The Role of Facial Features and Mannerisms in Detecting Deepfakes

Saanvi Bhargava

***Abstract*— This project seeks to develop a machine learning model to identify deepfakes to prevent the spread of misinformation in this era of technology. Politicians and celebrities are the most affected by deepfakes, since fake videos could endanger their reputation and their careers. Most of the current approaches attempt to create a single model across different videos and using that for detection, which does not yield very accurate results. This study focuses on deepfakes with a single face and attempts to use facial feature extraction for detection of deepfakes. I propose a novel approach of using facial features such as facial landmark detection, head pose estimation, facial action unit recognition, and eye-gaze estimation for classification. I conducted 10 different experiments building models for detection using classification algorithms and concluded that 9 of them had an accuracy higher than 95% using the facial feature extraction approach (using OpenFace2). The key finding of this research is that features extracted using the Openface2 library are extremely effective signals for classification of deepfakes involving a single face.**

I. Introduction

Deepfakes are a type of synthetic media where one's face is replaced with another's. They can be used during elections to spread fake news, or for bullying. The British Broadcasting Company (BBC) created a deepfake of Queen Elizabeth's Christmas speech to raise public awareness about deepfakes. As technology evolves, deepfakes will become more and more realistic and can be used in harmful ways. Research communities and universities have been working on many different approaches for detecting deepfakes using different methods. Using extracted facial features for classification is a better method for detection of videos with a single celebrity. Each person has unique characteristic like head tilting, eye movements, lip movements etc. which can be used for distinguishing deepfakes. This approach is effective for detecting deepfakes of identifiable celebrities as it is easy to find real videos of these figures, which can then be used to train the model and then sort the test video as a deepfake or real.

II. Methods

Data Collection: This study uses three different datasets – 1. Three deepfakes of celebrities that have been released in the public domain over the last few years, 2. Celeb-DF dataset, and 3. Facebook Deepfake Detection Challenge (DFDC) dataset.

The three popular deepfakes used included, one each of Donald Trump, Barack Obama, and Queen Elizabeth. Using deepfakes that have circled around social media is a great way to confirm that the algorithm works on real world celebrity deepfakes.

The Celeb-DF dataset contains both higher and lower quality dataset. The high-quality dataset was used for training and testing the model. The Celeb-DF dataset consisted of 590 real videos of celebrities, taken from YouTube, and 5639 deepfake videos of those celebrities. The folder structure consisted of two folders, one of deepfakes, and one of real videos of the celebrities. Since the folder structure was not ideal for the training process, the files were sorted so that each celebrity's real and fake videos were in a folder unique to the celebrity. Then, verification was run to ensure each celebrity folder had both a real and fake dataset.

The DFDC dataset is very large, and this research only used the preview dataset. The preview dataset contains approximately 5300 videos (real and deepfake) using two different manipulation algorithms. The tags (deepfake/real) for each video are in a json file.

Feature Extraction: In order to create the data points for training the model, the Facial Feature Extraction analysis task from OpenFace2 was used. OpenFace2[0] is a widely used facial behavior analysis toolkit. The feature extraction task was run on all videos. This produced a ".csv" file with more than 170 facial features for each frame in the video, such as facial landmark detection, head pose estimation, facial action unit recognition, and eye-gaze estimation. Each row of the ".csv" corresponds to a frame in the video, and each column contains value for each feature extracted.

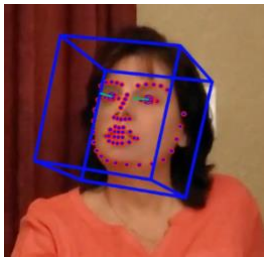

Figure 1: Facial Feature Extraction

Model Training: I conducted 10 different experiments across all the datasets. Table 1: Classification Experiments shows all the experiments conducted and their scope.

