Computer Vision
Fingerprint matching and Recognition

Mohamed Abdelwahed
Ibrahim Abdalaal

Abstract

The fingerprint system is widely used in office access nowadays. It is part of the different biometric systems, as each fingerprint for every human is unique. For that reason, it is used for a person's identification. The current systems use features based on minutiae points and ridge patterns. For trying to improve further, we use a new representation and matching scheme for fingerprint using Scale Invariant Feature Transformation (SIFT). We extract characteristic SIFT feature points in scale space and perform matching based on the texture information around the feature points using the SIFT operator. Using a dataset that is available for public use, we demonstrate our approach using SIFT.

Introduction

The biometric system is widely used for granting access to places to help increase the security and allow only the authorized people who are registered in the system. The most biometric system which is used is fingerprint recognition. Human fingerprints are practically unique. To allow persons to access a system, fingerprint scanners are used. Fingerprint scanners work by capturing the pattern of ridges and valleys on a finger. The information is then processed by the device’s pattern analysis-matching software, which compares it to the list of registered fingerprints on file. A successful match means that an identity has been verified, thereby granting access. The method of capturing fingerprint data depends on the type of scanner being used and the known scanners are three:

1. Optical sensor: These sensors capture a photocopy from the finger. They use an image sensor or light sensitive microchip to record a digital image.
2. Capacitive Sensor: They use electricity to determine fingerprint patterns. As a finger rests on the touch-capacitive surface, the device measures the charge; ridges exhibit a change in capacitance, while valleys produce practically no change at all. The sensor uses all this data to accurately map out prints.
3. Ultrasonic sensor: They work via sound waves. The hardware is designed to send out ultrasonic pulses and measure how much it bounces back. Ridges and valleys reflect sound differently, which is how ultrasonic scanners are able to create a detailed 3D map of fingerprint patterns. Ultrasonic sensors are currently being prototyped.

The fingerprint scanner can be put into a fixed robot arm which is then connected to a locked door only to be opened after the verification. The verification or identification process is done by two steps which are the preprocessing and extracting the fingerprint features which are ridges and valleys then matching these features with 1:M in the fingerprint identification scenario.

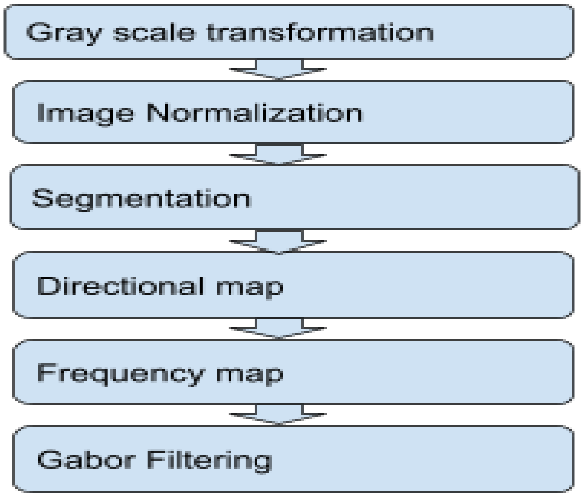
Methodology

Fingerprint recognition algorithm we used consists of two main parts as follows :

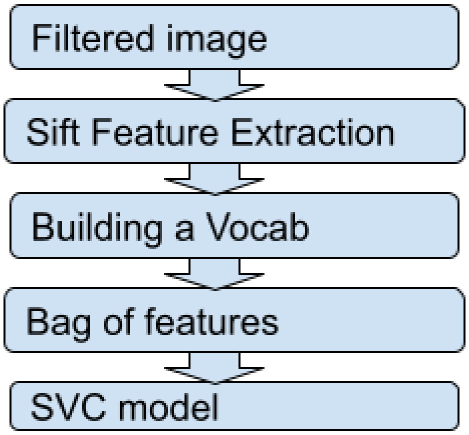
- Preprocessing step
- Extracting features and Matching then recognition

Preprocessing step

Preprocessing is a necessary and important step to enhance the images and improve the quality.Fingerprint images come with different widely different qualities so to step is done to assure the quality is good as possible and to further facilitate the next step of the fingerprint recognition.Different steps done for preprocessing in the next figure



Second part which is Extracting the biometric information of the fingerprint and recognition .Fingerprint biometric data is the mintuate (ridges ,circles ,blurfications).The steps includes in this part are in next figure :



Each of the steps included in both pre-processing and Extracting parts will be explained in detail.

