PATTERN RECOGNITION FINAL PROJECT GESTURE RECOGNITION

CS-669

3rd December 2012

Boga Saikiran

B10010, Computer Science and Engineering

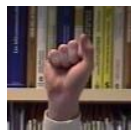
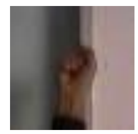
Vihari Piratla

B10025, Computer Science and Engineering

INTRODUCTION

Data Sets:

Initially data sets were procured from http://www.idiap.ch/scientific-research/resources/hand-posture-and-gesture-datasets/

The images procured were very challenging for skin detection technique that we have employed, but demands a much computationally intensive techniques like Haar classifier. This may not be the case in the real world where a person may try until his hand is detected and hence we have decided to make a dataset closer to what persists in the real-world.

We have created our own image data sets by capturing frames through webcam.

Methods Used:

- Support Vector Machines.
- K-Means.
- Canny Edge Detection.
- Gaussian Kernel Smoothening of Image.
- Active Contours.

PURPOSE:

The purpose of the project is to detect and classify gestures in real time with no constraints on the image sensor or computational device. Care is taken to make it computationally feasible.

EXPERIMENTS:

Initial datasets collection and feature extraction:

1. Image data sets of seven different hand gestures are collected from the frames of the videos taken through webcam. These data sets are taken each as a different class.
2. The images sets (rgba images) collected are converted into hsv images, which are more robust in lightening conditions, skin color quality and are more useful in gesture recognition type of problems.
3. These hsv images obtained are then grey scaled using threshold values of h, s, and v.
4. Those grey scale images are then smoothened for noise reduction.
5. Then edge detection is performed, which detects hand from rest of the image, and then contour is plotted around the detected edge.
6. The two dimensional points on the contour are extracted as the features of that particular image.

Gesture Recognition:

Training phase:

1. K-means clustering is performed on each class.
2. The output obtained in K-means clustering is given to SVM for training and a model is created on which testing can be done.

Testing phase:

1. Features are extracted from the images/frames obtained.

2. These feature vectors are tested with the model built with SVM and the frame is classified as belonging to on one of the classes on which SVM is trained.

Assumptions:

1. Hand is the biggest object of that particular color in the image.
2. Some general lightening conditions are assumed.

RESULTS:

- Experiments were made with various clusters of K means and also with various kernel types.
- We have considered 7 symbols in total (see the images in for more details).

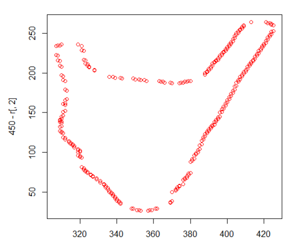


Fig a

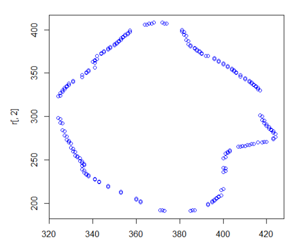


Fig b

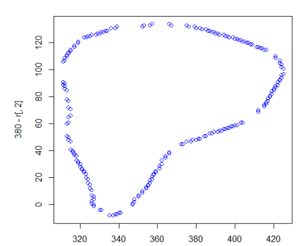


Fig c

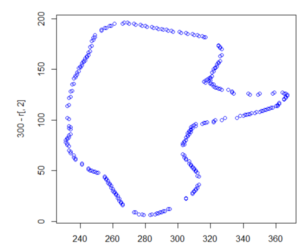


Fig d

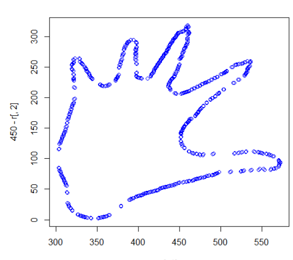


Fig e

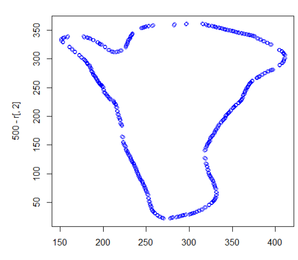


Fig f

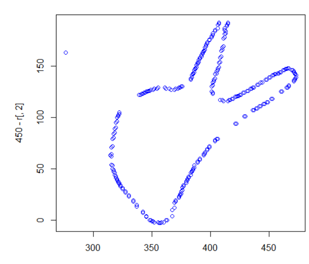


Fig g

- We experimented with 2 different values for number of clusters
  - Clusters of size 30 and 50 are considered.

- We have also experimented with different types of kernels:

  a) Linear kernel function
  b) Polynomial kernel function and
  c) Sigmoidal kernel function.
- It is observed that best performance was seen in the case of linear kernel.

Number of clusters: 50

<u>With flutter kind of symbols considered (Fig b,d,f)</u>

Liner kernel:

Accuracy = 91.5156% (1262/1379)

Confusion Matrix: $\begin{matrix} 105 & 5 & 8 \\ 13 & 629 & 39 \\ 5 & 47 & 528 \end{matrix}$

rbf kernel:

Accuracy = 76.9398% (1061/1379)

Confusion Matrix: $\begin{matrix} 98 & 606 & 120 \\ 0 & 0 & 0 \\ 25 & 75 & 455 \end{matrix}$

Polynomial kernel:

Accuracy = 50.2538% (693/1379)

Confusion Matrix:

$$\begin{matrix} 123 & 681 & 563 \\ 0 & 0 & 0 \\ 0 & 0 & 12 \end{matrix}$$

<u>Open flutter (Fig. d,e,f):</u>

Polynomial kernel:

Accuracy = 48.2636% (681/1411)

Confusion Matrix:

$$\begin{matrix} 681 & 154 & 576 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$$

Linear kernel:

Accuracy = 84.7626% (1196/1411)

Confusion Matrix:

$$\begin{matrix} 618 & 33 & 100 \\ 1 & 103 & 1 \\ 62 & 18 & 475 \end{matrix}$$

rbf kernel:

Accuracy = 70.8009% (999/1411)

Confusion Matrix:

586	91	190
0	27	0
95	36	386

Total for all the 7 classes:

Linear kernel:

Accuracy = 73.9515% (1587/2146)

Confusion Matrix:

152	10	2	31	7	19	24
1	57	1	2	0	1	3
2	3	58	2	2	3	0
45	27	6	574	26	86	36
1	0	1	1	101	2	2
46	13	1	49	10	447	13
22	12	0	22	8	17	198

Number of clusters: 30

Total for all the 7 classes:

Linear kernel:

Accuracy = 84.8555% (1821/2146)

	195	9	1	22	9	9	21
	2	96	0	0	0	1	1
	0	1	63	1	1	2	0
Confusion Matrix:	33	6	5	612	20	34	24
	2	1	0	4	115	0	4
	15	4	0	31	4	524	10
	22	5	0	11	5	5	216

Polynomial kernel:

Accuracy = 31.7335% (681/2146)

	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
Confusion Matrix:	269	122	69	681	154	575	276
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0

rbf kernal:

Accuracy = 58.6207% (1258/2146)

100	18	8	19	13	17	41
0	0	0	0	0	0	0
0	0	0	0	0	0	0
67	34	48	578	55	137	64
0	0	0	0	26	0	0
45	24	13	68	25	420	37
57	46	0	16	35	1	134

ROC curve: