summary

The aim of this project is to detect uppercase and lowercase English letters in an image based on computer vision and digital image processing techniques and recognize them using a neural network. An algorithm based on edge detection and morphological operations was used in Matlab to detect the letters in the image, and a neural network was designed and trained. In the end, we were able to implement the required and recognize the English letters with high accuracy.

Chapter 1

Project's introduction

In this chapter of the report, we present a general introduction, the objective of the project, its work stages, and the flow chart of the proposed algorithm.

General introduction

Detecting and identifying objects in an image is one of the most important challenges facing us today in the field of digital image processing. Among the objects that photo enthusiasts are interested in identifying are letters and numbers in an image, whether they are handwritten or computer-generated.

Character recognition is the mechanical or electronic process of translating an image into a computer-generated format. It began with scanners that read pages inbitmap format but to recognize them, additional software was required to do the job. This , software processed the scanned image to distinguish between the image and the text and identify the characters. However, with the development of the field of digital image processing, workers in this field began to propose several algorithms and publish research papers to perform this task. The matter later evolved and digital image processing was combined with neural networks that achieved higher efficiency than algorithms that depend on image processing only.

In this work, we present a method for detecting uppercase and lowercase English characters in images based on digital image processing and neural network techniques.

Project objective

This project aims to detect and recognize English characters based on digital image processing techniques and neural networks using MATLAB.

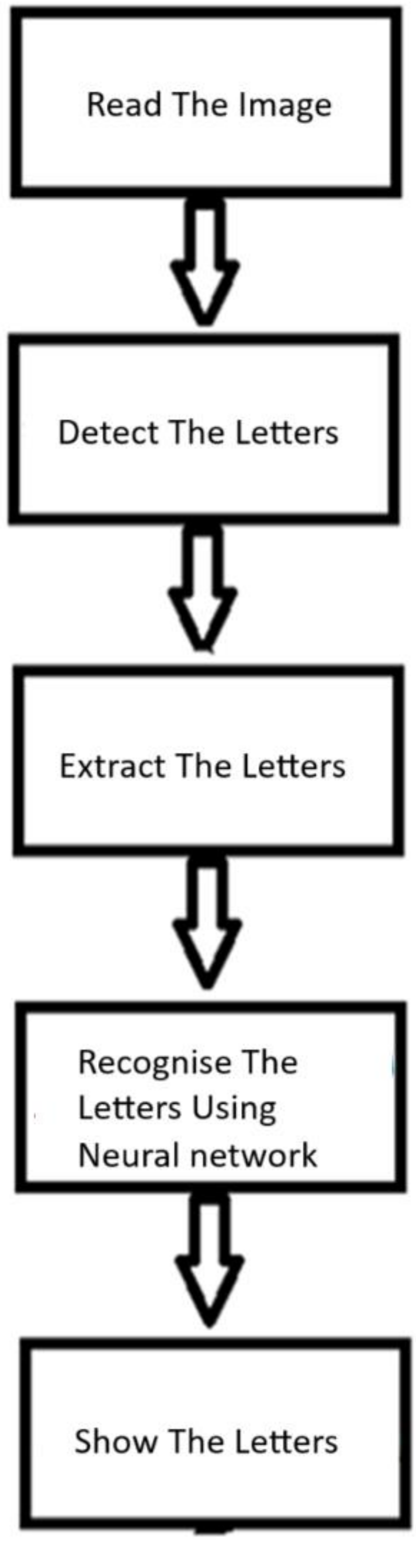
Project work stages

- Review the basics of image processing and MATLAB.
- Study of neural networks.
- Study and understand the work algorithm
- Apply the algorithm to MATLAB and verify its correct operation.
- Writing the report.

Flowchart of the algorithm followed

Figure (1.1) shows the flow chart of the algorithm followed in the project.

.Figure (1.1): Flowchart of the algorithm followed in the project



.Figure (1.1): Flowchart of the algorithm followed in the project

The algorithm is sequenced in five stages:

- The first stage: In this stage, the image in which the letters are to be recognized is read.

- The second stage: In this stage, the characters in the image are detected and cut out

- The third stage: The truncated characters are fed into a neural network for recognition.

- The fourth stage: reading the output of the neural network.

- The fifth and final stage: Showing the characters in the command window in the MATLAB program.

- But before that, a list of images of uppercase and lowercase English letters must be prepared in order to train the neural network.

Chapter 2

Image Processing Basics

In this chapter of the report we present basic ideas in image processing.

General Introduction

Humans are creatures that rely heavily on sight to sense and understand the world around them. Not only do humans look at objects around them to identify and .classify them, they can also examine and distinguish them with great precision ,They possess highly sophisticated visual skills that enable them to recognize faces distinguish colors, and process vast amounts of visual information at great speed.

The human need for the sense of sight in daily life has extended to all fields of science and technology, as the image is used in most modern scientific equipment as a means of conveying results to the operator. The success of space missions, for example, is measured by the quality of the captured images that are returned. This has necessitated the emergence of mechanisms for processing and developing :captured digital images, which are basically divided into two sections

- The first involves improving the information in the captured image to .make it easier for humans to interpret and understand
- The second section includes the transmission, storage and processing of .images to obtain results used in autonomous control of machines

The camera has become an integral part of our modern era, and almost every mobile device has a high-performance camera. The increasing use of images in many ,applications in various fields, which was accompanied by improvements in the size speed and low cost of digital computers, and with interesting developments in the related signal processing technology, has led to a significant development in the field of digital image processing. Image processing has found an important role in scientific, industrial, space and governmental applications such as digital ,communications, the Internet, medical imaging, multimedia systems, life sciences ,materials science, robotics, manufacturing, smart sensing systems, remote sensing .and the arts of drawing and printing

Historical Overview

Since ancient times, man has sought to preserve images of his life, starting with drawing in caves and then on walls (as in the ancient Egyptian civilization). Then came the Arab scientist Ibn al-Haytham, who devoted a large part of his life to conducting scientific research in light and optics. He noticed that light entering ,through a hole in one wall and falling on the other wall carried with it an inverted blurry image of a tree outside the room. Ibn al-Haytham described this incident in detail and recorded his observations, explaining the laws of light in this case and laying the foundations for what is called the camera obscura or pinhole camera. From this, the West derived the name camera from the Arabic word "Qamra", which .means the dark room

Scientists continued their efforts in this field until the French scientist Joseph Niepce was able to take the first real photograph in 1826, as shown in Figure (1.2). After that, research continued to develop photography devices and make them easier and clearer.

Figure (1.2): The first photograph taken.

After Niepce succeeded in taking the first photograph, the term photography began to appear. It is a Greek word composed of two parts, "photo" meaning light and graph" meaning drawing. The combined meaning of these words is "drawing by" light". Although cameras were not produced commercially in the sixty years following the taking of the first photograph, millions of photographs were taken during that period.

George Eastman first released the famous Kodak camera in 1888 with the slogan Press the button and we do the rest." The first pocket camera was released in the" American market in 1896. After that, the camera became widely spread and its accuracy and ease of use increased.

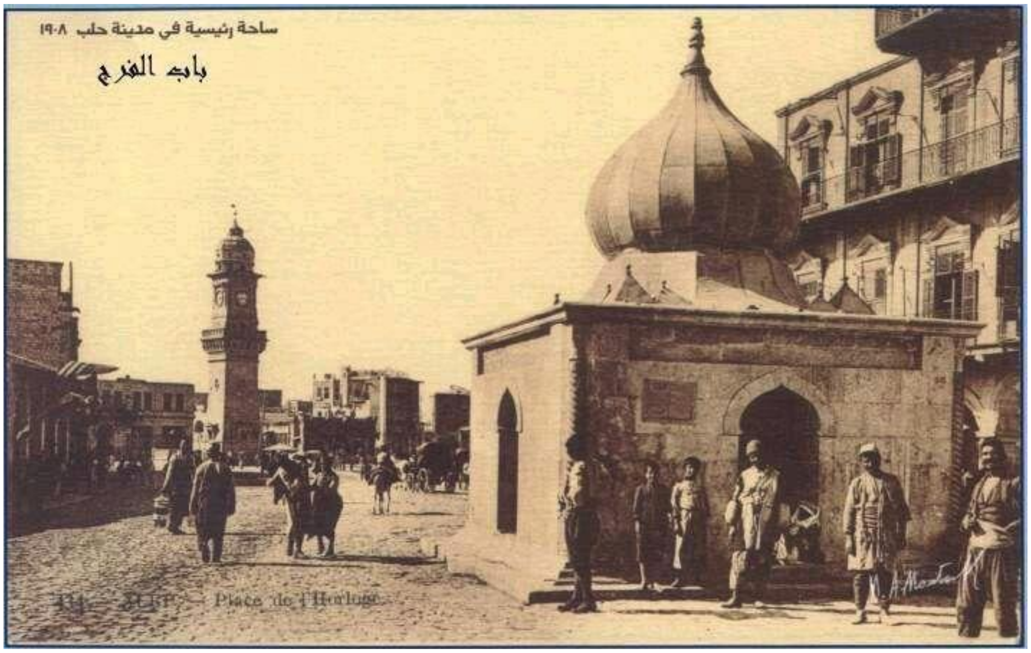


Figure (2.2): Bab al-Faraj Square (Aleppo) in 1908.

The history of the emergence of digital image processing is closely linked to the development of digital computers, especially those computer-supporting technologies such as increasing memory space, display devices, and transmission speed, which are among the most important requirements for digital image processing.

The first image to be converted into a digital image and transferred to a computer was produced in 1957 by Russell Kirsch, a pioneer in computers at the National Institute of Standards and Technology, who developed a device capable of transferring the image taken from the camera to the computer. Perhaps the first image he transferred was a picture of his newborn, and its dimensions were 176 x176 and it appears in Figure (3.2)

Figure (3.2): The first digital image taken.

The first computer capable of handling image processing tasks appeared in the early 1960s, and this was an announcement of the birth of what is known today as digital image processing. Since then, digital image processing has been continuously evolving, as it has been used in many medical, space, and other applications. This is to improve the contrast in images and encode gray levels and convert them into separate colors to facilitate the interpretation of images captured using X-ray techniques and other images used in industrial, medical, and biological science applications.

Capture the image

Most images are created by a combination of a light source and the reflection and absorption of the energy radiated from that source by the subject being imaged. This ,illumination may come from an electromagnetic energy source such as visible light radar, infrared, or X-rays, or it may come from a less conventional source such as ultrasound or even a computer-generated light pattern.

Image capture is defined as the process of obtaining an image from a source, such as a device such as a camera, a copier, or other, so that this image becomes ready for any future processing process. The image capture process is the first step in any image processing system. Since the image is a direct result of the devices, it can be considered as a primitive image that is not processed at all, which is very important in some fields in order to obtain a fixed base for work. The main goal of the capture process in image processing systems is to obtain a source of inputs that operates within fixed measurement and control principles to obtain images that can be processed and from which the information required by the system can be obtained.

