

Implementation of Robust HOG-SVM based Pedestrian Classification

Reecha P. Yadav
K.K.W.I.E.E.R
Nashik
India

Vinuchackravathy
Senthamilarasu and
Krishnan Kutty
KPIT Technologies Ltd.
Pune
India

Sunita P. Ugale
K.K.W.I.E.E.R
Nashik
India

ABSTRACT

Achieving pedestrian protection by means of computer vision is not a new topic in the field of computer vision research; however it is still being pursued with renewed interest because of the huge scope for performance improvement in the existing systems. Generally, the task of pedestrian detection (PD) involves stages such as pre-processing, ROI selection, feature extraction, classification, verification/refinement and tracking. Of all the steps involved in the PD framework, the paper presents the work done towards implementing the feature extraction and classification stages in particular. It is of paramount importance that the extracted features from the image should be robust and distinct enough to help the classifier distinguish between a pedestrian and a non-pedestrian, while a good classification algorithm would go a long way in precisely identifying a pedestrian as well as in simplifying the verification stage of the PD framework. The presented work focuses on the implementation of the Histogram of Oriented Gradients (HOG) features with modified parameters that can represent accurate intrinsic information of the image. Classification is achieved using Support Vector Machine (SVM). However instead of employing a readily available SVM library, the linear SVM implemented uses the Sequential Minimal Optimization (SMO) algorithm. The results observed by this HOG-SVM combination show promise to be the best feature extraction cum classification module for a full-fledged PD system.

General Terms

Computer Vision, Image processing.

Keywords

Pedestrian detection, Feature extraction, Classification, HOG, SVM, SMO.

1. INTRODUCTION

In view of the fact that 22 % of the world's road traffic deaths occur among pedestrians [1], measures intended to minimize pedestrian fatalities due to negligence on part of the driver or other such unfavorable circumstances, are becoming important. Pedestrian Detection Warning System (PDWS) in automobiles detects pedestrians in front of the vehicle and warns the drivers to take appropriate decision. Such an advanced warning system achieves pedestrian safety by

employing cameras along with computer vision and image processing algorithms.

Generally, the task of pedestrian detection (PD) involves stages such as pre-processing, Region of Interest (ROI) selection, feature extraction, classification and tracking [2]. Figure 1 shows the block diagram of a general pedestrian detection framework. Pre-processing (e.g. image smoothing, contrast enhancement etc.) is done on the input image/video in order to aid the performance of the subsequent stages. The enhanced image is then subjected to segmentation to obtain a set of ROIs, which in this case are the regions in the image which are more likely to contain a pedestrian. Feature extraction stage aims at acquiring certain image characteristics, which will most meaningfully represent the 'pedestrian' information in the image. In a PDWS, the features so extracted help in classifying the input image samples as those containing 'pedestrian' or not. Lastly, tracking is carried out to maintain the pedestrian estimates — obtained after classification — over the subsequent frames of a video, under real-time scenario.

The paper is aimed towards implementation of a sub-task of the complete PDWS — that of implementing a robust feature-classifier ensemble — which can significantly contribute to improving the overall PD accuracy. The presented work implements the Histogram of Oriented Gradients (HOG) feature proposed in [3], with modifications made to certain parameters to suit classifier performance as detailed in Section 3 below. Classification is achieved using a Support Vector Machine (SVM). However as opposed to using a readily available SVM library (e.g. SVMlight) as done in [3], the linear SVM is implemented using the Sequential Minimal Optimization (SMO) algorithm as described in [4].

The remainder of this paper is organized as follows. Section 2 provides a succinct review of the major feature-classifier approaches engaged in the past for PD. Section 3 details the HOG methodology and the proposed modification. The details of basic principle of SVM and its implementation using SMO form the content in Section 4. The results obtained by executing the two blocks i.e. HOG based feature extraction and SMO based classification are elaborated upon in Section 5. Finally, concluding remarks have been derived and relevant future work proposed as part of the epilogue in Section 6.

