OCR USING ANN

Optical Character Recognition using Artificial Neural Network

APRIL 17, 2019

Submitted by:
BHOR Tejas
CHAUDHARY sudhanshu
SINGH Arjun

Contents

ABSTRACT

This paper describes a geometry based technique for feature extraction applicable to segmentation-based word recognition systems. The proposed system extracts the geometric features of the character contour. This features are based on the basic line types that forms the character skeleton. The system gives a feature vector as its output. The feature vectors so generated from a training set, were then used to train a pattern recognition engine based on Neural Networks so that the system can be benchmarked.
Keywords—Geometry, Character skeleton, Zoning, Universe of Discourse, Line type, Segment , Direction feature.

In the running world, there is growing demand for the software systems to recognize characters in computer system when information is scanned through paper documents as we know that we have number of newspapers and books which are in printed format related to different subjects. These days there is a huge demand in "storing the information available in these paper documents in to a computer storage disk and then later reusing this information by searching process". One simple way to store information in these paper documents in to computer system is to first scan the documents and then store them as IMAGES. But to reuse this information it is very difficult to read the individual contents and searching the contents form these documents line-by-line and word-by-word. The reason for this difficulty is the font characteristics of the characters in paper documents are different to font of the characters in computer system. As a result, computer is unable to recognize the characters while reading them. This concept of storing the contents of paper documents in computer storage place and then reading and searching the content is called DOCUMENT PROCESSING.

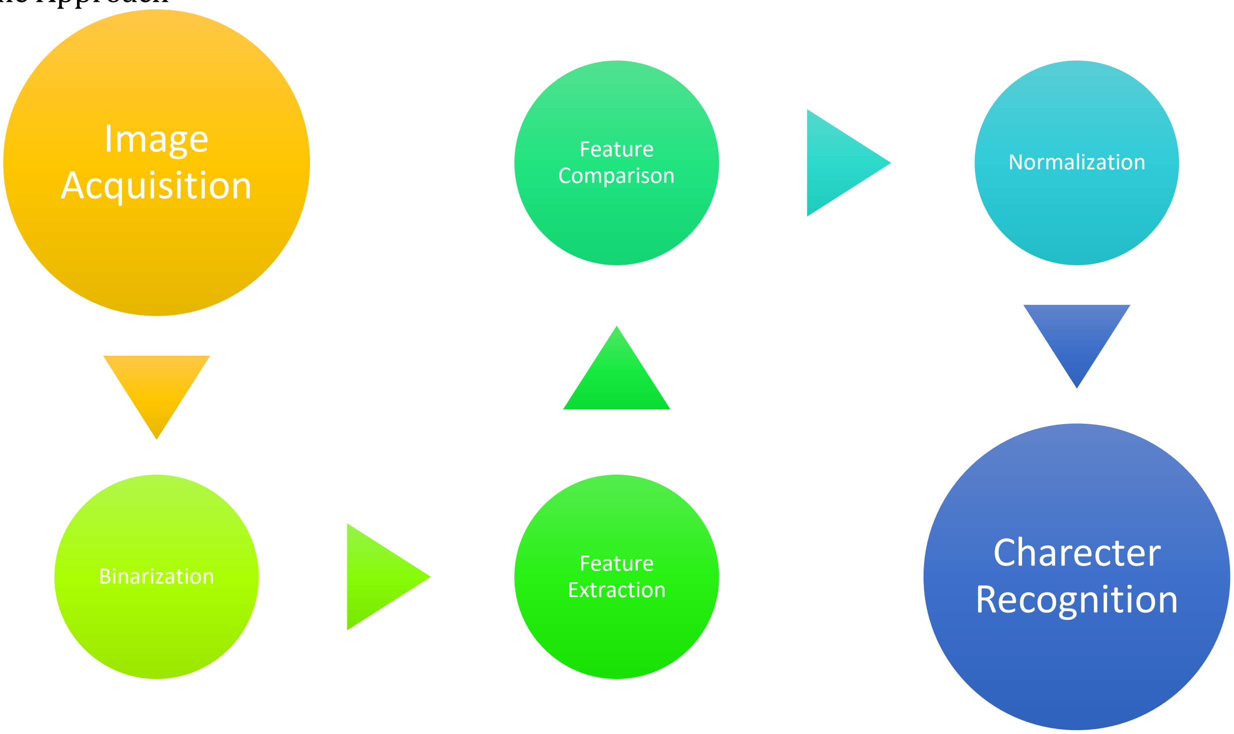
Sometimes in this document processing we need to process the information that is related to languages other than the English in the world. For this document processing we need a software system called CHARACTER RECOGNITION SYSTEM. This process is also called DOCUMENT IMAGE ANALYSIS (DIA). Thus our need is to develop character recognition software system to perform Document Image Analysis which transforms documents in paper format to electronic format. For this process there are various techniques in the world. Among all those techniques we have 2 chosen Optical Character Recognition as main fundamental technique to recognize characters. The conversion of paper documents in to electronic format is an on-going task in many of the organizations particularly in Research and Development (R&D) area, in large business enterprises, in government institutions, so on.

From our problem statement we can introduce the necessity of Optical Character Recognition in mobile electronic devices such as cell phones, digital cameras to acquire images and recognize them as a part of face recognition and validation. To effectively use Optical Character Recognition for character recognition in-order to perform Document Image Analysis (DIA), we are using the information in Grid format. . This system is thus effective and useful in Virtual Digital Library's design and construction.

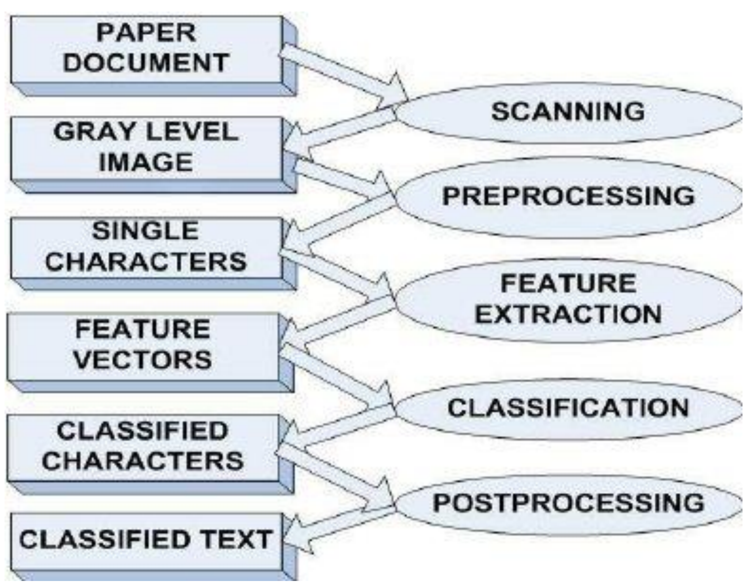
OVERVIEW

The project covers the methods in Image preprocessing but the objective is to extract features and study them. What makes them unique and how to use them to differentiate charecters.

The Approach



Fundamental of OCR



An image can be defined as a function f(x,y). Herex and y are spatial (plane) coordinates. The amplitude of function f at any point (x,y) is called intensity or grey level of an image at that point. Now an image is said to be a digital image when x,y and amplitude values of function f(x,y) are finite and discrete in nature. A digital image is composed of a finite number of elements (known as pixels), each of which has

- a particular value
- a particular location

A pixel is the smallest element in an image. In general, there are three levels of processing or three types of processes in digital image processing namely: low, mid and high-level processes.

- Low-level processing involves primitive operation such as image preprocessing to reduce noise, contrast enhancement, image sharpening, etc. In the low-level process, both input and output are images.
- Mid-level processing involves tasks such as image segmentation, description of images, object recognition, etc. In the mid-level process, inputs are generally images but its outputs are generally image attributes.
- High-level processing involves “making sense” from a group of recognized objects. This process is normally associated with computer vision.

