# Word Channel Based Multiscale Pedestrian Detection Without Image Resizing and Using Only One Classifier

Arthur Daniel Costea and Sergiu Nedevschi
Image Processing and Pattern Recognition Group (http://cv.utcluj.ro)
Computer Science Department, Technical University of Cluj-Napoca, Romania
{arthur.costea, sergiu.nedevschi}@cs.utcluj.ro

## Abstract

*Most pedestrian detection approaches that achieve high accuracy and precision rate and that can be used for real-time applications are based on histograms of gradient orientations. Usually multiscale detection is attained by resizing the image several times and by recomputing the image features or using multiple classifiers for different scales.*

*In this paper we present a pedestrian detection approach that uses the same classifier for all pedestrian scales based on image features computed for a single scale. We go beyond the low level pixel-wise gradient orientation bins and use higher level visual words organized into Word Channels. Boosting is used to learn classification features from the integral Word Channels.*

*The proposed approach is evaluated on multiple datasets and achieves outstanding results on the INRIA and Caltech-USA benchmarks. By using a GPU implementation we achieve a classification rate of over 10 million bounding boxes per second and a 16 FPS rate for multiscale detection in a 640×480 image.*

## 1. Introduction

Pedestrian detection represents a key problem in computer vision due to its wide range of applications: automotive industry, robotics, surveillance, semantic image annotation. The main challenge is to find an approach that is highly accurate and fast at the same time.

Detecting pedestrians in real life scenarios, such as traffic scenes, can be difficult. Pedestrians wear clothes or accessories with complex textures, have different attitudes and postures, and can be occluded by obstacles or other pedestrians. The size of pedestrians in an image depends on their distance from the camera. Current best performing approaches still struggle to detect far pedestrians.

Sliding window based pedestrian detection appears to be the most promising for low to medium resolution settings, in comparison to segmentation or keypoint based methods [12]. The sliding windows are classified into pedestrian or non-pedestrian based on image features. The image features should capture the required information for classification, while allowing fast computation.

Previous object detection approaches use a fixed size sliding window and resize the image [8] or use a fixed size image and resize the sliding window [29]. When using multiple sliding window scales, individual classifiers are trained for different scales. In this paper we propose a solution to pedestrian detection that does not require image resizing and uses only one classifier for all sliding window scales. The proposed approach introduces the use of word channels, inspired from codebook based semantic image annotation techniques for extracting classification features.

## 2. Related work

The literature on pedestrian detection is extensive. Valuable surveys exist, such as [12, 14, 17], presenting and comparing the most relevant approaches of the last decade. These approaches are evaluated on pedestrian benchmarks. The most frequently used benchmark is the INRIA dataset, currently dominated by methods based on the Integral Channel Features approach proposed by Dollar *et al*. [10]. Ten integral images are constructed: six for gradient orientations at different angles, one for the gradient magnitude and three for the LUV color channels. These channels are used to learn classification features for a boosting based classifier. A classification feature is defined by a rectangle having a specific size and position in the detection window and it is represented by the sum of responses on one of the channels inside that rectangle. The classification features are learned by the boosting algorithm from a total of 30000 random features. Benenson *et al*. [4] obtained better classification results using all 718 080 possible rectangles for a 64×128 pixel model. Pedestrians are detected at multiple scales by learning a single classifier for pedestrians with heights of 96 pixels. The input image is then repeatedly resized to correctly detect all pedestrians with one sliding window.

Computing the image features at each scale can be time consuming due to the high number of potential image scales (around 50). Dollar *et al*. compute in [11] the image

features only once every half octaves (five in an image) and approximate for the rest of the scales.

An innovative idea was the transfer of resize operations to training time, proposed by Benenson *et al*. in [3]. Instead of using one classifier and multiple image octaves, a single image scale is used with multiple classifiers for each octave. By using a GPU implementation the approach achieved a detection rate of 50 FPS. An even higher detection rate is achieved using stereo information due to the reduction of search space.

Another popular pedestrian detection benchmark is the Caltech dataset [9]. It consists of approximately 10 hours of video taken from a vehicle in an urban environment. The dataset is challenging due to the small size of pedestrians and different occlusion cases. Even if the evaluation is performed only on pedestrians of 50 pixel heights or taller, and who have a maximum occlusion of 35%, the best performing methods achieve a miss rate of around 40% at a precision of 1 false positive per 10 frames. Best performing methods use pedestrian context [5, 32] or multiresolution deformable part models [32] for achieving the lowest miss rates. However, they have a slow execution time (around 1 second per frame, or even more).

