DEEPFAKE DETECTION

Sarah Zhou
shi.q.zhou@mail.mcgill.ca

Problem Statement & Background

This is a current real-world problem where the objective is to accurately determine if a video contains deepfake content. Research papers have shown that human behaviours such as the natural blinking pattern of the eye is hardly mimicable by deepfake videos. Theoretical models have been discussed namely involving a mixture of LRCN and LSTM. However, the challenge remains to implement the detection of these faults in a machine learning model with over 50% accuracy [3].

Data

The large dataset of 400 videos of a total of 470GB is provided by the Kaggle challenge [1]. The label per video is REAL vs FAKE, used for binary classification and feedback. Two pretrained models were considered during this project. One was pretrained on the FaceForensics++ dataset and the other, on the Facenet dataset.

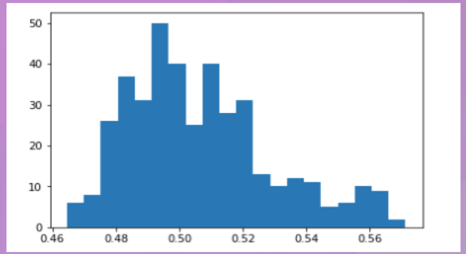

Figure 1.0 Final distribution on the test set (public validation set)

Methodology 1

The first pretrained facial detection model chosen is MTCNN and Inception Resnet models. Using Facenet by Pytorch, the face feature vector were calculated for all the face in each video. The distance from each face to the centroid for its video was then calculated and acted as means of discrimination [4].

Results 1

From figure 1.0, we see that the model performed poorly for a binary classification problem. The clusters are continuous and there is a lack of separation of the output. The small rage of 0.46 to 0.56 cannot be used to distinctly. Although a face detection pipeline was successfully created, it was not effective.

Methodology 2

The second model was released by **FaceForensics++** used **dlib package** and had a significantly better jump off point. It uses 3 types of image preprocessing and applied many models namely the Xception model, a 71-layer CNN pretrained on more than a million images from the ImageNet. The comparison of training results has shown that **the Face_detection Xception model with all compressed 23 images performed best** [2].

Results 2

The predictions were run on 50 frames at a time and were averaged into a single value. 0.5 is the value returned when unable to predict. For each video, we return the maximum, minimum and average all frames of "fake" prediction [2].

In the **final training predictions**, the model has a somewhat clear binary partition, with almost no confusion on real videos.

In the **final test predictions**, the model was confused about 80 videos.

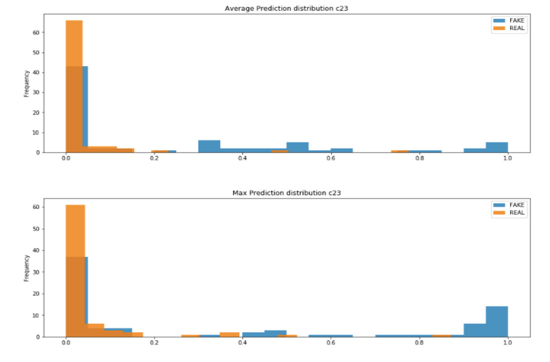

Figure 2.0 Final test predictions

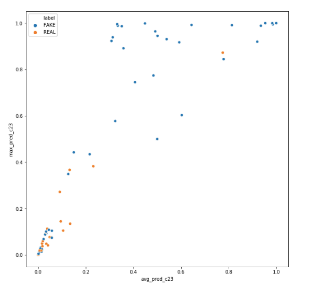

Figure 3.0 Plot the Average vs Max Prediction Probability - Fake vs Real

Conclusion

The rest of the videos follows a somewhat normal distribution.

Many combinations of CNNs and LSTM should be tried out and further tuned to solve this problem.

This challenge certainly holds high value in terms of application in real world products. A successful model can be commercialized to help the public detect fake news, help in cyber security, assist the police for criminal cases and serve as an educational tool.

References

[1] DFDC (2019). Deepfake Detection Challenge. [online] Kaggle.com. Available at: https://www.kaggle.com/c/deepfake-detection-challenge/ [Accessed 31 Jan. 2020]

[2] Mulla, Rob. (2020). FaceForensics++ Baseline (dlib & no internet). [online] Kaggle.com. Available at: https://www.kaggle.com/robikscube/faceforensics-baseline-dlib-no-internet/ [Accessed 28 Feb. 2020]

[3] Thanh Thi Nguyen *et al.* (2019)
Deep Learning for Deepfakes Creation and Detection. https://arxiv.org/pdf/1909.11573.pdf [Accessed 31 Jan. 2020]

[4] Timesler (2020). Facial recognition model in pytorch. [online] Kaggle.com. Available at: https://www.kaggle.com/timesler/facial-recognition-model-in-pytorch [Accessed 31 Jan. 2020]