Exploiting Neighbors for Faster Scanning Window Detection in Images

Pavel Zemčík, Michal Hradiš, and Adam Herout

Graph@FIT, Brno University of Technology, Bozetechova 2, Brno, CZ
{zemcik,ihradis,herout}@fit.vutbr.cz

Abstract. Detection of objects through scanning windows is widely used and accepted method. The detectors traditionally do not make use of information that is shared between neighboring image positions although this fact means that the traditional solutions are not optimal. Addressing this, we propose an efficient and computationally inexpensive approach how to exploit the shared information and thus increase speed of detection. The main idea is to predict responses of the classifier in neighbor windows close to the ones already evaluated and skip such positions where the prediction is confident enough. In order to predict the responses, the proposed algorithm builds a new classifier which reuses the set of image features already exploited. The results show that the proposed approach can reduce scanning time up to four times with only minor increase of error rate. On the presented examples it is shown that, it is possible to reach less than one feature computed on average per single image position. The paper presents the algorithm itself and also results of experiments on several data sets with different types of image features.

1 Introduction

Scanning window technique is commonly used in object detection in images. In combination with highly selective and fast classifiers, it provides state-of-the-art success rates under real-time constraints for various classes of target objects [14,6,3]. Although, in reality, much information is shared between neighboring (overlapping) image positions, they are normally classified independently. Making use of this shared information has a potential to reduce amount of computations during scanning.

In this paper, we propose an effective and at the same time simple and computationally inexpensive method which uses the dependency between neighboring image position to suppress computing the original detection classifier at nearby locations. The proposed method learns a new classifiers which predict the responses of the original detection classifier at neighboring positions. When the prediction is confident enough, computing the original classifier is suppressed.

We propose to use WaldBoost algorithm [11] to learn the suppressing classifiers in such way that they reuse computations of the original detection classifier. These reused computations can be image features in case of Viola & Jones' [14]

J. Blanc-Talon et al. (Eds.): ACIVS 2010, Part II, LNCS 6475, pp. 215–226, 2010.

like detectors or possibly also other temporal computation results. This reusing of computations is crucial and, in fact, is the only reason why faster detection can be achieved.

The task of learning the suppression classifiers is similar to emulating existing detectors by WaldBoost [12,13]. Formulating the neighborhood suppression task as detector emulation allows usage of unlabeled data for training and it does not require any modifications in learning of the detection classifier. Moreover, previously created detectors can used without any modifications.

Although the classifiers proposed for scanning window detection vary highly, they also share many similarities which result from common requirements and similar properties of the target objects. The main requirements are high selectivity (low false alarm rate) and, in case of real-time processing, very low average classification time per position.

The classifiers generally rely on efficient image features to extract relevant information from the image. In literature, Haar-like features [14], Multi-block Local Binary Patterns [15], Local Rank Patterns [5], Histograms of Oriented Gradient (HOG) [3,4] and others have been shown to perform well in detection tasks.

Another common attribute of the detection classifiers is some form of focus-of-attention structure. The exact form of the attentional structure ranges from simple ad-hoc solutions [7] through more sophisticated [14,1] to theoretically sound approaches which minimize decision time for given target error rate on training data [11,2]. These attentional structures greatly reduce average classification time by rejecting most of the non-object positions early in the decision process. In attentional structures, the classifier is generally formed from several stages. After each of the stages a decision is made if it is already known with high enough confidence that the position does not contain the target object or further information has to be still extracted.

The previous approaches, which exploit the information shared by neighboring image positions in context of scanning window object detection, focus solely on sharing image features between classifiers computed at nearby locations. Schneiderman [10] advocates feature-centric computation of features as opposed to the commonly used window-centric evaluation. He proposes to compute simple discrete-valued features on a dense grid covering the whole image. These dense features are then used as input to efficiently implemented linear classifier. However, the feature-centric approach is suitable only early in the attentional classifiers. Schneiderman uses attentional cascade [14] where only the first stage is feature-centric and the rest is window-centric. The benefit of this approach vanishes when very fast classifiers are available (some detectors may need less than 2 features per position on average as shown in Section 3).

Dalal and Triggs [3] also use feature-centric computation of features. They use dense Histograms of Oriented Gradients image representation and a linear classifier trained by Support Vector Machine. This approach provides good detection rates for pedestrian detection; however, it is too computationally expensive to be used in real-time applications.

