A MiniProject Phase 1 Report on

AUTOMATIC CONVERSION OF HANDWRITING FORMAT TO DIGITAL FORMAT

Submitted in partial fulfilment for the
degree of Bachelor of Technology in
Information Technology

Submitted by
Harshitha Aditham[01]
Rajeshwari Nadar[26]

Under the guidance of
Prof.Anita More

USHA MITTAL INSTITUTE OF TECHNOLOGY
2019

CERTIFICATE

This is to certify that Name of Student has completed the Harshitha Aditham and Rajeshwari Nadar.Project report on the topic name'AUTOMATIC CONVERSION OF HANDWRITING FORMAT TO DIGITAL FORMAT' satisfactorily in partial fulfillment for the Bachelor's Degree in INFORMATION TECHNOLOGY under the guidance of Prof. Anita More Guide Name during the year 2018-2019 as prescribed by SNDT.

Guide

Prof.Anita More

Head Of Department

Dr.Sanjay Shitole

Principal
Dr.Sanjay Pawar

Examiner 1

Examiner 2

Contents

1 INTRODUCTION **1**

2 REVIEW OF LITERATURE **2**
- 2.0.1 Early Scanners . . . 2
- 2.0.2 To the Digital Age . . . 2
- 2.0.3 Machine Learning . . . 2

3 Steps for Converting Handwritten to Digital format **3**

4 Step:1 NOISE REMOVING **5**
- 4.0.1 Non-local means algorithm . . . 5
- 4.0.2 Image Denoising with OpenCV . . . 5

5 Step:2 BINARIZATION **7**
- 5.0.1 Otsu Binarization Algorithm . . . 7
- 5.0.2 Histogram Analysis . . . 7
- 5.0.3 THRESHOLD IMPLEMENTATION . . . 9
- 5.1 Guassian Thresholding Algorithm . . . 9
- 5.2 THRESHOLD implemented image Analysis . . . 9

6 Step:3 EDGE DETECTION **11**
- 6.1 Algorithm for Canny Edge detection . . . 13

7 Step:4 DILATION **15**
- 7.1 Algorithm for erosion algorithm . . . 15

8 Step: 5 FILTERING **17**

9 Step: 6Character Segmentation **18**

10 SOFTWARE REQUIREMENTS AND INSTALLATION **22**

11 ADVANTAGES **23**

12 DISADVANTAGES **25**

13 APPLICATIONS **26**
- 13.1 REFERENCES: . . . 26

List of Figures

3.2 Handwritten conversion System 3
3.3 Context Level DFD diagram 4
3.4 level 1 dfd diagram 4

4.1 noise less image 6

5.1 Otsu binarized image 8
5.2 histogram of the threshold image 8
5.3 analysis using threshold values 10

6.1 Canny Edge detected Image 12
6.2 Canny Edge detected Image 12
6.3 Canny Edge detected Image 14

7.1 dilated image 16

9.1 spacing between the words 18
9.2 charecter segmented 19
9.3 charecter segmented 19
9.4 charecter segmented 19
9.5 divided text to words flowchart 20
9.6 dividing text into words 20
9.7 dividing rows into words 20
9.8 dividing words into charecters 21
9.9 the output of all the steps in Conversion 21

Abstract

Recognition of Handwritten English alphabets have been broadly studied in previous years. Optical character recognition (OCR) method has been used in converting printed text into editable text. OCR is very useful and popular method in various applications. Accuracy of OCR can be dependent on text pre-processing and segmentation algorithms. This project seeks to classify an individual handwrittenword so that handwritten text can be translated to a digital form. We used two main approaches to accomplish this task: classifying words directly and character segmentation. For the former, we use Convolutional Neural Network (CNN) with various architectures to train a model that can accurately classify words.

Chapter 1

INTRODUCTION

Everything is going digital these days, even currencies. While most companies, and students are increasingly using their smartphones to scan and send and receive important documents, what about handwritten notes. How do you turn your handwriting into digital notes. The handwriting to text is an idea which can be performed with ease and satisfaction. To convert handwriting to text this is undoubtedly the best program which can be used to overcome the problems which the other programs present. The process to convert handwritten notes to text which is followed by this program is awesome. Although Digital text is easier to edit, search and store, handwriting is still commonly used, since it's fast, easy and accessible. handwriting recognition and converting it to digital format is a great solution for those who still like the feel of pen on paper, but want to enjoy the benefits of digital. It is easy to use, fast and affordable. A perfect solution for students, with school notes and homework, in high school, college or university.

• Scan, recognize and convert handwritten notes into digital text that can be edited, searched in and stored on any device or cloud service.
• Any handwritten notes, such as letters, school notes, diaries, meeting notes and protocols, grocery lists, recipes etc. can now be scanned by your mobile device and be converted, by our handwriting recognition engine, into text, available for use in any digital platform.

Chapter 2

REVIEW OF LITERATURE

Recognition of Handwritten English alphabets have been broadly studied in previous years. Optical character recognition (OCR) method has been used in converting printed text into editable text. OCR is very useful and popular method in various applications. Accuracy of OCR can be dependent on text pre-processing and segmentation algorithms. This project seeks to classify an individual handwritten word so that handwritten text can be translated to a digital form. We used two main approaches to accomplish this task: classifying words directly and character segmentation. For the former, we use Convolutional Neural Network (CNN) with various architectures to train a model that can accurately classify words.

2.0.1 Early Scanners

The first driving force behind handwritten text classification was for digit classification for postal mail. Jacob Rabinows early postal readers incorporated scanning equipment and hardwired logic to recognize mono-spaced fonts

2.0.2 To the Digital Age

The first piece of software OCR used a more developed used for the matrix method (pattern matching). Essentially, this would compare bitmaps of the template character with the bitmaps of the read character and would compare them to determine which character it most closely matched with. To improve on the templating, OCR software began using feature extraction rather than templating.

2.0.3 Machine Learning

It next focused on using gradient-based learning techniques using multi-module machine learning models.The next major upgrade in producing high OCR accuracies was the use of a Hidden Markov Model for the task of OCR. This approach uses letters as a state, which then allows for the context of the character to be accounted for when determining the next hidden variable. Handwritten alphabets recognition system's precision of any image relay upon the sensitivity of the choice of features and category of classifier utilized. Therefore, so many feature removal and categorization techniques could be establish in the literature. Feature vector is exclusively created from image to utilizing arithmetical trusts among character and features, partially calculated characters are identified by evaluating through lexicon.Applied CNNs to the problem of taking text found in the wild (signs, written, etc)

Chapter 3

Steps for Converting Handwritten to Digital format

The main Methodology of the project is to review the idea behind making Handwriting System. Studying and resolving all the issues regarding Algorithms used for HRS (Handwriting Recognition System) in previous few years. The algorithm used not only accelerates the process but also increases the probability of detecting the detecting the alphanumeric character and extraction of characters, under certain set of constraints. Character segmentation performs an important role. The result shows higher accuracy of region removal other than character region and thus it results a better recognition of each letters and numbers after the segmentation process.

