Zewail City of Science and Technology
University of Science and Technology
Linear Algebra and Vector Geometry - SPRG2022

Linear Algebra and Vector Geometry
Project Report

Image Processing

By: Alaa Mostafa 202-000-971
Marwa Elbadry 202-001-052
Riham Mohamed 201-901-639

Dr. Hatem Adel Fayed.

Table of Contents:

Abstract:

The project is mainly about image processing using some linear algebra mathematical equations. Processing of images is a method in which some operations are performed on an image to get an output that can be the same image enhanced or features associated with that image. With the advent of computers, processing has come to be done on digital images acquired through the digitization process or directly using any digital device, using computer graphics algorithms. Doing image processing on digital images using a computer is called digital image processing. The mage can be represented as a matrix; as each pixel of the image can be considered as an entry of a matrix. If a white & black image was taken as an example, each entry is between 0 (might represent black) and 1 (might represent white). The more the matrix entries of a particular image are, the more pixels it has, and accordingly, the more enhanced the image is. Image processing has many applications like Face Recognition, Image editing, photoshop, and other technical needs.

Introduction:

Visual interpretations have become an important aspect of communication in the millennial era. The desire for greater image quality has risen with the emergence of image editing and capture software on the market however, photographs are not only limited to merely flaunting.

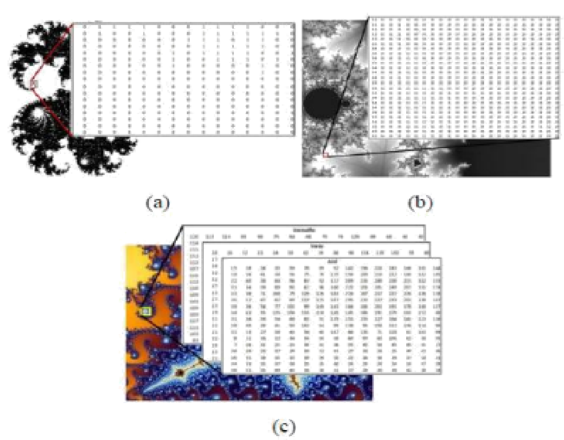
Nowadays, Images have become such a vital part of our lives that we often don't even realize we've come across one. We are used to clicking, filtering, improving, saving, uploading, and performing a variety of other things on images. They are really all around us.

But, what exactly is an image? An image is a two-dimensional graphical representation of a two-dimensional shape. Because a picture is made up of many rows and columns, it necessitates the use of a syntax called Image Matrix to express it.

Digital Image Processing refers to the applications that a digital computer uses to process digital images using an algorithm. The captured image is then utilized to run computer processes in order to create an enhanced version of it or extract the needed information from it. And now with the advancement of technology, digital image processing applications are rapidly infiltrating practically all areas.

Methodology:

The images can be read as 2D matrices by a computer. Each image contains pixels that can be considered as an entry of a matrix. In binary images, the values of each pixel are 255(white) and 0 (black). The most common grayscale image using 8 or 16-bit per pixel, for example, (2^8 = 256, 0 → 255, 256 *values*). In the RGB system, the intensity of the color of the pixels can be measured by a vector containing the values of r, g, and b.
Therefore, R is the amount of red in the pixel, G is the amount of green, and b is the amount of blue color. Their values are integer numbers between 0 and 255.



We can use the operations of matrices in image processing, for example:

1- Adding 2 matrices to merge 2 images together:

$$A = \begin{bmatrix} a_{11} & a_{12} & . & . & a_{1n} \\ a_{21} & a_{22} & . & . & a_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ a_{m1} & a_{m2} & . & . & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & . & . & b_{1n} \\ b_{21} & b_{22} & . & . & b_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ b_{m1} & b_{m2} & . & . & b_{mn} \end{bmatrix}$$

$$\begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & . & . & a_{1n}+b_{1n} \\ a_{21}+b_{21} & a_{22}+b_{22} & . & . & a_{2n}+b_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ a_{m1}+b_{m1} & a_{m2}+b_{m2} & . & . & a_{mn}+b_{mn} \end{bmatrix} \quad (2.1)$$

(a)

(b)

(c)