SCANNING

To extract characters from scanned images it is necessary to convert the image into proper digital image. This process is called text digitization. The process of text digitization can be performed either by a Flat bed scanner or a hand held scanner. Hand held scanner typically has a low resolution range. Appropriate resolution level typically 300 - 1000 dots per inch for
better accuracy of text extraction

PREPROCESSING

Preprocessing consists of number of preliminary processing steps to make the raw data usable for the recognizer. The typical preprocessing steps included the following process:

- Binarization
- Noise Detection & Reduction
- Skew detection & correction
- Page layout analysis
- Segmentation

BINARIZATION METHODS

Binarization is a technique by which the gray scale images are converted to binary images. Some binarization methods are given below:

1. Global Fixed Threshold:

The algorithm chooses a fixed intensity threshold value I. If the intensity value of any pixel of an input is more than I, the pixel is set to white otherwise it is black. If the source is a color image, it first has to be converted to grey level using the standard conversion

2. Otsu Global Algorithm:

This method is both simple and effective. The algorithm assumes that the image to be threshold contains two classes of pixels and calculates the optimum threshold separating those two classes so that their combined spread (intra-class variation) is minimal

3. Niblack's Algorithm:

Niblack‟s algorithm calculates a pixel wise threshold by sliding a rectangular window over the grey level image. The threshold is computed by using the mean and standard deviation, of all the pixels in the window

4. Adaptive Niblack's Algorithm:

In archive document processing, it is difficult to identify suitable sliding window size SW and constant k values for all images, as the character size of both frame and stroke may vary image by image. Improper choice of SW and k values results in poor binarization. Modified Niblack‟s algorithm allows automatically chosen values for k and SW, which is called adaptive Niblack‟s algorithm

5. Sauvola's Algorithm:

Sauvola‟s algorithm is a modification of Niblack‟s which is claimed to give improved performance on documents in which the background contains light texture, big variations and uneven illumination. In this algorithm, a threshold is computed with the dynamic range of the standard deviation

NOISE DETECTION & CORRECTION METHODS

Noise can be produced during the scanning section automatically. Two types of noises are common. They are background noise and salt & paper noise. We cannot eliminate wide pixels from upper or lower portion of a character because it may not only eliminate noise but also the difference between two characters like e and I and for some other characters. Some noise detection methods are given below:

The commonly used approach is, to low pass filter the image and to use it for later processing. The objective in the design of a filter to reduce noise is that it should remove as much of the noise as possible while retaining the entire signal

Mixed noise of Gaussian and impulse cannot be removed by conventional method at the same time. We can use enhanced TV (Total Variation) filter which can remove these two types of noise based on PSNR (Pick Signal to Noise Ratio) and subjective image quality. The main advantage of this method is it can eliminate mixed noise efficiently & quickly. Comparing with the conventional method such as averaging method and the median method, the proposed method in gets higher quality. This method is based on kalman filter.

Dots existing in a character like I may be treated as noise. In this method, first estimate the size of the dots in each region of the text. Then the minimum size of dots in each region is estimated based on the estimated size of dot in that region. The use of connected component information will eliminate the noise using statistical analysis for background noise removal. For other type of noise removal and smoothing we can use wiener and median filters. Connected component information is found using boundary finding method (such as edge detection technique). Pixels are sampled only where the boundary probability is high. This method requires elaboration in the case where the characteristics change along the boundary. In, noise is removed from character images. Noise removal includes removal of single pixel component and removal of stair case effect after scaling. However, this does not consider background noise and salt and paper noise.

FEATURE EXTRACTION

Feature extraction plays a major role in the performance of the OCR. The characteristics of the feature extraction techniques have to be independent of the scalable font characteristics such as type, size, style, tilt, rotation and should be able to describe the complex, distorted, broken characters effectively. A feature vector should be simple, reliable, complete, and compact to recognize any input character with high accuracy similar to human perception. A lot of research has been done on feature extraction techniques for OCR for the past few decades. The existing OCR methods will work successfully for one or two fonts and they have used the combination of existing features to improve the accuracy. Therefore, we feel there is still scope to work on feature extraction techniques for the recognition of multilingual characters and different font family.

We need Features to identify

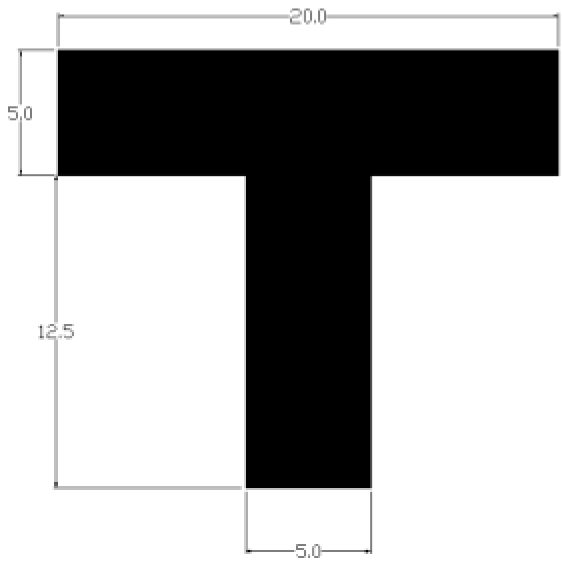
1. ‘Character properties’.
2. Features are extracted from the image by means of spatial pixel-based calculation.
3. A collection of such features help in defining a character uniquely.

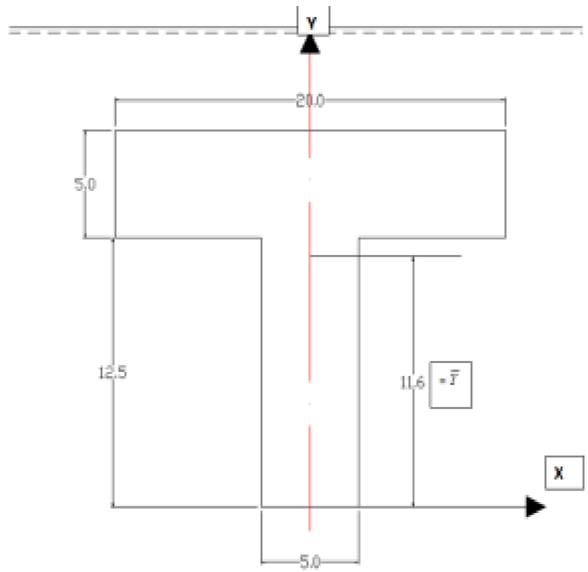
List of features

- Feature 1- Centroid
- Feature 2 - Area
- Feature 3 - Major Axis Length & Minor Axis Length
- Feature 4 - Filled Area of all value characters
- Feature 5 - Perimeter
- Feature 6 - Walsh Hadamard transformation
- Feature 7 - Fourier transform

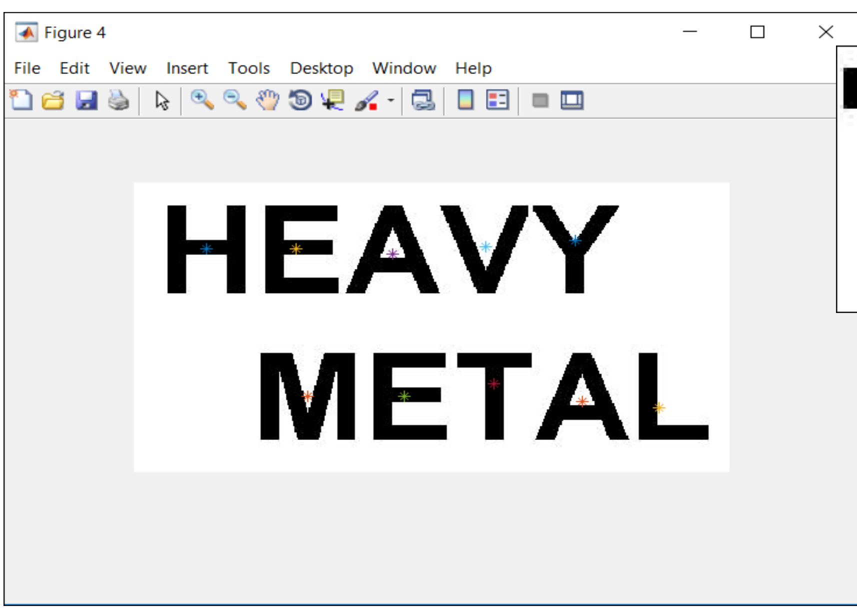
Feature 1- Centroid

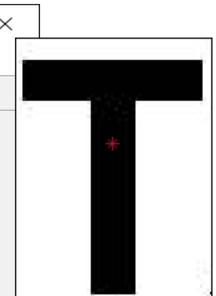
Centroid of T-section can be found easily with respect to reference x-axis. This is so because T section is symmetrical about Y-axis. And it becomes a unique distinguishing parameter for every character