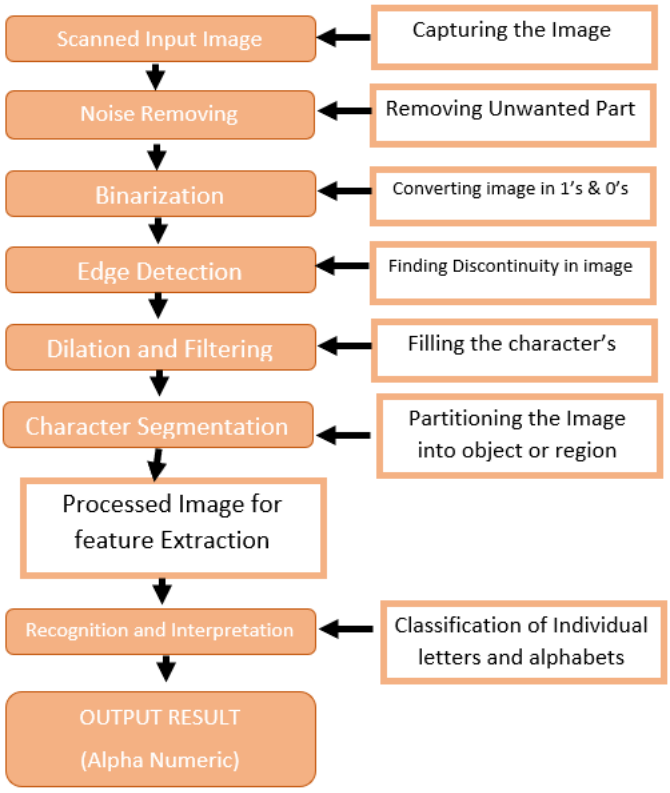

Figure 3.2: Handwritten conversion System

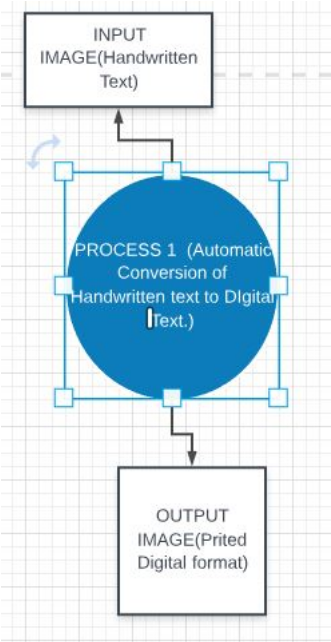

Figure 3.3: Context Level DFD diagram

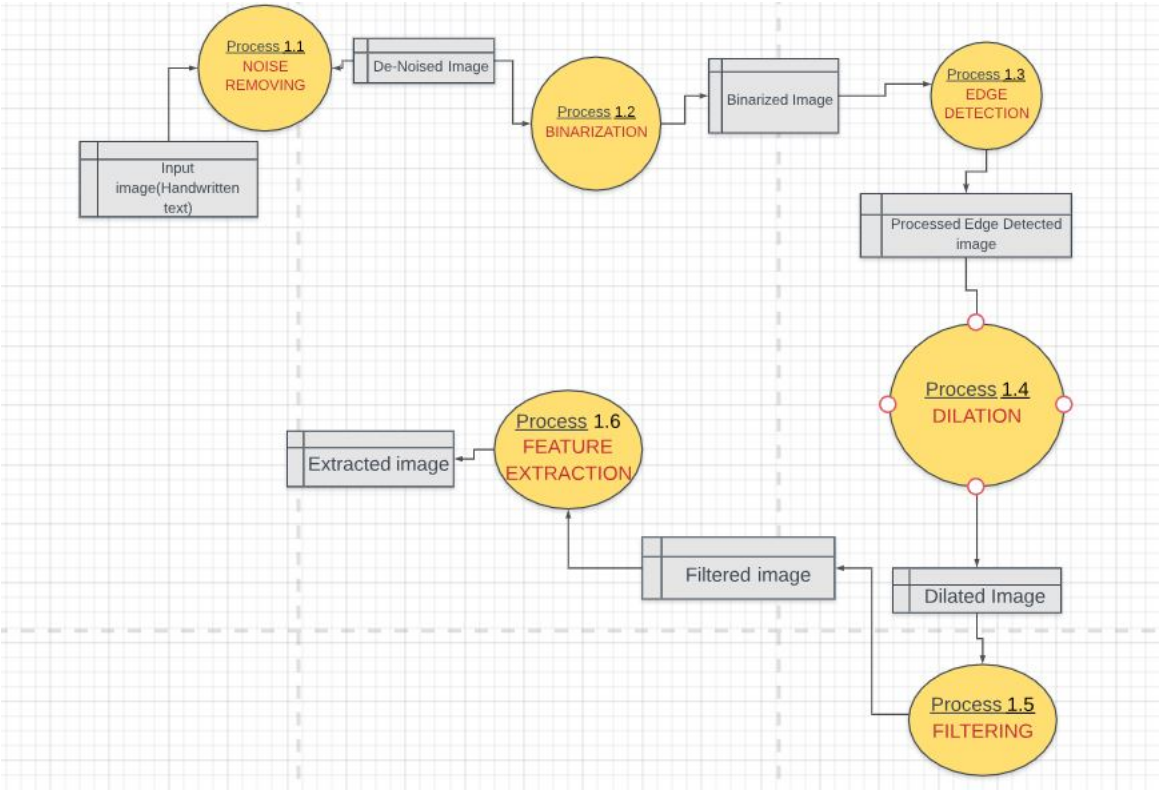

Figure 3.4: level 1 dfd diagram

Chapter 4

Step:1 NOISE REMOVING

One of the fundamental challenges in image processing and computer vision is image de noising. What de noising does is to estimate the original image by suppressing noise from the image. Image noise may be caused by different sources (from sensor or from environment) which are often not possible to avoid in practical situations. Therefore, image de noising plays an important role in a wide range of applications such as image restoration, visual tracking, image registration, and image segmentation. While many algorithms have been proposed for the purpose of image denoising, the problem of image noise suppression remains an open challenge, especially in situations where the images are acquired under poor conditions where the noise level is very high.

4.0.1 Non-local means algorithm

Step 1: Input the image.
Step 2: Find the repeated non local neighbourhood subimage.
Step 3: Find means of subimages.

p1 = p0 +n1

p2 = p0 +n2

. . . .

pN = p0 + nth

P=p1 + p2 +.....+pN/N

Step 4: Replace the pixels with the repeated non local subimages

4.0.2 Image Denoising with OpenCV

In this section, we'll use**cv2.fastNlMeansDenoisingColored()**
function which is the implementation of **Non-local Means Denoising Algorithm**.