The simplest model for obtaining images is the pinhole camera. This camera model reflects the mathematical relationship between the coordinates of a point in space and the reflection of this point on the image plane, as the light beam through the lens hole is considered perpendicular to the image plane. If there is an object located at a distancez and a heighty from the hole. The height of the reflection of this object is determined by two basic criteria: the depth of the camera boxf, ,which is constant in addition to the distance of the object from the center of coordinates. From the similarity of triangles, we find that:

$$x = -f\frac{X}{Z}, y = -f\frac{Y}{Z}$$

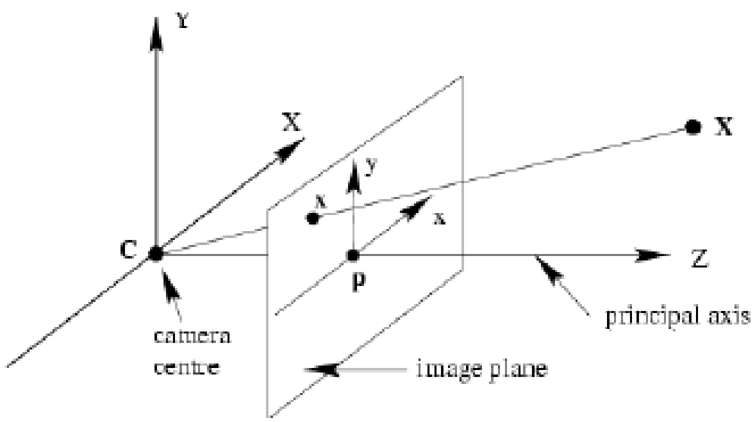


Figure (4.2): Pinhole camera model.

The resulting image scale changes withf- distance in a similar way to how the focal length determines the magnification of a commonly used camera. For example, in a still scene, small images with small focal lengths and a large angle of view are produced by wide-angle lenses with short focal lengths. Increasing the focal length results in larger images with smaller angles of view. Negative values in the previous relationships indicate that the resulting image is inverted with respect to the vertical and horizontal axes. That is, it is rotated by 180 degrees.

Convert image to digital image

The image acquisition stage produces an image that can be represented by a two-dimensional functionf(x,y) The value or amplitude of this function is a positive . magnitude whose physical meaning is determined by the source of this image at coordinates(x,y) When an image is generated by a physical process, the values of . this function are proportional to the energy emitted by the physical source.

Digital images rely on sensors that convert light into an electron charge, such as the CCD sensors used in most digital cameras, while some other digital cameras use CMOS technology to convert light into electrons. Millions of these sensors are combined in two-dimensional arrays to form the image.

Images are usually continuous in amplitude. To convert this image to digital form the function representing the image must be cut, producing a two-dimensional image MxN)M rows andN :columns) as in the following figure:

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1) & \dots & f(M,N-1) \end{bmatrix}$$

Both sides of the above equation are equivalent ways of expressing quantities in a digital image. The right side of the matrix contains real values and each element is called a picture element or pixel. The upper left corner is the origin of the coordinates in the image while the positive part of thex- axis extends downward and the positive part of they- axis extends to the right. Figure (5.2) shows an image with the coordinates pointing downward.

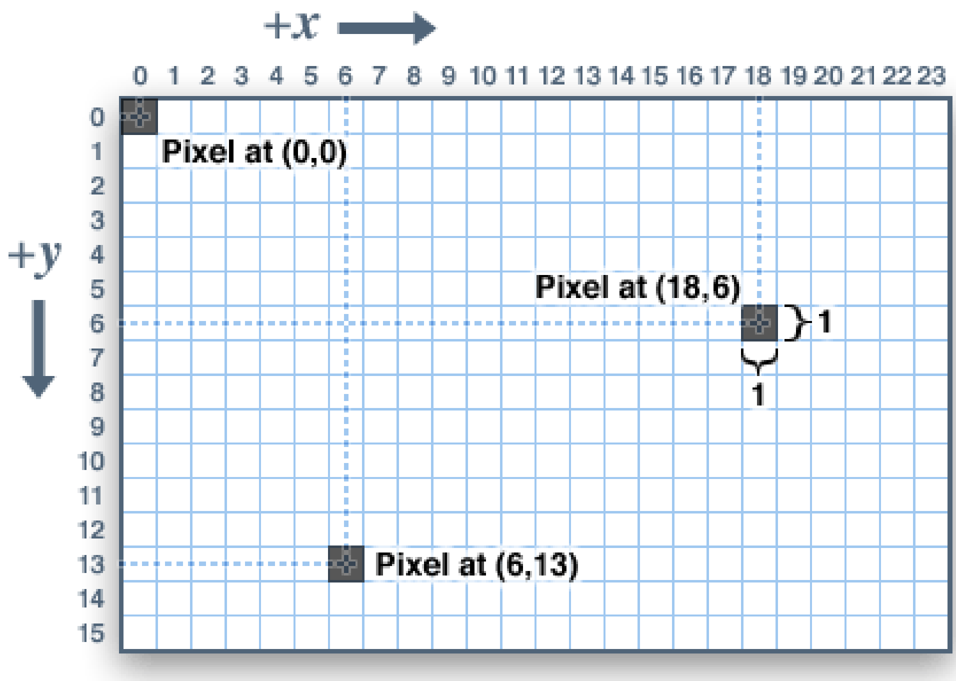


Figure (5.2): Orientation of axes in the digital image.

The process of digitizing an image on M and N coordinates is to segment it into L levels of light intensity . There are no restrictions on the M, N coordinates other than that they are positive integers. The number of light intensity levels is determined by segmentation and hardware considerations . The light intensity is usually specified by an integer power of 2. Thus, the pixel value can range from 0 to 2^k .

k the bit depth of the image. The number of bits per pixel depends on the is called type of image, whether it is grayscale, binary, or color. Below is a brief overview of some common types of images according to bit depth:

Grayscale images

It can be represented as a two-dimensional matrix where each pixel in the image is given a single value representing the light intensity or brightness, which is a positive For example, typical images use . 2^k integer value ranging from 0 to $k = 8$ bits or one byte per pixel, where the light intensity ranges from 0 to 255, where the value of zero represents the minimum brightness, i.e. the color black, while 255 represents the maximum brightness, i.e. the color white.

Allocating 8 bits per pixel is not sufficient for many specialized applications such as medical and aerospace applications. An image depth of 20, 14 or even 16 bits is often used.

Binary image

Binary images are a special type of grayscale image, but in these images the pixels are either black or white. Therefore, only one bit is allocated to each pixel in this ,type of image. Binary images are used to display fonts, graphics, saved documents fax and correspondence files, and in electronic printing.

Color images

Most color images are based on the primary colors, which are red, green, and blue (RGB) ,bits are often used to represent each color component in a pixel. That is 8 . .each pixel in a color image requires 24 bits to represent the three color components As with grayscale images, images can be represented with 30, 36, or even 42 bits per pixel in certain applications.

Although most color images are represented by three color components, there are some types of images that are represented by four or more color components, which is common, especially in color printing, which is based primarily on theCMYK (magenta-yellow-black) color system.

Indexed images are a special type of color images and differ from real images in the number of colors in the image, which is less in indexed images. The pixel value in this image (which takes up 8 bits maximum) refers to a specific table containing all the values of the selected colors.

Figure (6.2) shows a grayscale image, Figure (7.2) shows a binary image, and Figure shows a color image of the same image (8.2).

Figure (6.2): Gray image.

Figure (7.2): Binary image.

Figure (8.2): Color image.

Image Resolution

Since most images are rectangular, their size is determined directly from the width of the matrixM (number of columns) and its lengthN .(number of rows)

While image resolution is determined by the number of pixels in a unit of ,measurement, which is either dots per inch or lines per inch, in satellite images pixel resolution is determined by the number of pixels per kilometer. This means that the size of the image does not provide accurate information about spatial resolution, and to determine this resolution, the spatial units in the image must be taken into account.

Image size is used to compare the capabilities of cameras. For example, a 20 megapixel digital camera is expected to be able to show more detail in an image than an 8 megapixel camera. This is assuming that both cameras take the same image from the same distance.

Image file formats

There are some criteria that must be taken into consideration when choosing a suitable format for the image, which are:

- **Image type:**

.Includes binary, grayscale and color images

- **Storage space and image compression:**

It includes the importance of the image storage requirements in the application in ,which the image is to be used, and the efficiency of image compression methods especially when the resulting compression formats are taken into account.

- **Compatibility**:

It reflects the importance of exchanging, saving and archiving image data and the ease of reading the data by the computer.

- **Application area:**

The main scope in which the image data will be used includes whether it is intended for printing, publishing over the network, or as computer graphics, and whether .these images are medical, astronomical, or other
The most popular image file formats used are:

Tagged Image File (TIFF)

This flexible and widely used format was developed specifically to meet the ,professional needs of a wide range of industries. It is popular with graphic designers publishers, and printers, as well as amateur and professional photographers. It was designed byAldus and then developed byMicrosoft and laterAdobe It supports a . wide range of grayscale, indexed, and color images. It also supports special image types that have high storage capacity integers and floating-point images. These images can be uncompressed or compressed with information preserved without significant loss.
Files represented in this format take up much more storage space than their counterparts represented in other formats, as they take up memory space equal to the size of the image multiplied by the bit depth of the image, and thus the size of a singleTIFF file may be .more than 100 MB, making it unsuitable for emailing

Graphics Interchange Format (GIF)

This format was developed by CompuServein and has since become the most 1986 widely used format on the World Wide Web. Its popularity is due to its early support for indexed images with multiple bit depths, interlaced image loading, and its ability to encode simple animations by storing a number of images in a single file and .displaying them sequentially

These images are designed to handle color and grayscale images with a maximum depth of 8 bits. They do not support true color images and provide effective support for images containing 2 to 255 colors, allowing pixels to be represented with fewer bits.

Portable Network Graphics (PNG)

This format was developed as an alternative to theGIF format and was designed as .a general format for images used on the InternetPNG format supports three types of images:

- True color images: with a pixel depth of up to3x16 .
- .Grayscale images: up to 16 bits/pixel depth
- .Indexed images: up to 256 different colors

PNG surpasses GIF in everything except its ability to generate animated images. A PNG file can only containa single .image

Joint Photographic Experts Group format -JPEG

This is the most common file format for digital images and other digital graphics. When a file is saved inJPEG format it uses a lossy compression algorithm. That , is, the image quality will decrease as the file size decreases. The format is named after the Joint Photographic Experts Group, which created this file type to achieve acceptable information loss when compressing the image at a ratio of16:1 .

This format is the most common image format used in digital cameras. UnlikeGIF files ,which have poor image quality ,JPG allows the image file size to be reduced without significantly compromising on quality.

Windows Bitmap Images (BMP)

BMP ,files represent the way the Windows environment handles bitmap image files so they are not well known in other environments such asMac for example. This ,format supports grayscale, color and indexed images. It also supports binary images

,but in an inefficient way, as each pixel in the binary image reserves an entire byte not a single bit.

When you save an image inBMP format the color values of each pixel in the image , are stored without any compression. For example, if we have aBMP image with dimensions of10x10 .its size will be the bit depth of the pixel multiplied by 100 , This method results in saving clear, high-quality image data, but its size is large.

Histogram

A histogram is a graph that shows the distribution of gray levels in digital images. A histogram displays the number of pixels representing each color level between ,black and white. Black is at the far left and white is at the far right. In dark images .the histogram appears to be biased to the left and in bright images, to the right Figure (8.2) shows the histogram of an image.

The image histogram does not contain any information about the location of the color levels in the image, i.e. it does not contain any information about the spatial coordinates of the pixels in the image. This is because the histogram is a statistical function. That is, there is a distinct histogram for each image, but there is no distinct image for the histogram. Figure (9.2) shows a different image with the same histogram.

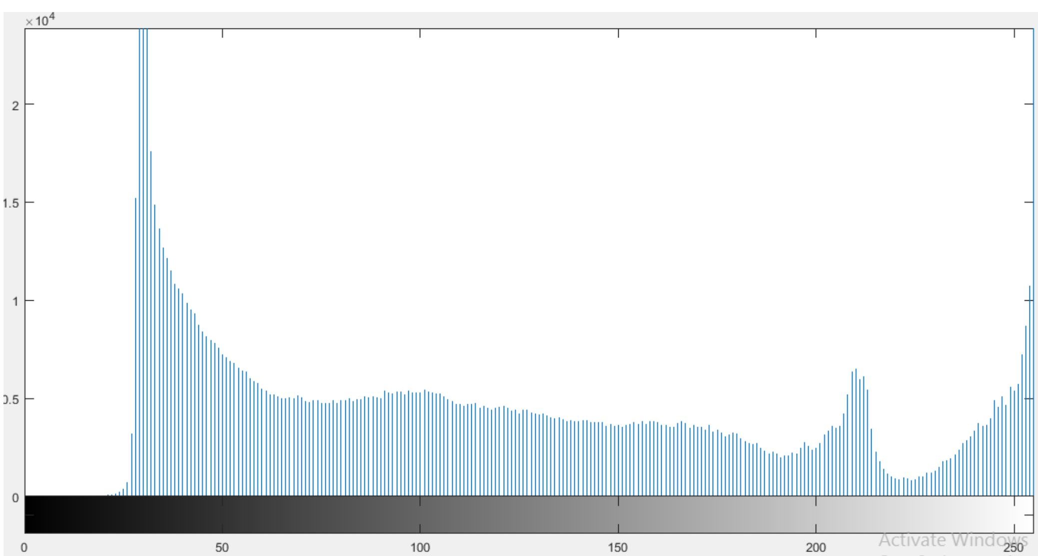


Figure (8.2): The histogram of the image in Figure (6.2).

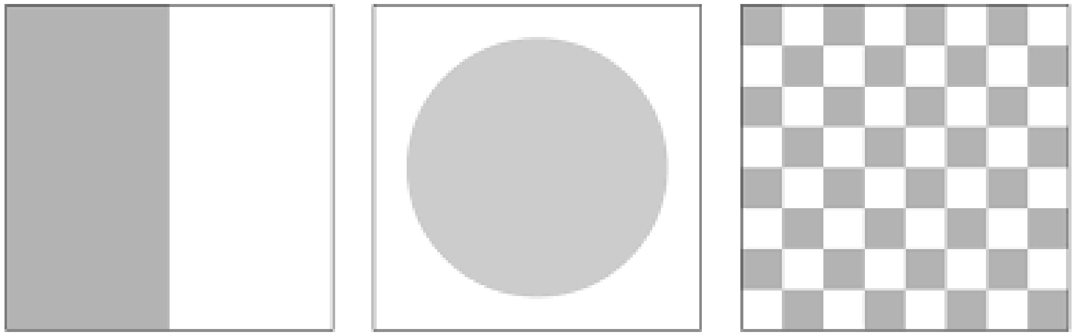


Figure (9.2): Three images with the same histogram.

Correction

The binarization process is a special case of gray-scale reduction, and is commonly used to obtain desired objects in an image based on their color. It produces a binary image by separating pixel values into two groups based on whether they are greater As follows a_{thresh} .or less than a certain threshold value:

$$B = \begin{Bmatrix} a_0 : if\,(\mathrm{a} >= a_{thresh}) \\ a1 : if\,(a <= a_{thresh}) \end{Bmatrix}$$

The threshold value is between the minimum and maximum gray levels in the image. If the image is to be converted to binary, thena0 = 1 anda1 = 1. Stitching is usually used to isolate or separate certain objects in an image based on pixel values. Below we show the results of using different threshold values.

Figure (10.2): The image resulting from pixelating the image in Figure (6.2) with a threshold equal to 0.5 (128/256).

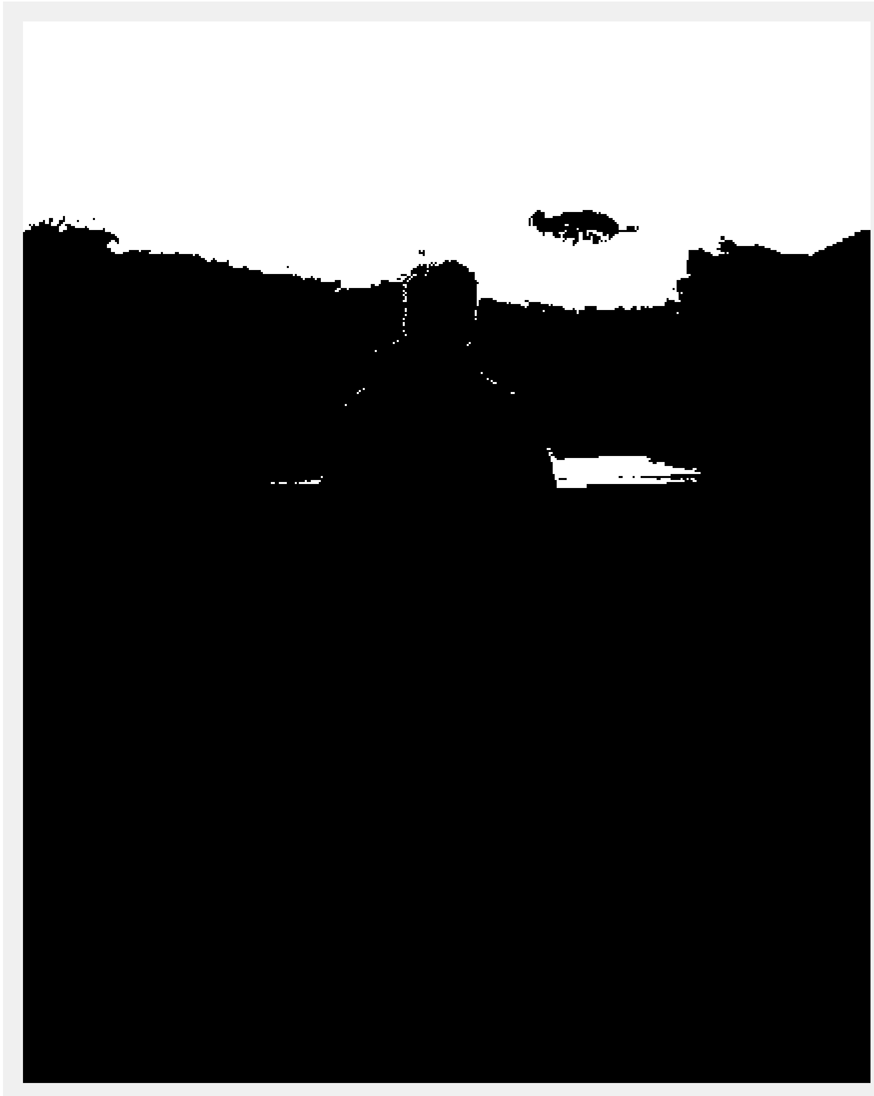


Figure (11.2): The image resulting from pixelating the image in Figure (6.2) with a threshold equal to 0.8.

Filters in the image

Filters are used in an image to remove noise or to detect edges. The filter is applied to the image through the filter window. The neighbors of a pixel are the pixels surrounding it, which can be expressed as the smallest matrix in which the pixel is located at its center. Most image processing algorithms that rely on neighbors use small square matrices with an odd number of pixels, such as a3x3 matrix as shown , in Figure (12.2). The term neighbors of a pixel can have different meanings. It is common to refer to the group of pixels above, below, to the right, and to the left of a reference pixel as the four neighbors of a pixel (Figure (13.2)). Meanwhile, all the pixels surrounding a reference pixel on all sides are referred to as the eight neighbors of a pixel, as shown in Figure (14.2).

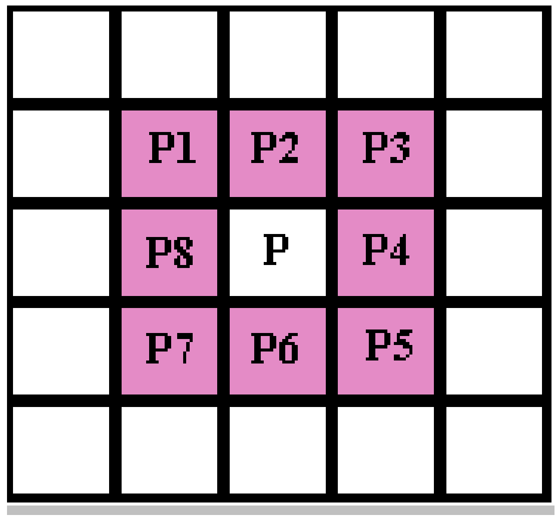


Figure (12.2): Neighbors of pixels in the image.

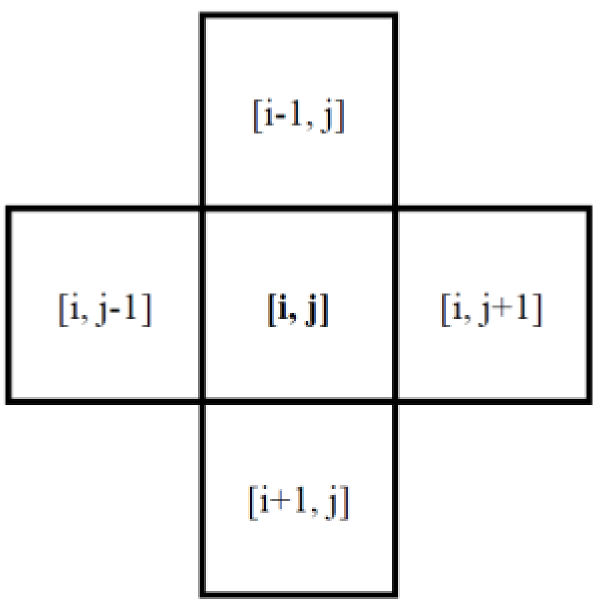


Figure (13.2): The four neighbors of a pixel.

