Traffic Sign Classification

Sasank - 110070051, Tharun - 10070048, Rajeev - 110070055

November 8, 2014

1 Introduction

We classify traffic sign images from German Traffic Sign Recognition Database using different methods. We used following algorithms

- Linear Discriminant Analysis (LDA)
- Fisher's Linear Discriminant/Fisherfaces
- Random Forests

We applied these on images and histogram of oriented gradients descriptors.

2 Dataset

We used German Traffic Sign Recognition Database for training and testing our classifiers. The dataset features 43 different signs under various sizes, lighting conditions, occlusions and is similar to real-life data. Training set includes about 39000 images while test set has around 12000 images. Images are *not* guaranteed to be fixed dimensions and sign is *not* necessarily centred in each image. Each image contains a 10% border around the actual traffic sign. However, annotations are provided for bounding box if we wish to remove this border. More details on the dataset can be found in [1]

3 Classification using Images

In this section, we describe methodology used for classification using images directly. Bounding box of traffic sign provided in annotations is used to crop each image and it is converted to grey scale. Then images are rescaled to 40×40 and are reshaped to 1×1600 row vector. Such row vector for each image are stored in a matrix X and corresponding labels/classes in y.

3.1 Linear Discriminant Analysis [2]

We use linear discriminant analysis on X and y to classify images. i.e, It is assumed that a sample vector from each class comes from a multivariate Gaussian distribution with different means μ_k but with same covariance matrix Σ and with a prior probability of π_k. Since, we do not know μ_k, π_k and Σ, they have to estimated from the training data (X, y)

- $\hat{\pi}_k = N_k/N$ where N_k is the number of class-k observations
- $\hat{\mu}_k = \sum_{y_i=k} x_i/N_k$
- $\hat{\Sigma} = \sum_{k=1}^{K} \sum_{y_i=k} (x_i - \hat{\mu}_k)^T (x_i - \hat{\mu}_k)/(N - K)$

(a)

(b)

(c)

(d)

(e)

(f)

Figure 1: Samples of images from dataset, scaled to double the size

where x_i is ith row of X and N and K are total number of observations in training data and number of classes respectively. Calculation of Σ_k can be vectorized as

$$\hat{\Sigma} = \sum_{k=1}^{K}(X_k - 1 * \hat{\mu}_k)^T(X_k - 1 * \hat{\mu}_k)/(N - K)$$

where 1 is matrix of ones of appropriate dimension and X_k is matrix with rows from X for which $y_i = k$

Once μ_k, π_k and Σ are estimated, we can use Bayes classification. This can be shown to be equivalent to

$$\hat{y}(x) = \text{argmax}_k\, x\Sigma^{-1}\mu_k^T - \mu_k\Sigma^{-1}\mu_k^T/2 + \log\pi_k$$

This computation can also be vectorized and further optimizations are possible to make this computation faster. However, we stop with vectorization as it is fast enough for our applications. Note that decision boundaries from this scheme are linear.

3.1.1 Results

This gives the worst classification of all the algorithms we implemented. This is because we do not expect raw image data to be linearly separable.

Correctly classified	Misclassified	Total test images
8539	4091	12630
63.61%	36.39%	100%

3.2 Fisher's Linear Discriminant/Fisherfaces [3]

This method selects a linear transformation W such that the ratio of the between-class scatter and the within-class scatter is maximized. Between-class scatter matrix is defined as

$$S_B = \sum_{k=1}^{K} N_k(\mu_k - \mu)^T(\mu_k - \mu)$$

Within-class scatter matrix is defined as

$$S_W = \sum_{k=1}^{K}\sum_{y_i=k}(x_i - \mu_k)^T(x_i - \mu_k)$$

Here μ is mean of all images in training set.

If S_W is non sigular, the optimal projection is chosen as the

$$W_{opt} = \text{argmax}_W \frac{|W^T S_B W|}{|W^T S_W W|} = [\mathbf{w_1}\mathbf{w_2}\ldots\mathbf{w_m}]$$

where $\mathbf{w_i}$ are generalized eigenvectors of S_B and S_W corresponding to m largest eigenvalues λ_i.

Usually S_W is singular. This is because rank of S_W is at most $N - K$ and N, number of training images, is usually much smaller than number of pixels n in each image. If so, PCA is done to reduce dimensions of data to $N - K - 1$ before applying FLD.

In our case, $N - K = 32609 - 43 = 32566$ is much larger than number of pixels in each image $40 \times 40 = 1600$. So, S_W is invertible and generalized eigenvectors can be used to find W_{opt}.

Once W_{opt} is found, we project each image in the training set using W_{opt} and save them in a database. For a given test image, we project it onto the vector space and class corresponding to nearest neighbour in the database is returned as prediction.

