# Introduction



## Optical Character Recognition

OCR is the acronym for Optical Character Recognition. This technology allows a machine to automatically recognize characters through an optical mechanism. Human beings recognize many objects in this manner our eyes are the "optical mechanism." But while the brain "sees" the input, the ability to comprehend these signals varies in each person according to many factors. By reviewing these variables, we can understand the challenges faced by the technologist developing an OCR system. First, if we read a page in a language other than our own, we may recognize the various characters, but be unable to recognize words. However, on the same page, we are usually able to interpret numerical statements - the symbols for numbers are universally used. This explains why many OCR systems recognize numbers only, while relatively few understand the full alphanumeric character range. Second, there is similarity between many numerical and alphabetical symbol shapes. For example, while examining a string of characters combining letters and numbers, there is very little visible difference between a capital letter "O" and the numeral "0." As humans, we can re-read the sentence or entire paragraph to help us determine the accurate meaning. This procedure, however, is much more difficult for a machine. Third, we rely on contrast to help us recognize characters. We may find it very difficult to read text which appears against a very dark background, or is printed over other words or graphics. Again, programming a system to interpret only the relevant data and disregard the rest is a difficult task for OCR engineers.

OCR is aimed to enable computers to recognize optical characters without human intervention. The applications of OCR have been found in automated guided vehicles(AGV), object recognitions, digital libraries, invoice and receipt processing, and recently on personal digital assistants (PDAs).

## Historical Perspective

The father of OCR is said to be Lawrence Roberts who conducted first experiments on automatic text recognition on the MIT in 1960. First practical appliances of OCR as hardware solutions appeared in 1965. Back then the recognition was limited to specially designed fonts like OCR-A and OCR-B. In 1976 Ray Kurzweil developed the first omnifont, i.e. font independent OCR system. With increasing computer performance software-based OCR solutions have gained more and more importance since the mid-eighties.

## Important Terms Used In Optical Character Recognition

### Character

A character is a standardized code for storing and displaying a letter, a digit or a special symbol on a computer. The set of all allowed characters is also called a [character set](http://www.ocr-systeme.de/english/ocrallg.htm#Zeichensatz).

### Character Recognition

[Character Recognition](http://www.ocr-systeme.de/english/glossary/zeichenerkennung.htm) is a synonym for the automatic conversion of printed [characters](http://www.ocr-systeme.de/english/ocrallg.htm#Zeichen) into editable text files.

### Character Set

A character set is the amount of all [characters](http://www.ocr-systeme.de/english/ocrallg.htm#Zeichen) displayable on a computer system. Depending on the application there are 8 bit characters sets with 256 characters (e.g. ANSI or ASCII) and 16 bit character sets with many thousand characters (Unicode). For 8 bit character sets there are many national variations, so called code pages for displaying language-specific characters.

### Color Depth

The color depth of a raster image determines the maximum number of different values for each image point. Usually the color depth is 1 Bit (back and white), 8 Bits (256 shades of gray) or 24 Bit (color). For [OCR](http://www.ocr-systeme.de/english/ocrallg.htm#OCR) purposes black-and-white images are fully sufficient as OCR engines cannot use grayscale or color information.

### Document Recognition

Document recognition is the process of converting a document pictured by a [raster image](http://www.ocr-systeme.de/english/ocrallg.htm#Rastergrafik) into an editable form. For document recognition printed documents will usually be scanned and converted into a text file (e.g. DOC, HTML, PDF, TXT) using an [OCR software](http://www.ocr-systeme.de/english/ocrallg.htm#OCR-Software) for further editing in a word processor.

### Feature Extraction

With feature extraction all characters will be divided into geometric elements like lines, arcs and circles and the combination of these elements will be compared with stored combinations of known characters.

### Ligature

In typography, a ligature occurs where two or more letters are joined as a single symbol, e.g. fi. For [OCR software](http://www.ocr-systeme.de/english/ocrallg.htm#OCR-Software)the difficulty with ligatures is to recognise them as two chracters during [Segmentation](http://www.ocr-systeme.de/english/ocrallg.htm#Segmentierung).

### Resolution

The resolution of a scanner is a measure of how dense the image will be sampled. It is measured in dpi (dots per inch). For [OCR](http://www.ocr-systeme.de/english/ocrallg.htm#OCR) purposes a resolution between 300 and 400 dpi will be sufficient, for scanning images for reproductional purposes resolutions of 1200 dpi and higher would be used.

### Scanning

Scanning in conjunction with [OCR](http://www.ocr-systeme.de/english/ocrallg.htm#OCR) is the process of optical rasterization of a printed document by using a [scanner](http://www.ocr-systeme.de/english/ocrallg.htm#Scanner). The document will be divided into image points and each point will be assigned a value describing black and white, grayscale or color information

### Segmentation

Segmentation is a processing step of [OCR](http://www.ocr-systeme.de/english/ocrallg.htm#OCR). During segmentation the text image will be divided into lines and the lines will be divided into [characters](http://www.ocr-systeme.de/english/ocrallg.htm#Zeichen) or [ligatures](http://www.ocr-systeme.de/english/ocrallg.htm#Ligatur).

# Theory Involved in Optical Character Recognition



## Basic Process of Optical Character Recognition

OCR usually includes two stages, “low-level” and “high-level”. Low level involves extracting features from images such as the boundary of a character or regions with the same texture. The task of “high-level is then to recognize these objects with the extracted features. This is concerned with both “low-level” and “high-level” stages, in particular with finding properties of an image which are invariant to image transformations including scale, translation and rotation.

This is accomplished by searching a match between the features extracted from the given character’s image and the library of image models. Ideally, we would like the features to be distinct for different character images so that the computer can extract the correct model from the library without any confusion. At the same time, we also want the features to be robust enough so that they will not be affected by viewing transformations, noises, resolution variations and other factors.

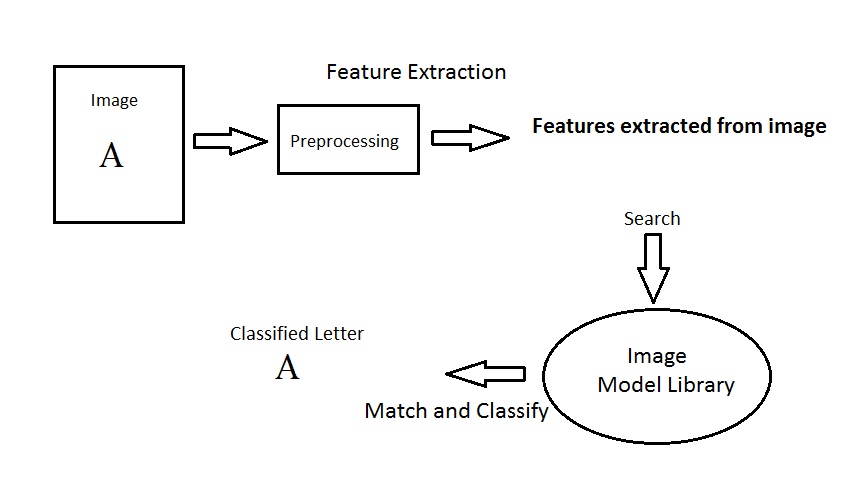


Figure ‑: The basic process of an OCR system

## Different Families of Character Recognition

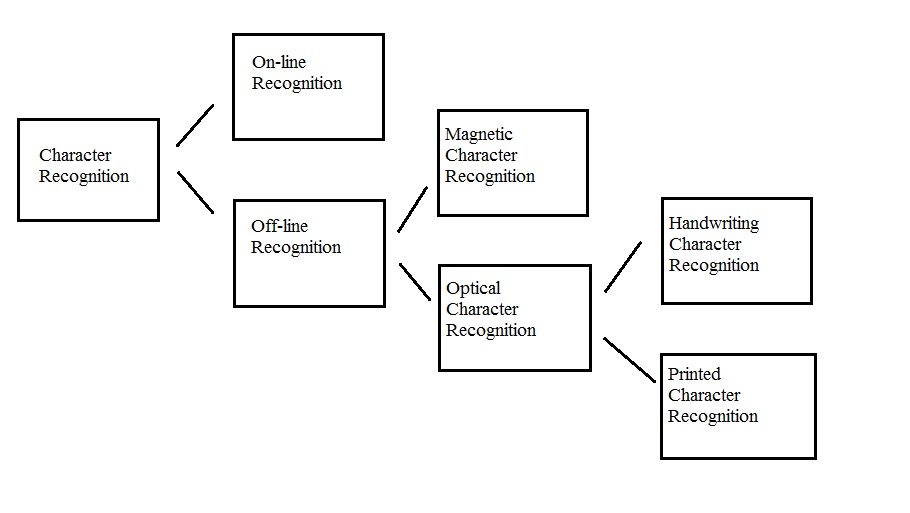


Figure ‑: Different Families of Character Recognition

### Online Recognition

Here the user is directly connected to the system using an electronic pen or a touch screen and recognition is carried out in real-time. On-line character recognition deals with a data stream which comes from a transducer while the user is writing. The typical hardware to collect data is a digitizing tablet which is electromagnetic or pressure sensitive. When the user writes on the tablet, the successive movements of the pen are transformed to a series of electronic signal which is memorized and analyzed by the computer.