[i-1, j-1]	[i-1, j]	[i-1, j+1]
[i, j-1]	**[i, j]**	[i, j+1]
[i+1, j-1]	[i+1, j]	[i+1, j+1]

Figure (14.2) The eight neighbors of a pixel.

Filtering operations are applied to the entire image by a series of local operations on the neighbors of the pixels using the sliding window principle so that each pixel .in the image is processed based on an operation that uses the neighbors of that pixel There are many options available for applying the filter to the image, which can be generally classified into linear filters and nonlinear filters. The new value of the pixel is calculated using a mathematical operation that is either linear or nonlinear.

A linear filter is defined as one in which the new values of a pixel are given by linear summation of the values of the pixel with its neighbors. Any filter that does not fit this definition is a non-linear filter. The linear summation of the neighbors of the pixel is determined by the filter kernel, or mask, which is a matrix of size exactly equal to the size of the pixel neighbor matrix containing weights assigned to each corresponding pixel of the target pixel's neighbors.

This process is carried out according to the following mechanism:

1- The filter kernel is slid over the original image$I$ such that the center of the kernel $H(0,0)$ coincides with the desired pixel with coordinates(u,v) at which .the filter value is to be found
2- $H(I,j)$ are multiplied by the corresponding pixels$I(u+i,v+j)$ and then all the .resulting values are summed
3- The resulting sum value is stored at the same coordinates of the desired pixel but in a new image$I'(u,v)$.

4- .The previous steps are repeated for each pixel in the input image

Examples of linear filters includethe Mean Filter and theGaussian Filter, and examples of nonlinear filters includethe Mean and Max Filters .

Detect edges in the image

Edge detection is a fundamental process in image processing that aims to find as many edges as possible in an image. These edges are then connected together to form meaningful boundaries or lines in the image, which leads to segmenting the .image into multiple regions

Edges are defined as the boundaries between two regions of different image properties such as gray levels, which are usually associated with a sharp difference in light intensity between parts of the image. Figure (15.2) illustrates the difference between ideal edges and oblique edges. Ideal edges have an abrupt change in light .intensity, while oblique edges have a gradual, slanting change in light intensity

Figure (15.2): Sloping edges (right) and perfect edges (left).

Applying a median filter usually results in smoothing the image and losing some detail. Therefore, another type of filter is used to bring out the edges and is called .differential filters

Edge detection is a process of detecting local changes in the intensity value of an image. The change in intensity level is measured by the gradient of the image. Since :an image is a function of two variables, the gradient of the image is given as follows

$$\begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{df}{dx} \\ \dfrac{df}{dy} \end{bmatrix}$$

:The direction of the gradient is given by the following relation

$$\theta(x, y) = \tan^{-1}(G_y / G_x)$$

The gradient calculates the change in light intensity level as well as the direction of this change. This change is calculated by calculating the difference in light intensity level between neighboring pixels. The gradient in a binary image on the X andY coordinates is approximated :by the following two relations

$$G_x = f(i+1, j) - f(i, j)$$

$$G_y = f(i, j+1) - f(i, j)$$

Calculating the gradient requires two masks, the first according to theX axis and the second according to the Y axis . The previous two gradients are combined into a vector whose amplitude expresses the intensity of the edge gradient. Figure .shows the image in which the edges are to be displayed (16.2)

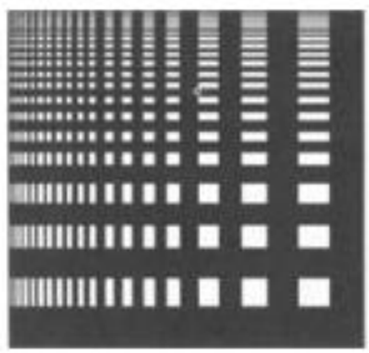


Figure (16.2): The image whose edges are to be shown.

Figure (17.2) shows the edges along theX- axis .

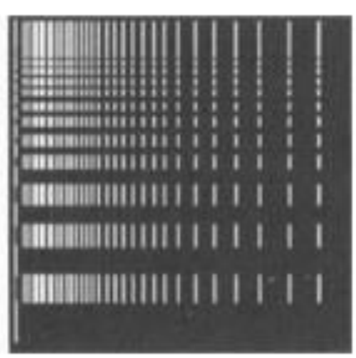


Figure (17.2): Edges according to theX- axis .

Figure (18.2) shows the edges along theY- axis .

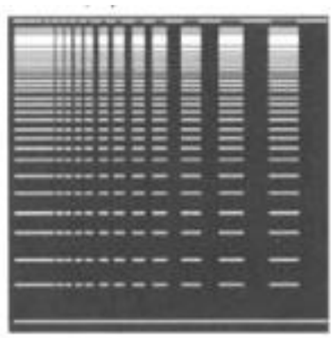


Figure (18.2): Edges according to theY axis .

Figure (19.2) shows the edges in the image after collecting the edges along theX-axis and the edges along theY- axis .

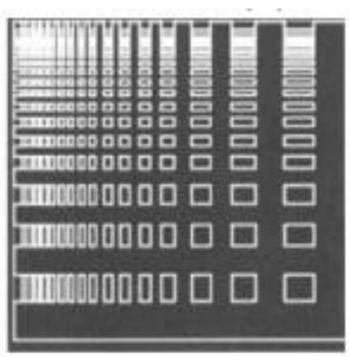


Figure (19.2): Edges in the image.

Roberts Filter

The Roberts filter is the simplest edge detection filter that relies on computing the gradient of the gray levels at a given point. The first derivative equation in the function is approximated to the previous two equations.

The Roberts filter is represented by two different masks, one representing its application on the horizontal axis and the other on the vertical axis.

$$H_x = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$H_y = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Sobel Filter

This filter uses two3x3 masks, one along theX axis and the other along theY axis These two masks are given as follows :

$$H_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & 1 \end{bmatrix}$$

$$H_y = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Figure (20.2) shows the image in which edges are to be detected. Figure (21.2) shows edge detection using a Sobel filter for a threshold value of 110.

Figure (20.2): The image in which edges are to be detected.

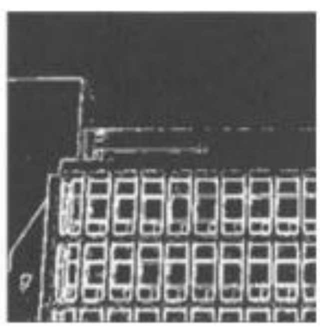


Figure (21.2): Edge detection after staggering with a value of 110.

Bruit Filter

A Bruett filter is known because it is an eight-mask filter. Each mask differs from .the previous one by a 90-degree rotation. Figure (22.2) shows four Bruett masks The remaining masks are obtained by successive 90-degree rotations.

1	1	1
0	0	0
−1	−1	−1

0	1	1
−1	0	1
−1	−1	0

−1	0	1
−1	0	1
−1	0	1

−1	−1	0
−1	0	1
0	1	1

Figure (22.2): Four Pruitt masks.

Figure (23.2) shows the result of edge detection using the Bruett filter.

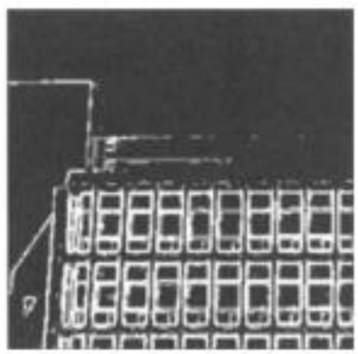


Figure (23.2): Edge detection using the Bruett filter.

Morphological processes

Morphological operations are a branch of image processing that provides the tools needed to represent, describe, and analyze shapes in an image. Because these operations are concerned with the geometric structure of image components, they are named after a branch of biology that deals with the study of the shapes and structures of animals and plants.

Mathematical morphology provides a set of useful tools for identifying and .extracting meaningful attributes based on the properties of shapes in an image Applications range from object counting to object searching. Morphology is a powerful and important set of rules and principles that are applied mathematically within the framework of set theory. It is also used in the pre- and post-processing of images containing significant shapes.

Structure Element

Similar to linear filter kernels, the properties of morphological operations are .determined by the values of the elements of a matrix called the construction matrix When dealing with binary images, the values of the construction matrix are limited to one of two values: 0 and 1.

Figure (24.2) shows examples of building elements.

1	1	1
1	1	1
1	1	1

	1	
1	1	1
	1	

		1	1	1		
	1	1	1	1	1	
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
	1	1	1	1	1	
		1	1	1		

1	1	
1	0	
1		0

1	1	1
1	10	1
1	1	1

Figure (24.2): Examples of morphological building blocks.

Expansion Dilation

It is a morphological process that results in the growth of organisms and is It can be defined as a group process as follows symbolized by the symbol $\oplus$:

$$I \oplus H \equiv \{(p+q) \mid p \varepsilon I, q \varepsilon H\}$$

the set of points resulting from the expansion process is the sum of all possible , pairs of coordinates of the points in the setsI andH. The stretching process stretches the white areas in the image. Figure (25.2) shows the result of the stretching process for some shapes.

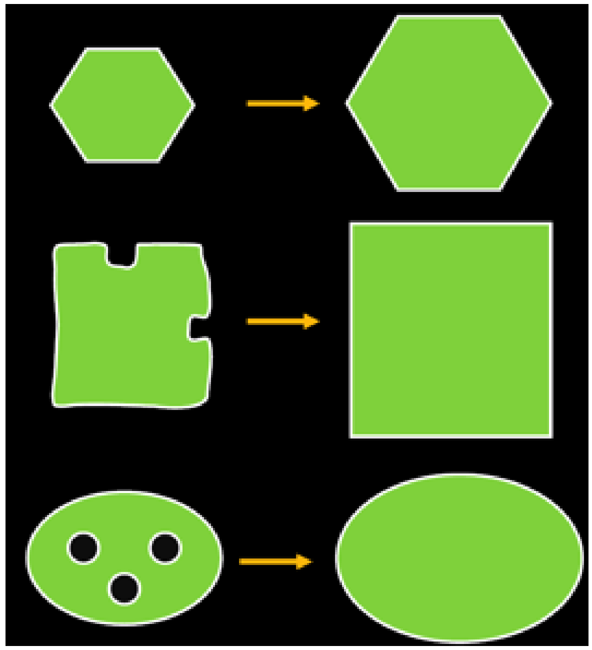


Figure (25.2): The result of the expansion process for some shapes.