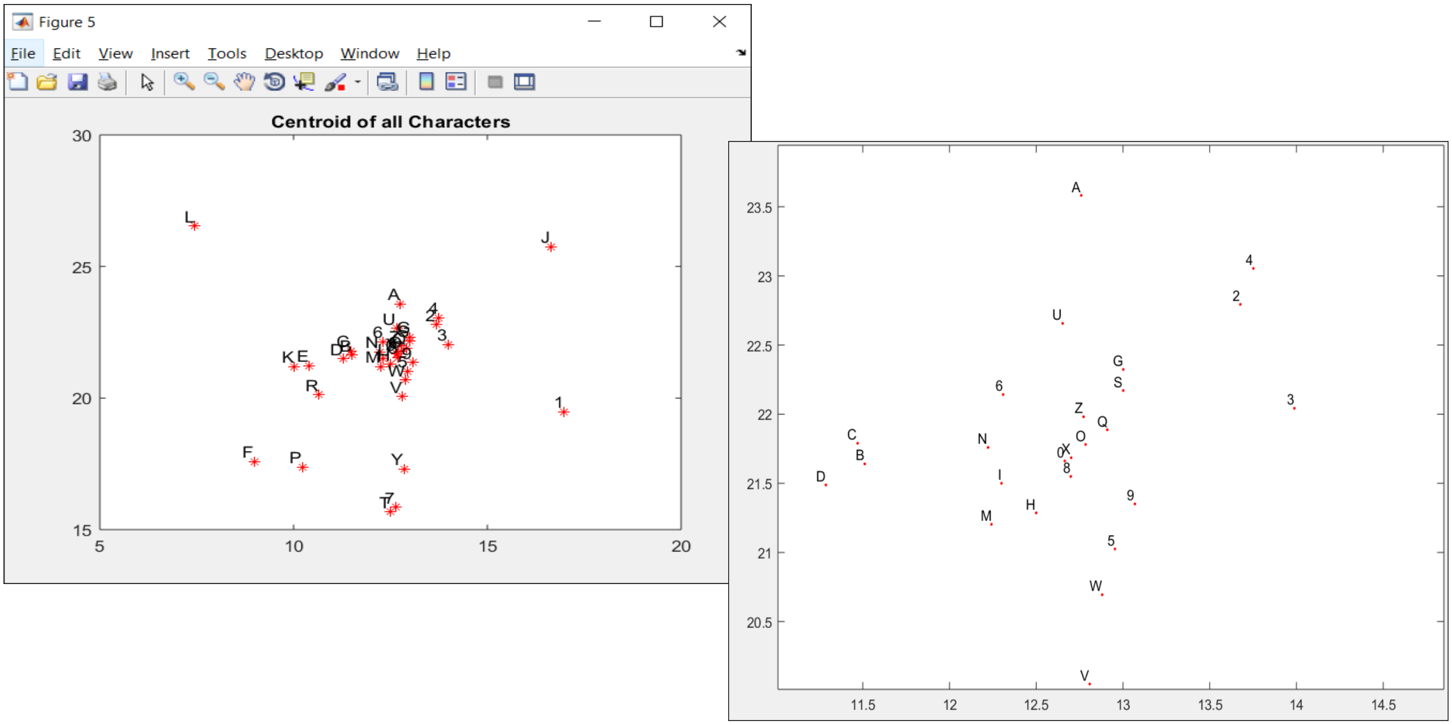




T-section is comprised of two rectangles.
Area of Large Rectangle = 20×5 = 100 cm.
Area of smaller Rectangle = 12.5×5 = 62.5 cm.
Sum of Area = 100 + 62.5 = 162.5 cm.
Centroid of Large Rectangle with respect to reference x-axis = Y = 12.5/2 = 6.25 cm.
Centroid of small rectangle with respect to reference x-axis = Y = 5/2 + 12.5 = 15 cm.
Moment of Area of Large rectangle = M_1 = 62.5×6.25 = 390.25 cm.
Moment of Area of small rectangle = M_2 = 100×15 = 1500 cm.
Sum of Moments = 390.25 + 1500 = 1890.625 cm.
$\bar{Y}$=Sum of of Moments Sum of Area
$\bar{Y}$=1890.625162.5=11.63cm

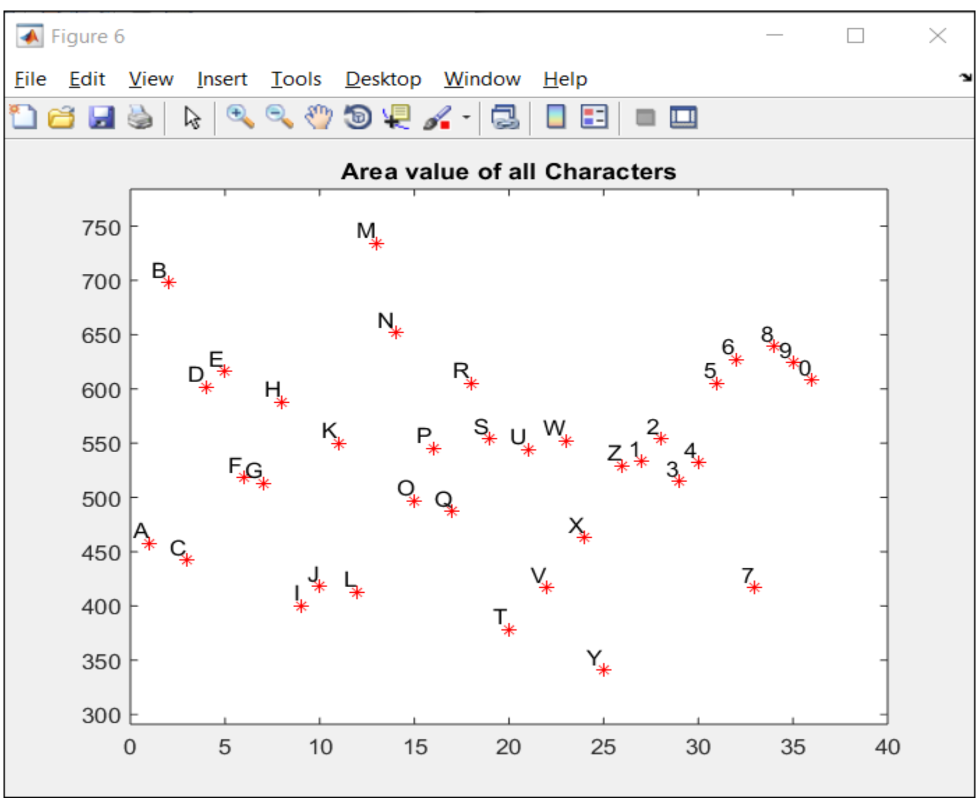






Feature 2 - Area

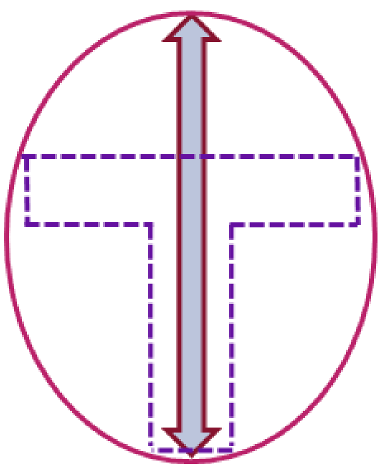
Area under every alphabet, it varies for every alphabet since the size is different even for the alphabet like M & W and B & 8 the difference is visible. It is the Actual number of pixels in the region, returned as a scalar.

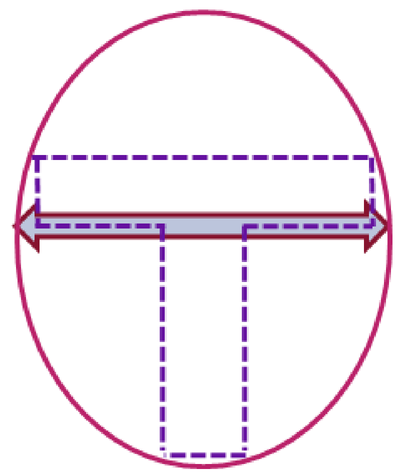


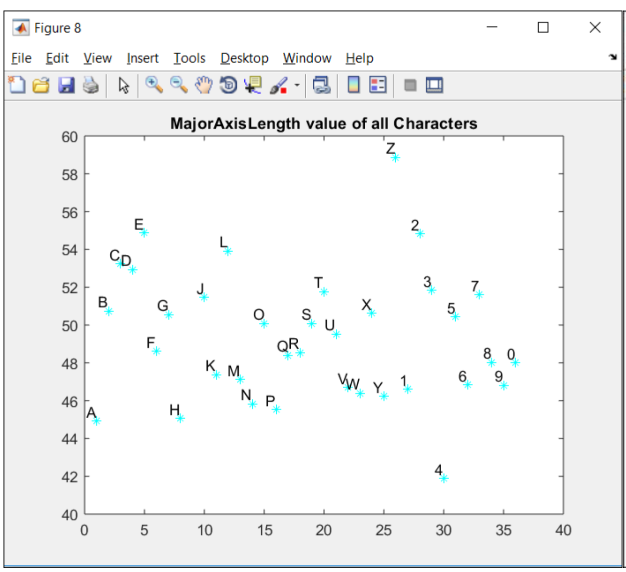
Feature 3 - Major Axis Length & Minor Axis Length