### Offline Recognition

Here recognition is carried out on handwritten text that is captured using a scanner or a camera.

Thus the text is treated as an image. Offline methods does not have any temporal information. However using certain heuristic methods and prior knowledge (for example in the Roman script, writing is always performed from left to right) we can determine the direction of the strokes with respect to time.

Off-line character recognition is performed after the writing is finished. The major difference between on-line and off-line character recognition is that on-line character recognition has time-sequence contextual information but off-line data does not.

The off-line character recognition can be further classified into MCR and OCR.:

#### Magnetic character recognition (MCR)

In MCR, the characters are printed with magnetic ink. The reading device can recognize the characters according to the unique magnetic field of each character. MCR is mostly used in banks for check authentication..

#### Optical character recognition (OCR)

OCR deals with the recognition of characters acquired by optical means, typically a scanner or a camera. The characters are in the form of pixilated images, and can be either printed or handwritten, of any size, shape, or orientation. The OCR can be subdivided into handwritten character recognition and printed character recognition.. Handwritten character recognition is more difficult to implement than printed character recognition due to the diversified human handwriting styles and customs. In printed character recognition, the images to be processed are in the forms of standard fonts like *Times New Roman*, *Arial*, *Courier*, etc.

## Image Features

Image features are unique characteristics that can represent a specific image. Image features are meaningful, detectable parts of the image. To overcome the vulnerabilities of template matching and reduce the computation cost, image features are employed. Two types of characteristics are usually referred by image features:

1. **a global property** **of an image**, for instance the average gray level of all pixels included in a gray level image.

2**. a part of the image** with special properties, for example the boundary length of an image.

Good image features should satisfy the following conditions:

1. **robust to transformations** – the image features should be as invariant as possible to image transformations including translation, rotation, and scaling, etc.

2**. robust to noise** – the image features should be robust to noises and various degraded situations.

3**. feature extraction efficiency** – image features can be computed efficiently.

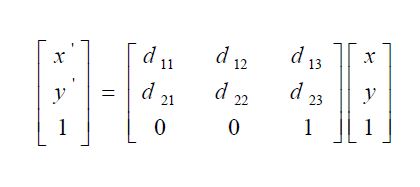
4**. feature matching efficiency** – the matching algorithms should only require a reasonable computational cost.

The selection of image features and corresponding extraction methods is probably the most important step in achieving high performance for an OCR system. At the same time, the image feature and the extraction method also decide the nature and the output of the image-preprocessing step. Some image features and the extraction algorithms work on color images, while others work on gray level or binary images. Moreover, the format of the extracted features must match the requirements of the classifier. Some features like the graph descriptions and grammar-based descriptions are well suited for structural and syntactic classifiers. Numerical features are ideal for statistical classifiers. Discrete features are ideal for decision trees.

## Affine Transformation

Images are the projection of 3D objects onto 2D planar surfaces. The spatial status of a solid 3D object is described by six parameters, three for translations and three for rotations. Image points are affected by these six parameters, and different views of coplanar object points are related by planar projective transformations. When the distance of the coplanar object points from the camera (depth) does not vary much compared with their average depth, the planar projective transformation can be approximated by the affine transformation.

The affine transformation is an important class of linear 2D geometric transformation that maps variables (*e.g.* pixel intensity values located at position (*x*, *y*) in an input image) into new variables (*e.g.* (*x’* , *y’*) in an output image) by applying a linear combination of translation, rotation, scaling and/or shearing operations. A general affine transformation can be expressed as:



Examples of affine transformations are shown in below.

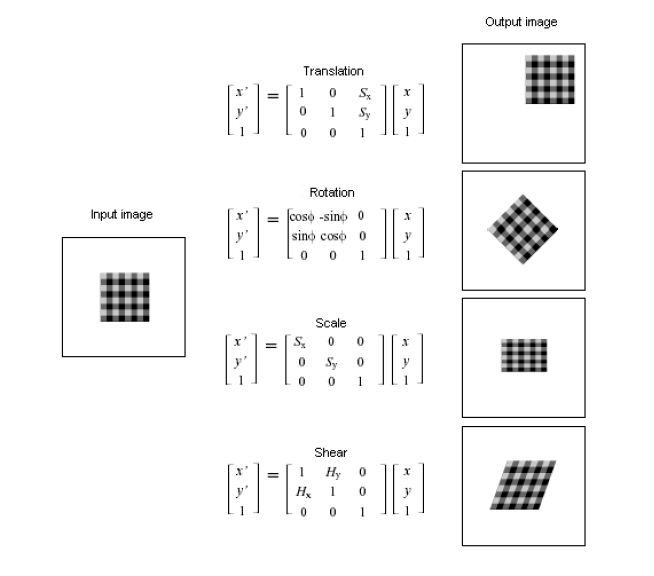


Figure ‑:Examples of affine transformations

The affine transformation preserves straight lines and parallelism. In other words, if you take a number of points that fall on a straight line from the input image and apply an affine transformation, the output image will still keep the points on a straight line. The parallel lines in the input image will still be parallel in the output image.

## Image Invariants

An important problem in OCR is the automatic recognition of a character in a scene regardless of its position, size, orientation, etc. In order to recognize different variations of the same character, image features which are invariant to certain transformations need to be used. Image invariants are features which have approximately the same values for samples of the same image which are, for instance, translated, scaled, rotated, skewed, blurred, or noise affected.

Shape-basedimage invariants are the most commonly used image features. Image recognition based on these invariants includes three major issues: shape representation, shape similarity measure and shape indexing. Among these issues, shape representation is the most important issue. Various shape representation methods and shape descriptors exist in literatures. These methods can be classified into two categories.

### Boundary-based invariants

In boundary-based invariants, only the contour information of the shape is explored.

Boundary is one of the most straightforward and important image features as human beings tend to perceive scenes as being composed of different individual objects, which can be best identified by their boundaries. Meanwhile, as far as the implementation is concerned, boundary is also very simple to calculate.

The common boundary-based invariants include chain code and Fourier descriptors.

for image representation and data reduction. Usually additional transformations are required to achieve the desired invariant properties for the image features.

#### Chain Codes

Chain code was introduced by Freeman as a means to represent lines or boundaries of shapes by a connected sequence of straight line segments of specified length and direction. A chain code has two components: the coordinates of the starting point and a chain of codes representing the relative position of the starting pixel and its followers. The chain code is generated by using the changing direction of the connected pixels contained in a boundary. The representation is based on 4- connectivity or 8- connectivity of the segments.

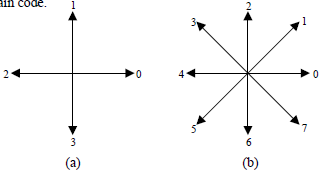


Figure ‑(a)4-connectivity chain code.(b)8-connectivity chain code

For a closed boundary, its chain code depends on the starting point of the boundary. To make it invariant to the starting point, the chain code can be normalized according to the following method: A chain code can be treated as a circular sequence of direction numbers. Therefore, the starting point can be redefined so that the resulting sequence of numbers forms a minimum integer.