Except for the feature-centric evaluation, other ways to exploit the shared information are possible. Image features could be selected in such way that they are reused as much as possible when scanning the image. This approach, however, requires more complex learning methods. Alternatively, response of classifier at one position could be used as starting point (or as a feature) at neighboring location. Such access to previous results should provide good initial guess as the responses of classifiers at neighboring positions are highly correlated. However, this approach would also increase complexity of the learning system and would most likely require iterative retraining of the classifier which would significantly prolong the learning. On the the hand, the proposed approach of learning suppression classifiers can be used with existing detectors and the suppression classifiers are learned much faster than the original detector.

The suppression of some positions could be especially beneficial for some types of detectors and on certain computational platforms. If features that need normalization are used (e.g. Haar-like features and other linear features), suppressing some positions removes the need of possibly expensive computation of the local normalization coefficient. Also, on some platforms, the suppression could lead to faster execution as possibly deep computational pipeline does not have to be started for some positions.

The proposed neighborhood suppression method is presented in detail in Section 2 together with an algorithm able to learn the suppression classifiers. Results achieved by this approach are shown and discussed in Section 3. Finally, the paper is summarized and conclusions are drawn in Section 4.

2 Learning Neighborhood Suppression

As discussed before, we propose to learn classifiers suppressing evaluation of detection classifiers in the neighborhood of the currently examined image window. Such approach can improve detection speed only if the suppressing classifiers require very low overhead. This can be achieved by reusing computations already performed by the detection classifier itself. Most naturally, these reused computations can be responses of image features which are part of most real-time detectors [14,10,11,1,2,4,6,15,13]. In our work, the focus is only on these real-time detectors as they are the hardest to further speed up and speed of slower detectors can be improved by already known techniques [12,13].

The amount of information carried by the reused features, which is relevant to the decision task at neighboring location, will surely vary with different types of features and objects. It will also decrease with the distance of the two areas as the mutual overlap decreases.

In the further text, it is assumed that the detector for which the neighborhood suppressing classifier needs to be learned is a *soft cascade* [1]. This does not limit the proposed approach as extending it to detectors with different attentional structures is straightforward and trivial.

The soft cascade is a *sequential decision strategy* based on a *majority vote* of simple functions $h_t : \chi \rightarrow \mathbb{R}$ which are called *weak hypotheses* in the context of boosting methods [8]:

$$H_T(\mathbf{x}) = \sum_{t=1}^{T} (h_t(\mathbf{x}))\,. \quad (1)$$

The weak hypotheses often internally operate with discrete values corresponding to partitions of the object space χ. Such weak hypotheses are called by Schapire and Singer [9] *space partitioning* weak hypotheses. Moreover, the weak hypotheses usually make their decision based only on a single image feature which is either discrete (e.g. LBP) or is quantized (e.g. Haar-like features and a threshold function). In the further text, such functions $f : \chi \rightarrow \mathbb{N}$ are reffered to in general simply as *features* and the weak hypotheses are combinations of such features and a *look-up table functions* $l : \mathbb{N} \rightarrow \mathbb{R}$

$$h_t(\mathbf{x}) = l_t(f_t(\mathbf{x})). \quad (2)$$

In the further text, $c_t^{(j)}$ specifies the real value assigned by l_t to output j of f_t.

The decision strategy S of a soft cascade is a sequence of decision functions $S = S_1, S_2, \ldots, S_T$, where $S_t : \mathbb{R} \rightarrow \sharp, -1$. The decision functions S_t are evaluated sequentially and the strategy is terminated with negative result when any of the decision functions outputs -1. If none of the decision functions rejects the classified sample, the result of the strategy is positive.

Each of the decision functions S_t bases its decision on the tentative sum of the weak hypotheses H_t, $t < T$ which is compared to a threshold θ_t:

$$S_t(\mathbf{x}) = \begin{cases} \sharp, & \text{if } H_t(\mathbf{x}) > \theta_t \\ -1, & \text{if } H_t(\mathbf{x}) \leq \theta_t \end{cases}. \quad (3)$$

In this context, the task of learning a suppression classifier can be formalized as learning a new soft cascade with a decision strategy S' and hypotheses $h'_t = l'_t(f_t(\mathbf{x}))$, where the features f_t of the original classifier are reused and only the look-up table functions l'_t are learned.

