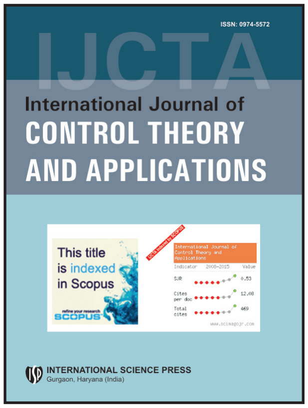


International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 24 • 2017

Pedestrian Detection of Video and Still Images for ADAS Application

J.D. Dorathi Jayaseeli[a], D. Malathi[b], A.L. Amutha[c], S. Gopika[d] and Shashank Mishra[a]

[a]*Assistant Professor, Department of Computer Science and Engineering, SRM University. Email: dorathijayaseeli.jd@ktr.srmuniv.ac.in*
[b]*Professor, Department of Computer Science and Engineering, SRM University. Email: malathi.d@ktr.srmuniv.ac.in*
[c]*Assistant Professor, Department of Computer Science and Engineering, SRM University. Email: amutha.al@ktr.srmuniv.ac.in*
[d]*Research Scholar, Department of Computer Science and Engineering, SRM University. Email: gopikas@kristujayanti.com*
[e]*B. Tech Student, Department of Computer Science and Engineering, SRM University. Email: 24shashankm@gmail.com*

Abstract: This paper aims at developing a daytime pedestrian detection system for Advance Driver Assistance System (ADAS). The objective of ADAS is to enforce driving safety and convenience. Pedestrian detection is a part of ADAS application. Few reasons that make pedestrian detection a difficult problem are huge variations in appearance, huge deformation in shapes, camera motion, illumination variations and highly prone to false positives. In view of these an attempt is made to develop an automatic detection of pedestrians on the road by camera (monocular or stereo) using Histograms of Oriented Gradients (HOG) features and linear Support Vector Machine (SVM). It also implements image pre-processing modules namely histogram equalization which is performed frame by frame basis and consecutive frame subtraction with appropriate thresholding to decrease the effect of the egomotion of the camera. Thus using the pre-processing modules, it investigates methods to reduce the false positives of the pedestrian.

Keywords: Feature Extraction, Histogram Equalization, Histograms of Oriented Gradients, Pedestrian Detection, Support Vector Machine.

1. INTRODUCTION

In the recent era computer vision system has become the research interest of many scholars. The reason behind that is these systems are capable of doing many peculiar and complicated tasks in the same way as humans are doing. But still they are not able to do some intellectual tasks which humans can do in a smart way. This include recognizing speech, objects, hand written characters and logical interpretation.

The current research concentrates on developing computer vision systems that perform human like thinking, remembering and reasoning capabilities. Out of these, object recognition is one of the skill that a human can do effortlessly and instantaneously. Pedestrian detection is a peculiar case of object detection where the target object is people. However, humans can identifya man or woman even if they have difference in their color, pose and background complexity. Many researchers working in computer vision are focused on bringing out object recognition systems that can be able to recognize the specific identity of an object being observed. One of the primary tasks to reach that goal is the implementation of accurate object detection, which means localizing, in terms of locality and scale, a particular object in a static image and video. It has been an active area of research because of the positive potential impact of these derived applications, such as surveillance systems, automotive safety, multimedia content analysis, robotics, assistive technology and advanced interactive interfaces.

Even though many approaches have already been proposed, pedestrian detection is proved to be a challenging task due to the wide range of pedestrian appearances. Moreover, it is also a competitive domain since the high research activity in this area uninterruptedly pushes the upper boundary of accuracy and efficiency to new levels. Existing pedestrian detection systems use combinations of basic features and trainable classifiers (SVM, boosted classifiers or random forests).

The purpose of this paper is to develop and evaluate a state of the art pedestrian detection approach. In particular, the work will achieve two specific goals:

1. Building a pedestrian detector from the origin and evaluate its performance, both in terms of recognition accuracy and speed. A pedestrian detection can be understood as a combination of two building blocks, a feature extraction algorithm and a classification method that uses the features to make the object/non-object decision. Our approach uses a boosted cascade of Histograms of Oriented Gradients (HOG) as features and Support Vector Machines as classifiers.
2. Perform a sensitivity analysis in order to study how the training parameters influence the progress of learning process and the final performance of the pedestrian detector.

This paper is organized in the following manner: Chapter II gives a brief historical account of pedestrian detection, Chapter III explains the design strategy and proposed methodology, Chapter IV deals with the implementation of pedestrian detection in video and still images, Chapter V shows the results and output of each module and finally Chapter VI concludes the paper with future work.

2. PEDESTRIAN DETECTION – A BRIEF HISTORICAL ACCOUNT

Pedestrian detection has been an active field of research recently; numerous researchers have reported significant results in this area. However, pedestrian detection is still a challenging task due to the wide variety of appearances (body pose, occlusions, clothing, lighting, background) that pedestrians can take in images**.** The variability of employed techniques for pedestrian detection purposes is very diverse. In general, proposed training methods can be classified into holistic approaches, which try to identify the full-human body using a single detection region, and part-based approaches in which each part of the body is identified separately and a human is detected if some or all parts are presented in a reasonable spatial configuration. Additionally, according to the testing procedure, we can distinguish between sliding window approaches, which densely scan the image at different positions and scales, and key point approaches, which first extract local interest points from the image, and then evaluate only regions around these points. At the same time, several types of features, ranging from low level features to complex visual descriptors, have been used in combination with different types of classifiers. There is an extensive literature on pedestrian detection techniques and their performance; there exist specific detailed surveys that concentrate the state of the art in pedestrian detection [1]. One of the earliest detectors, proposed by Papa Georgiou and Poggio [2] was a sliding window approach that used multi-scale Haar wavelets and support vector machines (SVM). Viola and Jones [3] developed these ideas to build a detector that integrated Haar-like features in a cascade of AdaBoost classifiers. Moreover, they pioneered the use of integral images for fast feature computation. These contributions continue today to be the basis of modern detection techniques.

