Pedestrian and Vehicle Classification

CS771A: Machine Learning: Tools, Techniques, Applications(2016)

Prof. Harish Karnick

Indian Institute of Technology, Kanpur

Group No. 34
Ishita Ankit(13316)
Kanupriya Agarwal(13338)
Shivam Malhotra(13660)
Gaurav Gautam(11917270)

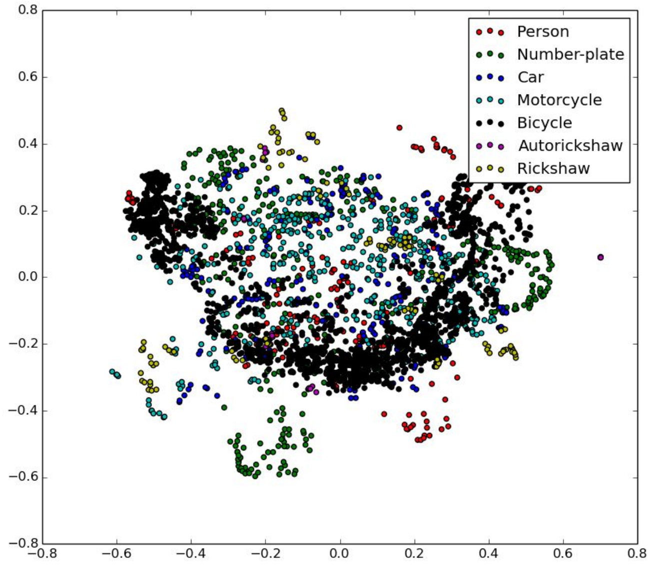
Problem Statement

The dataset given is surveillance video footage from IIT-K security video cameras. Our problem statement is to detect objects in the video and classify them as person or vehicle(non-person). The strategy followed by us is summarized as follows:

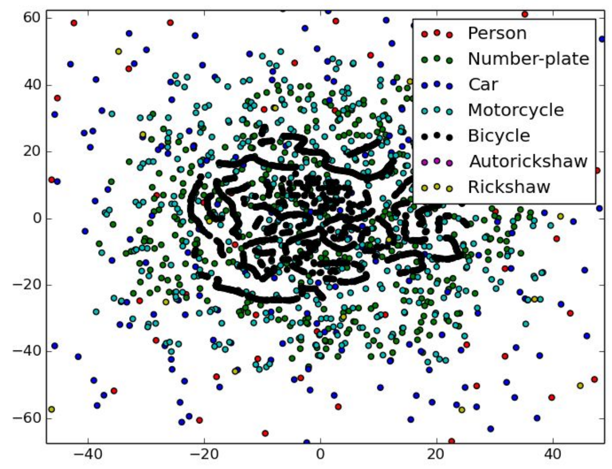
1. **Background subtraction**: A preliminary processing done in order to detect objects in video frames.
2. **Object Detection**: Blobs of foreground images in our video frames are obtained after background subtraction. In this stage these blobs are processed to get bounding boxes for foreground objects in the frame. Also a these objects are tracked across frames in the video.
3. **Dataset acquisition**: The annotated video data is cleaned and processed into labelled subimages of objects like person, car, bicycle, motorcycle, rickshaw etc.
4. **Object classification**: Various feature descriptors and learning algorithms are used on the dataset obtained to get trained models, which are then tested on a subset that was set aside from the beginning.

Problem Statement

Dimensionality reduction using PCA shows that the dataset is not very good in terms of separability.



Dimensionality reduction using TSNE also shows that the dataset is not very good in terms of separability.



Background subtraction

Background subtraction was performed in order to get bounding boxes for objects in the video frames. The first step is then to get the background image. To do this the following approaches were tried:

- **Average pixel intensity**: The fames were converted to grayscale. Then the pixels of the background image were estimated to have the intensity that is the average intensity value of the pixel over several frames. Since the videos were small initially the average was taken over the entire video. We were told in the presentation that we should have tried averaging over fewer frames as well. We have now implemented this and the results are shown in figure. 1. Since averages are static for a fixed time frame, thus this method produces comparatively worse results than the MoG method.

