<u>PROJECT REPORT</u>

TOPIC - OPTICAL CHARACTER RECOGNITION

GROUP - 11

19UCS185 - Harsh Kumar
19UCS214 - Subham Khetan
19UCS215 - Siddharth Akar
19UCS237 - Raghav Agarwal
19UCS245 - Abhinav Agarwal

Optical Character Recognition

***Optical Character Recognition* (OCR)** is a process of conversion of images of typed, handwritten, or printed text into machine-encoded text. It can be scanned documents or mechanically obtained documents.

OCR Technology deals with the problem of recognizing all kinds of different characters. Both handwritten and printed characters can be recognized and converted into a machine-readable, digital data format.

Think of any kind of serial number or code consisting of numbers and letters that you need digitized. By using OCR you can transform these codes into a digital output.

What OCR does not do is consider the actual nature of the object that you want to scan. It simply "takes a look" at the characters that you aim to transform into a digital format. For example, if you scan a word it will learn and recognize the letters, but not the meaning of the word.

Before OCR technology was available, the only option to digitize printed paper documents was by manually re-typing the text. Not only was this massively time-consuming, but it also came with inaccuracy and typing errors.

OCR is often used as a “hidden” technology, powering many well-known systems and services in our daily life. Less known, but as important, use cases for OCR technology include data entry automation, indexing documents for search engines, automatic number plate recognition, as well as assisting blind and visually impaired persons.

OCR technology has proven immensely useful in digitizing historic newspapers and texts that have now been converted into fully searchable formats and has made accessing those earlier texts easier and faster.

The use of OCR software in banks can also scan many customers’ important handwritten guarantee documents like their loan documents and more. Additionally, the incorporation of facial recognition software with OCR is also significantly remarkable because it provides two-layer security at ATMs.

AIM

Basically, The aim of our project is to design an OCR, ***Optical Character Recognition***, that converts an Image into text file. It not converts the image into a text file, but it also reduces the noise in the image, and returns the content of the picture, clear and readable.

The efficiency of the OCR can be varied with the use of various filters and processes that helps in reducing the noise. It fills the gap of scanning a physical document to convert it into an Image. It reduces the time in scanning documents and can be done in milliseconds by the use of OCR technology.

The OCR technology is an automatic process in which the image is converted into text in much ease. One simple way to store information to a computer system from these printed documents could be first to scan the documents and then store them as image files. But to re-utilize this information, it would be very difficult to read or query text or other information from these image files. Therefore a technique to automatically retrieve and store information, in particular text, from image files is needed.

The objective of our project is to achieve modification or conversion of any form of text or text-containing images such as printed or scanned text images, into an editable text format for deeper and further processing.

Opening

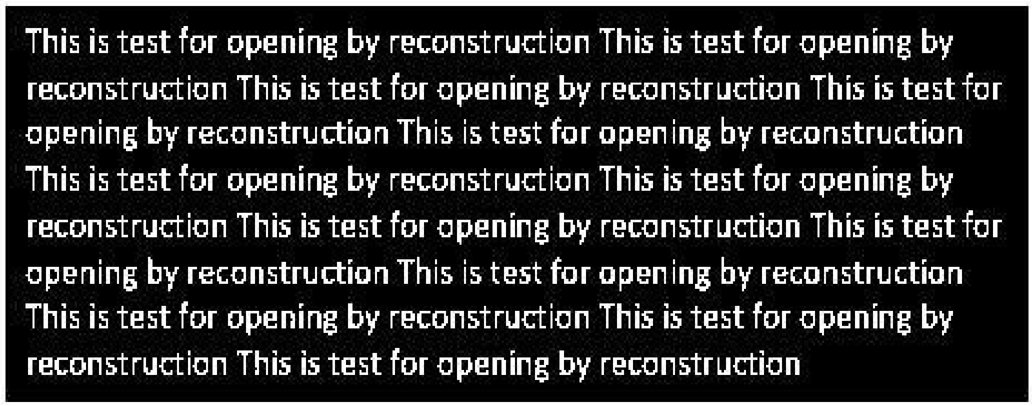
In mathematical morphology, opening is the dilation of the erosion of a set A by a structuring element B :

$A \circ B = (A \ominus B) \oplus B$, where $\ominus$ and $\oplus$ denote erosion and dilation, respectively.