Our method is also related to the semantic image annotation and segmentation domain. Annotations refer to the context (scene) of the image or to the presence of several image concepts (objects, materials, action). The visual codebook or "bag of words" model is a powerful tool to construct global image descriptors. The visual codebook consists of visual words obtained through training. An image is regarded as a collection of these words and the histogram of words is used as a descriptor vector Multiple methods, such as [6, 26], achieved outstanding results on the Pascal VOC Challenge [15] on the classification task using visual codebooks.

The Pascal VOC Challenge had also a semantic segmentation task. Top performing methods, like [18] and [19], used visual codebooks for multiple descriptor types. Using a trained classifier, individual pixels were classified as one of the semantic classes based on the surrounding visual words. A multiclass boosting classifier was trained with millions of pixel samples using visual word based classification features. Decision stumps were used as weak classifiers over visual word counts in learned rectangles. The results from individual pixel classifications were integrated as first order, unary potentials into a conditional random field which was then extended with multiple higher order potentials.

The robustness of the visual codebook based features in semantic image annotation techniques inspired us to use them for pedestrian detection. The main disadvantage of these descriptors is the high computational time required, suggesting why visual codebooks are not so popular in real time applications. The most time consuming process is the matching of all individual local image descriptors to the most similar word from the codebook. In this paper we show that it is possible to achieve real time performances using codebooks with lower word counts for descriptor types having lower dimensionality.

## 3. Image representation

Prior to classifying any rectangular region as pedestrian or non-pedestrian, the raw image data have to be transformed into potential classification features. Inspired from semantic image annotation techniques we propose the use of visual codebooks, also known as dictionaries, for representing images as distributions of visual words. A different codebook is trained for each individual local descriptor type.

### 3.1. Local descriptors

In our approach we use three descriptor types that can be computed densely at each pixel position. The first one is based on Dalal's and Triggs's HOG descriptor [8], the second one on Local Binary Pattern (LBP) and the third one on the LUV color channels. Our main focus is achieving fast computational time and reduced descriptor dimensionality.

Prior to computing the local descriptors, a Gaussian filter with $\sigma$=0.25 is used over the grayscale image for HOG and LBP and $\sigma$=1.0 over the color image in order to attenuate the image noise.

The HOG descriptor consists of several histograms of oriented gradients computed for multiple overlapping blocks inside a bounding box. We use one HOG block as a local descriptor because of fast computational time and reduced feature vector dimensionality constraints. We divide the 16x16 pixel block into four 8x8 pixel cells and for each cell we compute a histogram of oriented gradients using 6 orientation bins in the 0°–180° angle range (contrast insensitive). By concatenating the four histograms we obtain a 24 dimensional feature vector. This configuration provided good results in [8].

The basic LBP transform assigns an 8 bit value for each pixel. The raw intensity value of a pixel is compared to the intensity values of the 8 neighbor pixels. If the neighbor pixel has a higher value the corresponding bit is assigned "1", otherwise "0". Instead of using only 8 neighbors we use a larger 5×5 pixel neighborhood and compare the center pixel to the 24 surrounding pixels. A 24 dimensional descriptor vector consisting of '1's and '0's is then obtained.

In the experiments of Dollar *et al*. with *Integral Channel Features* [10] the LUV color channels, were strong and consistent cues in the face region for the detection of pedestrians. We use the LUV color channel values as pixel level feature vectors.