2.1 Learning Suppression with WaldBoost

Soft cascades can be learned by several different algorithms [1,2]. We chose the *WaldBoost* algorithm [11,13] by Šochman and Matas which is relatively simple to implement, it guarantees that the created classifiers are optimal on the training data, and the produced classifiers are very fast in practice. The WaldBoost algorithm described in the following text is a slightly simplified version of the original algorithm. The presented version is specific for learning of soft cascades.

Given a weak learner algorithm, training data $\{(x_1, y_1) \ldots, (x_m, y_m)\}, x \in \chi, y \in \{-1, +1\}$ and a target miss rate α, the WaldBoost algorithm solves a problem of finding such decision strategy that its miss rate α_S is lower than α and the average evaluation time $\bar{T}_S = E(\arg\min_i(S_i \neq \sharp))$ is minimal:

$$S^* = \arg\min_S \bar{T}_S, \text{ s.t. } \alpha_S < \alpha.$$

To create such optimal strategy, WaldBoost combines *AdaBoost* [9] and Wald's *sequential probability ratio test*. AdaBoost iteratively selects the most informative weak hypotheses h_t. The threshold θ_t is then selected in each iteration such that as many negative training samples are rejected as possible while asserting that the likelihood ratio estimated on training data

$$\hat{R}_t = \frac{p(H_t(\mathbf{x})|y = -1)}{p(H_t(\mathbf{x})|y = +1)} \tag{4}$$

satisfies $\hat{R}_t \geq \frac{1}{\alpha}$.

To learn the suppression classifiers we follow the classifier emulation approach from [13] which considers an existing detector a black box producing labels for new WaldBoost learning problem. However, when learning the suppression classifiers, the algorithm differs in three distinct aspects.

The first change is that when learning new weak hypothesis h'_t, only the look-up table function l'_t is learned, while the feature f_t is reused from the original detector. The selection of optimal weak hypothesis is generally the most time consuming step in WaldBoost and restricting the set of features thus makes learning the suppression classifier very fast.

The second difference is that the new data labels are obtained by evaluating the original detector on different image position than where the newly created classifier gets information from (the position containing the original features l_t). This corresponds to the fact that we want to predict response of the detector in neighborhood of the evaluated position.

The final difference is that the set of training samples is pruned twice in each iteration of the learning algorithm. As expected, samples rejected by the new suppression classifier must be removed from the training set. In addition, samples rejected by the original classifier must be removed as well. This corresponds to the behavior during scanning when only those features which are needed by the detector to make decision are computed. Consequently, the suppression classifiers can also use only these computed features to make their own decision. The whole algorithm for learning suppression classifier is summarized in Algorithm 1.

The neighborhood position is suppressed only when the suppression soft cascade ends with -1 decision. This way, the largest possible miss rate introduced by the suppression mechanism equals to α. The previous statement also holds when the detector is accompanied with multiple suppression classifiers which allows even higher sped-up still with controlled error.

Also, multiple neighboring position can be suppressed by a single classifier. Such behavior requires only slight change in Algorithm 1, where the training labels now become positive when the original detector gives positive result at any of the positions which should be suppressed.

2.2 Suppression in Real-Time Scanning Windows

The suppression with classifiers which reuse discrete-valued features is especially well suited for wide processor and memory architectures. On those architectures,

Algorithm 1. WaldBoost for learning suppression classifiers

Input: original soft cascade $H_T(x) = \sum_{t=1}^{T} h_t(x)$, its early termination thresholds $\theta'^{(t)}$ and its features f_t; desired miss rate α; training set $\{(x_1, y_1) \ldots, (x_m, y_m)\}, x \in \chi, y \in \{-1, +1\}$, where the labels y_i are obtained by evaluating the original detector H_T at an image position with particular displacement with respect to the position of corresponding x_i

Output: look-up table functions l'_t and early termination thresholds $\theta'^{(t)}$ of the new suppression classifier

