Towards Reliable Real-Time Person Detection

Silviu-Tudor Serban, Srinidhi Mukanahallipatna Simha, Vasanth Bathrinarayanan, Etienne Corvee, Francois Bremond

Towards Reliable Real-Time Person Detection

Silviu-Tudor SERBAN[1], Srinidhi MUKANAHALLIPATNA SIMHA[1], Vasanth BATHRINARAYANAN[1], Etienne CORVEE[1] and Francois BREMOND[1]

[1]*INRIA Sophia Antipolis - Mediterranee, 2004 route des Lucioles, Sophia Antipolis, France*

{*silviu-tudor.serban,srinidhi.mukanahallipatna_simha,vasanth.bathrinarayanan,etienne.corvee,francois.bremond*}*@inria.fr*

Keywords: Random sampling, Adaboost, Soft cascade, LBP channel features

Abstract: We propose a robust real-time person detection system, which aims to serve as solid foundation for developing solutions at an elevated level of reliability. Our belief is that clever handling of input data correlated with efficacious training algorithms are key for obtaining top performance. We introduce a comprehensive training method based on random sampling that compiles optimal classifiers with minimal bias and overfit rate. Building upon recent advances in multi-scale feature computations, our approach attains state-of-the-art accuracy while running at high frame rate.

1 INTRODUCTION

In most applications of person detection a high level of accuracy is not sufficient, as good detection speed is equally critical. It is thus essential to choose a mixture of learning algorithm, features and detection strategy that satisfies both requirements with minimal compromise.

We propose a versatile training system which allows automatic training optimization and possesses the ability to efficiently discriminate training samples, choose satisfactory subsets and cluster the training data. We capture substantial information at low computational cost by computing the Local Binary Pattern operator (T. Ojala and Maenpaa, 2002) and the Modified Census Transform (B. Froba, 2004) on several color channels of the training images. We implement a variant of the Adaboost classifier that uses *soft cascades* (C. Zhang, 2007) for lossless reduction of detection time.

1.1 Related work

We present a list of major contributions in the object detection field, with a focus on sliding window approaches. Most of them have influenced our work in some degree, providing both inspiration and motivation for improvement.

One of the first sliding window detectors was Papageorgiou et al. (Papageorgiou and Poggio, 2000). It focused on applying Support Vector Machines (Cortes and Vapnik, 1995) to a dictionary of multi-scale Haar wavelets. Viola and Jones [VJ] (Viola and Jones, 2004) improved on the idea by introducing integral images for fast feature computation and by using a cascade-like structure of Adaboost classifiers for increasing the efficiency of detection. The wide acceptance for gradient-based features began after Dalal and Triggs [HOG] (Dalal and Triggs, 2005) proposed histogram of oriented gradient (HOG) features for detection by showing substantial gains over intensity based features. Zhu et al. (Q. Zhu and Cheng, 2006) improved the original HOG implementation by using integral histograms. A vast majority of modern detectors are still HOG-based.

Shape features have also shown good promise. Gavrila and Philomin (Gavrila and Philomin, 1999)(Gavrila, 2007) used Hausdorff distance transform together with a template hierarchy to match image edges with a shape templates set. Wu and Nevatia (Wu and Nevatia, 2005) aimed to represent shape locally by using edgelet features, with boosted classifiers for full-body, head, torso and legs. A combination of features was used in order to provide complementary information. Wojek and Schiele (Wojek and Schiele, 2008) combined Haar-like features, shapelets (Sabzmeydani and Mori, 2007), shape context (G. Mori and Malik, 2005) and HOG features obtaining an detector that outperforms individual features of any kind. Wu and Nevatia (Wu and Nevatia, 2008) combined HOG, edgelet and covariance features. (T. Ojala and Maenpaa, 2002) combined a texture descriptor based on LBP with HOG.

Dollar et al. (P. Dollar and Belongie, 2009) proposed an extension of the Viola and Jones framework where Haar-like feature are computed over multiple channels of visual data including LUV color channels, grayscale, gradient magnitude and gradient magnitude quantized by orientation. In the Fastest Pedestrian Detector in the West (P. Dollar and Perona, 2010), this approach was extended to fast multi-scale detection given that features computed at a single scale can be used to approximate feature at nearby scales.

