Face Recognition Using OpenCV and Python

By

Shubham Saoji

Outline

1. Introduction
2. Applications
3. Face Recognition
4. Face Detection
5. Face Alignment
6. Feature Extraction
7. Implementation

INTRODUCTION

- Everyday actions are increasingly being handled electronically.
- This results in increased demand in fast and accurate user identification and authentication.
- A Face Recognition system is a technology capable of identifying and verifying a person from a digital image or video frame from video

Applications

- Face Recognition system can be used as access control in security systems
- It can be used to mark attendance
- Unlock mobile phones
- Tag friends on social media

Face Recognition

Face Recognition system has 4 steps

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

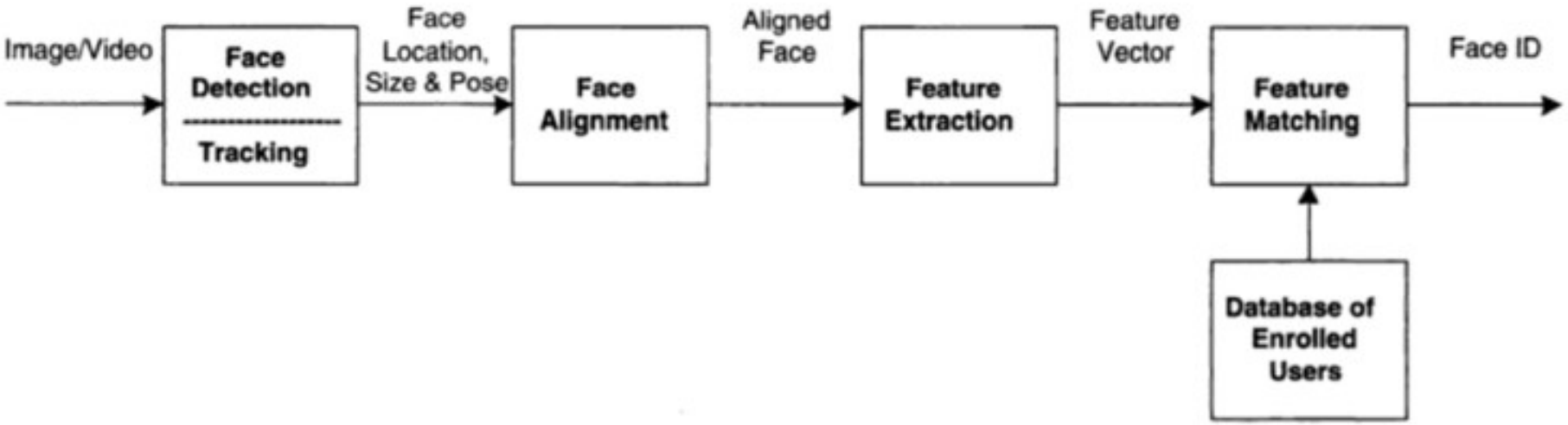


Fig. 1.2. Face recognition processing flow.

Face Detection

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

Face Alignment

- Normalize image of face and get face-centered image
- This makes face more readable to the system to get features.
- Operations such as translation, scaling, and rotation are used to make face alignment

Feature Extraction

- Extract relevant features from image
- 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

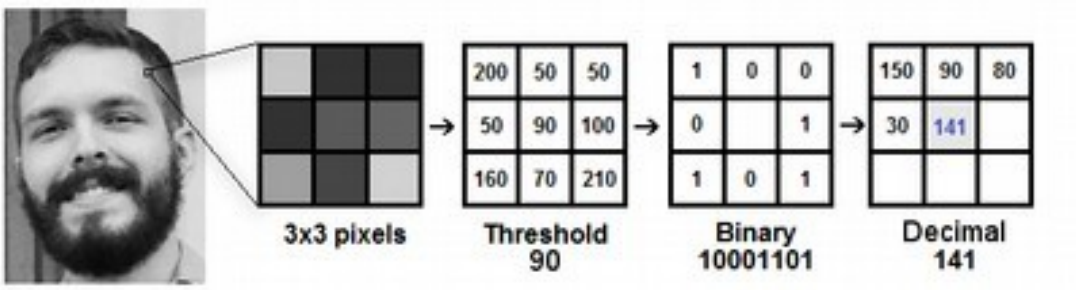
- We compare the 128-d embedding of the subject image to that of the image stored in the database and a score is calculated based on euclidean distance between each face in the new image and known person image.
- If this score is above the pre-set threshold then there is a match.

Implementation

- Face Detection – We have used Haar Cascade Classifier for face detection
- 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.
- Adaboost is used for feature selection to select the best features. In this, irrelevant features are discarded at the end of each classifier
- Some other face detectors - Histogram of Oriented Gradients (HOG), CNN Face detector

Implementation

- For face recognition we have used Local Binary Patterns Histogram (LBPH)
-



- Each image is represented by a separate histogram.
- For any input image, the histogram is calculated and compared with histogram images in the database by calculating the euclidean distance between two histograms.
- This distance is confidence value
- Lower the confidence value, the higher are the chances of a match

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