

# FACULTY OF ENGINEERING & TECHNOLOGY

Group Assignment

ERT3046: Machine Vision

Trimester 1 2015/2016

**Image Group: 2**

**Group name: The Engineers**

Major: RE Date of Submission: 15/9/2015 (week 12)

|  |  |
| --- | --- |
| **Name** | **Student ID** |
| 1. NABIL FIKRI BIN MOHD NAJIB (GL) | 1121117342 |
| 1. TAREK ALBAWAB | 1121118577 |
| 1. MUHAMMAD AMMAR BIN PUTRA ABD RAHMAN | 1122701199 |
| 1. AHMED TAWFIK EL FARRA | 1111112349 |
| 5. ABDULSALAM FARHAT | 1111113572 |

Table of contents:

Objectives …………………………………………………...3

Introduction ………………………………………………... 3

Job Distribution ……………………………………………. 4

Implementation planning Gunn Chart (time-line)…………. .4

Methodology ………………………………………………. 5

Results …………………………………………………….. 11

Conclusion ………………………………………………... 15

References ……………………………………………….... 15

**1.0 INTRODUCTION**

**Machine vision** is the current technology used to provide imaging-based automatic inspection and analysis for applications. This includes automatic inspection, process control, and robot guidance in industry. The scope of this technology is wide. The primary uses for machine vision are automatic inspection for product manufacturing and [industrial robot](http://en.wikipedia.org/wiki/Industrial_robot) control. Commonly, machine vision are applied in applications such as [quality assurance](http://en.wikipedia.org/wiki/Quality_assurance), sorting, material handling, robot guidance, and optical gauging.

In this experiment job scope, students are required to design a machine vision software that can inspect product quality through the use of image processing. Images are processed by comparing to the reference image of accepted quality. These can be done by using several methods and techniques that later will be discussed in details in methodology part.

**1.1 OBJECTIVES**

Trimester 1 15/16, under the subject of Machine Vision ERT 3046 we were given an assignment covering all chapters from the subject. The assignment is done via Visual C# .NET. In addition, 3rd party library such as Aforge.NET, Accord.NET, EmguCV and etc. are referred in order to complete our assignment. The topic is algorithm development in nature. Below are the objectives to meet:

* Apply binary image processing and feature extraction techniques. (a)
* Apply segmentation and contour extraction. (b)
* Analyse texture patterns. (c)
* Evaluate the performance of classification process. (d)
* Locate four side edges and form a rectangle. (e)
* Calculate the different in position and angle shift respect to reference image. (f)
* Perform contrast analysis and shape recognition on bright round area. (g)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PLANNING\ WEEK | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Study for overall requirements and work distribution |  |  |  |  |  |  |  |  |  |
| Locate four side edges and form a rectangle |  |  |  |  |  |  |  |  |  |
| Calculate the different in position and angle shift respect to reference image |  |  |  |  |  |  |  |  |  |
| Perform contrast analysis and shape recognition on bright round area |  |  |  |  |  |  |  |  |  |
| Trouble shooting the system and test the GUI using different PCs |  |  |  |  |  |  |  |  |  |
| Prepare the final report and finalize the software |  |  |  |  |  |  |  |  |  |

**1.2 IMPLEMENTATION PLANNING GUNN CHART (TIME-LINE)**

**1.3 JOB DISTRIBUTION LIST**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Group Member | I.D | Job Distribution |
| 1. | NABIL FIKRI BIN MOHD NAJIB (GL) | 1121117342 | Methodology and Coding |
| 2. | TAREK ALBAWAB | 1121118577 | Results and Coding |
| 3. | MUHAMMAD AMMAR BIN PUTRA ABD RAHMAN | 1122701199 | Cover page, Introduction, Objectives and Task distribution list |
| 4 | AHMED TAWFIK | 1111112349 | Gantt Chart and Table of Contents |
| 5 | ABDULSALAM FARHAT | 1111113572 | Conclusion and Printing |

**2.0 METHODOLOGY**

In this portion, the codes and the way of our codes work will be explained in brief.

1. The UPLOAD buttons

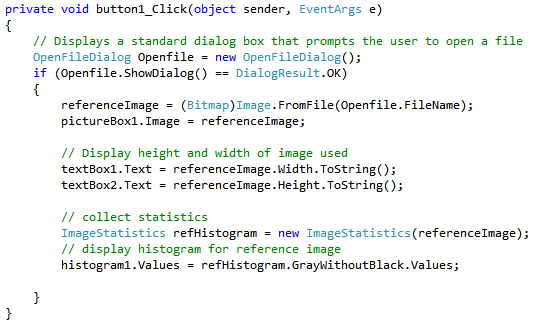
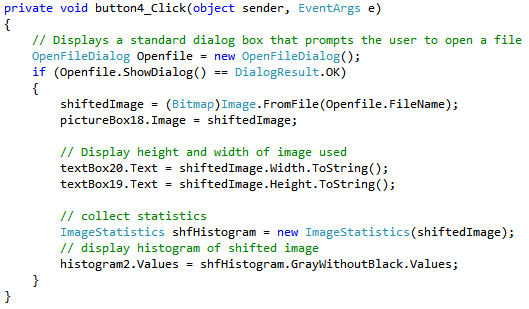


Figure 2.1 Upload button

Both of the image refer to the UPLOAD button execution. The first image describes the UPLOAD button for Reference Image and the second image is for Shifted Image. After each buttons are pressed, user will be asked to select images from folders and after selecting the images, histogram of GrayWithoutBlack will appear for both images. The histograms are used for indication of the grayscale images intensity without black colour. We tried to avoid using histogram of gray and black colour together because black colour happened to produce unwanted spikes on our histogram and hence prevents us from seeing the real representation of our images.

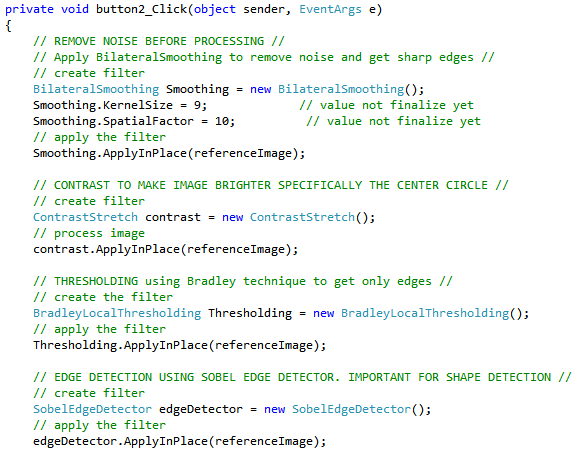
2. The PROCESS IMAGE buttons



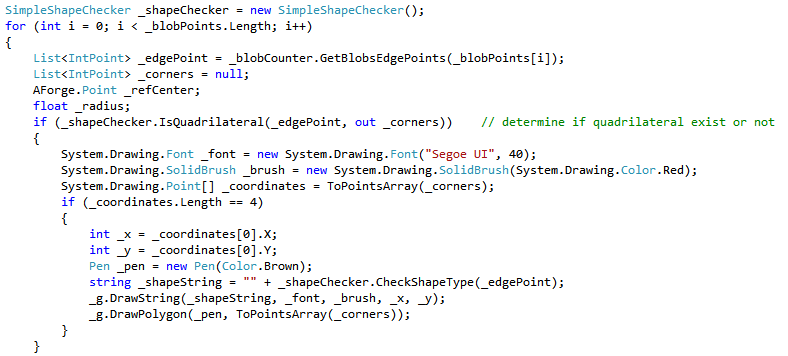
Figure 2.2 Process Image buttons

After uploading images, we proceed to process the images to fulfill the requirements listed. The algorithm in Figure 1.2 starts from:

1. Remove noises using ***Bilateral Smoothing Filter***. Bilateral filter conducts "selective" Gaussian smoothing of areas of same colour (domains) which removes noise and contrast artifacts while preserving sharp edges. We use KernelSize of 9 and SpatialFactor of 10 to get what the results that we wanted. \*Achieve (a)
2. Make the images brighter using ***Contrast Stretch Filter****.* It is a simple image enhancement technique that attempts to improve the contrast in an image by 'stretching' the range of intensity values it contains to span a desired range of values. \*Achieve (g)
3. Threshold the images using ***Bradley Local Thresholding*** technique. Every image's pixel is set to black if its brightness is *t* percent lower than the average brightness of surrounding pixels in the window of the specified size, otherwise it is set to white. We use this as it threshold the line to be the object and that is enough for us to do the processing later.

\*Achieve (a)

1. Detect edges of shape in images using ***Sobel Edge Detection*** method and form shapes. The filter searches for objects' edges by applying Sobel operator. Each pixel of the result image is calculated as approximated absolute gradient magnitude for corresponding pixel of the source image. Besides, the *Sobel Edge Detection* gives us thicker edges formation. \*Achieve (b) and (e)
2. Make the lines of shape thicker using ***Dilation*** filter. The filter assigns maximum value of surrounding pixels to each pixel of the result image. Surrounding pixels, which should be processed, are specified by structuring element: 1 - to process the neighbor, -1 - to skip it. \*Achieve (b)
3. Separate each component and label them with different colours using ***Connected Component Labeling*** method. The filter performs labeling of objects in the source image. It colors each separate object using different color. The image processing filter treats all none black pixels as objects' pixels and all black pixel as background. \*Achieve (b)
4. Extract standalone objects in images using ***BlobCounter*** filter. We extract only the biggest rectangle from images by specifying the boundary needed. \*Achieved (e) and (g)
5. From the extracted objects, perform quadrilateral and circle detection using ***SimpleShapeChecker*** class for provided set of points (shape's edge points). During the check the class goes through the list of all provided points and checks how accurately they fit into assumed shape. \*Achieved (e) and (g)
6. Using the centre coordinates of circles detected from reference and shifted images, perform image position shifting and angle shifting calculation. \*Achieved (f)
7. Extract the images features using ***GrayLevelCooccurrenceMatrix*** Class. The class used can capture properties of a texture. Features extracted are the values of entropy, energy, homogeneity and contrast. \*Achieve (c) and (d)



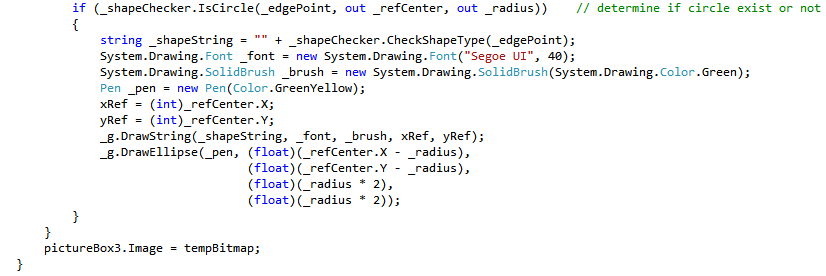


Figure 2.3 Shape Detection Algorithm

From Figure 1.3, shape detection is performed to determine quadrilateral shape and circle shape. Both UPLOAD buttons are using the same algorithm. However for shifted image processing, we include the calculation for determining the position and angle shifted.

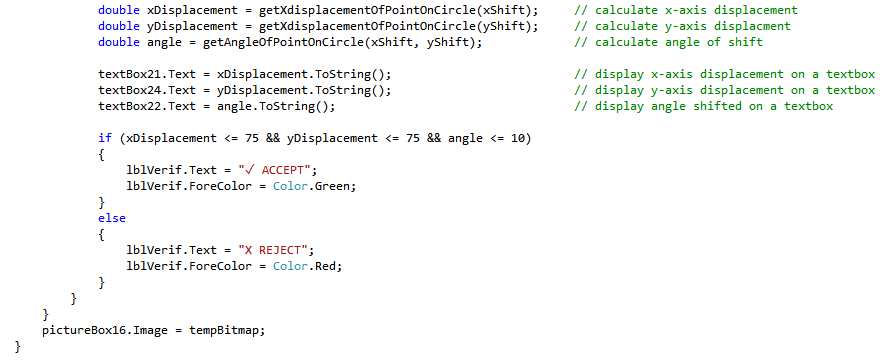
 Figure 2.4 Position and Angle Calculation

