Design Features Recognition Using Image Processing Techniques

Sreenivasulu Reddy, Poornachandra Sekhar, Hitheshwar Naik

Department of Mechanical Engineering, Sri Venkateswara University College of Engineering, Tirupati – 517502;
Andhra Pradesh; INDIA.
Email: seetharamadasubcm@gmail.com, nlpoorna321@gmail.com, pujari.hithesh@gmail.com

ABSTRACT: Design features refer as manufacturing information sets of shape related attributes of a work part. Features recognition is the most relevant methods of image analysis. In this paper, an ideal method is developed to extract and recognize different shape features using digital image processing techniques. The geometric data extraction algorithm is developed with SOBEL, CANNY Edge detection algorithms for features extraction and MINBOUNDSUITE algorithm for features recognition. The methods involved are colored (RGB) image to black and white image (Binary) conversion, boundary and edge detection, area based filtering, use of bounding box and its properties for calculating object metrics. The algorithm was developed in MATLab package.

Keywords : Feature extraction and recognition, Image processing techniques, geometric data extraction, SOBEL edge detection algorithm, Canny edge detection algorithm, MINBOUNDSUITE algorithm, RGB image.

1 INTRODUCTION

Design feature extraction and recognition information provides a connection between CAD and CAPP systems in the automated shop floor [1]. Developing such a system from a work part using image processing techniques is the present work. Image is a collection of points in a matrix format. Every point has some relation with other points nearby and all have different values [3]. Image can be used to recognize the object, or in some cases, compare the object with a predefined, flawless and expected object [4]. Features of perceptual representation of an image are shape, corner, colour, texture, etc. Shape features are most important visual features among them. The field of digital image processing has found many applications in the field of automation, Design, Inspection, etc. An approach involving digital image processing and geometric logic for recognition of shape features in image such as squares, circles, rectangles and triangles has been discussed in this paper. Input images taken into MATLab are part drawings drawn in CATIA v5 software. Complete processing is done using MATLAB image processing functions, which are written in the form of codes. The accuracy of the system largely depends upon the quality of visual features used to represent image information.

2 LITERATURE REVIEW

Automated feature recognition has been an active research area in solid modeling for past decade and is considered to be a critical component for integration of CAD and CAM fields. Design features recognition is one of the most relevant applications of image analysis [5]. Lot of research is still in progress to obtain a best method to perfectly recognize features from an image. A simple search with the phrase “Features Recognition” in the IEEE digital library throws 9422 results and about 1332 articles published in only one year 2014. In this paper a program is developed using MATLAB functions in form of codes for the proposed feature extraction and recognition system. Where the part’s form features are identified and listed. These results may be applied as input information for the generation of alternative solutions in production planning operations.

3 METHODOLOGY

Here the complete concentration is on image conversion from RGB to Black and White (binary format), thresholding the image, and reorganization of boundaries of objects etc. Fig.1 shows the Flow chart for Methodology.

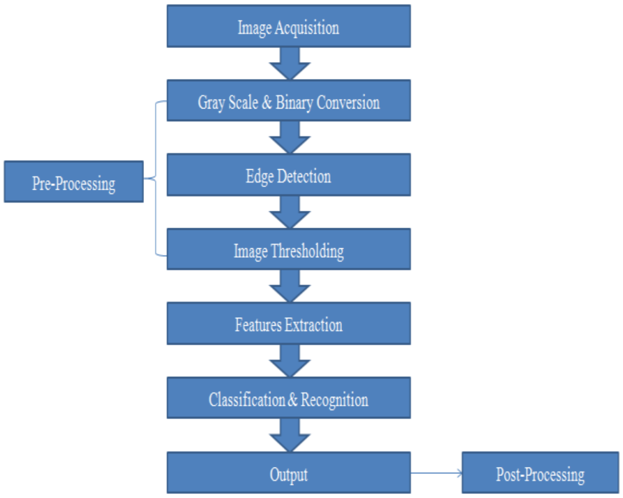


Fig.1 *Flow chart for Methodology*

3.1 Procedural Steps in Detail:

- Read/capture image: Image is acquired from existing files from the memory. Consider the acquired image be in RGB format which is a true color format for an image in MATLAB.
- Converting RGB image to Binary format: This process is done in two steps. The RGB image is first converted to a two dimensional grayscale image. This gray image is then converted to black and white (binary) image. A threshold is set and the luminance of each pixel is compared with this threshold. All values that are greater than this threshold are replaced with a logical one (white) and the values below this threshold are replaced by a logical zero (black).
- Thresholding the image: Thresholding the image is an important step in features extraction. Sobel and Canny

edge detector algorithms are employed at this stage. Removing noise, smoothening and effective edge detection is carried out in this step.

