DeepFake detection in videos

SIV Project Report

Thomas Reolon
thomas.reolon@studenti.unitn.it

Moreno D'Incà
moreno.dinca@studenti.unitn.it

***Abstract* — Nowadays the quality of deep fake videos is improving to the point where they can easily become a threat. Developing techniques to detect fake videos is now an important and open area of research. In this project we explore and develop (with the supervision of instructors) some of these techniques. We exploit OpenFace [1], a face detection tool, to extract meaningful features [2] from videos of a given person. Then we train a pipeline of 15 One Class Classifiers on the real videos of a specific person and test them on real and fake ones. Our dataset is composed by 6 people with a total time of three hours and fifteen minutes of videos. In our setting we reach an accuracy of 89.6% and a F1-score of 94.5% showing good starting results for future developments.**
Code available at github.com/Moreno98/SIV_project.

1 - Introduction

In the era of deep learning, deep fake videos are becoming more and more popular to the point where anyone can manipulate videos of important people in order to make them say whatever they want. Moreover the quality of deep fake videos has improved to the point where it is becoming difficult to determine if a given video is real or fake. These technologies are growing fast and they are beginning to threaten the world, for example we are able to get World Leaders to say whatever we want, this can easily lead to an increase in global tensions. Fake news is another sensible field where fake videos can be even more devastating. Fake news are news created ad-hoc to move the population ideas through a belief rather than another. This field leverages the naivety of the people to manipulate the population, unfortunately it is already a lot spread in our society even without fake videos, so the introduction of these videos can lead to a real and even more powerful threat. For instance, if we wanted to spread a new fake news, who could do it better than Obama?
In this context fake detection in videos is becoming a must in order to stop, or at least mitigate, as soon as possible all these malicious activities. In this project we exploit **OpenFace** [1], a face detection tool, to extract meaningful features from videos. At this point we train One Class Classifiers on the real videos in order to then be able to detect fake videos about a specific person.

Section 2 describes how OpenFace works, Section 3 explains the techniques we adopt to detect deep fakes, Section 4 describes our project's structure, Section 5 summarizes our results.

2 - OpenFace

OpenFace is a tool intended for computer vision and machine learning researchers, affective computing community and people interested in building interactive applications based on facial behavior analysis [1]. This tool is able to analyse both videos and images and it provides a large variety facial analysis tasks, including:

- Facial Landmark Detection
- Facial Landmark and head pose tracking
- Facial Action Unit Recognition
- Gaze tracking
- Facial Feature Extraction

These features are extracted in the following way: a mask (implemented by dlib) is used to crop the region of an image containing a face, then this portion of the image is processed using FaceNet [7] a deep neural network proposed by Google, which is capable of reconstructing a 3D mapping of a face.

In our project we primarily use Facial Feature Extraction in order to extract the facial expressions (Action Units) using the Facial Action Coding System (FACS) which allows openFace to code the facial expressions in a standard and objectively way. OpenFace is able to recognize a subset of Action Units (AU): 1, 2, 4, 5, 6, 7, 9, 10, 12, 14, 15, 17, 20, 23, 25, 26, 28, and 45. Each AU has two scores: presence and intensity, the first one provides if the AU is visible in the face, the second how intense is the AU (minimal to maximal) on a 5 point scale.
OpenFace is also able to track people in a multiple people video however, in our setting, this feature is not used since we are focusing on single person videos, namely the real and fake videos about a specific person, moreover this feature could decrease the quality of the results, as stated in the documentation of the tool.
The outputs from this tool are given via a CSV file containing all the information about the analysis made by OpenFace. We then extract the needed data from this file.
For further information about OpenFace and how it works, please visit OpenFace wiki.

3 - Methodologies

To detect deep fakes we start from the assumption that a person can be characterized by his AUs. More precisely, each individual has his own way to speak and communicate, so we can learn a person's usual face expressions and compare them with the ones extracted from a fake video.
We extract the following features from the videos, as described by Shruti Agarwal et al. [2] paper:

- 16 AUs
- Head rotation (x-axis)
- Head rotation (z-axis)
- 3D Horizontal distance between the corners of the mouth
- 3D Vertical distance between the lower and upper lips

