Segmentation of Fingerprint Impressions and Signatures

Ankit Saxena
J.Shakthi Yokesh

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, India

Segmentation of Fingerprint Impressions and Signatures

Project submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Ankit Saxena

(Roll 108CS031)

J.Shakthi Yokesh

(Roll 108CS068)

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela – 769 008, India

May 2012

Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, India. www.nitrkl.ac.in

Dr. Pankaj Kumar Sa
Assistant Professor

May 14,2012

Certificate

This is to certify that the work in the Project entitled *Segmentation of Fingerprint Impressions and Signatures* by *Ankit Saxena & J.Shakthi Yokesh*, is a record of an original research work carried out by them under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Bachelor of Technology* in *Computer Science and Engineering* at the National Institute of Technology,Rourkela. Neither this Project nor any part of it has been submitted for any degree or academic award elsewhere.

Pankaj Kumar Sa

Acknowledgment

This Project has benefited in various ways from several people. Whilst it would be simple to name them all, it would not be easy to thank them enough.

We would like to gratefully acknowledge the enthusiastic supervision and assistance of Prof. Pankaj Kumar Sa throughout this work. His constant support and unstinting guidance has been always been an immense source of motivation and encouragement.

We are very much indebted to Prof. A.K.Turuk, Head-CSE and Prof. P. M. Khilar for allotting us this project and also for the resources and facilities that were made available to us whenever we needed the same.

Thanks are due to Mr. Sambit Bakshi, Mr. Kalyan Singh Hati and other staffs of the Image Processing Lab of NIT, Rourkela for graciously providing us with all facilities and access to the resources vital to the completion of this project.

Our sincere thanks to Prof. S.K. Rath, Prof. S. K. Jena, Prof. B. D. Sahoo, Prof. Banshidhar Majhi, Prof. D. P. Mohapatra, Prof. S. Chinara, Prof. R. Baliarsingh, Prof. K. S. Babu, Prof. M. N. Sahoo, Prof. S. Mohanty, Prof. R.Mohapatra and Prof. S. K. Panigrahi for being our knowledge resource. Their help can never be penned in words.

We would like to thank all our friends for helping us and would also like to thank all those who have directly or indirectly contributed in the success of our work.

Last but not the least, big thanks to NIT Rourkela for providing us such a platform where learning has known no boundaries.

Ankit Saxena
J.Shakthi Yokesh

Abstract

Organizations like banks, post office, passport office, civil registrations get hundreds of applications daily for enrollment into their systems. Some of the most common data they require for enrollment are photographs, signatures, fingerprint impressions etc. These data require scanning by using a desktop scanner and subsequent precise cropping to remove unnecessary white spaces. A significant amount of labor is incurred when the amount of data is in the order of hundreds. In this project the aforementioned enrollment process has been automated by using simple image processing algorithms. The boundaries of each fingerprint impression are determined by using the scan lines followed by the region of interests for segmenting the fingerprints. In another work, different morphological operations are used to identify the fingerprint impressions. Segmentation becomes complex when the impressions do not follow any particular order. To counter such problems connected components of a pixel are used for segmenting the impressions.

Contents

List of Figures

Chapter 1

Introduction

1.1 Image Processing

Vision is the most advanced sense of human being. So, undoubtedly images play an important role in the human perception. Image processing automates the process of gathering and processing visual information. The process of image processing by a digital computer is known as Digital Image Processing. An image may be defined as a two-dimensional function f(x,y) where x and y are spatial co-ordinates and the amplitude f denotes the intensity of the image in a particular co-ordinate (x,y).When the spatial co-ordinates and the amplitude are finite, such an image is called Digital Image. Digital Image processing may be either of these two types:

- Both input and output are images.
- Input is an image and output is the attributes of the image.

Following is the list of different image processing functions that are widely used

- Image Acquisition
- Image Enhancement
- Image Restoration

- Color Image Processing
- Multi-resolution Processing
- Image Compression
- Morphological Processing
- Image Segmentation
- Representation and Description
- Object Recognition

For the first seven functions, both input and output are images whereas for the rest three functions input is an image and output is the attributes of the image. Except image acquisition and display most of the Image processing functions is implemented in software.

Out of the ten different sub-categories of digital image processing, this Project deals with image segmentation. Here, various segmentation methodologies are used and various inputs are restored using these methods.