Initialize sample weight distribution $D_1(i) = \frac{1}{m}$

for $t = 1, \ldots, T$

1. estimate new l'_t such that its

$$c_t^{(j)} = -\frac{1}{2} \ln \left(\frac{Pr_{i \sim D}(f_t(x_i) = j | y_i = +1)}{Pr_{i \sim D}(f_t(x_i) = j | y_i = -1)} \right)$$

2. add l'_t to the suppression classifier

$$H'_t(x) = \sum_{r=1}^{t} l'_r(f_r(x))$$

3. find optimal threshold $\theta'^{(t)}$
4. remove from the training set samples for which $H_t(x) \leq \theta^{(t)}$
5. remove from the training set samples for which $H'_t(x) \leq \theta'^{(t)}$
6. update the sample weight distribution

$$D_{t+1}(i) \propto \exp(-y_i H'_t(x_i))$$

multiple look-up tables l_t for a single feature f_t can be combined into single wide-word table such that single word contains $c_t^{(j)}$ values for all the classifiers. In such case, the required $c_t^{(j)}$ values can be loaded with single memory access, added to an accumulator register using single instruction and also efficiently compared with the rejection thresholds.

Obviously, such scheme is very well suitable for SIMD architectures, such as MMX/SSE instruction set extensions found in the contemporary PC processors. In such architectures, the wide registers can hold 4 32-bit numbers or 8 16-bit integer numbers. Consequently, the implementation of such scheme can be seen as nearly free of charge from the computational point of view.

The scheme is also applicable for programmable hardware or other hardware architectures. In such case, the scheme is beneficial in that addition of the extra prediction classifiers consumes only very little resources due to nearly unchanged structure and control subsystems.

On systems with wide enough data words but no SIMD support, the implementation can be similar to SIMD, except is must be assured that the multi-accumulator is not overflown (as piecewise addition is not possible in this case). While this assumption seems to be severe and binding, the reality is such that

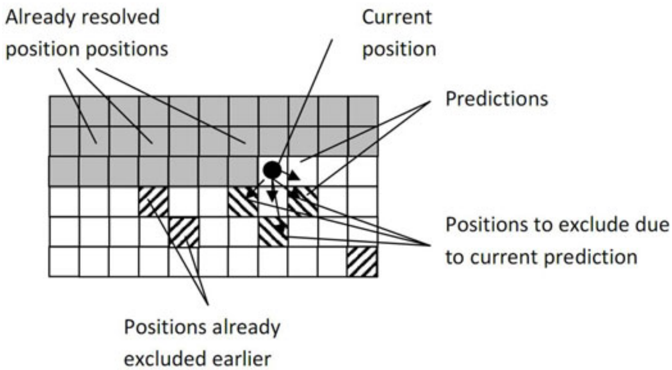

Fig. 1. Scanning an image in ordinary line-by-line fashion while using neighborhood suppression

Table 1. The benefit of neighborhood suppression for different features and datasets. ROCA is the percentage difference between area under ROC without and with area suppression. Time represents average number of features computed per position relative to the original detector without neighborhood suppression. "single" stands for suppressing single position. "12" stands for suppressing 12 positions with 12 suppression classifiers. Target error of the suppression classifiers was 5 %.

		Haar		LBP		LRD		LRP	
dataset	value	single	12	single	12	single	12	single	12
BioID	ROCA (%)	-0.02	0.07	-0.48	-3.44	-0.16	-1.08	-0.24	-2.04
	Time	0.96	0.68	0.78	0.33	0.92	0.54	0.82	0.37
PAL	ROCA (%)	-0.00	-0.39	-0.08	-0.21	-0.09	-0.85	-0.05	-0.44
	Time	0.96	0.71	0.77	0.31	0.91	0.51	0.82	0.36
CMU	ROCA (%)	-0.03	-0.36	-0.27	-1.92	-0.02	-0.49	-0.08	0.01
	Time	0.93	0.62	0.74	0.31	0.93	0.62	0.87	0.47
MS	ROCA (%)	-0.04	-0.54	-0.21	-1.02	-0.02	-0.27	-0.06	-0.65
	Time	0.93	0.60	0.73	0.29	0.93	0.60	0.87	0.45

it is easy to fulfill as the maximum possible value of each portion of the register can be calculated and predicted.