- **Image Normalization**
  Normalizing the image one by subtracting the mean and dividing by the standard deviation.The main goal from this step is to reduce the variance of the gray levels and avoid the unwanted effects such as illumination.

- **Image Segmentation** (Ridge Segment)
  This a very necessary step in the preprocessing part.Segmentation is used to get rid of the edges and the noisy parts of the image.to segment the images ,we followed the STD calculation approach for image blocks as follows :

1-Dividing the image into blocks of size 16*16
2-Looping through all the blocks and for each block do the following

1. Calculate the STD of the block
2. Compare it to predefined threshold
3. If STD of the block is above the threshold then its true useful part of the image
4. If not ,then it is considered as useless part (background part of the image) and will be ignored in the preprocessing

Segmentation make us able to reduce the size of the image by including only the useful parts which further optimize the algorithm

- **Directional map**
  To define the orientation of the striates (fingerprint lines) ,a Directional map is needed.
  This map will be used for generating the frequency map and in the gabor filtering step.
  Generating Directional map consists mainly of two parts as follows :

Orientation estimation :

Core step in the Directional map generation to estimate the orientation of each pixel through the following steps :

1- taking a neighborhood in which the desired pixel is the center of it
2-Compute the Gradients in both x ,y direction Gx,Gy using Sobel filter
3-Local direction in the neighborhood rows and columns are estimated as follows :

```
np.sqrt(np.power(Gxy,2) +
np.power((Gxx - Gyy),2))
```

4-Obtaining the sines and cosines of the doubled angle

Part2 is smoothing Directional map as follow :
1-Smoothen the estimation of the orientation obtained from the previous part
2- this is done by using LPF or specifically gaussian filter
3-finally using tan^-1 ,we obtain the non noisy directional map

Frequency map:

The purpose of this step is to estimate the local frequency of the streaks in each pixel. The frequency of the image I (i, j) is an image F (i, j). This frequency is calculated by the ratio (1 / T) where T represents the period calculated between two successive extrema. We calculate the frequency as follows:

1. By looping over the rows and the columns of the block size of the image, we take the renormalized image in the dimension [r:r + block_size][:, c:c + block_size] and the oriented image in the same dimension, then we calculate the block frequency by calling the block frequency function.

2. The block frequency function finds mean orientation within the block. This is done by averaging the sines and cosines of the doubled angles before reconstructing the angle again. This avoids wraparound problems at the origin. Then we rotate the image block so that the ridges are vertical. After that, we crop the image so that the rotated image does not contain any invalid regions. This prevents the projection down the columns from being mucked up. Then, we Sum down the columns to get a projection of the gray values down the ridges.
3. After we find the block frequency, we multiply it by the masking, then we find the mean, then we multiply the mean by the making to find the ridge frequency.

Gabor Filter:

The gabor filter even symmetry and is oriented at 0 degrees and it is used to select in the Fourier domain the set of frequencies that make up the region to be detected. We designed it as follows:

1. We round the array of frequencies to the nearest 0.01 to reduce the number of distinct frequencies we have to deal with.
2. We generate filters corresponding to these distinct frequencies and orientations.
3. We Generate rotated versions of the filter and the orientation image provides orientation along the ridges, hence +90 degrees,
4. and imrotate requires angles +ve anticlockwise, hence the minus sign.
5. We Find indices of matrix points greater than maxsize from the image boundary.
6. We Convert orientation matrix values from radians to an index value that corresponds to round(degrees/angleInc), and after that we do the filtering.

Extracting minutiaes and Matching

After the preprocessing and the enhancement done to the image ,we now move to next step which is feature extraction and fingerprint matching as follows :

Sift feature Extraction (Sift_descriptor function)

In this step we extract the key points from each image using sklearn Sift Function.