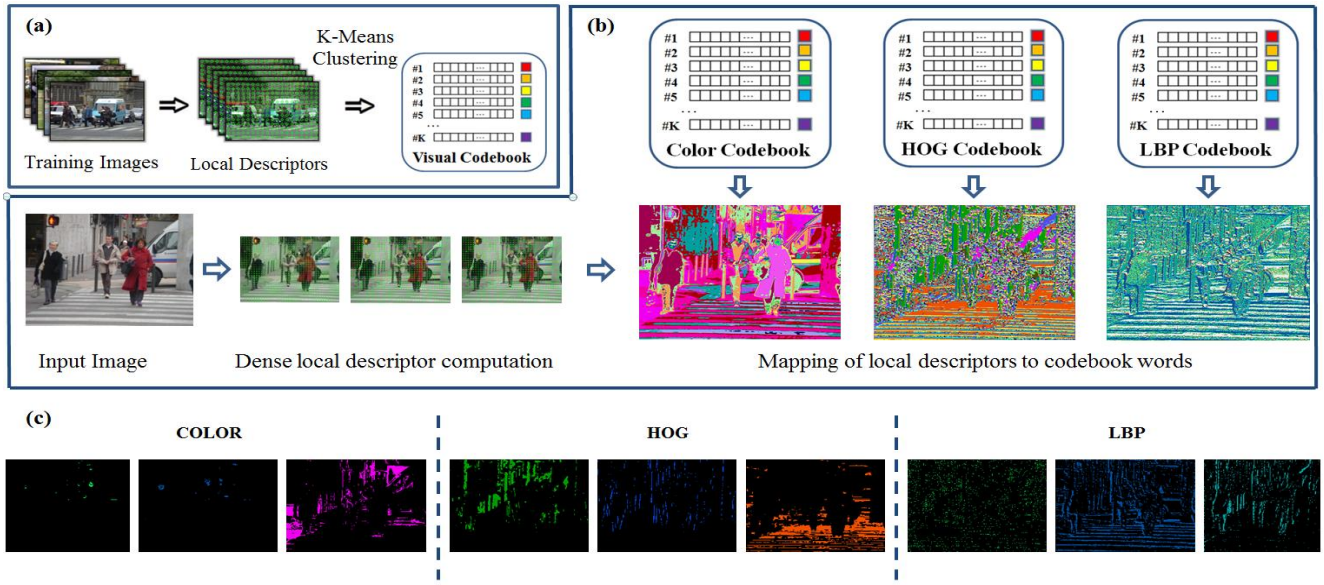


Figure 1: Generating *word channels* for an input image. (a) Training a *visual codebook* for one descriptor type; (b) Computation of *word maps* by matching dense local descriptors to visual codebooks for three different descriptor types; (c) A few examples of *word channels* extracted from the *word map* of each descriptor type.

### 3.2. Visual codebook

A visual codebook is defined as a set of local image descriptors, representing the most frequent and most characteristic responses. Initially this model was used for text classification [1], where the text was considered a "bag of words". The model was adapted to images, by considering them a set of visual words [7]. Applications include image classification and image segmentation. Any local descriptor in an image can be matched to one of the visual words from a learned codebook based on a similarity metric, such as the Manhattan or Euclidean distance. Only the codebook reference is therefore used in further processing rather than the whole descriptor vector.

A visual codebook can be constructed using any clustering algorithm. The most frequently used approach is K-Means clustering. The codebook consists of the resulting centroids. A different codebook is trained for each local descriptor type using a set of training images. Local descriptors are densely sampled from each image in order to obtain a large set of responses for clustering. This process is illustrated in Figure 1(a). For each used descriptor type an individual codebook is built. The clustering of 500000 descriptor samples into 64 clusters takes around 5 minutes using a CPU implementation.

### 3.3. Codebook maps and word channels

In our approach the local image descriptors are sampled at each individual image pixel. By matching each of these local descriptors to the most similar codebook word, we obtain the *codebook map* (Figure 1(b)). Any image region can therefore be easily described by the distribution of words in the map. The map can be decomposed into channels for each individual word. We call these channels w*ord channels* which are similar to the texton maps for texture features [28]. The visual codebooks capture specific properties of descriptor types in the context they were trained on. For an input image the word channels represent the distribution of these particular codebooks' words (Figure 1(c)). For computational efficiency we build integral images for each of the word channels.

It is difficult to choose the ideal number of words for a visual codebook. For visual categorization approaches based on codebooks, a higher number of words provide better results, however at higher computational costs. Common choices are around 100 to 1000 words [18, 19, 20, 26]. For a reduced context such as pedestrian detection (compared to general visual recognition) a lower word count can be used. In our experiments we successfully used 64 words for each of the three descriptor types resulting in a total of 192 word channels. The number of words was chosen intuitively in order to reduce computational costs. The analysis of the ideal number of words for each individual descriptor type is left for future work. The GPU based implementation allows the computation of all the 192 word channels in 12 ms (83 FPS). Each local descriptor is computed on an individual GPU thread and matched in the same thread with the codebook (the local descriptor is never saved on the GPU device memory). Because of the reduced size, the codebook can be cached in the shared memory of the GPU, making the access very fast.

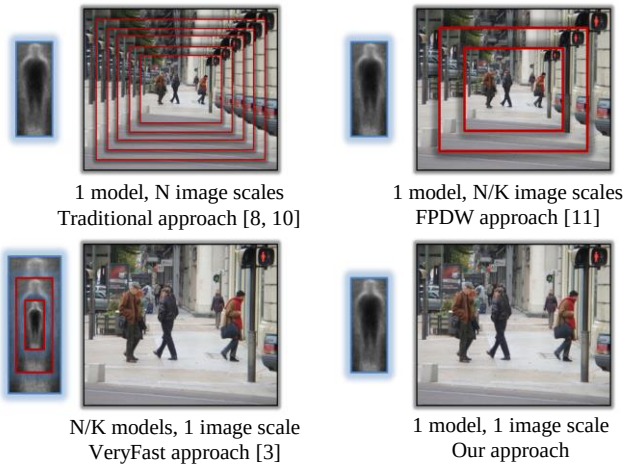


Figure 2: Different approaches for multiscale detection

## 4. Multiscale detection

The main implementation choices for multiscale pedestrian detection were presented in the related work section and are summarized in Figure 2. We propose an approach that uses image features computed at a single scale. The pedestrians are detected by scanning the fixed size image with sliding windows at different scales, but using a single classifier model for all scales. To our knowledge this is the first time when such an approach is used for multiscale detection. This is possible due to the scale independent classification features.