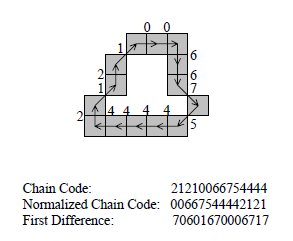


Figure ‑:An example of the 8-connectivity chain code

The chain code can also be normalized for rotation by using the *first difference* instead of the code itself. The *first difference* is generated by counting the number ofdirection changes (counterclockwise) of two adjacent pixels. The first code can becomputed by counting the direction changes between the end point and the beginningpoint of the chain. The *first difference* will keep the chain codeinvariant to rotation only if the boundary itself is invariant to rotation change.

#### Fourier Descriptors

Fourier descriptors were introduced to describe the shape of a closed, planar figure. Given a close shape in a 2D Cartesian plane, the boundarie*s* can be traced and re-sampled according to, say, the counterclockwise direction, to obtain an uniformly distributed *K* points as shown in Figure.



Figure ‑:A re-sampled boundary and its representation as a complex sequence

Each point’s coordinates can be expressed as *(x0, y0)*, *(x1,* *y1)*, *(x2, y2)*,…, *(xK-1, xK-1)*. These coordinates can be expressed in the form *x(k) = xk* and *y(k) = yk* . Under this condition, the boundary can be expressed as a sequence of complex numbers:

*s(k) = x(k) + j y(k) , k = 0, 1, 2, …, K-1*

That means the *x*-axis is treated as the real axis and the *y*-axis as the imaginary axis of a sequence of complex numbers. The coefficients of Discrete Fourier Transform (DFT) of this complex sequence *z(u)* is:

-(1)

The complex coefficients *z(u)* are called the Fourier descriptors of the boundary

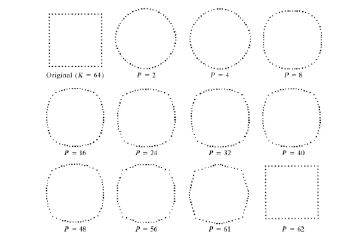
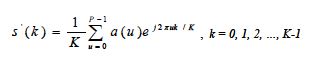


Figure ‑: Examples of reconstruction from Fourier descriptors. P is the number of Fourier coefficients used in the reconstruction process

According to the property of the Fourier transform, low-frequency components determine global shape, and high-frequency components determine the fine details.

We can reconstruct the boundary using the first *P* coefficients with the inverse Fourier transform:

-(2)

Although only *P* coefficients are used to obtain the boundary *s(k)* , *k* still ranges from *0* to *K-1*. In other words, the same number of points is in the approximated boundary, but not as many terms are used in the reconstruction of each point. Above Figure shows the example of the reconstruction results of a square boundary with different number of coefficients from 2 to 62. From this example we see that when higher-order coefficients used, the reconstructed contour is more like the original shape. The Fourier descriptors have the advantage of reducing a 2-D problem to a 1-D problem. The significant Fourier descriptors can be used as the basis for classifying different boundary shapes.

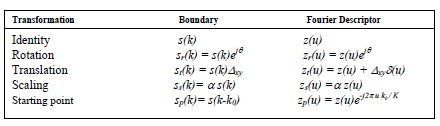


Figure ‑: Basic properties of Fourier descriptors under some transformations

The symbol Δ*xy =* Δ*x + j*Δ*y*. δ*(u)* is an impulse function which only has non-zero values at the origin and zero values everywhere else.

From the table, we can see that the magnitudes of *z(u)* are invariant to image rotations because:

 -(3)

The image translation consists of adding a consistent displacement to all sample coordinates of the boundary. The translation has no effect on the descriptors except for *u=0* because of the impulse function δ*(u)*. The first component of the Fourier descriptors depends only on the position of the shape, it is not useful in describing the shape and can be discarded.

The invariance to scale changes can be achieved by forming the ratio of two coefficients so that we can get rid of the parameter α.

The magnitudes of *z(u)* are invariant to the change of starting point because

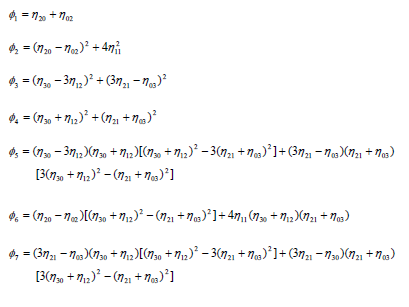
Capture 31 -(4)

### Region-Based Invariants

Boundary-based invariants such as chain code and Fourier descriptors explore only the contour information; they cannot capture the interior content of the shape. On the other hand, these methods cannot deal with disjoint shapes where single closed boundary may not be available; therefore, they have limited applications. For region-based invariants, all of the pixels of the image are taken into account to represent the shape. Because region-based invariants combine information of an entire image region rather than exploiting information just along the boundary pixels, they can capture more information from the image. The region-based invariants can also be employed to describe disjoint shapes.

#### Hu’s Moments

Based on normalized central moments, Hu introduced seven nonlinear functions which are translation, scale, and rotation invariant. The seven moment invariants are defined as:



Hu’s seven moment invariants are invariant to image transformations including scaling, translation and rotation. However, this set of moment invariants is not invariant to contrast changes.

Hu’s seven moment invariants have been widely used in pattern recognition, and their performance has been evaluated under various deformation situation including blurring, spatial degradations , random noise, skew and perspective transformations.

As Hu’s seven moment invariants take every image pixel into account, the Computation cost will be much higher than boundary-based invariants.

# OCR Engine



## OCR Engine

An OCR engine is a system which can load an image, preprocesses the image, extracts proper image features, computes the “distances” between the extracted image features and the known feature vectors stored in the image model library, and recognizes the image according to the degree of similarity between the loaded image and the image models.

The system proposed has two components : Character Segmentation and Character Recognition

### Character Segmentation

The character segmentation has been done in two parts -

1. Dividing the image into lines.
2. Segmenting every line into individual characters.

A preprocessing is done by converting all coloured images to black and white images, thresholding the image and removing all background noise. The seperation into lines is done by identifying a blank horizontal line of pixels as line dividers. For each line, the connected components are found with each connected component representing a possible character. For each connected component, the rectangle enclosing the connected component is taken as the character segment. The approach has been chosen keeping in mind the challenges to the system of robustness and the availability of low resolution images.



Figure ‑: Character Segmentation

### Character Recognition

The segmented characters are then characterised by the recognition process using the process of classification.

## Classification Process

There are two steps in building a classifier: training and testing.



Figure ‑: Classification Process

### Training

a. *Pre-processing* – Processes the data so it is in a suitable form

b. *Feature extraction* – Reduce the amount of data by extracting *relevant* information—Usually results in a vector of scalar values. (We also need to NORMALIZE the features for distance measurements!)

c. *Model Estimation* – from the finite set of feature vectors, need to estimate a model (usually statistical) for each class of the training data

### Testing

a. *Pre-processing*

b. *Feature extraction*

c. *Classification* – Compare feature vectors to the various models and find the closest match.

## OCR – Pre processing

The preprocessing stage aims to make the image be suitable for different feature extraction algorithms. Some feature extraction algorithms only deal with the contours of the image while some algorithms calculate every pixel of the image. On the other hand, the initial image may be noise affected, or blurred by other reasons. The preprocessing stage can make the initial image more suitable for later computation.

These are the pre-processing steps often performed in OCR

#### Binarization

Usually presented with a grayscale image, binarization is then simply a matter of choosing a threshold value.

#### Morphological Operators

Remove isolated specks and holes in characters.

#### Segmentation

Check connectivity of shapes, label, and isolate.