- Recognize boundaries of objects: Boundaries of the objects are recognized by first setting a single pixel on the object-background interface as a starting point and moving in a clockwise or counter-clockwise direction and searching for other object pixels. The pixels may be searched either diagonally (in 8-connected pixels) or edge-adjacent pixels (in 4-connected pixels). By hunting for object pixels in a fixed direction, the object's boundary can be recognized.
- Finding areas of objects and area filtering: Once the object boundaries have been recognized, the area of that object can easily be calculated by summing the number of pixels within the boundary extent with the help of following MATLab function "regionprops".
- Finding bounding box of the object: The bounding box of an object is an imaginary rectangle that completely encloses the given object. Below Fig.2 illustrates the concept of a bounding box.

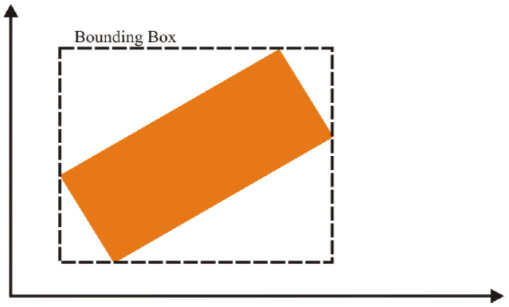


Fig.2 *Bounding box for a rectangle*

- Finding ratio of areas for a circle: The ratio of the area of an object to the area of its bounding box is known as the Extent (1) and is a very useful parameter in MATLab. Finding extent is to recognize features in the image with some geometric logic application. To classify a square and a rectangle whose extent values will be same, another parameter called AspectRatio is calculated. Aspect ratio is the ratio between width and height of the bounding box. This ratio will not be same for both the features.

$$Extent \quad = \quad \frac{Areaof\ the\ object}{Areaofboundingbox} \quad - (1)$$

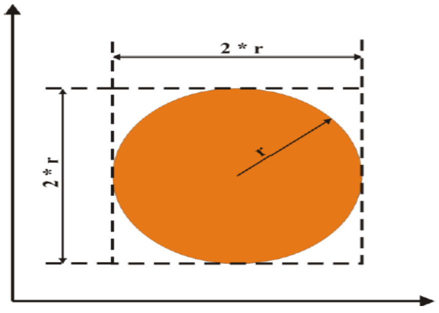


Fig.3 *Bounding box dimensions for a circle*

3.2 Image processing algorithms employed:

The purpose of edge detection in general is to significantly reduce the amount of data in an image, while preserving the structural properties to be used for further image processing. These algorithms include the Sobel, Prewitt, Roberts, Canny, and Laplacian of Gaussian methods. Sobel and Canny edge detection algorithms are employed in the present work for features extraction. For recognition of features Minboundsuit algorithm is applied.

3.3Sobel edge detector algorithm:

- The Sobel Algorithm isused in image processing and computer vision, particularly for edge detection of an image which emphasizes edges and transitions [6]. It is named after Irwin Sobel, who presented the idea of an "Isotropic 3ximage gradient operator" at a talk at the Stanford Artificial Intelligence Project (SAIP) in 1968.
- Sobel detector is employed by most operators because of its simplicity and common use.
- Sobel detector performs a 2-D spatial gradient measurement on images which enhances the removal of redundant data; as a result, unnecessary amount of data is removed for further edge detection.
- The Sobel detector is incredibly sensitive to noise in pictures, it effectively highlight them as edges. Hence, Sobel operator is less recommended, but stands as a standard detector for further detection.

Syntax: *BW = edge(I,'sobel');*

3.4 Canny edge detector algorithm:

- The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986 [7].
- Flow chart for canny edge detection algorithm in Fig.4 explains the procedure of edge detection.
- It is the most powerful edge-detector. This differs from the other edge-detectors that it uses two different thresholds (to detect strong and weak edges), and includes the weak edges in the output only if they are connected to strong edges. This method is therefore less likely than the others to be fooled by noise, and more likely to detect true weak edges [8].

Syntax: *BW = edge(I,'canny');*

Algorithm Steps:

- Step-1: Convert to gray scale and then to binary format.
- Step-2: Smoothening the input image with any filters.
- Step-3: Thresholding the image

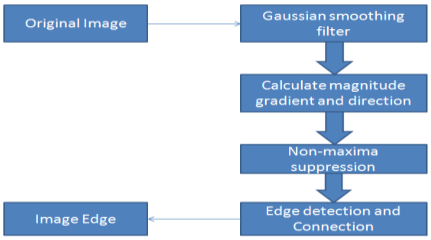

Original Image
Gaussian smoothing filter
Calculate magnitude gradient and direction
Non-maxima suppression
Edge detection and Connection
Image Edge

Fig.4 *Flow chart for canny edge detection algorithm*

3.5 Minboundsuit algorithm:

This technique comes under image segmentation phenomenon. Segmenting the data needed is very important in features recognition. Minboundsuit algorithm provides few techniques like bounding circle, bounding square and bounding rectangle or bounding box [9]. In the present context, bounding box is employed.

- Bounding Box:

It consists of a rectangle with horizontal and vertical sides that encloses the region and touches it's topmost, bottommost, leftmost, and rightmost points. As shown in Fig.5 below, there can be as many as eight distinct extremal pixels to a region: topmost right, rightmost top, rightmost bottom, bottommost right, bottommost left, leftmost bottom, leftmost top and topmost left. Each extremal point has an extremal coordinate value in either its row or column coordinate position. Each extremal point lies on the bounding box of the region.

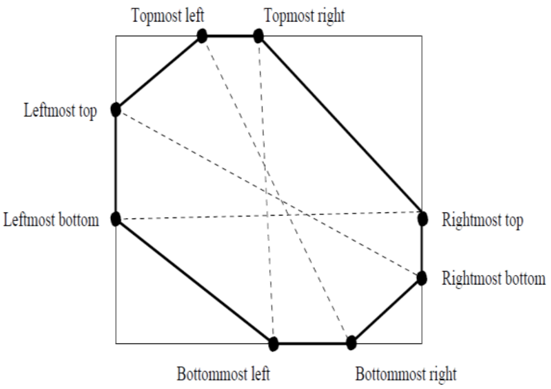


Fig.5 *Bounding box applied for a typical Boundary*

4 CASE STUDY

Part drawing drawn using CATIA software is taken as case study, whose features need to be recognized. Shown below Fig.6 is the part drawing drawn in CATIA for mentioned purpose.

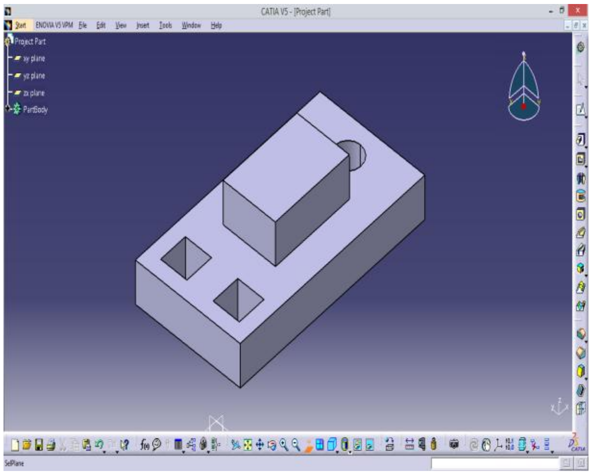


Fig.6 *Example part drawing for case study*

The entire block dimensions are 150x80x40 mm^3. Fig. 7 below shows the example part with dimensions of each feature.