Tuzel et al. (O. Tuzel and Meer, 2008) utilized covariance matrices computed locally over various features as object descriptors. The boosting framework was modified to work on Riemannian manifolds, leading to better performance. Maji et al. (S. Maji and Malik, 2008) presented a way to approximate the histogram intersection kernel for use with SVMs, which provided speed-ups significant enough to enable a non-linear SVM to be used in sliding-window detection.

Babenko et al. (B. Babenko and Belongie, 2008) proposed an approach for simultaneously separating data into coherent groups and training separate classifiers for each; (C. Wojek and Schiele, 2009) showed that both (S. Maji and Malik, 2008) and (B. Babenko and Belongie, 2008) gave modest gains over linear SVMs and AdaBoost for pedestrian detection, especially when used in combination (S. Walk and Schiele, 2010).

Several groups worked on efficiently utilizing large feature spaces. Feature mining was proposed by (P. Dollar and Belongie, 2007) to explore huge feature spaces using strategies like steepest descent search before training a boosted classifier. The notion of pose and body parts was investigated by a number of authors. Mohan et al. (Mohan and Poggio, 2001) successfully extended (Papageorgiou and Poggio, 2000) with a two stage approach: supervised training of head, arm and leg detectors, and detection that involved combining outputs in a rough geometric model. Keypoints represent the base for early contributions in unsupervised part learning, including the constellation model (M. Weber and Perona, 2000)(R. Fergus and Zisserman, 2003) and the sparse representation approach of (Agarwal and Roth, 2002). Leibe et al. (A. Leibe and Schiele, 2005) adapted the implicit shape model for detecting pedestrians. However, as few interest points are detected at lower resolutions, unsupervised part based approaches that do not rely on keypoints have been proposed.

Multiple instance learning (MIL) was employed in order to automatically determine the position of parts without part-level supervision (P. Dollar and Z. Tu, 2008)(Z. Lin and Davis, 2009). In one of the most successful approaches for general object detection to date, Felzenszwalb et al. (P. Felzenszwalb and Ramanan, 2008)(P. F. Felzenszwalb and Ramanan, 2009) proposed a discriminative part based approach that models unknown part positions as latent variables in an SVM framework. As part models seem to be most successful at higher resolutions, Park et al. (D. Park and Fowlkes, 2010) extended this to a multi-resolution model that automatically switches to parts only at sufficiently high resolutions.

In terms of detection speed, recent notable publications reveal outstanding results. Dollar et. al (P. Dollar and Kienzle, 2012), builds upon previous contributions (P. Dollar and Belongie, 2009)(P. Dollar and Perona, 2010) and uses Crosstalk cascades to further reduce detection time. Benenson et al (Rodrigo Benenson, 2012) propose a method of a similar nature, but use GPU for accelerating feature computation.

2 CLASSIFICATION

Our method combines detection techniques that greatly reduce computational time, without compromising accuracy. We use efficient LBP and MCT features which we compute on integral images for optimal retrieval of rectangular region intensity and nominal scaling error. Adaboost is used to create cascading classifiers with significantly reduced detection time. We further refine detection speed by using the *soft cascades* approach and by transferring all-important computation from detection stage to training stage.

2.1 LBP Channel Features

Local binary pattern(LBP) is a non-parametric kernel which summarizes the local spatial structure of an image and is invariant to monotonic gray-scale transformations. At a given image location, LBP is defined as an ordered set of binary comparisons of values between the center block and its eight surrounding blocks (Figure 1). In the same degree, the Modified

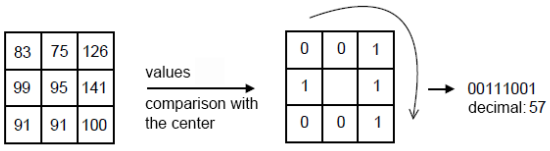

Figure 1: Extracting LBP response code (8-bit and decimal)

Census Transform (MCT) is defined as an ordered set of binary comparisons between all nine blocks and their mean value.

Inspired by the logic behind Integral Channel Features (P. Dollar and Belongie, 2009), at training stage feature extraction is done on 5 channels of the input image: Red, Green, Blue, Grayscale and Edges(Gradient Magnitude). Our approach combines two variants of the LBP feature: classic LBP with 8-bit response code and Modified Census Transform with 9-bit response code. This results in 10 different types of features, which we refer to as LBP channel features. (LBP x 5 color channels, MCT x 5 color channels). The fusion of informative channel features, along with the usage of integral image, allows comprehensive object description and fast feature computation.