The parameters are:

src : Input 8-bit 3-channel image.
dst : Output image with the same size and type as src .
h : Parameter regulating filter strength for luminance component. Bigger h value perfectly removes noise but also removes image details, smaller h value preserves details but also preserves some noise.
templateWindowSize: Size in pixels of the template patch that is used to compute weights. Should be odd. Recommended value 7 pixels

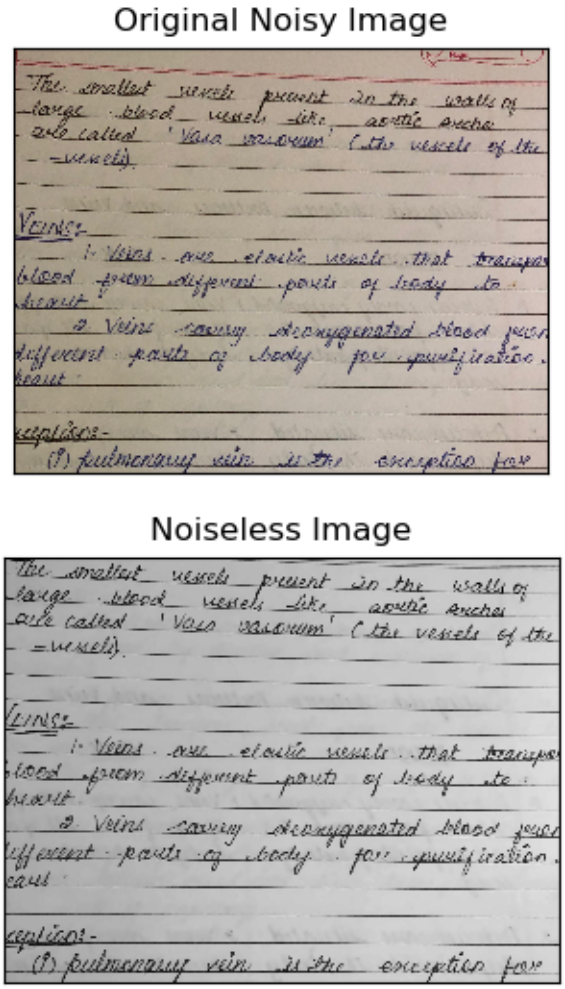

Figure 4.1: noise less image

searchWindowSize : Size in pixels of the window that is used to compute weighted average for given pixel. Should be odd. Affect performance linearly: greater searchWindowsSize - greater denoising time. Recommended value 21 pixels

Chapter 5

Step:2 BINARIZATION

The Otsu's binarization Now that you've seen when an image is bimodal and you are in possession of the tools to recognize it as such, you can see in detail the **Otsu's binarization.**

Returning to the previous figure, the bimodal image, presents the histogram into two distinct distributions, almost separable between them. In fact between the two mode there is a minimum point, where you might consider the possibility of separating the histogram into two parts. Well the Otsu's binarization helps you to automatically get that value.

This algorithm will allow you to quickly and automatically obtain the correct threshold value to choose between two histogram mode, so as to apply the thresholding in an optimal manner.

In OpenCV, the application of the Otsu's binarization is very simple. It will be sufficient to add as parameter within the **cv2.threshold ()** function, called

cv2.THRESH$_OTSY$

5.0.1 Otsu Binarization Algorithm

Step 1: Plot a Histogram on the basis of pixels value of the image
Step 2: Finding probabilitis of each pixels value seperately.
Step 3: Find mean variance
Step 4: Within Class Variance:
if pixels are classified into N classes categories turn the within class variance

where Wi is (of pixels in class i)/(totla pixel)

Step 5: Between Class Varinace:
if pixels are classified into N classes (categories) then the **between class variance**(Vo)=(Vt-Vw) if pixels are classified into 2 classes, then the between class variance Vt=W1W0(**Step 6:**Otsu Thresholding Find the Threshold 'T' and classify pixels into 2 classes(class 1 and class2) so that the Vw is minimum(Vo is maximum)

5.0.2 Histogram Analysis

Moreover, once this technique gained there will be no need to visualize and study the histogram in order to find the point, but everything will be done automatically.

By running the code this time you'll get the best results. In addition it is possible to see which is the optimal threshold value found by Otsu's binarization within the histogram. As you can see it is between the two modes, but not in the minimum point.

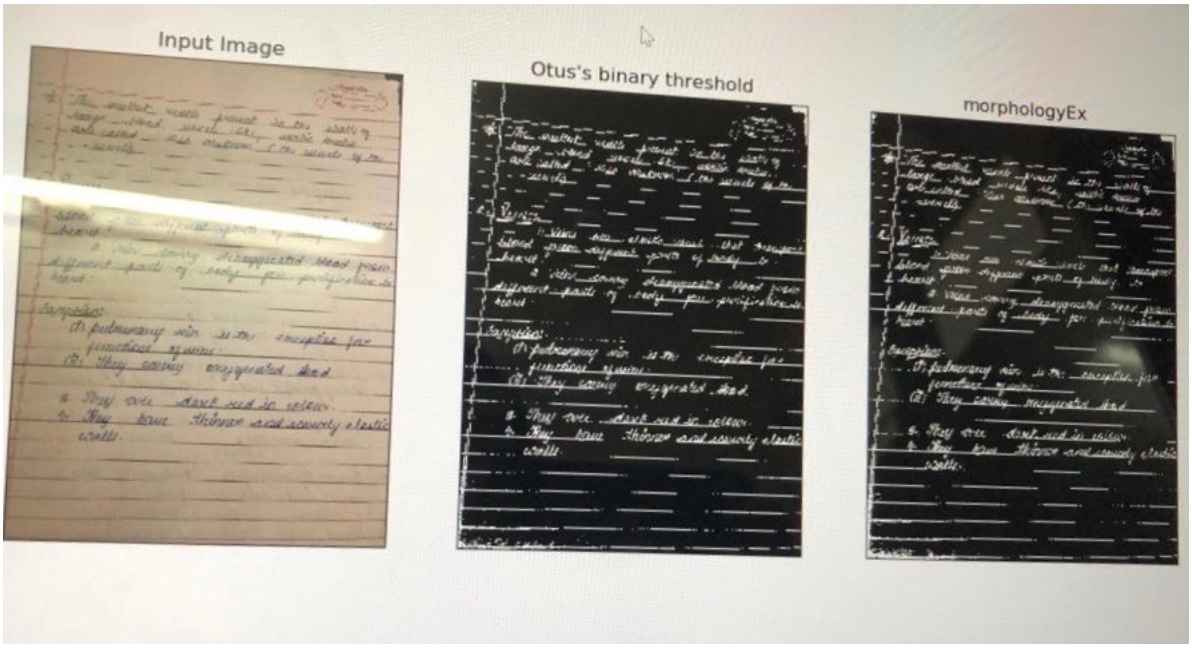

Figure 5.1: Otsu binarized image

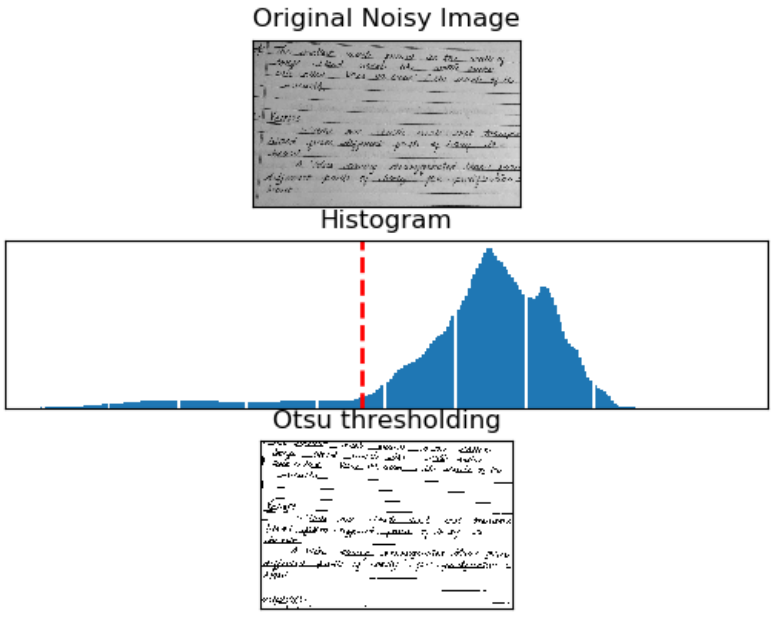

Figure 5.2: histogram of the threshold image

5.0.3 THRESHOLD IMPLEMENTATION

Thresholding
One of the most used techniques for the analysis of the images is that of the thresholding, ie the application of a threshold along a particular scale of values, to filter in some way an image.