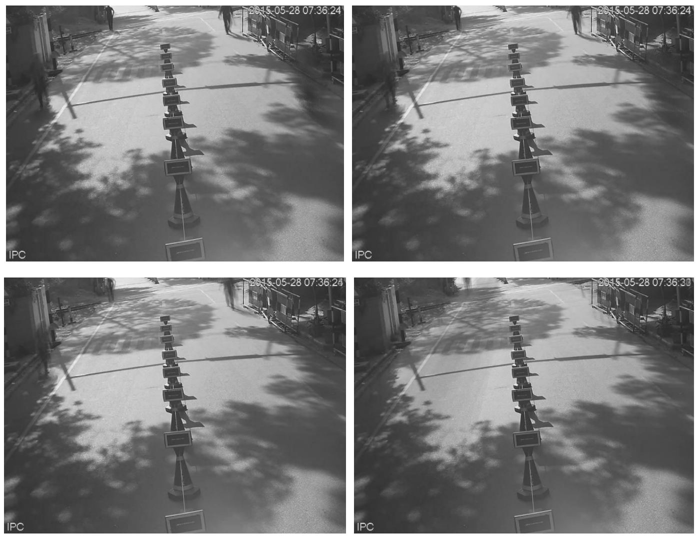


Figure 1: Clockwise, the background images calculated by averaging 10, 20, 40 and 200 frames.

- **Mode pixel intensity**: Another approach is to take the mode pixel intensity as the background value. The result from this approach was not very different from the first approach, as shown in figure. 2. The implementation of this method requires more memory since all the frames over which the mode is taken must be stored. But again background being static, the results were comparatively worse.

- **Mixture of gaussians**: In this method the intensity of every background pixel is learned using a mixture of gaussians as the underlying probability model. The opencv implementation of this algorithm was used. This method gave the best results as it is more responsive to changes than the previous methods but is the slowest among the three. **Therefore this method was chosen to do further processing in the project.**

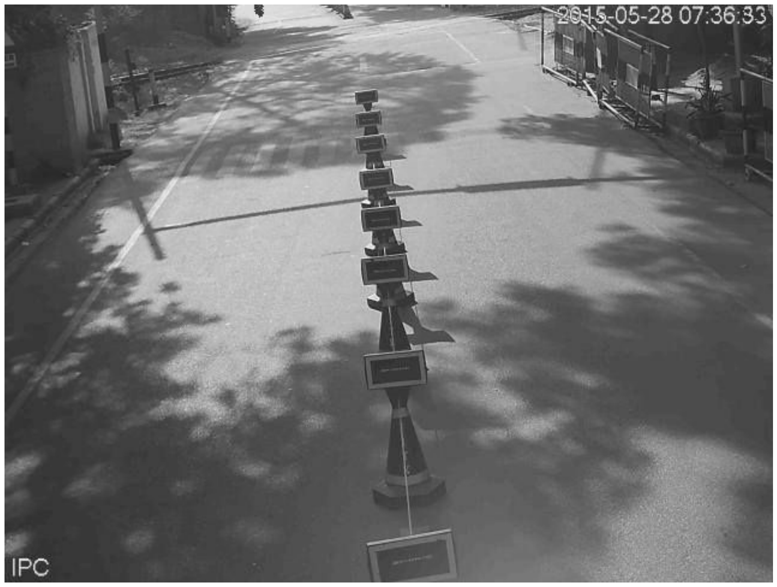


Figure 2: Background image calculated using the mode of pixel intensity.

Object detection

After getting the background image and subtracting it from the frames, all foreground objects in the video are obtained as blobs. However these blobs are of poor quality due to small holes that are present in their bodies. Their boundaries are also not smooth. A combination of morphological effects are applied to these blobs like erosion and dilation to close these holes and smooth the boundaries. Subsequently contours are obtained for the blobs. Finally opencv is used to draw bounding boxes for these contours. The effect of application of these operations is shown in figure.3.