Adoption of features based on gradient information brought important performance gains. Histogram of Oriented Gradients is the most popular gradient-based feature; it was introduced in conjunction with SVMs by

Dalal and Triggs [4]. At cost of computation time, the algorithm offers high robustness and accuracy even in complex image environments. In [5] Zhu et. al., improved the algorithm speed by using integral histograms [6]. A similar gradient-based feature, Edge Orientation Histograms (EOH), in combination with SVMs was presented by Shashua et. al., [7]

Shape and texture are other visual common exploited features [8]. Gavrila and Philomin [9]-[10] tested different template matching techniques to detect people. Mori et. al., [11] modeled human body configurations represented by local Shape Context. Furthermore, specific shape features, such as shapelets [12] and edgelets [13] have also been developed and learnt using boosting algorithms.

Advanced features and new gains appear when introducing motion information. Some of the approaches that include motion features are introduced in [14], that builds a detector for static-camera using Haar wavelets and a cascade boosting scheme, or in [15] that creates new descriptor called Histograms of Flow, designed for situations with a moving background or camera. All in all, taking advantage of motion information improves detection performance and reliability for practical applications.

Although some of the previous low-level features showed considerable good results, they are often not enough to describe accurately all visual information. One step beyond is done by combining different simple features, each additional feature provides complementary information and the overall performance is reasonably improved. Wu and Nevatia [16] combined HOG, edgelets and covariance features. Mao et. al., [17] developed a system to improve contour detection based on edgelets, Haar-like features and Viola's Adaboost cascade framework. Wang et. al., [18] combined a texture descriptor based on Local Binary Patterns (LBP) with HOG features. Wojek and Schiele [19] combined Haar-like features, shapelets, shape context and HOG features outperforming any individual features.

Part-based methods appear to overcome holistic representation methods limitations in spatial occlusion situations. Mohan et. al., [20] divide human body into four parts: head-shoulder, legs, left arm and right arm; and learns a part-based detector for each using SVM classifiers and Haar wavelet features. Mikolajczyk et. al., [21] have used seven parts (face/head for frontal view, face/head for profile view, head-shoulder for frontal and rear view, head-should for profile view and legs) based on local gradient features and follows the Viola and Jones learning approach. The three part-based detectors outperform their correspondent holistic method.

To sum up, while the used set of features and descriptors is quite diverse, the applied learning techniques tend to be discriminant classifiers such as AdaBoost or SVMs. According to different sources [19], HOG and dense Shape Context features perform better than other features independently of the employed classifier and a combination of multiple features tends to improve considerably the overall performance of individual detectors, and adopting a part-based solution seems to be the most reasonable option.

3. DESIGN STRATEGY AND PROPOSED METHODOLOGY

The design of our pedestrian detection system is shown in Figure 1. Our system works in two stages namely: The *learning stage* and the *runtime stage*. In the learning stage, first the input image is acquired from training dataset. Second, 756 HOG features are extracted from each input image. Third appropriate features are selected to train the binary classifier. Finally the binary classifier is learned and updated using the full training dataset images. In the runtime stage the system is tested with unseen images from test dataset. The system should be able to identify whether the image has pedestrian or not. This is done by scanning the whole image and applying the classification whether pedestrian or not over every region of the image.

Depending on the feature extraction method, we use an appropriate edge detection method to compute the image derivatives.

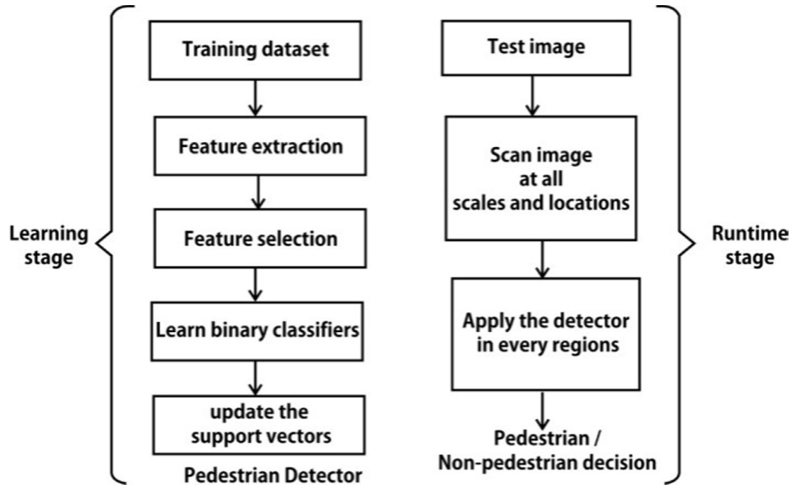


Figure 1: Pedestrian detection system

4. IMPLEMENTATION OF PEDESTRIAN DETECTION SYSTEM IN VIDEO AND STILL IMAGES

4.1. Pedestrian Dataset for Training and Testing

First step to develop pedestrian detection system is acquisition of image from a standard dataset. Developing our own dataset with large number of images and video will be a time consuming process and moreover the images and video may suffer from poor illumination, occlusion and orientation. The dataset should be good enough so that we can get all the information from the image and video. The dataset should contain sufficient number of positive and negative examples to train and test the developed system.