1.2 Image Segmentation

Image segmentation is the process of dividing an image into multiple parts. It is the process of segmenting an image into non-intersecting regions such that each segmented region Is homogeneous and no two adjacent regions are homogeneous. This is typically used to identify objects or other relevant information in digital images. Most scanned or clicked images have a huge file size (in Megabytes) which can be reduced to kilobytes so that the image can load faster using image segmentation. The goal of segmentation is to segment the given image to make it more meaningful and easier to analyze. Image Segmentation is mainly used to locate objects or curves in an image. Segmentation of non-trivial images is one of the most challenging tasks in image processing. The problem of segmentation is a well-studied one in literature and there are various approaches used to achieve them. Different algorithms are suited for different types of images and it is very difficult to measure the quality of the algorithm for a particular image quantitatively because various segmentation algorithms may produce correct segmentation result for a particular image. The various segmentation algorithms used for mono-chromatic images are mainly based on the two basic properties of image intensity

- Discontinuity
- Similarity

In the first category, we actually partition the image based on abrupt changes in the intensity values. In the second category, we partition the image into regions which have similar predefined criteria.

In our project, we have applied various Image segmentation techniques to segment fingerprints and signatures. This could be applied in various organizations like banks, post offices, passport office and civil registration office where every day hundreds of signatures and fingerprints gets enrolled and it has to be processed faster. By applying these algorithms in the aforementioned organizations, significant labor is avoided. In this project the aforementioned enrollment process have been automated using simple image segmentation algorithms.

1.3 Fingerprint Impressions and Signatures Used

The fingerprint ink impressions of thumb and fore finger were taken three times each for each and every person in the class in a pre-defined form and scanned the page in a scanner with 600 DPI settings and used that image(Figure 1.1) as an input for our project. The scanned image contained 24 fingerprints in an ordered manner .The signatures were also collected from some people in a pre-defined form three times for each person and scanned the page in a scanner with 300 DPI settings and used

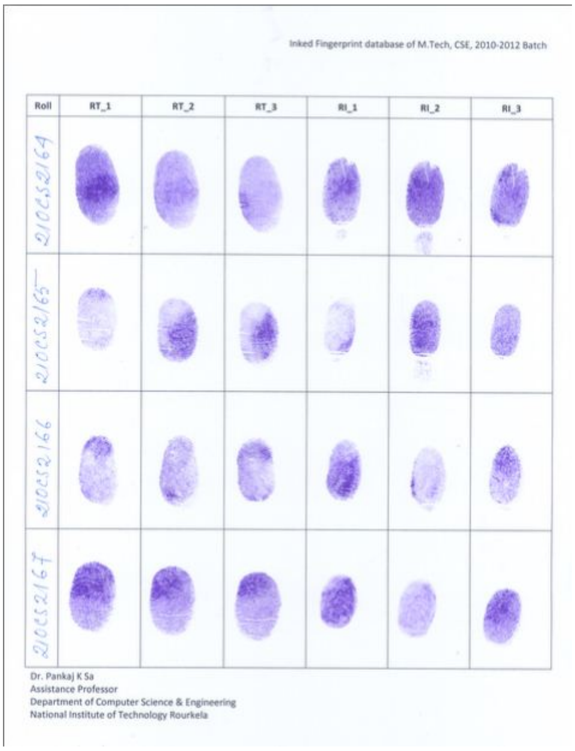

Inked Fingerprint database of M.Tech, CSE, 2010-2012 Batch
Roll
RT_1
RT_2
RT_3
RI_1
RI_2
RI_3
210CS2164
210CS2165
210CS2166
210CS2167
Dr. Pankaj K Sa
Assistance Professor
Department of Computer Science & Engineering
National Institute of Technology Rourkela