One of these techniques is for example the one that converts any image in grayscale (or color) in a totally black and white image. Often this is very useful for recognizing the regular shapes, contours within an image, or even to delimit and divide zones inside, to then be used in a different way in the subsequent processing.

So applied to a histogram, you will choose a value in which all the underlying values will be converted to*0 (white) and all those overlying to 255 (black)*, by converting an image to grayscale into black and white.

In **OpenCV** to perform the thresholding you can use the **cv2.threshold()** function
Take the case of the image of the previous leaf. Make the case that you need to recognize the shape of the leaf, but you can not use a histogram. As a first approach you'll try to apply a threshold (a threshold) at random, and then after several attempts able to find an optimal value.

The first value definitely worth trying is 127, which in the scale of 0-255 is perfectly in the middle. Then you apply this value to the **cv2.threshold()** function.

5.1 Guassian Thresholding Algorithm

Step 1: Input Image
Ste 2: Select Subregion
Step 3: Take weighted mean
Step 4: Mean= Threshold value(T) for the subregion
Step 5: Thresholding
if(f(x,y) ¡T)
f(x,y0 =255
else
f(x,y)=0

Step 6: Output

5.2 THRESHOLD implemented image Analysis

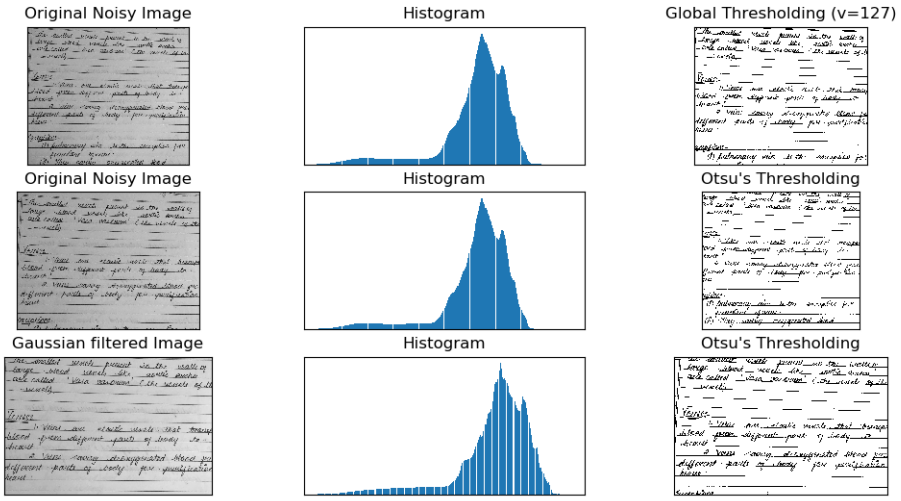

Figure 5.3: analysis using threshold values

Chapter 6

Step:3 EDGE DETECTION

Edge detection is one of the fundamental operations when we perform image processing. It helps us reduce the amount of data (pixels) to process and maintains the structural aspect of the image. We're going to look into many people think it as the ultimate edge detection Canny Edge Detection. With this detector, we get clean, thin edges that are well connected to nearby edges.
The canny edge detector is a 4-step detection process. The steps are:

1. Noise Reduction - 5x5 Gaussian filter

2. Calculating gradients - Finding Intensity Gradient of the Image

3. Non maximum suppression - upper threshold

4. Thresholding with hysterysis - upper/lower threshold

Canny Edge Detection is a popular edge detection algorithm. It was developed by **John F. Canny** in

It is a multi-stage algorithm and we will go through each stages. Noise Reduction

Since edge detection is susceptible to noise in the image, first step is to remove the noise in the image with a 5x5 Gaussian filter. We have already seen this in previous chapters.

Finding Intensity Gradient of the Image

Smoothened image is then filtered with a Sobel kernel in both horizontal and vertical direction to get first derivative in horizontal direction (Gx) and vertical direction (Gy. From these two images, we can find edge gradient and direction for each pixel as follows:

Edge Gradient(G)=

$$G2x + G2yAngle() = tan1(GyGx) \tag{6.1}$$

Gradient direction is always perpendicular to edges. It is rounded to one of four angles representing vertical, horizontal and two diagonal directions.

Non-maximum Suppression

After getting gradient magnitude and direction, a full scan of image is done to remove any unwanted pixels which may not constitute the edge. For this, at every pixel, pixel is checked if it is a local maximum in its neighborhood in the direction of gradient. Check the image below:

Point A is on the edge (in vertical direction). Gradient direction is normal to the edge. Point B and C are in gradient directions. So point A is checked with point B and C to see if it forms a local

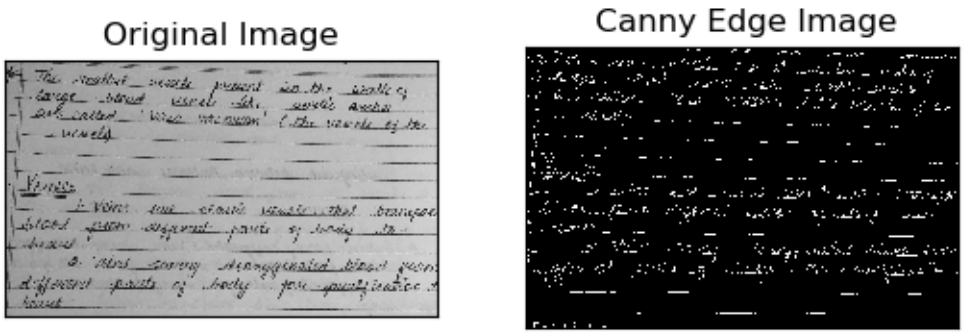

Figure 6.1: Canny Edge detected Image

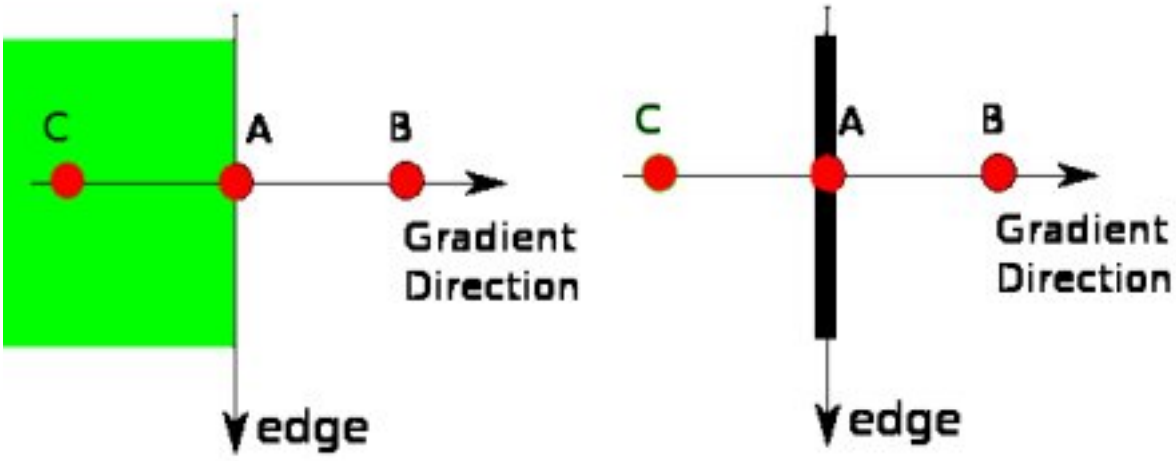

Figure 6.2: Canny Edge detected Image

maximum. If so, it is considered for next stage, otherwise, it is suppressed (put to zero).