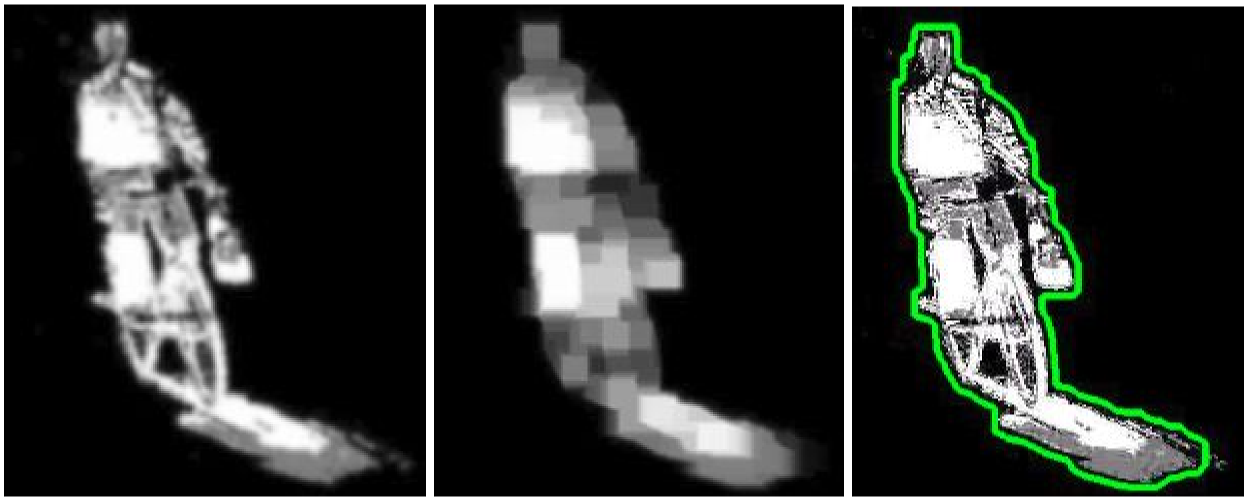


Figure 3: From left to right (a) Applying gaussian blur, (b) Applying morphological transformations like erosion and dilation, (c) Getting the contour for the blob.

Once the contours and bounding boxes are obtained, the boxes can be drawn on the video. A **weak form of tracking** was implemented by calculating the overlap between the boxes calculated for consecutive frames. For each bounding box calculated in a frame an overlap with all the boxes in the previous frame is calculated. The overlap is normalized by the area of the smaller among the overlapping boxes. Then we check if the overlap between the current box and any of the previous boxes is greater than a threshold value of 0.85 (we tuned this parameter). If true the box is identified as being the box from the older frame, otherwise it is identified as a new box. The result of detecting the objects and tracking them is shown in figure. 4. Also probable overlapping objects are predicted if there is a sudden growth in the area of a box, but this method needs improvements.

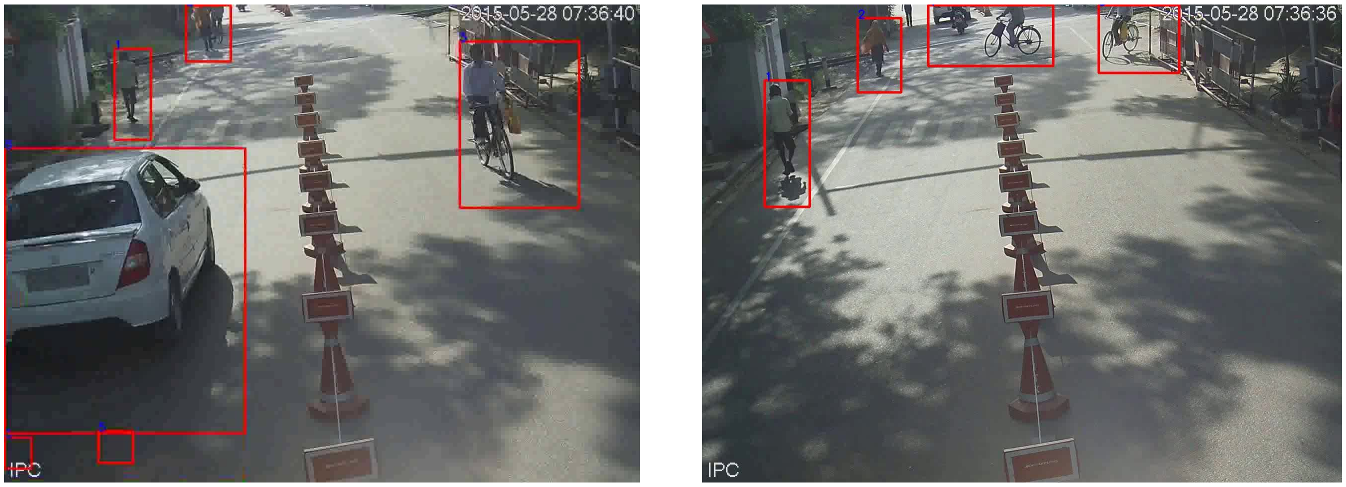


Figure 4: Objects detected and tracked in the video after applying background subtraction(mog2), and subsequent calculation of bounding boxes.

