Intro

This repository implements EigenFaces paper. This is an algorithm that tries to recognize and detect faces using PCA.

Usage

The `src.py` file contains the class EigenFaces, whose usage is given in `demo.py` over the ATnT face dataset.
Run `demo.py` for a sample run.

Requirements

```
python 3.7.0
opencv-python 3.4.2.16
opencv-contrib-python 3.4.2.16
numpy 1.15.2
scipy 1.1.0
sklearn 0.19.2
```

Testing

- We tested the class that we implemented by only letting the algorithm learn 38 subject images in the dataset above. We then tested the algorithm on a collection of all the 40 subjects in the dataset.

- We set the values of the threshold by looking at the sample values we were getting of the L2 norms of the different differences in weights.
- To test images that are non-faces, we used a random numpy array generated by `np.random.rand` to test for non-face detection by our algorithm.
- We chose the values of the hyperparameters that resulted in a good accuracy over all the accuracy measures, one can obviously change the values of the hyperparameters to get a different distribution of the accuracies according to a particular use case.
- Also we took the `train_test_split` ratio of `80-20`.

Results

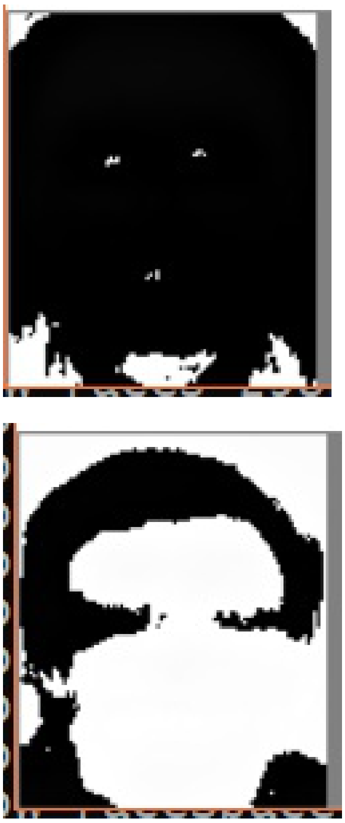
We used the ATnT face data set to train our algorithm and then we tested the implementation using some known faces, some unknown faces, and some non-faces, in each of the case we received accuracies of `Accuracy_knownFaces: 0.7157894736842105`, `Accuracy_unknownFaces: 0.8` and `Accuracy_nonFaces: 1.0`. The hyperparameters that we worked with were as below. One can change the hyperparameter at the start of demo.py to get different results.

We print the accuracies to the console and the main class that has been implemented resides in `src.py`. Have a look at that file to know the implementation details.

Hyperparameters used:

```
K = 30 # dimension of face_space
b = 2 # number of classes to keep unseen
te = 2100 # Threshold for the L2 distance for training weight vectors
tf = 13000 # Threshold for the L2 distance from the face space
unknownface = -1 # label to denote an unknownface
nonface = -2 # label to denote an nonface
```

Top 3 eigenfaces:



Conclusions

- Setting the hyperparameter of threshold `te` to a big value ensures that we get good recognition when the faces are known but keeping it low will increase the accuracy of detecting an unknown face at the cost of accuracy of recognizing a known face.
- Keeping `K` bigger ensures better recognition, at the cost of detecting an unknown face.
- One needs to tune the values of `te`, `tf`, `K` according to a particular use case. For example, taking `K=120` and `te = 3500` in our dataset testing results in `Accuracy_knownFaces: 0.9263157894736842`, `Accuracy_unknownFaces: 0.0` and `Accuracy_nonFaces: 1.0`.

Reference

- F. Samaria and A. Harter
  “Parameterisation of a stochastic model for human face identification”

2nd IEEE Workshop on Applications of Computer Vision

December 1994, Sarasota (Florida).