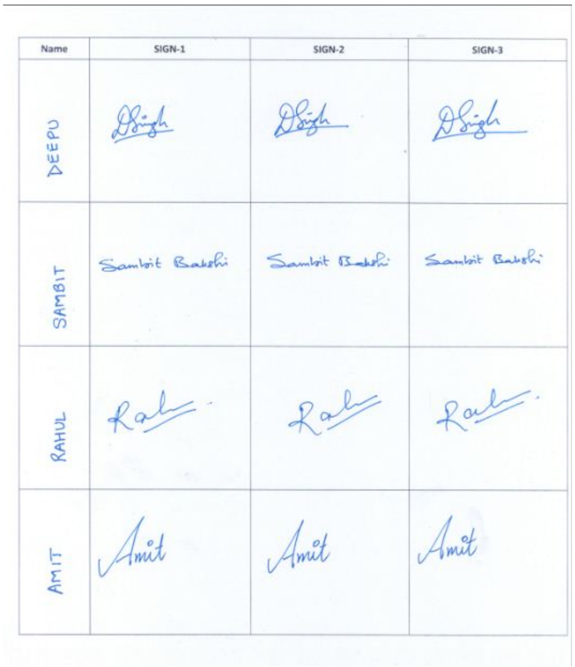
Figure 1.1: Multiple fingerprints in an ordered fashion

that image(Figure 1.2) as an input for our project. An image(Figure 1.3) in which fingerprints were at random positions was also used as an input in our project.

1.4 Problem Definition

The most common problems faced making it necessary for the image to be cropped are listed below

- The image size could be larger than what is required making it necessary for the image to be cropped for use.
- Most scanned or clicked images have a huge file size (in Megabytes) which has to be reduced to kilobytes so that the image can load faster.
- The image may have unnecessary background details which must be removed.



Name	SIGN-1	SIGN-2	SIGN-3
DEEPU	DSingh	DSingh	DSingh
SAMBIT	Sambit Bakshi	Sambit Bakshi	Sambit Bakshi
RAHUL	Rahul	Rahul	Rahul
AMIT	Amit	Amit	Amit

Figure 1.2: Multiple Signatures in an ordered fashion

The enrollment data such as fingerprints, signatures which comes for enrollment in organizations like banks, passport office, post office, civil registration office require scanning by using a desktop scanner and subsequent precise cropping to remove the unnecessary white spaces. A significant amount of labor is incurred when the amount of data is in hundreds. For example, if there are 500 fingerprints we have to either scan five hundred times or we can scan once and crop it 500 times.

1.5 Motivation

The main motivation of this project is to automate the aforementioned scanning process using simple image processing algorithms. We divided our objectives into three sub parts and they are as follows

- Segmenting a single image by ignoring the white spaces and noises.
- Segmenting a group of fingerprints which could be ordered or in some random

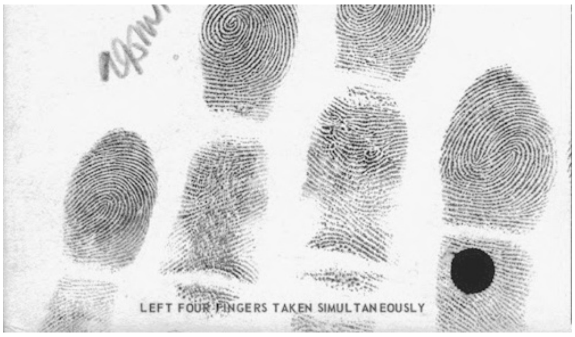


Figure 1.3: Multiple fingerprints at random locations

position and storing them separately.

- Segmenting a group of signatures and storing them separately.

1.6 Project Organization

The rest of the Project is as follows

- **Chapter2** introduces various algorithms (without using connected components) for image segmentation and discusses the steps involved in those algorithms and the results obtained from them.
- **Chapter3** introduces algorithm which uses connected components and signature segmenting algorithms and describe them in detail, steps involved in it and the results obtained from it.
- **Chapter4** presents the concluding remarks with scope for further research work.

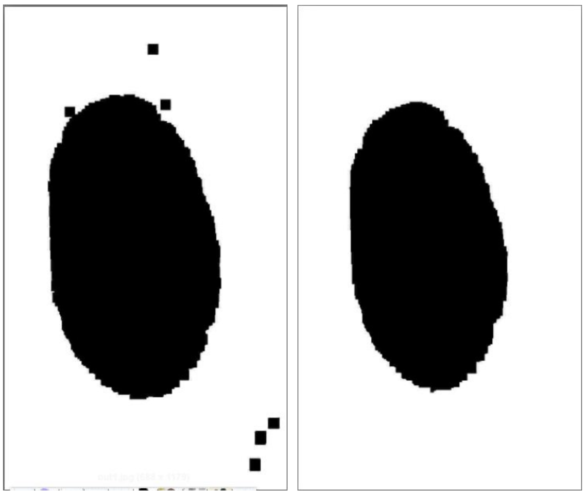
Chapter 2

Image Segmentation using Morphological Processing

2.1 Segmenting a single fingerprint image

The steps for segmenting a single fingerprint image is as follows

1. Load the image and convert it into binary image.
2. Do morphological operations and get the blob of the fingerprint which helps in noise removal (Figure 2.1).
3. Scan the image top to bottom row-wise until you get a group of black pixels and store the location in an array.
4. Scan the image left to right column-wise until you get a group of black pixels and store the location in an array.
5. Scan the image right to left column-wise until you get a group of black pixels and store the location in an array.
6. Scan the image bottom to top row-wise until you get a group of black pixels and store the location in an array.



