Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Computer Graphics and Interaction

DIPLOMA THESIS ASSIGNMENT

Student: **Bc. Jan Pokorný**

Study programme: Open Informatics
Specialisation: Software Engineering

Title of Diploma Thesis: **Sharing local information in scanning-window detection**

Guidelines:

1. Familiarize yourself with methods for object detection based on boosted cascaded scanning windows search [1], [2], [3], [4].
2. Explore methods that exploit the dependence of adjacent subimages. Make use of information shared between neighboring windows.
3. Reimplement the algorithm proposed in [6] and evaluate it.
4. Propose, implement and evaluate an improved version, possibly propose and test other methods for intelligent interaction between neighboring windows.

Bibliography/Sources:

[1] Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features.- CVPR 2001
[2] Sochman, J., Matas, J.- Waldboost - learning for time constrained sequential detection.- CVPR 2005
[3] Sochman, J., Matas, J.- Learning fast emulators of binary decision processes - IJCV 2009
[4] Freund Y., Schapire R. E. – A decision-theoretic generalization of on-line learning and an application to boosting – ECCLT, 1995
[5] Adam Herout, Michal Hradis, Pavel Zemcík: EnMS: early non-maxima suppression - Speeding up pattern localization and other tasks. Pattern Anal. Appl. 15(2): 121-132 (2012)
[6] Pavel Zemcík, Michal Hradis, Adam Herout: Exploiting Neighbors for Faster Scanning Window Detection in Images. ACIVS (2) 2010: 215-226

Diploma Thesis Supervisor: prof.Ing. Jiří Matas, Ph.D.

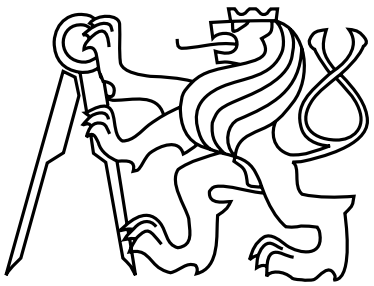
Valid until the end of the summer semester of academic year 2015/2016

prof. Ing. Jiří Zára, CSc.
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, November 4, 2014

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science and Engineering



Master's Thesis

Sharing local information in scanning-window detection

Bc. Jan Pokorný

Supervisor: Prof. Jiří Matas

Study Program: Open Informatics

Field of Study: Software Engineering and Interaction

January 5, 2015

Aknowledgements

I would like to thank Prof. Jiří Matas for supervising this thesis and Ing. Jiří Trefný for his collaboration on the experiments. Furthermore, I would like to thank my family, my girlfriend and our cat for their infinite support.

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publications used.
I have no objection to usage of this work in compliance with the act §60 Zákon č. 121/2000Sb. (copyright law), and with the rights connected with the copyright act including the changes in the act.

In Prague on January 5, 2015 .....

Abstract

Object detection is a classic task in computer vision. WaldBoost algorithm is a state-of-the-art method for object detection due its high detection accuracy and real-time speed. However, since the traditional scanning window method classifies all the windows independently and doesn't make use of the information shared among overlapping windows, there is still a possibility of a significant speed-up by exploiting this property.

We evaluate number of scanning patterns and predictors for spatially adjacent windows, inspired by work of Hradiš et. al. Furthermore, we generalize this idea from spatially adjacent widows to multiple scales and propose WaldBoost with Crosstalk Prediction. Evaluating on a state-of-the-art dataset for face detection, we show that a significant speed-up can be achieved with WaldBoost with Crosstalk Prediction with no or a little loss of precision, outperforming the reference method of Hradiš et. al.

Abstrakt

Detekce objektu je klasická úloha počítačového vidění. WaldBoost je jeden z nejlepších algorimů současnosti pro detekci objektu díky vysoké přesnosti detekce a rychlosti v reálném čase. Standardní metoda skenovacího okna klasifikuje všechna okna nezávisle na sobě, ačkoli překrývající se okna sdílí velké množství informace. Prozkoumání této vlastnosti může vést k výraznému zrychlení standardní metody.

Inspirováni prací Hradiše a spol. vyhodnotíme několik rýzných vzorců skenování a prediktorů pro okna překrývající se v prostoru. Dále generalizujeme tuto myšlenku od sousedních oken napříč škálami obrázku a navrhneme detektor WaldBoost with Crosstalk Prediction. Metodu vyhodnotíme na jednom z nejlepších současných datasetů pro detekci obličejů, ukážeme, že je možné standardní detektor výrazně zrychlit s žádnou, případně malou ztrátou kvality detekce, zároveň předčíme referenční metodu Hradiše a spol.