## OCR-Feature Extraction

Given a segmented (isolated) character , useful features are extracted. In our project we have used Hu’s moments. The basic process of computing process of Hu’s seven moment invariants is shown in the figure below:

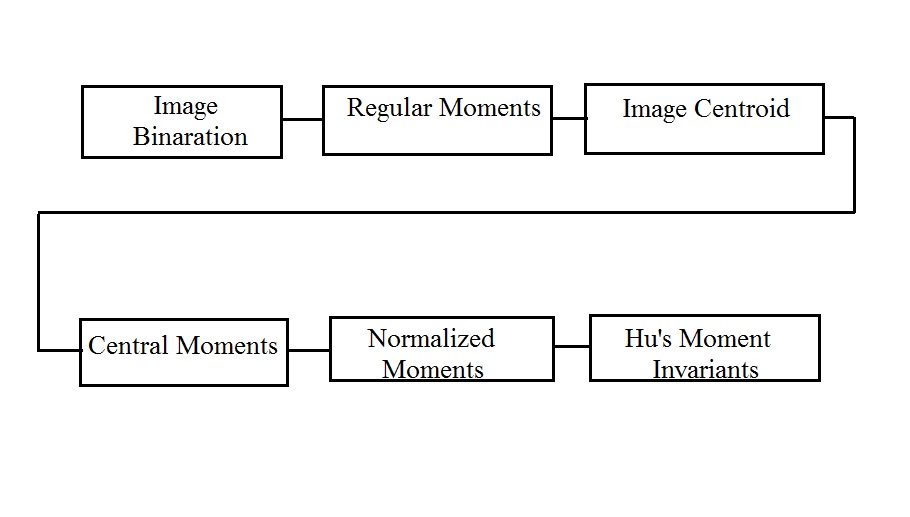


Figure ‑: Block diagram of computing Hu’s seven moment invariants

The image is first converted to binary format. The function to compute the regular moments is in the format: [*m*] = *moment* (*fig*, *p*, *q*). *fig* is the input binary image, and *p*, *q* are predefined moment’s order. With these parameters available, we do the summation according to the regular moments’ definition

 -(5)

by going through all of the pixels in the image. For a binary image, *m00* is the total number of white pixels of the image. The centroid of the binary image is computed according to 10 00 *x* = *m* /*m* and 01 00 *y* = *m* /*m* . Based on the centroid of the image, similar to the regularmoments, the central moments function is in the format: [μ] = *central moment*. This is computed according to the definition:

Capture 37 -(6)

## OCR- Model Estimation

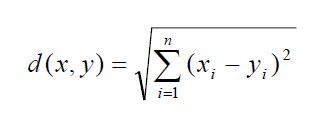
Given *labeled* sets of features for many characters, where the labels correspond to the particular classes that the characters belong to, we wish to estimate a statistical model for each character class.

## OCR-Classification

According to Tou and Gonzalez, “The principal function of a pattern recognition system is to

yield decisions concerning the class membership of the patterns with which it is confronted.” In the context of an OCR system, the recognizer is confronted with a sequence feature patterns from which it must determine the character classes.

We have used the minimum distance classifier (Euclidian Distance Classifier) The class to which a test character is assigned is that with the minimum distance. Euclidean distance is one of the most common metric distance functions and is defined as:

 -(7)

# Images Processing in MATLAB



## Representing a Binary Image in MATLAB

Binary scale images are represented as matrices of dimensions m, n,

Where,

m= number of pixels in y dimension (vertical) of the image

n= number of pixels in x dimension (horizontal) of the image

The elements in the matrix are either 0 or 1, where 0 represents black and 1 represents white. Each element represents a pixel in the same (x, y) location in the image.

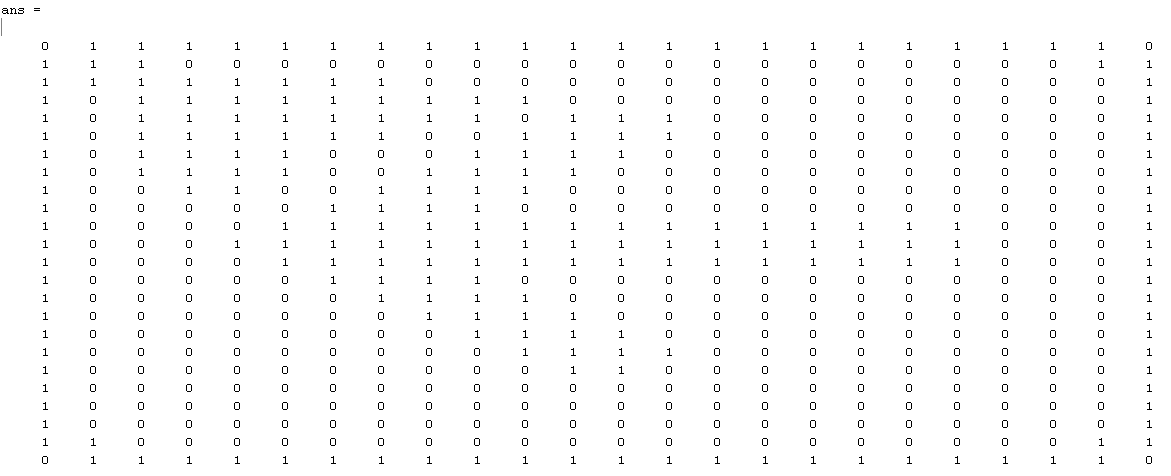
 e.g. The image  (The image dimensions and the matrix dimensions are 24, 24 ) is represented in MATLAB as :

Figure ‑ : Representation of a Binary Image in MATLAB

## Representing a Grey Scale image in MATLAB

Grey Scale images are represented as matrices of dimensions m, n, where,

m= number of pixels in y dimension (vertical) of the image

n= number of pixels in x dimension (horizontal) of the image

Here, images are usually represented the 8-bit unsigned integer class, with each pixel taking any value between 0 and 255. 0 represents black, while 255 represents white, the numbers in between represent different shades of gray.

e.g. The gray scale image  (The image dimensions and the matrix dimensions are 24, 24) is represented in MATLAB as:

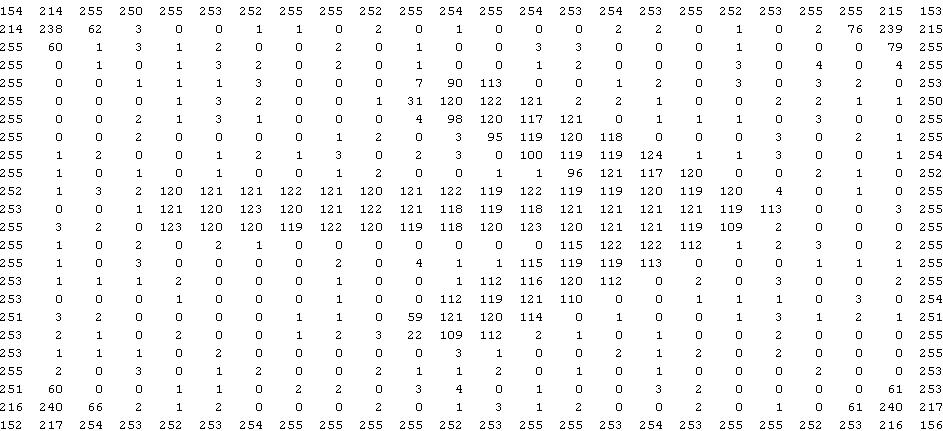
**

Figure ‑ : Representation of a Grey Scale image in MATLAB

## Representing RGB images in MATLAB

RGB images are represented in MATLAB by matrices of dimension (m, n, 3)

Where,

m= number of pixels in y dimension (vertical) of the image

n= number of pixels in x dimension (horizontal) of the image

3 represents the number of layers, Red, Green, and Blue in the image.

The 24,24 image is represented by a matrix of dimensions 24,24,3.