(a) (b)

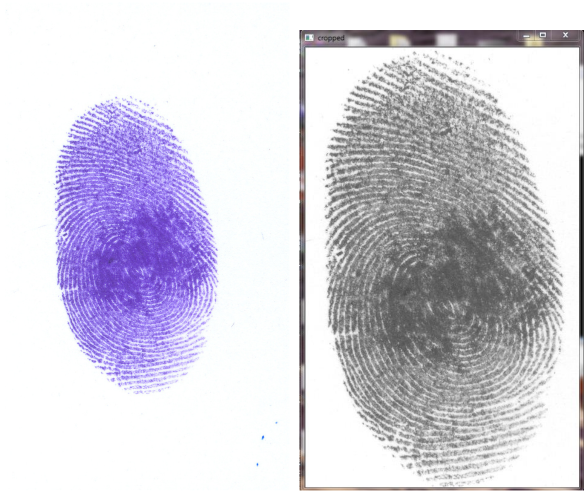
Figure 2.1: (a) Image after Erosion, (b) Dilation of the eroded image.

7. Using these four co-ordinates draw a bounding box around the fingerprint using ROI(Region of Interest).

The limitations of this algorithm are as follows

1. This algorithm works only when an image contains single fingerprint.
2. It fails when there more than one fingerprint.

To overcome these limitations another algorithm is proposed which works well for multiple fingerprints along with lines which are ordered



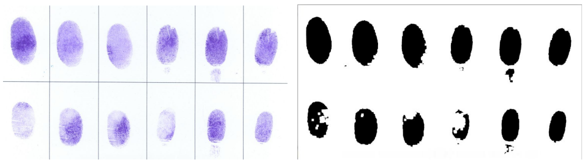
(a) (b)

Figure 2.2: (a) Input Image, (b) Output Image.

2.2 Segmenting a group of ordered fingerprints with lines

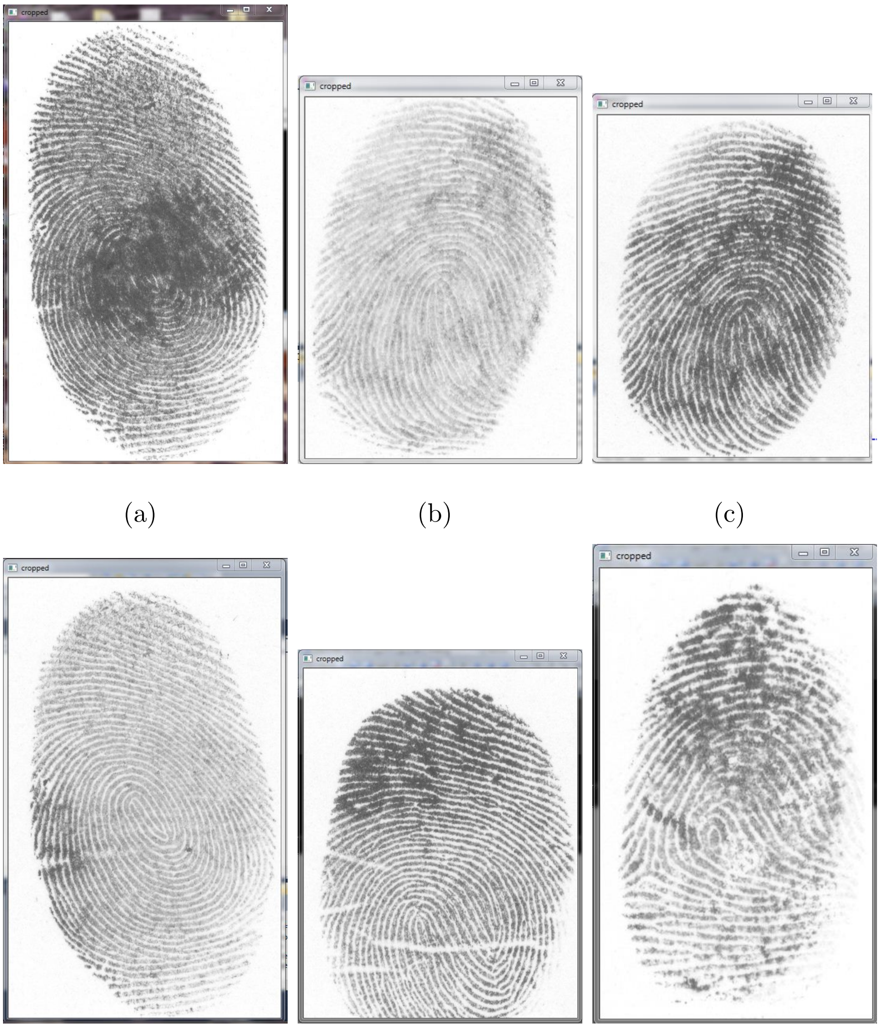
The steps for segmenting multiple fingerprints in an ordered fashion with lines are as follows

