Deep Fakes Detection

Akul S. Malhotra[1] and Sana Naz[1]

[1]American University, College of Arts of Sciences, Washington, D.C., USA

Introduction

Deep fakes leverage powerful techniques from machine learning and AI to manipulate or generate visual and audio content with a high potential to deceive. Large amount of readily accessible data and advanced computational power has made it easier to generate Deep fakes.

The motivation for this project is to build and evaluate an algorithm that detects if a particular media is fake or real. Accurate and efficient detection technology will prevent:

- Hampering societal peace
- Attacks on national democratic structure
- Prevent national and global security threat
- Stop the spread of fake news and its effects

Contribution

We extract features from and build soft biometric models of individual's faces within a dataset of original and manipulated videos. Then, we evaluate the deep fake detection models in their ability to distinguish between real and fake videos and show the efficacy of this approach on a large amount of data.

Classification Models

Logistic Regression

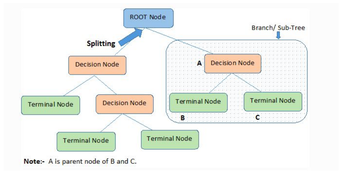
This is used to predict the binomial outcome of a response variable using one or several predictor variables. The predictors can be binomial, categorical, or numerical. It is a way to map a continuous function of predictors to the probability of a binary response from 0 to 1.

$$ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x$$

$$=> P = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

Decision Trees

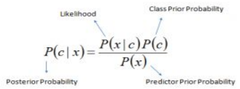
The decision tree for a classification task is built through binary recursive partitioning- starting at the tree root and splitting the data into partitions on the feature that results in the largest information gain. Then, splitting it further on each of the nodes until the leaves are pure.



Naive Bayes

This classifier is based on the Bayes theorem of probability to predict the class of the test dataset. It makes an assumption of independence amongst the predictors.

Bayes theorem provides a way of calculating posterior probability P(c/x) from P(c), P(x), and P(x/c):

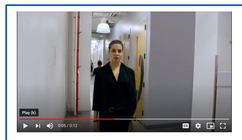


$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c)$$

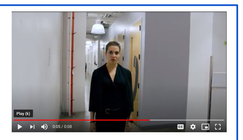
Data Extraction Using OpenFace2

OpenFace2 is a forensic technique that is designed to detect deep fakes of an individual based on the principle that that when they speak, they exhibit relatively distinct patterns of facial and head movements that are disrupted by all 3 types of Deep fakes.

OpenFace2 is a open-source facial behavior analysis toolkit. All shortlisted videos contain 1 primary and camera-facing person in the frame.



Original Sample



Manipulated Sample

We fed a sample of 150 original and manipulated videos to OpenFace2 and extract upper-body movements for speaker identification.

The output for each video are several features representing movements in each frame. Measurements like the following are later transformed into features:

- 2D and 3D facial landmark positions
- Head pose
- Eye gaze
- Facial action units (16)

Data Organization & Feature Description

Each observation matrix $X^{[k]}$ contains feature vectors which correspond to the movements of facial muscles commonly termed as Action Units (AU). In other words, each video contains frames at which certain facial expression were captured at a particular timestamp.

A total of 714 features were captured, for each video, using feature extraction methodology provided by OpenFace2.

Below is the hyperlink to follow for detailed information on the features metadata information:

https://github.com/TadasBaltrusaitis/OpenFace/wiki/Output-Format

Initial data exploration led us to incorporate a concise version of this dataset having 19 features.

Below is a snippet of cleaned data:

	AU01_r	AU02_r	AU04_r	AU05_r	AU06_r	AU07_r	AU09_r	AU10_r	AU12_r	AU14_r	AU15_r	AU17_r	AU2(
0	0.12	0.0	0.63	0.34	0.0	0.00	0.0	0.0	0.0	0.0	0.81	0.90	
1	0.12	0.0	0.78	0.39	0.0	0.05	0.0	0.0	0.0	0.0	0.76	0.77	
2	0.17	0.0	0.81	0.37	0.0	0.05	0.0	0.0	0.0	0.0	0.88	0.58	
3	0.35	0.0	0.67	0.37	0.0	0.00	0.0	0.0	0.0	0.0	0.96	0.48	
4	0.66	0.0	0.68	0.42	0.0	0.00	0.0	0.0	0.0	0.0	1.04	0.50	

(875, 19)

We use the Pearson correlation to measure the linearity between these features in order to characterize an individual's motion signature. With a total of 19 facial/head features, we compute the Pearson correlation between all 19 of these features, yielding 171 unique pairs of features across all video clips.

The next step involves creating final dataset for our analysis using these correlation values by taking the upper diagonal of the correlation matrix and vectorizing unique values row wise, belonging to each video together, to form a dataframe.

At this stage, the target variable "Real" was added to identify which observation belonged to a fake video and vice versa. The features in this dataset are correlation values obtained from the correlation matrix for each video. Below is the snippet of the final dataset and its dimensions.