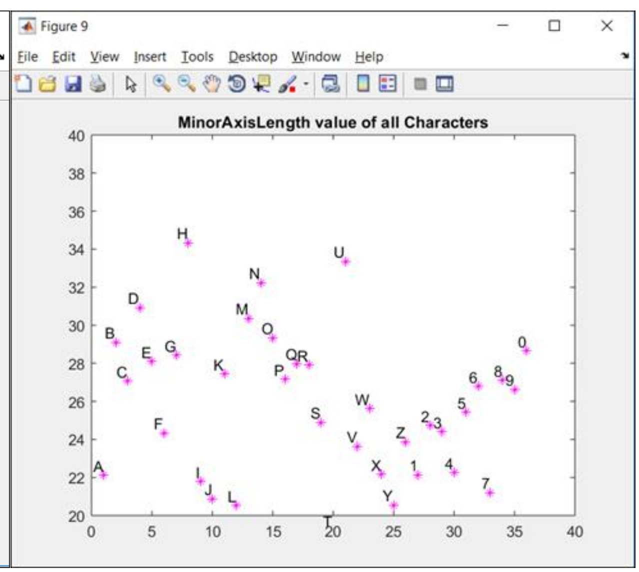
Major Axis Length: Length (in pixels) of the major axis of the ellipse that has the same normalized second central moments as the region, returned as a scalar.

Minor Axis Length: Length (in pixels) of the minor axis of the ellipse that has the same normalized second central moments as the region, returned as a scalar.



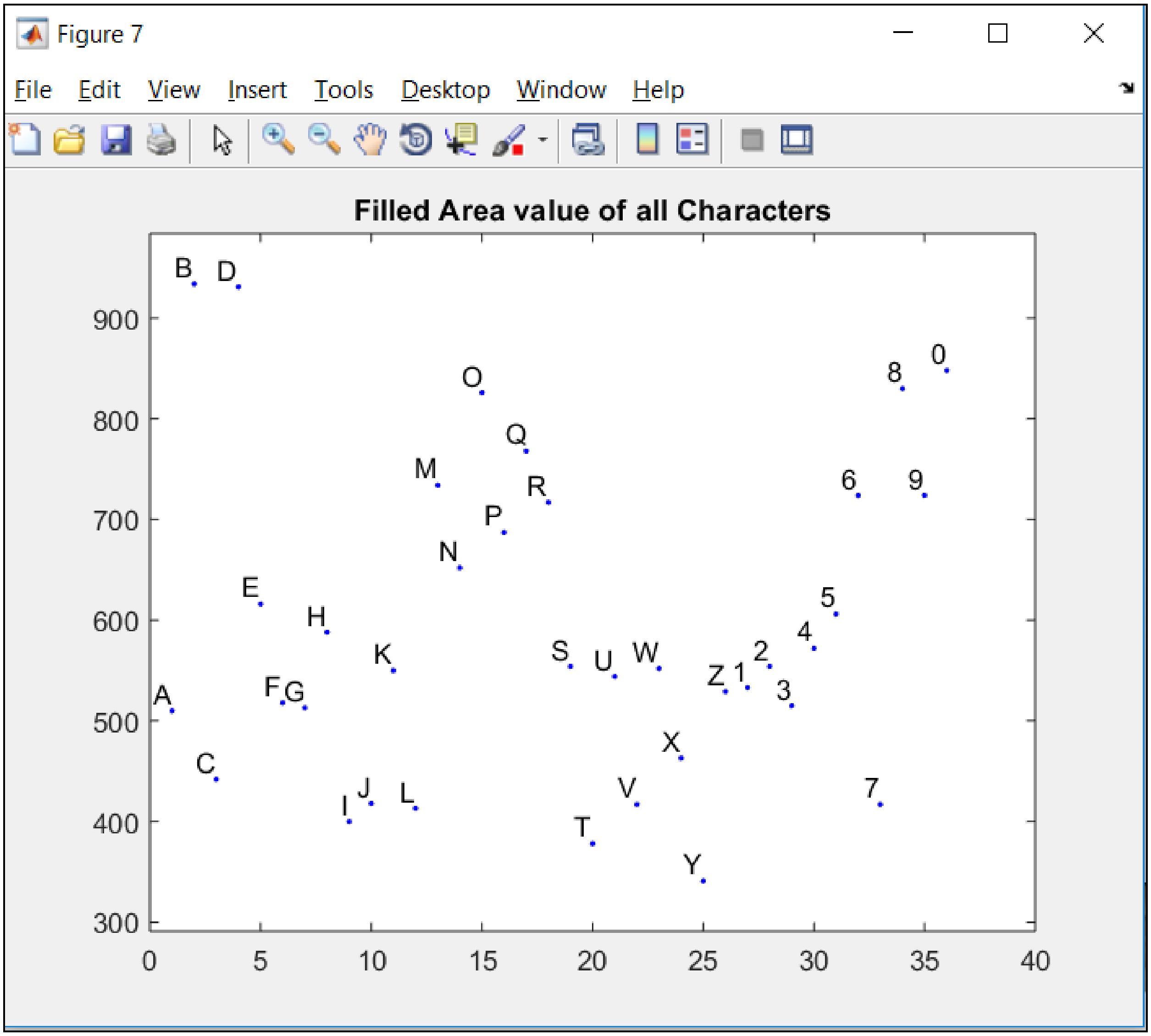






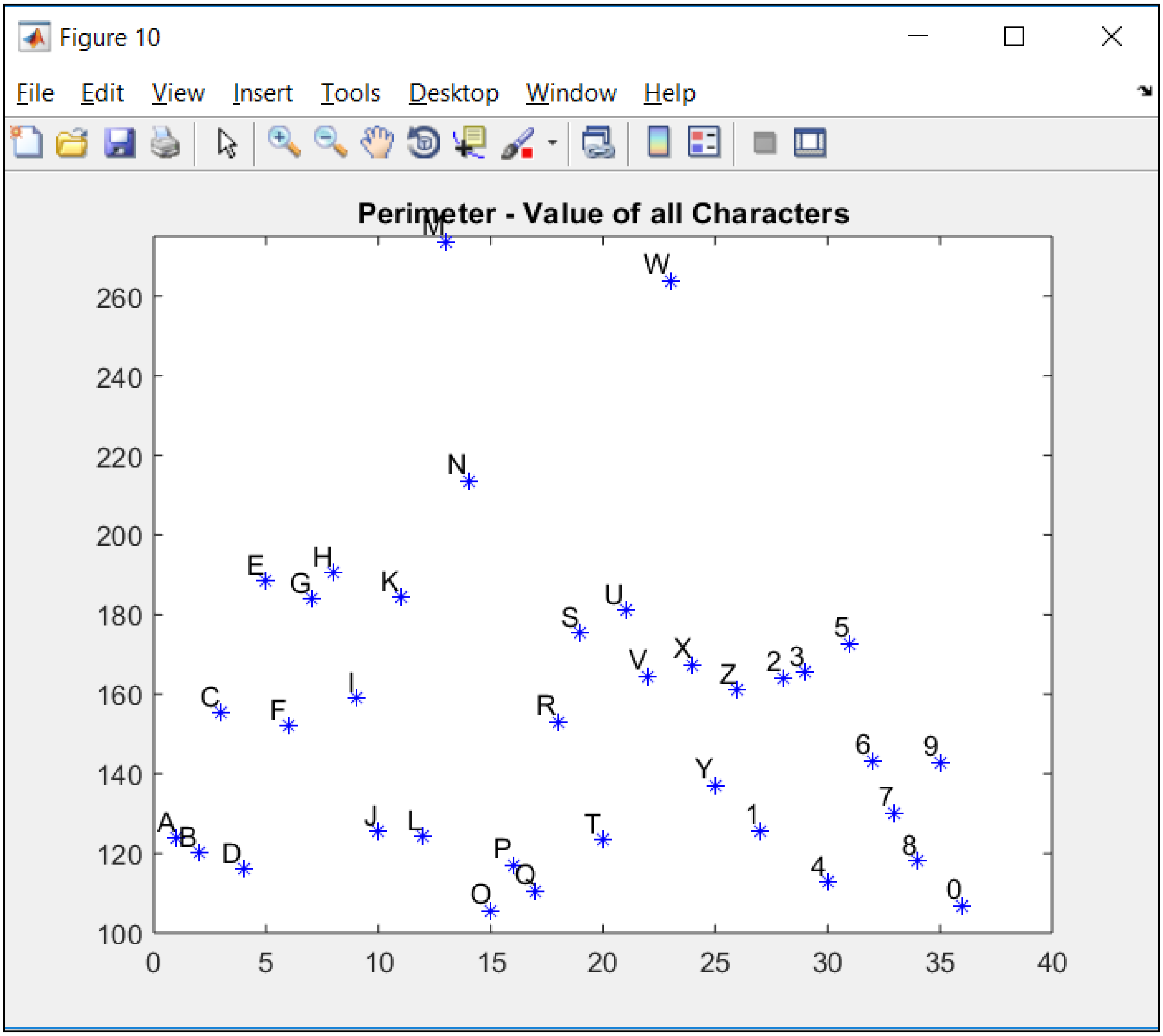
Feature 4 - Filled Area of all value characters

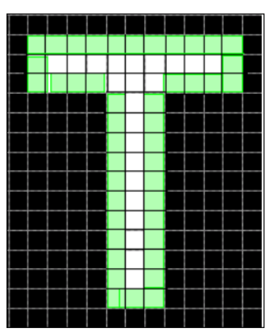
Number of on pixels in Filled Image, returned as a scalar.