Dataset acquisition

The dataset of annotated videos was cleaned using scripts and then by hand. We first cut out all annotated subimages from the videos using opencv removing all images with a significant overlap. Then all images where the object was only partially visible or very blurry were manually removed. All sub-images were resized to the size of an average person in the video. Finally some transforms were applied to the images that were thus obtained, such as flipping along the vertical axis and displacing the image to the left, right, top or bottom to generate more images from the remaining samples. Doing all this generated a dataset of 302 persons and 268 non-person images where all the objects were clearly visible and completely in the image.

Object classification

The dataset is now in the form of annotated images of a uniform size. A portion is set aside for testing. The next task is to develop a representative representation for the images in the form of a vectors and learn classifiers from the training set for our task. Several feature descriptors and learning algorithms were tried for this purpose. The results for these experiments are shown here.

Grayscale pixel intensities

The simplest descriptor is obtained by taking the grayscale pixel intensities of the images as the feature vector. The results are tabulated here:

Classifier	Accuracy
SVM	**79.10 %**
Random Forest	82.08 % (max_depth = 16, n_estimators =200)
Adaboost	80.59 % (max_depth = 4, n_estimators =120)

Table 1: Classifier performance for person/vehicle classification using grayscale pixel intensities as the descriptor

We have tuned the parameters for our classifiers and the best parameters have been reported above. The performance of random forest saturates around 200 estimator tress of a maximum depth of 16. Similarly for Adaboost, good performance was obtained for around 200 decision tree estimators of max depth 4.

Further on these grayscale intensity images, **K-means clustering(K=8)** was performed for both categories to visualize the average images as the computer sees them. Also for classification phase, **k-nearest neighbors algorithm** with majority vote was used to classify images. An accuracy of **78.35%** was obtained using this method when using **3 neighbors**. Visualization of some of the class representatives obtained by doing K-means clustering are shown in figure.5.

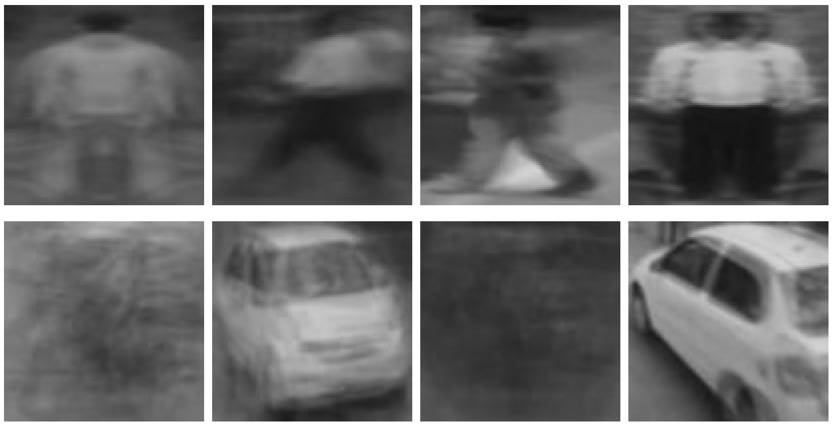


Figure 5: Class representatives of person and vehicles obtained by doing K-means clustering.

Histogram feature

Histogram feature of the grayscale images were calculated by binning the pixel intensities into 128 bins. The result of using different classifiers for on this feature vector is shown in Table. 2.

Classifier	Accuracy
SVM	**73.13 %**
Random Forest	67.91 % (max_depth = 16, n_estimators =200)
Adaboost	67.16 % (max_depth = 4, n_estimators =120)

Table 2: Classifier performance for person/vehicle classification using grayscale pixel intensity histogram as the descriptor

As can be seen, performance is worse than the previous method, probably because histogram loses spatial information on binning. Thus the representation is not very descriptive. Still SVMs give best performance for classification task.