1. Load the image and convert it into binary image.
2. Do morphological operations and get the blob of the fingerprints which helps in noise removal and removal of lines.
3. Scan the image top to bottom row-wise until you get a black pixel and mark it as a1.
4. Continue scanning until you get a row in which there is no black pixel and mark it as a2.
5. Scan the image left to right column-wise until you get a black pixel and mark it as X1.
6. Continue scanning until you get a column in which there is no black pixel and mark it as X2.
7. Continue this process until you cover all the fingerprints
8. Again scan top to bottom row-wise from a1 to a2 and when the first black pixel is found between Xi and Xj mark it as Yi.
9. Scan bottom to top row-wise from a2 to a1 and when the first black pixel is found between Xi and Xj mark it as Yj.
10. After you get Xi, Xj, Yi and Yj you can draw bounding boxes around the fingerprints using these co-ordinates using ROI for all available values of i and j.



(a) (b)

Figure 2.3: (a) Before Morphological Operations, (b) After Morphological Operations.



(a) (b) (c)

(d) (e) (f)

Figure 2.4: Sample outputs

The limitation of this algorithm is as follows

1. This method fails when there are some texts present in the image along with the group of fingerprints.

2. This method also fails when the fingerprints are in a random order.

The first limitation can be overcome by using the following algorithm which ignores the text which is present in the image.

2.3 Segmenting fingerprint images which contains group of texts

The previous algorithm is applied in this case also. But it will crop the group of texts also. We can remove it by calculating the aspect ratio of the fingerprint. The aspect ratio of the fingerprint would be above a certain threshold value (in our case it is between 1.6 and 2). But still this fails when the fingerprints are in a random order.

To segment the fingerprints which are located in random order, we use connected components. The algorithm which uses connected components to segment the image is described in detail in the next chapter.

Chapter 3

Image segmentation using connected components

3.1 What are connected components?

In image processing, connected components are the regions of connected pixels which have the same intensity value and follow pixel connectivity rules.

Pixel Connectivity Rules:

Connectivity is a simple relationship between pixels which pixels are next to which others?

Suppose we consider as neighbors only the four pixels which share an edge with the pixel (x, y) in question: (x+1, y), (x-1, y), (x, y+1) and (x, y-1). These are called 4-connected neighbors of the pixel (x,y).

Consider the following case

The black pixels on the main diagonal in Fig. 3.2 are not 4-connected. However, they act as an efficient insulator between the two sets of white pixels, which are also not 4-connected across the black pixels. This creates topological anomalies. An alternative is to consider a pixel as connected not just pixels on the same row or column, but also the diagonal pixels. The four 4-connected pixels with the diagonal

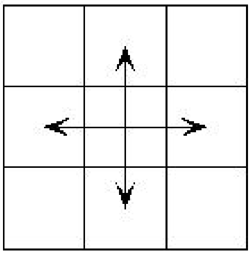


Figure 3.1: Four Connected Neighbors

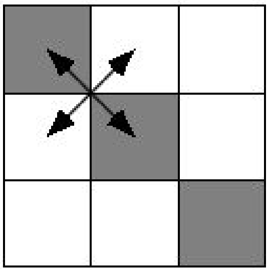


Figure 3.2: Paradox of 4 connected neighbors

pixels are called 8-connected neighbors.

