Segmentation of English Sentences(non-cursive)

Gaganjeet Reen ICM2015003
Vinay Surya Prakash IHM2015004
Swapnesh Narayan ISM2015002
Leetesh Meena ISM2014008
Group - D

***Abstract*— With a faster-changing world, the need to create a flexible learning environment and eradicate the redundancies in the traditional course curriculum is very paramount. Therefore, we intend to make a digital tool which would segment the sentences accurately.**

I. INTRODUCTION

Recognizing handwritten text is one of the most challenging areas of pattern recognition. Segmentation of handwritten text has lots of applications in todays world like mailing postal code interpretation, bank cheque processing, etc.Our main intention is to segment the sentences accurately so that it can be used for identification of characters.

A) Binary Image
A binary image is a digital image that has only two possible values for each pixel. Typically, the two colors used for a binary image are black and white. The color used for the object(s) in the image is the foreground color while the rest of the image is the background color. In the document-scanning industry, this is often referred to as ”bi-tonal”.Binary images are also called bi-level or two-level. This means that each pixel is stored as a single biti.e., a 0 or 1

B) Gray Scale Image
a grayscale or greyscale image is one in which the value of each pixel is a single sample representing only an amount of light, that is, it carries only intensity information. Images of this sort, also known as black-and-white or gray monochrome, are composed exclusively of shades of gray. The contrast ranges from black at the weakest intensity to white at the strongest.Grayscale images are distinct from one-bit bi-tonal black-and-white images which, in the context of computer imaging, are images with only two colors: black and white (also called bilevel or binary images). Grayscale images have many shades of gray in between.

C) Morphology
Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, you can construct a morphological operation that is sensitive to specific shapes in the input image.

D) Dilation
Dilation (usually represented by Ł) is one of the basic operations in mathematical morphology. Originally developed for binary images, it has been expanded first to grayscale images, and then to complete lattices. The dilation operation usually uses a structuring element for probing and expanding the shapes contained in the input image.

E) Erosion
The basic idea in binary morphology is to probe an image with a simple, pre-defined shape, drawing conclusions on how this shape fits or misses the shapes in the image. This simple ”probe” is called structuring element, and is itself a binary image (i.e., a subset of the space or grid).It removes pixels on object boundaries.

F) Contours
Edges are computed as points that are extrema of the image gradient in the direction of the gradient. if it helps, you can think of them as the min and max points in a 1D function. The point is, edge pixels are a local notion: they just point out a significant difference between

neighbouring pixels.Contours are often obtained from edges, but they are aimed at being object contours. Thus, they need to be closed curves. You can think of them as boundaries (some Image Processing algorithms and libraries call them like that). When they are obtained from edges, you need to connect the edges in order to obtain a closed contour.

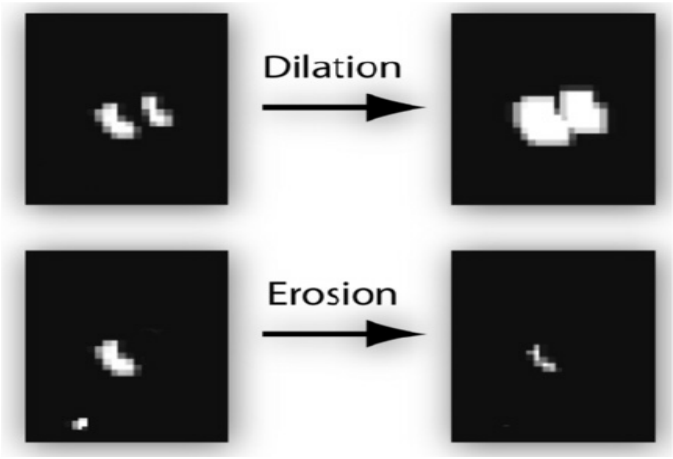

Fig. 1. Example of erosion and dilation

II. MOTIVATION

Segmentation of handwritten text is a very widely researched problem.It has a lot of applications in the real world and is the primary pre-processing step for character recognition.It is also closely related to the field of graph theory as all the characters in the english language can be interpreted as graphs and can be considered to be consisting of one or more connected components.Thus finding characters broadly consists of using depth first search algorithm and finding the corresponding connected components.

III. METHODS AND DESCRIPTION

The following methods have been used for pre-processing an image and then performing segmentation of characters.

A) **Grayscale and binarization of Image**
First phase of our project is to convert a given image into such a form so that it can be processed by CNN. To do this first step involves gray-scaling an image. During gray-scaling of image, pixels having dark colour will be coloured black and pixels having light colour will be coloured white. Once image is gray-scaled we have to binarize it. In binarization image is represented as 2-D array consisting of 0s and 1s. Pixels coloured black in gray-scaling are marked as 1 and white ones as 0.

