STUDY ON DEEPFAKES DETECTION

A Project Report Submitted

in Partial Fulfilment of the Requirements

for the

B.Tech Project

in

Computer Science And Engineering

by

ASHISH GUSAIN

(Roll No. 2017BCS0012)

to

DEPARTEMENT OF COMPUTER SCIENCE

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

KOTTAYAM-686635, INDIA

November 2020

DECLARATION

I, **Ashish Gusain** (**Roll No: 2017BCS0012**), hereby declare that, this report entitled **"Study on Deep Fakes Detection"** submitted to Indian Institute of Information Technology Kottayam towards partial requirement of **B.Tech Project** in **Computer Science And Engineering** is an original work carried out by me under the supervision of **Dr. Ebin Deni Raj** and has not formed the basis for the award of any degree or diploma, in this or any other institution or university. I have sincerely tried to uphold the academic ethics and honesty. Whenever an external information or statement or result is used then, that have been duly acknowledged and cited.

Kottayam-686635 **Ashish Gusain**

November 2020

CERTIFICATE

This is to certify that the work contained in this project report entitled **"'Study on Deep Fakes Detection"** submitted by **Ashish Gusain** (**Roll No: 2017BCS0012**) to Indian Institute of Information Technology Kottayam towards partial requirement of **B.Tech Project** in **Computer Science And Engineering** has been carried out by him under my supervision and that it has not been submitted elsewhere for the award of any degree.

Kottayam-686635
November 2020

Dr. Ebin Deni Raj
Project Supervisor

ABSTRACT

In recent years, many deep learning based models have made it easy to create believable face swaps with only a few seconds video. Some of the scenarios where these realistic fake videos can be seen making impact are political distress, blackmailing someone or fake terrorism events.

The present project focuses on developing a deep-learning based deep fakes detection model that can distinguish original images from deep fake images. Our system uses a CNN model to get face embeddings and then training ML classifiers over those embeddings. Evaluation and training of the model is done over a huge face forensics dataset to get better results.

Contents

List of Figures

Chapter 1

Introduction

In 1865, the first attempt known to swap someone's face was found and the portrait was of the U.S. President Abraham Lincoln. Recent advances in the last few years have radically changed the ways of image and video manipulation. The access of modern tools such as Tensorflow or Keras have made these activities even more easy. Generative adversarial network (GAN) and autoencoders models have made tampering any kind of images and videos easily done within the reach of any individual with a computer.

There are many applications available in the market like FaceApp and FakeApp that can create deep fakes for free. These tools can swap faces, generate a new face and many such activities. FaceApp even allows one to change gender, hairstyle, age and other facial attributes. FakeApp is an app that allows one to create deep fake video and manipulated video clips. He used tensorFlow, social media websites and public videos available to insert someone else's face onto each frame of the videos. These activities can be used negatively and hence a solution to resolve this issue must be found.

Chapter 2

Problem Description

The trend of deep fakes started with the Barack Obama campaign of 2008, when the Republican opposition came with a cropper against the former's social media onslaught. Such cases are the results of using artificial intelligence and deep learning solutions in a negative way, to believably mimic the real world, be it images, music,or any kind of speech. Higher the public availability of video footage of an individual in front of the general public, the stronger the possibility of algorithmically generating their fake videos.

There are three major problems with deep fakes that make them particularly worrisome. The first problem relates to the narrative that is created in our minds by a moving image. These deep fake videos will trouble us some or the other day because from inside, whatever we read and view makes an impact psychologically. The second problem is that denying deep fake videos becomes a completely difficult task because of the way in which they are created using advanced GAN and autoencoder technologies. Even videos and the audio clips doctored using less advanced technologies are not easy to dis-

tinguish and in future will come as one of the challenging tasks to compete with.

The third problem is the influence of political parties over the people using these technologies. These kinds of videos are widely spread during a political campaign and create many false beliefs among the people. For this additional reason too, independent organizations like the Election Commission of India must focus on this issue and address the deepfake problem before it becomes a problem that is unmanageable. It has to be understood that a normal human brain cannot process these scenarios by himself and cannot distinguish these videos by himself.

