CS205: Final Report

Real or Fake Face?

Team Member: Abdullah Chaudhry, Adhith Karthikeyan, Chun Ho Wong

Abstract/Motivation

In recent weeks, our team presented a paper presentation study on the video generation AI, "Make a Video," showcasing its capabilities and limitations. The technology demonstrated promising results, however it is still easy to tell if something is generated by AI because of the model capabilities at the time, however, the recent debut of OpenAI's SORA has sparked a new wave of concern among content creators, particularly YouTubers, about the potential for job displacement due to AI advancements. This not only leads to fake news but can become much more convincing simply because it looks realistic.

Naturally (after taking this class), we wanted to make a tool that can distinguish between real and fake faces. It's important to get it done quickly because generative AI is only going to get better over time. Our project ties into AI ethics and how AI should be used safely.

Dataset

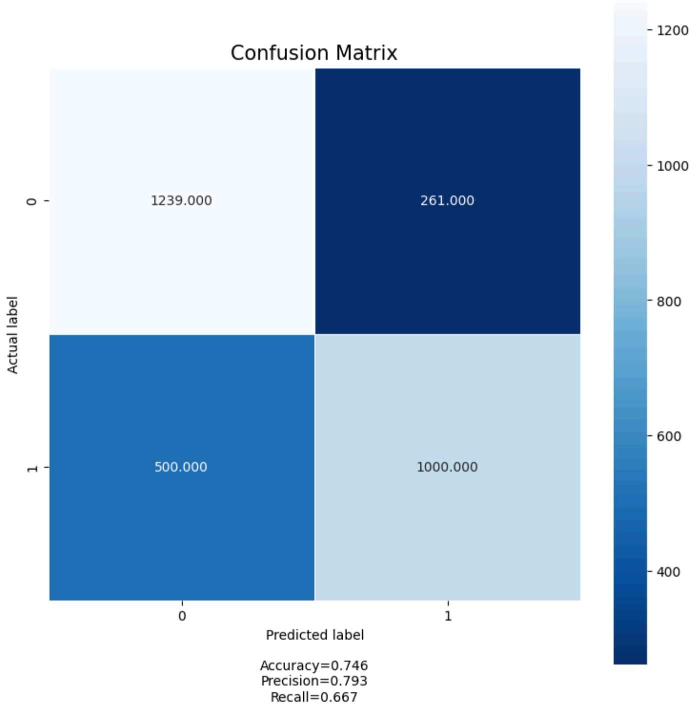
We used a publicly available dataset found on Kaggle. It's around 300 megabytes in size, which is large enough to give us meaningful results but it's also reasonable to run on machines with less storage. While this dataset is image based, we can later extend the same model on video by capturing faces on videos and converting it to images.

Structure of the Model

We decided to implement a convolutional neural network for this project, as it seemed to lend itself well to a classification problem such as this. It uses the adam optimizer, as SGD was reporting NaN loss values at each epoch (likely a result of exploding gradients) and binary cross entropy. Our model size was largely constrained by Colab GPU usage, although preliminary testing using larger models on local notebooks didn’t seem to improve performance substantially enough to outweigh runtime concerns.

The metric used for evaluation was primarily accuracy, although precision, recall, and f1-score were also observed to ensure that none of these metrics were falling significantly behind the others.

Evaluation



We tracked accuracy, precision, and recall as previously mentioned, and created a confusion matrix to more accurately determine where false positives, false negatives, etc. were occurring. The performance of the system, while competent, absolutely has room for future improvements should we choose to iterate on it.

Potential Improvements

We initially wanted to implement a logistic regression model as well as the CNN so we had a baseline to compare to, but decided against it because we were consistently exceeding Colab's GPU usage restrictions – this would absolutely be a nice feature to implement in the future. Early stopping is another consideration to make, but due to time constraints we simply decided to manually tune the epoch number in an attempt to avoid overfitting. Finally, k-fold validation is something else we can consider for more rigorous validation in the future.

Conclusion

Overall, while the problem itself turned out to be a little more difficult than we initially anticipated, we had a lot of fun working on the project and do not regret choosing the initial problem at all. We believe that we have some good fundamental groundwork that can be iterated upon to create a surprisingly competent model in the future.