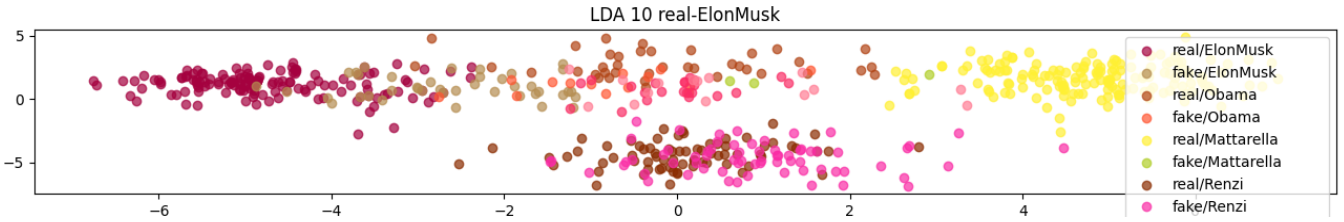

Figure 1: Feature reduction with LDA applied to different people and videos

The first 16 AUs are: inner brow raiser (AU01), outer brow raiser (AU02), brow lowerer (AU04), upper lid raiser (AU05), cheek riser (AU06), lid tightener (AU07), nose wrinkler (AU09), upper lip raiser (AU10), lip corner puller (AU12), dimpler (AU14), lip corner depressor (AU15), chin raiser (AU17), lip stretcher (AU20), lip tightener (AU23), lip part (AU25), jaw drop (AU26) [2].
For the x-axis and z-axis rotations we respectively extract the pose_Rx and pose_Rz features from OpenFace.

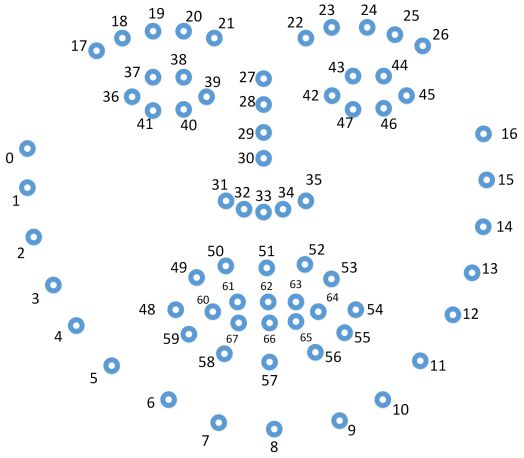

Figure 2: Face features points

In order to extract the last two set of features we exploit the direct feature points shown in Figure 2, the 3D Horizontal distance between the corners of the mouth is selected using the X, Y and Z coordinates of the 48 and 54 points, resulting in the distance:

$$H\_Dist = ||(X\_54, Y\_54, Z\_54) - (X\_48, Y\_48, Z\_48)||$$

Finally the 3D Vertical distance between the lower and upper lips is extracted using the 51 and 57 points:

$$V\_Dist = ||(X\_51, Y\_51, Z\_51) - (X\_57, Y\_57, Z\_57)||$$

Now we have a total of 20 features, but we still need to characterize the individual's motion signature, namely how much a pair of features is correlated, to do so we use the pearson correlation to measure the linearity between the features [2].

The pearson correlation between the features A and B is defined as follow:

$$\rho_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B}$$

Where:

- Cov is the covariance between the two features
- σ is the standard deviation of the feature

With the extracted 20 features, we compute the pearson correlation between each pair of feature, yielding to $\frac{20*19}{2} = 190$ total features [2]. These features are the final one used by our system to detect fake videos.
Given a video we need to choose how many clips we want to process, it allows us to capture small behaviours, while analyzing the video as a whole would smooth these behaviours, that is we would have a set of features characterized by noise introduced by camera changes, angle rotations and all the common adjustments in a video.
We choose to sample the features every 300 frames, so in a video with 30 fps we will sample every 10 seconds.

The fake video detection task can be performed in a variety of ways, we focus on two main methods:

- single model for all the people
- one model for each person

Initially we explore the first setting, here we have multiple videos of multiple people and we try to capture patterns, if there are, on the data about real and fake motions. In this exploration we try the SVM [8] and LDA [9] with different kernels or parameters, training on both real and fake videos. After this study we conclude that this kind of approach is not suitable for our problem, since the data, as shown in Figure 1, are not easily separable with a line leading the classifiers to poor performances.

For this reason we move to the approach proposed in [2] which, instead of training on both real and fake videos, an outlier detector is used (where fake videos are the outliers). This approach allows us to use only real videos to train the model (real videos are easier to find) and produce stronger results, especially when increasing the size of the training set.The classifier adopted is a OneClassSVM implemented by scikit-learn.