In order to develop a robust system which will be able to find pedestrians from the input image or video for all possible pedestrian variations, we need a large dataset. There are so many publicly available datasets for pedestrian detection.

In this work, we use the *MIT Pedestrian Dataset* and some images from *INRIA person dataset*, for both the learning and runtime stages. Figure 2 shows the examples of positive and negative samples of INRIA person dataset. Figure 3 and Figure 4 show the negative and positive samples of MIT pedestrian dataset respectively. These are the well-known and widely accepted dataset for pedestrian detection. It is the dataset actually produced for the HOGs, and it has been a reference for much pedestrian detection research, including this research work.

The dataset contains 1208 positive training samples and 566 positive testing samples. In view of increasing the number of positive training and testing samples, these images are mirrored left-right. As a result the number of positive training and testing samples are doubled. Thus a total of 2416 positive training samples and 1132 positive testing samples are created. In our work we used normalized samples of size 96×160 which contain a single person centered in a 64×128 window. Figure 5 shows the generated positive samples.

On the other hand, the negative training set does not contain persons at all. It consists of images with flowers, beach, scenery etc., objects such as cars, bicycles, motorbikes, furniture and many more. The dataset contains 1218 negative training samples and 453 negative testing samples. In order to increase the number of negative training and testing samples they can also be mirrored. Again these negative samples are also normalized to a 64×128 pixel window to generate negative samples. Figure 6 shows the generated negative samples. From the two dataset resources we prepare a dataset having 1000 positive templates and 1000 negative templates, and we have used 80% of the templates for training and 20% of the templates for testing. Figure 7 shows a sample of a sequence of test video frames.