Feature 5 - Perimeter

Distance around the boundary of the region returned as a scalar. It is Computed by calculating the distance between each adjoining pair of pixels around the border of the region.



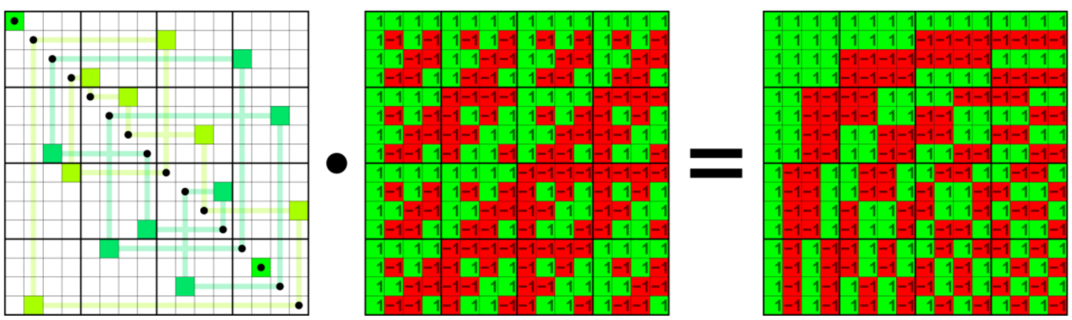


Feature 6 - Walsh Hadamard transformation

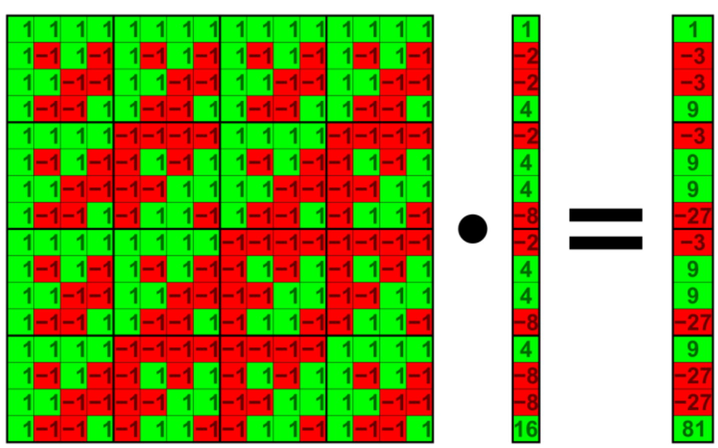
A **Walsh matrix** is a specific square matrix of dimensions 2^n, where *n* are some particular natural number. The entries of the matrix are either +1 or −1 and its rows as well as columns are orthogonal, i.e. dot product is zero. The Walsh matrix was proposed by Joseph L. Walsh in 1923.Each row of a Walsh matrix corresponds to a Walsh function.
The *naturally ordered* Hadamard matrix is defined by the recursive formula below, and the *sequency-ordered* Hadamard matrix is formed by rearranging the rows so that the number of sign changes in a row is in increasing order.Confusingly, different sources refer to either matrix as the Walsh matrix.
The Walsh matrix (and Walsh functions) are used in computing the Walsh transform and have applications in the efficient implementation of certain signal processing operations.

We use the ahdamard matrix and do a dot product(kronecker product) to get the transformed matrix and plot the signal

2-D image = Had* 2d_image*Had(transpose)



1-D image = Had* vector(1d)



Generating Hadamard matrix

The Hadamard matrices of dimension 2^k for $k \in N$ are given by the recursive formula (the lowest order of Hadamard matrix is 2)

$$H_2 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ -H_2 & H_2 \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \\ -\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

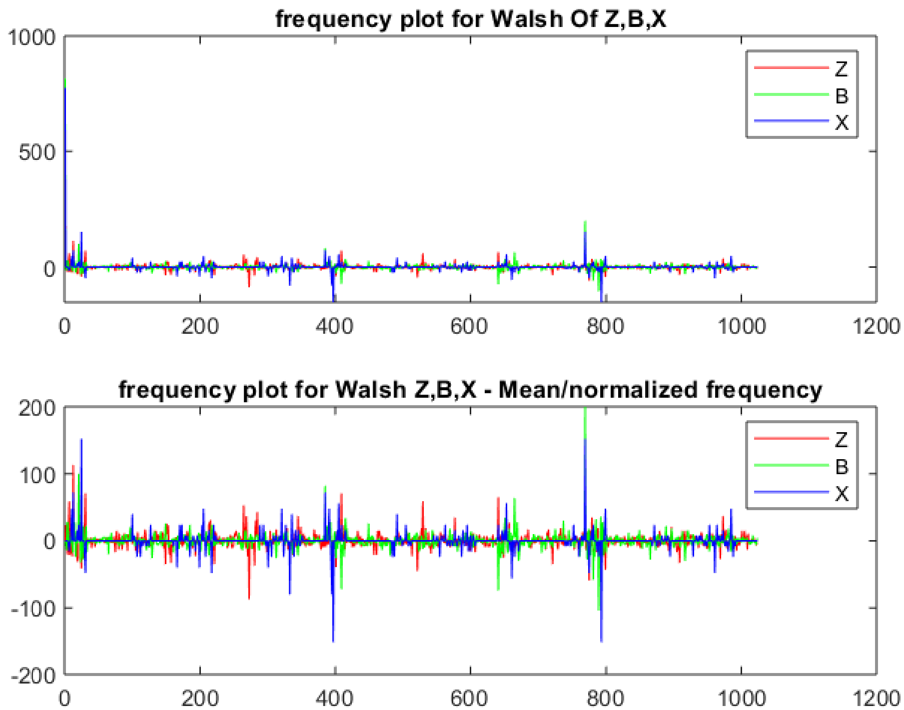
The image transform is as follows


Hadamard matrix (64 x64)

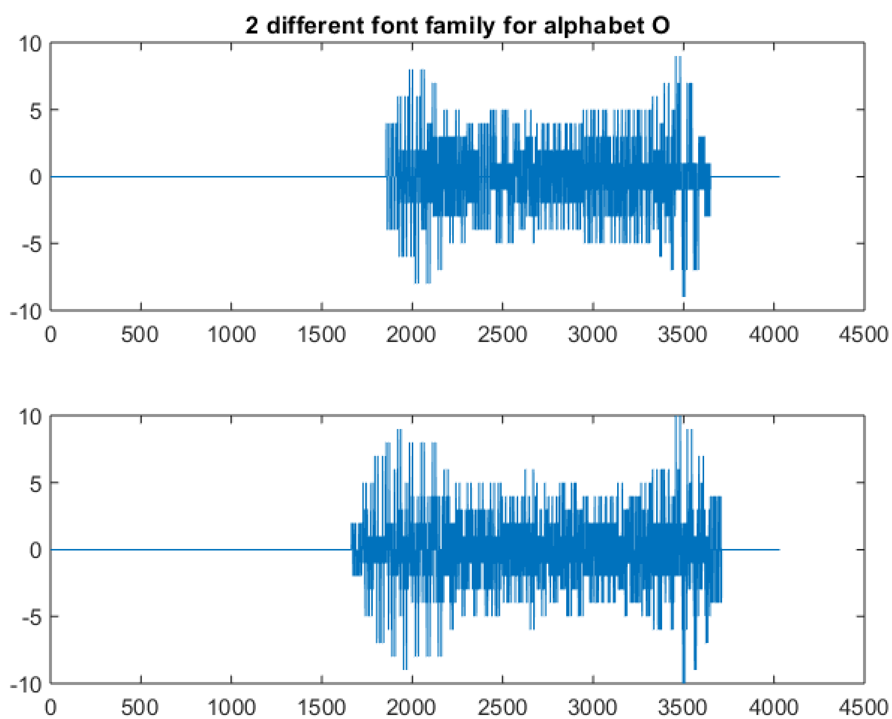
Greyscaled and Binarized image

The transformed matrix
It still retains the property of
the alphabet

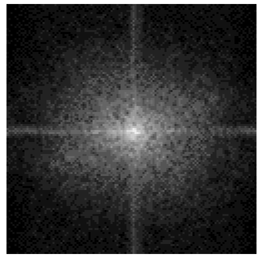
The Plot of 1D vector transform of hadamard transform applied to 3 different alphabet image B,Z,X



The plot for 2 diff O - the resemblance in pattern



Feature 7 - Fourier transform



Common Names: Fourier Transform, Spectral Analysis, Frequency Analysis

The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the *Fourier* or frequency domain, while the input image is the spatial domain equivalent. In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image.
The Fourier Transform is used in a wide range of applications, such as image analysis, image filtering, image reconstruction and image compression.