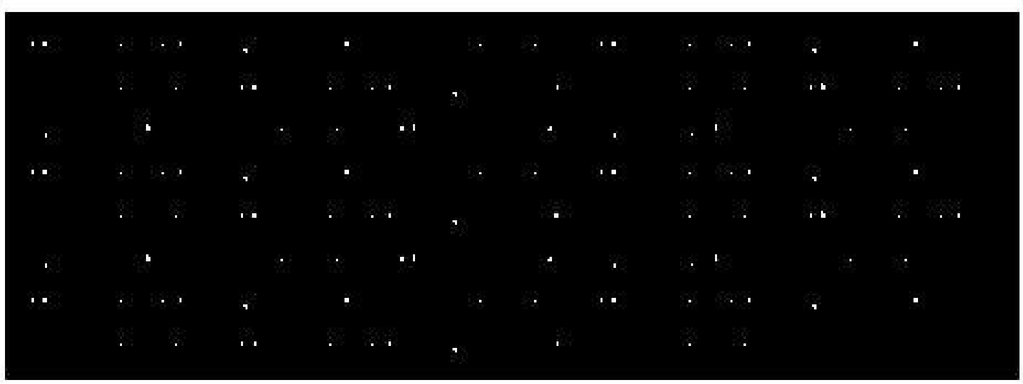
Together with closing, the opening serves in computer vision and image processing as a basic workhorse of morphological noise removal. Opening removes small objects from the foreground (usually taken as the bright pixels) of an image, placing them in the background, while closing removes small holes in the foreground, changing small islands of background into foreground. These techniques can also be used to find specific shapes in an image. Opening can be used to find things into which a specific structuring element can fit (edges, corners, ...).

One can think of B sweeping around the inside of the boundary of A, so that it does not extend beyond the boundary, and shaping the A boundary around the boundary of the element.

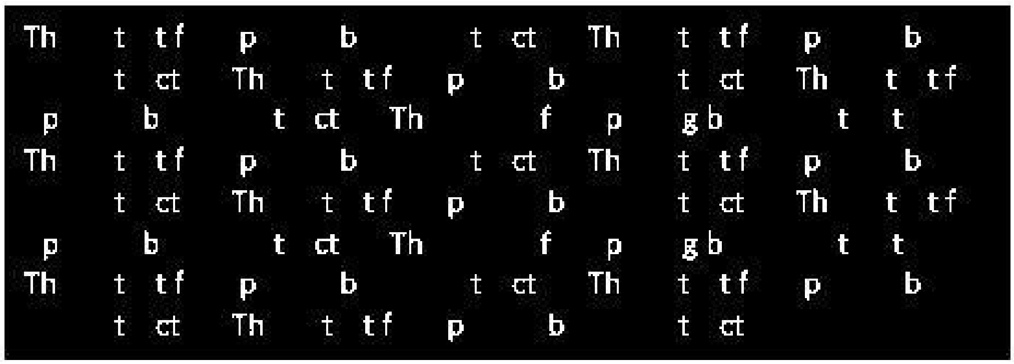
In morphological opening $(A \ominus B) \oplus B$, the erosion operation removes objects that are smaller than structuring element B and the dilation operation (approximately) restores the size and shape of the remaining objects. However, restoration accuracy in the dilation operation depends highly on the type of structuring element and the shape of the restoring objects. The opening by reconstruction method is able to restore the objects more completely after erosion has been applied.