Different code words would be activated for a near pedestrian compared to a far pedestrian. However, the features are invariant to smaller scale changes, because of the relatively small codebook size. 64 visual words are used to represent a very large range of possible visual responses for pedestrians, vehicles, buildings and different background textures. If the scale of the pedestrian changes from 60 pixels height to 70 pixels, there is a high probability that specific visual codes, representing pedestrian structures or textures, will remain unaltered. For larger scale changes, the visual codes would probably not match, even if a small number of codes is used, but this issue is solved by the inclusion of various pedestrian sizes in the training set.

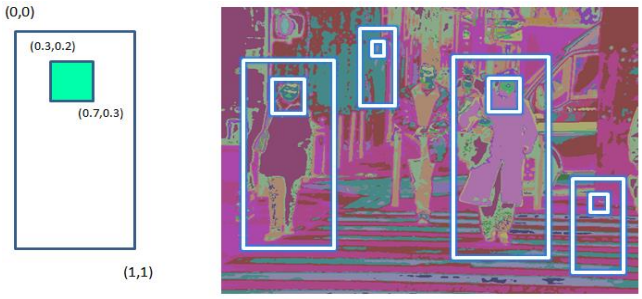


Figure 3: Left: Example of a classification feature defined by a word from the color codebook specific for pedestrian faces and a rectangle normalized to the detection window. Right: the corresponding regions in the color codebook map for different detection windows.

### 4.1. Classification features

The classification features are based on the integral images computed on word channels. The sum of the responses inside a rectangle for a specific word channel represents the count of that word. We choose to classify the bounding boxes based on the distribution of the codebook words which can be achieved by counting specific words in specific rectangles. In order to make the classification features independent of the bounding box size, we use normalized rectangles and normalized word counts. The rectangles are normalized to the bounding box size and the word counts are normalized to the rectangle size. Figure 3 illustrates a normalized classification feature and its use for different bounding boxes in an image. A similar normalization strategy was used by Wang *et al*. in [30] to handle scales and aspect ratios for object detection.

We need to generate a pool of potential classification features from which a feature selection algorithm can select the relevant features for classification. For this we need to specify a set of rectangles. Benenson *et al*. trained a pedestrian detector in [4] based on channel features using all 718 080 possible rectangles for a 64x128 pixel model (with a shrinking factor of 4). The training process took 2.5 days on a GPU enabled, multi-core, large RAM work-station. The detector performed significantly better compared to a detector that was trained with 30000 random rectangle features. We want to use fewer, but more relevant rectangles, considering the higher number of channels in the case of word channels. Benenson *et al*. provided informative statistics in [4] over the rectangles most frequently selected as classification features. Inspired from these statistics we also generate all possible rectangles for a detection window divided into 18 rows and 9 columns. However, we only use the rectangles that have an aspect ratio of 0.33 (vertical bars), 1 (squares) or 3 (horizontal bars) and an area between 3 and 12 cells, resulting in 556 rectangles.

The relevant classification features are learnt using boosting. Decision stumps over classification features are used as weak classifiers. A weak classifier can be learnt by evaluating all classification features using different values for decision stump thresholds and selecting the one that performed best on the training data. As in [28], in our experiments we evaluated only 1% of the classification features that were sampled randomly at each boosting round. This strategy reduces significantly the training time

without affecting the classification performance if the number of boosting rounds is high (5000 boosting rounds were used for a sampling rate of 0.5% in [28]).

Each boosting round learns a single weak classifier. Selecting the ideal number of boosting rounds is a delicate issue. Too many boosting rounds result in classifier overfitting, while too few boosting rounds result in underfitting. In our experiments we train new weak learners until the classification error on the training set reaches 0 and add 10% additional rounds. Even if the classification error over the training set reaches 0, the classification error over a validation set can still decrease with additional boosting rounds learnt using weighted training samples.

## 4.2. Classification scheme

Because the classification features do not depend on the bounding box size only one classifier is learnt. In order to obtain a robust classifier, it is important to have a training set that has many pedestrians at each scale. We train a cascade of multiple boosted classifiers.

