Final Report on the Project Titled *"Image Processing based Fingerprint Matching for Person Identification"*

Submitted by:
Mohaiminul Al Nahian
RIN: 662026703
Course No.: ECSE 4540

1. Introduction:

Fingerprints are considered a unique ID to an individual. As of today, no two individuals have been found with same fingerprints. According to literature, fingerprints are unique patterns on the fingers or toes of a person composed of friction ridges and valleys. The patterns are given different names such as loops, whorls and arches (see the following figure). Fingerprint matching or detection falls in the area of biometric recognition, which works with identification of people through the measurements of some of the anatomical, physiological or behavioral characteristics. Fingerprint detection has usage in forensics, access control, security and person identification. In our daily life, we are using our fingerprints to unlock the smartphone, log into bank accounts, verify online payments and many more. As fingerprints are unique to a person, it is a very good biometric feature of an individual. The recognition system for fingerprint matching must be able to reliably identify a person every time without failing. In case of access control for large group of people, the recognition system must reliably give access to authorized persons and deny the same to the people whose digital prints are not authorized in the database.

Figure 1: Fingerprint Patterns

Fingerprints are captured mainly in two approaches; optical scanner that takes some sort of digital image of the fingers and process it or capacitive scanners found in many smartphones that uses small capacitor arrays to store the electric charge difference created by the finger.

The goal of this project is to develop a well-performing fingerprint identification platform that uses digital images as the input for storing fingerprint data and then matching the stored data with another set of images to re-identify the person. The way this project has been designed; it can be thought of as an access control system for a large group of people. We first create a database of fingerprints and then test different fingerprints images to see if the developed system can distinguish between an authorized and unauthorized person.

2. Related Work:

The task of fingerprint detection is not new in the literature. Over the years many people have proposed different methodologies to do fingerprint recognition. The main three approaches of doing the task are *correlation-based, minutiae-based and non-minutiae feature based* such as ridge shape, texture information etc. Arun Ross et.al [1] proposed a fingerprint alignment and matching algorithm based on ridge feature map. A set of Gabor filters were used to capture the local ridge strengths at various orientations and a 2D correlation was used to determine the matching score. Shehu et. Al [2] proposed a deep CNN-based method to determine the types of fingerprint alterations. Jiang Li et.al [3] described a fingerprint matching algorithm that combines minutiae based matching method along with correlation based matching method. First, they use minutia based matching algorithm to extract the list of matched minutia pairs from the two fingerprint images and then take Correlation of the local neighborhood regions around each matching minutiae pair that represents local similarity. Peng Shi et.al [4] proposed a matching algorithm based on minutiae sets together with the mean ridge width and normalized quality estimation of the image. Chatterjee et.al [5] used minutiae feature such as thinned ridges from the fingerprint images and these feature matrices were applied as input data set to the Artificial Neural Network. Cavusoglu et. al [6]. used a variance-based segmentation method to segment the fingerprint image. They used local ridge orientation determined using Sobel operators. Apart from that, many newer methods seem to be resorting to Neural Network based methods to do the task automatically.

Each method has some advantages as well as disadvantages over the other. In this project, the minutiae-based method has been chosen as the main approach for fingerprint detection. Knowledge gained from different topics studied in the coursework have been used to perform various low-level and mid-level image processing tasks along with application of some the most established theories to develop a system that can distinguish between an authorized and an unauthorized person with high degree of reliability. It is safe to assume that any person taking this course should be able to recreate this project based on the knowledge gained from coursework and based on reading this report.