-> Original image for opening by reconstruction

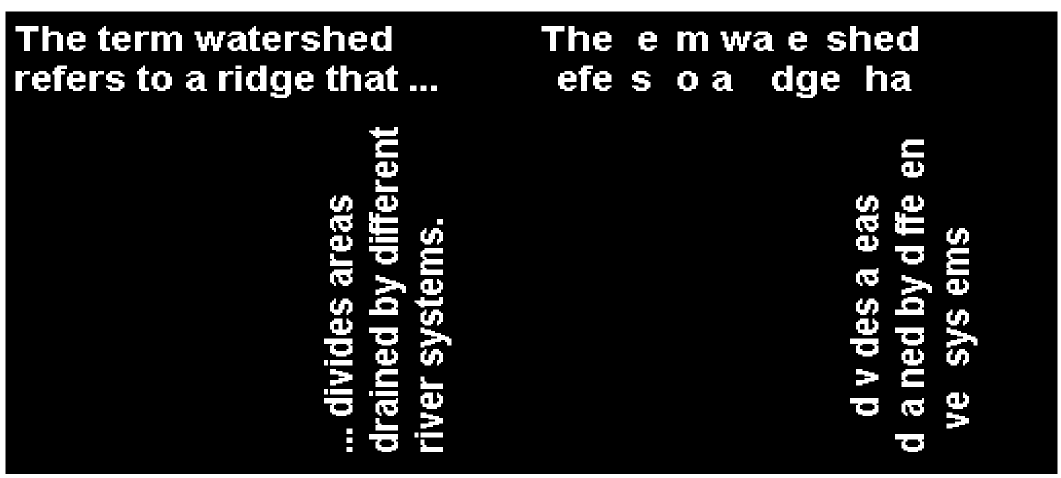


-> *Marker image*



-> *Result of opening by reconstruction*

BW2 = bwareaopen(BW,P) removes all connected components (objects) that have fewer than P pixels from the binary image BW, producing another binary image, BW2. This operation is known as an area opening.



Original Image *Processed Image*

Thresholding

Thresholding is a type of *image segmentation*, where we change the pixels of an image to make the image easier to analyze. In thresholding, we convert an image from color or grayscale into a *binary image*, i.e., one that is simply black and white.

Thresholding is a very important technique for segmenting images. It can be applied to intensity, color components, etc.It is useful in discriminating foreground from the background.

Thresholding is a function of

- Spatial Coordinates i.e (x,y)
- Grey level of the pixel i.e f(x,y)
- Some local properties of the image i.e A(x,y)

Therefore Thresholding operation can be expressed as

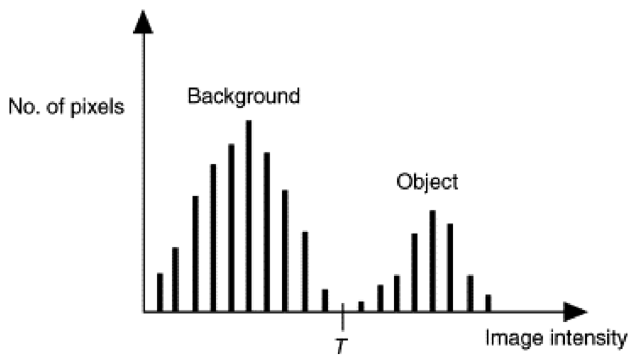
T=T[x,y,f(x,y),A(x,y)]

If Thresholding operation depends only on grey scale value it is known as **Global Thresholding.**in case if neighbourhood properties are also taken into consideration it is called **Local Thresholding.** If it depends on the pixel coordinates it is called **Dynamic/Adaptive Thresholding.**

The quality of threshold algorithm depends on the selection of a suitable threshold.tools that help to find the threshold is histogram. let F(x,y) be an image composed of light objects on a dark background.

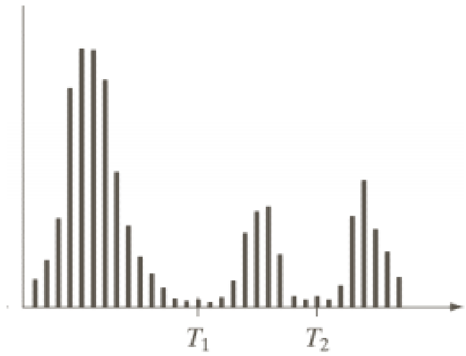
i.) Single Level Thresholding:

The objects can be extracted by comparing pixel values with a threshold T.
at any point(x,y)
 if F(x,y)>T =>object
 if F(x,y)<T => background.



ii.) Multilevel Thresholding:

It helps us to extract objects having specific intensity ranges.
at any point(x,y)
 if T1<F(x,y)<=T2 => object class 1
 if F(x,y)>T2 =>object class 2



thresholding is a critical step in Optical Character Recognition (OCR). It is also essential for other Character Image Extraction (CIE) applications, such as the processing of machine-printed or handwritten characters from carbon copy forms or bank checks, where smudges and scenic backgrounds, for example, may have to be suppressed.

GRAYSCALE IMAGE

In digital photography, computer-generated imagery, and colorimetry, a grayscale or 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. Grayscale images, a kind of 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.

Grayscale images can be the result of measuring the intensity of light at each pixel according to a particular weighted combination of frequencies (or wavelengths), and in such cases they are monochromatic proper when only a single frequency (in practice, a narrow band of frequencies) is captured. The frequencies can in principle be from anywhere in the electromagnetic spectrum (e.g. infrared, visible light, ultraviolet, etc.).