Again, following topological anomaly occurs in the case shown in Figure 3.2. The black pixels on the main diagonal are connected, but again so are the white background pixels. Few pixels are connected across the links between other connected pixels! The solution is to use 4-connectivity for the foreground with 8-connectivity for the background or vice versa.

Another type of connectivity is mixed-connectivity (Fig. 3.5, a type of 8-connectivity that considers diagonally-adjacent pixels to be connected if no shared 4-connected neighbor exists. (In other words, follow 4-connectivity where possible and 8-connectivity where not.)

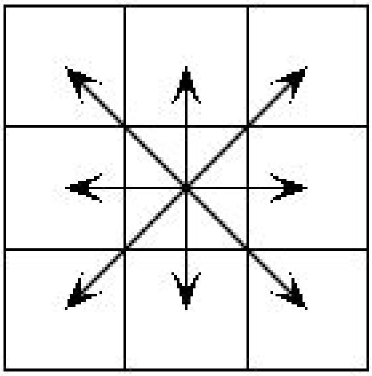


Figure 3.3: "8 connected" neighbors

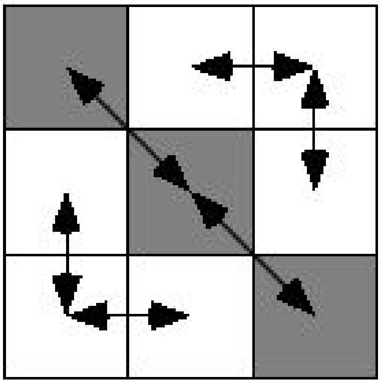


Figure 3.4: One sample solution suggested

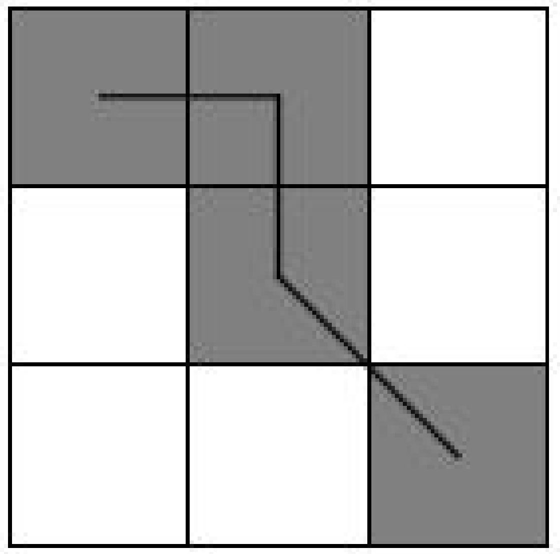


Figure 3.5: Mixed Connectivity

3.2 Connected Component Detection

Connected component detection works by scanning an image, pixel-by-pixel (from top to bottom and left to right) to identify connected pixel regions, i.e. regions of adjacent pixels which share the same set of intensity values V. (For a binary image V=1; But, in a gray-level image V will take a range of values, for example: V=51, 52, 53, ..., 77, 78, 79, 80.)

Connected component detection works on binary or gray-level images and different possible measures of connectivity. However, in our case we assume binary input images and 8-connectivity. The connected components detection operator scans the image by moving along a row until it comes to a point p (where p denotes the pixel to be detected at any stage in the scanning process) for which V=1. When this is true, it identifies the four neighbors of p which have already been covered in the scan (i.e. the neighbors (i) to the left of p, (ii) above it, and (iii and iv) the two upper diagonal terms). After this, the labeling of p occurs as follows:

- If all four neighbors are 0, assign a new label to p, else
- if only one neighbor has V=1, assign its label to p, else
- if more than one of the neighbors have V=1, assign one of the labels to p and make a note of the equivalences.

After the scan, the equivalent label pairs are sorted into equivalence classes and a unique label is assigned to each class. As a final step, a second scan is made through the image, during which each label is replaced by the label assigned to its equivalence classes. For display, the labels may be different gray-levels or colors.