3. Data Collection:

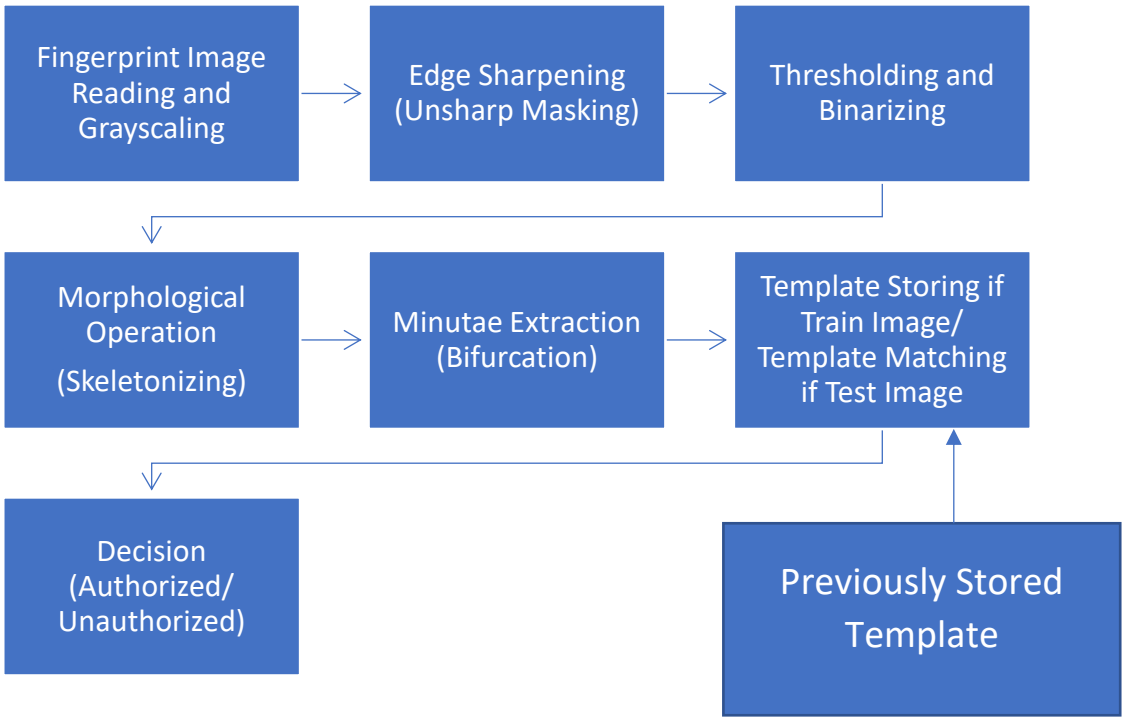
For this project, a publicly available dataset published by Kaggle has been chosen. The name of the dataset is Sokoto Coventry Fingerprint Dataset (SOCOFing) [7]. This is a biometric dataset designed for academic research purposes. It consists of fingerprint images of 600 African persons. The dataset has 6000 real images of 10 fingers of the 600 persons and many images that are altered version (obliteration, central rotation, and z-cut) of the original images. In other words, a certain finger of a certain person has a real picture

as well as digitally altered version of the same. Some of the sample images are shown below

Figure 2: 3 Different Images of the same finger from the database.

4. Technical Approach:

In the developed system for fingerprint identification, several basic to high level image processing techniques have been applied for fingerprint identification. The block diagram of the workflow has been shown below



We shall describe each of the process mentioned in the block diagram in this section below:

a. Image Reading and Grayscaling: A raw image is read and converted into grayscale in this step.

b. Edge Sharpening (Unsharp Masking): In order to enhance the ridges of the fingerprint of a person, we apply an edge sharpening technique called unsharp masking. The unsharp masking can be formulated by the equation

$$\textit{Sharpened Image= Original Image+ (Original-Blurred)*amount}$$

In other words,

i. $g_{mask}(x,y) = f(x,y) - f_{smooth}(x,y)$
ii. $g(x,y) = f(x,y) + k * g_{mask}(x,y)$

At first we take the raw image f(x,y) and apply a Gaussian blurring filter on to it. In our case, after several trial and error, we have used a Gaussian filter with sigma=3 to get a blurred image $f_{smooth}(x,y)$. We subtract the blurred image from the original to get the mask g_{mask} (x,y). We then add this mask multiplied by a value k to the original image to get the sharpened image g(x,y). In our case, we selected k=2 after several trial and errors to get the best possible results.

c. Thresholding and Binarizing: After creating the sharpened image, we need to threshold the image in order to binarize it for further processing. In order to threshold the image, we use an automated thresholding algorithm known as the Otsu's Method of thresholding. The algorithm returns a single intensity threshold that separate pixels into two classes, foreground and background. This threshold is determined by minimizing intra-class intensity variance, or equivalently, by maximizing inter-class variance [8]. In this method we select a threshold value and for that, we determine the inter-class and intra-class variances. We tune k so that eventually the inter-class variance is maximized and the two classes are well-separated. The process is shown below:

$$\begin{aligned}
C_1 &= [I(x,y) | I(x,y) < k] \\
C_2 &= [I(x,y) | I(x,y) \geq k] \\
P_1 &= P(C_1) = \sum_{i=0}^{k} P(I(x,y) = i) \\
P_2 &= P(C_2) = 1 - P_1 \\
m_1 &= \frac{\sum_{i=0}^{k} i \times P(I(x,y) = i)}{P_1} \\
m_2 &= \frac{\sum_{i=k+1}^{255} i \times P(I(x,y) = i)}{P_2} \\
\sigma_B^2 &= P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2
\end{aligned}$$

Here, m_G and σ_B^2 are global mean and variance. We tune the value of k to maximize the value of σ_B^2 and that value of k is our threshold.