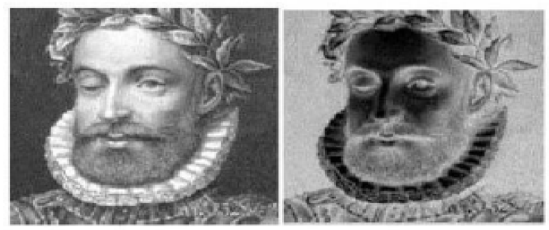
2- Subtraction can be used to delete the background of an image or produce a negative image:

$$A - B = A + (-1 \times B)$$

$$= \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & \cdot & \cdot & a_{mn} \end{bmatrix} - \begin{bmatrix} b_{11} & b_{12} & \cdot & \cdot & b_{1n} \\ b_{21} & b_{22} & \cdot & \cdot & b_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{m1} & b_{m2} & \cdot & \cdot & b_{mn} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} - b_{11} & a_{12} - b_{12} & \cdot & \cdot & a_{1n} - b_{1n} \\ a_{21} - b_{21} & a_{22} - b_{22} & \cdot & \cdot & a_{2n} - b_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1} - b_{m1} & a_{m2} - b_{m2} & \cdot & \cdot & a_{mn} - b_{mn} \end{bmatrix} \qquad (2.2)$$

(a) (b) (c)

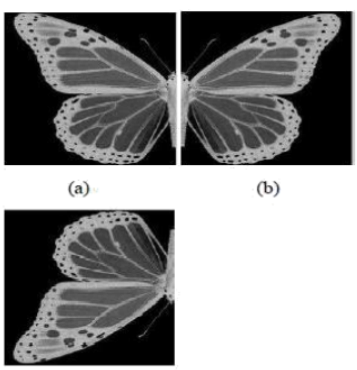


(a) (b)

3- Multiplications are to invert the image:

$$A \times B = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & a_{1p} \\ a_{21} & a_{22} & \cdot & \cdot & a_{2p} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & \cdot & \cdot & a_{mp} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} & \cdot & \cdot & b_{1n} \\ b_{21} & b_{22} & \cdot & \cdot & b_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{p1} & b_{p2} & \cdot & \cdot & b_{pn} \end{bmatrix}$$

$$= \begin{bmatrix} c_{11} & c_{12} & \cdot & \cdot & c_{1n} \\ c_{21} & c_{22} & \cdot & \cdot & c_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ c_{m1} & c_{m2} & \cdot & \cdot & c_{mn} \end{bmatrix} \qquad (2.3)$$



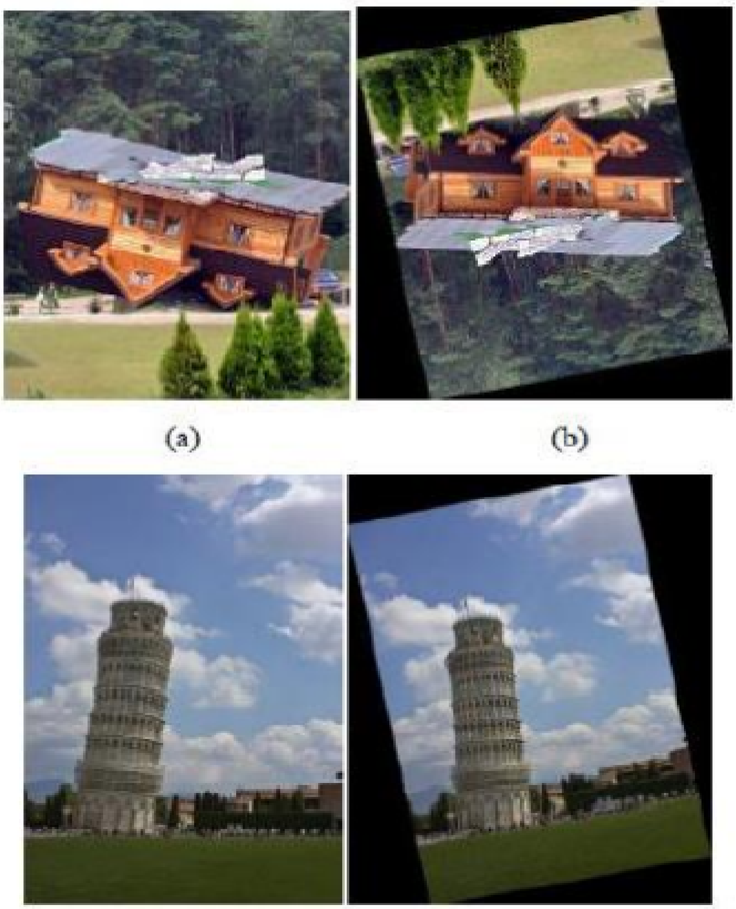
(a) (b)

(c)

4- Rotation is to rotate all pixels of the image around the origin without any deformations to the images.