For each of the classifiers in the cascade the positive train samples consist of pedestrian bounding boxes extracted at their original scale and use four times as many negative samples. For the first classifier in the cascade the negative samples are generated randomly from images containing no pedestrians. The following classifiers use the false positives of the previous classifier. In our experiments after 4-5 classifiers in cascade there were no false positives detected on the training set. To avoid overfitting we used 3 classifiers trained on hard negatives from images without pedestrians for the final cascade.

Two frequent cases of false detection on images with pedestrians were small bounding boxes detected inside the pedestrian or too large bounding boxes around the pedestrian. To solve these issues the classifier cascade was extended with two more classifiers. The first one was trained with false detections obtained on training images with pedestrians, where the bounding boxes were much larger and the second one was for too small bounding boxes inside the pedestrian.

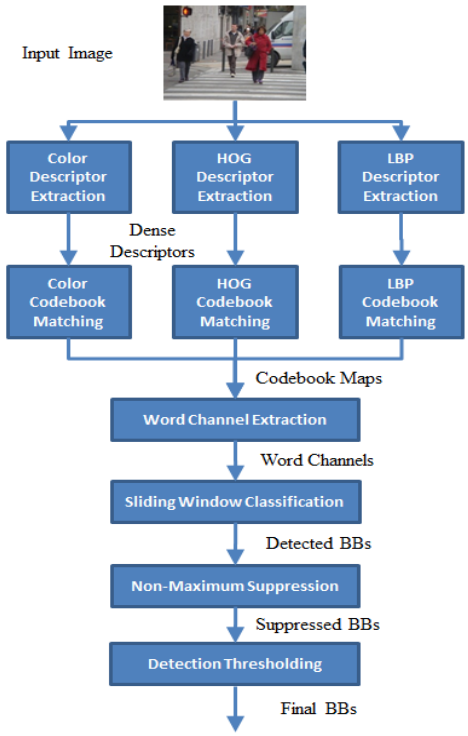


Figure 4: Pedestrian detection flow

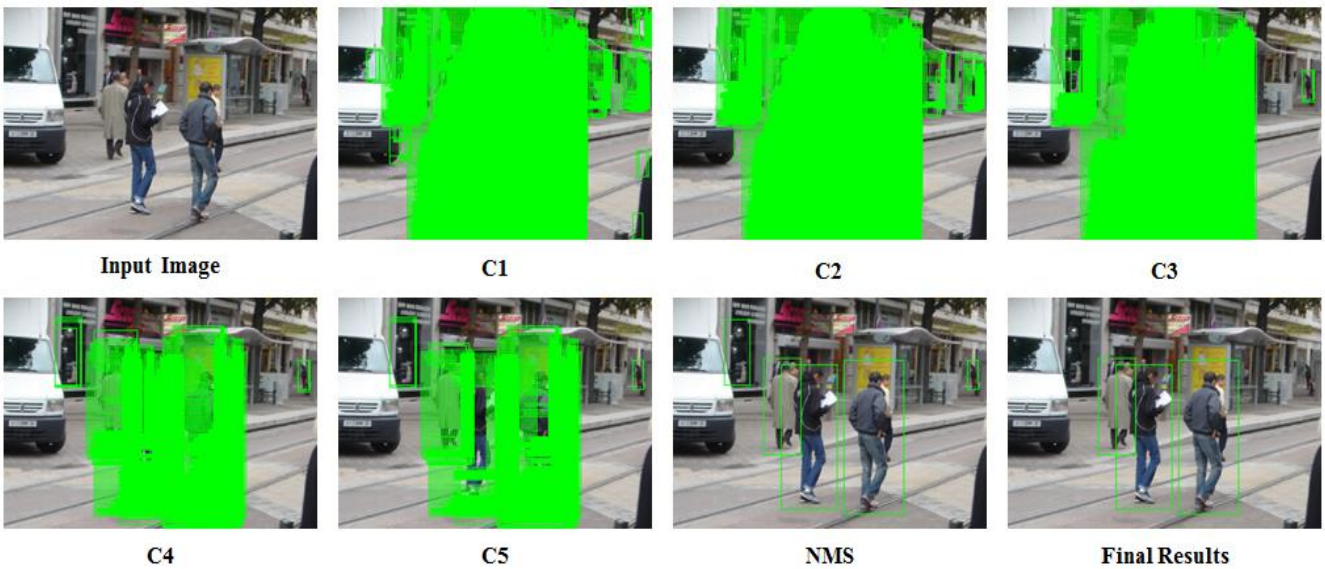


Figure 5: Pedestrian detection: classification cascade, non-maximum suppression and rejection of detections with low confidences.

