Facial Detection and Recognition Study for Security Applications

Ali Arefi-Anbarani, Ezgi Gumusbas, Yujin Lee
Master of Applied Data Science

Introduction

Computer vision has many applications from a security standpoint. Facial recognition is one of the most common applications of computer vision. It is widely used for biometric identification. Facial spoofing is a popular risk of facial recognition systems.

Our aim in this project is to make a robust facial recognition system by using state of the art models to achieve high performance and solve the problem of spoofing. Our goal is to answer the question of how we can authenticate a user with high accuracy and detect the spoofing activities.

Data

The dataset contains headshot photos of people in the plain background from Adobe Stock. For ArcFace model, the test dataset contains headshot, half and full body of shot photos of Chinese people in the various backgrounds from V3 dataset.

To see the details:
https://github.com/ezgigm/cyberdata_capstone

Methodology

This project consists of two main parts. First one is anti-spoofing and second part is ArcFace modeling.

For the anti-spoofing application, some images from the dataset were printed using a black and white printer. Data augmentation methods are applied to have a larger dataset and data was divided into train and test set with approximately 80-20 ratio.

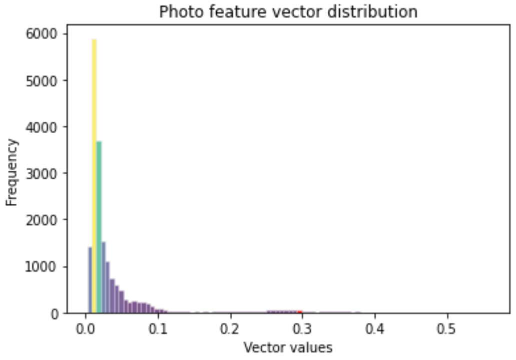
We used local binary patterns(LBP) to differentiate between original photos and printed images. LBP is a texture classifier based on a gray scale co-occurrence matrix that computes a local representation of texture in an image.

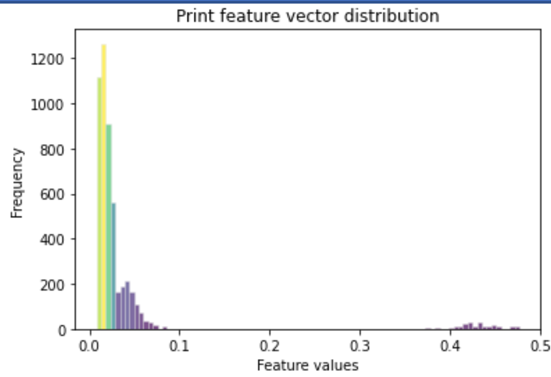
After LBP, we created feature vector plots to see if there is a difference between the distribution of digital photo and the printed(paper).

By using Lazy Classifier, we found the best performer models. After applying, K-fold Linear Discriminant Analysis model was found as a best model for this session.

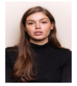
After anti-spoofing part, We used ArcFace modeling to find the similarity between two faces to give an access to the user or not.

Results & Discussion





These figures show how the vector values of the images are distributed by frequency. As it is seen obviously, the original images vector distribution is very different from printed images' vector distribution. It proves that LBP did a good job on texture differences.

	 test_0.png	 test_1.png	 test_420.png	 test_421.png	 test_00001.jpeg	 test_00000.jpg	 test_00002.jpg
test_0.png	0.000000	0.942101	2.349689	1.964111	2.434061	1.742653	1.959101
test_1.png	0.942101	0.000000	2.081174	1.760810	2.193204	2.051017	2.109672
test_420.png	2.349689	2.081174	0.000000	1.036252	2.101191	2.178593	2.250163
test_421.png	1.964111	1.760810	1.036252	0.000000	2.091505	2.250613	2.375853
test_00001.jpeg	2.434061	2.193204	2.101191	2.091505	0.000000	1.423741	1.237057
test_00000.jpg	1.742653	2.051017	2.178593	2.250613	1.423741	0.000000	0.704062
test_00002.jpg	1.959101	2.109672	2.250163	2.375853	1.237057	0.704062	0.000000

In this photo matrix, the green box shows the correct person that matches. 0 (zero) values show the same photo, and as expected in the matrix it is diagonal. The important part is that lower than a certain threshold means it is the same person. So, when the value is decreasing, it means that the similarity between two faces are increasing. Model catches the similarities below a certain threshold. There is one exception here that models find 6th and 7th photos a bit similar also, although they are not the same person. Other than that, when we look at the results row by row, we can see that the lower results point to the same person and model did a good job on this.