How It Works
As we are only concerned with digital images, we will restrict this discussion to the *Discrete Fourier Transform* (DFT).
The DFT is the sampled Fourier Transform and therefore does not contain all frequencies forming an image, but only a set of samples which is large enough to fully describe the spatial domain image. The number of frequencies corresponds to the number of pixels in the spatial domain image, *i.e.* the image in the spatial and Fourier domain are of the same size.
For a square image of size N×N, the two-dimensional DFT is given by:

$$F(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j)\; e^{-\iota 2\pi \left(\frac{ki}{N} + \frac{lj}{N}\right)}$$

where *f(a,b)* is the image in the spatial domain and the exponential term is the basis function corresponding to each point *F(k,l)* in the Fourier space. The equation can be interpreted as: the value of each point *F(k,l)* is obtained by multiplying the spatial image with the corresponding base function and summing the result.

The basis functions are sine and cosine waves with increasing frequencies, *i.e .*
F(0,0) represents the DC-component of the image which corresponds to the average brightness and *F(N-1,N-1)* represents the highest frequency.In a similar way, the Fourier image can be re-transformed to the spatial domain. The inverse Fourier transform is given by:

$$f(a,b) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l)\; e^{\iota 2\pi \left(\frac{ka}{N} + \frac{lb}{N}\right)}$$

Note the $\frac{1}{N^2}$ normalization term in the inverse transformation. This normalization is sometimes applied to the forward transform instead of the inverse transform, but it should not be used for both.$$

To obtain the result for the above equations, a double sum has to be calculated for each image point. However, because the Fourier Transform is *separable*, it can be written as

$$F(k,l) = \frac{1}{N} \sum_{b=0}^{N-1} P(k,b)\, e^{-\iota 2\pi \frac{lb}{N}}$$

where

$$P(k,b) = \frac{1}{N} \sum_{a=0}^{N-1} f(a,b)\, e^{-\iota 2\pi \frac{ka}{N}}$$

Using these two formulas, the spatial domain image is first transformed into an intermediate image using *N* one-dimensional Fourier Transforms. This intermediate image is then transformed into the final image, again using *N* one-dimensional Fourier Transforms. Expressing the two-dimensional Fourier Transform in terms of a series of *2N* one-dimensional transforms decreases the number of required computations.

Even with these computational savings, the ordinary one-dimensional DFT has N^2 complexity. This can be reduced to $N \log_2 N$ if we employ the *Fast Fourier Transform* (FFT) to compute the one-dimensional DFTs. This is a significant improvement, in particular for large images. There are various forms of the FFT and most of them restrict the size of the input image that may be transformed, often to $N = 2^n$ where *n* is an integer. The mathematical details are well described in the literature.

The Fourier Transform produces a complex number valued output image which can be displayed with two images, either with the *real* and *imaginary* part or with *magnitude* and *phase*. In image processing, often only the magnitude of the Fourier Transform is displayed, as it contains most of the information of the geometric structure of the spatial domain image. However, if we want to re-transform the Fourier image into the correct spatial domain after some processing in the frequency domain, we must make sure to preserve both magnitude and phase of the Fourier image.

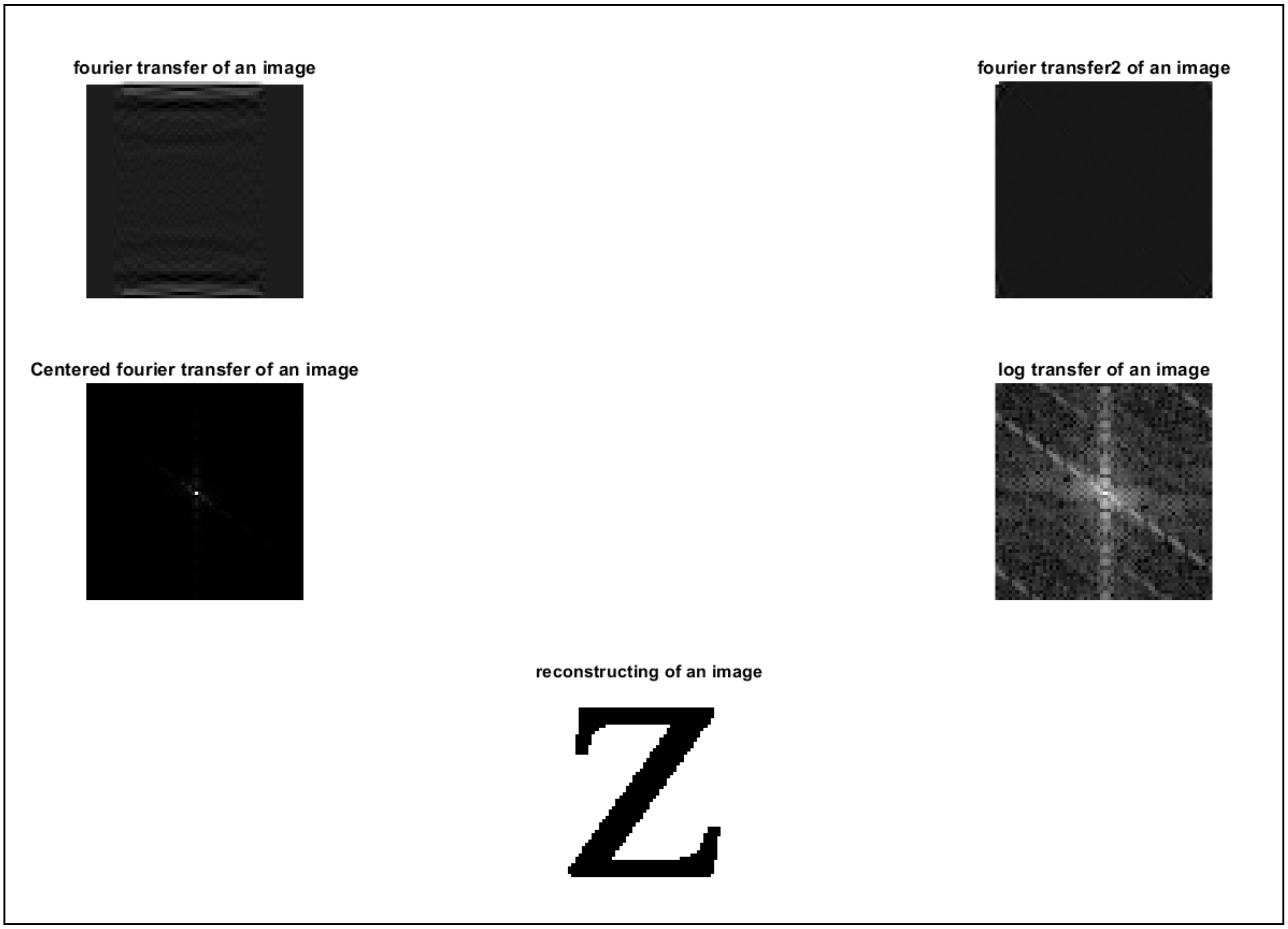
The Fourier domain image has a much greater range than the image in the spatial domain. Hence, to be sufficiently accurate, its values are usually calculated and stored in float values.