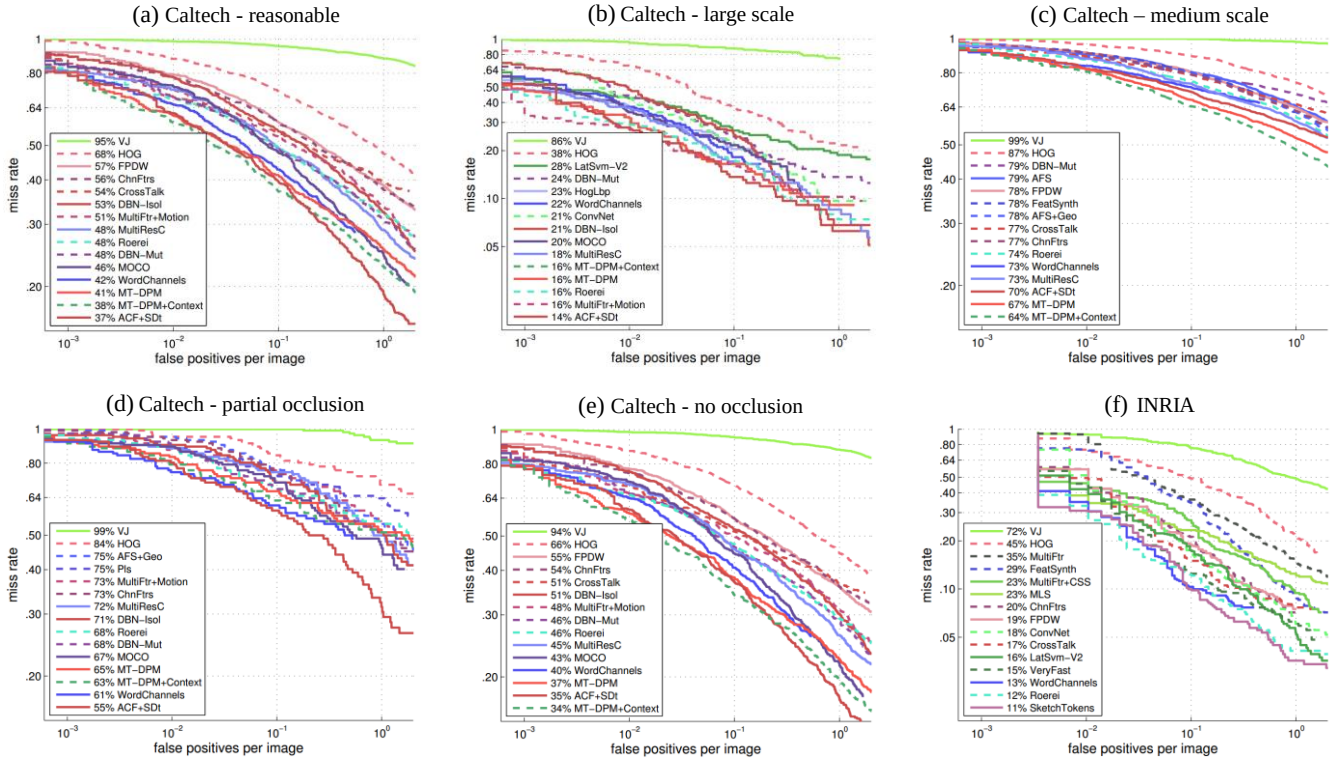


Figure 6: Benchmark results for different test scenarios.

### 4.3. Detection process

The main steps for detecting pedestrians in an input image are presented in Figure 4. The first step is to compute the local descriptors for the three descriptor types. The descriptor responses are computed densely at each pixel location. The responses are matched to the visual codebooks that were learnt from the training database. This results in three different visual word maps. These word maps are decomposed into word channels, with an individual channel for each word of each descriptor type. In order to allow efficient classification feature computation integral images are constructed for each word channel.

Using the integral word channels and the trained classifier cascade any detection window from the image can be classified. We use an aspect ratio of 0.41 for detection windows as suggested in [9]. For multiscale detection the image is scanned with detection windows of multiple sizes using steps of 4 pixels. We use a minimum height of 50 pixels and a scale factor of 1.07. This results in 38 different detection window scales and a total number of around 400 000 bounding boxes for a 640×480 pixel image. For the Caltech dataset a pedestrian of 50 pixels height corresponds to a pedestrian at a distance of approximately 40 meters from the camera. The sum of the boosting classification costs from each classifier cascade is used as confidence for the classification of each bounding box. Usually there are multiple overlapping detections around pedestrians and therefore we apply non-maximum suppression to solve this issue. We use the greedy approach proposed in [10]. The main idea is that for any two bounding boxes that overlap each other significantly only the one with higher confidence is retained. We set 0.6 as the overlap threshold which represents the ratio between the intersection and the union of the bounding box areas.