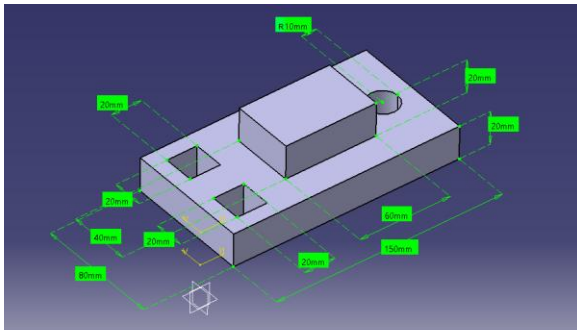
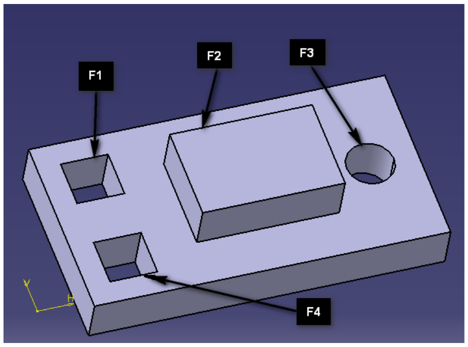


Fig.7 *Part drawing with Dimensions*

Example part drawing with features shown with arrows on the part itself and form features listed sideways is shown in below Fig.8.

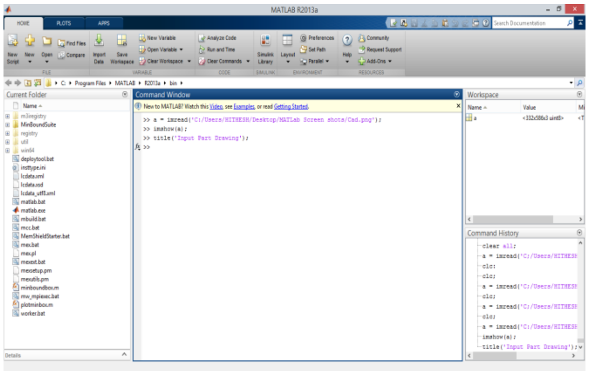


FEATURES LIST:

F1 = SQUARE
F2 = RECTANGLE
F3 = CIRCLE
F4 = SQUARE

Fig.8 *Part drawing with Form Features listed*

As mentioned in the methodology, complete programming is done in MATLab using functions in the form of codes. Fig.9 shows the home page of MATLab package. Command window, workspace window and command history window are the major operational windows in home page of MATLab. Complete programming methodology as explained in procedural steps like *Read/capture image, Converting RGB image to Binary image, Thresholding the Image, Recognize boundaries of objects, Finding areas of objects and area filtering, Finding bounding box of the object, Finding ratio of areas for given object*, classification of features based on object metrics calculated applying geometric logic and displaying the recognized features on the image itself is done accordingly.



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
>> a = imread('C:/Users/HITHESH/Desktop/MATLab Screen shots/Cad.png');
>> imshow(a);
>> title('Input Part Drawing');
>>
```

Name	Value
a	<332x586x3 uint8>