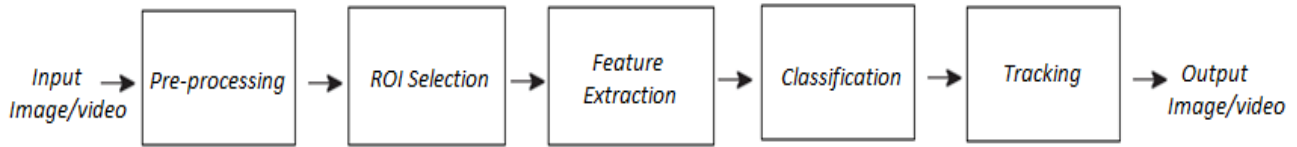


Figure 1: Block diagram describing major modules in vision based pedestrian detection system [2]

2. LITERATURE SURVEY

This section derives help from the work done in [5] to provide a brief overview of the feature extraction and the classification methodologies adopted by various researchers towards achieving PDWS. Amongst all the features employed across the literature, the most discriminative stand-alone feature is the HOG feature [3]. Much of its advantage comes from its ability to reliably capture the local edge/gradient information; along with a built-in invariance to local illumination condition. Almost all modern detectors employ HOG as a stand-alone detector or in combination with some other features. For instance, HOG in combination with Local Binary Patterns (LBP) and Local Ternary Patterns (LTP) has been used in [6], with improved gains in detection results. Enhancements to HOG also include the works described in [7], [8], [9], [10] amongst others.

Some other shape-based features have also been successfully employed. Most noticeable works amongst them, being based on shape context [11], edgelet [12] and shapelet [13] features. Edgelets, which are short line segments/curves, act as efficient local shape descriptors (e.g. useful for describing the head-shoulder curve) and hence provide robustness against occlusion [2]. Shapelet features are based on the principle of extracting gradients from the salient regions of an image, which are more likely to contain a pedestrian [11]. Both, shapelet and edgelet features are learned using a boosting-based approach. Shape Context, similar to HOG descriptors are based on location and orientation of edges, however are represented using a log-polar histogram [11]. Covariance descriptors [7] are another popular approach for pedestrian detection. They employ a covariance matrix sensitive to parameters such as positions, gradients, gray-level of pixels, etc. Features can also be extracted from optical flow images. Dalal et al. [14] obtained the Histogram of Oriented Flow (HOF) features (describing motion) using optical flow images. However, the results obtained using motion features showed only little improvement over its counterparts.

Upon obtaining the feature space representation of the ROIs, a classifier is employed to divide this feature space into the class ‘pedestrian’ and ‘non-pedestrian’. One of the popular classifiers is Support Vector Machine (SVM). It solves a binary classification problem by defining a decision boundary

between two distinct classes, so as to have maximum margin between the classes. In case of non-linear data, kernels can be applied to linearize the data in some higher dimension, thus enabling their linear classification. Linear SVM gives good performance when used in conjunction with some discriminative feature like HOG or its variants [3], [14]. Employing a kernel-based non-linear SVM [15] yields slightly better results but at the cost of increased computation time and memory requirement. A general form of SVM, called as latent SVM, has also been used in part-based pedestrian detection scheme in [16]. Latency here refers to the initially unknown part locations in a particular detection.

Boost-based classifiers are another popular classification approach. They combine various weak learners, over a number of iterations to build a robust classifier [17], [18]. Though the boosting framework requires more time to train, it is capable of making real-time detections. Cascade classifiers or a fusion of classifiers can also be used to improve detection speed and accuracy by using the output of a fast yet weak classifier, to drive a strong but slow classifier [19].

ANN (Artificial Neural networks), with their ability to store experiential data, can also be employed for the classification task. Multiple layers of neurons allow for non-linear decision boundaries between classes. The scheme used in [20] employs ANN to model parts and occlusions. Another classification approach in use currently is that of Decision forests [21]. Decision trees calculate the membership to a particular class by repeatedly partitioning a dataset into uniform subsets. Decision forests are obtained by combining the predictions of multiple independent Decision Tree Models to obtain a single prediction. Output of ANN and decision forest strategy has been found to be at par with each other [22]. However greater complexity incurred in these approaches, advocates the usage of SVM, which gives comparable performance with a much smaller implementation complexity.

3. IMPLEMENTATION OF HOG

From the literature survey performed, HOG emerged as the most successful stand-alone descriptor for supervised classification. This section describes feature extraction as proposed in [3] and its implementation. Figure 2 shows the block diagram of the HOG extraction process. The following sub-sections detail each of the constituent blocks.

