A comparative study of pedestrian detection methods using classical Haar and HoG features versus bag of words model computed from Haar and HoG features

Raluca Brehar
Computer Science Department
Technical University Of Cluj-Napoca
Email: Raluca.Brehar@cs.utcluj.ro

Sergiu Nedevschi
Computer Science Department
Technical University of Cluj-Napoca
Email: Sergiu.Nedevschi@cs.utcluj.ro

***Abstract*—The bag of words model has been actively adopted by content based image retrieval and image annotation techniques. We employ this model for the particular task of pedestrian detection in two dimensional images, producing this way a novel approach to pedestrian detection. The experiments we have done in this paper compare the behavior of discriminative recognition approaches that use AdaBoost on codebook features versus Adaboost trained on primitive features that may be extracted from a two dimensional image. By primitive features we refer in this paper to Haar features and Histogram of Oriented Gradients both being extremely used in object recognition in general and in pedestrian detection in particular. The conclusion of our experiments is that the codebook representation performs better than the primitive feature representation.**

I. Introduction

The detection of humans in still images and especially in traffic scenarios is an important problem for artificial vision and pattern recognition. A robust solution to this problem would have various applications to autonomous driving systems, video surveillance, image retrieval.

In general, the goal of pedestrian detection is to determine the presence of humans in images and videos and return information about their position. The problem of detecting pedestrians has a high degree of complexity because of the large intra-class variability, as pedestrians are highly deformable objects whose appearance depends on numerous factors:

- variability of appearance due to the size, color and texture of the clothes, or due to the accessories (umbrellas, bags etc) that pedestrians may carry;
- irregularity of shape: pedestrians may have different heights, weights ;
- variability of the environment in which they appear (usually pedestrians exist in a cluttered background in complex scenarios whose look is influenced by illumination or by weather conditions);
- variability of the actions they may perform and positions they may have (run, walk, stand, shake hands etc).

The inter and intra-class variability of pedestrians makes the task of a uniform representation extremely difficult. That is the width and height of the 2D image corresponding to perfectly framed pedestrians varies a lot. That is why the number of visual features that can be extracted for a pedestrian is not constant. Also, we may choose from a variety of features and as the number of features increases so does the complexity of the recognition algorithms.

In this paper we perform a study of two types of representations used for pedestrian detection:

- Representation based on primitive features: we have extracted relevant visual features in the field of pedestrian detection, namely Haar and Histogram of oriented Gradient (HoG).
- Representation based on codebook computed from primitive features.

We apply the same classification algorithm to both representation, for features computed on images in benchmark datasets. We notice that the codebook representation provides better detection results than the representation using primitive visual features.

The novelty of the paper resides in the application of the codebook representation for Haar and Histogram of Oriented Gradient features to the particular task of pedestrian detection in two dimensional images.

II. Related Work

The two-dimensional approaches to pedestrian detection scan the entire image space (process that can be very slow and prone to false detections). The development of 2D pattern analysis techniques for recognizing pedestrians has shown a strong progress. These pattern analysis techniques include methods to detect the pedestrian shape or walking motion.

Shape techniques rely on detecting spatial human features, and are the most commonly used methods. For example [1], [2], [3] used a shape finding method called the Chamfer system to find pedestrian shapes with a set of hand crafted image templates of pedestrians in different poses.

Haar wavelet transform (essentially multiple scaled edge detection) is used by [4], [5] to extract a pedestrian shape

representation. A Support Vector Machine is then trained to learn a model of pedestrians in a front/rear pose, producing strong results. [6] introduced gradient orientation histograms for pedestrian detection, and developed a recognition system using Support Vector Machines. In [7] the classification is performed based on the vertical symmetry that the human figure exhibits. [8] have developed very efficient cascade classifier to recognize image patterns. Originally created for face detection and then applied to pedestrian detection for surveillance cameras [9], these classifiers are trained by an exhaustive selection of the best weak classifiers, then combining these weak classifiers they form a strong classifier, with impressive results.

Motion techniques are used to detect human walking patterns in image sequences. [10] use a model of the human gait in various poses to detect temporal patterns which resemble walking pedestrians. However this method will not detect stationary pedestrians nor any pedestrian not moving across the camera's field of view.

The bag-of-words model has been used intensively in object recognition but few algorithms are implemented for the particular task of pedestrian detection in intensity images. For example [11] use the model for representing high-level concepts in images; the high-level concepts correspond to a vocabulary used for Content Based Image Retrieval. The BoW model is used by [12] to predict the presence of an object within an image and it helps to accurately segment instances of object classes in images without any human interaction.

III. Method Description and System Architecture

This paper makes a study of the behavior of AdaBoost classification algorithm with respect to the task of pedestrian detection in two-dimensional images. The study compares the detection rate of the classification when using primitive features (Haar, HoG) and when using the bag-of-words model for the same features. Figure 1 depicts our approach. The main steps are the following:

1) Extract primitive features from different datasets. By primitive features we mean: Haar wavelets, HoG features.
2) Randomly choose a number of positive images and generate for them the codewords. Then take all the images in the positive and negative training set and compute the extended codebook.
3) Feed the primitive features to a classification module (AdaBoost).
4) Send the codebooks for each feature to the classification module.
5) Compare the detection results of the previous two steps.

A. Datasets

We have used reference datasets in pedestrian detection in intensity images.

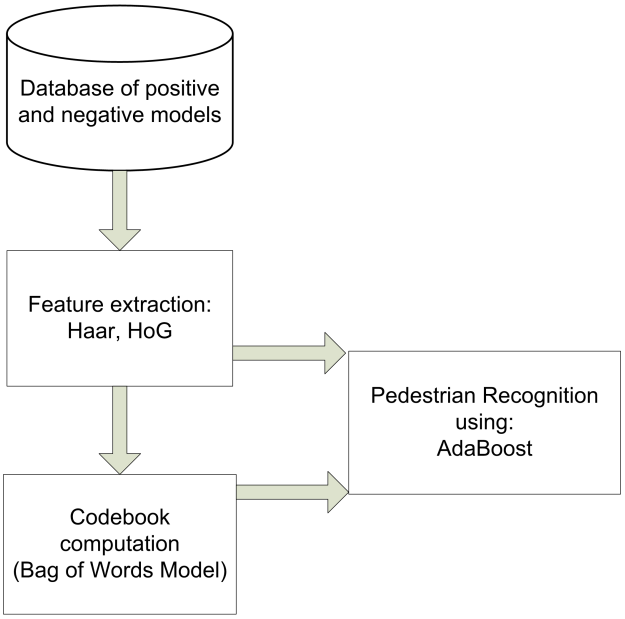

Fig. 1. Methodology: pedestrian detection based on primitive features and based on the bag-of-words model of the primitive features

1) Daimler: [1]
Pedestrian Classification Benchmark Dataset introduced in [13] contains three training sets, each consisting of 5000 negative samples and 4800 positive samples, hence a total of 15000 negatives and 14400 positives. All the images have a dimension of 18×36 pixels.

The database contains two test sets each being formed of 5000 negatives and 4800 positive images. Some samples from the database are depicted in Figure 2.

Fig. 2. Samples from Daimler Pedestrian Behchmark Dataset

2) NICTA Pedestrian Dataset: [2]
It is a relatively new benchmark dataset for pedestrian detection [14]. It contains positive and negative images at different resolutions (in our experiments we have used 16×40 and 64×80 image dimensions). "The final dataset contains 25551 unique pedestrians, allowing for a dataset of over 50 000 images with mirroring". Samples from the image data are presented in Figure 3.

[1] http://www.gavrila.net/Research/Pedestrian_Detection/Daimler_Pedestrian_Benchmarks/daimler_pedestrian_benchmarks.html

[2] http://www.nicta.com.au/research/projects/AutoMap/computer_vision_datasets

Fig. 3. Sample images from NICTA pedestrian dataset

We may notice that the datasets contain color or grayscale images. In our work we have converted all images to grayscale and we have extracted the features on this color representation.

B. Feature extraction

1) Haar: Haar-like features have been used by [15] for object detection (faces and pedestrians). For computing Haar-like features one has to consider adjacent rectangular regions at a specific location in an image, compute the sum of pixel intensities in these regions and calculate the difference between the sums. These features have been divided into

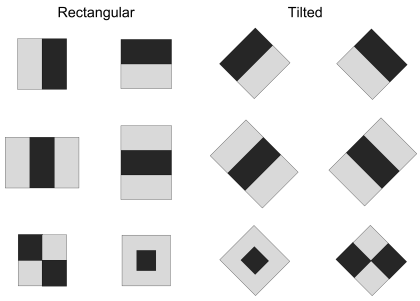

Fig. 4. Examples of Haar features

rectangular features and tilted features. Figure 4 depicts some of them.

We have worked with rectangular features. All Haar-like features take advantage of a fast computation due to the integral image representation[15]. The integral image, denoted $ii(x, y)$, at location (x, y) contains the sum of the pixel values above and to the left of (x, y) where $i(x, y)$ is the input image.

$$ii(x, y) = \sum_{x' \leq x; y' \leq y} i(x', y') \tag{1}$$

The integral image can be computed in one-pass over the image using the following recurrence relation:

$$s(x, y) = s(x, y - 1) + i(x, y) \tag{2}$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \tag{3}$$

where $s(x, y)$ denotes the cumulative row sum and $s(x, -1) = ii(-1, y) = 0$.

Figure 5 shows how to compute the value of the sum of the pixels from a rectangular region using the integral image representation. The relations are as follows:

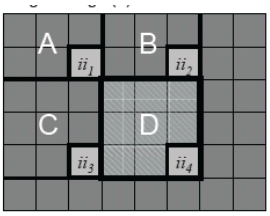

Fig. 5. Computing the sum of pixels in a rectangular region using the integral image

$$\begin{aligned} ii_1 &= \sum A; \\ ii_2 &= \sum A + \sum B \\ ii_3 &= \sum A + \sum C; \\ ii_4 &= \sum A + \sum B + \sum C + \sum D \\ \text{hence } \sum D &= ii_4 + ii_1 - ii_2 - ii_3 \end{aligned} \tag{4}$$

2) Histogram of Gradient Orientation Features HoG): HoG features have been introduced by [6] that propose an efficient algorithm for pedestrian detection. HoG features have a great advantage of being invariant to illumination conditions and they are based on the idea that local object appearance and shape within an image can be described by the distribution of edge directions.

The computation of HoG features involves the following steps:

1) Gradient computation that consists in filtering the color or intensity data of the image with the following kernels: $[-1, 0, 1]$ and $[-1, 0, 1]^T$. Using the filtering result one can compute the magnitude and orientation of each pixel in the image.
2) Orientation binning: consists in dividing the image into non-overlapping cells of equal dimension. Within each cell a histogram of orientations is computed. The vote of each pixel is weighted by the magnitude of the gradient. The cells can either be rectangular or radial in shape, and the histogram bins are evenly spread over 0 to 180 degrees or 0 to 360 degrees, depending on whether the gradient is "unsigned" or "signed".
3) Descriptor blocks: the cells are grouped into overlapping blocks in order to locally normalize the gradient.
4) Block normalization: within each block a normalization scheme is applied. We have used L2-Hys norm that provided best results for [6].

The final feature vector is formed of the histogram entries in each cell after performing the four steps previously mentioned.

C. AdaBoost Machine learning algorithm

The algorithm has been introduced by [16]. It is a meta-classification algorithm hence it can be used in combination with other learning algorithms in order to obtain better classification performance.

The general idea of the algorithm is as follows: consider $h_1, h_2, \ldots h_T$ a set of simple hypothesis and consider the

composite ensemble of hypothesis:

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{5}$$

There are many approaches for selecting the coefficients α_t and the base hypothesis h_t in equation 5. We have used the algorithm provided by [17]. The basic idea of the algorithm is presented in what follows:

1) **Input**: S=$(x_1, y_1), \ldots, (x_n, y_n)$, Number of iterations T
2) **Initialize:** $d_n^{(1)} = \frac{1}{N}$ for all $n = 1, \ldots, N$
3) for $t = 1, \ldots, T$ **do**
   a) Train classifier with respect to the weighted sample set $S, d^{(t)}$ and obtain hypothesis $h_t : x \to \{-1, +1\}$, i.e. $h_t = L(S, d^{(t)})$
   b) Calculate the weighted training error e_t of h_t:
   $$e_t = \sum_{n=1}^{N} d_n^{(t)} I(y_m \neq h_t(x_n)) \tag{6}$$
   c) Set:
   $$\alpha_t = \frac{1}{2} log \frac{1 - e_t}{e_t} \tag{7}$$
   d) Update the weights:
   $$d_n^{(t+1)} = \frac{d_n^{(t)} e^{-\alpha_t y_n h_t(x_n)}}{Z_t} \tag{8}$$
   where Z_t is a normalization constant such that $\sum_{n=1}^{N} d_n^{(t+1)} = 1$
4) **Break if:** $e_t = 0$ or $e_t \leq \frac{1}{2}$ and set $T = t - 1$
5) **Output:** $f_T(x) = \sum_{t=1}^{T} \frac{\alpha_t}{\sum_{r=1}^{T} \alpha_t} h_t(x)$

where x is the pattern to be classified, y is its target label and $f(x)$ is the decision function.

A weight $d^{(t)} = (d_1^{(t)}, \ldots, d_N^{(t)})$ is assigned to the data at step t and a weak learner h_t is constructed based on $d^{(t)}$. This weight is updated at each iteration. The weight is increased for the examples which have been misclassified in the last iteration.

The weights are initialized uniformly: $d_n^{(1)} = 1/N$.

To estimate if an example is correctly or badly classified, the weak learner produces a weighted empirical error defined by:

$$\epsilon_t(h_t, d^{(t)}) = \sum_{n=1}^{N} d_n^{(t)} I(y_n \neq h_t(x)_n) \tag{9}$$

Once the algorithm has selected the best hypothesis h_t , its weight $\alpha_t = \frac{1}{2} log \frac{1-\epsilon_t}{\epsilon_t}$ is computed such that it minimizes a loss function. One of the possible loss function considered in AdaBoost is:

$$G^{AB}(\alpha) = \sum_{n=1}^{N} e^{-y_n(\alpha h_t(x_n) + f_{t-1}(x_n))} \tag{10}$$

where f_{t-1} is the combined hypothesis of the previous iteration given by:

$$f_{t-1}(x_n) = \sum_{r=1}^{t-1} \alpha_r h_r(x_n) \tag{11}$$

The iteration loop is stopped if the empirical error ϵ_t equals 0 or $\epsilon_t \geq \frac{1}{2}$. If $\epsilon_t = 0$, the classification is optimal at this stage and so it is not necessary to add other classifiers. If $\epsilon_t \geq \frac{1}{2}$, the classifiers do not respect the weak condition anymore. They are not better than random selection so AdaBoost cannot be efficient.

Finally, all the weak hypotheses selected at each stage h_t are linearly combined as follows:

$$f_T(x) = \sum_{t=1}^{T} \frac{\alpha_t}{\sum_{r=1}^{T} \alpha_r} h_t(x) \tag{12}$$

The final classification is a simple threshold which determines if an example x_i is classified as positive or negative.

1) Weak learners: The adaptive boosting method calls the "weak" or "base" learning algorithm repeatedly, each time feeding it a different subset of training examples (actually, a different distribution or weighting over the training samples). Each time it is called, the base learning algorithm generates a new weak prediction rule, and after many rounds the boosting method must combine these weak rules into a single prediction rule, that hopefully, will be more accurate than any one of the weak rules.

For weak learners one can use:

- Bayes decision rule
- Decision trees having a root node and two children and using as splitting criteria:
  - Miss-classification error
  - Entropy
  - Gini index
- Other decision rules

D. Algorithm for codebook generation from a set of features

The Bag of Words (BoW) model has been introduced by natural language processing techniques and during the last years it has been used extensively in computer vision for the object recognition task.

To represent an image using BoW model, an image can be treated as a document. For the image context we need to define the "word" concept. This concept has different meanings and representations depending on the task we need to solve, on the images and on the features extracted for them.

Three main computational steps are employed by the bag-of-words model [18]:

1) feature detection: extract several local patches (or regions), which are considered as candidates for basic elements, "words".
2) feature description: each image is abstracted by several local patches. Feature representation methods deal with how to represent the patches as numerical vectors. These methods are called feature descriptors.
3) The final step is to convert vector represented patches to "codewords" that are representative for several similar patches. One simple method is performing K-means clustering over all the vectors.

We have modified the classic approach of the bag-of-words model by transforming the local patches into features. Our idea is to find the most representative features by clustering and then, for each image compute a histogram representation that stores the information about how many features are in the clusters. The steps of our algorithm are:

1) Randomly choose p images from the training data set.
2) For each image compute the features. The number of features may differ from image to image. We denote f_i the number of features computed for the i^{th} image.
3) Construct a large feature space by putting all the features for all images in a single feature vector.
4) Perform a supervised clustering using all the features of the large feature vector.
5) The features representing the centers of the clusters will be the codewords.

These steps are done for each class of objects, in our case for the Pedestrian class and for the NonPedestrian class.

The second major step consists in representing an image by its codebook. To obtain the codebook representation of an image we do the following:

- Compute all the features of the image;
- Find the cluster to which each feature belongs;
- Count how many features are in each cluster;

The final codebook representation feature vector associated to an image has a number of elements equal to the number of clusters and the value of the element at position i is given by the number of features that belong to cluster i for the given image.

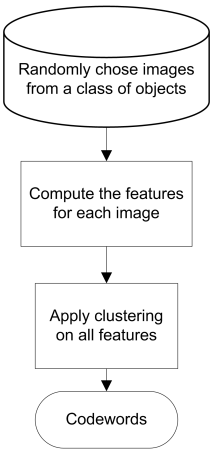

Fig. 6. Codeword generation for images in a given class

IV. Experiments and Results

For both features, Haar and HoG, and for both representations (codebook and primary) we have used the AdaBoost classification method. Our experiments have been done with the machine learning library, WEKA [3] [17]. The parameters of the ensemble learning algorithm were the following:

- Weak learner type: decision stump.
- Number of iterations: 10.

As clustering method for generating the codewords we have used k-means [17]. The general idea of k-means is that being given a set of observations $(x_1, x_2, \ldots, x_n)$, where each observation is a d-dimensional real vector, the algorithm tries to partition the n observations into k sets ($k \leq n$) $S = S_1, S_2, \ldots, S_k$ so as to minimize the within-cluster sum of squares (WCSS):

$$\text{argmin}_S \sum_{i=1}^{k} \sum_{x_i \in S_i} ||x_j - \mu_i||^2 \tag{13}$$

where μ_i is the mean of points in S_i.

A. Haar features

The first experiment comprised Haar features computed on Daimler dataset.

1) Classification results based on primitive Haar features: For Daimler dataset we have randomly chosen 5000 negatives images and 4800 positives images and we have created a classification model. The number of Haar features was equal to 840 for each image.

For the test set we have considered all the other images, that is: 20000 negatives and 19195 positives. The results are the following:

- Correctly Classified Instances = 32981 that is 84.1459 %;
- Incorrectly Classified Instances = 6214 that is 15.8541 % of the total number of samples;
- Kappa statistic = 0.6831;
- Mean absolute error = 0.2644;
- Root mean squared error = 0.3515;
- Relative absolute error = 52.9121 %;
- Root relative squared error= 70.3168 %;

The detailed accuracy by class is:

Class	Measure	Value
Pedestrian	TP Rate	0.866
Pedestrian	FP Rate	0.182
Pedestrian	Precision	0.82
Pedestrian	Recall	0.866
Pedestrian	ROC-Area	0.9
NonPedestrian	TP Rate	0.818
NonPedestrian	FP Rate	0.134
NonPedestrian	Precision	0.864
NonPedestrian	Recall	0.818
NonPedestrian	ROC-Area	0.9

TABLE I
PRIMITIVE HAAR DETAILED ACCURACY ON DAIMLER DATASET

The confusion matrix is given in table II.

# Pedestrians	#NonPedestrians	Classified as
16617	2578	Pedestrians
3636	16364	NonPedestrians

TABLE II
CONFUSION MATRIX FOR PRIMITIVE HAAR ON DAIMLER DATASET

[3]http://www.cs.waikato.ac.nz/ml/weka/

2) Classification results based on Haar codebook representation: For the codebook representation of Haar features extracted on Daimler dataset we have randomly chosen 500 positives and 500 negatives. We have used them for generating a bag of words model for the positive data and a bag of words model for the negative data. Each of the models contained 70 codewords (we have generated 70 clusters). Next we have randomly chosen other 4800 positive images that we have used for generating the codebooks of positives and we have picked randomly 5000 negatives for generating the codebook of negatives. All the remaining images have been used for creating the test data that contained: 19208 pedestrians and 19984 non-pedestrians. The results are much better than in the case of primitive Haar features:

- Correctly Classified Instances = 39148 that is 99.8877 %;
- Incorrectly Classified Instances = 44 that is 0.1123 % of the total instances;
- Kappa statistic = 0.9978;
- Mean absolute error = 0.0014;
- Root mean squared error = 0.0259;
- Relative absolute error= 0.2776 %;
- Root relative squared error = 5.1845 %;

The detailed accuracy by class is given in table III:

Class	Measure	Value
Pedestrian	TP Rate	0.999
Pedestrian	FP Rate	0.001
Pedestrian	Precision	0.999
Pedestrian	Recall	0.999
Pedestrian	ROC-Area	1
NonPedestrian	TP Rate	0.999
NonPedestrian	FP Rate	0.001
NonPedestrian	Precision	0.999
NonPedestrian	Recall	0.999
NonPedestrian	ROC-Area	1

TABLE III
CODEBOOK HAAR DETAILED ACCURACY ON DAIMLER DATASET

The confusion matrix is show in table IV.

# Pedestrians	#NonPedestrians	Classified as
19180	16	Pedestrians
28	19968	NonPedestrians

TABLE IV
CONFUSION MATRIX FOR CODEBOOK HAAR ON DAIMLER DATASET

The second experiment comprised Haar features computed on NICTA pedestrian dataset.

3) Classification results based on primitive Haar features: For NICTA training set we have considered 7000 positive images and 7000 negative images having a resolution of 16×40 pixels. The number of features extracted for each image equals 1152. For testing we have used 13785 pedestrian images and 23100 negative images. The classification results are as follows:

- Correctly Classified Instances = 32605 that is 88.3221 %;
- Incorrectly Classified Instances = 4311 that is 11.6779 % of the total instances;
- Kappa statistic = 0.7509;
- Mean absolute error = 0.1851;
- Root mean squared error = 0.291;

The detailed accuracy by class is provided in table V:

Class	Measure	Value
Pedestrian	TP Rate	0.84
Pedestrian	FP Rate	0.091
Pedestrian	Precision	0.999
Pedestrian	Recall	0.848
Pedestrian	ROC-Area	0.949
NonPedestrian	TP Rate	0.909
NonPedestrian	FP Rate	0.16
NonPedestrian	Precision	0.999
NonPedestrian	Recall	0.904
NonPedestrian	ROC-Area	0.949

TABLE V
PRIMITIVE HAAR DETAILED ACCURACY ON NICTA DATASET

The confusion matrix is provided by table VI.

# Pedestrians	#NonPedestrians	Classified as
11695	2221	Pedestrians
2090	20910	NonPedestrians

TABLE VI
CONFUSION MATRIX FOR PRIMITIVE HAAR ON NICTA DATASET

4) Classification results based on Haar codebook representation: For the codebook representation on the NICTA dataset we have randomly chosen 1000 positives and 1000 negatives with which we have generated the centers of the clusters. Then, the codebook representation for the training set was formed of 8000 positive images and 8000 negative images. The evaluation was done on a set that contained 13915 positives and 22999 negatives.

The results are:

- Correctly Classified Instances = 36882 (99.9133 %),
- Incorrectly Classified Instances = 32 (0.0867 %),
- Kappa statistic = 0.9982;
- Mean absolute error = 0.0012;
- Root mean squared error = 0.0255;
- Relative absolute error = 0.2331 %;
- Root relative squared error = 5.1203 %

The detailed accuracy by class is depicted in table VII:
The confusion matrix is displayed in table VIII.

B. Hog features

For Histogram of Gradient orientation features we have used the standard parameters of computation:

- cell size of dimension (8×8)
- block size of dimension (16×16)
- block stride of 8×8
- unsigned gradient representation
- L2Hys normalization

Class	Measure	Value
Pedestrian	TP Rate	0.999
Pedestrian	FP Rate	0.001
Pedestrian	Precision	0.998
Pedestrian	Recall	0.999
Pedestrian	ROC-Area	1
NonPedestrian	TP Rate	0.999
NonPedestrian	FP Rate	0.001
NonPedestrian	Precision	1
NonPedestrian	Recall	0.999
NonPedestrian	ROC-Area	1

TABLE VII
CODEBOOK HAAR DETAILED ACCURACY ON NICTA DATASET

# Pedestrians	#NonPedestrians	Classified as
13908	7	Pedestrians
25	22974	NonPedestrians

TABLE VIII
CONFUSION MATRIX FOR CODEBOOK HAAR ON NICTA DATASET

As the Daimler dataset contains small images ($18x36$) the number of HoG descriptors having standard parameters is relatively small. That is why we have concentrated our experiments with HoG features on NICTA database working with images of dimension 64×80.

1) Classification results based on primitive HoG features: For the training set we have used 8000 positive images and 8000 negative images. For each image we have extracted 144 features. For testing we have used 16413 pedestrian images and 20503 non-pedestrian images.

The obtained results are as follows:

- Correctly Classified Instances = 30923 (83.7658 %),
- Incorrectly Classified Instances = 5993 (16.2342 %);
- Kappa statistic = 0.6662;
- Mean absolute error = 0.2136;
- Root mean squared error = 0.3352;

The detailed accuracy by class is shown in table IX:

Class	Measure	Value
Pedestrian	TP Rate	0.874
Pedestrian	FP Rate	0.185
Pedestrian	Precision	0.741
Pedestrian	Recall	0.874
Pedestrian	ROC-Area	0.922
NonPedestrian	TP Rate	0.815
NonPedestrian	FP Rate	0.126
NonPedestrian	Precision	0.915
NonPedestrian	Recall	0.815
NonPedestrian	ROC-Area	0.922

TABLE IX
PRIMITIVE HOG DETAILED ACCURACY ON NICTA DATASET

The confusion matrix is displayed in table X.

2) Classification results based on HoG codebook representation: For the codebook representation we have randomly chosen 1000 positives and 1000 negatives for which we have generated the clusters. The number of clusters is equal to 60. For computing the codebook representation of the training set we have chosen 8000 positives and 8000 negatives, while for testing we have analyzed 13825 positives and 23089 negatives.

# Pedestrians	#NonPedestrians	Classified as
12168	1748	Pedestrians
4245	18755	NonPedestrians

TABLE X
CONFUSION MATRIX FOR PRIMITIVE HOG ON NICTA DATASET

The classification results are:

- Correctly Classified Instances = 35404 (95.9094 %);
- Incorrectly Classified Instances = 1510 (4.0906 %);
- Kappa statistic = 0.9128;
- Mean absolute error = 0.0601;
- Root mean squared error = 0.175 ;
- Relative absolute error = 12.0399 %;
- Root relative squared error = 35.0631 %.

The detailed accuracy by class is provided in table XI:

Class	Measure	Value
Pedestrian	TP Rate	0.943
Pedestrian	FP Rate	0.031
Pedestrian	Precision	0.949
Pedestrian	Recall	0.943
Pedestrian	ROC-Area	0.992
NonPedestrian	TP Rate	0.969
NonPedestrian	FP Rate	0.057
NonPedestrian	Precision	0.965
NonPedestrian	Recall	0.969
NonPedestrian	ROC-Area	0.992

TABLE XI
CODEBOOK HOG DETAILED ACCURACY ON NICTA DATASET

The confusion matrix is shown in table XII.

# Pedestrians	#NonPedestrians	Classified as
13115	800	Pedestrians
710	22289	NonPedestrians

TABLE XII
CONFUSION MATRIX FOR CODEBOOK HOG ON NICTA DATASET

A general conclusion that can be drawn from the experiments we have performed is that, in terms of accuracy, the codebook representation overcomes the representation based on primitive features. Another advantage of the codebook is the dimension of data space that is much smaller and the classification algorithms work faster. Nevertheless, for a test image we still need to compute all the features in order to generate the codebook, hence the feature computation time is not reduced.

V. CONCLUSIONS AND FUTURE WORK

The paper has presented a study and experiments on the pedestrian detection task for features extracted from two-dimensional images. We have approached two directions: (a) detection based on Haar and HoG features and (b)detection implemented on top of the codebook representation of Haar

and HoG features. The codebook representation is a novel approach in pedestrian detection and we have shown that it has better performance than the usual representation.

As future work we propose the extension of the method to the application on images that have a larger dimension than the training models, the introduction of other features (for example SIFT and SURF) and even the application of more complex classification algorithms.

REFERENCES

[1] D. Gavrila and V. Philomin, "Real-time object detection for smart vehicles," in *International Conference on Computer Vision*, September 1999.

[2] D. Gavrila, "Pedestrian detection from a moving vehicle," in *European Conference on Computer Vision*, 2000.

[3] S. Munder and D. Gavrila, "Multi-cue pedestrian detection and tracking from a moving vehicle," in *International Journal of Computer Vision*, 2006.

[4] T. Evgeniou, C. Papageorgiou, and T. Poggio, "A trainable pedestrian detection system," in *IEEE Intelligent Vehicle Sympossium*, 1998.

[5] C. Papageorgiou and T. Poggio, "Trainable pedestrian detection," in *International Conference on III Pattern*, 1999.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision & Pattern Recognition*, C. Schmid, S. Soatto, and C. Tomasi, Eds., vol. 2, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005, pp. 886–893. [Online]. Available: http://lear.inrialpes.fr/pubs/2005/DT05

[7] A. Fascioli, A. Broggi, M. Bertozzi, and M. Sechi, "Shape-based pedestrian detection," in *IEEE Intelligent Vehicle Symposium*, 2004.

[8] M. Jones and P. Viola, "Rapid object detection using a boosted cascade of simple features," in *Conf. Computer Vision and Pattern Recognition*, 2001.

[9] D. Snow, P.Viola, and M. Jones, "Detecting pedestrians using patterns of motion and appearance," in *International Journal of Comupter Vision*, 2005.

[10] C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, and W. von Seelen, "Walking pedestrian recognition," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 3, September 2000, pp. 155–163.

[11] J. Vogel and B. Schiele, "On performance characterization and optimization for image retrieval," in *Computer Vision - ECCV 2002*, ser. Lecture Notes in Computer Science, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds. Springer Berlin Heidelberg, 2006, vol. 2353, pp. 51–55.

[12] D. Larlus, J. Verbeek, and F. Jurie, "Category level object segmentation by combining bag-of-words models with dirichlet processes and random fields," *Int. J. Comput. Vision*, vol. 88, pp. 238–253, June 2010. [Online]. Available: http://dx.doi.org/10.1007/s11263-009-0245-x

[13] S. Munder and D. M. Gavrila, "An experimental study on pedestrian classification," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 28, no. 11, pp. 1863–1868, 2006.

[14] G. Overett, L. Petersson, N. Brewer, L. Andersson, and N. Pettersson, "A new pedestrian dataset for supervised learning," in *Intelligent Vehicles Symposium 2008 IEEE*, June 2008, pp. 373 – 378.

[15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," 2001, pp. 511–518.

[16] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, pp. 119–139, August 1997. [Online]. Available: http://portal.acm.org/citation.cfm?id=261540.261549

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, November 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[18] L. Fei-fei, "A bayesian hierarchical model for learning natural scene categories," in *In CVPR*, 2005, pp. 524–531.