The suppression itself can be handled by binary mask covering positions to be scanned. The positions marked as suppressed are then excluded from further processing. The scanning order can remain the same as in ordinary scanning window approach, even though it restricts the positions which can be suppressed to those which are to the left and bottom of the currently classified position (see Figure 1). Possibly, more efficient scanning strategies can be developed, but such strategies are beyond the scope of this paper.

3 Experiments and Results

We tested the neighborhood suppression approach presented in the previous text on frontal face detection and eye detection. In both task, two separate test sets

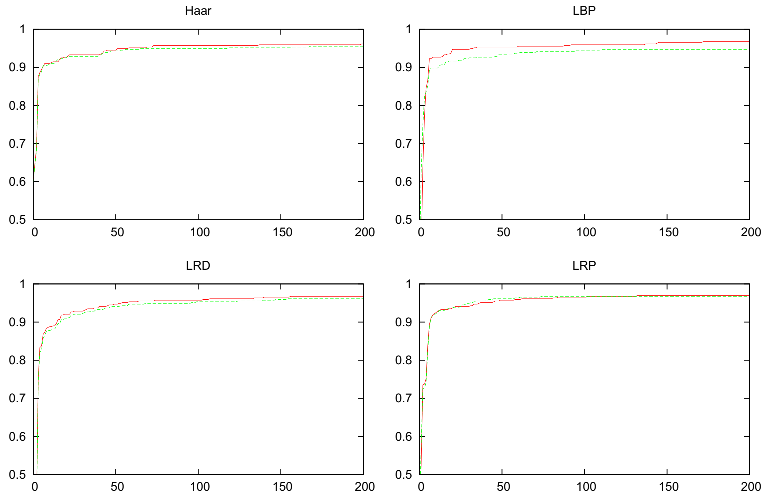

Fig. 2. The ROC curves on MIT+CMU dataset without suppression (full line) and with 12 suppression classifiers (dashed line). Target miss rate α of the suppression classifiers is 5 %.

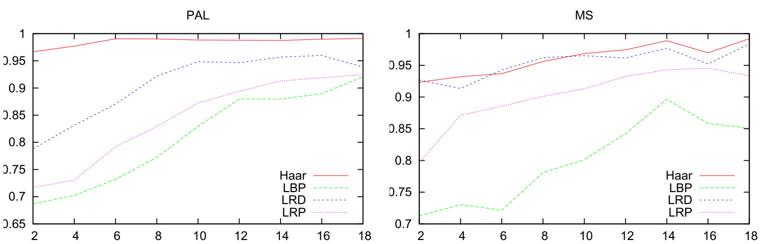

Fig. 3. Reduction of detection time (y-axis) when suppressing single positions in different horizontal distance from the classified position (x-axis). Target error of the suppression classifiers is 5 %.

were used - one with less constrained poses and lower quality images and one with easier poses and good quality images. For face detection, the harder dataset was standard MIT+CMU frontal face detection set (CMU) and the easier was a collection of 89 images of groups of people downloaded from the Internet. The easy set is denoted as MS and contains 1618 faces and 142M scanned positions. The eye detection classifiers were trained on XM2VTS[1] database and tested on BioID[2] database (104M positions, 3078 eyes) and on a easier dataset PAL[3] (111M positions, 2130 eyes) which is similar to XM2VTS. When scanning, shift of the window was two pixels at the base detector resolution and scale factor was 1.2. The suppression classifiers were trained on a large set of unannotated images containing faces.

[1] http://www.ee.surrey.ac.uk/CVSSP/xm2vtsdb/
[2] http://www.bioid.com/downloads/facedb/index.php
[3] https://pal.utdallas.edu/facedb/

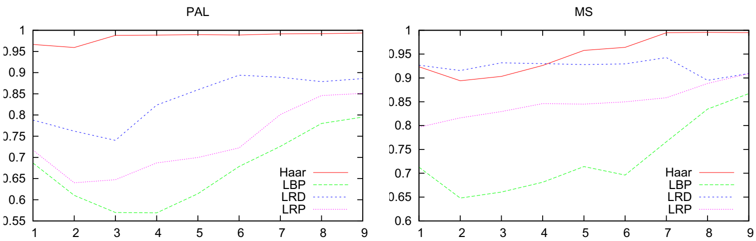

Fig. 4. Reduction of detection time (y-axis) when suppressing multiple positions on single image line by single classifier. x-axis is the number of suppressed positions. Target error of the suppression classifiers is 5 %.