A colorimetric (or more specifically photometric) grayscale image is an image that has a defined grayscale colorspace, which maps the stored numeric sample values to the achromatic channel of a standard colorspace, which itself is based on measured properties of human vision.

If the original color image has no defined colorspace, or if the grayscale image is not intended to have the same human-perceived achromatic intensity as the color image, then there is no unique mapping from such a color image to a grayscale image.\

CORRELATION

The correlation of a filter $w(x, y)$ of size $m \times n$ with an image $f(x, y)$, denoted as $w(x, y) \star f(x, y)$

$$w(x, y) \star f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

a = (m - 1)/2, b = (n - 1)/2,

Snippets of our code

Ocr.m -

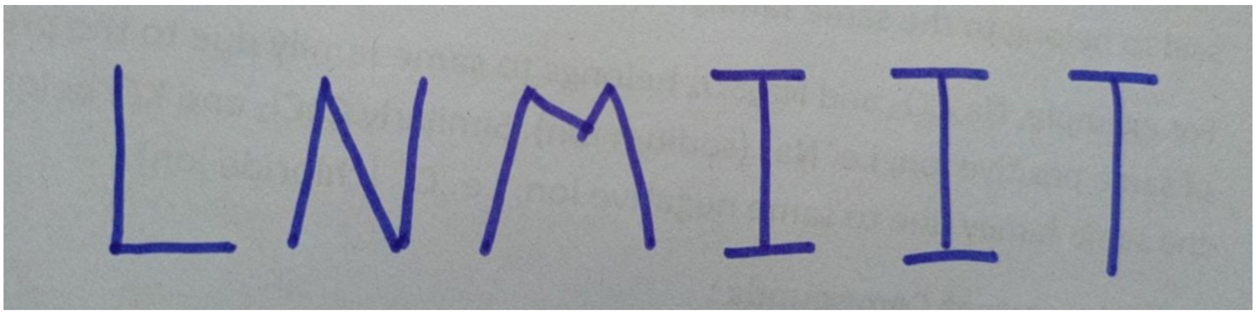
```
% PRINCIPAL PROGRAM
warning off %#ok<WNOFF>
% Clear all
clc, close all, clear all
% Read image
imagen=imread('TEST_1.JPG');
% Show image
imshow(imagen);
title('INPUT IMAGE WITH NOISE')
% Convert to gray scale
if size(imagen,3)==3 %RGB image
    imagen=rgb2gray(imagen);
end
% using median filter
imagen = medfilt2(imagen,[15 15])
% Convert to BW
threshold = graythresh(imagen);
imagen =~im2bw(imagen,threshold);
% Remove all object containing fewer than 30 pixels
imagen = bwareaopen(imagen,30);
%Storage matrix word from image
word=[ ];
re=imagen;
%Opens text.txt as file for write
fid = fopen('text.txt', 'wt');
```

```
% Load templates
load templates
global templates
% Compute the number of letters in template file
num_letras=size(templates,2);
while 1
    %Fcn 'lines' separate lines in text
    [fl re]=lines(re);
    imgn=fl;
    %Uncomment line below to see lines one by one
    %imshow(fl);pause(0.5)
    %-------------------------------------------------------------
    % Label and count connected components
    [L Ne] = bwlabel(imgn);
    for n=1:Ne
        [r,c] = find(L==n);
        % Extract letter
        n1=imgn(min(r):max(r),min(c):max(c));
        % Resize letter (same size of template)
        img_r=imresize(n1,[42 24]);
        %Uncomment line below to see letters one by one
         %imshow(img_r);pause(0.5)
        %---------------------------------------------------------
        % Call fcn to convert image to text
        letter=read_letter(img_r,num_letras);
```