	AU01_r_AU02_r	AU01_r_AU04_r	AU01_r_AU05_r	AU01_r_AU06_r	AU01_r_AU07_r	AU01_r_AU09_r	AU01_r_AU10_r	AU01_r_AU12_r	AU01_r_AU14_r	AU01_r_AU15_r	AU01_r_AU17_r	AU01_r_AU23_c	AU01_r_AU25_c	AU01_r_AU26_c	A pc
0	0.545948	-0.388635	0.290504	-0.354376	-0.358156	0.142932	-0.539013	-0.283119	-0.384306	0.489531	0.192848	-0.039162	-0.063942	0.067019	-0.(
1	0.640464	0.168792	0.450582	-0.378303	0.049258	-0.159566	-0.227057	-0.260812	-0.306965	0.283609	0.026230	-0.225895	-0.091503	0.131829	-0.(
2	0.717926	0.356868	0.437273	0.000000	0.090841	0.148148	0.268729	0.287867	-0.029882	-0.035514	-0.160199	0.000000	0.000000	0.000000	-0.(
3	0.767385	0.133881	-0.176998	0.009181	-0.100936	-0.001857	-0.123262	-0.093450	0.425487	-0.097959	-0.209486	0.522661	0.268355	0.000000	-0.(
4	0.301815	-0.109730	0.001275	-0.047978	0.181038	0.446990	0.660318	0.222923	0.000000	-0.092383	0.041384	-0.052828	0.000000	0.000000	0.(
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
145	0.780693	-0.110567	0.335352	-0.426225	-0.392553	-0.140003	-0.398819	-0.495971	-0.265410	0.151155	0.385311	0.088282	-0.302393	-0.002116	-0.(
146	0.798995	-0.042572	0.299118	-0.275520	-0.159584	-0.051014	-0.182434	-0.316859	-0.314976	0.144033	0.137788	0.209015	-0.170320	0.006466	-0.(
147	0.571866	0.405490	0.212149	-0.754562	0.461715	-0.193776	-0.277363	-0.081254	0.518084	-0.459135	0.292362	0.337531	0.484508	0.464925	0.(
148	0.822920	-0.166823	0.564532	-0.300718	-0.180616	-0.194148	-0.294204	-0.164956	-0.255305	0.086711	-0.007937	0.044818	0.099202	0.351061	-0.(
149	0.936425	0.043154	0.270262	-0.528773	-0.425941	-0.170750	-0.427841	-0.452984	-0.423210	0.285037	0.074550	0.099444	-0.357549	-0.099475	-0.(

150 rows × 685 columns

Logistic Regression - Results

Logistic regression approach provides an accuracy of 50% which is equivalent to an equally likely chance of a model detecting a fake video as real and vice versa. Possible justification behind accuracy being ½ can be high dimensionality. In other words, number of observations (N) are less than the number of features /variables (P) which leads to incorporate less samples and encounter an undersampling issue.

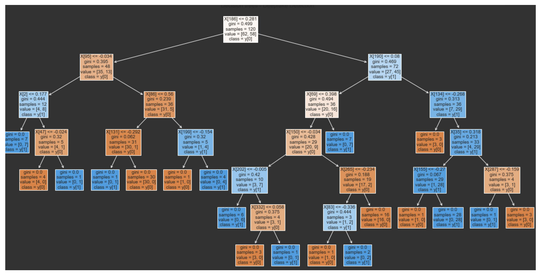
Below is the visual representation of confusion matrix for Logistic Regression.

Support for this work is greatly acknowledged from American University's School of Arts and Sciences (Data Science Department) and Professor Zois Boukouvalas.

1. Protecting World Leaders Against Deep Fakes (Agarwal, Shruti, Farid, Hany, Gu, Yuming, He, Mingming, Nagano, Koki and Li, Hao), In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, IEEE, 2019
2. Nguyen, Thanh & Nguyen, Cuong M. & Nguyen, Tien & Nguyen, Duc & Nahavandi, Saeid. (2019). Deep Learning for Deepfakes Creation and Detection: A Survey.
3. Mal-uses of AI-generated Synthetic Media and Deepfakes: Pragmatic Solutions Discovery Convening, June 2018

Decision Trees - Results

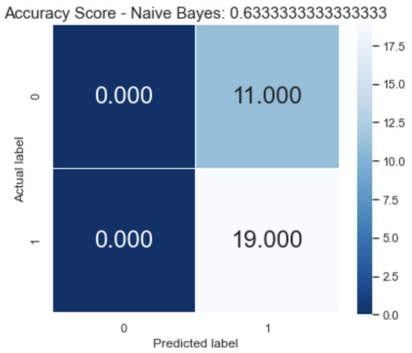
Decision tree algorithm was further utilized to improve accuracy. The percentage increased to 56%. Below is the decision tree created using the gini criterion. Blue box indicates the probability of a video as real and orange box indicates the probability of being fake. Darker shade represents higher probability.



Naive Bayes - Results

The Naive Bayes classification algorithm provides a higher accuracy than decision trees i.e. it provides 63.3% accuracy. A confusion matrix has been designed in order to visualize the reasoning behind its higher accuracy. It is evident from the matrix that the model is able to perfectly classify the real videos. However, it is totally misclassifying the fake videos as real. Potential issue in this case are in line with issues behind poor classification accuracy provided by logistic regression i.e. undersampling and high dimensionality. Some features with zero vectors should be removed for attaining better accuracy.

Below is the visual representation of confusion matrix for Naive Bayes.



Conclusion & Future Direction

The Naive Bayes and Decision Trees framework exploits the relationship between independent correlated facial features and our binary target variable in order to detect deep fakes.

Future work will focus on:

- Reducing dimensionality of the data using PCA and Feature Selection Methods
- Using advanced modeling techniques such as Convolutional Neural Network, 2D CNN/ RNN combined and 3D ConvNet