Figure 2: Examples of negative and positive images from the INRIA Person Dataset

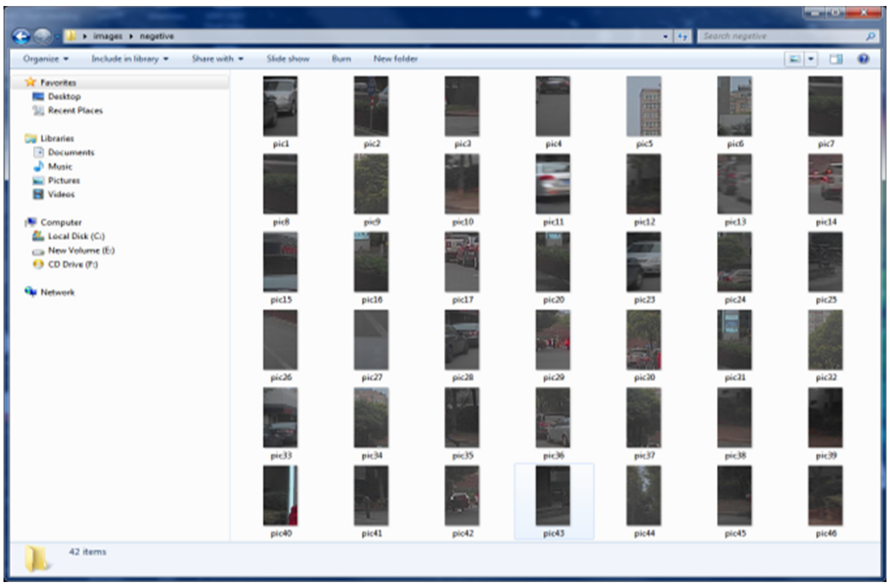


Figure 3: Negative samples

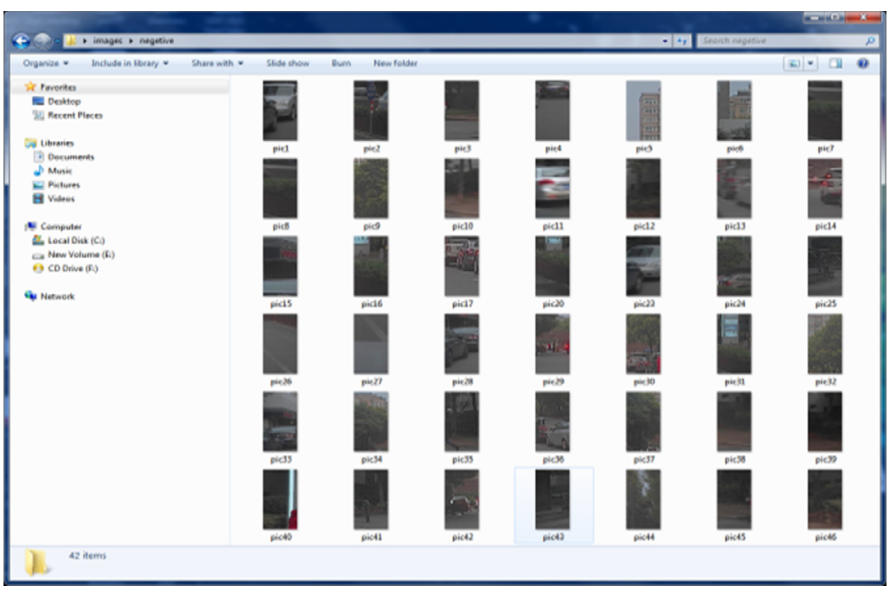


Figure 4: Positive samples of MIT pedestrian dataset

Figure 5: Generation of positive samples

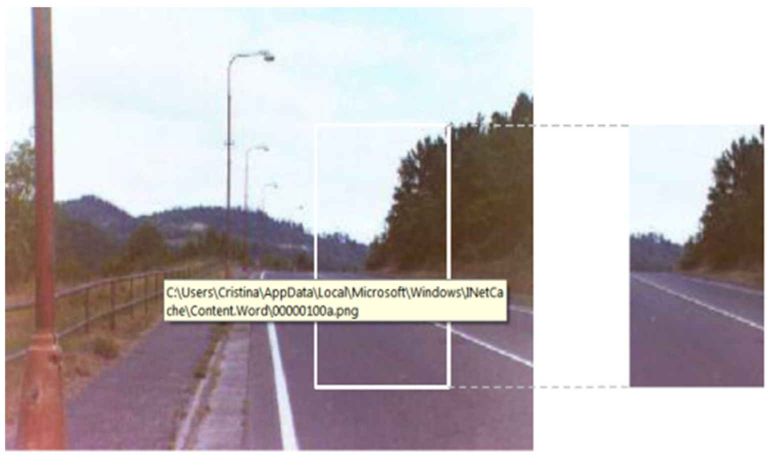


Figure 6: Generation of negative samples

Figure 7: A sample of a sequence of test video frames

4.2. Feature Extraction

The second step is to extract features from the normalized image of size 64 × 128 pixel. Features are extracted from all the images and these are in turn used to train the binary classifier. We have extracted 756 HOG features from each image. This idea is borrowed from Dalal and Triggs [4], they used a type of visual gradient-based descriptor to detect humans from the images. The actual process is explained as follows, first divide the image into 16 × 16 cell and the image will contain 4 blocks of 16 × 16 cell as the size of the image is 64 × 128. Then 9-bins are used and we have to find θ using the formula $\tan^{-1}(dx/dy)$ where dx and dy are the boundary values along x-axis and y-axis respectively. The calculated θ value is mapped to these 9-bins. This 16 × 16 cell will slide over the full image to get $(7 \times 3 \times 9 \times 4 = 756)$ 756 features.

4.3. Pedestrian Dataset Training

For training the pedestrian training dataset we have used the package LIBSVM 3.20. Libsvm is available at *http://www.csie.ntu.edu.tw/~cjlin/libsvm*. In the LIBSVM 3.20 package we have used the module svm-train.c to train our dataset.

4.4. Pedestrian Dataset Testing

For testing the pedestrian test dataset and compute the accuracywe use the module from LIBSVM package svm-predict.c

4.5. Preprocessing Result

For classification of pedestrians fromnon-pedestrians we have used primal form of decision function. svm-train.c outputs a model file for the learned parameters in the form of a text document. The support vectors are recorded in that text file, so we extract the support vectors (*sv*), the coefficients and the bias to compute the decision value of a particular test image and according to the sign we classify it as a pedestrian from a non-pedestrian.

5. RESULTS AND OUTPUT

5.1. Preprocessing Results

We provide a snapshot of the results of frame differencing between two consecutive video frames at time t and $t+1$ in Figure 8 and in Fig 9. We show the results of both normal frame differencing and the results of the frame differencing followed by thresholding with a fixed sized window in Figure 10, the output shows clearly that our method works better for segmenting out the pedestrian region of interest if the camera is in motion.