Hierarchical histogram feature

To improve upon the previous method, we used hierarchical histograms[2] as shown in the image below. Basically histograms of subsections of the given imageare taken and then the vectors are appended together to obtain the final feature vector. This is shown in figure. 6. The result of classification using the hierarchical histogram features are shown in Table. 3.

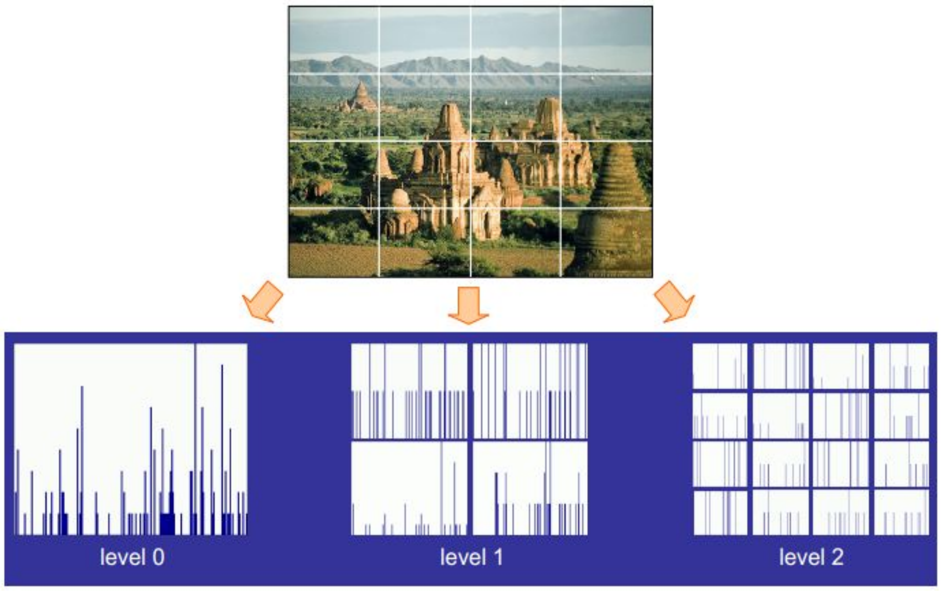


Figure 6: *Image taken from Grauman & Darrell (2005).* Hierarchical histograms are calculated on separate sections of the image and then appended together.

Classifier	Accuracy
SVM	78.35 %
Random Forest	**79.85 %** (max_depth = 16, n_estimators =200)
Adaboost	77.61 % (max_depth = 4, n_estimators =120)

Table 3: Classifier performance for person/vehicle classification using grayscale pixel intensity hierarchical histogram as the descriptor

On tuning the parameters, we obtained best performance when 128 bins used for the histograms and number of levels in our hierarchy were 3. This means overall 21 128 bin histogram vectors appended together to obtain the final feature vector. Also as can be seen, the performance improves considerably over simple histograms, as expected.

HOG features[4]

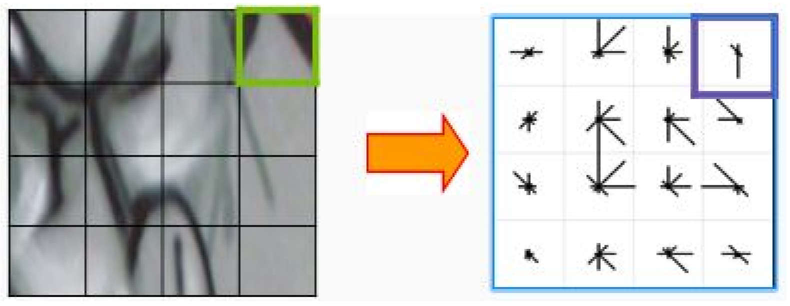


Figure 7: *(images from slides by Deva Ramanan and Kristen Grauman)*Calculation of HOG features by binning pixel intensity gradients.

The previous methods used only depend on the distribution of image intensities for describing the image. But as noted in vision tasks, the gradient store more information than just the intensities. This information is captured by HoG descriptors. HoG(Histogram of oriented gradient) features are calculated by binning the gradients of pixel intensities instead of the pixel intensities directly. The image is divided into small cells of (16x16) pixels and intensity gradients are binned in 8 directions. The histograms thus obtained from each cell are then merged to get the descriptor vector. This process is shown in figure. 7. The results for this feature vector are shown in Table. 4.