B) **Dilation and Contour Separation**
The segmentation process in the tool is based on continuous contour detection. For the case of very thin text, if we analyze the image at pixel level, we can find out that the contour is discontinuous at several points and thus a contour which was expected to be considered as a single contour would be splitted into many contours which would eventually produce wrong outputs. Thus to overcome this problem the image is dilated to thicken the contours, which makes them continuous. After the dilation contours are separated horizontally to avoid overlapping of portions of different contours during segmentation. A variation of flood-fill algorithm is used to horizontally space the contours.

C) **Flood-Fill Algorithm**

Flood fill, also called seed fill, is an algorithm that determines the area connected to a given node in a multi-dimensional array. It is used in the ”bucket” fill tool of paint programs to fill connected, similarly-colored areas with a different color, and in games such as Go and Minesweeper for determining which pieces are cleared.The flood-fill algorithm takes three parameters: a start node, a target color, and a replacement color. The algorithm looks for all nodes in the array that are connected to the start node by a path of the target color and changes them to the replacement color. There are many ways in which the flood-fill algorithm can be structured, but they all make use of a queue or stack data structure, explicitly or implicitly.Depending on whether we consider nodes touching at the corners connected or not, we have two variations: eight-way and four-way respectively.

D) **Horizontal Segmentation**
Segmenting the image containing the text is the main part of the text recognition. Accuracy of the

project depends majorly on this part. We have carried out segmentation based on continuous contour approach.

IV. PROPOSED ALGORITHMS

Algorithm 1 Flood-Fill Algorithm (floodFillUtil)

```
if x or y is outside the screen then
    return
end if
if color of screen[x][y] is not same as prevC then
    return
end if
Recur for north, south, east , west, north-west, north-east, south-west, south-east.
floodFillUtil(screen, x+1, y, prevC, newC);
floodFillUtil(screen, x-1, y, prevC, newC);
floodFillUtil(screen, x, y + 1, prevC, newC);
floodFillUtil(screen, x, y - 1, prevC, newC);
floodFillUtil(screen, x+1, y + 1, prevC, newC);
floodFillUtil(screen, x-1, y - 1, prevC, newC);
floodFillUtil(screen, x+1, y - 1, prevC, newC);
floodFillUtil(screen, x-1, y + 1, prevC, newC);
```

V. IMPLEMENTATION

The code for image detection, erosion, dilation and segmentation has been written in python.The libraries that have been used are OpenCV and Numpy. We read the image as a grayscale image and the perform the corresponding operations on it pixel by pixel.

VI. RESULTS

Github Repository Link : `https://github.com/piano-man/graph_theory.git`

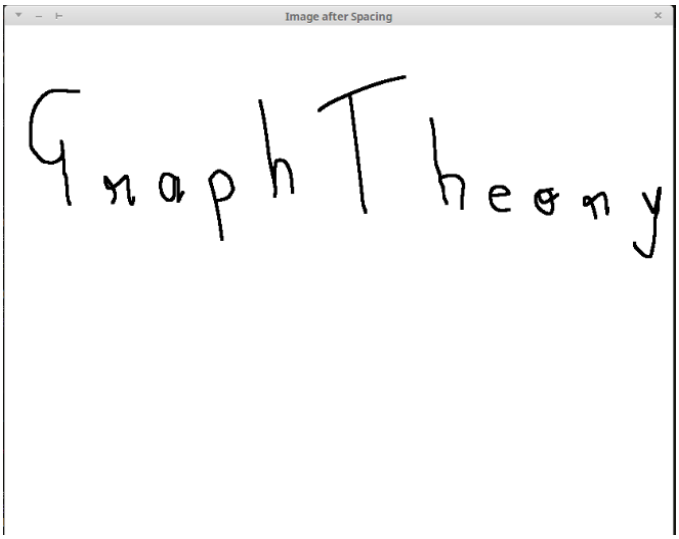

Fig. 2. English Sentence to be segmented

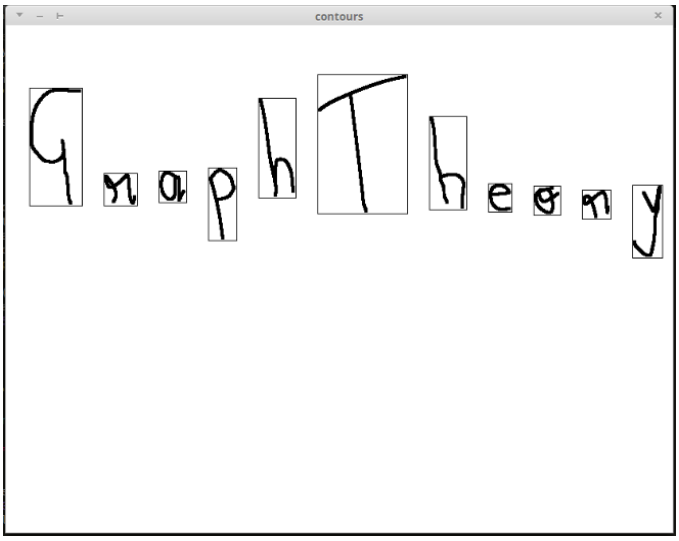

Fig. 3. Segmented English Sentence

Each individual rectangle in Fig. 3 represents a segmented character.