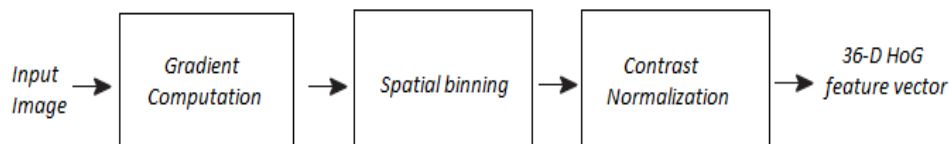


Figure 2: Block diagram for extracting HOG from an input image

3.1 Gradient Computation

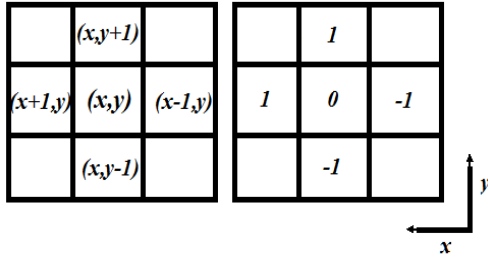


Figure 3: A 3x3 image region and its corresponding gradient kernel

Gradient of an image measures the change in the gray level for each pixel in the image along with the direction of the change. If we represent the gradient as a vector, the length of the vector represents the gradient magnitude, while its direction gives the gradient orientation.

The gradient information along horizontal and vertical direction of the image are calculated by convolving the given gray-scale image with the gradient kernel $[-1, 0, 1]$ and $[-1, 0, 1]^T$ respectively, as shown in Figure 3. It is clear that the horizontal gradient (I_x) is obtained by taking the difference between the column values (x-direction), while the vertical gradient (I_y) involves difference calculation between the row values (y-direction).

For a pixel (x,y) , the convolution of kernel over the image intensity mathematically implies equations (1) and (2). The gradient magnitude (m) and orientation (θ) is calculated using the equation (3) and (4) respectively.

$$I_x = I(x+1, y) - I(x-1, y) \quad (1)$$

$$I_y = I(x, y+1) - I(x, y-1) \quad (2)$$

$$m = \sqrt{I_x^2 + I_y^2} \quad (3)$$

$$\theta = \tan^{-1} \frac{I_y}{I_x} \quad (4)$$

In case of an input color image, the image gradient is computed independently for each of the three constituent color planes. However, for each pixel, the color channel corresponding to the maximum gradient magnitude contributes to the final gradient magnitude and orientation values for that pixel.

3.2 Spatial / Orientation Binning

In this step, the input image is divided into n cells of equal area. For every pixel in the cell, the gradient orientation and magnitude are calculated and are accumulated into a histogram. The histogram comprises of nine bins obtained by equally dividing the interval $0-180^\circ$ into bins of interval 20° . Consider the 8×8 cell region as shown in Figure 4. Assuming that the pixel $(1,1)$ corresponds to a gradient orientation value of 30° , it can be mapped to the bin corresponding to the interval $21-40^\circ$. In other words, the gradient magnitude corresponding to the orientation value of 30° , will be added to the histogram bin of $21-40^\circ$ for that pixel. Similarly, the next pixel $(1,2)$ with a gradient orientation value of 55° , is mapped with its gradient magnitude into the bin corresponding to $41-60^\circ$ in the histogram of the cell under consideration.

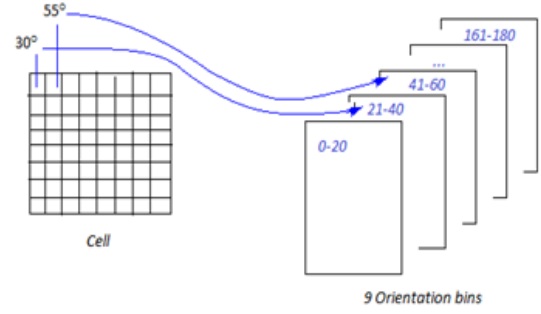


Figure 4: Gradient Orientation Binning

In this way, the gradient orientation of each pixel is weighted by its magnitude into appropriate bins of a cell-based histogram. This ensures that the object shape is captured reliably in terms of distribution of local gradient (or edge) information.

Each cell contributes a 9 dimensional (D) feature vector, thereby producing $9 \times n$ feature vector for the given input image. Reference [3] claims that a best detection performance can be achieved with a cell size of 8×8 pixel region. However, a minimum cell size of 16×16 pixel region is chosen in the presented work to speed up the HOG computation without much loss in the classifier performance.