Chapter 3

Literature Survey

3.1 Face embeddings

Facial detection and recognition is included among the most exciting applications of deep learning. The rise in facial recognition and detection systems has been phenomenal in recent years, however, there are many artificial intelligence enthusiasts who think that the usage of these facial recognition techniques must be properly regulated in order to prevent any kind of misuse that could be done.

FaceNet model provides a unique CNN architecture for performing multiple tasks like face recognition and clustering of various faces into similar labels. In order to achieve such a level of accuracy, it uses deep convolutional networks along with triplet loss. This provides unique embeddings for facial recognition and clustering tasks and then maps each of the face into a euclidean space such that the distances between different faces in that space correspond to face similarity, i.e. an image A of a person will be placed closer

to the image B of same person, due to similarity in their embeddings and away from an image C of a person if they have difference in their embeddings.

FaceNet uses a deep convolutional neural network (CNN) model and the complete architecture is trained in such a way that the squared L2 distance between all the facial embeddings corresponds to the face similarities between other embeddings. The images used for the training purpose must be cropped to the facial level so that the bounding box can focus over only the face.

The main difference between other techniques and the FaceNet model is that facenet learns the mapping from the images, generally faces and creates embeddings for them rather than using any distinct bottleneck layer for recognition or other similar tasks. Once the embeddings are created, in that scenario all the other tasks like facial recognition, verification etc. can be performed using these newly generated embeddings as the feature vectors as their representation.

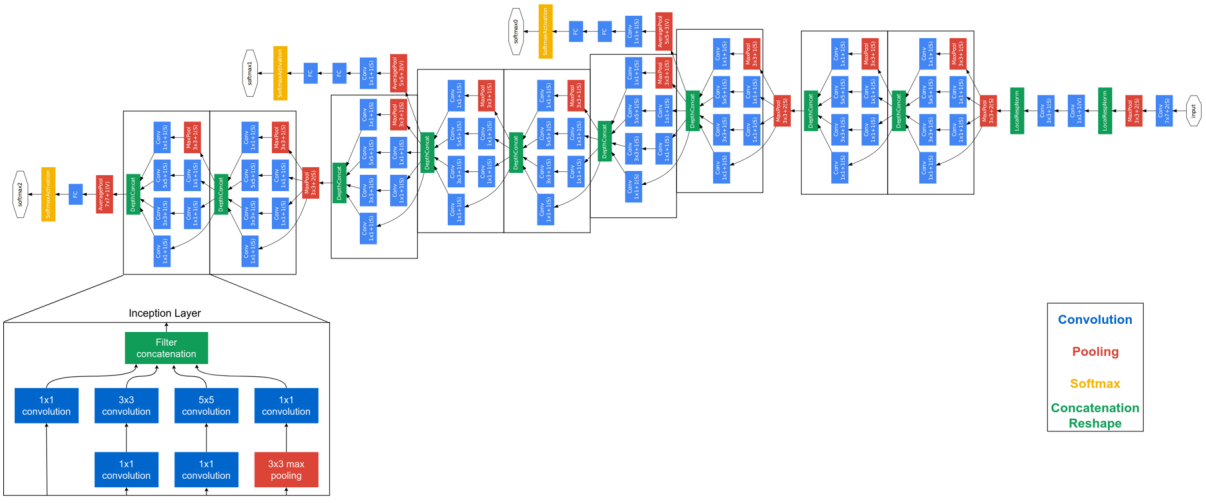

Figure 3.1: GoogleNet architecture used for face embeddings.

3.2 Intuition for Triplet loss

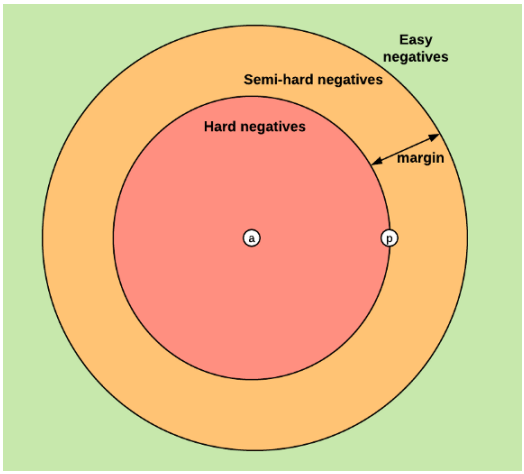

Figure 3.2: Types of triplet losses

The main goal of the triplet loss function is to make sure that two identities with same labels must have their vector embeddings close together while two identities with different labels must have their vector embeddings away from each other. To generalise the above statement, the loss will be defined in a way over the triplets of embeddings for an anchor, a negative image of a different class and a positive image of the same class.