Classifier	Accuracy
SVM	**95.37 %**
Random Forest	94.02 % (max_depth = 16, n_estimators =200)
Adaboost	91.93 % (max_depth = 4, n_estimators =120)

Table 4: Classifier performance for person/vehicle classification using hog as the descriptor

As can be seen, the best performance on our dataset reaches as high as 95% for SVM classifier.

SIFT features[5]

SIFT(Scale Invariant Feature Transform) features are able to detect descriptive patches at different scales in the image and different orientations. 20 sift interest points per image were calculated (example in figure. 8.)and were then appended together to give a single feature vector per image. The results when using this descriptor are shown in Table 5.

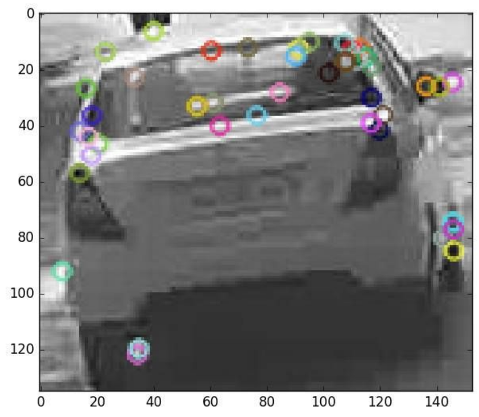


Figure 8: SIFT interest points calculated for one image.

Classifier	Accuracy
SVM	67.76 %
Random Forest	71.07 % (max_depth = 16, n_estimators =200)
Adaboost	**73.55 %** (max_depth = 4, n_estimators =120)

Table 5: Classifier performance for person/vehicle classification using SIFT features for 20 points as the descriptor

Thus as can be seen, the efficiency comparatively worse than other methods probably because of lesser number of SIFT points used.

Failed attempts

1. **Haar classifier:** A haar classifier was trained using the dataset but the classifier didn't predict persons correctly. This is probably because of the small size of the training dataset. An example of the prediction of persons made by the haar classifier is shown in figure. 10.

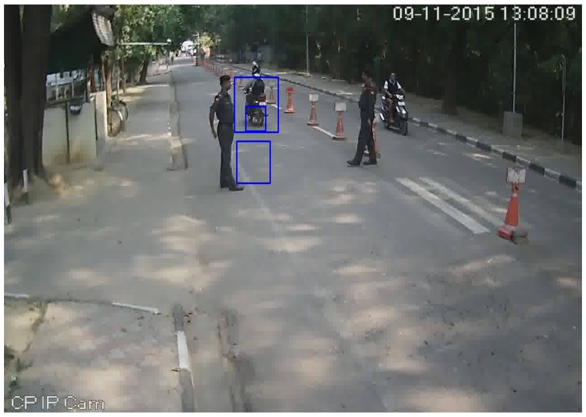


Figure 10: Haar classifier prediction of person in for one frame. The classifier couldn't be trained to give correct predictions on the dataset.

2. **SIFT variation:** Another approach with the SIFT features was tried which failed. In this approach a maximum of 50 SIFT interest points were chosen from each image and they were all given the label of the image. Thus a classifier was trained on these vectors obtained from all training images. Further for any test image, its interest points were extracted, classified by the above classifier and majority vote was taken to give a label to the image. The performance for this method was poor, ie. the cross - validation efficiency with SVMs was close to 40%.

Classification using pre-trained neural network

We were told during the presentation that we should have tried neural networks for classification. We have done this now and used the neural network on the test images of persons from our dataset. We used an pretrained model by Ross Girshick called Fast R-CNN [3] available at github. The file tools/demo.py in the repo was slightly modified by us and is present in the code we submitted under 3_Classify/demo.py . Some examples of detections using this classifier are shown in figure. 9. This classifier was able to give 46.21% accuracy over the person images we had in our dataset. The efficiency was less probably because there is a big difference in the images used for training this model and the test images provided by us.