The tests were performed with four types of image features which have been shown to perform well in real-time object detection. The features used were Haar-like features [14] (Haar), Multi-Block Local Binary Patterns [15] (LBP), Local Rank Differences [5] and Local Rank Patterns [5] (LRP). The real-valued responses of Haar-like features were normalized by standard deviation of local intensity and then quantized into 10 bins. The detection classifiers were learned by WaldBoost [11] algorithm and each contained 1000 weak hypotheses. The base resolution of the classifiers was 24 pixels wide.

In the first experiment, we focused on what is the the achievable speed-up using the neighborhood suppression of single and also twelve positions for moderately fast detection classifiers (4.5 - 6 features per position) and moderate target miss rate ($\alpha = 0.05$) and also on what is the influence of neighborhood suppression on precision of the detection. These results are shown in Table 1 and Figure 2. The results indicate large differences between individual image features. While the average number of weak hypotheses computed per position was reduced with twelve suppressed positions down to 30 % for LBP and 40 % for LRP , only 55 % was achieved for LRD and 65 % for Haar-like features. This can be explained by generally higher descriptive power of LBP and LRP. In general, the detection rate degraded only slightly with neighborhood suppression - by less than 1 % except for all twelve positions and LBP on datasets CMU and BioID and also LRP on BioID.

We have also evaluated the suppression ability with respect to distance form the classified position. Figure 3 shows that suppression ability decreases relatively slowly with distance and large neighborhood of radius at least 10 pixels can be used for the tested LBP and LRP classifiers.

As mentioned before, single suppression classifier can suppress larger area than just single position. Relation between speed-up and the size area of suppressed by a single classifier is shown in Figure 4. The results show that by suppressing larger area it is possible to reach higher speeds. However, the benefit is lower for frontal face detection and multiple suppression classifiers would always achieve higher speed-up.