2.2 Adaboost

The AdaBoost learning algorithm stands at the core of our training system. Boosting offers a convenient, fast approach to learning given a large number of candidate features. It has all the desirable attributes that a linear classifier can provide, has good generalization properties, automatically selects features based on a strategy that minimizes error and produces a sequence of gradually more complex classifiers. AdaBoost constructs a *strong* classifier by linearly combining *weak* classifiers.

Algorithm 1: The AdaBoost Algorithm

Given
$(x_1, y_1), \ldots, (x_m, y_m); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$
Initialise weights $D_1(i) = 1/m$.
For $t = 1, ..., T$
  Find $h_t = \arg\min_{h_j \in \mathcal{H}} \varepsilon_j = \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)]$
  If $\varepsilon_t \geq 1/2$ then stop
  Set $\alpha_t = \frac{1}{2}\log(\frac{1-\varepsilon_t}{\varepsilon_t})$
  $D_{t+1}(i) = \frac{D_t(i)exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
end
  Output the final classifier
  $H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

2.3 Cascading Classifiers

A Classifier Cascade represents the concatenation of several classifiers, using all information collected from the output from a given classifier as additional information for the next classifier in the cascade. In our cascading ensemble we compile classifiers by boosting LBP channel features.

When we train a classifier, an extensive array of features is extracted (training images x feat. locations x feat. sizes x feat. types). The array serves as input for the AdaBoost meta-learning method that constructs a classifier in an iterative fashion. In each iteration, a non-complex learning algorithm which selects the feature with the lowest discrimination error is called. This feature becomes a weak learner and is attributed a weight signifying its importance in the strong classifier. When the cumulus of weak learners can correctly discriminate all the training samples, iterating stops and the strong classifier is stored.

At each stage we combine the trained classifiers in a partial cascade which filters-out negative samples for training the next classifier. The cascade is considered complete when combined classifiers reach the desired level of performance. At detection time, the cascade approach works as a multistage False Positives filter for a given set of candidates.

2.4 Cascading and Random sampling

We use random sampling to construct cascades of optimized classifiers. A random sampling classification stage is a seemingly exhaustive process. Repeatedly, random positive and negative samples are chosen to be trained via our boosting method, creating a temporary classifier, then tested as shown in (Figure 2) . It iterates until a training goal is met and the classifier is stored.

Weights can be used to guide the construction of each classifier. After each test, training samples will be mapped with a performance table which keeps count of how many times they were correctly identified by all temporary classifiers. If samples with high counts are given more chances to be selected for training, the final classifier will learn to discriminate a vast majority of training samples in less time. If samples with low counts are given more chances to be selected, the final classifier will very accurately recognize all True Positives, but will suffer in terms of speed and robustness.

3 RANDOM SAMPLING TRAINING

Our system was designed to obtain best possible classification with little or no supervision while tackling existing dataset problems (Antonio Torralba, 2011)