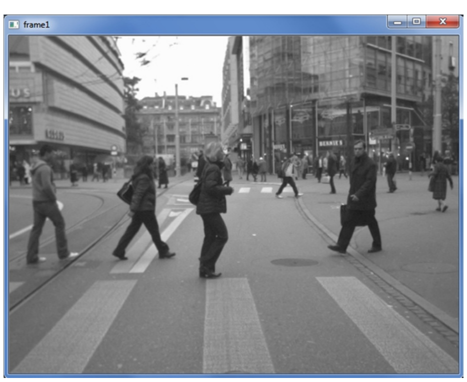


Figure 8: Test image of a video frame at time t

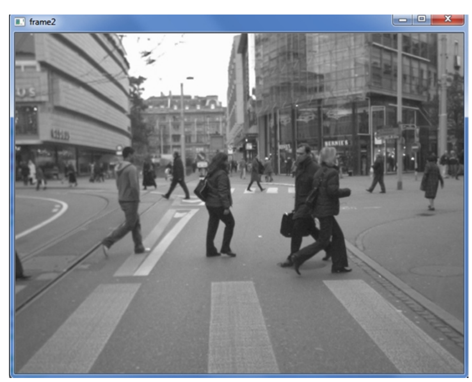


Figure 9L Test image of a video at time $t + 1$

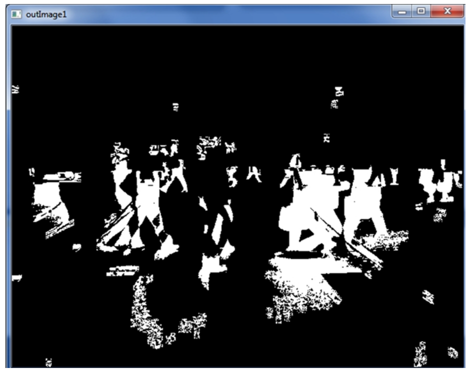
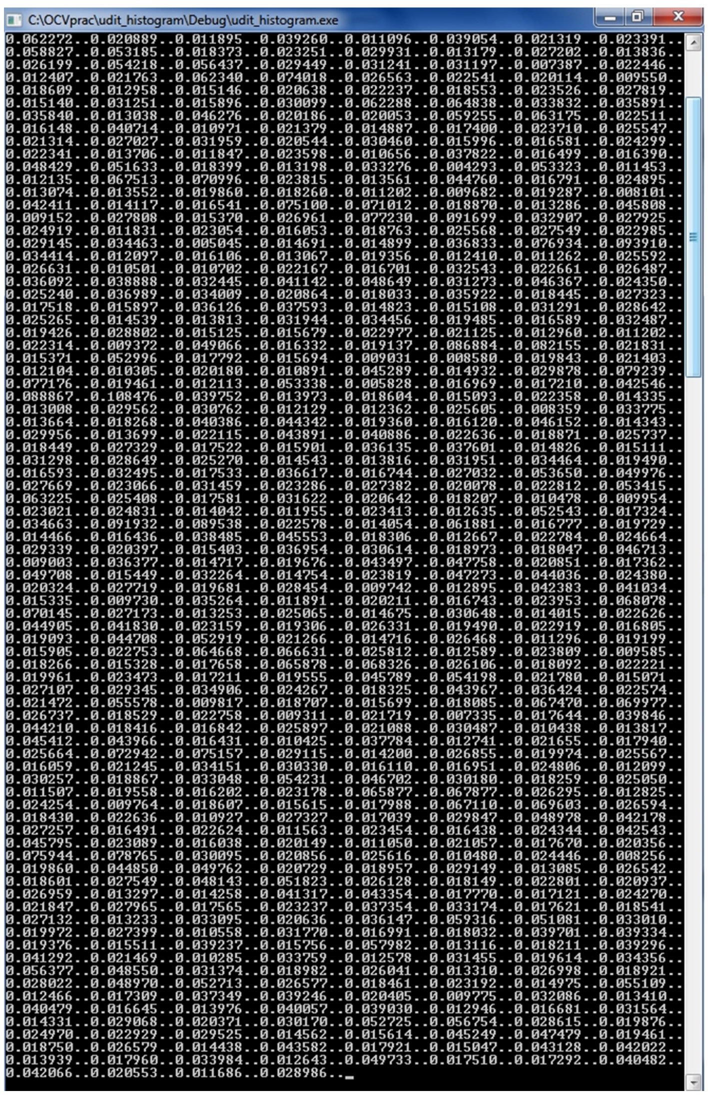


Figure 10: Output image after frame difference at *t* and *t* + 1 followed by thresholding