```
clear all;
a = imread('C:/Users/HITHESH
clc;
clc;
a = imread('C:/Users/HITHESH
clc;
a = imread('C:/Users/HITHESH
clc;
a = imread('C:/Users/HITHESH
imshow(a);
title('Input Part Drawing');
```

Fig.9 *MATLAB home page with input image read function called*

As mentioned in the methodology, RGB image conversion to binary image format is done in two steps i.e., The RGB image is first converted to a two dimensional grayscale image. This gray image is then converted to black and white (binary) image. This is done using following MATLab functions: *a1 = rgb2gray(a); BW = im2bw(a1);* results are shown in below figures (Fig.10 and Fig.11).

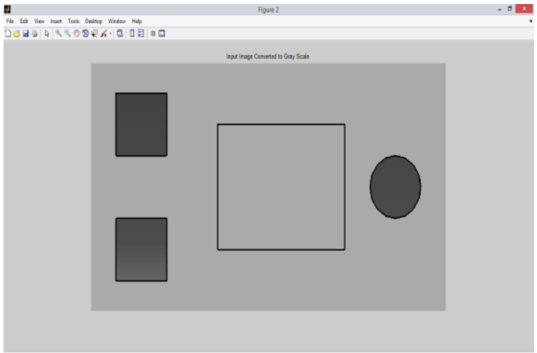


Figure 2
File Edit View Insert Tools Desktop Window Help
Input Image Converted to Gray Scale

Fig.10 *Input image converted to Gray scale format*

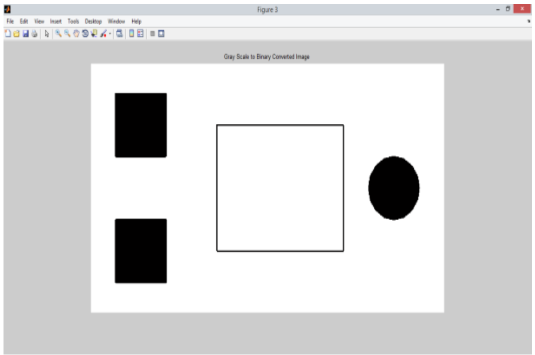


Figure 3
File Edit View Insert Tools Desktop Window Help
Gray Scale to Binary Converted Image

Fig.11 *Gray scale image to Binary format*

After converting input image to binary format, next step is to apply Sobel and Canny edge detection algorithms respectively for extraction of features from binary image. The result after applying Sobel edge detection algorithm is shown below Fig.12.

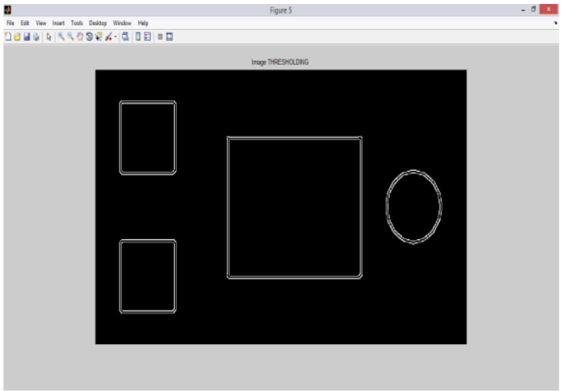


Fig.12 *Resu7lt window after applying Sobel algorithm*

For removing noisy edges and to extract the features perfectly canny algorithm is applied and Fig.13 below shows the features extracted window.

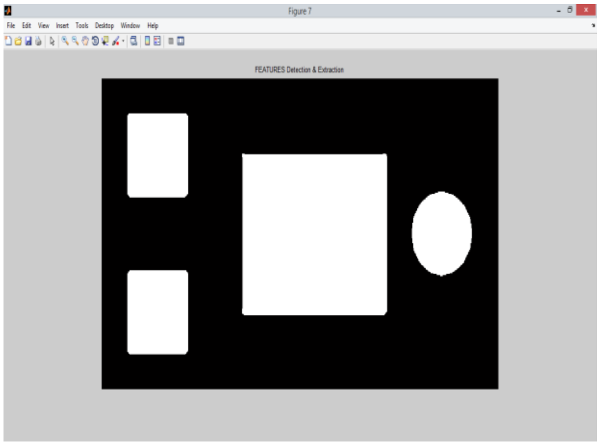
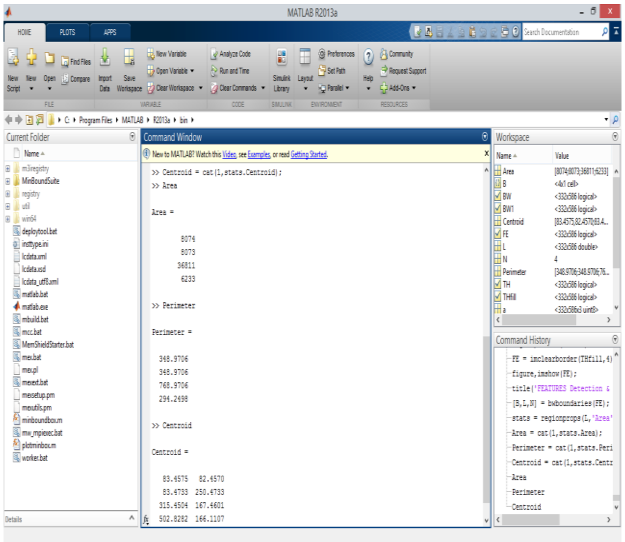


Figure 7
File Edit View Insert Tools Desktop Window Help
FEATURES Detection & Extraction

Fig.13 *Features extracted after applying Sobel and Canny algorithms*

Finding out the region properties for calculating various object metrics is an important step in the methodology. The '*regionprops*' function does the trick. Below Fig.14 shows region properties like Area, Perimeter and Centroid values calculated.



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
>> Centroid = cat(1,stats.Centroid);
>> Area

Area =

        8074
        8073
       36811
        6233

>> Perimeter

Perimeter =

  348.9706
  348.9706
  768.9706
  294.2498

>> Centroid

Centroid =

   83.4575   82.4570
   83.4733  250.4733
  315.4504  167.4601
  502.8282  166.1107
```