Figure 1.4 show that calculation is being done and it will be displayed using textboxes. We check if the x - y axis and angle shifting not more than 75 pixels, 75 pixels and 10 degrees, the image can be accepted. It depends on manufacturers to adjust the conditions that they need in order to match their product requirements.

3.0 Calculation Formula

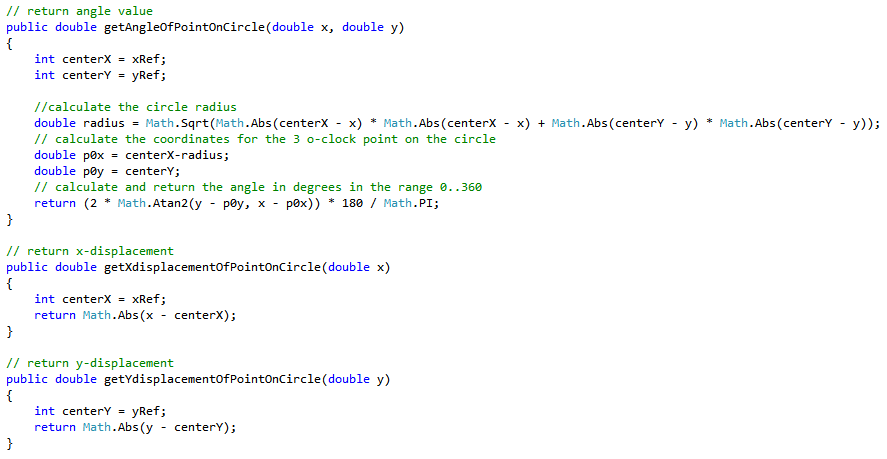


Figure 2.5 Formula for Position and Angle Shifting

Figure 1.5 illustrated the simple formula that we used in order to determine the position and angle shifting. The x and y reference point we took it from the centre coordinate of reference image circle. The formula that we used are as follows:

For angle shifted calculation:

1. Start from 3 o’clock axis, we calculate

For position shifted calculation:

4. Texture Analysis

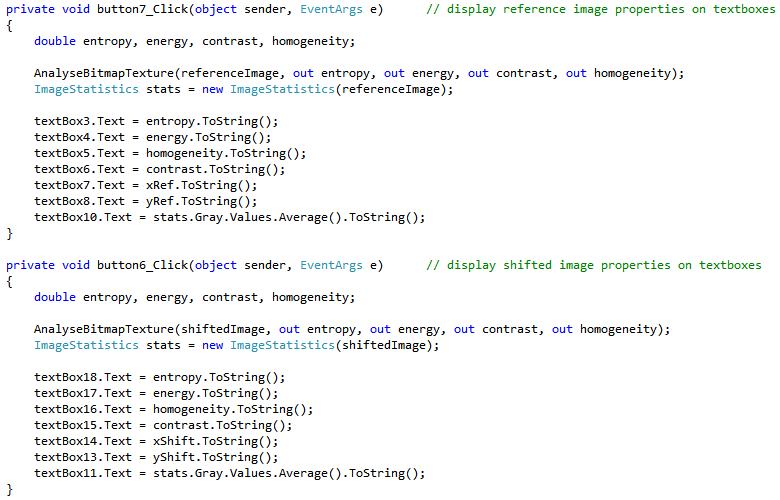


Figure 2.6 Texture Properties Displayed in Textboxes

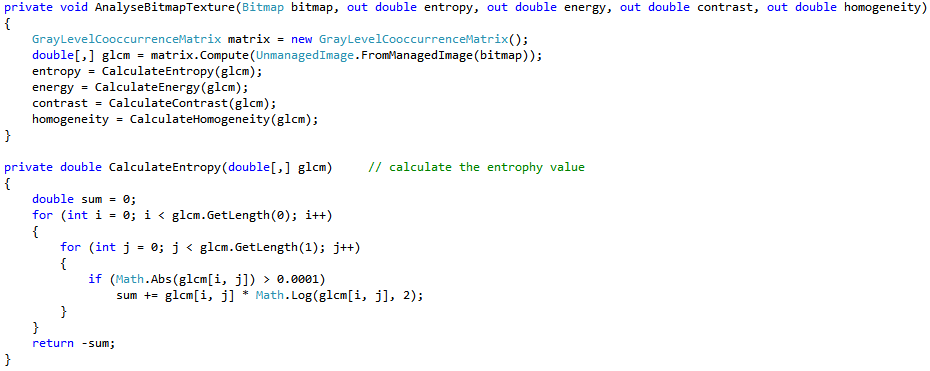
Figure 1.7 shows some formula that we used to calculate the value of entropy, energy, homogeneity and contrast. These formula are: \*Achieve (d)











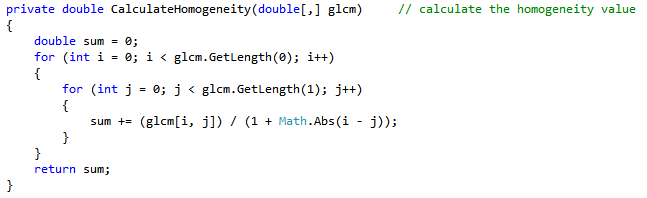
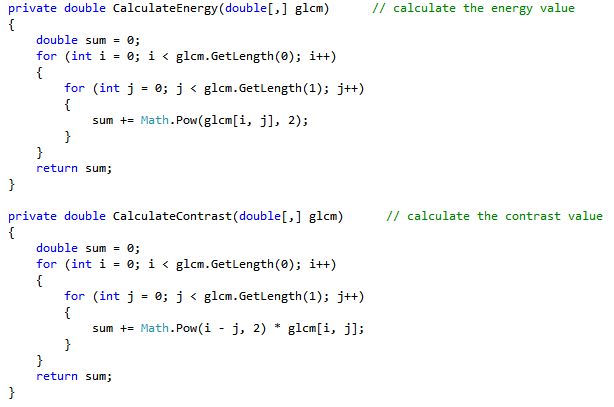