Analysis of Algorithms:-
The time complexity for the floodfill algorithm is $O(mn)$ where n and m stand for the number of rows and columns respectively.This is because the maximum number of pixels that can be covered by foodfill algorithm are m*n.The time complexity of final Segmentation is $O(Ct)$ where C is the number of

Contours detected in the image and t is the maximum number of characters in a particular contour in the image.

VII. APPLICATIONS

Character segmentation has widespread applications in the real world. It is the preliminary step for character recognition. It can be used as a preliminary step for the following :-

- Postal Code interpretation
- Number Plate recognition
- Manuscript Digitalization
- Expression Evaluation

VIII. DISCUSSION

In this paper we have proposed a method to segment non cursive english script.The method basically utilises the flood fill algorithm and the concept of contours to find connected characters.It also makes use of erosion and dilation as pre-processing steps for segmentation for characters.While the method works perfectly for english sentences, it cannot work for the letter 'i' as the 'i' consists of two disconnected components and the algorithm thus recognises it as two characters instead of one.

IX. CODE SNIPPETS

```
kernel = numpy.ones((3, 3),numpy.uint8)
binarized = cv2.erode(binarized, kernel, iterations = 1)
# binarized = cv2.dilate(binarized, kernel, iterations = 1)
binarized = space_contours.spacing(binarized)
cv2.imshow("Image after Spacing", binarized)
cv2.waitKey(0)
(cnts, heirarchy) = cv2.findContours(binarized.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

contours = []

for k in range(len(cnts)):
        if heirarchy[0][k][3] == 0:
                contours.append(cv2.boundingRect(cnts[k]))

contours.sort()


        #uncomment the below code to view segmented tokens
temp = binarized.copy()
for x, y, w, h in contours:
        cv2.rectangle(temp, (x, y), (x + w, y + h), 1)
cv2.imshow("contours", temp)
cv2.waitKey(0)
```

Fig. 4. Contour Detection

```
def flood_fill(img, mat, r, c, mat_c):
        h, w = img.shape
        sink = []
        sink.append((r, c))

        diff = mat_c - c
        maxi = c
        while len(sink):
                r, curr_c = sink[-1]
                sink.pop()
                if r < 0 or r >= h or curr_c < 0 or curr_c >= w:
                        continue

                if img[r, curr_c] != 0:
                        continue

                # if mat[r, curr_c + diff] == 150:
                img[r, curr_c] = 255
                mat[r, curr_c + diff] = 0
                maxi = max(maxi, curr_c)
                sink.append((r - 1, curr_c - 1))
                sink.append((r - 1, curr_c))
                sink.append((r - 1, curr_c + 1))
                sink.append((r, curr_c - 1))
                sink.append((r, curr_c + 1))
                sink.append((r + 1, curr_c - 1))
                sink.append((r + 1, curr_c))
                sink.append((r + 1, curr_c + 1))
        return maxi - c
```

Fig. 5. Flood Fill Algorithm

X. CONCLUSIONS

We can effectively segment english sentences using the methods suggested in the paper.

REFERENCES

[1] https://stackoverflow.com/questions/17103735/difference-between-edge-detection-and-image-contours
[2] https://en.wikipedia.org/wiki/Flood-fill
[3] https://www.mathworks.com/help/images/morphological-dilation-and-erosion.html
[4] Santosh, Dr. Jenila Livingston L.M. Text Detection from Documented Image Using Image Segmentation,2013
[5] Mathworks, Automatically Detect and Recognise Text in Natural Images
[6] Department of electrical and computer engineering, The university of Michigan-Dearborn MI 48128-1491 USA, Machine Printed Character Segmentation
[7] Amjad Rehman Khan and Dr. Zulkifli Mohammad A Simple Segmentation Approach for Unconstrained Cursive Handwritten Words in Conjunction with the Neural Network,2008
[8] M. Blumenstein and B. Verma, A New Segmentation Algorithm for Handwritten Word Recognition,1999
[9] HongLee and Brijesh Verma, Binary segmentation algorithm for English cursive handwriting recognition, 2011