Name	Value
Area	[8074;8073;36811;6233]
B	<4x1 cell>
BW	<332x586 logical>
BW1	<332x586 logical>
Centroid	[83.4575,82.4570;83.4...
FE	<332x586 logical>
L	<332x586 double>
N	4
Perimeter	[348.9706;348.9706;76...
TH	<332x586 logical>
THfill	<332x586 logical>
a	<332x586x3 uint8>

```
FE = imclearborder(THfill,4)
figure,imshow(FE);
title('FEATURES Detection &
[B,L,N] = bwboundaries(FE);
stats = regionprops(L,'Area'
Area = cat(1,stats.Area);
Perimeter = cat(1,stats.Peri
Centroid = cat(1,stats.Centr
Area
Perimeter
Centroid
```

Fig.14 *Region properties calculated*

After finding region properties, as mentioned in methodology bounding box for each object is found i.e., minboundsuite algorithm is applied for features recognition. Extent values of each feature need to be found, to classify the recognized features. The extent value for square and rectangle is same. So to recognize rectangle feature, another parameter called Aspect ratio is found. It is the ratio of width to height of bounding box. "RGB = label2rgb(L); imshow(RGB);" function is called to display the output image. Object metrics with which geometric logic is applied to recognize different features are combined using Combined=[]; function. Different colors indicate the variations in intensity values of the input image. Final output image is shown in Fig.15 with its different features labeled on the image itself.

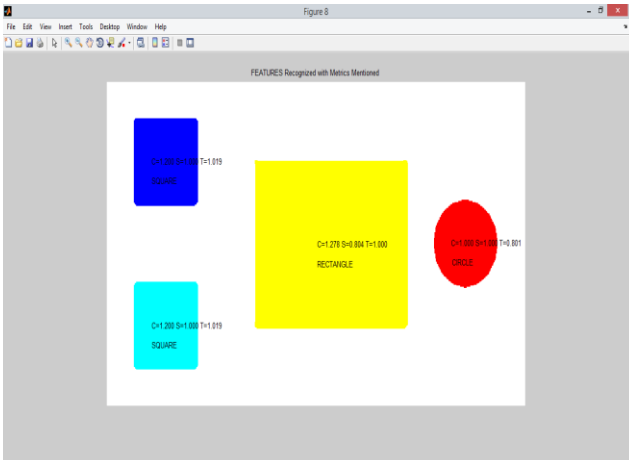


Fig.15 *Final output of developed recognition system*

5 RESULTS AND DISCUSSIONS

Table-1 below is the results from MATLab final output screen. Different features are recognized based on their object metrics (Circular Metric, Square Metric and Triangle Metric) with geometric logic applied.

Table-1: *Recognized features with their metric values*

Feature No.	Circular Metric	Square Metric	Triangle Metric	Recognized Feature
F1	1.200	1.000	1.019	Square
F2	1.278	0.804	1.000	Rectangle
F3	1.000	1.000	0.801	Circle
F4	1.200	1.000	1.019	Square