To obtain more robust results we also adopted the following techniques:

- changing feature space: 190, 250, 60:
  190 features from the paper [2] + 60 features found by us, which are the mean, std and max of the AUs
- tuning classifiers' hyperparameters:
  we made an exhaustive search of the possible hyperparameters like kernel type, polynomial degree, C, nu
- ensemble learning with 15 OneClassSVMs:
  Adding classifiers with diversity between them can make the predictor more robust and bring significant improvements (as in our case). Diversity between classifiers is granted varying: input feature space, hyperparameters and training set

4 - Project Structure

Code available at: github.com/Moreno98/SIV_project
Demo example available on: google_colab

The project is structured in two main parts: `openfaceapi` and `videoanalizer`, the former is a python wrapper around OpenFace (which was compiled in C) and allows us to easily interact with OpenFace, while the latter is an implementation of the ideas from [2].
`videoanalizer` exposes some functions that can autonomously train a model given a folder containing real videos about a person. For example, calling `videoanalizer.train_OneClassSVM('../folder')` will do the following:

1. check if the videos in the folder have been processed, if not: process them with `openfaceapi`
2. extract the features as described in the previous sections
3. train multiple classifiers (ensemble learning) with different hyperparameters on the features
4. return a classifier that predicts if a video is fake/real based on a weighted sum of the votes of the trained classifiers

You can then use the returned classifier to predict if a video is real or fake calling `clf.predict_video('vid.mp4')` which will return 1 if the video is predicted as real and -1 otherwise. This function has another parameter called landmark_video, if it is set to true `videoanalizer` will create a new .avi of the original video plus a box (red if fake, green if real) around the analyzed person's face (Figure 3 and 4).
`videoanalizer` exposes some other methods like:

- `process_video`:
  extract the feature vectors by calling openfaceapi, splitting videos into clips of 10 seconds and computing the correlation between the action units
- `split_train_test`:
  extract feature vectors and split them so that a video clip can be only in the train or in the test set
- `plot_features`:
  uses feature reduction (PCA or LDA) to plot the clips of the videos in a scatter plot (Figure 1)

For further information about the project structure and how it works, please take a look at the source code here.

5 - Conclusions

We test our system on a dataset of about 1170 video clips of 10 seconds. These clips are extracted from videos downloaded from youtube (for Obama and Elon Musk) and from videos of ourselves.
The average accuracy for detecting deep fakes is varying a lot depending on the analyzed person (Thomas, Moreno, Obama, Musk), but the results always show better than chance classifications, even with just one classifier.
After introducing ensemble learning techniques we reach an average accuracy of 89.6% and a F1-score of 94.5%.
Even if the results we obtained confirmed that this method is really effective, Shruti Agarwal et al. [2] reported an accuracy of 96%. We think we could reach these results with a bigger dataset. Another interesting approach would be to train a model over every video in the dataset and then fine tune it for a specific person in the hope of aggregating the general knowledge about real and fake videos with the specific knowledge about a single individual.

Figure 3: Landmarks generated from a deep fake video

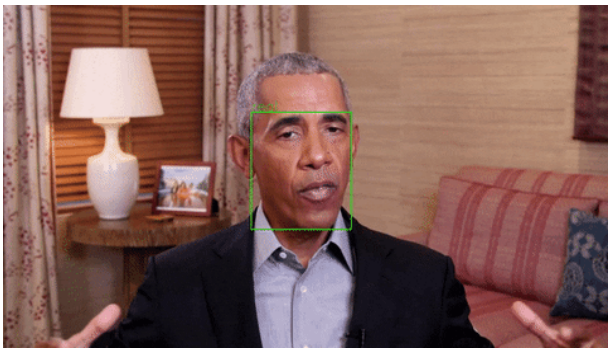

Figure 4: Landmarks generated from a real video

References

[1] OpenFace 2.0: Facial Behavior Analysis Toolkit Tadas Baltrušaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency, IEEE International Conference on Automatic Face and Gesture Recognition, 2018 - GitHub

[2] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting World Leaders Against Deep Fakes" 2019

[3] Convolutional experts constrained local model for facial landmark detection A. Zadeh, T. Baltrušaitis, and Louis-Philippe Morency. Computer Vision and Pattern Recognition Workshops, 2017

[4] Constrained Local Neural Fields for robust facial landmark detection in the wild Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. in IEEE Int. Conference on Computer Vision Workshops, 300 Faces in-the-Wild Challenge, 2013.

[5] Rendering of Eyes for Eye-Shape Registration and Gaze Estimation Erroll Wood, Tadas Baltrušaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling in IEEE International. Conference on Computer Vision (ICCV), 2015

[6] Cross-dataset learning and person-specific normalisation for automatic Action Unit detection Tadas Baltrušaitis, Marwa Mahmoud, and Peter Robinson in Facial Expression Recognition and Analysis Challenge, IEEE International Conference on Automatic Face and Gesture Recognition, 2015

[7] FaceNet: A Unified Embedding for Face Recognition and Clustering, Florian Schroff, Dmitry Kalenichenko, James Philbin, CVPR, 2015

[8] Support Vector Classifier (SVM) SVM - Scikit

[9] Linear Discriminant Analysis (LDA) LDA - Scikit