Figure 2.7 Calculation Formula for Texture Analysis

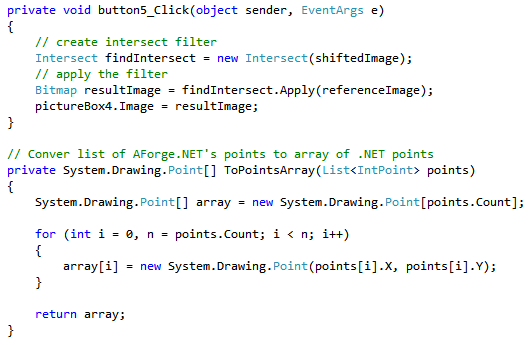
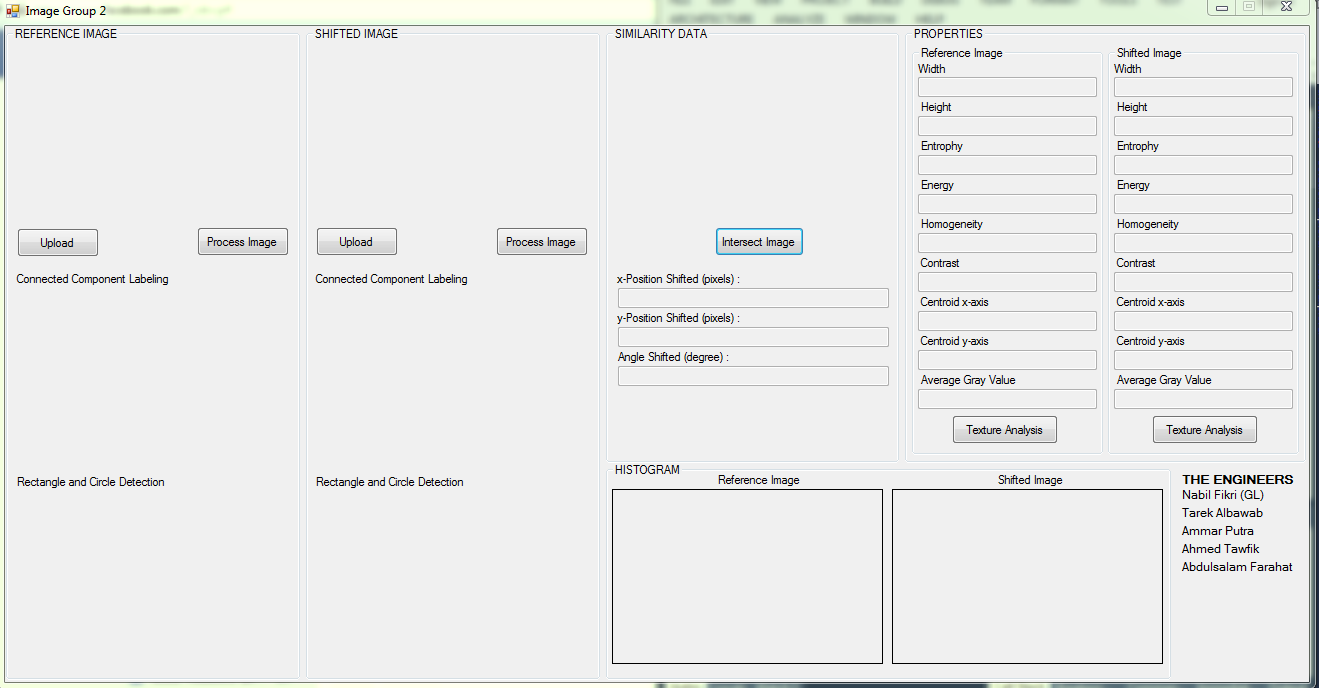


Figure 2.8 Intersect the Image Button

Figure 1.8 show the Intersect filter that we used to check the image result after we are trying to intersect them. If the image turns out with very smooth rectangle and circle, we can consider the image to be accepted and vice versa.

**3.0 RESULTS AND DISCUSSION**

When starting the program, a graphical user interface (GUI) will start (Fig.4.1).

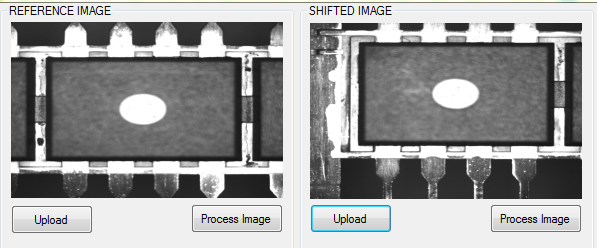


**Fig.3.1**

Press Upload button to upload the images that you want to start the processing (Fig.4.2).

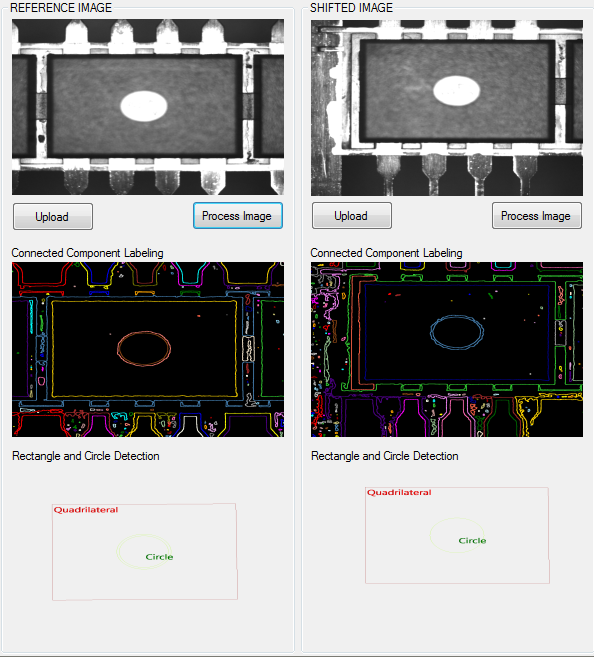
Two kinds of images, Reference Image and Shifted Image. The image that you want to set it as the main image for your processing, is called the Reference image.