3.3 Normalization

Normalization is an important step in robust HOG extraction process. It helps HOG maintain its discriminative property and perform consistently even against parameters like local illumination variations and foreground-background contrast in the input image. Normalization is done using 'block' as a fundamental region of operation. Each block is a region comprising of a square array of 4 cells. Also, each new block is defined with a 50% overlap with the previous block.

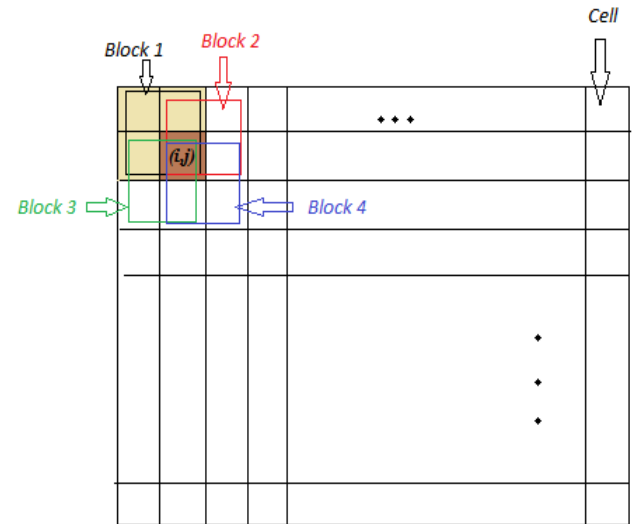


Figure 5: A single cell (i, j) contributes to the final feature vector w.r.t four different blocks

Normalization ensures that the cell-based local gradient information is invariant to local illumination conditions. It can be done by calculating a measure of 'gradient energy' [16] — using the gradient magnitudes obtained in the previous stage — over each of the blocks defined over the input image. Each of the cell (i, j) is then normalized w.r.t the gradient energy measure calculated for each block containing this cell, as

described in [16]. Figure 5 depicts a cell (i, j) , which is a part of four blocks. Hence, this particular cell is normalized separately w.r.t the gradient energy calculated over each of the four blocks individually. Thus, each such cell contributes a 9D vector four times to the final feature vector. In other words, each cell contributes a 36D feature vector ($4 \times 9D = 36D$ vector).

4. IMPLEMENTATION OF SVM

SVM is a state-of-the-art classification method and is widely used in supervised classification in machine learning applications. Apart from its simplicity and accuracy, the SVM classifier is popular because of the ease with which it can deal with high-dimensional data. It performs binary classification by defining a hyper-plane that classifies the input data into two classes.

As shown in Figure 6, SVM has two variants - Hard margin SVM and Soft-margin SVM. Hard margin SVM requires all data points to be classified correctly into their respective classes. However, the more popular soft-margin approach allows controlled misclassification of difficult or noisy examples — using a parameter ‘C’ — to achieve a maximum margin linear classifier by avoiding over-fitting and hence, the usage of kernels.

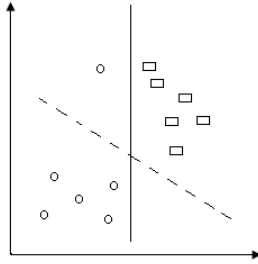


Figure 6: Hard-margin (solid line) and Soft-margin (dotted line) SVM Classifier

The presented work employs a soft-margin SVM due to its ability to deal with non-linear data linearly. Obtaining the separating hyper-plane, with a maximum margin between the two classes, amounts to solving the maximization problem of Equation (5), given the constraints defined by Equations (6) and (7).

$$\max L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (5)$$

$$\sum_i \alpha_i y_i = 0 \quad (6)$$

$$0 \leq \alpha_i \leq C \quad (7)$$

Equation (5) is a quadratic programming (QP) problem in α_i , where α_i is called as Lagrange Multiplier, which is introduced to solve any constrained optimization problem. The variable x_i in Equations (5) represents the feature vectors obtained from the training data, while y_i represents the class labels associated with each of the training samples. For the two class learning problem of pedestrian classification, y_i is taken as +1 for positive training samples and -1 for negative training samples. Equation (5) can be solved efficiently using Sequential Minimal Optimization (SMO) algorithm, as proposed by John Platt [4]. The main advantage of SMO algorithm comes from the fact that it avoids the use of a time-consuming numerical QP optimization. It solves the optimization problem by choosing — in each iteration — a pair α_i and α_j to update next, using some heuristic. The heuristic employed is such that it tries to pick a pair α_i and α_j that will help make the