Erosion Erosion

The process of erosion can be considered the opposite of the process of expansion and is known in groups as follows:

$$I \odot H \equiv \{p \varepsilon Z^2 \mid (p+q) \varepsilon I,\ for\_every\_q \varepsilon H\}$$

That is, the coordinate pairp is present in the result of the erosion operation if and only if the structure elementH located at pointp is completely contained in the foreground pixels of the original image. Figure (26.2) shows the result of the erosion operation for some shapes.

Erasing is a process that reduces shapes in an image and removes smaller areas of a building element.

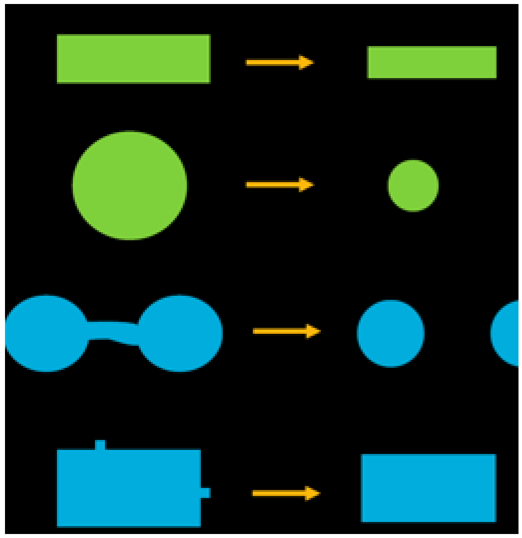


Figure (26.2): The result of the erosion process for some shapes.

Conquest open

Opening is a complex process. Opening begins with an image erosion process followed by a stretching process. Figure (27.2) shows the result of the opening process for area Ausing structural shapeB.

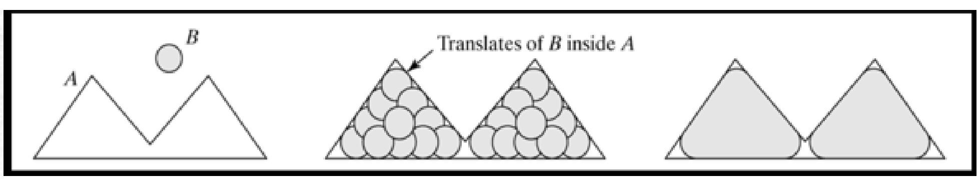


Figure (27.2): The result of the opening process for areaA using structural shape B.

Closure close

Closure is a complex process resulting from an expansion process followed by an erosion process. Figure (28.2) shows the result of the closure process for area A using structural formB.

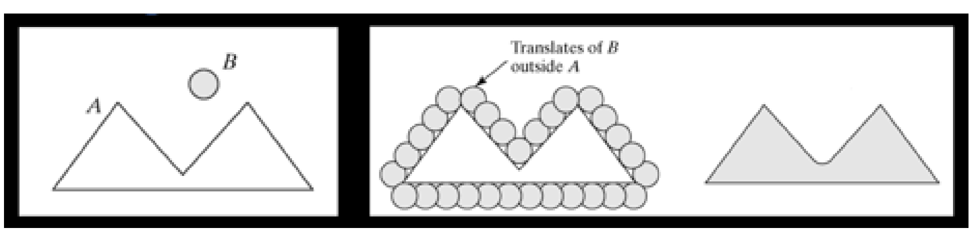


Figure (28.2): The result of the closure process for region Ausing structural shape B.

Chapter 3

Neural networks

In this chapter of the report, we present an explanation of neural networks in
.general

Artificial Neural Networks

Artificial neural networks are computational techniques designed to simulate the way the human brain performs a particular task, by processing Massive distributed in parallel, and composed of simple processing units . These units are nothing but) computational elements called neurons or nodesNeural Networks which have a (neural property in that they store scientific knowledge and experimental information to make it available to the user by adjusting weights. Figure (1.3) shows a biological neuron.

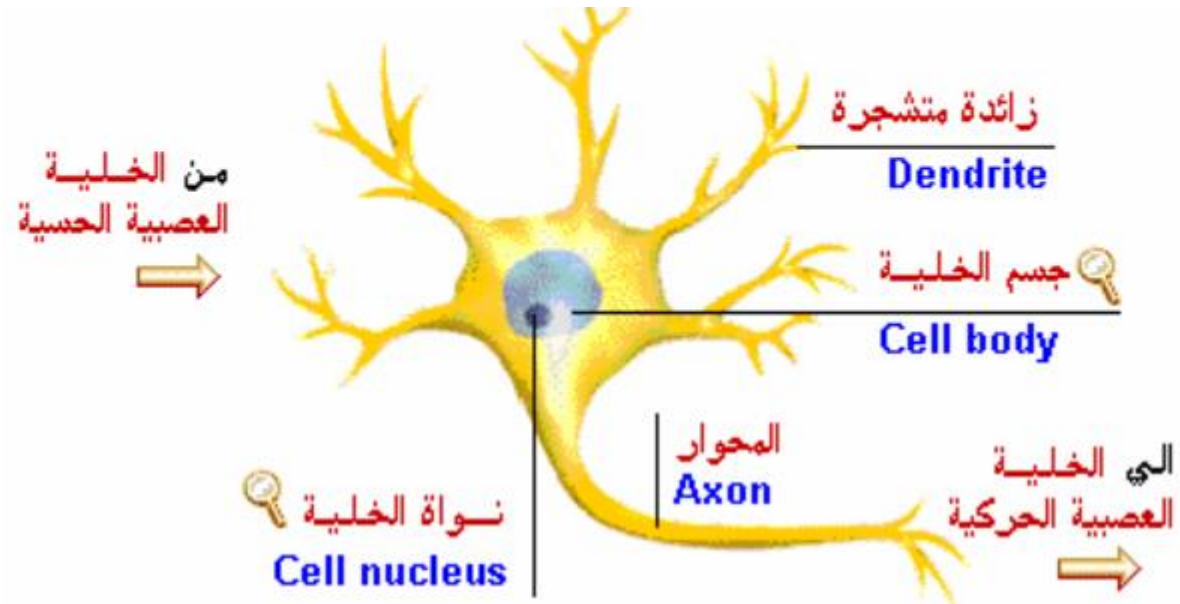


Figure (1.3): Real neuron.

is similar to the human brain in that it acquires knowledge So the artificial neuron through training and stores this knowledge using connections within neurons called synaptic weights. There is also a biological neural similarity that gives scientists the opportunity to Biology relies onANN to understand the evolution of biological phenomena.

Just as humans have input units that connect them to the outside world, which are their five senses, neural networks also need input units and processing units in which mathematical operations are performed to adjust the weights and through which we obtain the appropriate response for each input to the network. The input units form a layer called the input layer, and the processing units form the processing layer, which outputs the network's outputs. Between each of these layers there is a layer of Interconnections that connect each layer to the next layer and in which the weights of each interconnection are set, The network has only one layer of input units, but it may have more than one layer of input units. Processing . Figure (2.3) shows the general form of a neural network.

The variablesa ,represent the network inputsw ,the weightsx ,the network output andJ .the processing element.

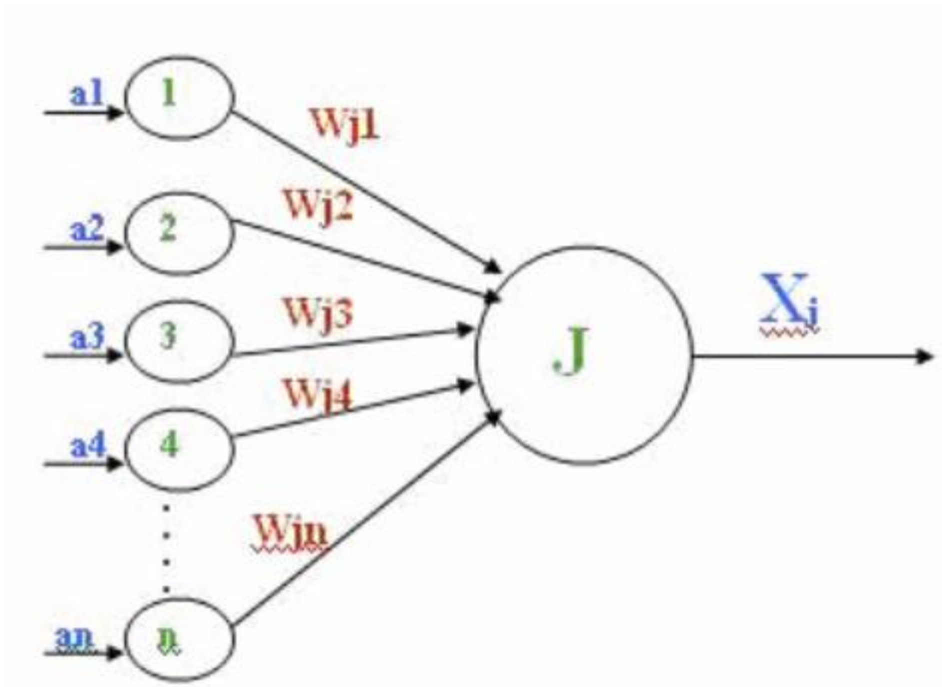


Figure (2.3): General shape of the neural network.

:The processing element is divided into two parts

1- .Collector: which collects signals into the weighted input
2- ActivationFunction This function determines the output of the neuron by : scaling it to the range[-1,1] .

Transformation functions

The transformation function determines the output of a neuron and must have the following properties:

It must be a continuous differentiable function -1.

It should be fluid and not decreasing -2.

There are three types of transformation functions:

Follow the threshold or continue the step

This function limits the output of the neuron to one when the input is greater than zero and to -1 when the output is less than zero. Figure (3.3) shows the graph of the step function.

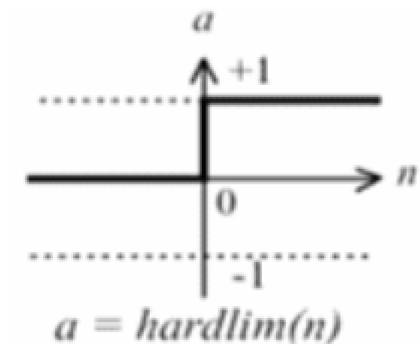


Figure (3.3): The graph of the step function.

Stepwise function

Figure (4.3) shows the graph of the step function. This function is used in linear adaptive filters.

Exponential function

This function takes input values between negative and positive infinity and makes the output limited between 0 and 1. It is the most widely used function because of its ease of use and derivation. Figure (4.3) shows the graph of the exponential function.