e.g. The image  is represented in MATLAB as

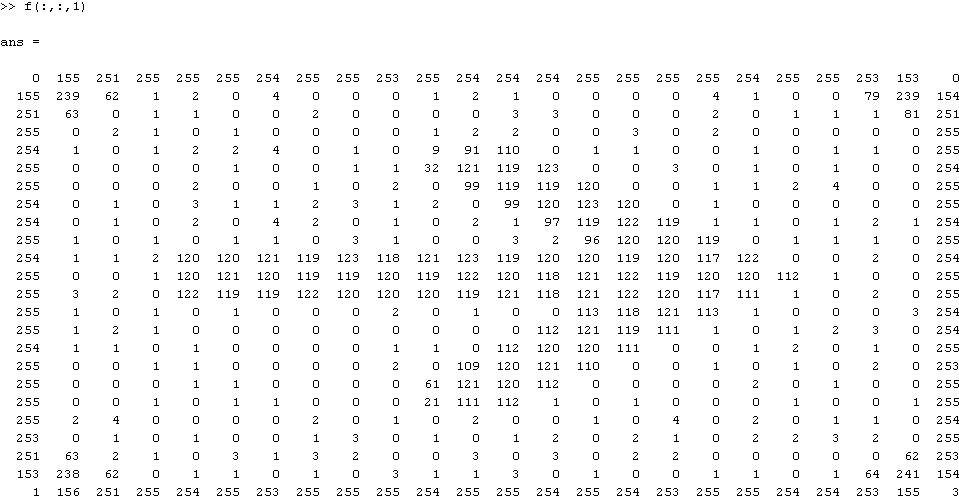
**

Figure ‑ : Representation of First layer of RGB image in MATLAB

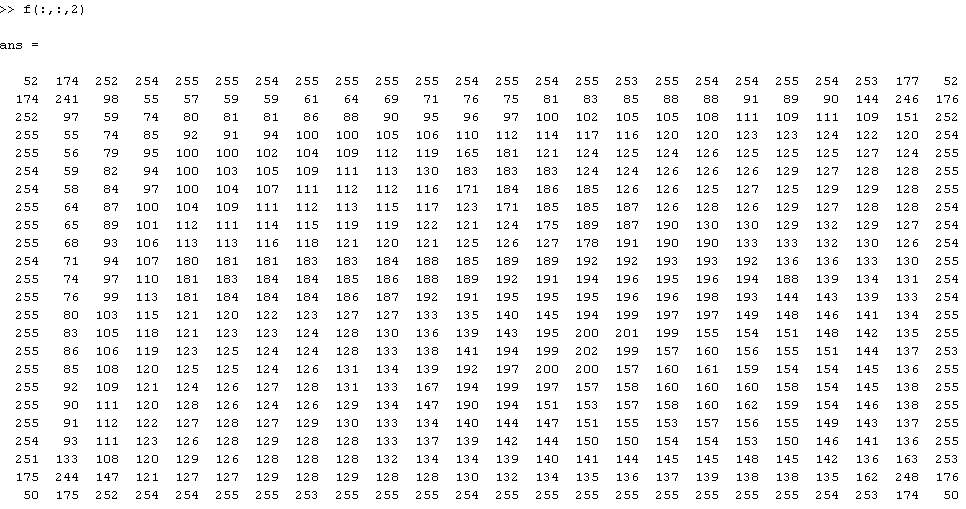
**

Figure ‑ : Representation of Second layer of RGB image in MATLAB

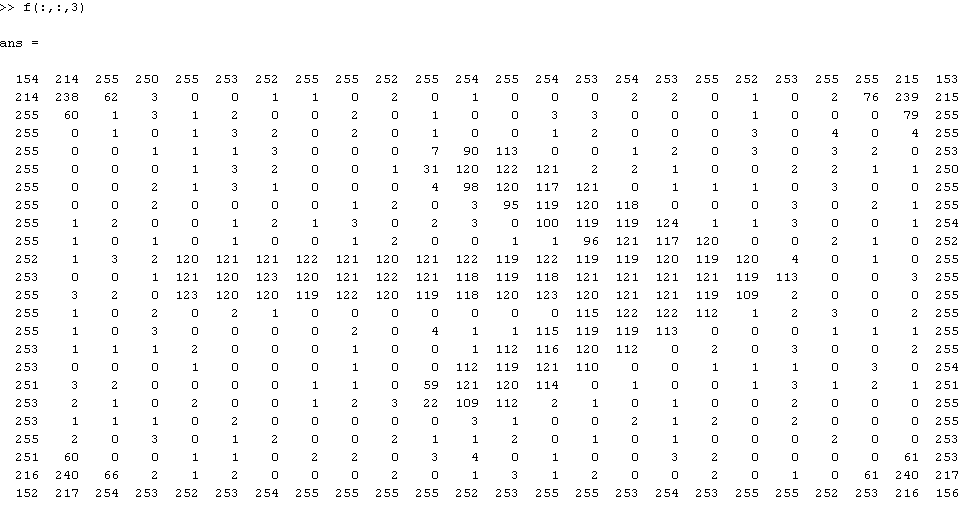
**

Figure ‑ : Representation of Third layer of RGB image in MATLAB

The first layer, accessed in MATLAB by the index (:,:,1) represents the Red layer of the RGB image. Pixel values are represented by the 8 bit unsigned integer class with 255 representing red and 0 representing black.

The second layer, accessed in MATLAB by the index (:,:,2) represents the Green layer of the RGB image. Pixel values are represented by the 8 bit unsigned integer class with 255 representing green and 0 representing black.

The third layer, accessed in MATLAB by the index (:,:,3) represents the Blue layer of the RGB image. Pixel values are represented by the 8 bit unsigned integer class with 255 representing green and 0 representing black.

## Image processing features used and MATLAB functions involved

### Thresholding

Because of its intuitive properties and simplicity of implementation, image thresholding enjoys a central position in applications of image segmentation.

During the thresholding process, individual [pixels](http://en.wikipedia.org/wiki/Pixels) in an image are marked as “object” pixels if their value is greater than some threshold value (assuming an object to be brighter than the background) and as “background” pixels otherwise. This convention is known as threshold above. Variants include threshold below, which is opposite of threshold above; threshold inside, where a pixel is labeled "object" if its value is between two thresholds; and threshold outside, which is the opposite of threshold inside. Typically, an object pixel is given a value of “1” while a background pixel is given a value of “0.” Finally, a binary image is created by coloring each pixel white or black, depending on a pixel's label. An example of “threshold” above is shown:



Figure ‑ : Image before Thresholding



Figure ‑ : Image after Thresholding

The first image (figure 4.6) is the original image and the second image (figure 4.7) shows the effect of thresholding on the image. It’s to be noted that thresholding is applied to gray scale images and results in binary images.

In RGB images, thresholding is done at all three levels(R,G,B) individually. The binary decision process of thresholding in gray scale images is represented by

 Equation .1

Here,

is the gray scale image

are the co-ordinates of the pixel

is the threshold value.

### Erosion

Erosion is one of two fundamental operations (the other being [dilation](http://en.wikipedia.org/wiki/Dilation_(morphology))) in [Morphological image processing](http://en.wikipedia.org/wiki/Morphological_image_processing) from which all other morphological operations are based. It was originally defined for [binary images](http://en.wikipedia.org/wiki/Binary_image), later being extended to [grayscale](http://en.wikipedia.org/wiki/Grayscale) images, and subsequently to [complete lattices](http://en.wikipedia.org/wiki/Complete_lattice).

In binary morphology, an image is viewed as a [subset](http://en.wikipedia.org/wiki/Subset) of an [Euclidean space](http://en.wikipedia.org/wiki/Euclidean_space)\mathbb{R}^d or the [integer](http://en.wikipedia.org/wiki/Integer) [grid](http://en.wikipedia.org/wiki/Lattice_graph) \mathbb{Z}^d, for some dimension d.

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](http://en.wikipedia.org/wiki/Structuring_element), and is itself a binary image (i.e., a subset of the space or grid).

Let E be a Euclidean space or an integer grid, and A a binary image in E. The erosion of the binary image A by the structuring element B is defined by 4.2

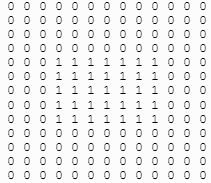
 Equation .2

Where,

is the translation of by the vector z, i.e.,

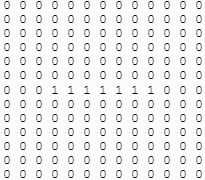
 Equation .3

To represent the above equations and words in tangible form, consider the example below: Example: Suppose A is a 13 \* 13 matrix and B is a 5 \* 1 matrix:

A= 

And B= 

Assuming that the origin B is at its center, for each pixel in A superimpose the origin of B, if B is completely contained by A the pixel is retained, else deleted. The Erosion of A by B is given by



This means that only when B is completely contained inside A that the pixels values are retained, else it gets deleted or in other words it gets eroded. The MATLAB function for erode is

**BW2 = bwmorph(BW,operation)**

Where, BW is the binary image on which erosion is to be performed

Operation 'erode' performs erosion using the structuring element ones (3) i.e. a 3x3 matrix with all elements ‘1’. Erosion is performed on the selected object to ‘filter out’ noise or irregularities at the border.

### Labeling the segmented objects

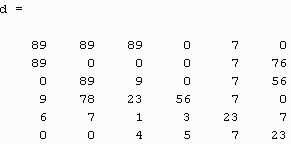
After, the objects have been segmented using thresholding, the multiple objects have to be labeled in order to differentiate between them. This is done in MATLAB by using the bwlabel function

**L = bwlabel(BW,n)** returns a matrix L,

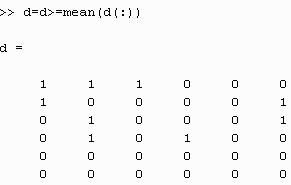
of the same size as BW, containing labels for the connected objects in BW. n can have a value of either 4 or 8, where 4 specifies 4-connected objects and 8 specifies 8-connected objects; if the argument is omitted, it defaults to 8.