In short, the result you get is a binary image with "thin edges".

6.1 Algorithm for Canny Edge detection

"Canny Edge Detection Algorithm" is the optimal alorithm among the edge detection algorithms. The three main criteria's of the canny edge detection are as follows: 1. Low error rate: It is important that edge occurring in image should not be missed and there should be no response for non-edge.
2. Good Localization: The distance between the edge pixels as found by the detector and the actual edge is to be minimum.
3. Single Response: To have one response to a single edge. The algorithm mainly has five steps in it. They are:
Step 1: Computing the horizontal (Gx) and vertical (Gy) gradient of each pixel in an image.
Step 2: Using the above information the magnitude (G) and direction (of the each pixel in the image is calculated.
Step 3: In this step all non-maximas are made as zero that is suppression the non- maximas thus the step is called Non-Maximal Suppression.
Step 4: The high and low thresholds are measured using the histogram of the gradient magnitude of the image.
Step5: To get the proper edge map hysteresis thresholding is employed which will link between the weak and strong edges. The weak edges are taken into consideration if and only if it is connected to one of the strong edges or else it is eliminated from the edge map. The strong edge is the one whose pixel is greater than the high threshold and weak edge is one whose pixel value lays between high and low threshold.

Hysteresis Thresholding
This stage decides which are all edges are really edges and which are not. For this, we need two threshold values, minVal and maxVal. Any edges with intensity gradient more than maxVal are sure to be edges and those below minVal are sure to be non-edges, so discarded. Those who lie between these two thresholds are classified edges or non-edges based on their connectivity. If they are connected to "sure-edge" pixels, they are considered to be part of edges. Otherwise, they are also discarded. See the image below:

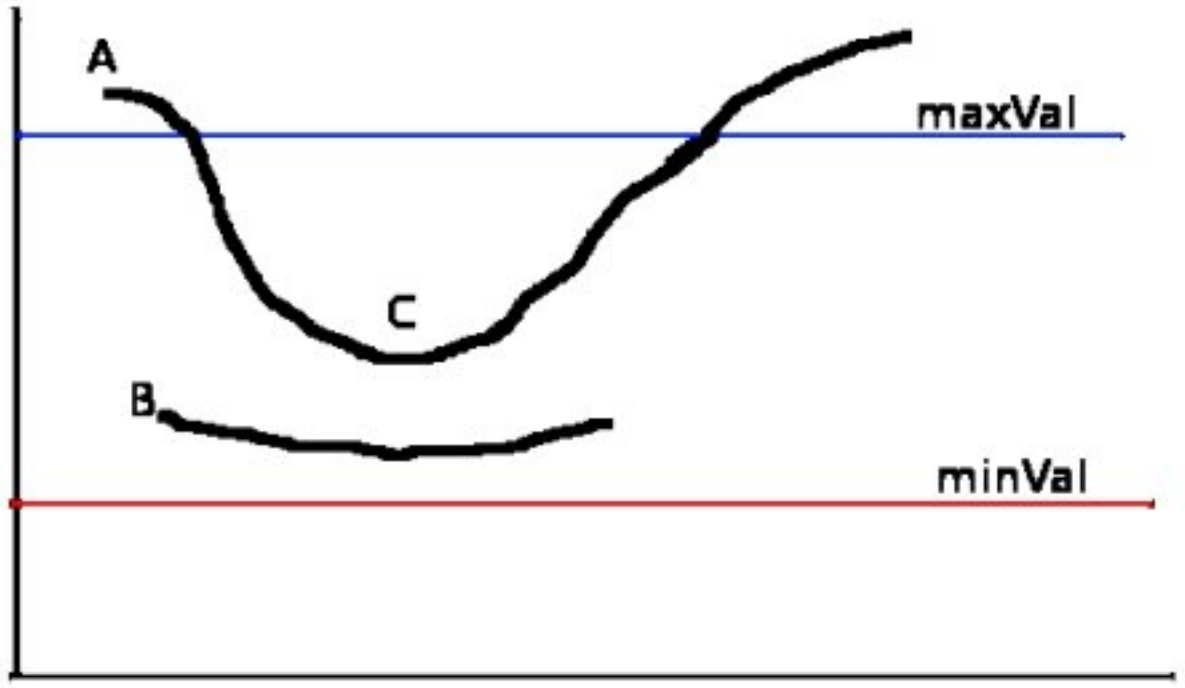

Figure 6.3: Canny Edge detected Image

Chapter 7

Step:4 DILATION

This operations consists of convoluting an image A with some kernel (B), which can have any shape or size, usually a square or circle.

The kernel B has a defined anchor point, usually being the center of the kernel.

As the kernel B is scanned over the image, we compute the maximal pixel value overlapped by B and replace the image pixel in the anchor point position with that maximal value. As we can deduce, this maximizing operation causes **bright regions within an image to grow.** (therefore the name **dilation**).

Here is an example of applying the dilation operation:

7.1 Algorithm for erosion algorithm

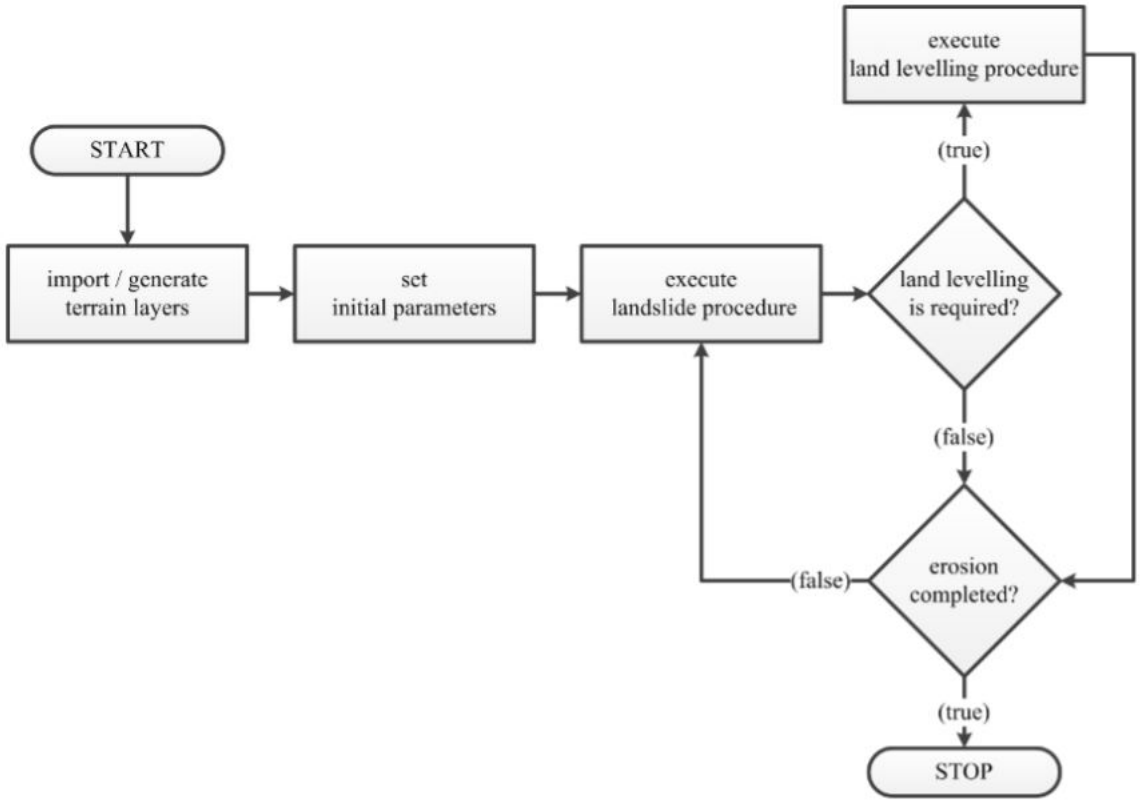

Figure 2. Structure of erosion algorithm.

Algorithm for erosion algorithm Step 1:

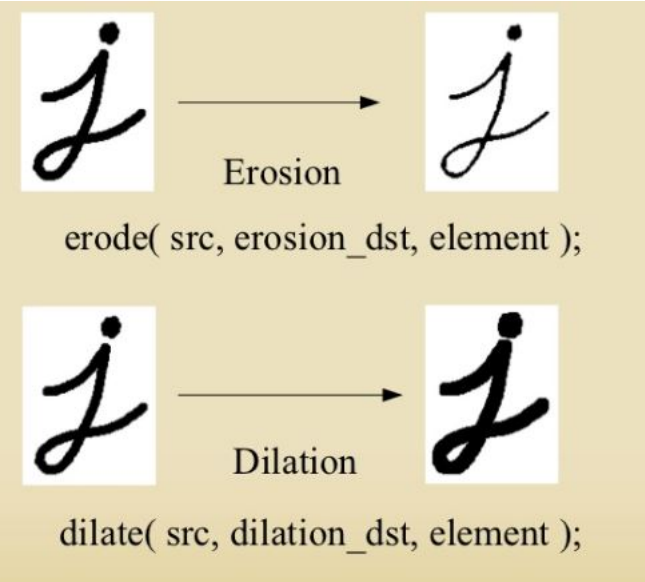

Figure 7.1: dilated image

Chapter 8

Step: 5 FILTERING

Median filters are used when, The image has so-called salt-and-pepper noise or impulse noise. That is, it is corrupted by extreme values or outliers (0s or 255s). Practically speaking, I have seen this occurs most when I've scanned old photographs. We don't want to replace a pixel with an artificially calculated value (mean) which may not exist in the image window. We would rather use a value that actually exists in the window. This point is a bit of an academic quibble.

Mean filters are used when, We actually want to blur the image or part of the image (generally you would use a Gaussian function though, rather than a simple moving average, but the principle of convolution applies in both). Think of a case where you want to blur out someone's face from a photo.
Median filters are computationally more expensive than mean filters. Averaging operation just involves addition and division while median operation involves sorting.

Improved Median Filtering Algorithm
A. Improvement of the filtering mask The filtering mask is mainly u nn square mask or cross mask. Considering of the symmetry of the mask, n is commonly odd. The smaller the mask is, the better the image details are retained, the weaker the noise reduction performance is; the larger the mask is, the less the image details are retained, the stronger the noise reduction performance is. To solve the contradiction, we introduce the adaptive filtering algorithm. In the filtering process, it can adaptively resize the mask according to noise levels of the mask. In the mask, max is the maximum value of gray levels, min is the minimum value of gray levels, average is the average value of gray levels, med is the median value of gray levels, jif),(is the central value of the mask, n is the size of the mask. The adaptive filtering requires two steps: Step 1: adaptively resizing the mask
(1) Initialization: let n= 3 ;
(2) Computation: A1=med-min,A2=med-max
(3) Judgment: if A ! 01 and A !02 , then turn to the step 2; if not, then enlarge the size of the mask, let n=n+2 and turn to (2).
Step 2: median filtering.

The time complexity of the improved algorithm is less than the conventional algorithm. Steps of the improved algorithm are shown as below. (1) The mask slides over the image, overlaps the center of the mask with the pixel on the image to search the center element jif),(; (2) To read the values of the corresponding pixels of the mask; (3) To compute the average value (average) of the mask; (4) To compare the value of each pixel with average, if the value of each pixel is greater than average, then searching the median value and let),(medjif ; otherwise, retaining the original value of the pixel unchanged; (5) Repeating the step (4), until nji .

Chapter 9

Step: 6Character Segmentation

The next steps in the OCR process after the line segmentation, word and character segmentation, isolate one word from another and separate the various letters of a word.

In most cases, separating words is not that hard. There's always a distinct white space between them. (Which means that a word often includes a punctuation symbol.)

The horizontal space between words in a line is called /textbf "interword space". Which leads us to the phenomenon of justification: typographists adjust the interword space and the interletter space to achieve justification! Note how the space between the letters inside a word **("interletter space")** on the 4th line is bigger than the space between two words ("interword space") on the 3rd line...!
Separating characters is a different matter. There are many elements besides justification that play a role. (In extreme cases, justification creates spaces between individual letters that may be bigger than the space between two words on the same page. We just illustrated as much!)

When the segmentation breaks up each image or zone of interest into small units for recognition, each character "cell" ideally corresponds to a single character. But a segment can just as well contain a group of connected characters... or only part of a single character. In that case, the segments have to be broken up further in several characters or various discrete segments have to recombined to recognize a letter!

