**Pattern Recognition**

**Project Report**

**Group 2**

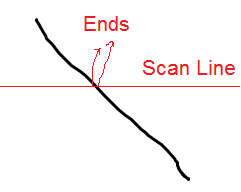
**Team Members**

1. Ayman Mohamed.
2. AbdelRahman Ahmed.
3. AbdelRahman Fawzy.
4. AbdelRahman Nasr.

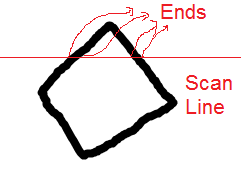
**Used Features**

**Number of horizontal & vertical ends**

We are using the number of horizontal ends as it differentiate between the lines as each line has max distribution of 2 ends per horizontal scan line, as well as the vertical scan line as shown in figure.

  
Figure: Scan Line and the Number of Ends in Line image

Where in Diamonds and Ellipses there are more than 2 ends per horizontal or vertical scan line as shown in the figure.

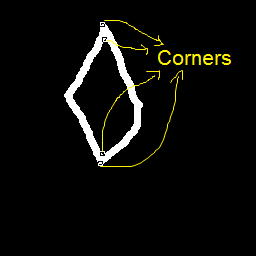
  
Figure: Scan Line and the Number of Ends in Diamond Image

**The number of Corners**

The number of corners is another feature that differentiates between Diamond & Ellipse with high accuracy, we used the Curvature Scale Space (CSS) Method to detect corners for the image, and the steps of CSS Method are:

1. Extract the edge contours from the input image.
2. Fill Small Gaps in edge contours.
3. Compute curvature on the edge contours at high scale.
4. The corner points are defined as the maxima of the absolute curvature above a threshold.
5. Track the corners through lower scales to improve localization.
6. Compare gaps filled with other corners and remove very close corners.

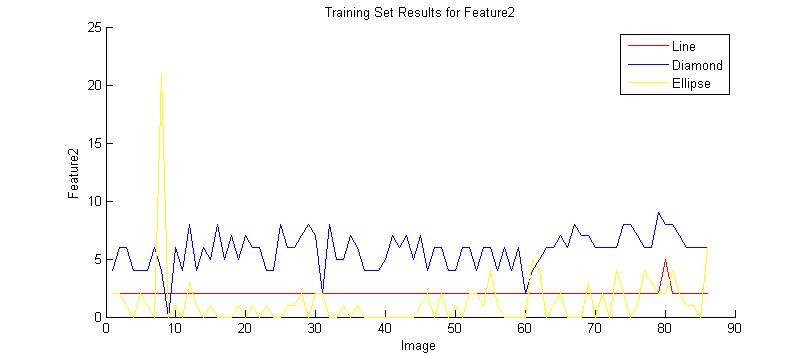
We used a ready coded function composed by “He Xiaochen” from HKU EEE Dept. ITSR, with canny edge detector and extracting curves before computing smoothed curvature using Gaussian patch and getting the maxima.

  
Figure: the output Corners using Canny filter with Hi = 0.35 & Low = 0,   
and using Gaussian filter with patch of size 5x5 with sigma = 7.6,   
the algorithm is settled by maximum corner angle = 130  
and 1.5 difference between the large & small axis.

The 2 Features defines good decision regions for each image as shown in figures.

`





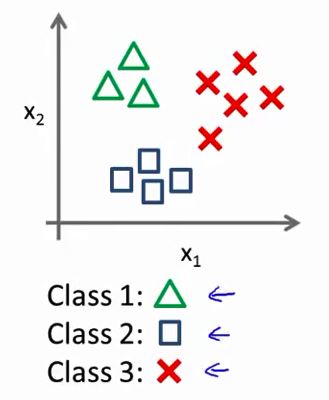
**Used Classifiers**

**Logistic Regression Classifier**

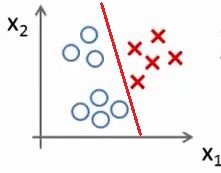
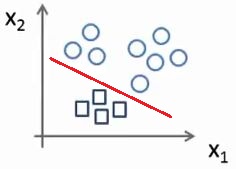
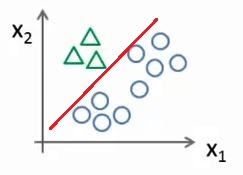
We used logistic regression one vs. all method to train k classifiers, each classifier predicts the probability of a training example belonging to a specific class, and we take the classifier with the highest probability as the matching class.

The Classifier maps the 2 features to polynomial features used in the regularization.

**Example:**



* First we have multiple classes and want to design a classifier for each class like in the previous image.



* Then we take each class and design its own classifier (learning parameters) with respect to the other classes.
* Then we take the class which outputs the highest probability for the training pattern as the correct class.

**Neural Network Classifier**

We tried to use Neural Network of 1 level 25 nodes Network but it results 95.5% accuracy for the Training set while it results 90.9% accuracy for the validation set that makes the Logistic Regression better than Neural Networks.

**Validation Results using Logistic Regression**

|  |  |  |
| --- | --- | --- |
|  | Training Set Accuracy | Validation Set Accuracy |
| validating with 1st 22 images per class | 95.833333% | 98.484848% |
| validating with 2nd 22 images per class | 96.875000% | 96.969697% |
| validating with 3rd 22 images per class | 96.875000% | 96.969697% |
| validating with 4th 22 images per class | 97.916667% | 93.939394% |

**Total Results**

Training Set Accuracy: 96.875000%

Validation Set Accuracy: 96.590909%