Face Recognition with OpenCV and Python

Shubham Saoji
Department of Computer Science
University of Florida
Gainesville, United States of America
shubham.saoji@ufl.edu

Abstract

Face Recognition has received significant attention in past several years. One of the reasons is that face recognition has many applications in commercial as well as law enforcement domain. Another reason is the availability of feasible technologies after many years of research which includes but not limited to availability of high-resolution cameras, cheaper and more memory or cloud storage for images, higher computation power of CPU etc. Face Recognition is classification problem and features of face such as face structure, nose width, width of eyes, pupils are extracted from image and compared to class of features from images introduced during training phase of model and the face is categorized according to closest match. In this study, we will try to learn more about face detection and recognition/classification system.

1. Introduction

A Face Recognition system is a technology capable of identifying and verifying a person from a digital image or a video frame from a video. It is typically used as access control in security systems and can be compared to other bio-metrics such as fingerprints or eye iris recognition systems.

One of the most commonly used examples is using face recognition technology to unlock the phone, marking the attendance system. The idea of the face recognition system is it identifies a particular face from an image and compares it with other faces that we have in our database and returns the name or identity.
Face recognition involves 4 steps:

- Face Detection
- Face Alignment
- Feature Extraction
- Face Recognition

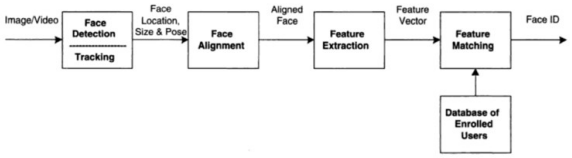


Fig. 1.2. Face recognition processing flow.

Face Detection:
It is a process to automatically detect human faces in visual media. To identify an image we use features such as noes, eyes, face cut or face structure, etc. Face detection is based on 68 face landmark points. Landmarks like eyebrows, lips, nose, mouth, etc.

Face alignment:
After the face is detected we need to normalize this detected face to improve the accuracy of face recognition. In face detection, we mark the boundaries of the face with a rectangle, in face alignment, we output the image which is face-centered. The output image has a line joining the center of two eyes parallel to the horizontal line and the image is resized according to the length between two eyes. This makes face more readable to the system to get features. In other words, this is a pre-processing step to face recognition. Operations such as translation, scaling, and rotation are used to make face alignment. The translation is shifting image from one position to another position without a change in height or width or any angular rotation. Rotation is changing the angle of image or orientation of image and scaling is changing the height and width of the image.

Feature Extraction:

After the pre-processing step now we need to extract features for feature vector. So, we'll have all features relevant to face recognition. In this step, the face recognition system produces a 128-d embedding, which is a 128 numbered measurements from the image. The system uses these 128 numbers to compare with measurements of images in the database in later steps.

Face Recognition/Classification
We compare the 128-d embedding of the subject image to that of the image stored in the database and a score is calculated based on euclidean distance between each face in the new image and known person image. If this score is above the pre-set threshold then there is a match. The face recognition is used for face recognition as it uses a pre-trained model based on ResNet architecture.

Face detection is based on 68 face landmark points. Landmarks like eyebrows, lips, nose, mouth, etc. Whereas for face recognition we require 128-d embeddings.

2. Face Detection

It is a process to automatically detect human faces in visual media. To identify an image we use features such as noes, eyes, face cut or face structure, etc. Face detection is based on 68 face landmark points. Landmarks like eyebrows, lips, nose, mouth, etc.

Some of the methods used in face recognition are:

1. Haar Cascades Classifier:
This is the first machine learning-based cascading classifier. This method is fast and can be run on low-power CPUs. This is the reason this method finds application in low power CPU's like mobile phones and cameras with less computing power.

The classifier is trained with many positives (images with faces in it) and an equal number of negatives (images with no faces in it).

It concatenates several classifiers into one classifier. An image has many features out of which many features are irrelevant. We need to select only small features that are relevant and have weight. Adaboost is used for feature selection to select the best features. In this, irrelevant features are discarded at the end of each classifier.

Thus increasing classifiers in cascading reduces irrelevant features and our system will have only those features that have weight. Increasing the classifiers will, in turn, increase the accuracy of face detection. We keep on adding classifiers until the accuracy requirement is met and the error rate reduces.