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(a) was rotated ($-\pi$ /16) to produce the image (b)
image (c) was rotated ($-\pi$ /16) radians to produce the image (d)



(a) (b) (c) (d)

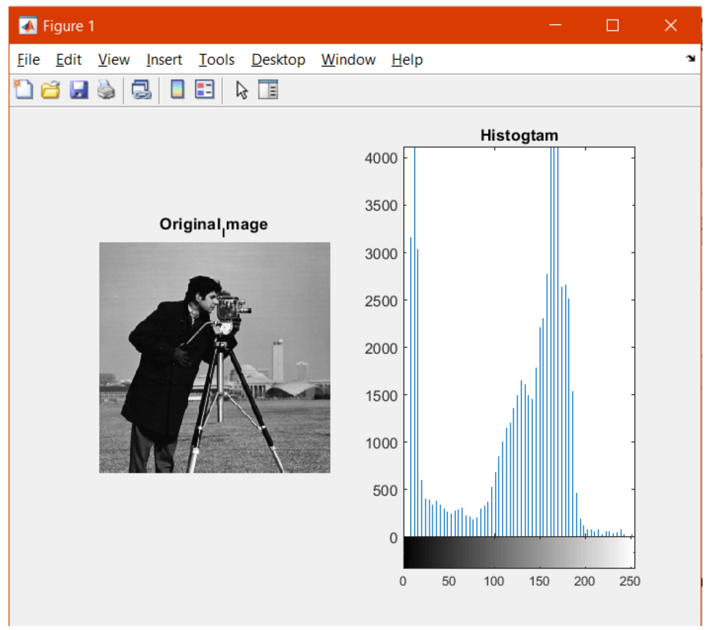
<u>Matlab Code: Histogram Processing</u>

The histogram of an image represents the distribution of the pixel intensity values. For an 8-bits grayscale image, there are 256 different intensity values from 0 to 255; (0 is black, 255 is white, and values in between are shades of grey). The histogram shows the distribution of pixels among those 256 grayscale values. By observing the histogram of an image, the image can be dedicated to whether it’s dark or light (i.e. the brightness of the image) and whether the intensity values it has; small or large dynamic range (i.e. contrast of image). By knowing this information about the image, it can be processed to improve its quality.

The Algorithm steps:

- Loading the image which is desired to have the edit in.
- Displaying it and its histogram side by side in one figure (using subplot() function).

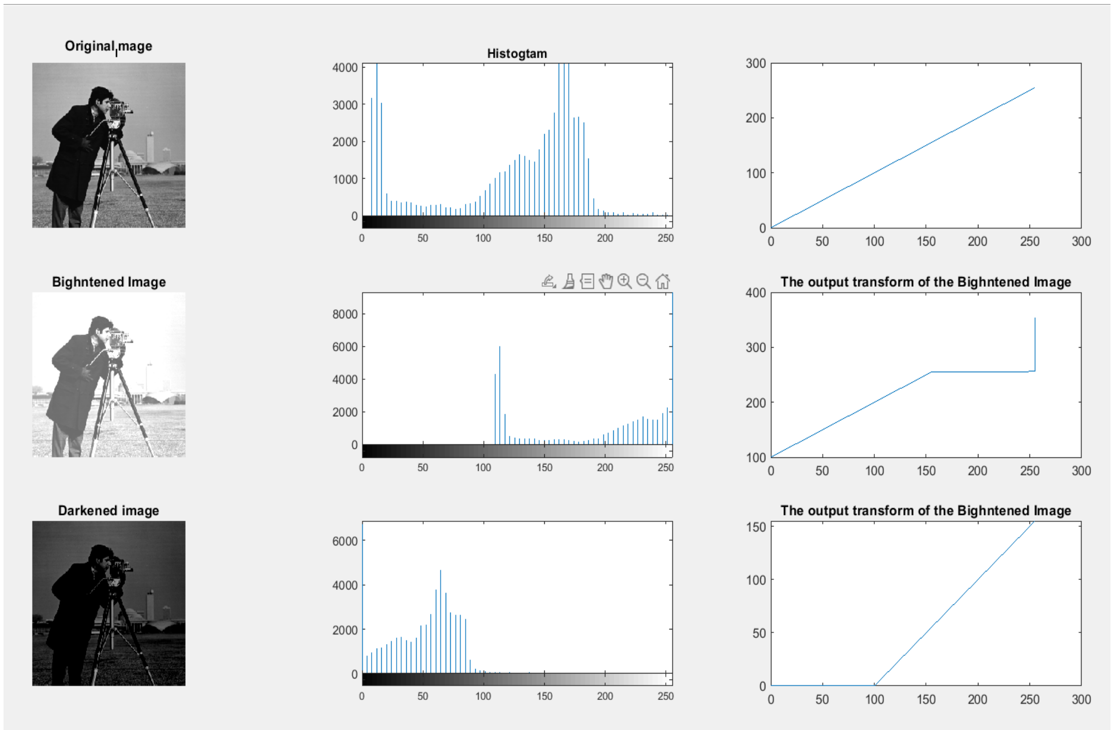
```
Image=imread('cameraman.tif'); %reading the image
figure
subplot(1,2,1);
imshow(Image);  %plotting the image
title("Original_Image");
subplot(1,2,2);
imhist(Image,64);  %plotting histogram of image
title("Histogtam");
```