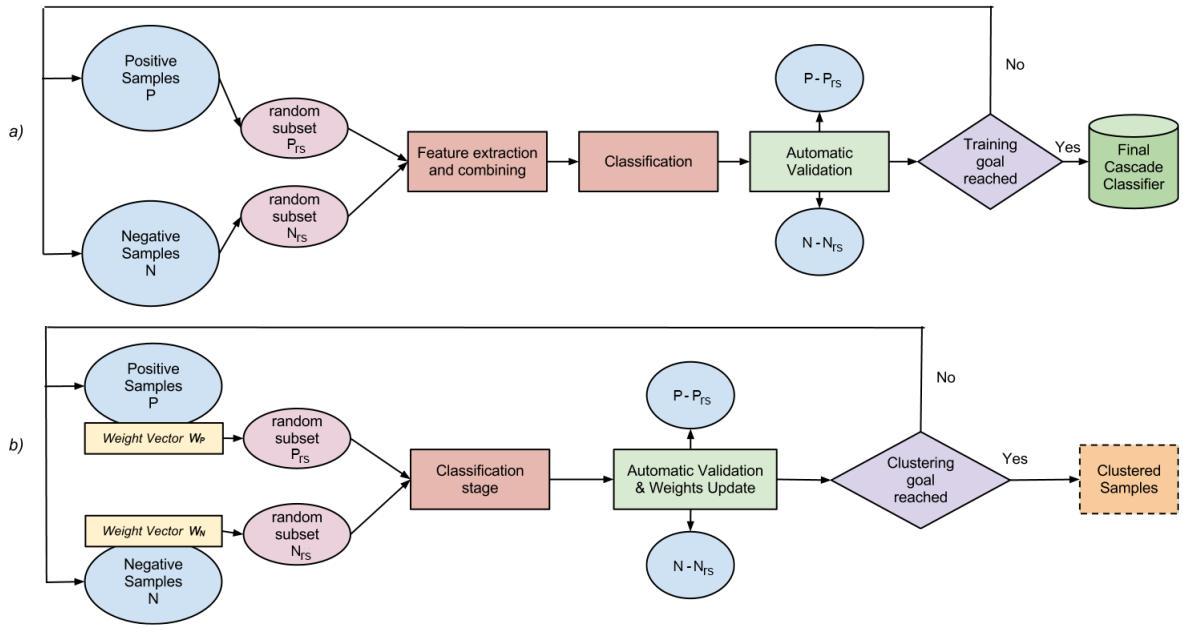

Figure 2: a)Random sampling approach for classification optimization. b)Weighted random sampling for dataset clustering.

by using the statistical method of random sampling. This minimizes bias, better estimates a general model and simplifies analysis of results.

The approach reveals an unsupervised mechanism of training sample selection and classification optimization. We describe the general work flow of the iterative system, see (Figure 2.a), and highlight it's most important features and advantages.

Automatic generation and validation. From given positive sample dataset **P** a subset $\mathbf{P_{rs}}$ is randomly chosen and from given negative sample dataset **N**, a subset $\mathbf{N_{rs}}$ is chosen. Feature extraction is performed on $\mathbf{P_{rs}}$ and $\mathbf{N_{rs}}$ and a cascade classifier candidate **C** is trained via AdaBoost. Upon generation, the performance of **C** is validated on testing sets $\mathbf{t_P = P - P_{rs}}$ and $\mathbf{t_N = N - N_{rs}}$. We define the training goal **G** as a set of rules that include a maximum number of allowed features per classifier and the threshold percentage of accuracy obtained by the classifier on $\mathbf{t_P}$ and $\mathbf{t_N}$. When **G** is satisfied, **C** is stored and the system begins training the next classifier cascade. If **G** is not satisfied, **C** is dismissed, $\mathbf{P_{rs}}$ and $\mathbf{N_{rs}}$ are regenerated and the classifier cascade training is restarted.

Dataset bias and overfitting reduction. We aim to reduce bias by concatenating several person datasets in order to obtain a diverse and ample dataset. Also, by randomly selecting a subset of training samples from the training dataset, we minimize the level of similarity between training samples and effectively decrease the overfitting effect.

Efficient big dataset handling. The random sampling technique allows training large datasets without the need of supercomputers. By choosing the subset that represents the entire set with minimal error, computation charges of classification are greatly reduced and the resulting classifier are similar or better.

Detection optimization. An outcome of using a selected subset of the total samples is that just a handful of features are needed to correctly discriminate object class. During our experimentation, we have obtained up to 10-fold speed-up in feature computation time while in terms of quality, we show State-of-the-Art detection rates on all the evaluated datasets.

Dataset clustering. Dataset segregation can be achieved by using weight vectors in conjunction with our random sampling approach. In this technique, weight vectors $\mathbf{W_P}$ and $\mathbf{W_N}$, store the number of chances any members of **P** and **N** has to be selected for training. Preliminary to training a cascade classifier, weights of all members are set to default, $\mathbf{W_P}[1..size_{(P)}] = 1$ and $\mathbf{W_N}[1..size_{(N)}] = 1$. When classification is concluded, the resulting trained classifier **C** is validated on $\mathbf{t_P}$ and $\mathbf{t_N}$ (as shown in Figure 2.b). Correctly identified samples receive an additional chance to get randomly selected ($\mathbf{W_P}[z] = \mathbf{W_P}[z] + 1$, where z is sample ID). We define as Recall score **RS**, a distribution that reveals how many times each test sample has been correctly identified over several validation stages (iterations). The clustering goal $\mathbf{G'}$ is composed by maximum number of itera-

tions **T** and a set of thresholds for automatically separating **P** and **N** into subsets with similar **RS**. When the clustering goal $\mathbf{G}'$ is reached, the weight vectors $\mathbf{W_P}$ and $\mathbf{W_N}$ will be stored in RS_P (Figure 4) and RS_N. In the case of person detection, samples with low **RS** represent the images that are difficult to discriminate using general models (hard positives - view examples in (Figure 5) and hard negatives).