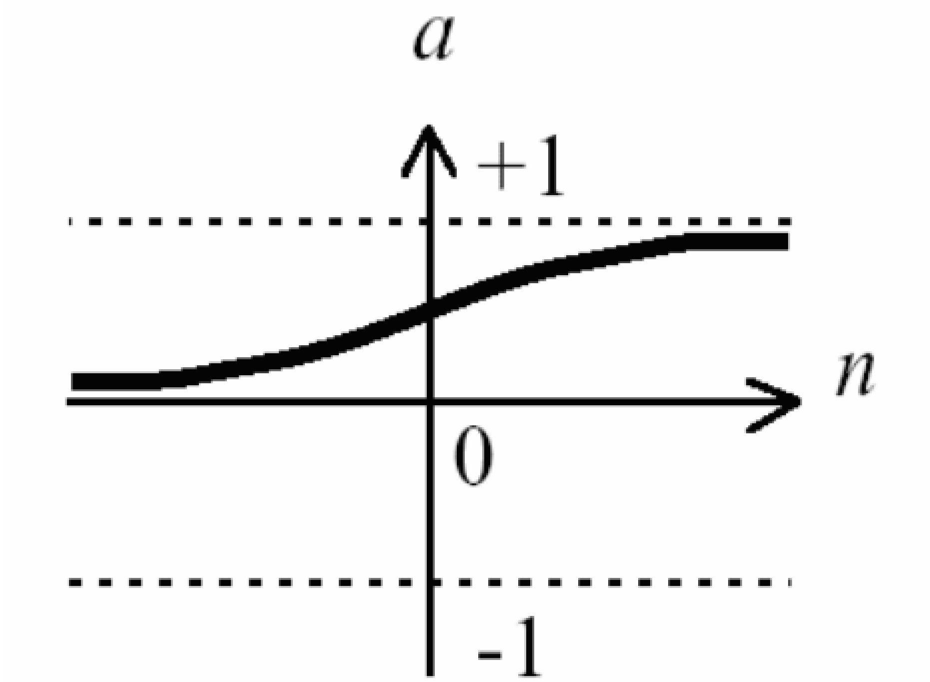


Figure (4.3): The graph of the step function.

The connection structure of the neural network

The structure of a neural network is the way in which neurons are connected to each other to form the network and this is related to the training algorithm. Figure (5.3) shows a neural network with one hidden layer.

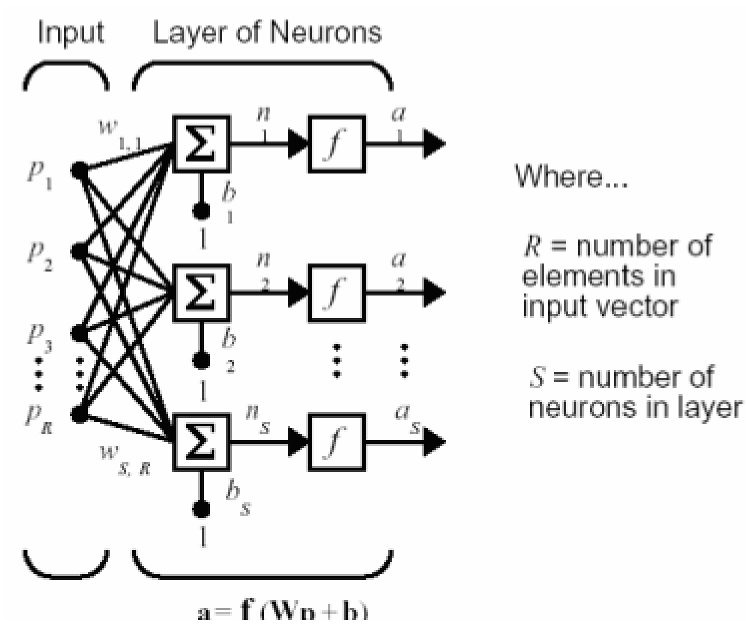


Figure (5.3): A neural network with one inner layer.

Each neuron has a summer connection that sums the weighted input with the displacement to form the numerical output of the neuron, and in The result is that the output components of the neuron layer form the output vector , which is a single-column matrix given by the following relation:

$$\mathbf{a} = \mathbf{f}\,(\mathbf{W}\mathbf{p} + \mathbf{b})$$

The input vector components are multiplied by a weight matrix defined by the following relation:

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ & & & \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

The row indices of the elements of this matrix indicate the target neuron, while the column indices indicate the input components . Source. That is, the indicators in elementW1,2 It indicates that this weight is related to the first neuron, and that the input components of this neuron are the second components.

can consist of several layers, in which case each layer has a weight matrix W, a displacement ,vector b, ,and an output vectora. For distinction, the layer number is added as a superscript for each of the variables used by the constructed network : Input vehicle , neuron in the first layer, neuron in the second layer, and so on in the same manner. It is also noted that the output of each intermediate layer is the input to the layer that follows it, and thus each layer in This network is like a single-layer network. The layer that gives the output is called the output layer, while the input is not considered a layer, and the rest of the layers are called hidden layers. We can draw the three-layer network shown in the previous figure using .the abbreviated drawing shown in Figure (6.3)

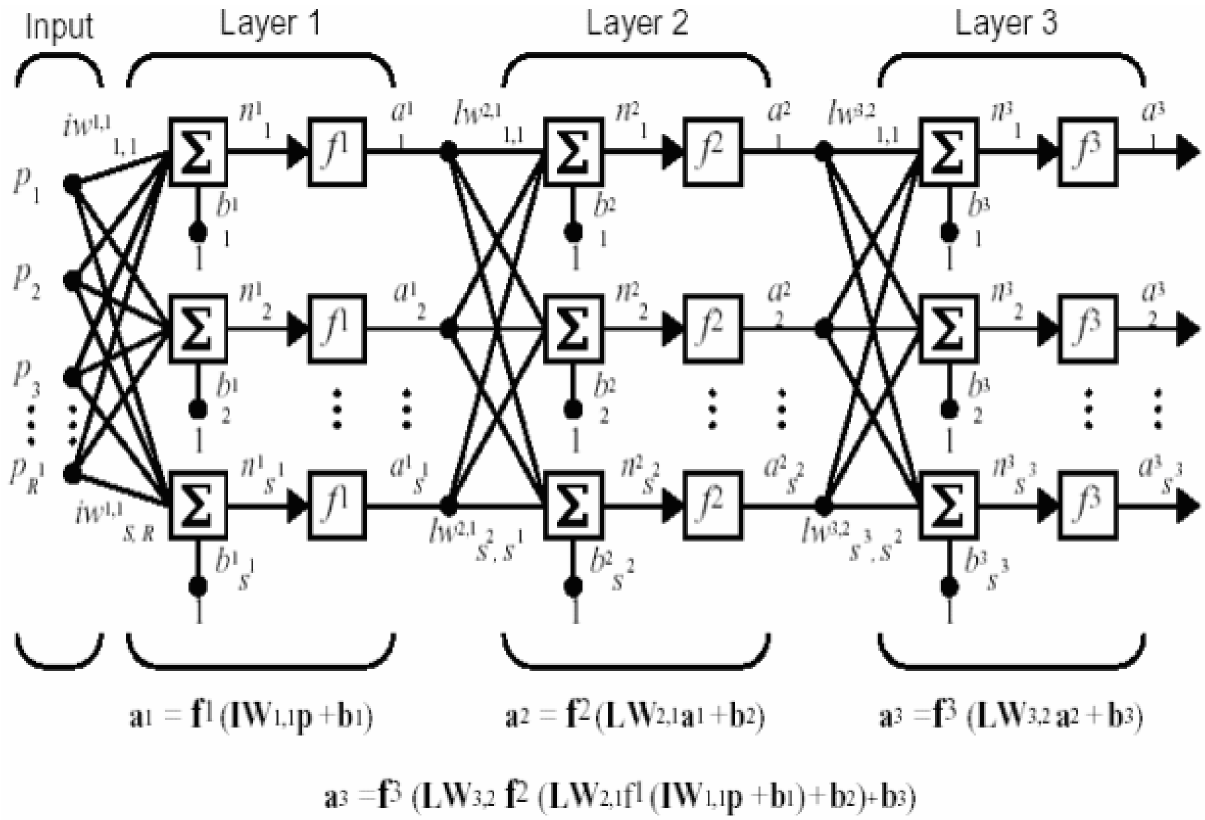


Figure (6.3): Multi-layer neural network.

,Multilayer networks are very effective networks, especially two-layer networks which are very widely used , as these networks can solve many complex problems but training them takes longer ,.

Neural Network Teaching Methods

The network learns by giving it a set of examples, which must be carefully selected, because this will contribute to The speed of the network learning. This . set of examples is called the training set

Neural network training methods are divided into two sections according to the training category presented to the network, which are:

1- SupervisedLearning :
   The training or education methods by the artificial neural network teacher are based on the idea of presenting the training data to the network in the form of a pair of forms, which arethe input and theoutput .
2- unsupervised learning:

In this method, the training class is just the input vector without displaying the target on the network, and this is called The self-learning method, where artificial neural networks build learning methods based on their ability to discover the distinctive characteristics of the shapes and patterns presented to them and their ability to develop an internal representation of these shapes without Prior knowledge and without giving examples of what she should produce, contrary to the principle followed in the teaching method By teacher.

This type of learning is divided into two types:

- Teacher-free education based on error correction:

This type of training is used to teach single-layer linear networks that are used to solve linear mapping problems. Between input and output, where the network calculates the error signal from the difference between the neuron's output and the output Required, and the weight values are modified by the error function called the cost function in order to minimize the difference by Derivation of this function with respect to synaptic weights. This method of teaching is considered one of the most . important and common methods of teaching by a teacher

- Education by a teacher based on memory:

is stored in the neural network, i.e. the training set is stored. It is the input beam and its corresponding output beam. This type of education requires a criterion to . determine the similarity of the beams and the presence of Education base

.

==
=================

Chapter 4

==
=================

Practical implementation on MATLAB

In this chapter we present the practical implementation on MATLAB and the .results

stages of algorithm operation

Design and training of the neural network

At this stage, the neuron is trained based on a pre-prepared list of uppercase and lowercase English letters. After the training is complete, the neural network is ready for testing. The training process used is untrained training, where images of characters are fed into the neural network and the ideal output is defined. To make .things more realistic, noise is added to the input and output vector

The neural network we designed is a feedforward neural networkand has an inner hidden layer of only 30 nodes. The input matrix is a matrix with dimensions50x81 and the output matrix is a matrix with dimensions 27x81. Figure (1.4) shows the neural network that was designed in MATLAB.