- Adjusting the brightness of the image; Increasing the brightness of it, by adding a positive value to every pixel in the image (The output pixel values above 255 are saturated at 255)
- Decreasing brightness, by adding a negative value to every pixel (The output pixel values below 0 are saturated at 0).
- Displaying the original image, brightened image, and darkened image in one figure (using subplot() function).
- Displaying the histogram for each one.

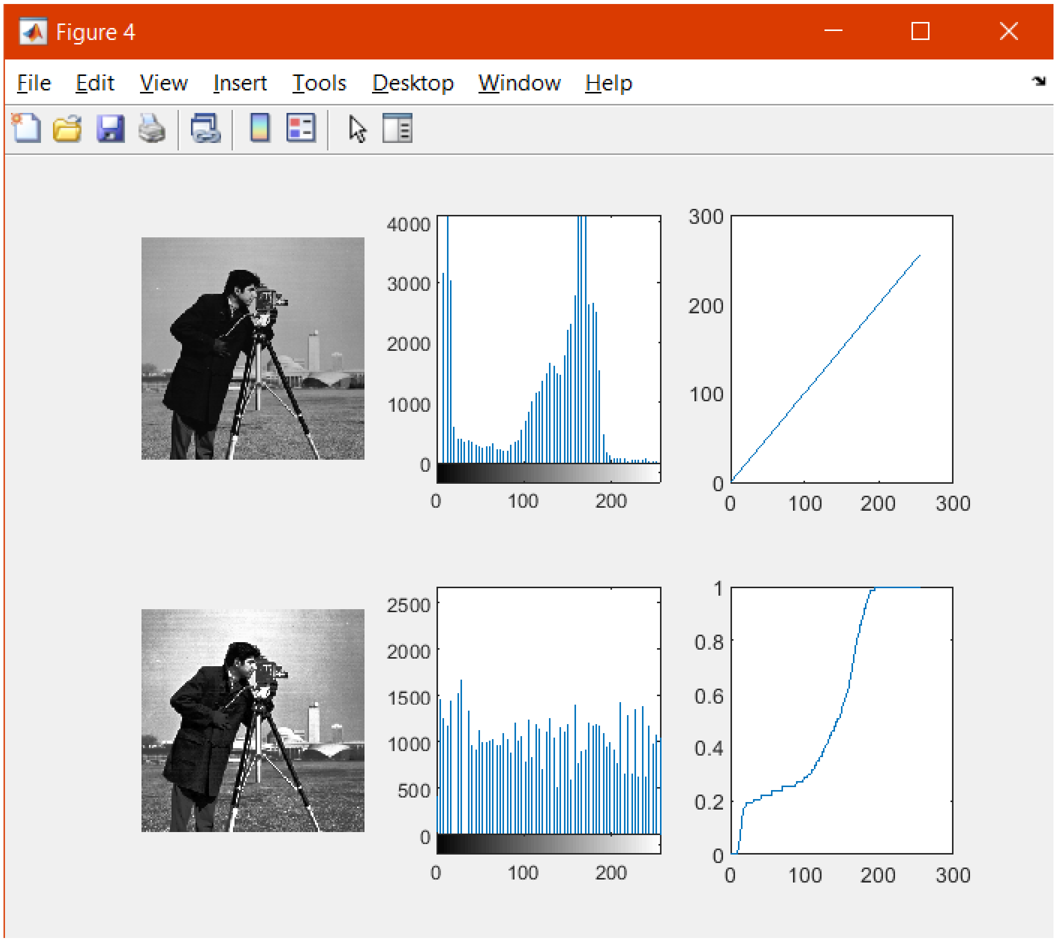
```
X = 0:1:255;
Y = X;
figure
subplot(3,3,1);
imshow(Image);  %plotting the image
title("Original_Image");
subplot(3,3,2);
imhist(Image,64);  %plotting histogram of image
title("Histogtam");
subplot(3,3,3);
plot(X,Y);  %plotting the X,Y axes
[row, col] = size(Image); %identifying the size of the image
brighted_Image = uint8(zeros(row,col)); %The Brighted Image
darkened_Image = uint8(zeros(row,col)); %The Darkened Image
for R = 1:row
    for C = 1:col
        New_Image = 100 + Image(R,C); %Adding 100 to every pixel in the image
        if New_Image > 255
            New_Image = 255;
        end
        brighted_Image(R,C) = New_Image; %Now the new image is The Brighted Image
    end
end
Y = X + 100;
for i=1:255
    if Y(i) > 255
        X(i) = 255;
    end
end
subplot(3,3,6);
plot(X,Y); %plotting the output transform of the Bighntened Image
```

```
title("The output transform of the Bighntened Image");
subplot(3,3,4);
imshow(brighted_Image); %Plotting the Bighntened Image
title("Bighntened Image");
subplot(3,3,5);
imhist(brighted_Image,64); %Plotting histogram of the Bighntened Image
for R = 1:row
    for C = 1:col
        new_Image = Image(R,C) - 100; %Subtracting 100 from every pixel in the image
        if new_Image < 0
            new_Image = 0;
        end
        darkened_Image(R,C) = new_Image;  %Now the new image is The Darkened Image
    end
end
Y = X - 100;
for i=1:255
    if Y(i) < 0
        Y(i) = 0;
    end
end
subplot(3,3,9);
plot(X,Y); %plotting the output transform of the Darkened Image
title("The output transform of the Bighntened Image");
subplot(3,3,7);
imshow(darkened_Image); %Plotting the Darkened Image
title("Darkened image");
subplot(3,3,8);
imhist(darkened_Image,64); %Plotting histogram of the Darkened Image
```