Figure 3: Random sampling training is performed with subsets of 1k, 2k, 3k and 4k size. The 3k subset (10.7% of parent set) reveals adequate discrimination and requires a low number of weak learners

Our positive training set consists of 28.000 image samples which have been extracted from MIT [1], DAIMLER [2], NICTA [3] and INRIA [4] person datasets. The negatives samples are generated from a comprehensive set of background images.

We train, in parallel, classifiers with same training goal but different number of samples (see Figure 3). Training goals serve great purpose in making training automatic and enforcing quality. By adjusting the training goal to a default 95% accuracy threshold, our system compiles classifier cascades in trivial time (around 6 hours for our configuration) and highlights classifiers with minimal number of features and classifiers with highest accuracy levels.

A qualitative improvement of the classifier cascades can be obtained by raising the accuracy threshold, at the cost of increased training time.

[1] http://cbcl.mit.edu/software-datasets/PedestrianData.html

[2] www.gavrila.net/Datasets/datasets.html

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

[4] http://pascal.inrialpes.fr/data/human

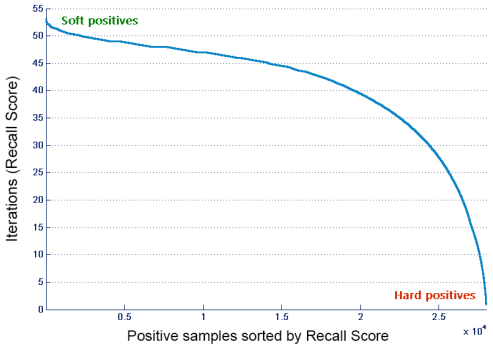

Figure 4: Ordered Recall score distribution (28k positive samples, 55 iterations of Random Sampling Clustering). The graph should be read in the following manner: The 1st sample has been correctly identified by the highest number of random sample classifiers (53) and holds a recall score of 53/55, while the 28000th sample holds a score of 1/55.

Figure 5: Top row: Hard Positives, Bottom row: Soft Positives

Output data such as the distribution in (Figure 4) is of high value, as it highlights atypical training samples. For instance, a subset of samples that registers low recall scores may offer great insight in regard to the limitations of the current classifier configuration.

3.1 Classifier Comparison

We compare the average performance of our standard Adaboost classification, Random Sampling method and Weighted Random Sampling method. (Figure 6) reveals the trade-off in terms of detection, training time and detection speed between the 3 approaches. The standard boosting approach has a minimal training time. However it falls behind the other approaches in terms of True Positives and detection speed. The random sampling method maximizes detection accuracy, improves detection speed, at the cost of increased training time. With the use of weights, difficult positive samples are given a lesser chance of selection in the classification process. This results in a

faster generation of Optimal classifiers and reduced False Positives. A negative, but negligible, side-effect is a mild decrease in True Positives.

In regards to training time, in our experimentation, generating a classifier using the Random Sampling approach took, on average, 4 times longer than training in the classical manner, and only 2 times longer when using weights. On the other hand, classifiers generated with either random sampling approaches use less weak learners thus minimizing detection time(up to 15fps on VGA resolution input).