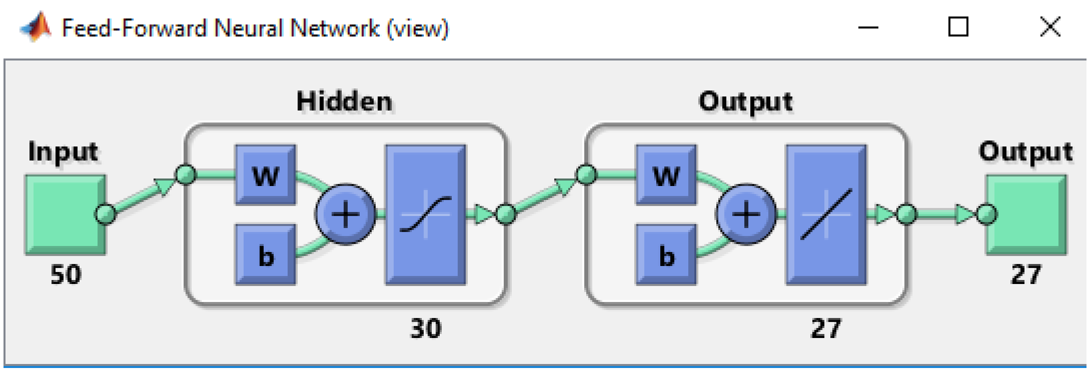


Figure (1.4): The neural network designed in MATLAB.

To make things more realistic, noise was added to the input, where 35 copies of the ,noisy input were created and fed into the neural network and trained on this basis since the image that will be fed into the neural network will not be perfect and free of noise, and the same applies to the output.

Forming the income statement for training the neural network 2.1.4

To train a neuron using the untrained method, we must have an ideal input and a corresponding ideal output. The ideal input was formed by a set of images .containing upper and lowercase English letters (the images are in the project file) The dimensions of each image are10x5 However, to feed these images into the . neural network, each image was converted into a single vector with dimensions50x1 The corresponding ideal output is a unitary matrix with dimensions .26x26 ,In fact . the number of characters fed into the neural network is 27, not 26, as a separator character/ was added to separate the words in the images. This will become clear in the examples included in the report.

Neural Network Training 3.1.4

As mentioned, the project was implemented in MATLAB. We made use of the ready-made libraries in MATLAB to form and train the neural network. The Levenberg-Marquardt training algorithm was adopted and performance was measured based on the mean squared error. The training process was completed after 28 iterations with a time of 12 minutes and 35 seconds on aCore i3 computer with 4 GB of RAM. Figure (2.4) shows the training process in MATLAB.

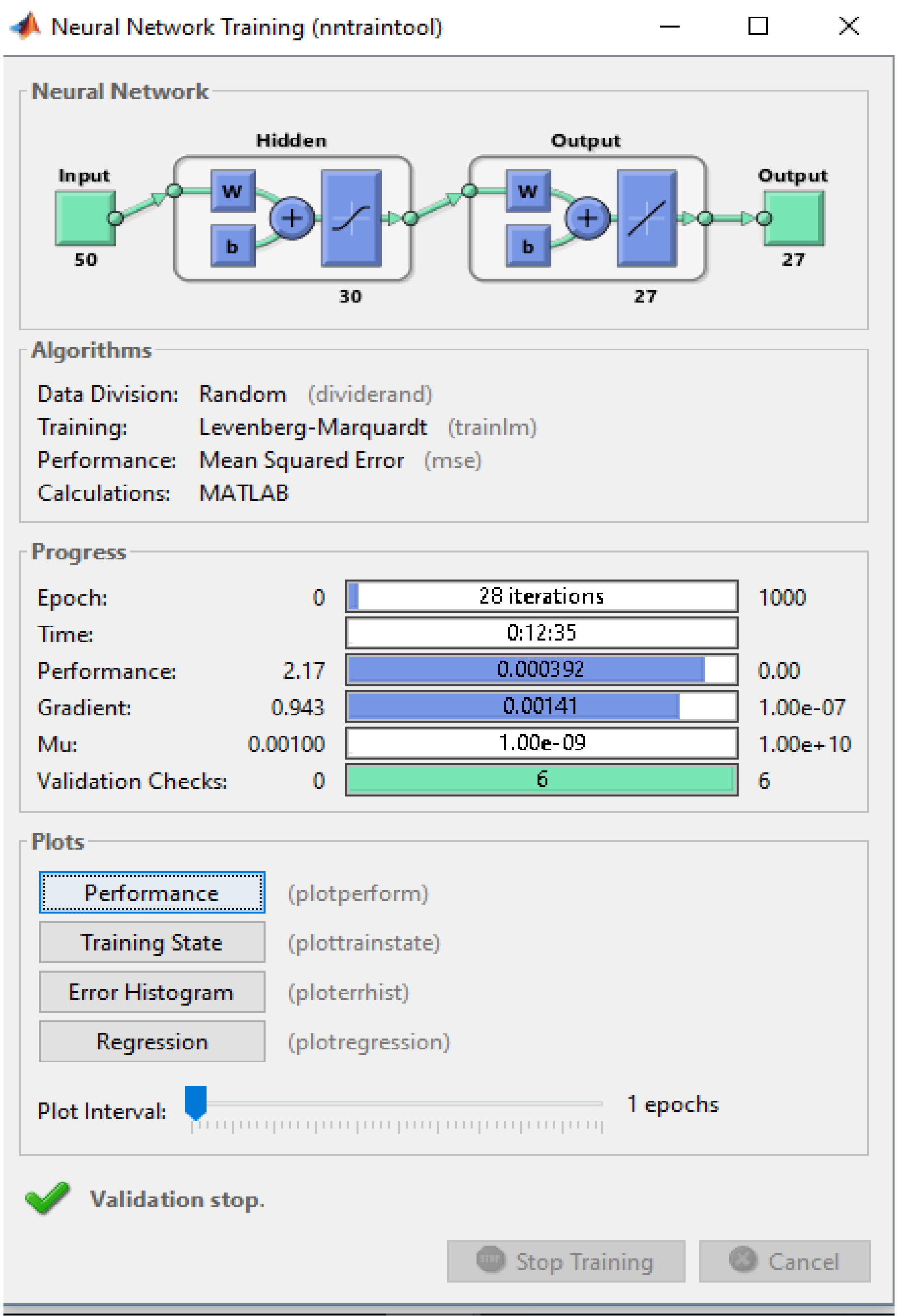


Figure (2.4): Training process in MATLAB.

Figure (3.4) shows the performance after the training process is completed.

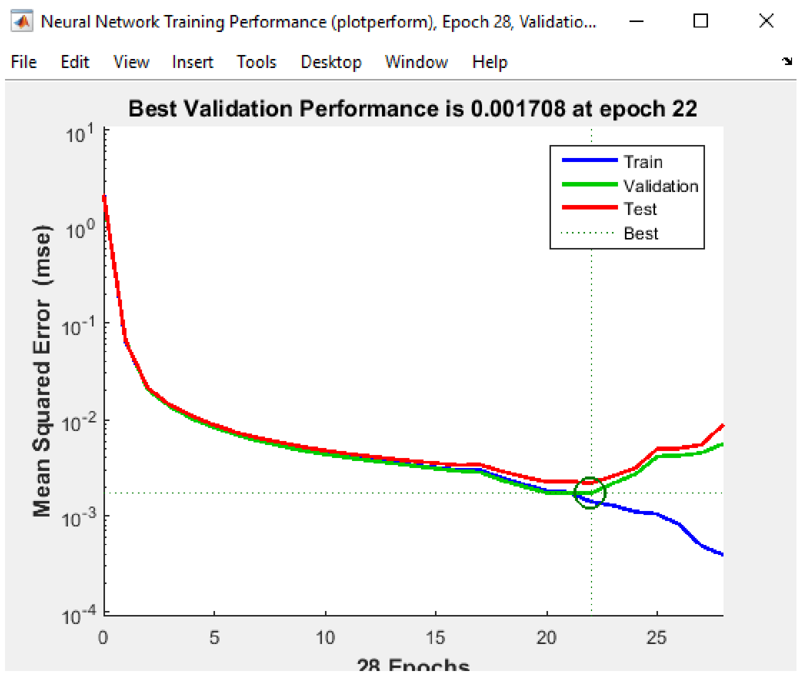


Figure (3.4): Performance of the neural network after completing the training process.

Figure (4.4) shows the training status after the neural network is finished training.

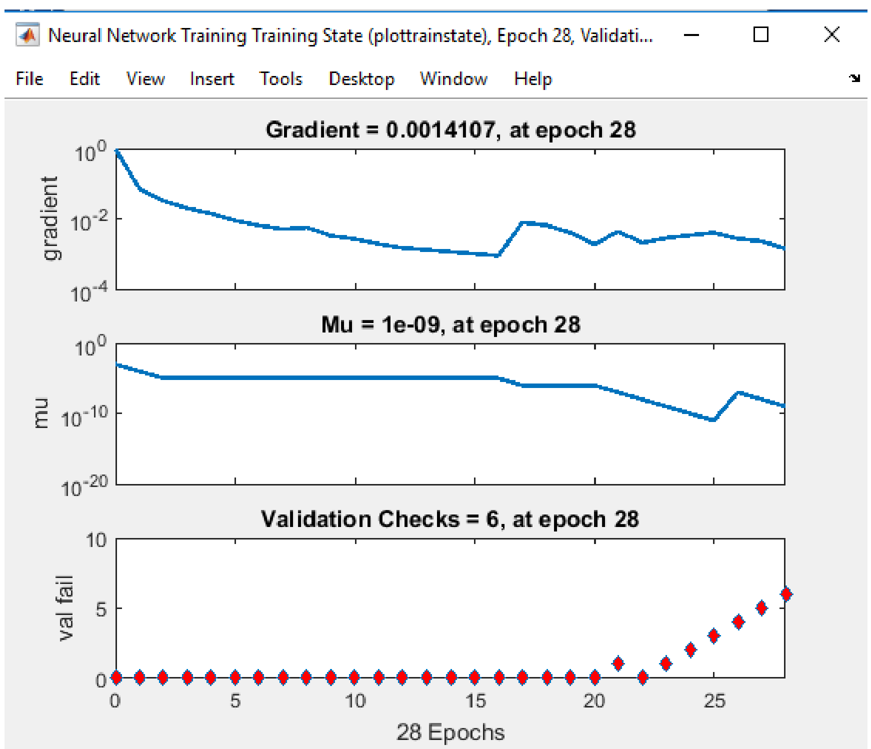


Figure (4.4): Neural network training case.

Figure (5.4) shows the histogram of the error.