biggest progress towards the global maximum [4]. The SMO algorithm proceeds to update the original function $L(\alpha)$, by optimizing w.r.t. the pair α_i and α_j , while keeping other α_k 's (where $k \neq i, j$) constant. The Karush-Kuhn-Tucker (KKT) [23] conditions are checked to test for the convergence of the SMO algorithm. KKT conditions are the conditions that are used to solve problems with inequality constraints. Reference [23] gives detailed derivation of the KKT conditions as applicable to the SVM optimization problem of Equation (5), given the constraint defined by the inequality in Equation (7).

Upon solving Equation (5), only those values of α_i come out to be non-zero, which correspond to the training samples acting as support vectors for the separating hyper-plane. The equation of such a separating hyper-plane is given by Equation (8), where w corresponds to the normal to the desired hyper-plane and b is the minimum distance from the origin to the hyper-plane.

$$w^T x + b = 0 \quad (8)$$

$$w = \sum_i \alpha_i y_i x_i \quad (9)$$

$$f(u) = w^T u + b \quad (10)$$

The parameter w is calculated using Equation (9), while b is computed as the average of the b values corresponding to all the training examples, as explained in [23]. For a given unknown test vector (i.e. a test image segment) u , the sign of $f(u)$ as calculated using Equation (10) is used for classifying between the two trained classes i.e., if $f(u)$ is positive, then u belongs to class +1 or otherwise.

5. RESULTS

Initially, the HOG features were calculated for segments of pedestrian and non-pedestrian images. The HOG visualization in Figure 7 and Figure 8 is an approximate representation of the actual HOG descriptor. They show the most prominent edge corresponding to each cell for a pedestrian and a non-pedestrian image respectively. This edge information is derived from the most weighted gradient orientation bin corresponding to the histogram of each cell.



Figure 7: Image (right) showing most prominent edge corresponding to each cell for a pedestrian image (left [24]) using HOG feature



Figure 8: Image (right) showing most prominent edge corresponding to each cell for a non-pedestrian image (left [24]) using HOG feature

For better visualization of the edge information, the results shown in Figure 7 and Figure 8 were obtained using a cell-size of 8x8. However, while calculating the HOG features for the image segments which are input to the classifier, the cell-size employed is 16x16. This is done to reduce the amount of calculations involved in HOG computation, at the same time avoiding any degradation in the classifier performance.

The SMO algorithm based SVM classifier is implemented using the pseudo-code as proposed in [4]. For the training and testing of the classification stage, the CVC-02 Pedestrian Dataset [24] is employed. It was chosen because it provides a separate subset of database focused towards the different tasks involved in a PDWS, namely candidate generation, classification and overall system evaluation. Also, it replicates the cluttered background commonly seen by a PDWS in any urban scenario. For training of classifier, it provides 1016 cropped positive (or pedestrian) images and 7650 cropped negative (or non-pedestrian) images. A classifier trained with more training data would be better adept at accurately detecting a pedestrian. Hence to bring in more pedestrian pose scenarios the CVC-02 dataset employs mirror images of the positive dataset. The presented work uses 1000 pedestrian images along with their corresponding mirror images and 2000 non-pedestrian images for training of classifier. Figure 9 and Figure 10 show some example images from both the positive and the negative class of the CVC-02 Pedestrian Dataset. For testing, the presented work employs a total of 1138 pedestrian images, including their mirror image and an equal number of non-pedestrian images. All the images in the CVC-02 dataset are used after rescaling them to 120x48 image sizes.

Table 1 summarizes the important parameters pertaining to the training and testing of the classifier. To achieve a maximum-margin linear SVM classifier, the parameter ‘C’ in the table is used to control the degree of misclassification of some difficult examples in the training data. The best classifier performance is obtained when C value is taken as 0.09.