3.3 Segmenting the fingerprint images at random location

The steps for segmenting the images at random location are as follows

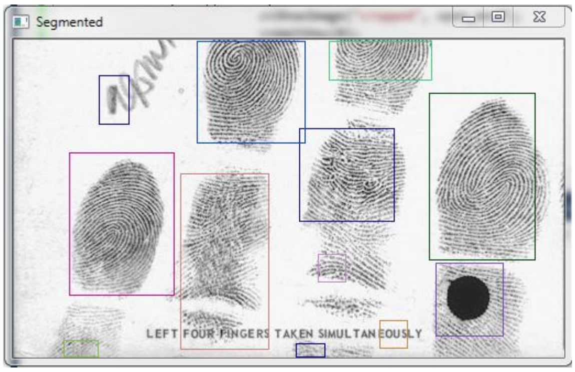
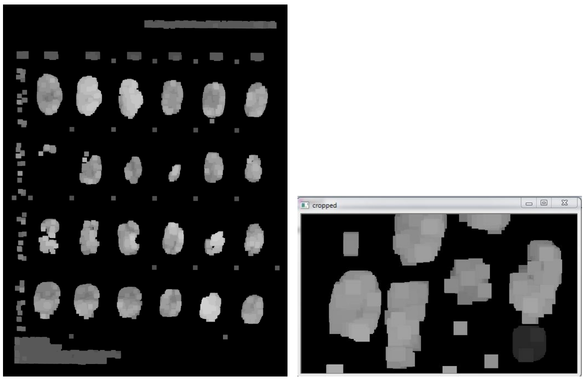


Figure 3.6: Connected Components Detected

1. Load the image in RGB.
2. Initializing the variables for contours.
3. Create storage needed for contour detection.
4. Smooth, threshold and apply morphological filters to the image.
5. Duplicate image for contour, Search contours in pre-processed image , Optimize contours, reduce points.
6. For each contour found, Generate Random colors which is used to draw the rectangles around the letters.
7. We can draw the contour of object or detect bounding rectangle of contour.
8. Set ROI from the bounding rectangle by applying suitable noise removal filters.

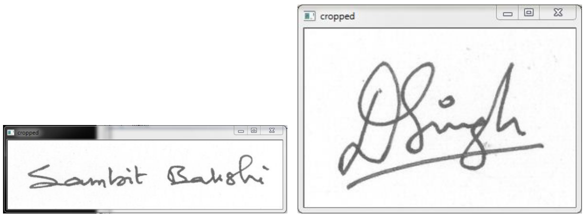


(a) (b)

Figure 3.7: Partial output of segmentation using connected components

3.4 Segmenting signatures

Segmenting a signature is far more different than segmenting a fingerprint. A signature cannot be differentiated from the simple text by merely on aspect ratio basis. Sometimes signature is not a single word but a group of words with white spaces. It may contain special characters like dots, lines etc. It may be in different languages also, so we need at-least some predefined form like a table to segment signatures. Signatures in random positions cannot be differentiated from simple text even by connected component analysis.



(a) (b)

Figure 3.8: Sample Outputs

Chapter 4

Conclusions

The work in this Project primarily focuses on segmentation of fingerprints and signatures in scanned images. The work reported in this Project is summarized in this chapter. Section 4.1 lists the achievements of the work. 4.2 lists the limitations and 4.3 provide some scope for further development

4.1 Achievements

Methods for segmenting images with and without connected components were studied under great detail and implemented. The method using connected components was found to tackle the segmentation problems of the scanned fingerprints, when the fingerprints are in random order.

4.2 Limitations of the work

The following limitations were encountered during the course of this project:

- The methods without connected components were not able to segment the fingerprints when they are located in random order.

- The method with connected components could not segment fingerprints with less amount of ink.
- The method with connected components could not segment signatures located in random order.
- This method also cannot efficiently segment the signatures if the signature contains date

4.3 Further Development

This method with connected components can be extended to consider fingerprints with less amount of ink .A technique can also be developed for segmenting signatures located in random order and also those containing dates along with signatures.

Bibliography

[1] Anil K. Jain. *Fundamentals of Digital Image Processing.* Prentice-Hall of India, 1989.

[2] Rafael Gonzalez and Richard Woods. *Digital Image Processing.* Addison Wesley, 1992.

[3] Gary Bradski and Adrian Kaehler. *Learning OpenCV.* O'REILLY.

[4] *OpenCV Documentation.* 2009.

[5] Opencv yahoo groups. `http://tech.groups.yahoo.com/group/OpenCV/`.