The image that you want to be processed and compared with the Reference image is called Shifted Image.



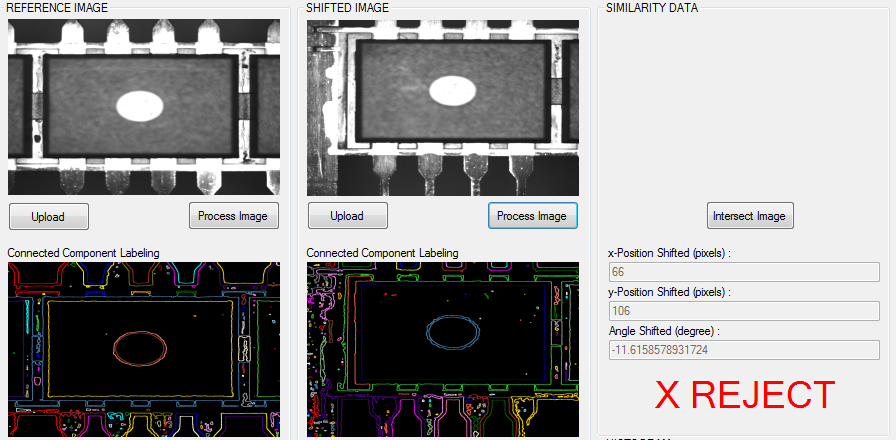
**Fig.3.2**

If Process Image button is pressed, the image will be processed, the processing will detect the Rectangle and Circle in the image (Fig.4.3).

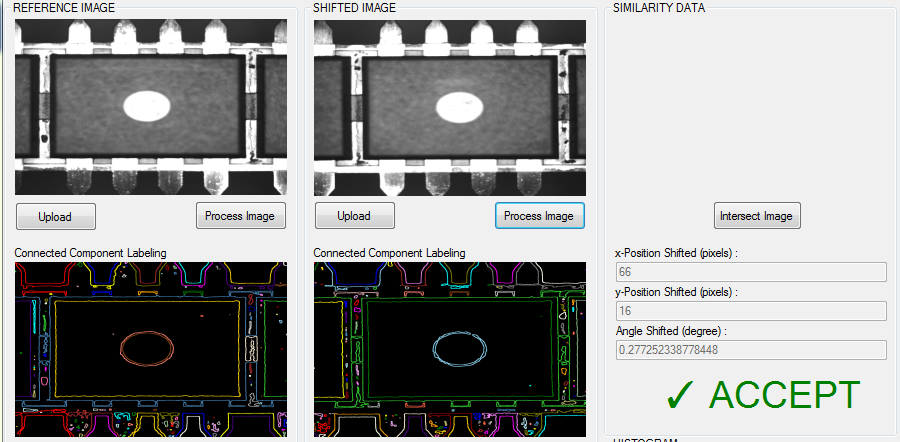


**Fig.3.3**

Furthermore, the process image will show if the Shifted Image is accepted or rejected with respect to the Reference Image (Fig.4.4) & (Fig.4.5).