5.2. HOG Features

In Figure 11, we provide a snapshot of the HOG features that are computed.




```
0.062272..0.020889..0.011895..0.039260..0.011096..0.039054..0.021319..0.023391..
0.058827..0.053185..0.018373..0.023251..0.029931..0.013179..0.027202..0.013836..
0.026199..0.054218..0.056437..0.029449..0.031241..0.031197..0.007387..0.022446..
0.012407..0.021763..0.062340..0.074018..0.026563..0.022541..0.020114..0.009550..
0.018609..0.012958..0.015146..0.020638..0.022237..0.018553..0.023526..0.027819..
0.015140..0.031251..0.015896..0.030099..0.062288..0.064838..0.033832..0.035891..
0.035840..0.013038..0.046276..0.020186..0.020053..0.059255..0.063175..0.022511..
0.016148..0.040714..0.010971..0.021379..0.014887..0.017400..0.023710..0.025547..
0.021314..0.027027..0.031959..0.020544..0.030460..0.015996..0.016581..0.024299..
0.022341..0.013706..0.011847..0.023598..0.010656..0.037822..0.016499..0.016390..
0.048429..0.051633..0.018399..0.013198..0.033276..0.004293..0.053323..0.011453..
0.012135..0.067513..0.070996..0.023815..0.013561..0.044760..0.016791..0.024895..
0.013074..0.013552..0.019860..0.018260..0.011202..0.009682..0.019287..0.008101..
0.042411..0.014117..0.016541..0.075100..0.071012..0.018870..0.013286..0.045808..
0.009152..0.027808..0.015370..0.026961..0.077230..0.091699..0.032907..0.027925..
0.024919..0.011831..0.023054..0.016053..0.018763..0.025568..0.027549..0.022985..
0.029145..0.034463..0.005045..0.014691..0.014899..0.036833..0.076934..0.093910..
0.034414..0.012097..0.016106..0.013067..0.019356..0.012410..0.011262..0.025592..
0.026631..0.010501..0.010702..0.022167..0.016701..0.032543..0.022661..0.026487..
0.036092..0.038888..0.032445..0.041142..0.048649..0.031273..0.046367..0.024350..
0.025240..0.036989..0.034009..0.020864..0.018033..0.035922..0.018445..0.027323..
0.017518..0.015897..0.036126..0.037593..0.014823..0.015108..0.031291..0.028642..
0.025265..0.014539..0.013813..0.031944..0.034456..0.019485..0.016589..0.032487..
0.019426..0.028802..0.015125..0.015679..0.022977..0.021125..0.012960..0.011202..
0.022314..0.009372..0.049066..0.016332..0.019137..0.086884..0.082155..0.021831..
0.015371..0.052996..0.017792..0.015694..0.009031..0.008580..0.019843..0.021403..
0.012104..0.010305..0.020180..0.010891..0.045289..0.014932..0.029878..0.079239..
0.077176..0.019461..0.012113..0.053338..0.005828..0.016969..0.017210..0.042546..
0.088867..0.108476..0.039752..0.013973..0.018604..0.015093..0.022358..0.014335..
0.013008..0.029562..0.030762..0.012129..0.012362..0.025605..0.008359..0.033775..
0.013664..0.018268..0.040386..0.044342..0.019360..0.016120..0.046152..0.014343..
0.029956..0.013699..0.022115..0.043891..0.040886..0.022636..0.018871..0.025737..
0.018449..0.027329..0.017522..0.015901..0.036135..0.037601..0.014826..0.015111..
0.031298..0.028649..0.025270..0.014543..0.013816..0.031951..0.034464..0.019490..
0.016593..0.032495..0.017533..0.036617..0.016744..0.027032..0.053650..0.049976..
0.027669..0.023066..0.031459..0.023286..0.027382..0.020078..0.022812..0.053415..
0.063225..0.025408..0.017581..0.031622..0.020642..0.018207..0.010478..0.009954..
0.023021..0.024831..0.014042..0.011955..0.023413..0.012635..0.052543..0.017324..
0.034663..0.091932..0.089538..0.022578..0.014054..0.061881..0.016777..0.019729..
0.014466..0.016436..0.038485..0.045553..0.018306..0.012667..0.022784..0.024664..
0.029339..0.020397..0.015403..0.036954..0.030614..0.018973..0.018047..0.046713..
0.009003..0.036377..0.014717..0.019676..0.043497..0.047758..0.020851..0.017362..
0.049708..0.015449..0.032264..0.014754..0.023819..0.047273..0.044036..0.024380..
0.020324..0.027719..0.019681..0.028454..0.009742..0.012895..0.042383..0.041034..
0.015335..0.009730..0.035264..0.011891..0.020211..0.016743..0.023953..0.068078..
0.070145..0.027173..0.013253..0.025065..0.014675..0.030648..0.014015..0.022626..
0.044905..0.041830..0.023159..0.019306..0.026331..0.019490..0.022919..0.016805..
0.019093..0.044708..0.052919..0.021266..0.014716..0.026468..0.011296..0.019199..
0.015905..0.022753..0.064668..0.066631..0.025812..0.012589..0.023809..0.009585..
0.018266..0.015328..0.017658..0.065878..0.068326..0.026106..0.018092..0.022221..
0.019961..0.023473..0.017211..0.019555..0.045789..0.054198..0.021780..0.015071..
0.027107..0.029345..0.034906..0.024267..0.018325..0.043967..0.036424..0.022574..
0.021472..0.055578..0.009817..0.018707..0.015699..0.018085..0.067470..0.069977..
0.026737..0.018529..0.022758..0.009311..0.021719..0.007335..0.017644..0.039846..
0.044210..0.018416..0.016842..0.025897..0.021088..0.030487..0.010438..0.013817..
0.045412..0.043966..0.016431..0.010425..0.037784..0.012741..0.021655..0.017940..
0.025664..0.072942..0.075157..0.029115..0.014200..0.026855..0.019974..0.025567..
0.016059..0.021245..0.034151..0.030330..0.016110..0.016951..0.024806..0.012099..
0.030257..0.018867..0.033048..0.054231..0.046702..0.030180..0.018259..0.025050..
0.011507..0.019558..0.016202..0.023178..0.065877..0.067877..0.026295..0.012825..
0.024254..0.009764..0.018607..0.015615..0.017988..0.067110..0.069603..0.026594..
0.018430..0.022636..0.010927..0.027327..0.017039..0.029847..0.048978..0.042178..
0.027257..0.016491..0.022624..0.011563..0.023454..0.016438..0.024344..0.042543..
0.045795..0.023089..0.016038..0.020149..0.011050..0.021057..0.017670..0.020356..
0.075944..0.078765..0.030095..0.020856..0.025616..0.010480..0.024446..0.008256..
0.019860..0.044850..0.049762..0.020729..0.018957..0.029149..0.013085..0.026542..
0.018601..0.027549..0.048143..0.051823..0.026128..0.018149..0.022801..0.020937..
0.026959..0.013297..0.014258..0.041317..0.043354..0.017770..0.017121..0.024270..
0.021847..0.027965..0.017565..0.023237..0.037354..0.033174..0.017621..0.018541..
0.027132..0.013233..0.033095..0.020636..0.036147..0.059316..0.051081..0.033010..
0.019972..0.027399..0.010558..0.031770..0.016991..0.018032..0.039701..0.039334..
0.019376..0.015511..0.039237..0.015756..0.057982..0.013116..0.018211..0.039296..
0.041292..0.021469..0.010285..0.033759..0.012578..0.031455..0.019614..0.034356..
0.056377..0.048550..0.031374..0.018982..0.026041..0.013310..0.026998..0.018921..
0.028022..0.048970..0.052713..0.026577..0.018461..0.023192..0.014975..0.055109..
0.012466..0.017309..0.037349..0.039246..0.020405..0.009775..0.032086..0.013410..
0.040479..0.016645..0.013976..0.040057..0.039030..0.012946..0.016681..0.031564..
0.014331..0.029068..0.020371..0.030170..0.052725..0.056754..0.028615..0.019876..
0.024970..0.022929..0.029525..0.014562..0.015614..0.045249..0.047479..0.019461..
0.018750..0.026579..0.014438..0.043582..0.017921..0.015047..0.043128..0.042022..
0.013939..0.017960..0.033984..0.012643..0.049733..0.017510..0.017292..0.040482..
0.042066..0.020553..0.011686..0.028986..
```