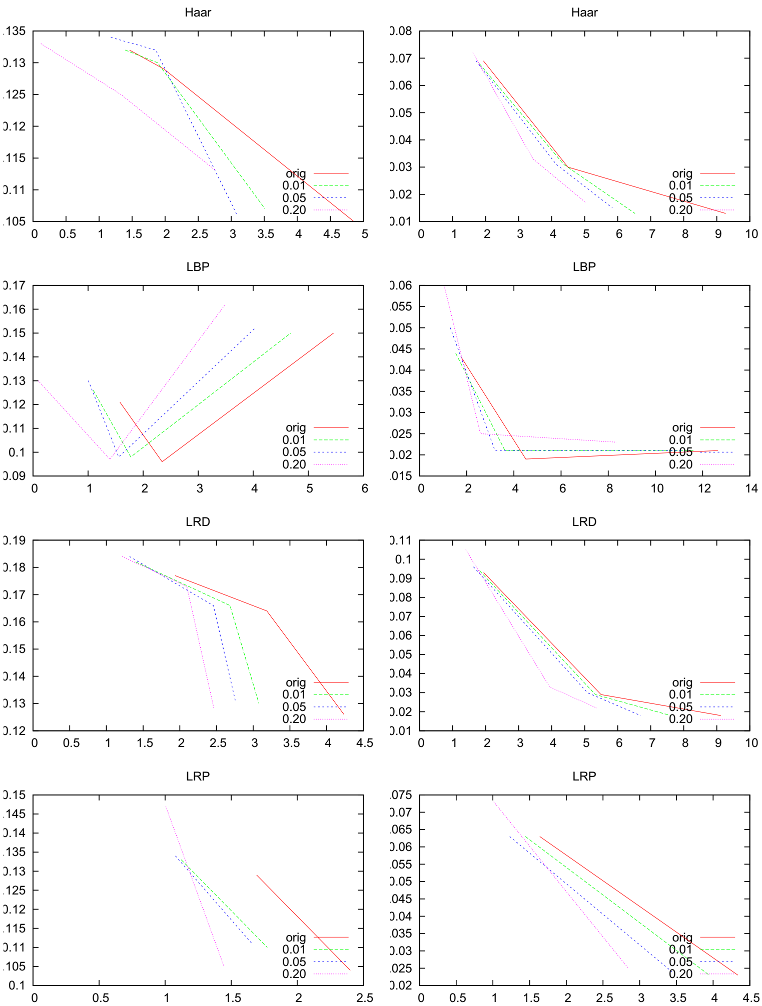

Fig. 5. Speed-up achieved by suppressing single position for different speeds of the original detector and different target miss rates α. Each line represents results for different α for three original detectors of different speed. X-axis is the speed of classifier in number of weak hypotheses evaluated on average per single scanned position (left is faster). Y-axis is area above ROC (lower is more accurate). On the left are results on eye detection PAL dataset and right are results on frontal face detection MS dataset.

For the neighborhood suppression to be useful, it must provide higher speed than simple detector for the same precision of detection. To validate this, we have trained number of detectors with different speeds (in terms of average number of features computed per position) for each feature type. Then, we learned three suppression classifiers with α set to 0.01, 0.05 and 0.2 for each of the detectors. The corresponding speeds and detection rates are shown in Figure 5. Even thought, only a single suppression classifier is used in this case for each of the detectors, the results clearly show that by using neighborhood suppression, higher speed can be reached for the same detection rate.

4 Conclusions

This paper presents a novel approach to acceleration of object detection through scanning windows by prediction of the neighbor positions results using new classifiers that reuse the image features of the detector. The approach has been demonstrated on frontal face and eye detection using WaldBoost classifiers. The results clearly show that the proposed approach is feasible and that it can significantly speed up the detection process without loss of detection performance.

Further work includes evaluation of the approach on further data sets, other features, and possibly also different classification mechanisms, such as SVM. Further work will also focus on real-time implementation of the proposed method on CPU, GPU, and programmable hardware (FPGA). Also of interest will be possible improved image scanning patterns that can benefit even more from the neighborhood suppression.

Acknowledgements

This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research program LC-06008 (Center for Computer Graphics) and by the research project "Security-Oriented Research in Informational Technology" CEZMSMT, MSM0021630528.

References

1. Bourdev, L., Brandt, J.: Robust object detection via soft cascade. In: CVPR (2005)
2. Cha, Z., Viola, P.: Multiple-instance pruning for learning efficient cascade detectors. In: NIPS (2007)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), Washington, DC, USA, vol. 1, pp. 886–893. IEEE Computer Society Press, Los Alamitos (2005)
4. Hou, C., Ai, H.Z., Lao, S.H.: Multiview pedestrian detection based on vector boosting. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part I. LNCS, vol. 4843, pp. 210–219. Springer, Heidelberg (2007)

5. Hradis, M., Herout, A., Zemk, P.: Local rank patterns - novel features for rapid object detection. In: Proceedings of International Conference on Computer Vision and Graphics 2008. LNCS, pp. 1–12 (2008)
6. Huang, C., Ai, H.Z., Li, Y., Lao, S.H.: High-performance rotation invariant multi-view face detection. PAMI 29(4), 671–686 (2007)
7. Rowley, H.A., Baluja, S., Kanade, T.: Neural network-based face detection. IEEE Transactions On Pattern Analysis and Machine intelligence 20, 23–38 (1998)
8. Schapire, R.E.: The boosting approach to machine learning: An overview. In: MSRI Workshop on Nonlinear Estimation and Classification (2002)
9. Robert, E.: Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions 37(3), 297–336 (1999)
10. Schneiderman, H.: Feature-centric evaluation for efficient cascaded object detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 29–36 (2004)
11. Sochman, J., Matas, J.: Waldboost - learning for time constrained sequential detection. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), Washington, DC, USA, vol. 2, pp. 150–156. IEEE Computer Society, Los Alamitos (2005)
12. Sochman, J., Matas, J.: Learning a fast emulator of a binary decision process. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part II. LNCS, vol. 4844, pp. 236–245. Springer, Heidelberg (2007)
13. Sochman, J., Matas, J.: Learning fast emulators of binary decision processes. International Journal of Computer Vision 83(2), 149–163 (2009)
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, p. 511 (2001)
15. Zhang, L., Chu, R., Xiang, S., Liao, S., Li, S.Z.: Face detection based on multi-block lbp representation. In: ICB, pp. 11–18 (2007)