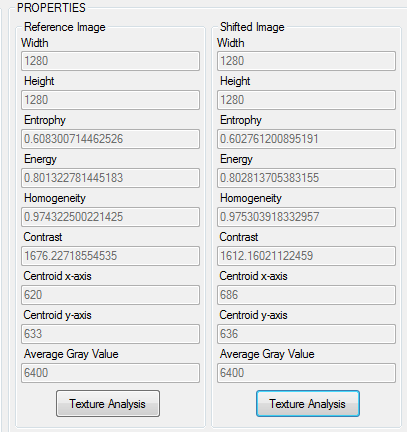
**Fig.3.4**



**Fig.3.5**

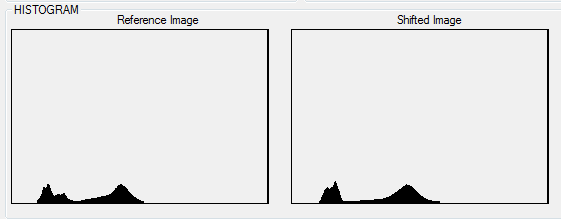
When the Texture Analysis button is pressed, properties of the photos will be shown.

Properties such as: Entropy, Energy, Homogeneity, Contrast, Centroid x-axis, Centroid y-axis, and Average Grey Value (Fig.4.6).



**Fig.3.6**

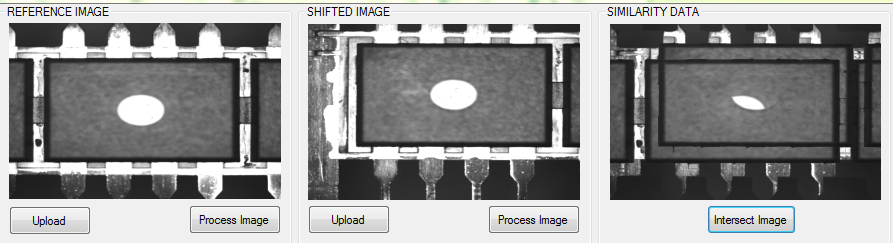
Program can also show the Homogeneity data (Fig.4.7).



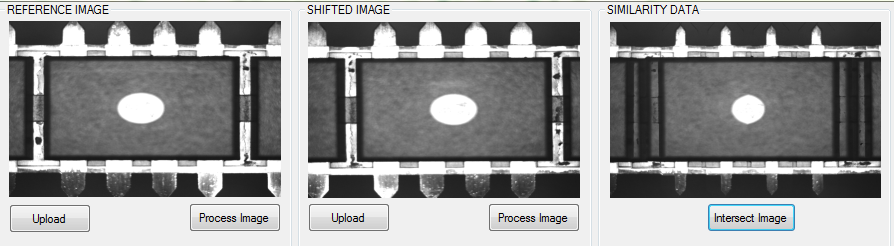
**Fig.3.7**

In addition to all the features that our program has, an extra feature is added to our program, which is Intersect image.

Intersect image basically shows how similar one image to another by intersect them in one image (Fig.4.8) & (Fig.4.9).



**Fig.3.8**



**Fig.3.9**

**4.0 CONCLUSION**

Our program gives the ability to a normal camera, to be smart. Writing the codes to enable the processing, which will therefore give the brain to the processor to process real world conditions and generate data and makes actions out of it. In other words, we have achieved the Machine Vision, by giving the Vision to the machine, to compare and analyse between different given data.

We have successfully achieve the objective of this assignment, by detecting the four sides edges with forming a rectangle, calculating the different in position and angle shift respect to reference image, performing contrast analysis and shape recognition on bright round area, further more adding another feature as our enhancement which is the Similarity Data.

**5.0 REFERENCES**

1. AForge.NET :: Framework Features. (n.d.). Retrieved from <http://www.aforgenet.com/framework/features/> (September 14, 2015)
2. Adrian. Calculating the angle between two points on a circle. (2009, March 23). Retrieved from <https://beradrian.wordpress.com/2009/03/23/calculating-the-angle-between-two-points-on-a-circle/> (September 14, 2015)
3. Souza, C. (n.d.). Primary objects/Accord.NET. Retrieved from <https://github.com/primaryobjects/Accord.NET/blob/master/Sources/Accord.Imaging/GrayLevelCooccurrenceMatrix.cs> (September 14, 2015)