The elements of L are integer values greater than or equal to 0. The pixels labeled 0 are the background. The pixels labeled 1 make up one object, the pixels labeled 2 make up a second object, and so on.

For example, consider the matrix



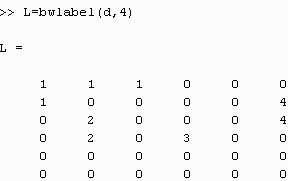
‘d’ is thresholded using the mean value of all pixels in d and the resultant is a binary matrix as shown below



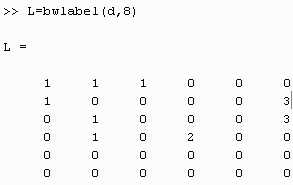
Before going further and using the ‘bwlabel’ function on ‘d’, let’s see what 4-connected and 8-connected actually means.

4-Connected pixels are neighbors to every pixel that touches one of their edges. These pixels are connected horizontally and vertically. In terms of pixel coordinates, every pixel that has the coordinates

is connected to the pixel at . e.g.

.

8-connected pixels are neighbors to every pixel that touches one of their edges or corners. These pixels are connected horizontally, vertically, and diagonally. In addition to 4-Connected pixels, each pixel with coordinates is connected to the pixel at . For e.g.

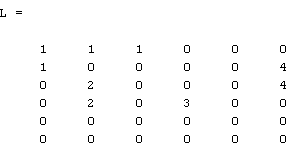


Note how the object ‘2’ in the 4-Connected matrix is a part of the object ‘1’ in the 8-Connected matrix, hence there are 3 8-connected objects in‘d’ while there are 4 4-connected objects.

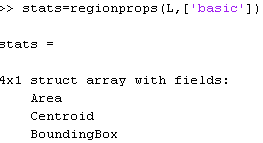
### Properties of the labeled objects

The target tracking problem requires information like the co-ordinates of the centroid (in the x-y plane) of the selected object and size of the object. The previous techniques (thresholding, bwlabel) are used to identify the objects, to extract the above mentioned properties of the object, the MATLAB function ‘regionprops’ is used.

Consider the labeled matrix ‘L’ for 4-connected objects (which was previously explained)



Applying the ‘regionprops’ function to L

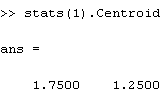


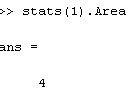
‘basic’ means that only the 3 basic properties, i.e. Area, Centroid and the Bounding Box , need to be calculated for the labeled objects.

Note that the ‘regionprops’ function returns a 4x1 structured array because there are 4 labeled objects in L.

The structure contains the Area, Centroid and Bounding Box measures of the individual labeled objects. We shall interest ourselves only with the Centroid and Area properties since these are relevant to the target tracking problem.

As an example, let us consider these properties for the first object, i.e. the object labeled ‘1’ in the matrix ‘L’.





stats(1).Centroid returns the (x, y) co-ordinates of the centroid of the first object.

stats(1).Area returns the area(or total pixels) of the first object.

The other objects can similarly be referred to by changing the index in stats().Area and stats().Centroid.

