False Face Detection with Multispectral NIR Image
Udacity Machine Learning Nanodegree Capstone Proposal

Eduardo Carvalho Nunes

February 28, 2020

Domain Background

Biometric System with Facial Recognition is an active topic of research in the last decades [1]. Currently, organizations are planning or using facial recognition systems, some examples: the Chinese government that implemented a facial recognition system on surveillance cameras around the city in Xinjiang [2]; in London at Gatwick airport, it will be the first airport in the United Kingdom to use facial recognition cameras for passenger identification [4] and in Brazil had his first arrest with the help of facial recognition [3].

Biometric Systems that use perform face authentication need fraud detectors more reliable. A system to able to detect this task automatically and correctly brings a number of practical advantages in the field of biometric authentication. For this problem, an anti-spoofing is developed and serves as a pre-step before face recognition. The proposed approach for false face detection is to use NIR infrared camera and machine learning.

Problem Statement

Some facial recognition systems may fail to detect false faces. For example, a test conducted by Consumentenbond (a Dutch non-profit organization that promotes consumer protection) in April 2019 found that of 110 mobile phones, 42 failed to detect false faces. The researchers used photographs of printed faces, masks and 3D heads for the tests [**?**].

Datasets and Inputs

The dataset was built by the author himself. So far, the data set has 15 participants (my friends), between 22 and 30 years old. Each participant has 10 images of real faces and 20 images of fake faces (10 fake digital images and 10 fake images on paper). In total, the data set contains 450 images.The Figure 1 shows an example of a participant where we have 10 images of real faces and 20 fake faces.

I will use the OpenFace library to extract features. According to the website [**?**], OpenFace is a facial recognition library with *deep learning*.

The OpenFace procedure for single input image are:

- Detect face with pre-trained dlib or OpenCV models;
- Cut out the detected face and use it as an entry in the deep neural network;
- Use a *deep learning* to represent the face in a unitary hypersphere of 128 dimensions (characteristics);
- After acquiring the 128 characteristics, apply *cluster* or classification techniques to complete the facial recognition task.

Figure 2 below shows an example using OpenFace, where the input is an image with a face and the output is a vector with 128 values that represent the characteristics of the face.

At the moment we have images of faces from 15 participants. The dataset will have 450 instances, with 300 fake faces and 150 real faces.

Figure 1: Images of real face and fake face of a participant

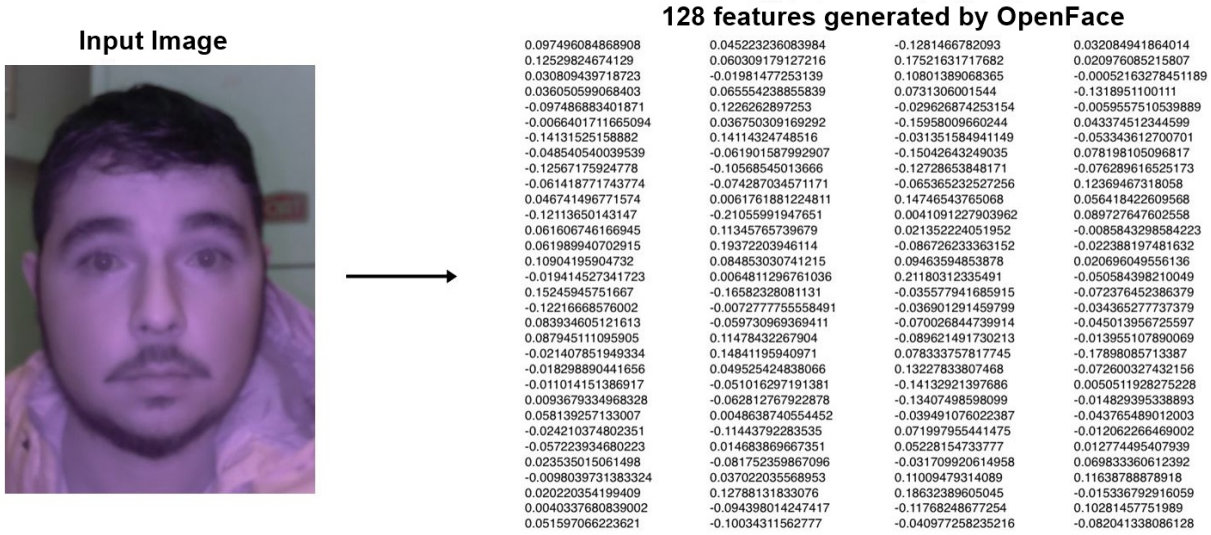

Figure 2: Facial features of a real face using OpenFace

Solution Statement

This is clearly a supervised learning problem because the dataset is labeled (fake face or real face). The goal is to create a model that receives an input face and decide whether the face is fake or not.

