet network architecture. We plan to explore more by tuning the hyperparameters of the triplet network to see if that makes a difference in the quality of feature segregation and extraction. We have demonstrated performance close to the one shown in our base literature and we aspire to improve our model further to outdo the current achievements.

References:

[1].FaceForensics++: Learning to detect manipulated facial images https://arxiv.org/abs/1901.08971#:~:text=https%3A//doi.org/10.48550/arXiv.1901.08971

[2]Xiangyu Zhu, Hao Wang, Hongyan Fei, Zhen Lei, and S. Li. 2021. Face Forgery Detection by 3D Decomposition. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2928–2938

[3]Detecting deepfakes with metric learning https://arxiv.org/abs/2003.08645#:~:text=https%3A//doi.org/10.48550/arXiv.2003.08645

[4] Shreyan Ganguly, Aditya Ganguly, Sk Mohiuddin, Samir Malakar, Ram Sarkar/ViXNet: Vision Transformer with Xception Network for deepfakes based video and image forgery detection.

[5]Xception: Deep Learning with Depthwise Separable Convolutions

https://doi.org/10.48550/arXiv.1610.02357

[6] FaceNet: A Unified Embedding for Face Recognition and Clustering

https://arxiv.org/abs/1503.03832