Chapter 7

Result and Discussion

Here we describe the experiments under different schemes as mentioned above. The four classification schemes used are:

i. multiple kernels learning (MKL)
ii. support vector machines (SVM)
iii. closest neighbor (NN) classification utilizing c2 measurement as a closeness degree and
iv. Histogram of Oriented Gradients (HOG).

The following outputs have been obtained using a sample natural scene image. The pictures of the output are given in sequential order.

1. **Scene Captured from Nature-** In this we have captured the natural image from the sign board, poster and pamplets in Kannada and given the input with the original input image with text as displayed in the Figure7.1.

2. **Conversion of RGB to Gray Scale Image-** After feeding of the image then the image is converted from RGB to Gray Scale, where colored image is converted to Gray Scale. In this image text becomes gray as shown in Figure7.2.

3. **Calculating Threshold of Image-** After converting the image into Gray Scale here, we are calculating Threshold value for the Images .Each image will have it's own threshold value and that will be displayed in the image as shown in the Figure7.3.

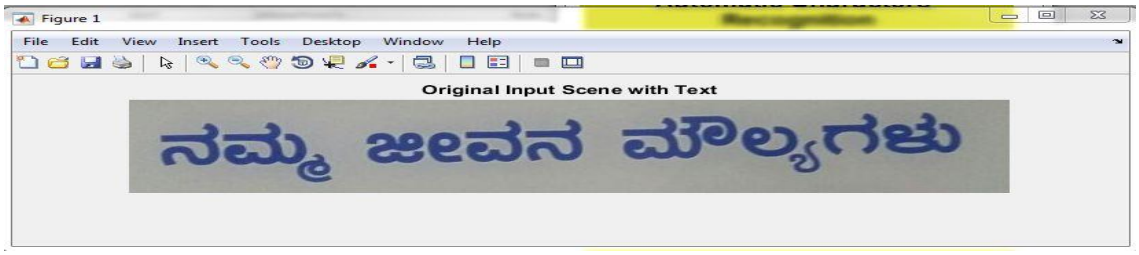

Figure 7.1: Original Image from Sample Database

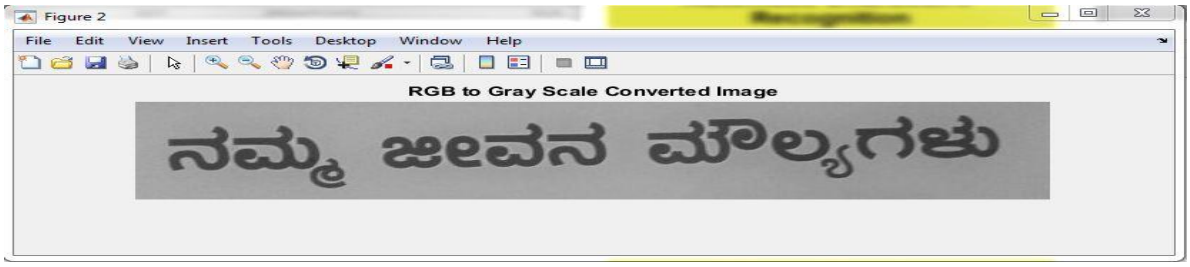

Figure 7.2: Converting RGB to Gray scale Image

Figure 7.3: Calculating Threshold of Image

4. **Canny Edge Detection-** For segmentation of a particular text in the image we have used Canny Edge Detection Algorithm. This algorithm is used for detection of the boundary of the text given as input, it detects both inner as well as outer boundary of the text as shown in Figure7.4.

5. **Removal of Unwanted Objects with Lesser Pixels-** After detection of the text boundary, then unwanted objects in the image is removed with less than 10 pixels. Also, if noisy image is there then the noisy part of the image is removed and image is smooth and clear as shown in Figure7.5.

6. **Feature Extraction Using – HOG-** Feature Extraction is used for the extracting of text using boundary box technique. Each text is bounded specifically and extracted as shown in Figure7.6 and after that HOG(Histogram Oriented Gradient) is obtained for the text in the image where text part is there it is shown with bullets and for space it is shown with dot as shown Figure 7.8.

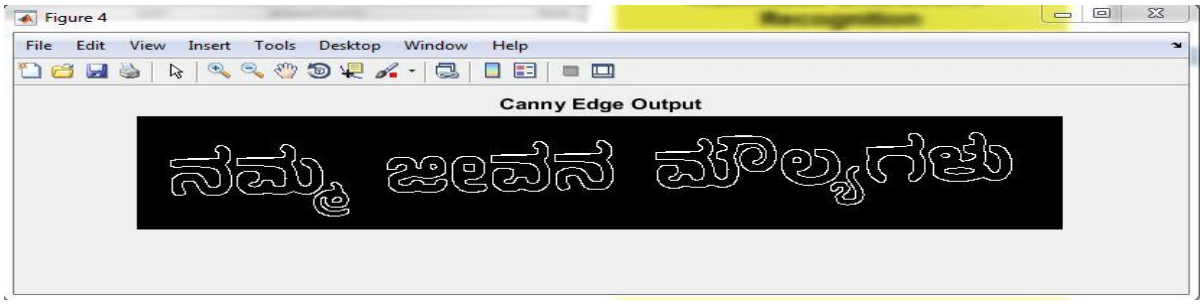

Figure 7.4: Canny Edge Detection

Figure 7.5: Removing Unwanted Objects less than 10 pixels

Figure 7.6: Feature Extraction

Figure 7.6.1: Feature Extraction – HOG

7. **Training and Testing-** After Segmentation of the Image with Various Steps then comes to the Training and Testing phase. Training is done for all 251 datatset we have taken where positive and some negative datasets are there. Training can be done once for a particular image. Through, training we get to know about efficiency of the dataset matched with 251 characters as shown in Figure7.7.

Testing is done after training, testing is done line by line whether it has single or multiple line text as input and then text is extracted using boundary box technique. In this every text is extracted with boundary and all the characters of the text are detected in the segment as shown in Figure7.8. This all process is a step wise process with different steps involved in it and all the process is shown in the Figure7.8.1.

```
Computing descriptors for 251 training windows:   251

Training linear SVM classifier...
 Iteration    FunEvals     Step Length     Function Val      Opt Cond
         1           2     2.41653e-03     7.71065e-01     3.57309e+02
         2           3     1.00000e+00     8.29996e-02     2.10283e+01
         3           4     1.00000e+00     7.86730e-02     1.58662e+01
         4           5     1.00000e+00     7.01067e-02     1.26109e+01
         5           6     1.00000e+00     6.25725e-02     1.32498e+01
         6           7     1.00000e+00     5.59889e-02     6.99075e+00
         7           8     1.00000e+00     5.45956e-02     1.93974e+00
         8           9     1.00000e+00     5.43152e-02     1.88109e+00
         9          10     1.00000e+00     5.42197e-02     1.47152e+00
        10          11     1.00000e+00     5.41132e-02     2.59030e-01
        11          12     1.00000e+00     5.41052e-02     2.14562e-01
        12          13     1.00000e+00     5.40994e-02     1.30836e-01
        13          14     1.00000e+00     5.40953e-02     6.70054e-02
        14          15     1.00000e+00     5.40897e-02     1.24702e-01
        15          16     1.00000e+00     5.40791e-02     2.84086e-01
        16          17     1.00000e+00     5.40564e-02     5.05524e-01
        17          18     1.00000e+00     5.40010e-02     7.97585e-01
        18          19     1.00000e+00     5.38843e-02     1.22656e+00
        19          20     1.00000e+00     5.37165e-02     1.19194e+00
        20          22     1.67050e-01     5.36715e-02     1.32288e+00
        21          23     1.00000e+00     5.35028e-02     6.06172e-01
        22          24     1.00000e+00     5.34694e-02     1.39143e-01
        23          25     1.00000e+00     5.34663e-02     1.87681e-02
        24          26     1.00000e+00     5.34662e-02     9.18153e-03
        25          27     1.00000e+00     5.34662e-02     7.25667e-03
        26          28     1.00000e+00     5.34662e-02     9.40461e-04
Directional Derivative below TolX

Training accuracy: (249 / 251) 99.20%
```

Figure 7.7: Training

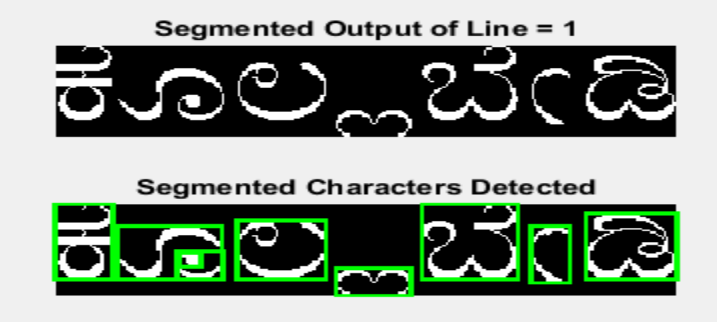

Figure 7.8: Testing

```
Searching 2096 detection windows...
  Image Scale 1.00, 384 windows - 384 matches total, 18.3% done
  Image Scale 0.95, 345 windows - 729 matches total, 34.8% done
  Image Scale 0.91, 273 windows - 1002 matches total, 47.8% done
  Image Scale 0.86, 240 windows - 1242 matches total, 59.3% done
  Image Scale 0.82, 209 windows - 1451 matches total, 69.2% done
  Image Scale 0.78, 153 windows - 1604 matches total, 76.5% done
  Image Scale 0.75, 128 windows - 1732 matches total, 82.6% done
  Image Scale 0.71, 105 windows - 1837 matches total, 87.6% done
  Image Scale 0.68, 84 windows - 1921 matches total, 91.7% done
  Image Scale 0.64, 65 windows - 1986 matches total, 94.8% done
  Image Scale 0.61, 48 windows - 2034 matches total, 97.0% done
  Image Scale 0.58, 33 windows - 2067 matches total, 98.6% done
  Image Scale 0.56, 20 windows - 2087 matches total, 99.6% done
  Image Scale 0.53, 9 windows - 2096 matches total, 100.0% done
  Image scale 0.51 is not large enough for descriptor, stopping search.
Image search took 5.52 seconds
```

Figure 7.9: Step wise process of Testing

Given below is the performance chart for 20 images that we tested with their result of the characters identified as shown in Figure 7.10.

Image ID	No. Of Characters	No. Of Characters Identified	Performance
1	6	3	50%
2	2	1	50%
3	4	4	100%
4	8	8	100%
5	7	7	100%
6	2	1	50%
7	49	49	100%
8	49	49	100%
9	3	0	0%
10	10	10	100%
11	14	14	100%
12	9	9	100%
13	5	5	100%
14	6	6	100%
15	4	4	100%
16	4	1	25%
17	10	10	100%
18	4	4	100%
19	7	0	0%
20	6	0	0%
TOTAL	209	185	88.52%

Figure 7.10: Performance Chart