3.2.1 Results

There is a considerable improvement in performance compared to LDA.

Correctly classified	Misclassified	Total test images
10914	1716	12630
86.41%	13.59%	100%

4 Histogram of Oriented Gradients (HOG) [4]

These are feature set originally proposed in [4] to detect humans in an image. However these can also be used for classification/detection of other objects like cars or traffic signs.

The basic idea is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradients or edge positions. This is implemented by dividing the image window into small spatial regions ('cells'), for each cell accumulating a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell. It is also useful to contrast-normalize the local responses before using them. This can be done by accumulating a measure of local histogram 'energy' over somewhat larger spatial regions ('blocks') and using the results to normalize all of the cells in the block

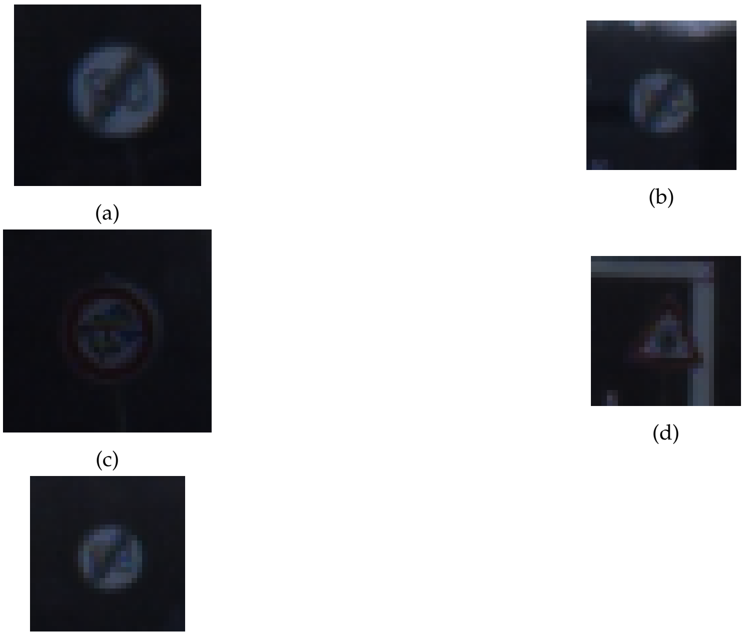
Figure 2: Visualization of histogram of oriented gradients

Three sets of precomputed HOG descriptors are provided with dataset. To compute these features, each image is scaled to 40×40 pixels and following sizes of cells and overlapping blocks are used.

Name	cell size	block size	binning resolution	length
HOG 1	5×5	2×2	8 (sign ignored)	1568
HOG 2	5×5	2×2	8	1568

5 Classification using HOG descriptors

Precomputed HOG vectors are used for classification for both LDA and FLD. There's a substantial improvement in performance.



(a) (b) (c) (d) (e)

Figure 3: Some Misclassified Images when LDA is applied on HOG 1 descriptors

5.1 LDA Results

Descriptor	Correctly classified	Misclassified	Total test images
HOG 1	11766 (93.16%)	864 (6.84%)	12630(100 %)
HOG 2	11766 (95.68%)	545 (4.32%)	12630(100 %)

5.2 FLD Results

Descriptor	Correctly classified	Misclassified	Total test images
HOG 1	11766 (94.32%)	717 (5.68%)	12630(100 %)
HOG 2	12197 (96.57%)	433 (3.43%)	12630(100 %)

6 Random Forests (RF)[2]

We apply technique of random forests on HOG descriptor data to classify signs. Random forests are ensemble of bagged randomized decision trees in which a random set of features are selected for best split. Each randomized decision trees above is trained with bootstrap sample of training data. i.e, sample of same size as training data is drawn from training data picking each sample uniformly randomly *with replacement*.

6.1 Results

Our implementation of RF gives very bad results. This is because splits are premature at each node. We couldn't debug this. These are the results from sklearn package's RandomForestClassifier class. Number of trees in the ensemble trained are 50.

Descriptor	Correctly classified	Misclassified	Total test images
HOG 1	11709 (92.70%)	921 (7.30%)	12630(100 %)

7 Improvements possible

Our random forests code was not able to find good splits. We could've debugged this had we had more time.

References

[1] Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition, *J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel*, August 2012, Neural Networks (32), pp. 323-332

[2] The Elements of Statistical Computing, *T.Hastie, R.Tibshirani,J.Friedman*

[3] Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection *P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman*

[4] Histograms of oriented gradients for human detection, *N. Dalal, B.Triggs* Proceedings of the IEEE conference on computer vision and pattern recognition, 2005 (pp. 886,893).