Benchmark Model

The classifiers Decision Tree, Random Forest, KNN (K-closest neighbors) and SVM (Support Vector Machine) will be implemented as a reference for this project because all are simple to implement and interpret.

Evaluation Metrics

For evaluation, I will use the confusion matrix. From the confusion matrix, I will use:

- Accuracy: It is the ratio between the subjects correctly labeled. Example: how many real faces do we correctly label from all faces?

- Recall (aka Sensitivity): Corresponding to the performance rate of the positive class Example: of all the faces that are real faces, how many do we predict correctly?
- Specificity: Corresponding to the performance rate in the negative class Example: of all the fake faces, how many of them do we predict correctly?

Project Design

First of all, we need to detect faces using the dlib library. After detecting the faces, the OpenFace library will be used to extract features and then the data will be normalized. The size of the dataset is estimated to be 450 instances with 128 characteristics.

The dataset will be divided into a training and testing set (holdout method). Models: Decision Tree, Random Forest, KNN and SVM will be implemented, trained and tested. A search grid will be made for all models. To measure the performance of each model, accuracy, recall, and specificity will be used.

The best parameters of each model will be tested in the test set. Based on the accuracy, recall, and specificity we can analyze the best classifier to detect fake face or real face.

Figure 3 illustrates the procedure for detecting fake face and real face.

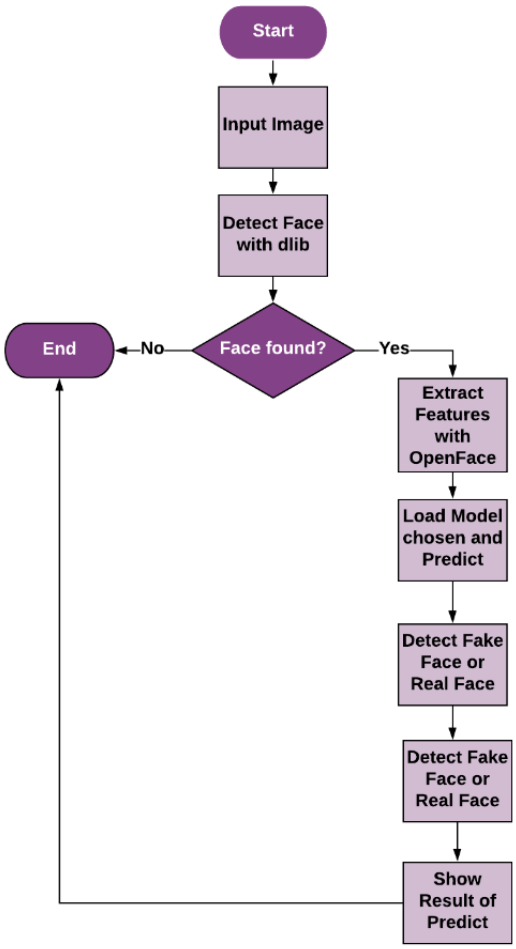

Figure 3: Flowchart for detecting fake face or real face

References

[1] Jain, Anil K., and Stan Z. Li. Handbook of face recognition. Vol. 1. New York: springer, 2011.

[2] China's massive investment in artificial intelligence has an insidious downside. (2020). Retrieved 17 February 2020, from https://www.sciencemag.org/news/2018/02/china-s-massive-investment-artificial-intelligence-has-insidious-downside

[3] Carnival has first arrested via facial recognition camera in Brazil. (2020). Retrieved 17 February 2020, from https://www.tecmundo.com.br/seguranca/139262-carnaval-tem-primeiro-preso-via-camera-reconhecimento-facial-brasil.htm

[4] Gatwick Airport commits to facial recognition tech at boarding. (2020). Retrieved 17 February 2020, from https://www.bbc.com/news/technology-49728301