Experiment ID	Dataset	Algorithm	Scope
1	Celeb-DF	KNN (n=3)	Model per face
2	Celeb-DF	KNN (n=5)	Model per face
3	Celeb-DF	KNN (n=15)	Model per face
4	Celeb-DF	XGBoost	Model per face
5	Celeb-DF	SVM	Model per face
6	Celeb-DF	SGD	Model per face

7	Celeb-DF	KNN (n=5)	Single Model, Seen faces (#faces=5)
8	DFDC	KNN (n=15)	Single Model, Seen faces
9	3 Celebs (public domain)	KNN (n=5)	Model per face
10	DFDC	KNN (n=15)	Single Model, Unseen Faces

Table 1: Classification Experiments

Each experiment had three variables – Dataset, Algorithm and Scope. Various different classification algorithms were tested with mostly default parameters (unless specified). Scope was one of the following: Model per face, Single Model Seen Faces and Single Model Unseen faces. *Model per face* – In this scope, I trained a model for each face id. *Single Model Seen Faces* - Single model for all faces in the training set, and the test set included the same faces. *Single Model Unseen Faces* - Single model for all the faces in training set, and the test set included faces not in the training set.

I tested five different classification algorithms using the Python scikit machine learning library, KNN (K-Nearest Neighbors), XGBoost, SVM (Support Vector Machine), and SGD (Stochastic Gradient Descent). I tested the KNN algorithm with 3 different parameters for nearest neighbors. I used the default parameters (unless specified) for all algorithms.

III. Results

The goal of this study was to measure the effectiveness of using facial features from videos for detection across different classification algorithms and scope. To test the effectiveness of these models, I collected four different scores - accuracy, precision, recall, and F1, shown in Table 2: Experiment Results. The best metric for effectiveness is accuracy, and nine out of ten of the algorithms had over 95% accuracy.

Experiment ID	Accuracy	Precision	Recall	F1
1	1.000	1.000	1.000	1.000
2	1.000	1.000	0.999	0.999
3	0.999	0.997	0.994	0.996
4	0.999	1.000	0.995	0.994
5	0.990	0.964	0.925	0.943
6	0.978	0.910	0.834	0.868
7	0.996	0.985	0.973	0.979
8	0.953	0.980	0.799	0.880
9	1.000	1.000	1.000	1.000
10				

Table 2: Experiment Results

The results were very close for different scopes on those nine experiments including model per face and single model for all faces. The variation of accuracy across different algorithms with different datasets is very minimal. This suggests that facial features are a highly effective signal for classifying deepfakes and can be used along with any of the classification algorithms for deepfake detection.

IV. Future Research

As discussed earlier in the results section, nine of the experiments were tested on a face which was already part of the training set, and that yielded high accuracy and conclusive results. The tenth experiment, where the scope was *Single Model With Unseen Faces,* did not yield conclusive results.

Since the accuracy scores had a lot of variation across videos, I analyzed the data manually for many videos in the test set. Many of the real videos were predicted with a low accuracy, while the fake videos had a generally high accuracy. This suggests that the model was predicting most of the videos to be fake.

One of the areas that needs further investigation is to try and use a unary classification algorithm along with facial features to try and detect deepfakes. Also I believe that the approach taken so far may not be very effective when we encounter a new face so the facial features may need to be augmented with some other features from the video for training.

Acknowledgment

Put sponsor acknowledgments in the footnote. Upload your files and submit the form by January 15, 2019. Thank you for submitting to the 2019 Columbia Junior Science Journal.

References

[1] Baltrusaitis, Tadas, et al. OpenFace 2.0: Facial Behavior Analysis Toolkit. IEEE, 2018, par.nsf.gov/servlets/purl/10099458.

[2] Harrison, Onel. “Machine Learning Basics with the k-Nearest Neighbors Algorithm.” Medium, Towards Data Science, 14 July 2019, towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761.

[3] Jung, Tackhyun, et al. “DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern.” IEEE Access, IEEE Access, 20 Apr. 2020, ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9072088.

[4] Nguyen, Thanh Thi, et al. “Deep Learning for Deepfakes Creation and Detection.” Arxiv, 25 Sept. 2019, arxiv.org/pdf/1909.11573v1.pdf.

[5] Lyu, Siwei. “Deepfakes and the New AI-Generated Fake Media Creation-Detection Arms Race.” Scientific American, 20 July 2020, www.scientificamerican.com/article/detecting-deepfakes1.

[6] Dolhansky, Brian, et al. “Deepfake Detection Challenge Dataset.” Ai.facebook.com, 2019, ai.facebook.com/datasets/dfdc. Accessed 29 Aug. 2021.

[7] https://openaccess.thecvf.com/content_CVPRW_2019/papers/Media%20Forensics/Agarwal_Protecting_World_Leaders_Against_Deep_Fakes_CVPRW_2019_paper.pdf