- Adjusting the contrast of the image is by performing histogram equalization.
- Displaying the original image and enhanced image in one figure.
- Displaying the histogram for each.

```
[Enhanced_image, Enhanced_transform] = histeq(Image); %to return the output transform of Enhanced image.
figure
subplot(2, 3, 1);
 imshow(Image); %plotting the original image
 subplot(2, 3, 2);
 imhist(Image,64); %plotting the histogram of original image
 X = 0:1:255;
Y = X;
subplot(2, 3, 3);
plot(X,Y);
subplot(2, 3, 4);
imshow(Enhanced_image); % plotting the Enhanced image
subplot(2, 3, 5);
imhist(Enhanced_image,64); % plotting the histogram of the Enhanced image
subplot(2, 3, 6);
 plot(Enhanced_transform); %the output transform of Enhanced image.
```



Code, test image, and all Matlab figures will be found through this drive.

Applications for image processing:

IP has many applications that we can see in our daily life

1. Image sharping and restoration:

   It is the process that was made on a captured photo to reach the desired result. It can be simply described as what Photoshop does like zooming, blurring, sharpening, grayscale to color conversion, detecting edges and vice versa, Image retrieval and Image recognition.

Blurred Background

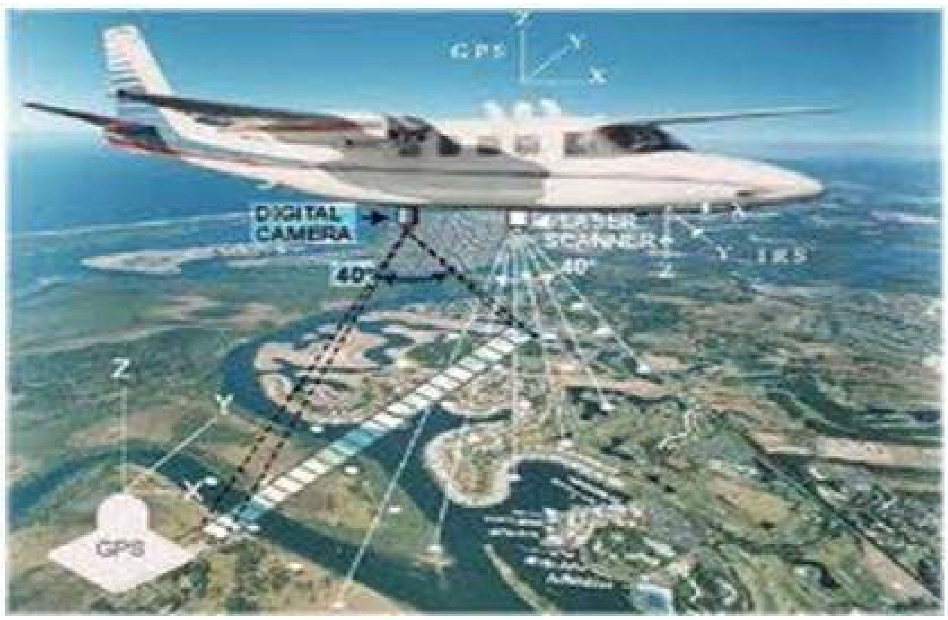
Original Image

After Zooming

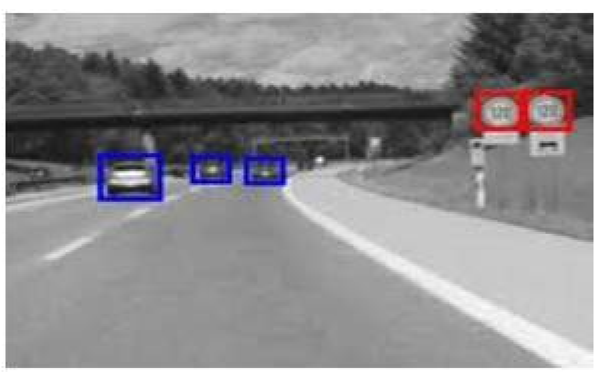
2. UV Imaging:

A satellite or a very high ground scans a region of the earth, which is subsequently processed to get information. The detection of infrastructure damages induced by an earthquake is one application of digital image processing in the field of remote sensing.
Even if substantial damages are concentrated, it takes longer to understand the harm. Because the impacted region is sometimes so large, it is impossible to inspect it with the naked eye in order to quantify damages. Even if it is, then it is a very hectic and time-consuming procedure. As a result,

digital picture processing provides a solution. A photograph of the affected region is taken from above ground and examined to determine the different types of damage caused by the earthquake.



3. Detecting stumbling blocks:
   It is used to detect different objects in a picture and then to estimate the distance between the robot and the hurdles.



4. Facial Recognition:

This application is nowadays part of our daily life as it is used in smartphones to strengthen security by allowing only the owner or someone whose face the system recognises to use the device. That technology uses a camera and stored data to verify an individual's identification. The programme is built on algorithms that recognise the individual's particular qualities while also memorising them in the data so that they may be distinguished from other people later. More than simply features are captured by face recognition systems. It is programmed in such a way that it can capture all of the distinctive features of a face from numerous angles. It measures and remembers the distance between a person's eyes and lips, for example. It operates in the same way as a 2D or 3D model does, in that the face prints of individuals are collected and saved from various angles. The programme is capable of totally ignoring the ageing effect and memorising aspects that do not change over time. When the data is saved on the device, the programme can readily recognise the individual even in a picture or video, and it can also discriminate between numerous individuals.

5. Robot Vision:

   Robot vision is the technique of employing a mix of camera hardware and computer algorithms to enable robots to analyse visual input from the environment. Digital image processing is carried out by a number of robotic machines as they navigate using image processing techniques. Robots are virtually blind without digital picture processing.

   Robots employ vision to perform complicated tasks in a continuously changing environment. Digital cameras have sophisticated technology that allows them to shift high-resolution pixel arrays to the robot's computer. Digital image processing algorithms enhance and interpret these pictures. Now, It is critical for the robot to be able to see objects, recognise them, and detect any obstacles, allowing people to make use of technology and digital image processing.

Conclusion:

Image processing has many techniques to be done. Any image whether it’s on grayscale or coloured can be processed using many software applications and programming languages. Image processing has many applications; It’s not limited to only editing and enhancing images. Face recognition which’s one of the most important applications for image processing can be used to catch criminals and find lost people and children. This leads to how vital and many uses image processing can be used in. Matlab is one of the best tools that can be used to process images.

References:

[1] Gonzalez, R.C., and Woods, M.P. Digital image processing. Prentice Hall, 2nd edition, 2002.
[2] Pratt, W.K. Digital image processing. John Willey & Sons, Inc., 3rd edition, 2001.