Deepfake Video Detection

Aleksander Dash, Nolan Handali

{adash. nolanh}@stanford.edu

Youtube link: https://youtu.be/XhxGPhLxSH8

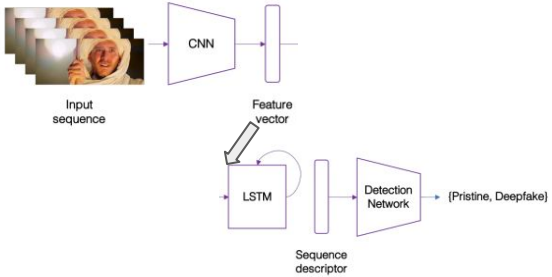
Introduction

Deepfake detection is becoming a much more popular topic among today's computer vision world. Deepfakes refer to when a performance by an actor is superimposed onto a photo or video of a target person to make it appear like the target is performing the actions that the actor is doing. The creation of deepfakes has been enabled by recent AI/ML advances, and modern deepfakes are virtually imperceptible from real people to human eyes. This technology is devastating to people targeted by them, as politicians can be made to give speeches they never would have, archive footage can be doctored, or celebrities can be superimposed onto pornographic footage. In this project, we experiment with using a convolutional LSTM architecture to detect faked videos.

Data

The dataset comes from Kaggle, which is currently running a competition where they provide a 500GB dataset of 1920x1080 30fps 10-second colored real and deepfake videos. The original dataset is heavily imbalanced and too large, so we created a 60 GB balanced subset with 4800 videos in the training set. The validation/test sets both consist of 480 videos, 240 from each class.

Model Architecture



Methods

Baseline:

1. Pretrained VGG-16 weights to extract 512 dimensional embeddings
2. 4 Layer feed forward net to predict fake or not

(The first baseline trains on a single frame per video, the second baseline trains on 50 frames per video without incorporating temporal information)

Face Detection Layer:

1. Select a subset of 10 frames per input video file (one frame per second)
2. Perform facial detection on the frame using Haar Cascades
3. Use the following heuristic to incorporate temporal information in tracking of faces across the frames, where (p_x, p_y) is the center of the bounding box identified in the previous frame, and (cx_i, cy_i, conf_i) is the center of the bounding box and associated confidence for each face detected in the current frame:

$$\text{score}_i = \frac{\sqrt{(p_x - cx_i)^2 + (p_y - cy_i)^2}}{\text{conf}_i}$$

4. Crop input frames around detected faces, rescale to 299x299 pixels

CNN Architecture:

1. This portion consists of a Inception-v3 model with weights pretrained on Imagenet
2. Removed final linear layer and froze weights to extract 2048 dimensional embedding from facial features

RNN Layer:

1. Passed the dimensional embedding from Inception-v3 into a LSTM with hidden size 512

Linear Layer:

1. Final output from LSTM is passed into a linear layer, which outputs the predicted class

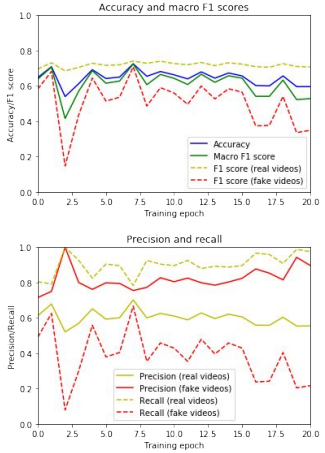
Training Details:

Used Binary Cross Entropy Loss, Adam optimizer with lr = 0.0005
Training was done on AWS p2.xlarge instance

References

Lee C. Hsu, C.; Zhuang. Deep fake image detection based on pairwise learning.preprints,2019.

Results



Conclusion

The model trains well until epoch 7, after which it starts to overfit to the training set. The model also predicts a balanced amount of real and fake videos on the validation set for this epoch -- 268 real vs 212 fake predictions.

After epoch 7, the model overfits and begins to overwhelmingly predict real videos on the validation set -- by the end of training it predicts 422 real videos and 58 fake videos, which can be seen as the model's recall on the fake class of videos in the validation set decreases sharply.

Inthe future, we would experiment with other methods of temporal feature extraction, such as a 4d convolutions. Additionally, we wanted to experiment with using a GAN to create deepfakes, and then using the discriminator model to distinguish between real or fake videos, but did not have time to explore this idea. Additionally, given more time and compute, we would train on the whole dataset.