Table 1: Parameters employed in training and testing of classifier

Parameter	Value
Segment/Crop size	120 x 48
No. of training images	4000 (2000 positive and 2000 negative)
No. of positive test images	1138
No. of negative test images	1138
Cell-size	16x16
Feature vector length/cell	36
C (for controlled misclassification)	0.09



Figure 9: CVC-02 Pedestrian Dataset — Positive Training images

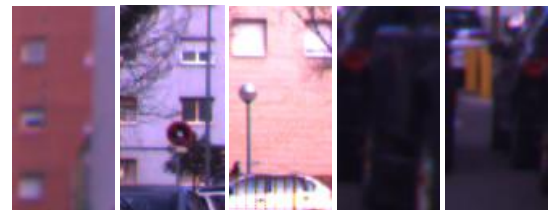


Figure 10: CVC-02 Pedestrian Dataset — Negative Training images



Figure 11: Misclassified pedestrians (False Negatives)

Out of the 1138 positive test images, 1079 cropped pedestrian images were classified as pedestrians by the implemented feature-classifier pair, amounting to a true positive classification accuracy of 95%. Figure 11 depicts some of the pedestrian test images which were not classified correctly by the implemented HOG-SVM pair. The primary reason for their misclassification could be the low resolution levels of these images, coupled with the low contrast between foreground and background regions. With 1138 negative images, the classifier yielded a false positives classification percentage of 10% i.e., 116 non-pedestrian images were misclassified as pedestrians. Table 2 gives a summary of the results obtained, while Figure 12 provides a graphical representation of the same.

Table 2: Classification Results

Classification	Total number of test images		Number of classified images		Percentage Classification
	Positive	Negative	True	False	
True Positives (TP) (Pedestrians classified as pedestrians)	1138	Not Applicable (NA)	1079	NA	95%
False Negatives (FN) (Pedestrians classified as non-pedestrians)	1138	NA	NA	59	5 %
True Negatives (TN) (Non-pedestrians classified as non-pedestrians)	NA	1138	1022	NA	90%
False Positives (FP) (Non-pedestrians classified as pedestrians)	NA	1138	NA	116	10%

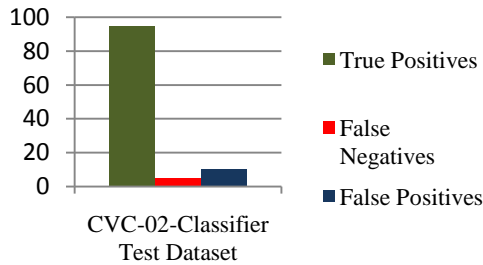


Figure 12: Graphical representation of classification results

Table 3: Classification performance parameters

Classifier Performance Parameters	Formula	Value	Percentage
Precision (Positive predictions that are correct)	$TP / (TP+FP)$	0.9	90%
Recall / Sensitivity (Positive labeled segments that were predicted as positive)	$TP / (TP+FN)$	0.95	95%
Specificity (Negative labeled segments that were predicted as negative)	$TN / (TN+FP)$	0.9	90%
Accuracy (Predictions that are correct)	$(TP+TN) / (TP + TN + FP+FN)$	0.92	92%

Table 3 evaluates the classifier performance using some commonly used metrics. From the graph in Figure 12 and the parameter values obtained in Table 3, it can be concluded that the implemented HOG-SVM feature-classifier pair shows promising results, with a high true positive classification accuracy and minimal number of misclassification as shown in Figure 11. The percentage of the pedestrians which were wrongly classified is represented in Figure 12 as ‘False negatives’. In order to reduce the number of non-pedestrian images from being classified as pedestrians, a short review of major verification strategies employed in the literature has been provided in [2]. Employing a verification strategy will help validate the classified pedestrian ROIs, while filtering out the false positives.

6. CONCLUSION AND FUTURE WORK:

In this paper, implementation of two key blocks in the pedestrian detection framework, namely feature extraction and classification are presented. HOG features are implemented for feature extraction, with a cell-size of 16x16 (for computational speed-up) and an efficient normalization strategy (for illumination invariance). A soft-margin linear SVM, based on the simple yet efficient SMO algorithm is used for implementation of the classification module. The classifier uses a subset of the CVC-02 Pedestrian Dataset, which is specifically aimed at the training and testing of the classification stage. The results obtained show high pedestrian classification accuracy (TP) of 95% and an overall classification accuracy (TP+TN) of 92%. Hence, the implemented feature-classifier ensemble can act as a fast and robust building block for a complete PDWS. However, to further improve the results of the presented work, measures such as increasing the amount of training data of the classifier and using a suitable verification strategy — to reduce the number of false positives incurred — can be undertaken. Future work can also be aimed at the implementation of an equally efficient segmentation approach, which would complement the presented feature-classifier modules.