Figure 9.1: spacing between the words

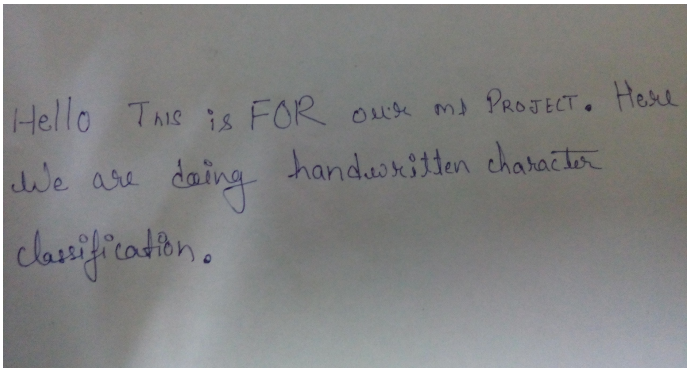

Figure 9.2: charecter segmented

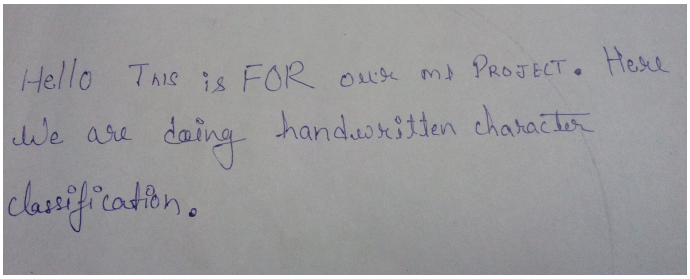

Figure 9.3: charecter segmented

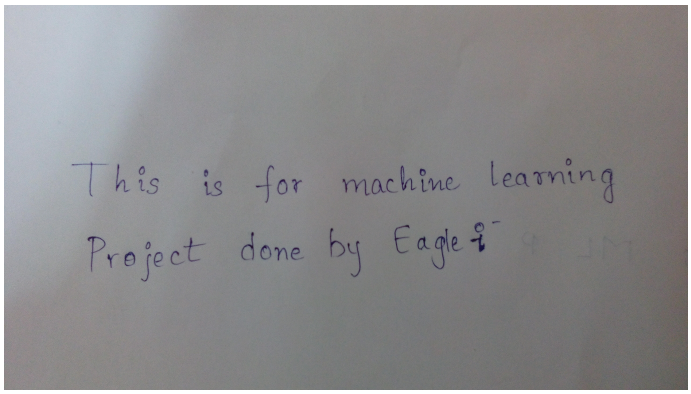

Figure 9.4: charecter segmented

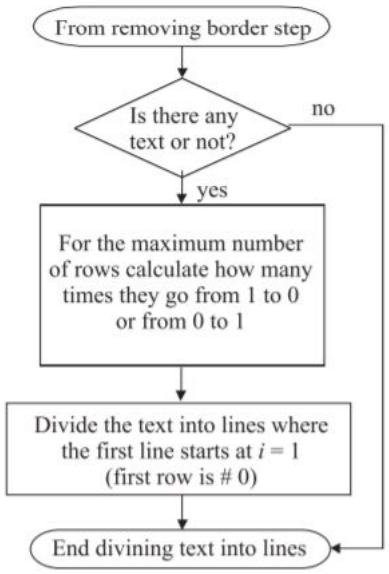

Fig. 8. Divided text into rows flowchart.

Figure 9.5: divided text to words flowchart

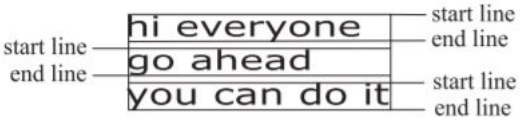

Figure 9.6: dividing text into words

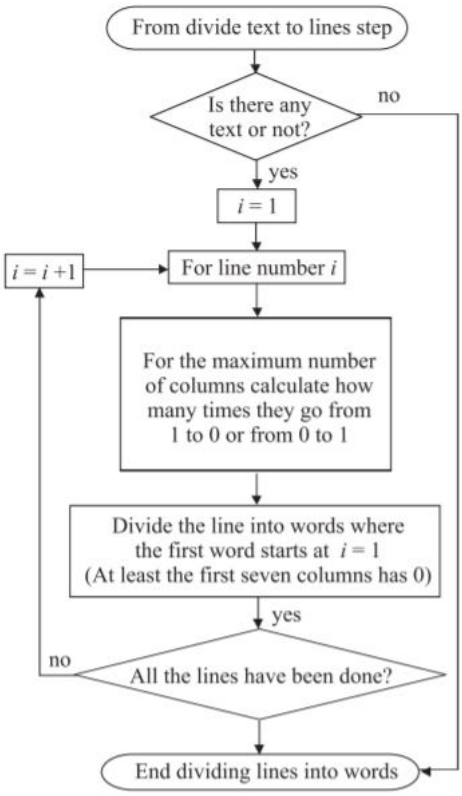

Figure 9.7: dividing rows into words

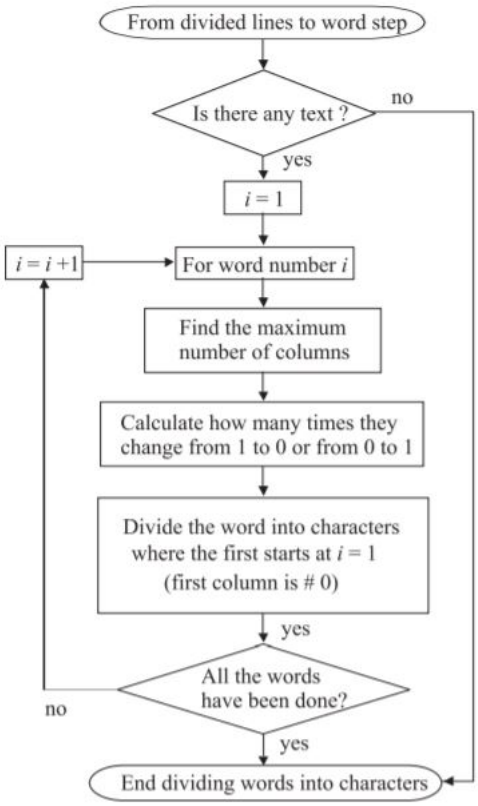

Figure 9.8: dividing words into charecters

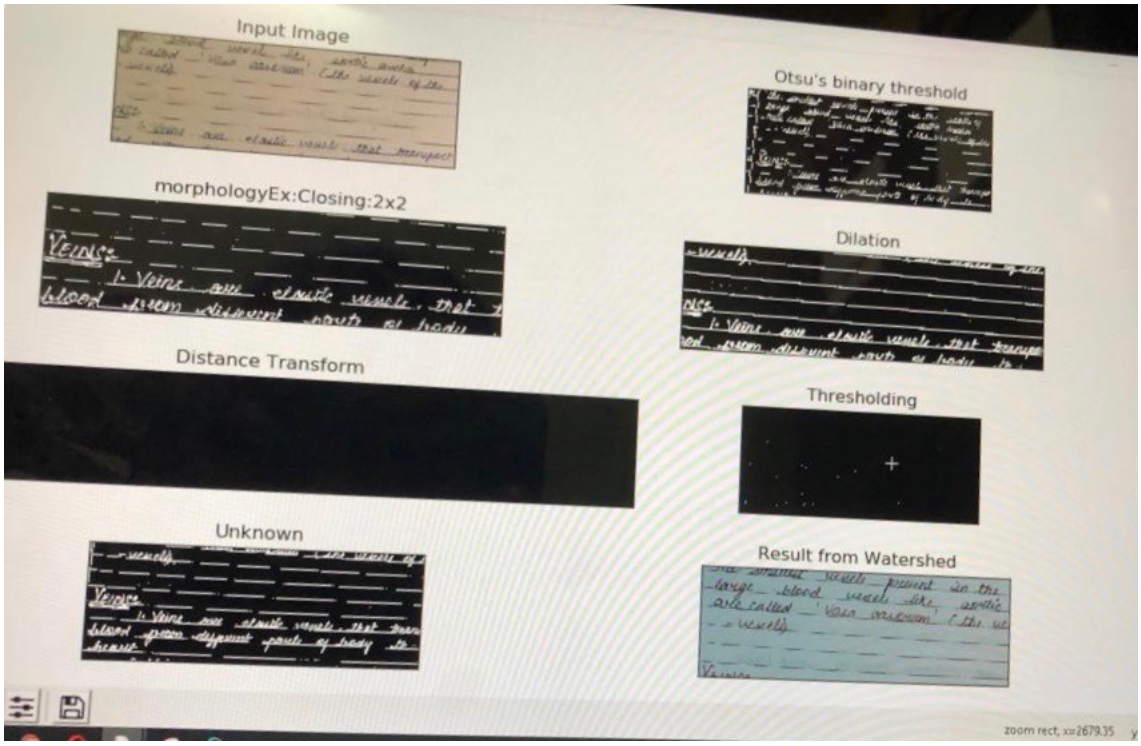

Figure 9.9: the output of all the steps in Conversion

Chapter 10

SOFTWARE REQUIREMENTS AND INSTALLATION

Installing OpenCV from prebuilt binaries
1. Below Python packages are to be downloaded and installed to their default locations.
1.1. Python-2.7.x./Python-3.5.x.
1.2. Numpy.
1.3. Matplotlib (Matplotlib is optional, but recommended since we use it a lot in our tutorials).
2. Install all packages into their default locations. Python will be installed to C:/Python27/.
3. After installation, open Python IDLE. Enter import numpy and make sure Numpy is working fine.
4. Download latest OpenCV release from sourceforge site and double-click to extract it.
7. Goto opencv/build/python/2.7 folder.
8. Copy cv2.pyd to C:/Python27/lib/site-packeges.
9. Open Python IDLE and type following codes in Python terminal.
Download and install necessary Python packages to their default locations 2.1. Python 2.7.x 2.2. Numpy 2.3. Matplotlib

output

¿¿¿ import cv2
¿¿¿ print cv2.$_{version}$ $_{If the results are printed out without any errors, congratulations!!! You have installed OpenCV-Python successfully.}$

Chapter 11

ADVANTAGES

Handwriting Recognition has many advantages that made it grow rapidly in the technology word now. There is much different kind of technologies that abide to enable others to take advantage of the handwriting recognition. The way this work was when people write letters a different way and they let the computer know what the intended letter was and change in to a text document. But the problem with this was the different way that the letters are written and this could make unnatural feel to the person who is writing it. Another way of appreciating this technology is that just write and the computer changes it to a text document but at the same time, the computer doesn't always get the right word and sometimes it insert the wrong letters.

Certain cell phones have the handwriting recognition system in it. The advantage of this is that it allows people to write on the cell phones using stylus and then the phone software translates the written words to the phone in text. But at the same time, the problem or called as disadvantage of this again is the same problem that the original software face, which is the unnatural letters strokes and sometimes the software predict the letters wrongly. This is one of the reasons that not all phones have this software in it.

Additionally, there are many more advantages of handwriting recognition system. Data storage, for an example, there are many files, contracts and some personal records that contains some handwritten information and notes. Certain of these documents contains original signature or notes that might not be electronically stored but this problems are overcome with the use of handwriting recognition system. Handwriting recognition software allows user to translate all those signature and notes into electronic words in a text document format. The advantage of this electronic storage is that this data only requires far less physical space than the storage of the physical copies. Another advantage of the electronic storage is also that it requires fewer employees to sort the documents through, organize and to keep the data storage warehouse.

Other than that, data retrieval is another advantage of handwriting recognition. Physical data retrieval always requires personnel to sort through physical copies of old information. The data must always have been stored and correctly organized and also it must have proper maintenance and upkeep on the physical copies. To retain this information or data, we perform electronic data retrieval by using a file search by using specific keywords, for example, like the names and the dates of the file or document. Handwriting recognition software allows the old files to be saved in a proper electronic format. This is how handwriting recognition software helps in saving the old files or important document. For an example, some clinics prefer keeping their patient's medical records and handwriting recognition helps in this situation by keeping these medical records safe in the computer. This data or files can be reviewed and updated time to time when needed without worrying if the date will be lost.

Moreover, another advantage of handwriting recognition is historical preservation. Historical papers mostly exists has physical format. Examples of historical papers are genealogical information,

written manuscripts, old family records, and some personal diaries and sometimes even shared old past stories. But still sometimes, these historical papers might be damaged or corrupted due to some accidents and there is when handwriting recognition software is really very helpful. Handwriting recognition helps to transform the writings in the papers to a text document format which can also be said as readable electronic format. By this way, historical facts can be stored, reviewed and shared easily too many people.

Lastly, the advantage is textual studies. A textual study is a category of literature studies. Literature studies involve reviewing the original manuscripts of literature in comparison with the printed version. This means that the story has been dug and changed at the hands of the editor's hand after coming from the hands of the author. This is why original manuscripts are cared very well but still this requires a complete review of the manuscript. Handwriting recognition helps to keep safe these original manuscripts in an electronic format and it can be reviewed without damaging the original copy of the manuscripts.

Chapter 12

DISADVANTAGES

The disadvantage of handwriting recognition technologies is that not everyone's handwriting is the same, everyone writes differently. This starts the problem in the handwriting recognition technology when it need to translate a person's handwriting into type and because of this problem many companies failed to perform well because many couldn't effectively use the program well enough.

In a nutshell, handwriting recognition is very useful software that really helps to safe and keeps data and documents well. But at time it also has its disadvantage such as that sometimes if fails to read certain people's handwriting and due to this many people do not prefer to use the handwriting recognition software that much. Even though handwriting recognition has its disadvantages but still it is growing rapidly in the technology world. Handwriting recognition is used when there are certain people who prefer writing on the screens rather than writing it on a paper. As far as seen, handwriting recognition will still grow fast in the technology world if it is upgraded more well.

Chapter 13

APPLICATIONS

The conversion of handwritten text to digital format is used mainly in
(a) National ID number recognition
(b)Postal office Automation

Application of handwritten Character recognition: (a) National ID number recognition system (b) Posta
with code number recognition on Envelope (c) Automatic license plate recognition and (d) Bank auton

(C) Automatic licence plate detection (d) Baank Automation

13.1 REFERENCES:

[1] Online and off-line handwriting recognition: a comprehensive survey - IEEE Journals Magazine

[2] Methods of combining multiple classifiers and their applications to handwriting recognition - IEEE Journals Magazine

[3] A handwritten character recognition system using directional element feature and asymmetric Mahalanobis distance - IEEE Journals Magazine

[4]An overview of character recognition focused on off-line handwriting - IEEE Journals Magazine

Mohanad Alata — Mohammad Al-Shabi, text detection and character recognition using fuzzy image processing, Journal of elcetrical engineering, VOL. 57, NO. 5, 2006, 258–267

[6]AMANDEEP KAUR, SEEMA BAGHLA, SUNIL KUMAR, STUDY OF VARIOUS CHARACTER SEGMENTATION TECHNIQUES FOR HANDWRITTEN OFF-LINE CURSIVE WORDS: A REVIEW, International Journal of Advances in Science Engineering and Technology, ISSN: 2321-9009

[7]Fajri Kurniawan, Mohd. Shafry Mohd. Rahim, Ni'matus Sholihah, Akmal

[8]Rakhmadi Dzulkifli Mohamad , Characters Segmentation of Cursive Handwritten Words based

on Contour Analysis and Neural Network Validation, ITB J. ICT, Vol. 5, No. 1, 2011, 1-16

Acknowledgement

I have a great pleasure to express my gratitude to all those who have contributed and motivated during my project work. Here you have a liberty to write anything and express your feeling to all those who have helped you.

...

Date:25th February 2019

Harshitha Aditham

Rajeshwari Nadar