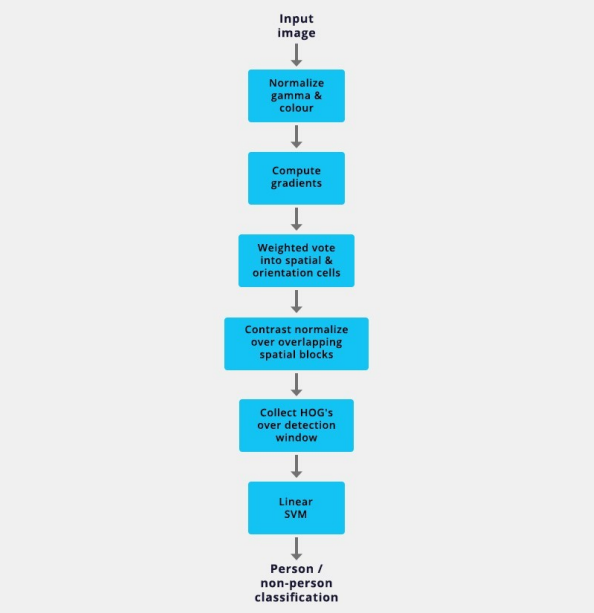
2. Histogram of Oriented Gradients (HOG):
This method uses a Histogram of Oriented Gradient features as image descriptor and trained Support Vector Machine (SVM) classifier to create a highly accurate human detector. A gradient is the direction of intensity change in pixel. Any type of object detector can be created by training HOG accordingly. A HOG relies on the distribution of intensity gradients or edge directions.

The advantage of HOG are -

- Very lightweight
- Works in small occlusion such as a face with glasses on a person's face.

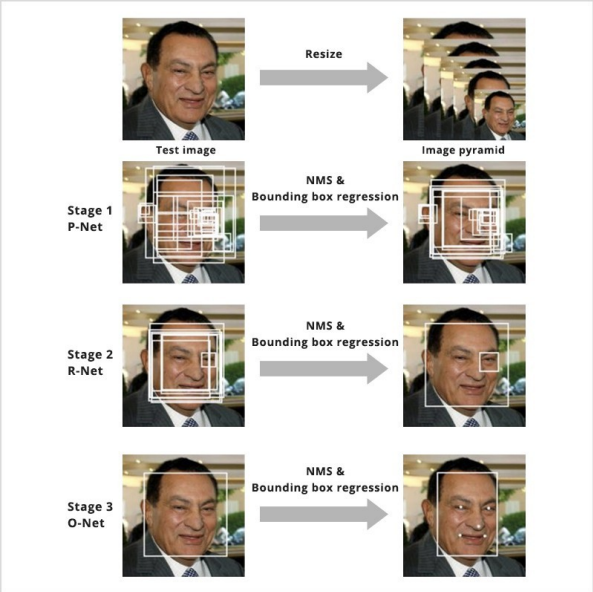
The disadvantage of HOG are -

- Min size of the face should be 80x80 pixels.
- It can detect only front-facing images. It does not work on side faces and high extremes of non-frontal images.
- It doesn't work on heavy occlusion.



HOG implementation pipeline. [**Source:** From paper Histogram of Oriented Gradients for Human Detection, Dalal and Triggs, 2005]

3. CNN Face detector in Dlib:
This is one of the methods which is based on a deep learning approach. It uses cascading of Convolution Neural Networks of three stages to train and predict face and five landmark locations which help in face alignment. In the first step, the image is passed to the Proposal Network (P-Net). Using the regression technique it predicts bounding boxes. After that, non-maximum suppression (NMS) is used to merge highly overlapped candidates. In the second step, previously output candidates are the input to another CNN, called Refine Network (R-Net). This CNN will also reject a large number of false candidates and predict more accurate bounding boxes and apply NMS to remove overlapped boxes. In the third step, Output Network (O-Net), is similar to the second stage, but this network will also predict five facial landmark's position. The training of this detector is simple and doesn't require large training data.



MTCNN pipeline [**Source:** Joint Face Detection and Alignment using Multi-task cascaded convolution Networks, Zhang, IEEE]

The advantages of this method are -

- A detector can detect multiple face orientations.
- Works with medium occlusion

Disadvantages are -

- The detector is very slow on CPU. For faster speed, GPU is required.

3. Related Work in Face Recognition

There are many efficient approaches for face recognition system. Some of them are Face Recognition using the Eigenface approach, Face recognition using Convoluted Neural Networks approach, Face Recognition using the fisherface method where the reduction in feature vector is done by Principal component Analysis(PCA). The methods differ in accuracy and efficiency resulting from using any approach and selection depends on the application for which face recognition is required. Approaches like Convoluted Neural Network for face recognition provide higher accuracy, however, they require high computing power. So, it is a trade-off. In scenarios where accuracy is an important factor, such as biometrics, the use of such approaches is justified, however at places where the accuracy of the face recognition system is not paramount, we can go with some other low-end approach. The working of all approaches is somewhat similar. The model needs to be trained using many training images. The model detects the face from an image by using features link nose, eyes, etc. The detected face is then aligned so as feature extraction process can follow and extract the most relevant features. The main difference between different approaches is the way the features are selected or rejected to ensure accurate face recognition using minimum features. So, in this step, redundant features are rejected. Then using a classifier, a comparison is done between these features and features of images in the database and an image is classified accordingly.