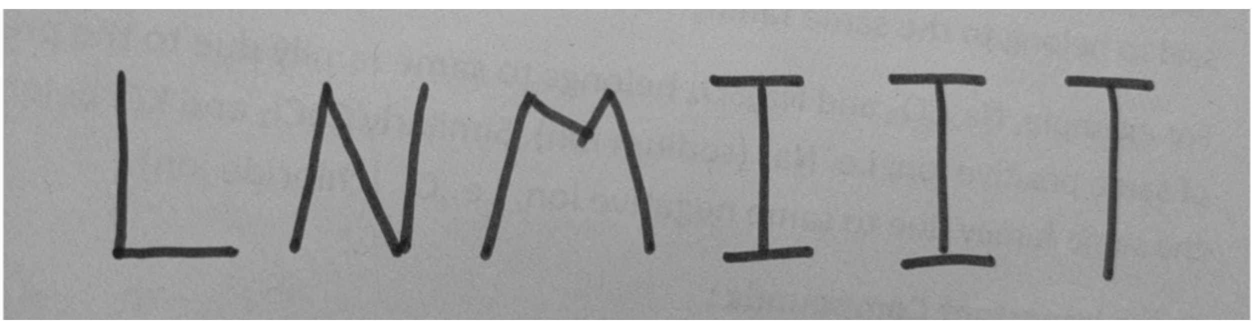
```
        % Letter concatenation
        word=[word letter];
    end
    %fprintf(fid,'%s\n',lower(word));%Write 'word' in text file (lower)
    fprintf(fid,'%s\n',word);%Write 'word' in text file (upper)
    % Clear 'word' variable
    word=[ ];
    %*When the sentences finish, breaks the loop
    if isempty(re)  %See variable 're' in Fcn 'lines'
        break
    end
end
fclose(fid);
%Open 'text.txt' file
winopen('text.txt')

clear all
```

SAMPLE INPUT AND OUTPUT

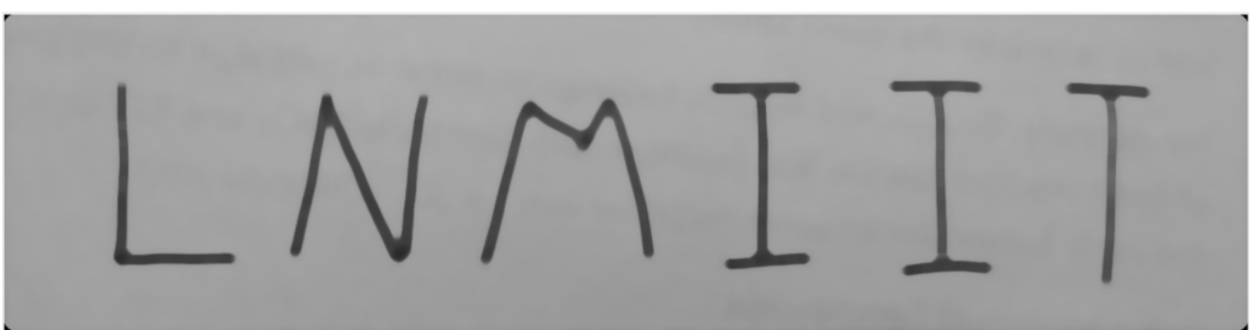
1. Input Image



2. Image after applying gray scaling to it



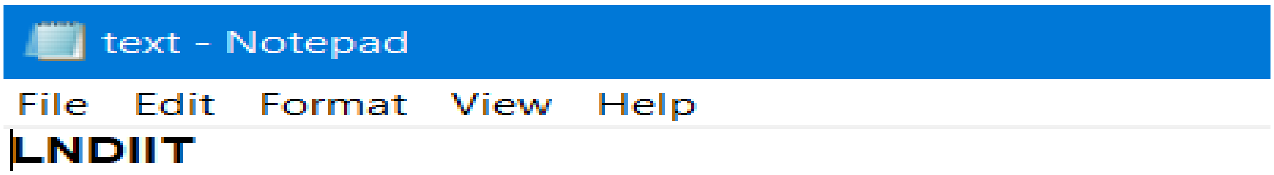
3. Now as we can see that there is a lot of salt and pepper noise so we will apply median filter to reduce the noise



4. Now we will binarize this image so that our letters can be more clearly visible and can be matched with sample letters which are also of similar form

5. Now to increase the width of the letters and make them more recognizable we will apply opening which is a morphological process

6. Now we will try to recognize each letter and output them into a text file, our model work with 80% accuracy on hand written text



CONCLUSION

We would like to conclude with the fact that there were instances where our OCR design has failed in producing satisfactory results, whereas in many cases, it has produced results that are very much expected.

Our design has produced expected results for the cases where the input was an electronically typed image, it was able to read the input image carefully and had produced satisfactory results. In cases where some of the inputs were handwritten, the output had some words that were incorrect with respect to the given input. Overall the reliability of the project stands on some expected level and can be further improved with use of median filters.

Again, it was a great learning opportunity. We are thankful to ***Dr. Ram Prakash Sharma***, for giving us this learning opportunity. With this project, we had a great insight into the field of Image Processing and can be used to further explore fields like Computer Vision. This has also helped us in improving our knowledge as well as helped us in learning from our mistakes.