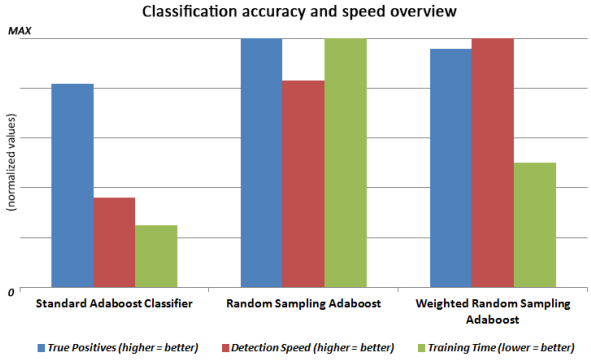

Figure 6: Classifier comparison

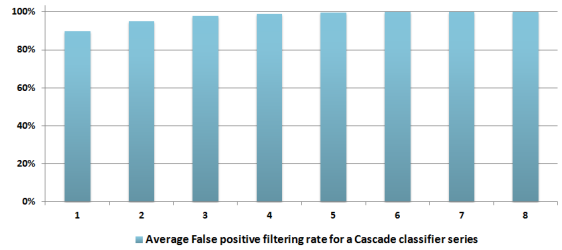

Figure 7: Cascade Significance graph

4 EXPERIMENTAL RESULTS

Our object detector is tested on 2 public datasets namely PETS [5], ETISEO [6] and 2 private datasets namely VANAHEIM [7], Hospital [8]. The sequences contain single to various objects in challenging conditions that include illumination change, low resolution, appearance change, pose change and partial occlusions. All sequences have been processed with default configuration settings: 10 different scan window sizes and search step of 2 pixels. The classifier threshold is set to 50%, classic for boosting.

Results				
Dataset	Metric	OpenCV	DPM	**Ours**
VANAHEIM	Precision	0.82	0.72	**0.86**
	Recall	0.41	**0.73**	0.62
	F-Score	0.54	**0.72**	**0.72**
Hospital	Precision	0.66	0.77	**0.89**
	Recall	0.7	0.76	**0.83**
	F-Score	0.67	0.76	**0.85**
ETISEO	Precision	**0.91**	0.83	**0.91**
	Recall	0.48	**0.77**	**0.77**
	F-Score	0.62	0.8	**0.84**
PETS	Precision	0.92	**0.95**	**0.95**
	Recall	0.42	0.71	**0.83**
	F-Score	0.57	0.81	**0.88**

Table 1: Comparison of detection results.

The detector presented a commendable behavior on all testing data and even more so on the PETS dataset. We have performed comparative tests with the OpenCV[9] HoG detector (Dalal and Triggs, 2005) and state of the art Deformable Parts Model detector (DPM) (P. F. Felzenszwalb and Ramanan, 2009). On Hospital ,ETISEO and PETS datasets our approach outperforms both OpenCV and DPM, while on the VANAHEIM dataset it is on par with DPM and outperforms OpenCV. A visual comparison is available on the final page (Figure 8).
The detection results are shown in (Table 1), where: TP - True Positives, FP - False Positives, FN - False Negatives, P - Precision, R - Recall, F - F-Score,

$$\left| \mathrm{P} = \frac{TP}{TP+FP}, \mathrm{R} = \frac{TP}{TP+FN}, \mathrm{F} = \frac{2 \times P \times R}{P+R} \right|$$

When possible, our approach takes advantage of context information. Here the context information refers to the camera calibration details which includes camera intrinsic, extrinsic information and 3D measurements of the mobile objects.

When performing detection on 640x480 resolution images, with default configuration, we attain a constant detection speed of 15FPS running on a new generation processor, single core with 3.0Ghz clock speed. In some datasets, context information allows us to limit the scan window search range. Using that we have reached up to 50FPS detection speed without any loss in terms of quality.

[5] http://pets2012.net - Dataset S2.L1-walking.
[6] http://www-sop.inria.fr/orion/ETISEO/
[7] http://www.vanaheim-project.eu/
[8] http://www.demcare.eu/
[9] http://docs.opencv.org/

5 CONCLUSION

Our comprehensive training method based on random sampling is a powerful tool for training optimization. Coupled with an efficient configuration of feature extraction, classification and detection strategy, it compiles a competent classifier, efficient in both speed and detection quality. In our evaluation we have shown that our method outperforms actual state-of-the-art approaches.

We hope to extend feature descriptors by using Local Ternary Patterns (X. Tan, 2010) and enhance detection speed by enhance detection speed by applying techniques presented in (P. Dollar and Kienzle, 2012), (Rodrigo Benenson, 2012).

Acknowledgement. The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 - Challenge 2 - Cognitive Systems, Interaction, Robotics - under grant agreement n 248907 - VANAHEIM.

REFERENCES

A. Leibe, E. S. and Schiele, B. (2005). Pedestrian detection in crowded scenes . *CVPR*.

Agarwal, S. and Roth, D. (2002). Learning a sparse representation for object detection. *ECCV*.

Antonio Torralba, A. A. E. (2011). Unbiased Look at Dataset Bias. *CVPR*.