For this purpose, generally we keep a threshold value to decide whether the distance is high or low. According to the above-mentioned analogy, in a 128-dimensional space, one can say that two faces are different if the distance between any two encoded points is high and are the same if the distance is low. Now, while training, the model will adjust it's weights in such a way that the distance between the embeddings of an anchor image and the positive image will be low while the anchor image and the negative image will be high.

Let A, P and N stands for anchor, positive and negative image respectively.There are three different type of triplets generation methods based upon the distance between the embeddings of anchor, positive and negative faces:

- **Easy Triplets:** The distance between the embeddings of negative and anchor faces must be greater than the distance between the embeddings of anchor and positive faces along with a margin. The loss propagated here is zero and this does not helps the network to learn much.

  d(A,P) + margin <d(A,N)

- **Semi-hard Triplets:** The distance between the embeddings of anchor and negative faces is the distance between the embeddings of anchor and positive faces and the distance between the embeddings of anchor and positive faces plus a margin. The loss propagated here is positive or zero in this case.

  d(A,P) <d(A,N) <d(A,P) + margin

- **Hard Triplets:** The distance between the embeddings of anchor and negative faces is less than the distance between the embeddings of anchor and positive faces plus some margin. Hence, the loss propagated here is backwards and thus always positive.

  d(A,N) <d(A,P) + margin

Any model can be trained using triplet loss by using either offline or offline triplet mining procedure. In Offline Triplet Mining, the triplets are first generated manually and then the data is fit to the network. In Online Triplet Mining, training data is provided in batches and then triplets are generated using all the examples in the batch and finally the loss is calculated over it. This approach allows us to randomize the triplets and increase the chance to find triplets with high losses and further will help in training the model faster.

Chapter 4

Working model

4.1 Model training

The dataset that we have contains 8000 deep fake videos and 2000 original videos. From each video 20 frames are extracted randomly and faces are extracted from all of those. Now, each face is passed through a facenet model to get 512 vector embeddings for each of the images, that is original and deep fakes. Now, a network architecture is defined, with an Input layer of the shape (m, 512) and an Output layer of size (m, 64), representing the final embeddings.

We then define the model in such a way that the Triplet Loss function receives all the embeddings from each batch and along with it all the labels are also provided to the batch. This implementation will be shown in the model on the next page. For this we have to define an input layer for the labels and then it is concatenated to the embeddings.