To recognize circular feature, geometric logic applied is: *CircleMetric=(Perimeter.^2)./(4*pi*Area).*
When calculated, CircleMetric value will be 1 i.e., when circular metric value for any feature becomes 1, then the recognized feature will be circle. To classify a square and a rectangle whose Extent values will be same i.e., (area of object/area of bounding box) another parameter called AspectRatio is calculated. *SquareMetric(k) = aspectRatio;* this ratio will not be same for both the features. So when square metric value is other than 1, then that feature is recognized as Rectangle. Also when square metric value becomes 1, square feature is recognized. To recognize triangle feature, geometric logic applied is: *TriangleMetric(k)= Area(k)/boxArea;* When calculated its value falls under 0.6. So when Triangle metric value is <0.6, triangle feature is recognized. In the above Table-1all the features are recognized accordingly. Following are the MATLab functions used to classify the features based on geometric logic.

```
isCircle = (CircleMetric<1.1);
isTriangle = ~isCircle&(TriangleMetric<0.6);
isSquare = ~isCircle&~isTriangle&(SquareMetric>0.9);
isRectangle = ~isCircle&~isTriangle&~isSquare;
```

5.1 Features window for part drawing:

After recognizing all the features, dimensions and location of each feature in the example part are tabulated in below Table-2.

Table-2: *Results of part drawing*

Feature No.	Feature Name	Radius (mm)	Length (mm)	Width (mm)	Height (mm)	Location (X,Y)
F1	Square	-	20	20	20	39,40
F2	Rectangle	-	60	40	20	208,81
F3	Circle	10	-	-	20	459,161
F4	Square	-	20	20	20	39,209

These results of the Case study may be applied to Production planning problems.

6 CONCLUSIONS

In the present paper, a prototype of design features extractor and recognizer is developed using image processing techniques to recognize various shape features in a part drawing developed using CATIAv5 software.

- The proposed feature recognition system is able to find the shape features like Circle, Rectangle, Square and Triangle.
- The results of the feature recognition system contain the information such as feature types with object Metrics and their location which may be applied as input to Production Planning problems.

The applications are not only restricted to design, Production processes field but can extend to fields like artificial robotic intelligence, bio-medical field, computer vision, recognition of vehicles at toll booths or traffic signals for traffic status determination, etc.

7 REFERENCES

[1] Huang Z, Yip-Hoi D. (2002), High-level feature recognition using feature relationship graphs. *Computer Aided Design*, No. 34, pp. 561–582.

[2] Emad S. Abouel Nasr and Ali K. Kamrani. (2006), A new methodology for extracting manufacturing features from CAD system, *Computers & Industrial Engg*, No. 51, pp. 389–415.

[3] Shalinee Patel, Pinal Trivedi, and Vrundali Gandhi, “2D Basic Shape Detection Using Region Properties”, International Journal of Engineering.

[4] Du .C.-J. and Sun .D.-W, “Recent developments in the applications of image processing techniques for food quality evaluation”, Trends in Food Science & Technology 15 (2004), 230–249.

[5] Kenneth. R. Castleman, Z. G. Zhu. "Digital Image Processing". Publishing House of Electronics Industry, Beijing, 1999.

[6] Irwin Sobel, 2014, "History and Definition of SOBEL Edge Detection Algorithm".

[7] Canny, J., "A Computational Approach to Edge Detection", IEEE Trans. Pattern Analysis and Machine Intelligence, 1986.

[8] Moeslund, T. (2009, March 23). "Canny Edge Detection". Retrieved December 3, 2014

[9] Freeman, R. Shapira, "Determining the Minimum-Area Encasing Rectangle for an Arbitrary Closed Curve", Communications of the ACM Vol. 18 No. 7 (1975).

[10] Zhang K.F, Wright A.J. and Davies B.J. (1989), A feature-recognition knowledge base for process planning of rotational mechanical components, *International Journal of Advanced Manufacturing Technology*, Vol. 4, pp.13–25.

[11] Shah, J. J., Mantyla, M. and Nau, D., Advances in Feature Based Manufacturing. Elsevier/North-Holland, Amsterdam, 1994.

[12] Proyecto Fin de Carrera, "Feature Recognition Algorithms" June 16, 2010

[13] Meisel .W, "Computer-Oriented Approaches to Pattern Recognition". Academic Press. New York, 1972.

[14] Zhou, P., Ye, W., & Wang, Q. (2011). "An Improved Canny Algorithm for Edge Detection". Journal of Computational Information Systems