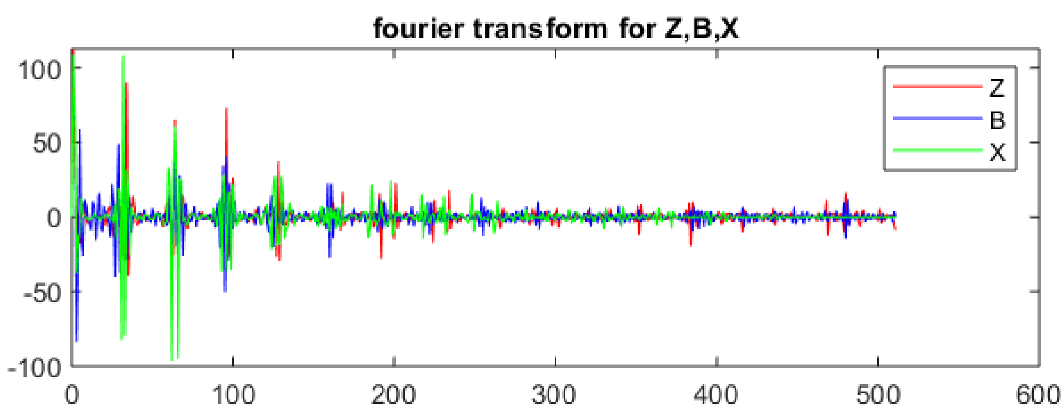
The Fourier Transform is used if we want to access the geometric characteristics of a spatial domain image. Because the image in the Fourier domain is decomposed into its sinusoidal components, it is easy to examine or process certain frequencies of the image, thus influencing the geometric structure in the spatial domain.

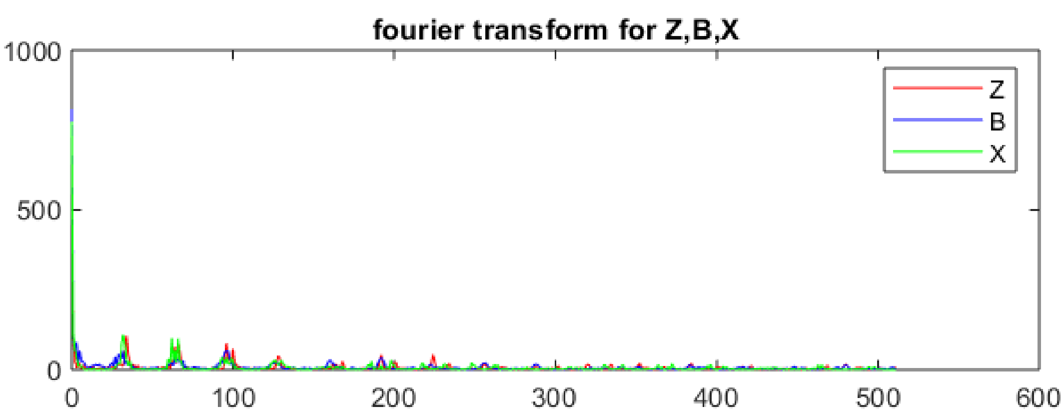
In most implementations the Fourier image is shifted in such a way that the DC-value (*i.e.* the image mean) *F(0,0)* is displayed in the center of the image. The further away from the center an image point is, the higher is its corresponding frequency.

A Fourier-Transformed image can be used for frequency filtering.

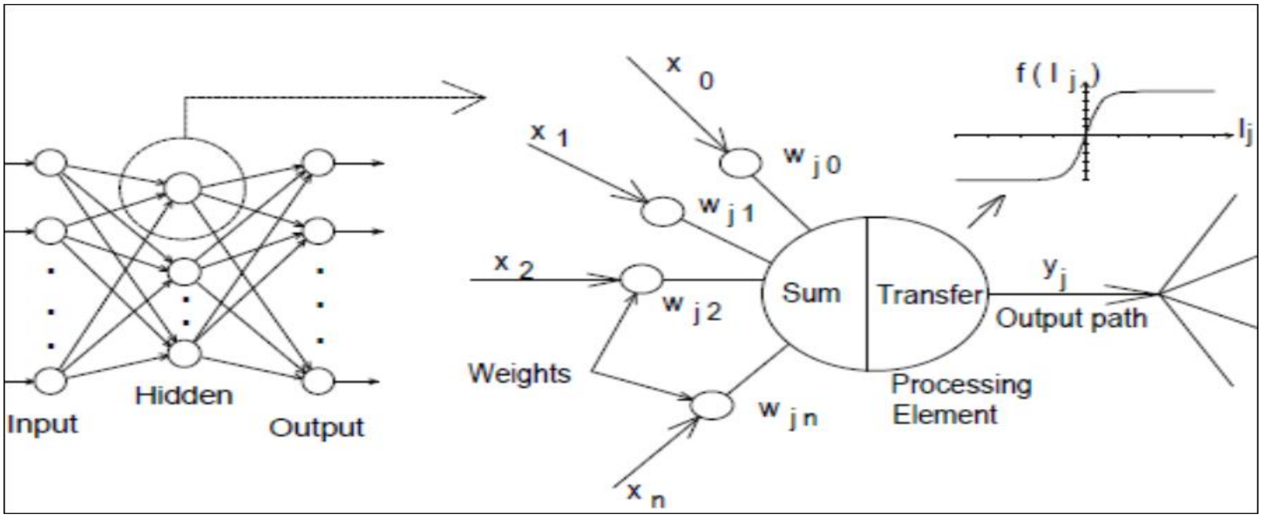
The different transformation using fourier


fourier transfer of an image
fourier transfer2 of an image
Centered fourier transfer of an image
log transfer of an image
reconstructing of an image
Z


fourier transform for Z,B,X
Z
B
X
100
50
0
-50
-100
0
100
200
300
400
500
600


fourier transform for Z,B,X
Z
B
X
1000
500
0
0
100
200
300
400
500
600

ARTIFICIAL NEURAL NETWORK



An **artificial neural network** is a network of simple elements called artificial neurons, which receive input, change their internal state (activation) according to that input, and produce output depending on the input and activation.

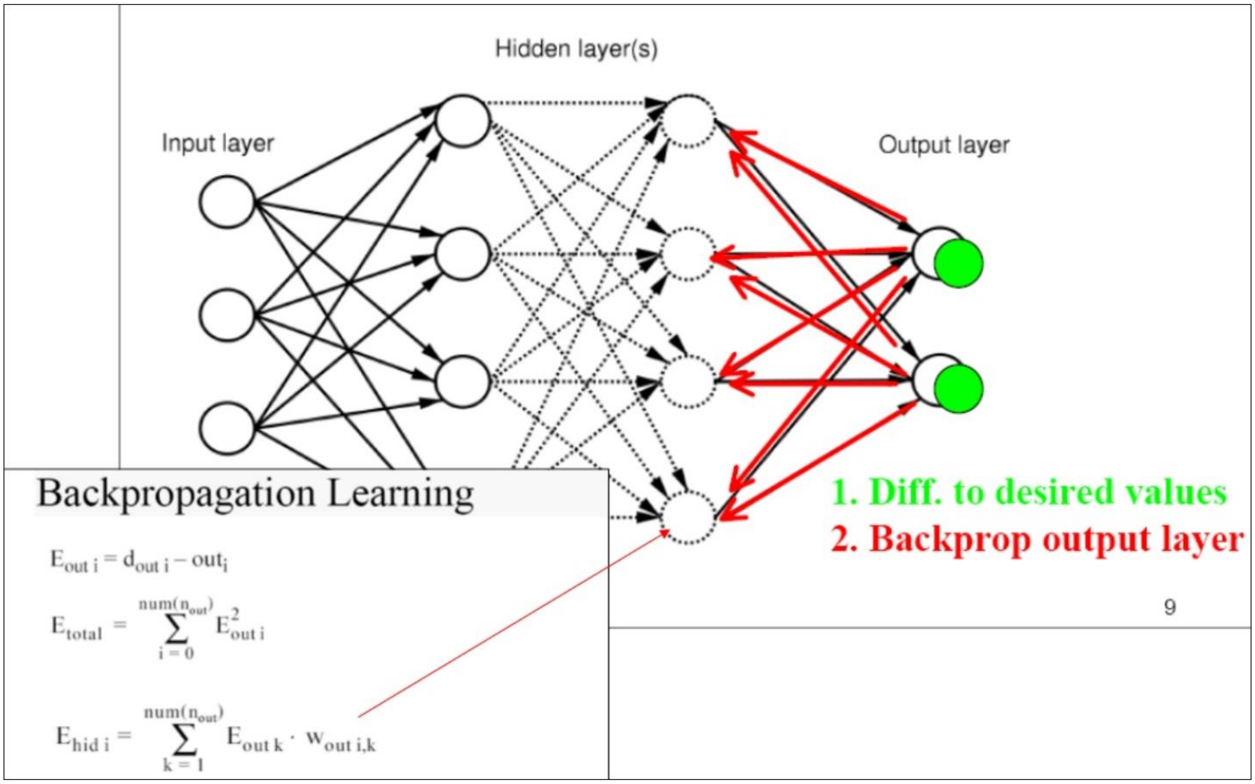
An artificial **neuron** mimics the working of a biophysical neuron with inputs and outputs, but is not a biological neuron model.