Figure 11: HOG values of a pedestrian template

5.3. N Fold Cross Validation Result

Figure 12 shows that our cross validation gives an accuracy of *93.966%.*

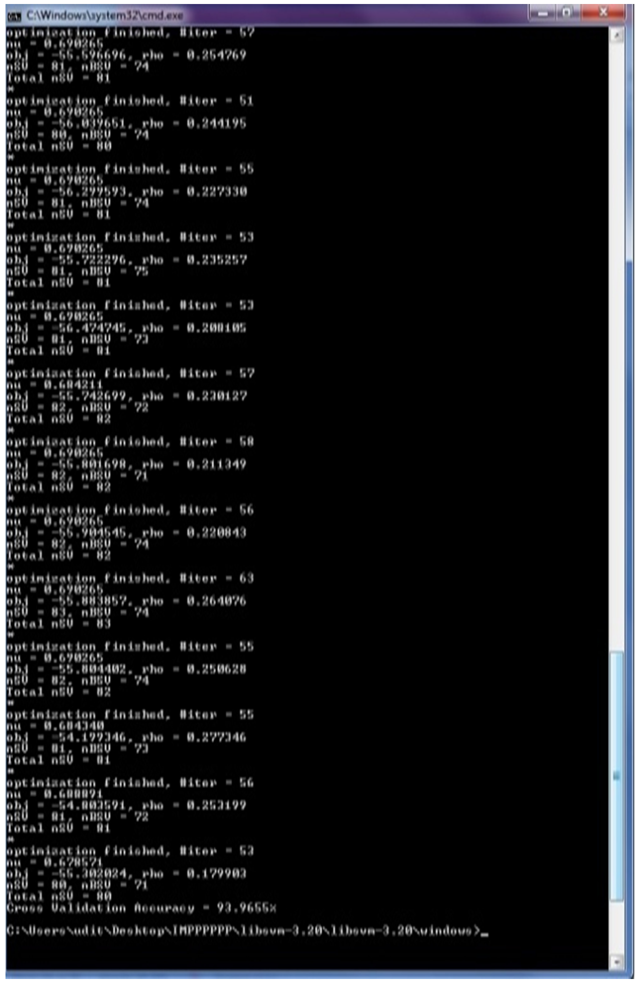
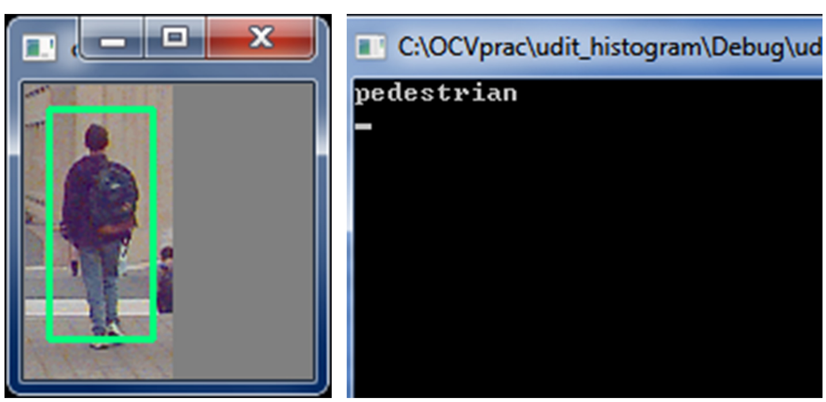


Figure 12: Cross Validation Result

5.4. Pedestrian Detection in still images

In Figure 13, we show the pedestrian inside a bounding box and detect non pedestrian also.



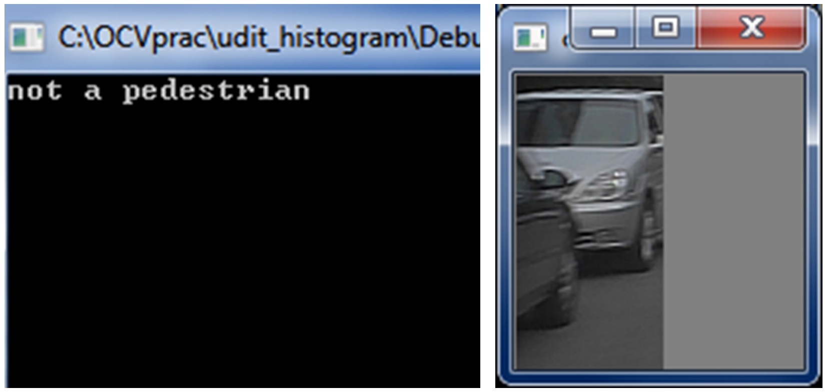


Figure 13: The pedestrian classified from a non-pedestrian

6. CONCLUSION AND FUTURE WORK

This proposal helped us to first, get deeper knowledge in pedestrian detection using MIT and INRIA datasets. Then, we built a pedestrian detection system using HOG features and linear SVM. Finally we evaluated the behavior of our system and proved accuracy of 93% with 756 HOG features.