Figure 5 illustrates the iterative outputs of the cascade classifiers for a test image from the INRIA dataset.

## 5. Evaluation

In order to evaluate the performance of the proposed approach we use the Caltech and INRIA pedestrian detection benchmarks. A different classifier model was trained for each benchmark using the provided corresponding training datasets. The two widely used metrics for detection quality are miss rate and precision rate measured in false positives per image. There is a tradeoff between miss rate and precision. This tradeoff can be controlled by applying higher or lower thresholds over the detection confidences. The easiest way to represent the detection performance of an approach is by using ROC curves. Dollar *et al.* made available on the Caltech Pedestrian Detection Benchmark webpage[1] the detection results of each method evaluated in [12]. The set of

[1] http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/

Figure 7: Pedestrian detection results on images from the Caltech evaluation set.

approaches was updated with newer state of art methods that appeared since. We present a comparison between our method and the current top 12 approaches on both benchmarks that include *VeryFast* [3], *Roerei* [4], *ChnFtr* [10], *FPDW* [11], *CrossTalk* [13], *MOCO* [5], *DBN* [23], *MultiRes* [24], *ACF+SDt* [25], *MultiFtr* [31], *FeatSynth* [2], *SketchTokens* [21], *ConvNet* [27], *LatSvm* [16], *MLS* [22] and *MT-DPM* [32]. We add also the *HOG* [8] and *VJ* [29] approaches which provided the basis for most of the current approaches.

### 5.1. Detection performance

Figure 6 compares our approach with the current state of the art on the Caltech benchmark for multiple test scenarios [12], with different scale and occlusion settings. The approaches are ordered according to the log-average miss rate for the precision range of $10^{-2}$ to $10^{0}$ false positives per image. Our approach compares well with the current state of art. Using only the visual word based classification features inside the detection window, it outperformed multiple methods that used pedestrian context, part based models or motion information. The *MT-DPM* approach obtained better performances for reasonable pedestrians by using multiresolution deformable part models and similarly *MT-DPM+context* which is an extension that uses a car detector for eliminating false positives. For partially occluded pedestrians only one method performed better that was using also motion information. Figure 7 presents examples of detection results for Caltech evaluation images.

The INRIA benchmark is more challenging for our detector, because the context of pedestrians is more varied than for the Caltech benchmark with only urban traffic scenarios. Because of the heterogeneity of scenes captured in the INRIA dataset, in comparison to vehicle video sequences, it is difficult to guarantee that each pedestrian scale is sufficiently covered in the training database.

The images have various sizes and therefore we resize them to a maximum dimension of 640 for width and height while maintaining the original aspect ratio. We used the same minimum sliding window size, scale factor and window step as for the Caltech benchmark (specified in Section 4.3). Figure 6 illustrates the performance of our detector when compared to other top performing methods on the INRIA benchmark. Our approach obtained good miss rate in comparison with the state of the art methods.

### 5.2. Computational costs

The average execution times for 640x480 images using a GPU implementation on an Nvidia 780 GTX graphics card are given below:

- Pixel-wise local descriptor computation: 4 ms
- Codebook matching: 8 ms
- Integral image computation: 11 ms
- Classification of each bounding box: 39 ms
- Total detection time: 62 ms

Over 400 000 bounding boxes are classified in 39 ms, i.e. a rate of over 10 million classifications per second. A bounding box is classified using up to a few thousand decision stumps over classification features that can be computed in O(1) from integral images. Each bounding box is classified on an individual GPU thread. 128 decision stumps are evaluated at a time and this way the weak classifier models can be easily stored in the 48 KB fast shared GPU memory.

The whole detection process achieves a rate of around 16 FPS. From the current top performing approaches only the *VeryFast* and *Crosstalk* approaches achieve faster classification rates, but at lower detection accuracies. The speed of the detection process can be further optimized by:

- reducing the dimensionality of the descriptors using principal component analysis
- using soft cascades [33] for each of the boosting classifiers in the detection cascade
- reducing the search space for pedestrians.

Training the classification cascade is also time efficient. The total training time for the Caltech dataset was around 30 minutes. The time for training all of the boosting classifiers from the cascade for the Caltech dataset was around 5 minutes, using a multithreaded CPU implementation on an Intel Core i7-2770k processor. The most time consuming tasks are classification feature precomputation and reevaluation of the classifier cascade.