7. REFERENCES

- [1] World Health Organization (WHO). 2013. Global Status Report on Road Safety 2013: Supporting a Decade of Action.

- [2] Geroñimo D., Loópez A. M., Sappa A. D., and Graf T., “Survey of Pedestrian Detection for Advanced Driver Assistance Systems”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010 .
- [3] Dalal N. and Triggs B. 2005. Histograms of Oriented Gradients for Human Detection. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*.
- [4] Platt J. 1999. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods—support vector learning* 3.
- [5] Yadav R. P., Senthilarasu V., Kutty K., Vaidya V., Ugale S. P. 2015 (in press). A Review on Day-Time Pedestrian Detection. *SAE 2015 World Congress*.
- [6] Hussain S. and Triggs B. 2010. Feature sets and dimensionality reduction for visual object detection. In *British Machine Vision Conference*.
- [7] Liu Y., Yao J., Xie R., and Zhu S. 2013. Pedestrian Detection from Still Images Based on Multi-Feature Covariances. In *Proceeding of the IEEE International Conference on Information and Automation*.
- [8] Jun B., Choi I., and Kim D, “Local Transform Features and Hybridization for Accurate Face and Human Detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.
- [9] Enzweiler M. and Gavrila D. M., “A Multilevel Mixture-of-Experts Framework for Pedestrian Classification”, *IEEE Transactions on Image Processing*, 2011.
- [10] Li B., Li Y., Tian B., Zhu F. H., Xiong G., Wang K. 2013. Part-based pedestrian detection using grammar model and ABM-HOG features. *IEEE International Conference on Vehicular Electronics and Safety*.
- [11] Wojek C. and Schiele B. 2008. A Performance Evaluation of Single and Multi-feature People Detection. G. Rigoll (Ed.): *DAGM Symposium*.
- [12] Wu B. and Nevatia R. 2007. Simultaneous Object Detection and Segmentation by Boosting Local Shape Feature Based Classifier. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.
- [13] Sabzmeydani P. and Mori G. 2007. Detecting Pedestrians by Learning Shapelet Features. In *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*.
- [14] Dalal N., Triggs B., and Schmid C. 2006. Human Detection Using Oriented Histograms of Flow and Appearance. In *Proceedings of European Conference on Computer Vision*.
- [15] Munder S. and Gavrila D.M., “An Experimental Study on Pedestrian Classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.
- [16] Felzenszwalb P., Girshick R., McAllester D. and Ramanan D., “Object Detection with Discriminatively Trained Part Based Models”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.
- [17] Tang Y., Liu W., Jianwei W. 2010. An Improved Pedestrian Detection Algorithm Based on AdaBoost Cascading Structure. In *Proceedings of the 8th World Congress on Intelligent Control and Automation*.
- [18] Cerri P., Gatti L., Mazzei L., Pigoni F. et al. 2010. Day and Night Pedestrian Detection Using Cascade AdaBoost System. In *IEEE Annual Conference on Intelligent Transportation Systems*.
- [19] XuY. W., Cao X. B., Qiao H., and Wang F. Y. 2006. A cascaded classifier for pedestrian detection. In *Proceedings of IEEE Intelligent Vehicle Symposium*.
- [20] Ouyang, W., Wang, X. 2013. Joint deep learning for pedestrian detection. In *ICCV*.
- [21] Tong W., Hong H., Fang H., Xie Q., et al. 2003. Decision Forest: Combining the Predictions of Multiple Independent Decision Tree Models. Published by the American Chemical Society.
- [22] Benenson R., Omran M., Hosang J., Schiele B. 2014. Ten years of pedestrian detection, what have we learned? *ECCV 2014*.
- [23] Mak G. 2000. The Implementation Of Support Vector Machines Using The Sequential Minimal Optimization Algorithm. Master's Thesis, McGill University.
- [24] Gerónimo D., Sappa A. D., Ponsa D. and López A. M. 2010. 2D-3D Based on-board pedestrian detection system. In special issue on Intelligent Vision Systems in *Computer Vision and Image Understanding*.