After getting the Otsu threshold, we binarize the image by making all pixel value above threshold as 1 and below threshold as 0. This makes the fingerprint ridges to 0 and background to 1. In order to do further processing, we invert the image so that the ridges become 1 and background becomes 0.

d. Morphological Operation (Skeletonizing): The next step is to skeletonize the binary image. Skeletonization is a process for reducing foreground regions in a binary image preserving the connectivity of the original region while throwing away most of the original foreground pixels. The skeletonization operation uses morphological thinning that successively erodes away pixels from the boundary (while preserving the end points of line segments). We keep eroding the image with some structuring element until no more thinning is possible, at which point what is left approximates the skeleton which is basically thin lines with only 1 pixel width. To extract the minutiae properly, this skeletonization is a very important step.

e. Minutiae Extraction (Bifurcation): In the field of biometrics, minutiae refer to certain small features of a fingerprint image. In essence, minutiae can be applied to any type of image, but they are often discussed in relation to a biometric fingerprint image. Some example minutiae include *bifurcations* and the endings of ridges *(termination)* on a fingerprint pattern. Scar contours and edges may also be considered minutiae. An example picture is shown below:

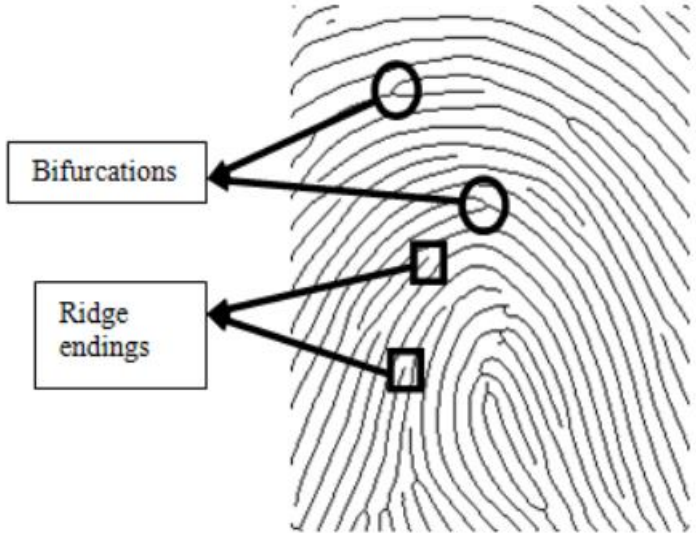


Figure 3: Minutiae Example

Here, Ridge ending and bifurcation examples are shown. Ridge ending is where the ridges of the fingerprint terminate. It is also called termination minutiae. Bifurcation is the region

where the ridges divide into 2 separate lines. Bifurcation looks like a ‘Y’ or a ‘T’ pattern in the fingerprint. In this project, we have used only bifurcation as the minutiae for fingerprint matching. We process the image to get the position of the bifurcation points from the skeletonized image. We do local region/ block processing of the skeletal image. We define 3x3 blocks of pixels and divide the entire image by many 3x3 blocks. In each block. we look for Y or T patterns. A few example patterns could look like this

Figure 4: Bifurcation points Visual example in a 3x3 block

So, we can see that in each block, there are only 4 pixels that has a value of 1. So, we write our algorithm such that in each 3x3 block, if the sum of the pixel value is 4, then we define the centroid of that block as a bifurcation point. As, the image is already skeletonized, there is hardly any chance that the skeletal image has lines with 2 or more pixel width and any points other than the bifurcation could not possibly result in a sum of 4 in a 3x3 block. Algorithmically,

i. *check:* sum(block pixels)
ii. *If sum== 4:* Take the position of block centroid as a bifurcation point

f. Template Storing: The train images or the images used to create the database are used for template creation. The images are taken and all the processing up to above section are done on them. Once we have the minutiae positions, we do not save the entire template image. Rather, we save the position index of the bifurcation points of all the template images into a file. In this way, we get two benefits. The stored database is smaller in size if we compare it with a database having template images as a whole, secondly, it makes the verification of a test fingerprint faster, as we do not have to load all the individual template images during the inference phase.

The coding platform used for this project is python. The major libraries needed for this project are *numpy* for matrix calculations, *matplotlib* to plot results, *openCV* and *scikit-image* to do different processing on the images.

g. Template Matching and Decision Making: In order to take a decision on the fingerprint verification task, we need to match the minutiae of the test image with that of the database images to see which of the database image most closely match with the test image. The metric that we use to match the templates is called The Structural SIMilarity (SSIM) Index proposed by Wang et. Al. [9]. SSIM is a perception-based model that considers image degradation as change in structural information. The idea is that the pixels have strong inter-dependencies when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene. If we have two images X and Y, the SSIM between them is defined as:

$$SSIM(X,Y) = \frac{(2\mu_x\mu_y + C1)(2\sigma_{xy} + C2)}{(\mu_x^2 + \mu_y^2 + C1)(\sigma_x^2 + \sigma_y^2 + C2)}$$