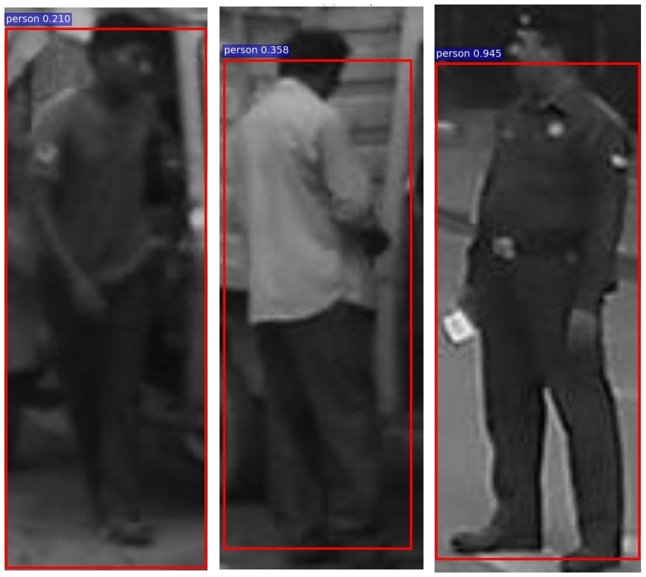


Figure 9: Neural network detection of person in the person test data.

Classifying the bounding boxes obtained by background subtraction using the hog descriptor and SVM classifier

This combination of descriptor and classifier gave the overall best performance for our task on our dataset. So for a new input video, bounding boxes are obtained using the background subtraction method, and then the trained classifier is used to predict the labels for these objects. Some examples of the predictions done by the classifiers are shown in figure. 10.

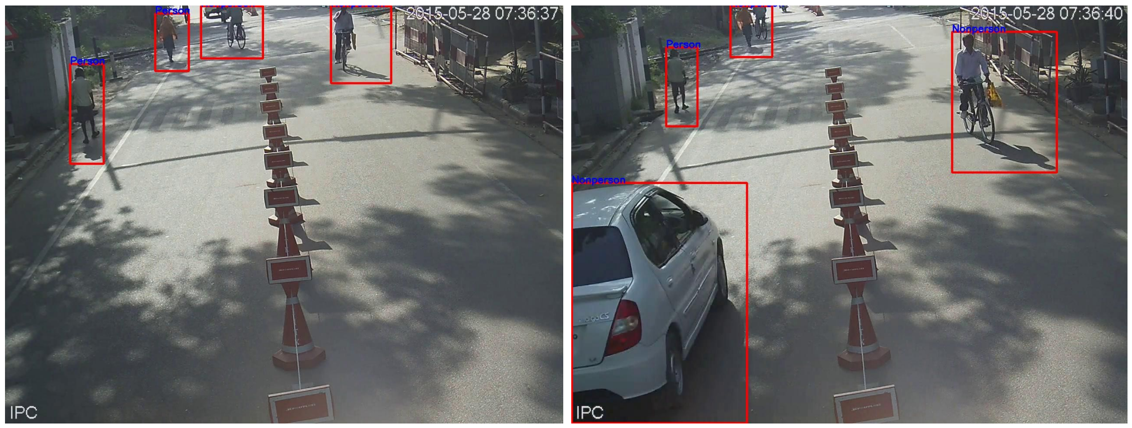


Figure 10: Classification of bounding boxes obtained by background subtraction.

Final results

When we tabulate the performance of all classifiers against all descriptors we see that the **HOG features perform the best.**

	Gray Image	Histogram	Hierarchical Histogram	HOG	SIFT 1
Clustering and k-NN	78.35 %	-	-	-	-
SVM	**79.10 %**	**73.13 %**	78.35 %	**95.37 %**	67.76 %
Random Forests	82.08 %	67.91 %	**79.85 %**	94.02 %	71.07 %
Adaboost	80.59 %	67.16 %	77.61 %	91.93 %	**73.55 %**

Acknowledgements

We would like to express our sincere gratitude towards our project mentor Dr. Harish Karnick for his constant support and invaluable suggestions. We would also like to express our gratitude towards the institute for providing state-of-the-art facilities to support this project.

References

[1] *Lecture slides by Deva Ramanan and Kristen Grauman*
[2] Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories *Grauman & Darrell (2005)*
[3] Fast R-CNN, Ross Girshick, arXiv:1504.08083
[4] Histogram of oriented gradients for human detection. 2005., Navneet Dalal and Bill Triggs.
[5] Distinctive Image Features from Scale-Invariant Keypoints,David G. Lowe
[6]Oliveira, M.; Santos, V. Automatic Detection of Cars in Real Roads using Haar-like Features