# Results



## Recognition of a Handwriting Sample



Figure ‑: Original Image



Figure ‑: Gray Scale Image

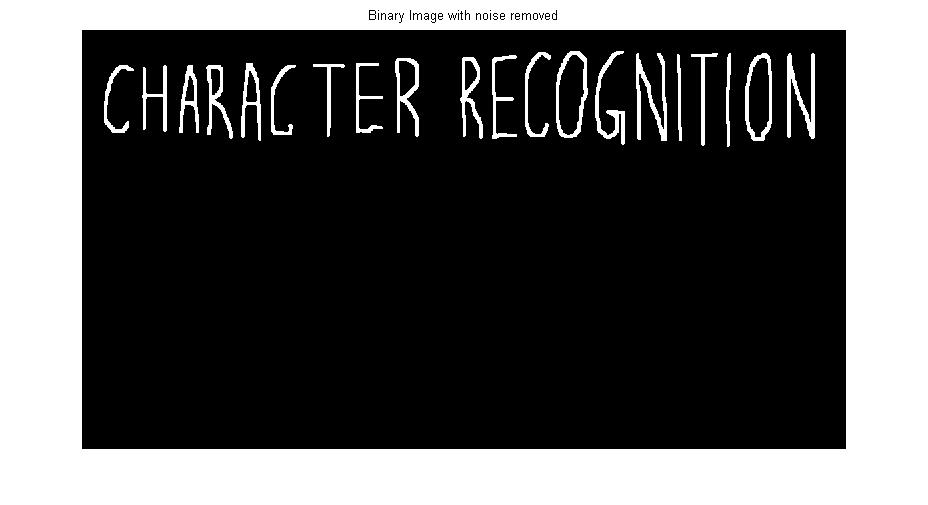


Figure ‑: Binary Image



Figure ‑: Binary Image with Labeled Objects

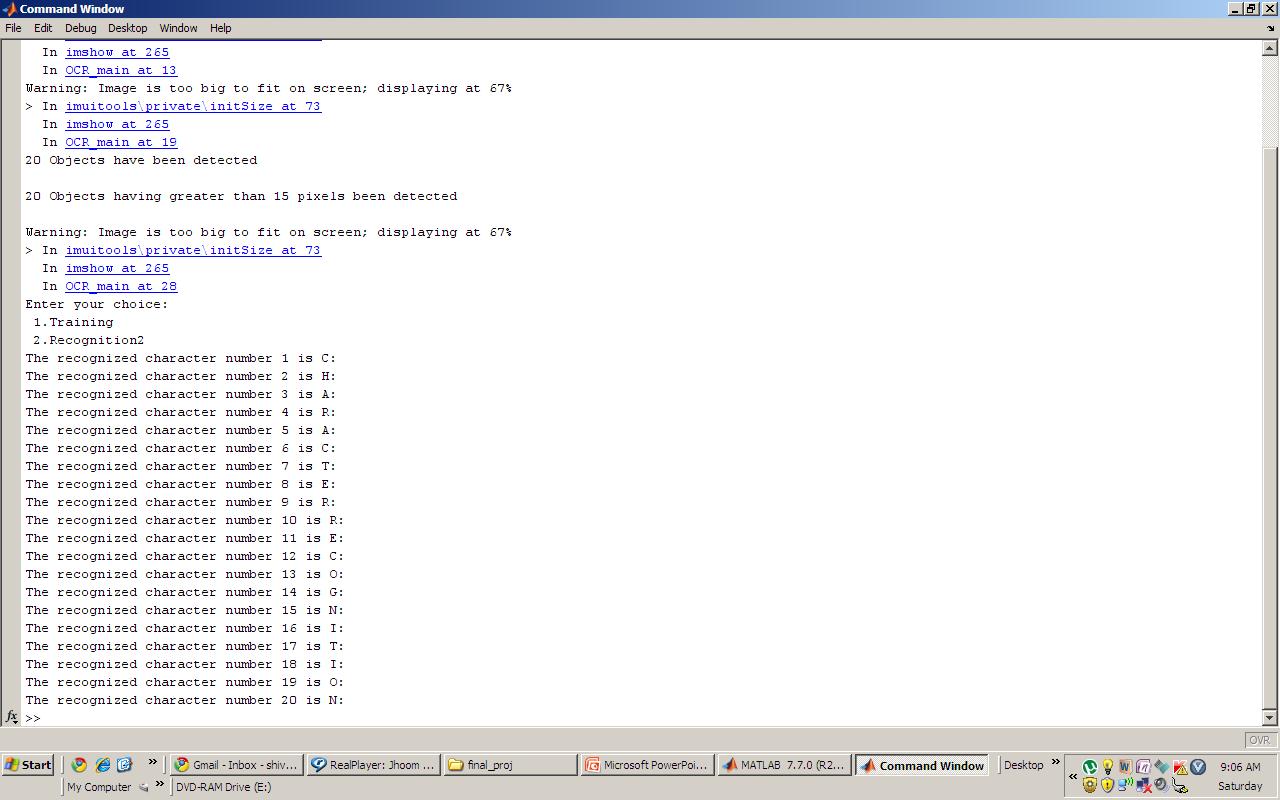


Figure ‑: Recognized Text

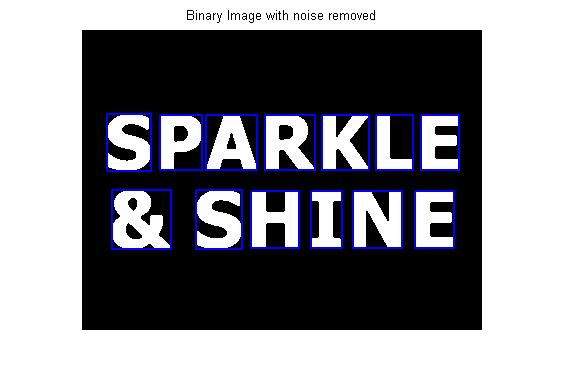
## Recognition of Machine Printed text



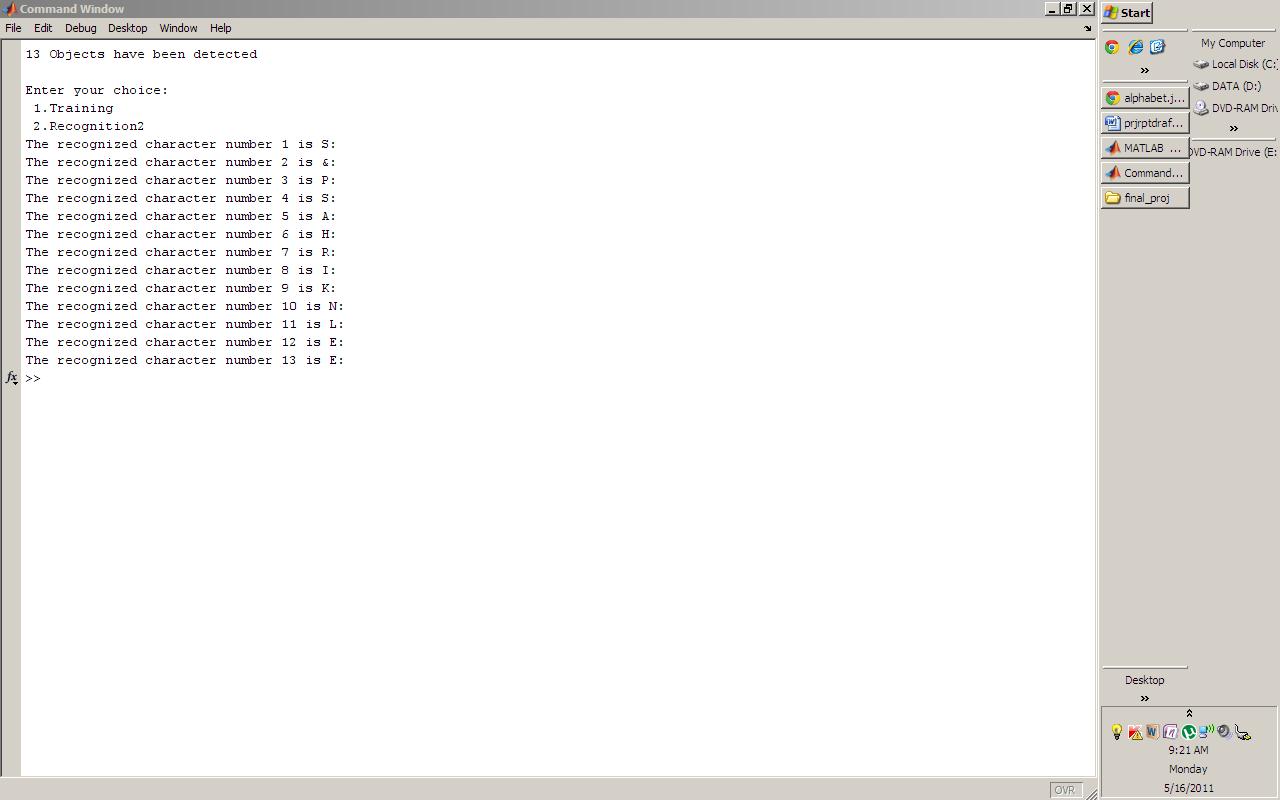
‑: Original Image



‑:Gray Scale Image



‑: Binary Image with Noise Removed



‑: Recognized Text

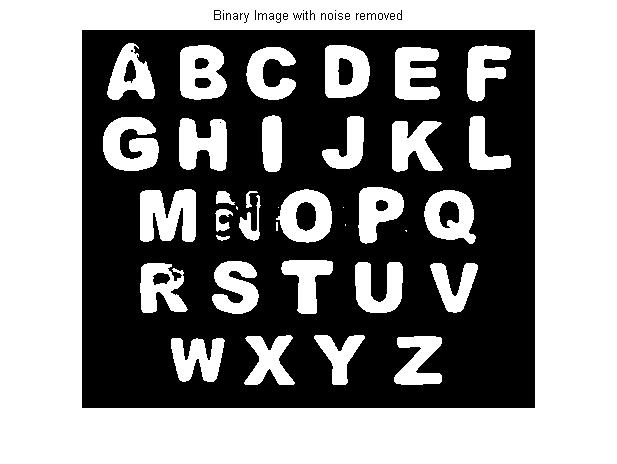
## Recognition of Document Text with Watermark Image