The network forms by connecting the output of certain neurons to the input of other neurons forming a directed, weighted graph. The weights as well as the functions that compute the activation can be modified by a process called learning which is governed by a **learning rule.**

Before the character recognition can happen, the ANN is 'trained', so it can build up the ability of mapping different mappings to the required yields and viably classify different characters. For preparing the ANN, we utilize the **'Vectors'** produced by the **'Database Templates'** utilizing the previously mentioned Feature Extraction methods.

The previously mentioned 7 unique sorts of highlights have been utilized to produce 11 parameters (some of same kind however extraordinary qualities), which are encouraged to the ANN. In this way, a lattice of 11x36 qualities is encouraged to the ANN to get 36 distinct qualities at the yield, one for each character in the database.

BACKPROPAGATION



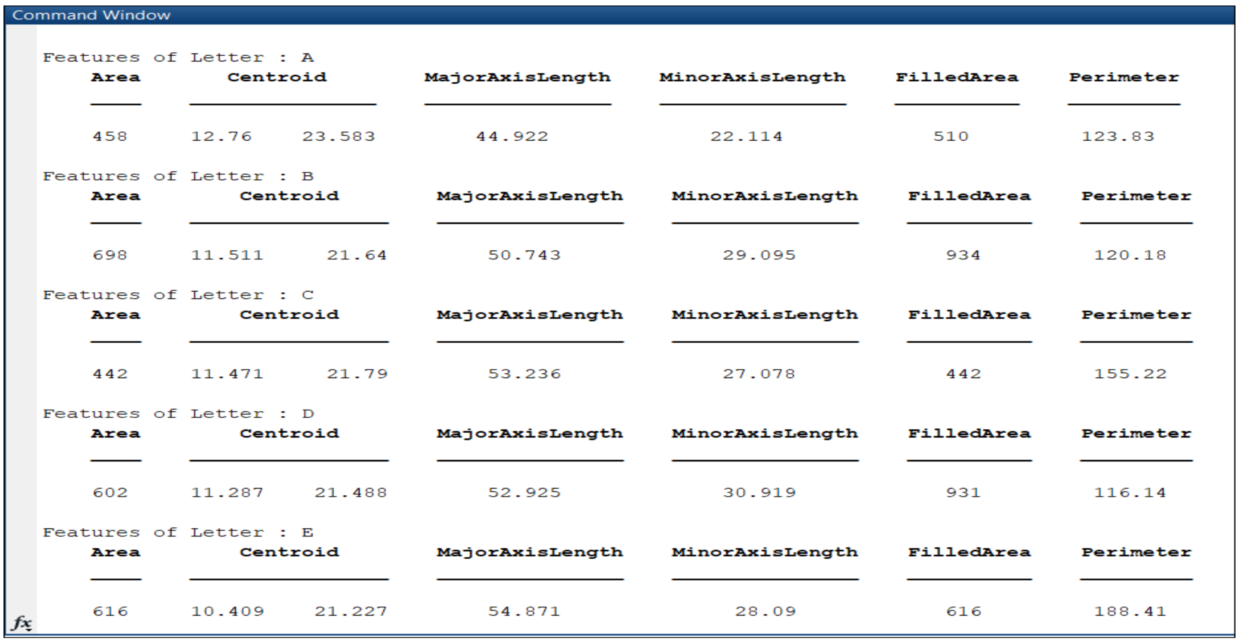
ANN can be discriminatively **trained** with the standard **backpropagation** algorithm. Backpropagation is a method to calculate the gradient of the loss function (produces the cost associated with a given state) with respect to the weights in an ANN.

It might be noticed that the ANN utilizes Backpropagation calculation for Learning. The 'Objective' values are indicated by the framework developer to oblige for little recognition errors, which might be changed from application to application.

The ANN was trained for 1000 iterations, which took around 5 seconds to finish. The Training Function was set to utilize **'Sum Squared Error'** as opposed to 'Mean Squared Error', on the grounds that the framework expected to ascertain the impact of joint errors in every one of the parameters, instead of by and large error. An error objective of 0.0001 or 0.01% was accomplished by the ANN.

EXPERIMENTAL Outcome & Summary

The ANN based system has shown promising results due to the fact that despite being trained only on a single set of templates. The system was tested upon characters and it shows a recognition rate of approximately 95% for the single font family, available for use, i.e. it recognizes 34 characters correctly out of 36.


Command Window

Features of Letter : A

Area	Centroid		MajorAxisLength	MinorAxisLength	FilledArea	Perimeter
458	12.76	23.583	44.922	22.114	510	123.83

Features of Letter : B

Area	Centroid		MajorAxisLength	MinorAxisLength	FilledArea	Perimeter
698	11.511	21.64	50.743	29.095	934	120.18

Features of Letter : C

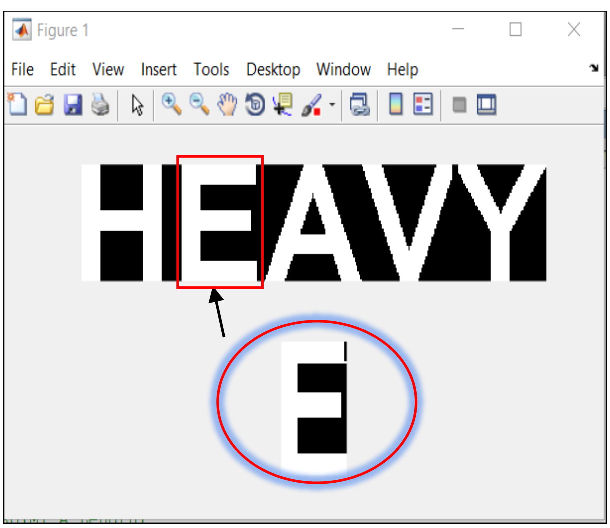
Area	Centroid		MajorAxisLength	MinorAxisLength	FilledArea	Perimeter
442	11.471	21.79	53.236	27.078	442	155.22

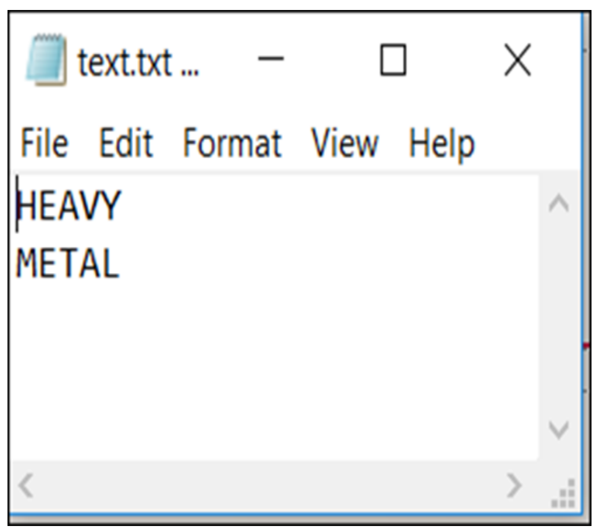
Features of Letter : D

Area	Centroid		MajorAxisLength	MinorAxisLength	FilledArea	Perimeter
602	11.287	21.488	52.925	30.919	931	116.14

Features of Letter : E

Area	Centroid		MajorAxisLength	MinorAxisLength	FilledArea	Perimeter
616	10.409	21.227	54.871	28.09	616	188.41





FUTURE SCOPE

- What does the future hold for OCR? Given enough entrepreneurial designers and sufficient research and development dollars, OCR can become a powerful tool for future data entry applications. However, the limited availability of funds in a capital-short environment could restrict the growth of this technology. But, given the proper impetus and encouragement, a lot of benefits can be provided by the OCR system. They are:-
    1. The automated entry of data by OCR is one of the most attractive, labor reducing technology
    2. The recognition of new font characters by the system is very easy and quick.
    3. We can edit the information of the documents more conveniently and we can reuse the edited information as and when required.
    4. The extension to software other than editing and searching is topic for future works. The Grid infrastructure used in the implementation of Optical Character Recognition system can be efficiently used to speed up the translation of image based documents into structured documents that are currently easy to discover, search and process.

- FUTURE ENHANCEMENTS
  The Optical Character Recognition software can be enhanced in the future in different kinds of ways such as
    1. Training and recognition speeds can be increased greater and greater by making it more user-friendly.
    2. Many applications exist where it would be desirable to read handwritten entries. Reading handwriting is a very difficult task considering the diversities that exist in ordinary penmanship. However, progress is being made

REFERENCES

1. For the complete reference and understanding of neural networks refer jeff heaton's chapter 1 from www.jeffheaton.com
2. For the complete reference and understanding of OCR refer jeff heaton's chapter 7 from www.jeffheaton.com
3. Image Processing toolbox by matlab https://www.mathworks.com/help/images/index.html
4. OCR basics 101 by buzztech https://buzztech.in/fundamental-steps-in-digital-image-processing/