Here, $\mu_x, \mu_y, \sigma_x^2, \sigma_y^2, \sigma_{xy}$ are the average of x, average of y,variance of x, variance of Y and covariance of X,Y

The C1 and C2 parameters help avoiding a possible divide by zero result. The result of SSIM is a value between 0 and 1 where high value indicates more structural similarities. We take a test image and process it to generate the minutiae. Then we compare the image with the templates of all the database images and calculate the SSIM. The highest SSIM should correspond to the prediction of the fingerprint. However, in case the test person's finger is not in the database, then the algorithm should raise a No Match flag. To ensure that, we have declared a minimum SSIM value that at least one of the template images have to cross to declare the test finger as an authorized person. This ensures to minimize the False Positive or unauthorized access of a person at the expense of having some False Rejection Scenarios, where due to a bad scan of the test finger, an authorized person might be declared as unauthorized due to lower SSIM value with the saved templates.

5. Intermediate Results:

We apply the algorithm described in Section 4 to all the images of our selected dataset to get minutiae for all of them. We load an image and then make it gray scale. Then we apply the unsharp mask to get an edge enhanced more contrasty image. The results typically look like the following

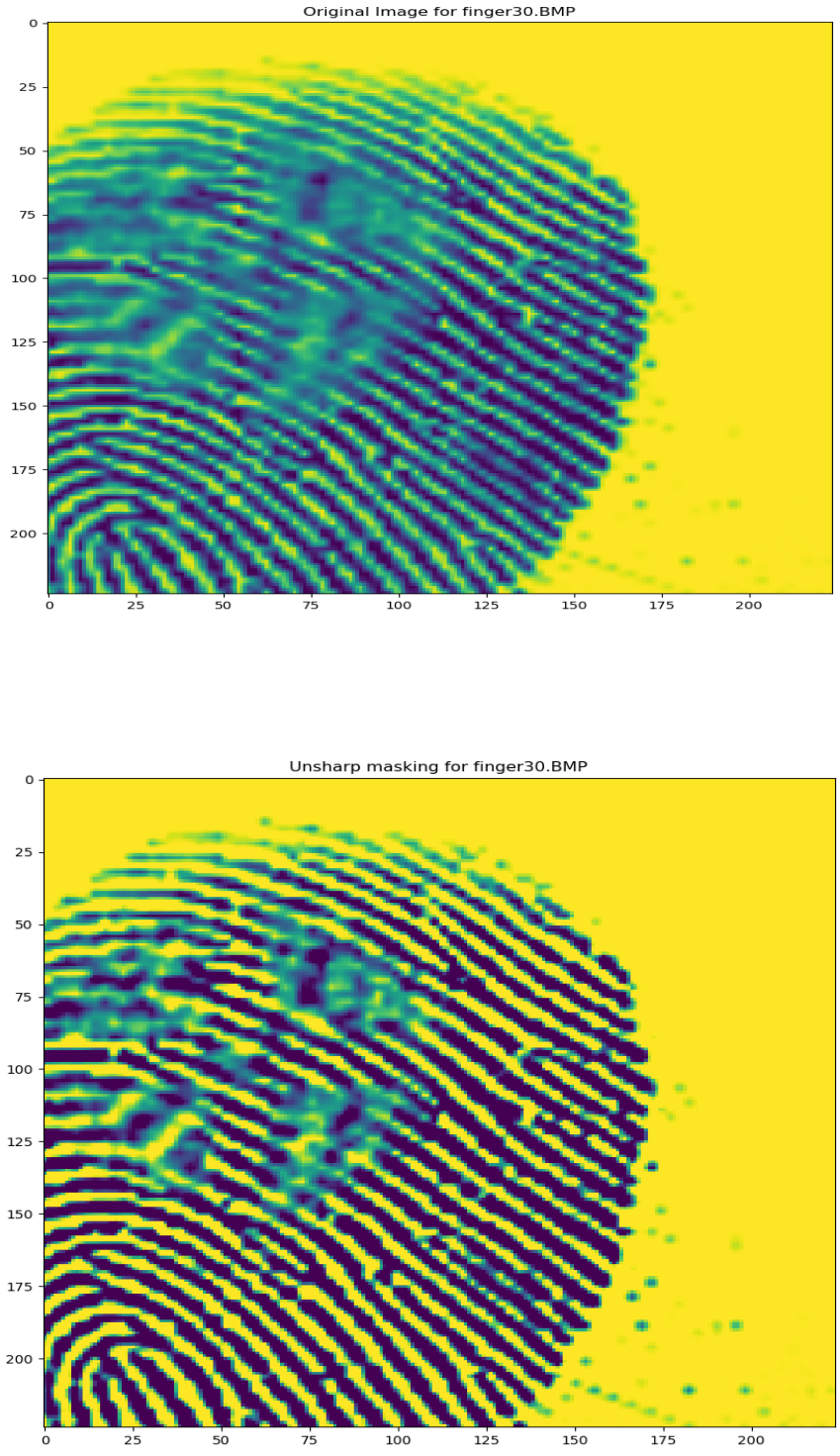


Figure 5: Original Image (Grayscale) and Unsharp Mask Result

We have tried different values of sigma for the Gaussian blur filter and took the one that gave the best performance (sigma=3). A graph is shown below:

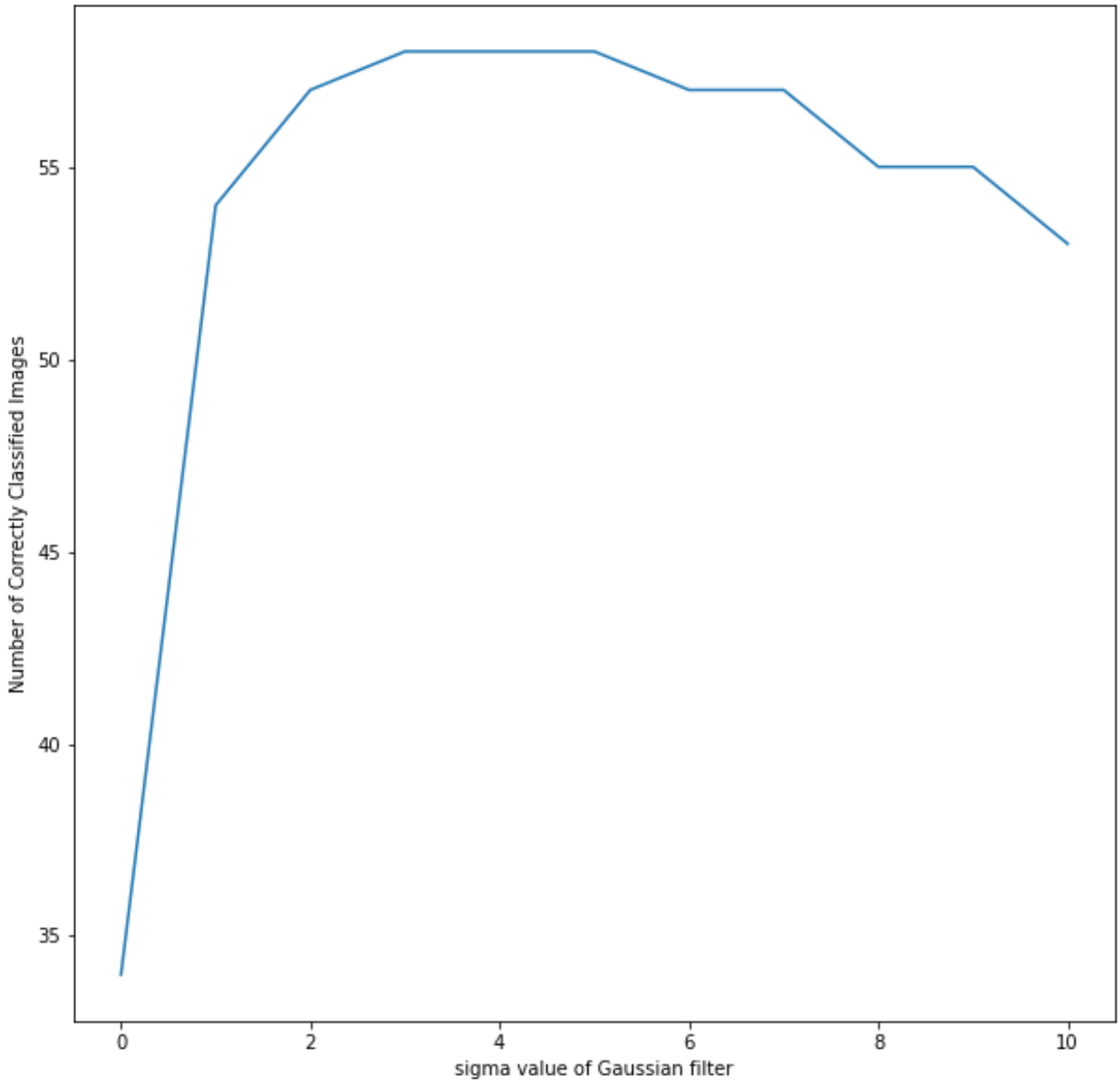


Figure 6: Model Performance based on different sigma value

After that, we do Otsu thresholding and Skeletonizing to the images. The resultant images look like the following:

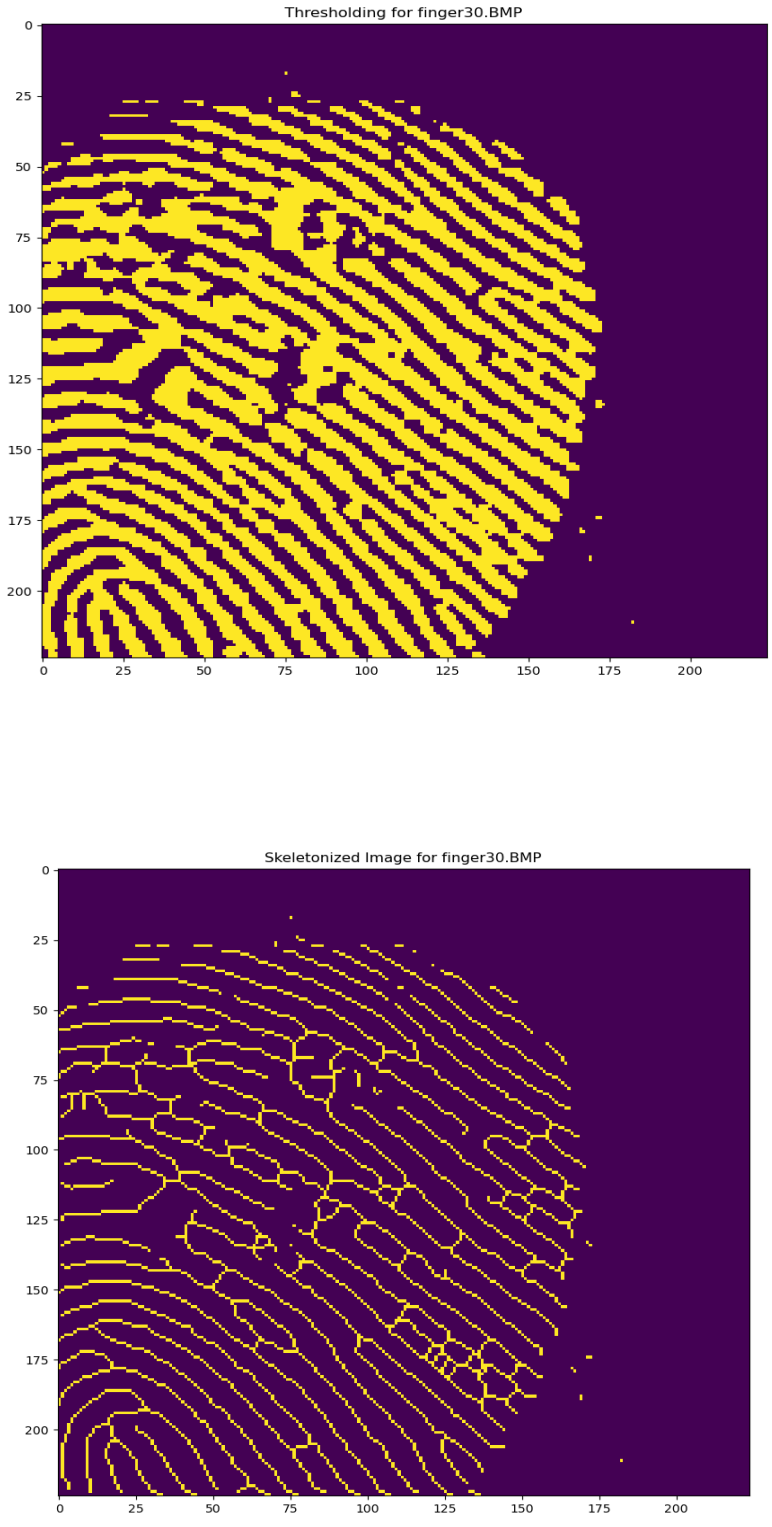


Figure 7: Binary Image produced by Otsu Threshold and corresponding Skeletal Image

After that, based on the method described previously, we calculate minutiae or bifurcation positions from the skeletal image. A typical resultant image after bifurcation detection looks like this