## 6. Conclusions

The main contribution of this paper is a new recognition approach based on single image size, single scale image features and a single classifier for all sliding window scales. This achievement was possible due to the higher discrimination power of the word channel based features and due to the scale independent behavior of the normalized features. The classifier was trained with images of pedestrians of various sizes using their original scale and with different types of hard negatives sets.

The current processing performance is of 16 frames per second and it is achieved by a GPU based implementation.

The achieved results of the proposed pedestrian classification method are on the same level with the state of the art methods. The performances in detection accuracies and processing time on different benchmarks, without using motion information, pedestrian context, multiple feature scales or multiple classifier models highlight the power of the WordChannel features and the great potential for further improvements of the proposed approach.

Amongst future improvements are the design of more efficient codebooks with reduced, but more relevant words based on a better evaluation of each channel contribution, the inclusion of the pedestrian context information and the reduction of the search space.

## Acknowledgment

This work was supported by the Romanian Ministry of National Education, MULTISENS project PNII-ID-PCE-2011-3-1086.

## References

[1] R. Baeza-Yates, and B. Ribeiro-Neto. Modern Information Retrieval. ACM Press, 1999.

[2] A. Bar-Hillel, D. Levi, E. Krupka, and C. Goldberg. Part-based feature synthesis for human detection. ECCV, 2010.

[3] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. Pedestrian detection at 100 frames per second. CVPR, 2012.

[4] R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool. Seeking the strongest rigid detector. CVPR 2013.

[5] G. Chen, Y. Ding, J. Xiao, and T. Han. Detection Evolution with Multi-order Contextual Co-occurrence. CVPR, 2013.

[6] Q. Chen, Z. Song, Y. Hua, Z. Huang, and S. Yan. Hierarchical Matching with Side Information for Image Classification. CVPR, 2012.

[7] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual Categorization with Bags of Keypoints. ECCV, 2004.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. CVPR, 2005.

[9] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. CVPR, 2009.

[10] P. Dollar, Z. W. Tu, P. Perona, and S. Belongie. Integral channel features. BMVC, 2009.

[11] P. Dollar, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. BMVC, 2010.

[12] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. PAMI, 2011.

[13] P. Dollar, R. Appel, and W. Kienzle. Crosstalk cascades for frame-rate pedestrian detection. ECCV, 2012.

[14] M. Enzweiler and D. M. Gavrila. Monocular pedestrian detection: Survey and experiments. PAMI, 2009.

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. IJCV, 2010.

[16] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester and D. Ramanan. Object detection with discriminatively trained part-based models. PAMI, 2010.

[17] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. PAMI, 2009.

[18] J. M. Gonfaus, X. Boix, J. van de Weijer, A.D. Bagdanov, J. Serrat, and J. Gonzalez. Harmony potentials for joint classification and segmentation. CVPR, 2010.

[19] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr. Associative Hierarchical Random Fields. PAMI, 2013.

[20] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. CVPR, 2006.

[21] J. Lim, C. Lawrence Zitnick, P. Dollar. Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection. CVPR, 2013

[22] W. Nam, B. Han, and J. H. Han. Improving Object Localization Using Macrofeature Layout Selection. ICCV Workshop on Visual Surveillance, 2011.

[23] W. Ouyang, X. Zeng and X. Wang. Modeling Mutual Visibility Relationship with a Deep Model in Pedestrian Detection. CVPR, 2013.

[24] D. Park, D. Ramanan, C. Fowlkes. Multiresolution models for object detection. ECCV, 2010.

[25] D. Park, C. Lawrence Zitnick, D. Ramanan, and P. Dollar. Exploring Weak Stabilization for Motion Feature Extraction. CVPR, 2013.

[26] K. E. A. Van de Sande, T. Gevers, and C.G.M. Snoek. Evaluating Color Descriptors for Object and Scene Recognition. PAMI, 2010.

[27] P. Sermanet, K. Kavukcuoglu, S. Chintala, Y. LeCun. Pedestrian Detection with Unsupervised Multi-Stage Feature Learning. CVPR, 2013.

[28] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. ECCV, 2006.

[29] P. Viola and M. Jones. Robust Real-Time Face Detetection. IJCV, 2004.

[30] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for Generic Object Detection. ICCV, 2013.

[31] C. Wojek and B. Schiele. A Performance Evaluation of Single and Multi-Feature People Detection. DAGM, 2008.

[32] J. Yan, X. Zhang, Z. Lei, S. Liao, S. Z. Li. Robust Multi-Resolution Pedestrian Detection in Trafic Scenes. CVPR, 2013.

[33] C. Zhang and P. Viola. Multiple-instance pruning for learning efficient cascade detectors. NIPS, 2007.