Here we have used C and we have used openCV 2.1 for only reading and writing the images. Our future work would involve developing the parallel C implementation of the modules pipelines with the GPU.

Other possibilities to extend our work can be usage of large datasets with more number of positive and negative examples. Then in terms of feature extraction we can use texture features, color features and shape features. Finally when we come to classification, we can try classifiers such decision trees, map reduce, deep learning architectures etc., and their performance can be evaluated.

REFERENCES

[1] K. Anuradha and N. Sairam, Spatio-temporal based Approaches for Human Action Recognition in Static and Dynamic Background: A Survey, (2016) *Indian Journal of Science and Technology*, Vol 9(5).

[2] C. Papageorgiou, T. Poggio, A trainable system for object detection,(2000) *International Journal of Computer Vision*, 38 (1), pp. 15-33.

[3] P. Viola, M.J. Jones, Robust real-time face detection, (2004) *International Journal of Computer Vision*, 57(2), pp. 137-154.

[4] N. Dalal, B. Triggs, Histograms of Oriented Gradients for Human Detection, *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. 1 Pages; 886-893 Year of Publication: 2005 ISBN:0-7695-2372-2).

[5] Q. Zhu, M.-C. Yeh, K.-T. Cheng, S. Avidan, Fast human detection using a cascade of histograms of oriented gradients, *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. 2, Pages: 1491-1498 Year of Publication: 2006).

[6] F.M. Porikli, Integral histogram: a fast way to extract histograms in Cartesian spaces, *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. 1, Pages. 829-836 Year of Publication: 2005).

[7] A. Shashua, Y. Gdalyahu, G. Hayun, Pedestrian detection for driving assistance systems: single-frame classification and system level performance, *IEEE International Conference on Intelligent Vehicles*, (Pages. 1-6 Year of Publication: 2004 ISBN:0-7803-8310-9).

[8] E. Poongothai and A. Suruliandi , Survey on Colour, Texture and Shape Features for Person Re-Identification, (2016) *Indian Journal of Science and Technology*, Vol. 9(29).

[9] D.M. Gavrila, V. Philomin, Real-time object detection for smart vehicles, (1999) *IEEE International Conference on Computer Vision*, (Vol1, Pages: 87-93 Year of Publication: 1999 ISBN: 0-7695-0164-8).

[10] D.M. Gavrila, Multi-feature hierarchical template matching using distance transforms, P*roceedings of the 14th International Conference on Pattern Recognition*, (Vol1, Year of Publication: 1998 Pages: 439-444 ISBN:0-8186-8512-3).

[11] G. Mori, J. Malik, Estimating human body configurations using shape context matching, *Proceedings of the 7th European Conference on Computer Vision*– Part III, (Pages: 666-680 Year of Publication: 2014 ISBN:3-540-43746-0).

[12] P. Sabzmeydani, G. Mori, Detecting pedestrians by learning shapelet features, *IEEE Conference on Computer Vision and Pattern Recognition* (Pages. 9-12 Year of Publication: 2007 ISBN:1-4244-1179-3).

[13] B. Wu, R. Nevatia, Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors, (2007) *International Journal of Computer Vision,* 75(2) pp. 247-266.

[14] P. Viola, M.J. Jones, D. Snow, Detecting pedestrians using patterns of motion and appearance, *Proceedings of the 9th International Conference on Computer Vision*, (Vol1, Pages. 734-741 Year of Publication:2003 ISBN:0-7695-1950-4).

[15] N. Dalal, B. Triggs, C. Schmid, Human detection using oriented histograms of flow and appearance, *Proceedings of the 9th European Conference on Computer Vision*, (Pages: 428-441 Year of Publication: 2006 E-ISBN:978-3-540-33835-2).

[16] B. Wu, R. Nevatia, Optimizing discrimination-efficiency trade off in integrating heterogeneous local features for object detection, *IEEE Conference on Computer Vision and Pattern Recognition* (Pages: 1-8 Year of Publication: 2008)

[17] X. Mao, F. Qi, W. Zhu, Multiple-part based pedestrian detection using interfering object detection, *Proceedings of the 3rd International Conference on Natural Computation*, (Pages: 165-169 Year of Publication: 2007 ISBN:0-7695-2875-9).

[18] X. Wang, T. Han, S. Yan, An HOG-LBP Human detector with partial occlusion handling, *IEEE 12th International Conference on Computer Vision* (Pages: 32-39 Year of Publication: 2009 E-ISBN:978-1-4244-4419-9).

[19] C. Wojek, B. Schiele, A performance evaluation of single and multi-feature people detection, (2008) *Proceedings of the 30th DAGM symposium on Pattern Recognition*, (Pages: 82-91 Year of Publication: 2008 ISBN:978-3-540-69320-8).

[20] A. Mohan, C. Papageorgiou, T. Poggio, Example-based object detection in images by components, (2001) *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4), pp. 349-361.

[21] K. Mikolajczyk, C. Schmid, A. Zisserman, Human detection based on a probabilistic assembly of robust part detectors, *Proceedings of 8th European Conference on Computer Vision* (Pages. 69-82 Year of Publication: 2004ISBN:978-3-540-21984-2).