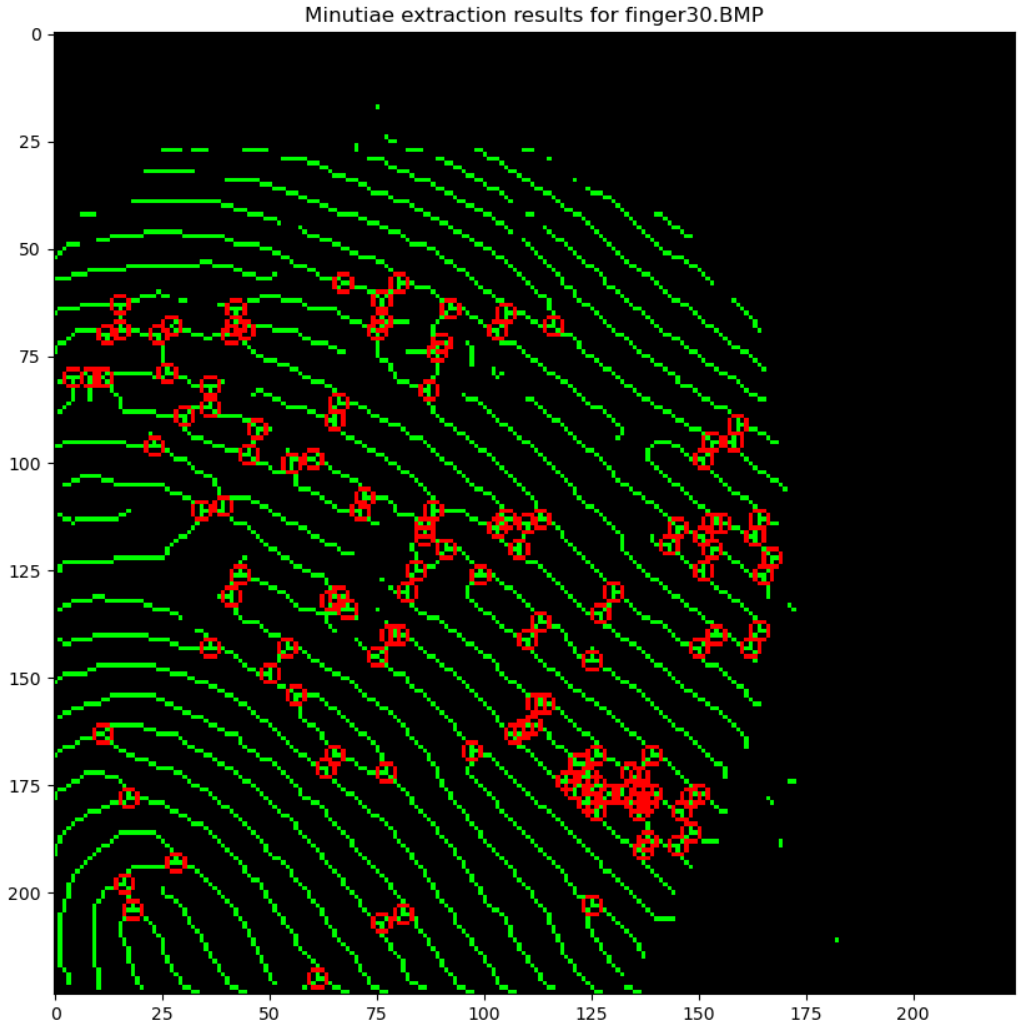


Figure 8: Minutiae positions on the Fingerprint Skeletal Image

The red circles denote the bifurcation points of the image where the fingerprint ridges take a Y or T shape.

After getting the minutiae points we calculate the SSIM between test image and all train image templates. We declare a threshold SSIM of 0.7 for the test image to be passed as an authorized person's finger. The threshold 0.7 is a relatively high SSIM value. After several trial and errors, this value is chosen. The reason of choosing high ssim threshold is that, if we compare with real life, even if an authorized test finger is rejected by the algorithm due to poor quality of scan and not being able to pass the ssim barrier, there is not much harm to it. We can scan again to get a better SSIM value. But if an unauthorized finger is predicted as one of the fingers of the database due to threshold being low and the unauthorized finger showing more than the minimum similarity with one of the images, then it will be predicted as an authorized finger, compromising the security of the system. So the goal has been to keep the False Acceptance to the lowest while the False Rejection is also not very high. We test the method in a subset of the entire dataset and see that we have achieved zero false acceptance rate while the false rejection was also relatively low. In a nutshell, the developed algorithm has performed extremely well on the chosen fingerprint dataset. However, one weakness of the algorithm is that we have chosen SSIM as the matching algorithm, which does not do very well in terms of scaling, rotation and translation of the images. The images in the chosen database were just artificially distorted spatially, as a result, even though the test images were different than the train template images, we could get high SSIM value for matching image pairs. But in case of rotated, scaled or translated images, this algorithm might struggle a bit. Another method proposed by the authors of SSIM paper, known as CW-SSIM or Complex Wavelength SSIM deals with the problem of rotation, scale, translation. So, using that method could make the developed algorithm more robust.

6. Final Results:

To show some quantitative results, we applied our algorithm to a subset of the SOCOFing dataset. The reason for taking small subset is that manually annotating ground truth for the entire dataset was a tedious job. However, we generalize our result by applying some statistics and probability to predict how well our model should perform for unseen or untested data. We show the statistical result in this section as well.