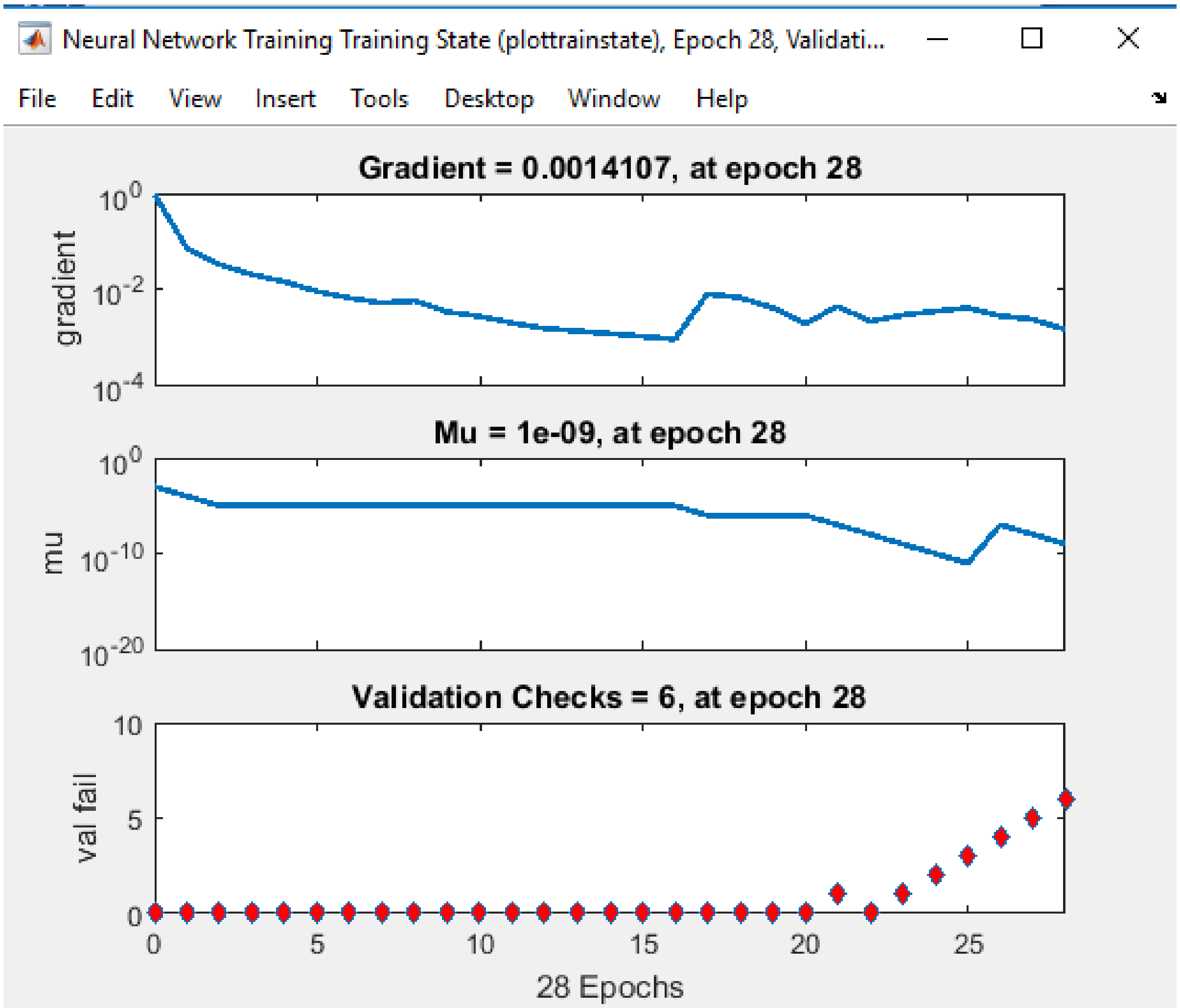


Figure (5.4): The histogram of the error.

Capture an image containing characters

At this stage, an image containing English characters, whether small or large, is) captured. Figure (1.4) shows an example of this. The character/ is considered a (separator between words.

Hi/i/am/From/Syria

Figure (1.4): An image containing the characters to be detected and recognized.

Extracting characters from an image

English characters are extracted from the image at this stage. First, the RGB image is convertedto grayscale and then to binary. Then, we perform the process of detecting the image regions and then extracting the image characters character by character.

Character recognition

After the characters are extracted from the image, they are fed into the neural network for recognition.

Display characters

At this stage the characters are displayed in the command window of the MATLAB program.

Implementing the algorithm in MATLAB

Image acquisition

At this stage, the image is read using theimread function and then displayed using the imshow function .Figure (6.4) shows this process .

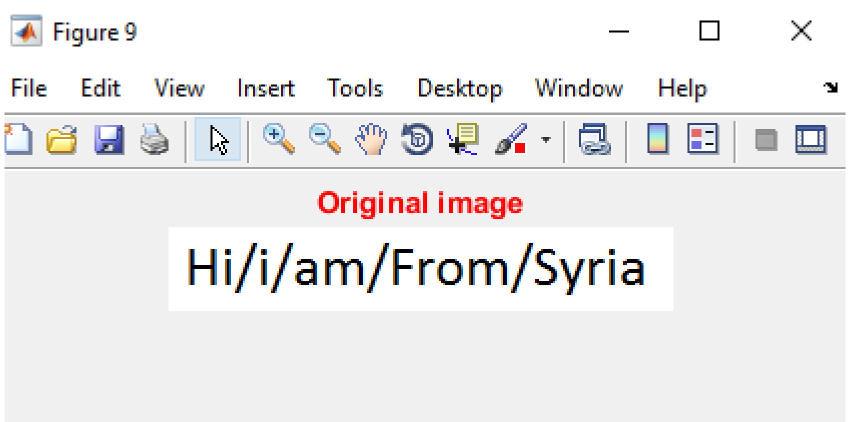


Figure (6.4): Reading the original image.

Character detection

First of all, the color image is converted to grayscale. Figure (7.4) shows the original image after converting it to grayscale. The aim of this process is to reduce the time required to detect characters in the image.

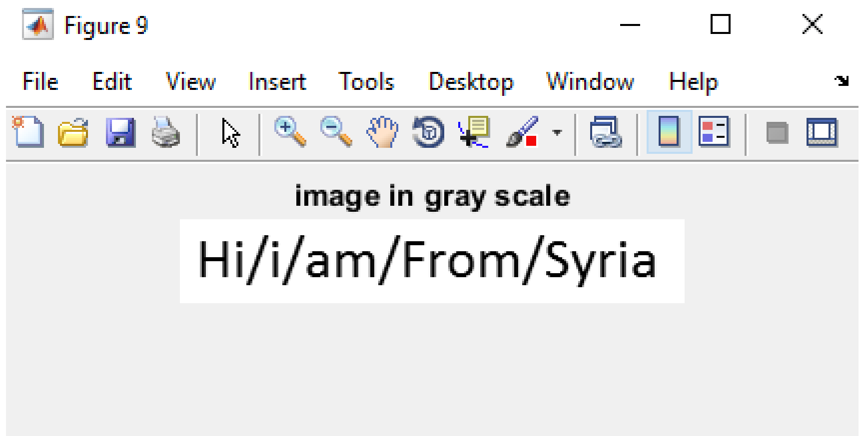


Figure (7.4): The original image after converting it to gray level.

The image is then converted to black and white. Figure (8.4) shows the original black and white image.

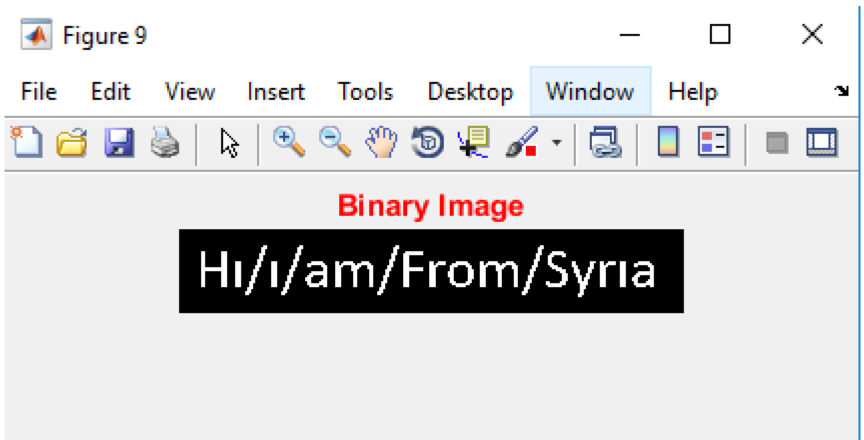


Figure (8.4): Binary image.

Image character recognition

After converting the image to binary, the regions of the image can now be identified. The regions in the image are the characters. Thebwlabel instruction can be used to identify the regions in the image. After that, theregionprops instruction can be used to give us the properties of the image. This instruction extracts the characters from the image (image regions) and stores them. Figure (9.4) shows the extracted letter H. Figure (10.4) shows the extracted letterF.

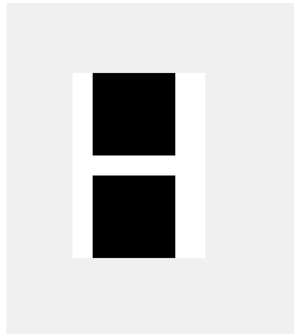


Figure (9.4): The truncated letterH.

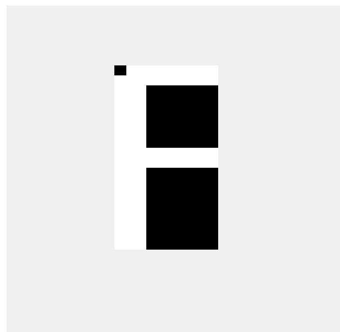


Figure (10.4): The truncated letterF.

Show characters

These characters are fed into the neural network, which recognizes them and then displays them in the command window. Figure (11.4) shows the characters displayed in the command window.

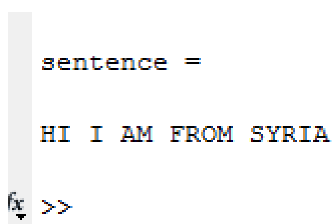


Figure (12.3): Displaying image characters in the command window.

Chapter 5

Conclusion and future prospects

In this chapter of the report, we present a conclusion and future prospects.

Conclusion

In this project, we implemented an algorithm to detect English characters in an image based on digital image processing and neural network. The task of the image processing algorithm is to detect characters in an image, cut them out, and then feed them into the neural network, which in turn recognizes the characters.

Future Prospects

The algorithm can be developed to recognize Arabic characters.

References

[1]: Bijender Mehandia, Ragini Bhat, “RECOGNITION OF VEHICLE NUMBER PLATE USING MATLAB ”, Gurgaon Institute of Technology & Management, Gurgaon, 2014, India.

[2]: Alhamzawi Hussein, Ali mezher, “Automatics Vehicle License Plate Recognition using MATLAB”, Faculty of Informatics/University of Debrecen, Hungary, 2017.

[3]: Homayoon SM Beigi, “An Overview of Handwriting Recognition,” Proceedings of the 1st Annual Conference on

[**4]:** Nadal, C. Legault, R. Suen and CY, “Complementary Algorithms for Recognition of Totally Uncontained Handwritten Numerals,” in Proc. 10th Int. Conf. Patter Recognition, 1990, vol. 1, pp. 434-449.