For each image we obtain a unique descriptor and then appending them to a list to be used in the next step

Build Vocabulary

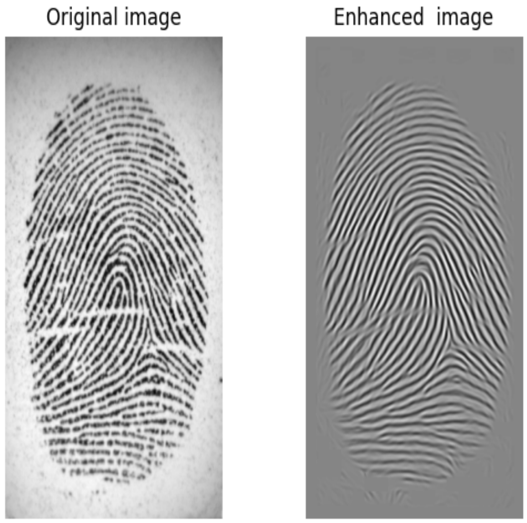
Form the massive matrix we got from above function, we apply Kmeans Returning the vocabulary with cluster centers only

Bag of words function

After building our Vocabulary of visual words (local features of all the images),we move now to build the bag of descriptors.

For each image ,we will count how many descriptor (feature) falls with in certain cluster by using the euclidean distance to get the closest cluster to that feature and then create histogram for each of them

- For each image,we extracted the sift descriptors (image feature vector)
- Getting the closest vocab word for each image



SVM Model to classify

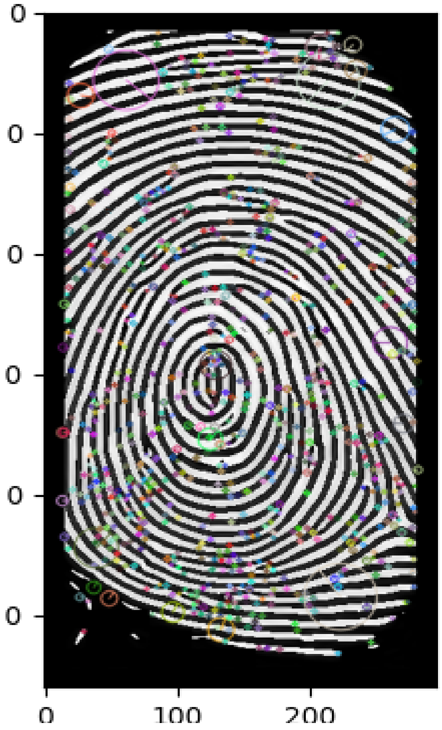
- This function is used to predict a category for every test image

- This function uses LinearSVC from sklearn.svm, and then we use the arguments to call it as (penalty='l2', loss='squared_hinge', dual=True, tol=0.0001, C=,1 multi_class='ovr', fit_intercept=True, intercept_scaling=1, class_weight=None, verbose=0, random_state=None, max_iter=1500)

Results:

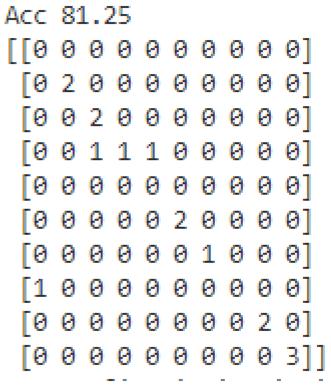
- After preprocessing steps and image enhancement ,it is clear that the quality of the image is much better with smaller size to optimize the algorithm.Obvious in the enhanced image is that we do not remove any important parts of the image.

- After SIFT feature Extraction and detection



Accuracy

Achieved accuracy : 81.25 %



```
Acc 81.25
[[0 0 0 0 0 0 0 0 0 0]
 [0 2 0 0 0 0 0 0 0 0]
 [0 0 2 0 0 0 0 0 0 0]
 [0 0 1 1 1 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 2 0 0 0 0]
 [0 0 0 0 0 0 1 0 0 0]
 [1 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 2 0]
 [0 0 0 0 0 0 0 0 0 3]]
```

Confusion matrix : is shown above

Discussion

Accuracy achieved is a good one but we believe further improvements could be made such as removing the false key points from the set of sift interest points.Removing the false features is expected to raise the accuracy higher.

In addition ,the results achieved is directly related to the quality of the images the sensor that capture the fingerprints and the status of the finger in the time of printing

Conclusion

Through this report ,detailed explanations of a fingerprint biometric recognition algorithm have been given.Starting with defining the problem moving to the solution.Then step by step implementation of the solution -has been proposed.Finally ,such systems could be used to for human identification and authentication