We selected fingerprint of 40 persons from the 600 total. For each finger, we have taken 3 images which are altered using different modality. So, we have 120 images in total of 40 fingers. The reason for taking 3 instances of the same finger is to make the prediction more reliable, so that the test image should match with at least one of them very closely. We create template database using these 120 images, along with ground truth. For testing the model, we have taken 60 images. 40 of which correspond to the 40 people of the train set. However, the test data images, even though of the same finger, are different that the train data images in terms of alteration levels. The last 20 test images are not present in the

template database, in other words, these are unauthorized fingerprints. The metrics used here are accuracy, Classification Error, True Positive Rate, False Positive Rate. Accuracy means ratio of how many of the test images were correctly predicted with respect to total number of datapoint. Error is just accuracy subtracted from 1. True Positive Rate means how many of the authorized test fingers were accurately predicted. False Positive Rate means how many of the unauthorized fingers were predicted as one of the authorized fingers. The results are shown below:

Metric Name	Value
Accuracy	96.67% (58 out of 60)
Error	3.33% (2 out of 60)
True Positive Rate	95% (38 out of 40)
False Positive Rate	0% (0 out of 20)

So, we have achieved fairly high accuracy and True Positive Rate. Also, False Positive Rate of 0% indicates, no unauthorized fingers have been authorized in the test dataset. However, since it is a small dataset, we generalize the error probability for general unseen test dataset using Hoeffding bound. Here, number of test sample, N=60. Test error is 3.33% or 0.0333. Let's take a tolerance value $\delta = 0.1$. So, according to the Hoeffding bound, we have

$$E_{out} \leq E_{test} + \sqrt{\frac{1}{2N} log \frac{2}{\delta}}$$
$$so, E_{out} \leq 0.0333 + 0.13 \approx 0.17$$

As $\delta = 0.1$, so we have 90% confidence that our model should have an error less than 0.17 or 17% for any arbitrary sized fingerprint dataset. However, we can expect a lot less error from our model based on the trend of the result that we got on the toy dataset.

7. Discussion and Future:

In this project, we have successfully implemented a fingerprint matching system based on the knowledge gained from the coursework and the state-of-the-art literature. The developed system works very well on the publicly available SOCOFing dataset. By

carefully choosing our image processing algorithms and different decision thresholds, we have achieved very good performance in terms of accuracy, error, TPR and FPR. Especially, the FPR has been 0 for all the selected test data. However, there is always room for improvement. After choosing the dataset, my goal had been to achieve good performance on this dataset. But, as discussed before, in this chosen dataset, the images are not rotationally and scale wise different, so the algorithm developed for this project might need to be tested and tweaked with a more robust database in the future in order to be more generally usable as a fingerprint matching platform. Also, as a matching criterion, other methods apart from SSIM, such as correlation, SIFT features extraction for scale invariance, CW-SSIM these could be employed. Different minutiae such as Termination could be added as well along with bifurcation to increase reliability. Again, if I got time, it would have been interesting to see if the developed method or some upgradation of the developed method can be used to match fingerprint images taken directly from cellphone cameras either in normal mode or in macro lens mode.

Reference:

1. Arun Ross, James Reisman and Anil Jain, (2002), “Fingerprint Matching Using Feature Space Correlation”, Biometric Authentication. Springer Berlin Heidelberg, pp. 48-57.
2. Shehu, Y.I., Ruiz-Garcia, A., Palade, V., James, A. (2018) “Detection of Fingerprint Alterations Using Deep Convolutional Neural Networks” in Proceedings of the International Conference on Artificial Neural Networks (ICANN 2018), Rhodes – Greece, 5th - 7th October 2018. Springer-Verlag Lecture Notes in Computer Science
3. Jiang Li, Sergey Tulyakov and Venu Govindaraju, (2007), “Verifying Fingerprint Match by Local Correlation Methods”, First IEEE International Conference on Biometrics: Theory, Applications,and Systems, pp.1-5.
4. Peng Shi, Jie Tian, Qi Su, and Xin Yang, (2007), “A Novel Fingerprint Matching AlgorithmBased on Minutiae and Global Statistical Features”, First IEEE International Conference on Biometrics: Theory, Applications, and Systems, pp. 1-6.
5. Atanu Chatterjee, Shuvankar Mandal, G.M.Atiqur Rahaman and Abu Shamim Mohammed Arif,(2010), "Fingerprint Identification and Verification System by Minutiae Extraction UsingArtificial Neural Network", JCIT, Vol. 01, Issue 01, pp.12-16
6. Abdullah Cavusoglu and Salih Gorgunoglu, (2007), ” A Robust Correlation Based Fingerprint Matching Algorithm for Verification”, Journal of Applied Sciences 7, pp. 3286-3291
7. Yahaya Isah Shehu, Ariel Ruiz-Garcia, Vasile Palade, Anne James, “Sokoto Coventry Fingerprint Dataset”, https://arxiv.org/abs/1807.10609v1
8. Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". IEEE Trans. Sys. Man. Cyber. 9 (1): 62–66.
9. Zhou Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, April 2004, doi: 10.1109/TIP.2003.819861