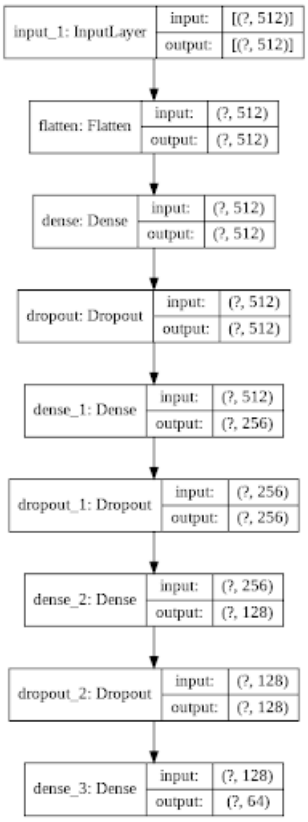

Figure 4.1: Initial part of DL model trained over triplet loss

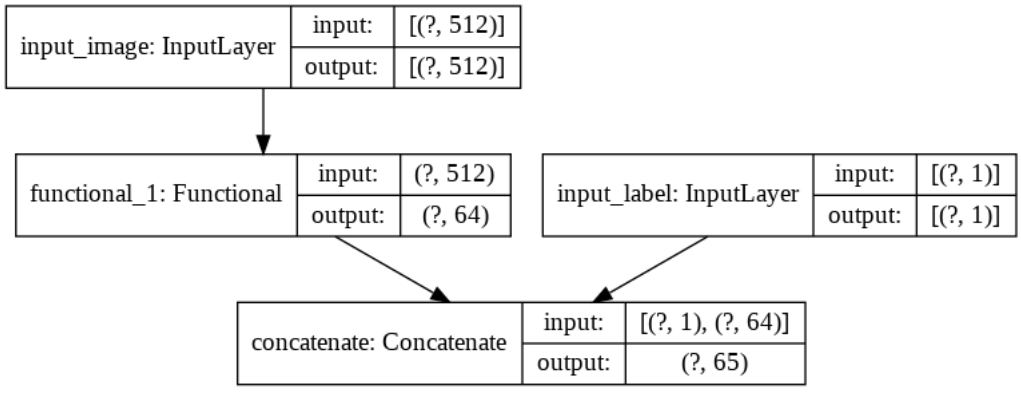

Figure 4.2: Final part of DL model trained over triplet loss

4.2 ML classifier

After the model is trained we get the desired weights with a 64 vector embeddings in the final layer. Now, all the different ML classifiers can be applied to it like logistic regression, decision tree, naive bayes, random forest, K nearest neighbour. I have tried to get the results both with 512 vector embeddings and 64 vector embeddings. Their comparison can be seen in the next section. 512 embeddings are the ones directly obtained from the facenet model while 64 are the ones trained over triplet loss.

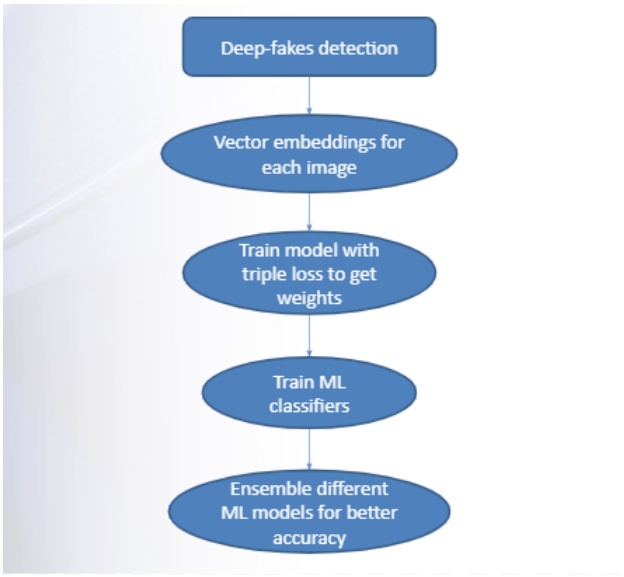

Figure 4.3: The complete architecture used

Chapter 5

Experimental Results

All the results are based on the test dataset separated from the original FaceForensics dataset which includes 10% in quantity.

Ensembling played a major role in increasing accuracy. Basically, we use different models together and the output label is chosen that has most counts from all the models. This technique is widely used everywhere and helps in increasing accuracy. The boost in the accuracy is between 2%-3%. Extracting a number of frames from the videos also played a role in improving accuracy. Initially 1 frame from each video was analyzed. This leads us to data imbalancing as deep fake videos were in large numbers. Finally, using 20 frames from deep fake videos and 50 frames from original videos were taken in count that gave us desired results.

Training ML classifiers directly for 512 vector embeddings could achieve overall 72% max accuracy. In general tree based models did not work here as they require distinct features which cannot be obtained here while simple ML models gave more preferable results. Highest acc is achieved when we

ensemble all models together and their comparison can be seen below:

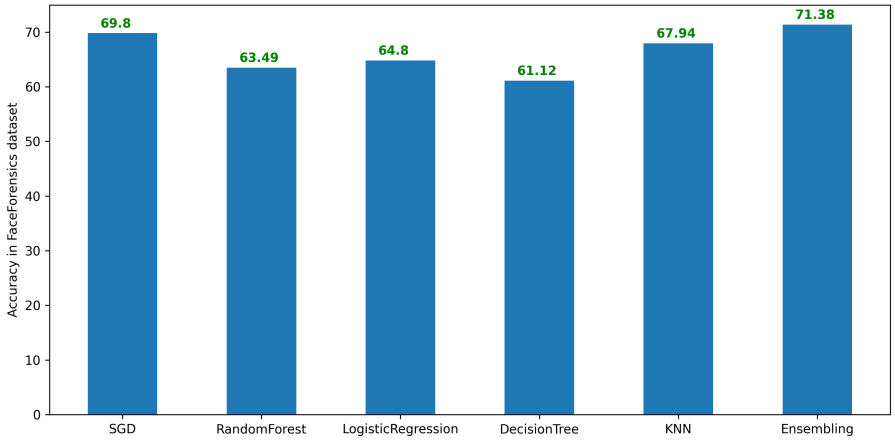

Figure 5.1: Comparison of ML models over untrained 512 vector embeddings

Now, the results are for 64 vector embeddings trained with triplet loss and a different neural network. The max accuracy achieved is 85%. The final comparison can be seen below:

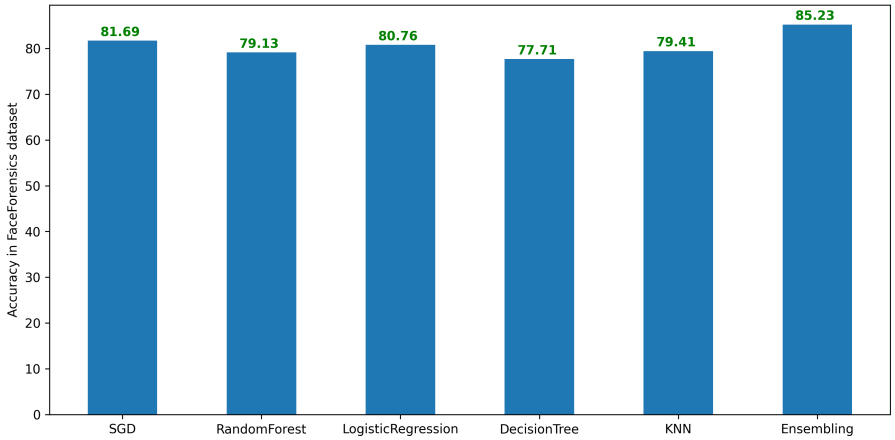

Figure 5.2: Comparison of ML models over trained 64 vector embeddings

The source code for all the experimental results shown above is present at: https://github.com/AshishGusain17/deep-fakes-detection

Conclusion

For the sole purpose of detecting deep fakes, I have tried multiple models and architectures. Many failed to provide results, even using CNN models directly gave a big blow as they just hung the accuracy at 50%. Finally, after trying different models, the one that worked was triplet loss over face embeddings. Every model and approach has its own pros and cons. For my model, it's pros are the implementation part and the cons include versatility in the dataset. The model could detect only deepfakes for only those techniques with which the dataset was generated like swapping face, neural textures. If in future a new technique is found for generating deep fakes, this model may not work for those particular cases. This possibility is not only for my model but all the models available in the market.

Bibliography

[1] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and Alberto Del Bimbo. Deepfake video detection through optical flow based cnn. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[2] Nicolò Bonettini, Edoardo Daniele Cannas, Sara Mandelli, Luca Bondi, Paolo Bestagini, and Stefano Tubaro. Video face manipulation detection through ensemble of cnns. *arXiv preprint arXiv:2004.07676*, 2020.

[3] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv preprint arXiv:1812.02510*, 2018.

[4] Alkazhami Emir. Facial identity embeddings for deepfake detection in videos, 2020.

[5] Disheng Feng, Xuequan Lu, and Xufeng Lin. Deep detection for face manipulation. *arXiv preprint arXiv:2009.05934*, 2020.

[6] Ali Khodabakhsh and Hugo Loiselle. Action-independent generalized behavioral identity descriptors for look-alike recognition in videos. In *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6. IEEE, 2020.

[7] Akash Kumar, Arnav Bhavsar, and Rajesh Verma. Detecting deepfakes with metric learning. In *2020 8th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2020.

[8] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A new dataset for deepfake forensics. *arXiv preprint arXiv:1909.12962*, 2019.

[9] Raju Nekadi. Siamese network-based multi-modal deepfake detection. 2020.

[10] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.