B. Babenko, P. Dollar, Z. T. and Belongie, S. (2008). Simultaneous learning and alignment: Multi-instance and multi-pose learning. *ECCV*.

B. Froba, A. E. (2004). Face detection with the modified census transform. *In Proc. of 6th Int. Conf. on Automatic Face and Gesture Recognition*, pages 91–96.

C. Wojek, S. W. and Schiele, B. (2009). Multi-cue onboard pedestrian detection. *CVPR*.

C. Zhang, P. A. V. (2007). Multiple-Instance Pruning For Learning Efficient Cascade Detectors. *NIPS*.

Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*.

D. Park, D. R. and Fowlkes, C. (2010). Multiresolution models for objdetection. *ECCV*.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. *CVPR*.

G. Mori, S. B. and Malik, J. (2005). Efficient shape matching using shape contexts. *TPAMI*, pages 1832–1837.

Gavrila, D. M. (2007). A bayesian, exemplar-based approach to hierarchical shape matching. *TPAMI*.

Gavrila, D. M. and Philomin, V. (1999). Real-time object det. for smart vehicles. *ICCV*.

M. Weber, M. W. and Perona, P. (2000). Unsupervised learning of models for recognision. *ECCV*.

Mohan, C. P. and Poggio, T. (2001). Example-based object det. in images by components . *TPAMI*, 23, no. 4:349–361.

O. Tuzel, F. P. and Meer, P. (2008). Ped. det. via classification on riemannian manifolds . *TPAMI*, 30 no 10:1713–1727.

P. Dollar, Z. Tu, H. T. and Belongie, S. (2007). Feature mining for image classification. *CVPR*.

P. Dollar, Z. Tu, P. P. and Belongie, S. (2009). Integral channel features,. *BMVC*.

P. Dollar, R. A. and Kienzle, W. (2012). Crosstalk Cascades for Frame-Rate Pedestrian Detection. *ECCV*.

P. Dollar, S. B. and Perona, P. (2010). The fastest pedestrian detector in the west. *BMVC*.

P. Dollar, B. Babenko, S. B. P. P. and Z. Tu, M. (2008). Multiple component learning for object detection. *ECCV*.

P. F. Felzenszwalb, R. B. Girshick, D. M. and Ramanan, D. (2009). "Object detection with discriminatively trained part based models. *TPAMI*, 99.

P. Felzenszwalb, D. M. and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. *CVPR*.

Papageorgiou, C. and Poggio, T. (2000). A trainable system for object detection. *IJCV*, 38:111–136.

Q. Zhu, S. Avidan, M. Y. and Cheng, K. (2006). Fast human detection using a cascade of histograms of oriented gradients. *CVPR*.

R. Fergus, P. P. and Zisserman, A. (2003). Object classMVA recognition by unsupervised scale-invariant learning. *CVPR*.

Rodrigo Benenson, Markus Mathias, R. T. L. J. V. G. (2012). Pedestrian detection at 100 frames per second. *CVPR*.

S. Maji, A. B. and Malik, J. (2008). Classification using intersection kernel SVMs is efficient. *CVPR*.

S. Walk, K. S. and Schiele, B. (2010). Disparity statistics for pedestrian detection: Combining appearance, motion and stereo . *ECCV*.

Sabzmeydani, P. and Mori, G. (2007). Detecting pedestrians by learning shapelet features. *CVPR*.

T. Ojala, M. P. and Maenpaa, T. (2002). Multiresolution grayscale and rotation invariant texture classification with local binary patterns. *TPAMI*, 24 no 7:971–987.

Viola, P. A. and Jones, M. J. (2004). Robust real-time face detection. *IJCV*, 57 no. 2:137–154.

Wojek, C. and Schiele, B. (2008). A performance evaluation of single and multi-feature people detection. *DAGM*.

Wu, B. and Nevatia, R. (2005). Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detection. *ICCV*.

Wu, B. and Nevatia, R. (2008). Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection. *CVPR*.

X. Tan, B. T. (2010). Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions. *IEEE Transactions on Image Processing*, 19(6):1635–1650.

Z. Lin, G. H. and Davis, L. S. (2009). Multiple instance feature for robust part-based object detection. *CVPR*.

Figure 8: First column: DPM, Second Column: Ours. Detection samples of VANAHEIM, Hospital, ETISEO and PETS