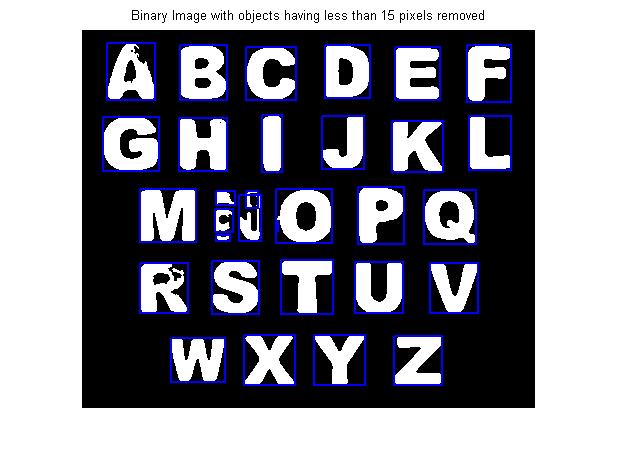
‑:Text with Watermark Image



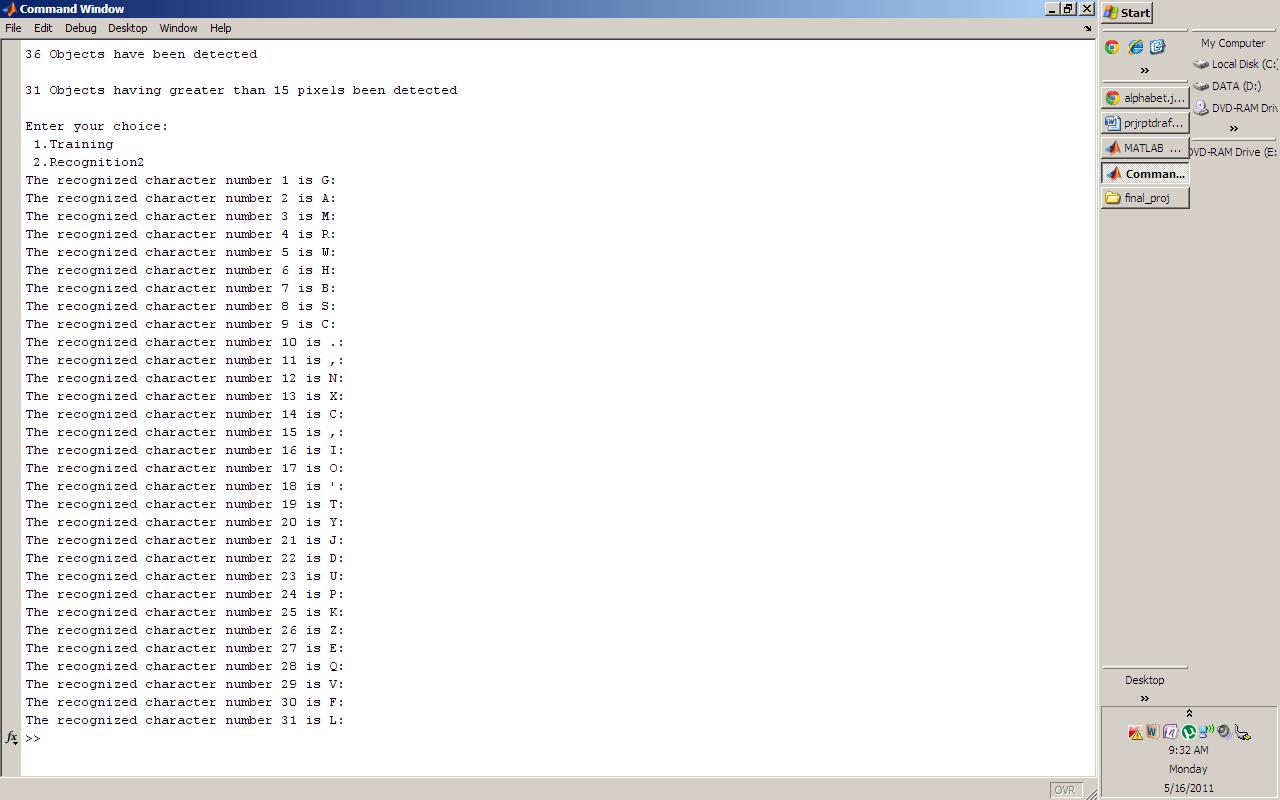
‑:Gray Scale Image



‑: Binary Image



‑: Labeled Objects



‑: Recognized Text

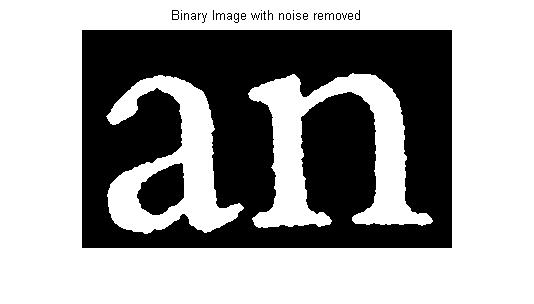
## Recognition Of Text in a Non-Document Image



‑:Text in a Non-Document Image



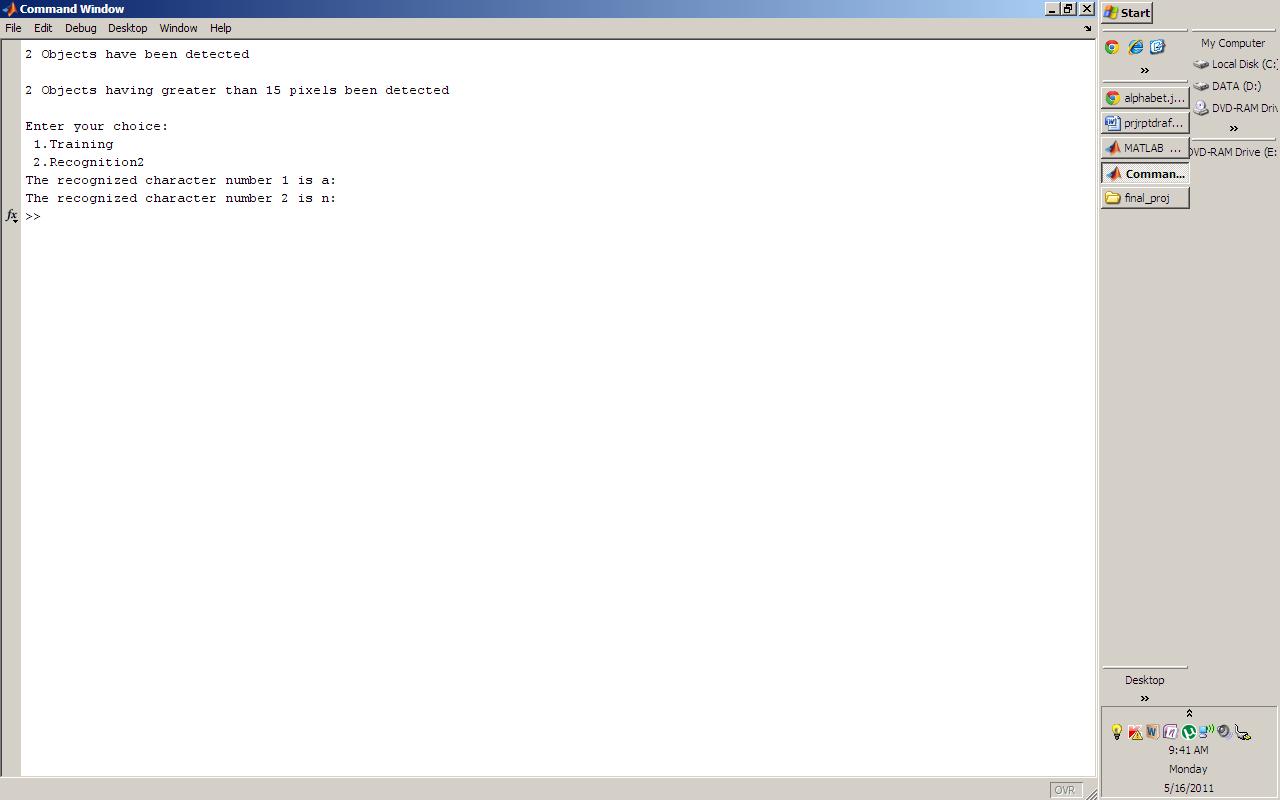
‑:Gray Scale Image



‑:Binary Image



‑:Extracted and Labeled Objects



‑:Recognized Text

# Conclusions and Future Work



## Conclusion

1. The proposed system is invariant to translation, rotation, scaling but is invariant to contrast changes.

2. bwlabel is used for extraction of objects from the image. It produces errors if more 2 objects have overlapping pixels or also if a single object has a missing pixel value.

3. The system can be trained for a number of values for a better performance.

## Proposed Improvements

1. Global thresholding using Otsu’s method has been used for segmentation. It works well for uniform backgrounds. However, for non-uniform background it does not produce the desired results. For this purpose, adaptive thresholding may be used, which takes into consideration individual pixel values.

2. Extended moment invariants may be used in place of Hu’s moments for better and accurate results.

3. The present system is not invariant to different font sizes. As a solution, extra features like Euler number etc can be used to identify a character.

## Future Work

The implemented system can be used in various applications.

### Banking

One widely known application is in banking, where OCR is used to process checks without human involvement. A check can be inserted into a machine, the writing on it is scanned instantly, and the correct amount of money is transferred.

### Legal

In the legal industry, there has also been a significant movement to digitize paper documents. In order to save space and eliminate the need to sift through boxes of paper files, documents are being scanned and entered into computer databases. OCR can be used to simplify the process by making documents text-searchable, so that they are easier to locate and work with once in the database.

### Application to Electronics Engineering

The implemented system can be used to read the numbers printed on the Integrated Circuits and also to read and interpret the data written on a printed circuit board..

# References



1. Ashish Agarwal, Rashish Tandon ,”Optical Character Recognition-Shape Invariant Approach and Implications of Spatial Resolution “

2. Glenn L. Cash, Mehdi Hatamian ,”Optical Character Recognition by Method of Moments “,Computer Vision,Graphics,and Image Processing 39

3. Gonzalez, Rafael C., Woods, Richard E. “Digital Image Processing”, Prentice Hall

4. Gonzalez, Rafael C., Woods, Richard E. ,Steven Eddins “Digital Image Processing Using Matlab”, Prentice Hall

5. Hriday Ravindranath :”Written Letter Analysis “

6. Jesse Hansen ,”Optical Character Recognition:”

7. Qing Chen, Emil M.Petriu :”Optical Character Recognition for Model-based Object Recognition Applictions “0-7803-81 08-4/03 0 2003 IEEE

8. Qivind Due Trier, Anil K Jain, Torfinn Taxt ,”Feature Extraction Methods for Character Recognition- A Survey” ,Pattern Recognition,Vol.29

9. S.Paschalakis, P.Lee :”Pattern Recognition in Grey Level Images using Moment Based Invariant Features” , Image Processing and its Applications, Conference Publication No. 465 0 IEE 1999

10. http://www.mathworks.com