**Weekly Report**

**Course Initial and name: Senior Project Design II ( 499B)**

**Project Name: Deepfake video detection**



Project Supervisor: Md. Ashrafuzzaman Khan (Azk)

Section: 07

Group Members:

Shayekh Mohiuddin Ahmed Navid-153124702

[Shamima Haque Priya-1620069042](https://github.com/fatema13/CSE499-04-FakeVideoDetection/tree/master/Shamima%20Haque%20Priya-1620069042)

[Umme Kulsum Ritu- 1511619042](https://github.com/fatema13/CSE499-04-FakeVideoDetection/tree/master/Umme%20Kulsum%20Ritu-%201511619042)

[Fatema Tuz Zohra- 1620191042](https://github.com/fatema13/CSE499-04-FakeVideoDetection/tree/master/Fatema%20Tuz%20Zohra-%201620191042)

Department of Electrical & Computer Engineering

North South University

Timeline:

Week 1 (21/1/2020-25/01/2020)

During our semester break, we decided to train our model with various architecture. The list of architectures included inceptionv3,ResNet50 and VGG-19. We also divided the dataset of the FaceForensics++ into the ratio 8:1:1. However, we were only able to train our model using 20% of the existing dataset. But we received an accuracy of 79% for inceptionv3, 78.9% for ResNet50 and 97.2% for VGG19 architecture.

After the semester started, we decided to participate in the Deep Fake detection challenge organized by facebook. We downloaded the first set of dataset from the competition. In order to train our model to conduct operations on the dataset, we needed to set up the environment for our data to work. After we set up the environment for our model to work, we decided to download a few more datasets from the competition. Conclusively, we downloaded in total, 4 out of 50 datasets from the competition. The task for the deep fake detection challenge was first to understand whether the model would be able to understand whether the provided data is fake or real. We implemented an algorithm in our code that can help us to extract the labels from the metadata.json file that was included with each downloaded datasets. Fortunately, frames were extracted in accordance with those labels and we put them in their designated folders: Real or fake.

For every image that was extracted, we used openCV to extract only one specific part : the facial part of the provided images.

On the following days, we extracted all the frames and faces from the datasets downloaded from deep fake detection challenge. For the next dataset, we used the architecture ResNet50 to train our model. The accuracy of the training session with the ResNet50 architecture was achieved to be 87%. However, for our testing session, our dataset lacked adequate amounts of data. During the training session, the metadata.json file was used to identify fake and real images.

Week 2 (26/01/2020-31/01/202)

After we worked on the previous datasets, we decided to download a few more datasets. Overall, we managed to download 10 datasets from the deep fake detection challenge organized by facebook. We divided the datasets into two sections. 9 datasets for training and 1 dataset for validation. We decided to train our remaining datasets with the ResNet50 and InceptionNet architecture. For ResNet50 the accuracy achieved was 50% while for InceptionNet , the achieved accuracy was 87%. However, for ResNet50 architecture, the accuracy began to eventually decrease.

Week 3 (02/02/2020-06/02/2020)

After much changes with different architectures, we decided to train our model using InceptionResnet architecture for our image extraction process. We used the concept of fine tuning and adjusted our parameters very precisely where we keenly observed the changes very precisely. We also achieved an accuracy of 89.7%.

Week 4 (9/02/2020-13/02/2020)

We submitted our score in the deep fake detection challenge and settled down with a rank. This week, we trained our model using InceptionResnet on our first trial. During our first trial, we trained our model with fully connected layers. For our second trial, we trained our model with a GAP layer. Our model during the training session with a GAP layer, provided better results with an accuracy of 87%.

Week 5 (16/02/2020-20/02/2020)

We reviewed few papers from the deepfake detection challenge competition to find solutions on how to improve the accuracy by changing the parameters of our model. One of the most efficient way to see an increase in the accuracy is to train the model using different architectures. This time we decided to train our model using VGG Face. The implementation included class weight techniques from Keras. Basically, the concept of transfer learning is applied here and pre-trained weights of VGG Face model is used. The model is consumed now as an auto-encoder which will represent images as vectors. In the real class category, we had approximately 6000 data and in the fake class category, we had 16000 data. Our model was biased towards the fake class as the amount of data for fake class category was greater than the real class category. Thus class weights function in keras was used and the model was trained sequentially, first with real frames and later with fake frames. However, only this time, each real instance was used 2.5 times more in order to balance the fake class. The validation accuracy however still didn’t increase, rather the validation loss decreased from approximately 0.032 to 4.

On 19th February, we tried to implement grid search. However, the keras API was acted as a barrier since the image generator only returns only a batch of X and Y’s and not the entire dataset on which the grid search should work and the result would be achieved.