Let us look at some of the face recognizers,

1. EigenFaces Face Recognizer -
This recognizer extracts features out of training images and selects only those features that cause more variance. Other features are rejected. This means, if we are using this recognizer to classify images between two individuals and train model using images of them, it will retain only those features that are causing maximum variance or change. The common features are discarded.

Advantages of this approach are -
1. Processing can be reduced as raw data can be directly used.
2. It is simple and efficient compared to other classifiers.

The disadvantages of this approach are -
1. The training of the model is time-consuming.

2. The overlapping between different classes increases due to the same feature vectors representing different classes then the recognition rate reduces.

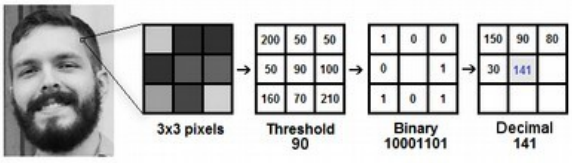
2. FisherFaces Recognizer -
This recognizer discards discriminating features and selects common features in faces of all people. It discards features that discriminate one person from another.

Advantages of fisherface recognizer -
1. It is more light invariant to light intensity and has better accuracy than eigenface recognizer.

Disadvantages are -
1. It needs more storage space for each face.
2. It requires more time than an eigenface recognizer to classify a face.

3. Local Binary Patterns Histogram -
The performance of both eigenface and fisherface recognizer decreases because of light conditions as it is not always possible to have optimum light conditions. LBPH algorithm tries to find the local structure of the image than finding local features to distinguish between faces. The structure can be built by comparing each pixel with the neighboring pixel.

Understanding LBPH face recognizer -
Let us look into the LBPH recognizer in detail as we'll be using this in our implementation.



So, we feed recognizer with grayscale images of faces and the recognizer converts it to a matrix of pixel values as shown in the first step in the above image. Then the threshold is taken as a central value in the matrix. The matrix is converted to binary by one simple rule that the neighboring cells which have a value less than the threshold are valued 0 and others are set to 1. This binary value that we get, as in the case given in the above image is 10001101, is then converted to a decimal value and assigned to the central cell of the matrix.

Following the above steps, we get a new image, and this image along with initial parameters X and Y passed as input will give us a histogram. Each image is represented by a separate histogram.

Face Recognition using LBPH -
For any input image, the histogram is calculated and compared with histogram images in the database by calculating the euclidean distance between two histograms.

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

For the image that gives us a minimum Euclidean distance, we label our image with its identifier.
This distance value is the confidence that this algorithm returns. So, the lower the confidence value, the higher are the chances of a match.

We define some threshold and if confidence value returned is less than this value, we have a match.

4. Face Recognition System with OpenCV Implementation

We are using python language because of the pool of libraries available and ease of use of these libraries. Libraries that we have used are -

NumPy library – This library can be used to deal with a number based multi-dimensional arrays.
ScikitLearn – It has many functions and data structures that can be used in data analysis projects.
OpenCV – This library is one of the most important libraries when it comes to computer vision or image processing related activities.

In our project, we train our model with multiple images of each identity that we wish our model to predict. The preprocessing part is converting this color image to a grayscale image as this will increase the detection rate of our model. We have used the Haar cascade classifier module for face detection. It detects multiple faces in one image with precision and returns images of faces for further processing.

We are using LBPH face recognizers in our project and we have set the threshold value to 60. If

confidence value returned is less than 60 then there's match and we label face with an identifier.
Note – In my project, I have used term uncertainty instead of confidence as it logically fits there and makes more sense. As uncertainty value increases there are fewer chances of a match.

5. Performance Evaluation

We trained our model with around 100 images of two actresses, Priyanka Chopra and Kangana Ranaut. Then we tested our model on many test images,

Below are performance statistics:
Accuracy
Face detection 100%
Face Recognition 87%

We tested this model on around 50 test images and came to these results.

The accuracy of the model based on OpenCV Haar Cascades and LBPH is low, however, it can be executed on a system with low computing power. We can use the face recognition module which is CNN based approach in case of availability of higher computing power systems or GPUs. The accuracy of this approach is much higher.

6. Conclusion

In the building of this system, we got a better understanding of the face detection and recognition system. This gave exposure to the working of classification models and an opportunity to learn about new approaches and algorithms in the field of face recognition.

7. References

[1] J. Lu and K. N. Plataniotis, "On conversion from color to gray-scale images for face detection," 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Miami, FL, 2009, pp. 114-119.

[2] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. 2003. Face recognition: A literature survey. ACM Comput. Surv. 35, 4 (December 2003), 399–458. DOI:https://doi.org/10.1145/954339.954342

[3] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no.1, pp. 103-108, Jan. 1990.