Contents

List of Figures

Chapter 1

Introduction

Object detection is a computer vision problem with many applications. Commonly, the applications require not only high accuracy in terms of low false negative and false positive rates but also high processing speed.

The scanning window technique combined with a rejection cascade of classifiers introduced by Viola and Jones [3] represents the state-of-the-art (e. g. work of Benenson et. al. on topic of face detection [4] or pedestrian detection [5]) and has been the dominant approach for object detection in recent years. Since its introduction, a large number of follow-up work has appeared in the literature.

In this work, we focus on the problem of increasing the speed of Viola-Jones type of methods. The WaldBoost [6] algorithm offers a competitive speed-precision trade-off using Wald's quasi-optimal sequential probability test and it achieves high detection rates for various object classes while keeping the ability to process tens of images per second. Recent advances in deep neural networks [7] have influenced state-of-the-art in object recognition significantly, however, fast object detection is still beyond its capabilities.

The standard scanning window detectors treat the decisions made about individual windows as independent despite the clear dependence of the signal in overlapping windows. This observation has been made by [8], [2] or [9].

A feature-centric approach was proposed by Schneidermann [8]. He proposed to pre-compute a set of feature values on a regular grid. The features are available for all the corresponding windows. This resulted in a significant speed-up of the algorithm. However, the reported speed for face detection was about 2 frames per second on 1.8GHz processor, which is not competitive even when the hardware speed-up since the publication of the paper is considered.

Hradiš et. al. [2] proposed a method that exploits the fact that information is shared between overlapping scanning windows. The method

introduces an auxiliary classifier for suppressing of evaluation at neighboring positions. While a window is being classified with the standard WaldBoost classifier, the response of the suppressing classifier is being computed virtually for free on the same features using only a different look-up table. If the confidence of the suppressing classifier reaches a threshold level, the neighboring position is discarded. However, if the confidence is low, the response of the suppressing classifier is ignored, even though it might contain a valuable information about the neighbor.

Dollár et. al. [9] use the correlation of pedestrian detector responses in nearby positions to build a sophisticated "crosstalk" cascade which enables neighboring detectors to communicate and achieve major computational gains. The problem we focus on, face detection, differs from pedestrian detection in the average number of evaluated weak classifiers per window – about 3 for face detection, approximately 30 for pedestrian detection – which makes the scheme impractical.

In this work, we follow this line of investigation and propose a new technique that benefit from inter-window dependences.

First, we generalize the idea of exploiting information from neighboring windows to multiple scales. The motivation behind this is to suppress all the sub-windows on lower image pyramid levels corresponding to a single window on a certain pyramid level based just on a single prediction response. This can lead to a significant speed-up of the classifier, given the number of all sub-windows grows exponentially with decreasing pyramid level.

Second, inspired by the work of Hradiš, we propose new predictors for spatially adjacent windows and evaluate their performance.

Third, the "suppression classifier" of Hradiš that makes a 0-1, suppress or don't suppress decision, is generalized and the predictor is treated as a (rather strong) weak classifier that is available at every location at no computational cost.

The rest of the thesis is structured as follows. AdaBoost algorithm, the basic element of WaldBoost classifier, is described in Chapter 2. Chapter 3 overviews the sequential analysis in object detection and WaldBoost. The idea of exploiting the neighbors for a faster detection is discussed in Chapter 4 together with an overview of [2] and introduction of WaldBoost with Crosstalk Prediction. Chapter 5 describes the implementation details and training data. The results of the work are presented and discussed in Chapter 6. Finally, the thesis is concluded in Chapter 7.

Chapter 2

AdaBoost

For object detection we use WaldBoost [6]. WaldBoost algorithm is build upon properties of two other algorithms: it uses AdaBoost [12] algorithm to select and order weak classifiers and Wald's sequential probability ratio test (SPRT) [10] to determine the decision thresholds. AdaBoost algorithm is a boosting algorithm, which means that it combines multiple weak classifiers to build a single strong classifier. The AdaBoost algorithm, as opposed to its predecessors, does not need an upper bound on the weak classifiers errors over training set weightings to be known a priori. It uses an adaptation to actual errors of weak classifiers on the training set.

AdaBoost selects and combines weak classifiers $h^{(t)} : \mathcal{X} \to \mathbb{R}$ by summing up their responses

$$f_T(\mathbf{x}) = \sum_{t=1}^{T} h^{(t)}(\mathbf{x}), \tag{2.1}$$

which for 2-class task can be denoted as

$$H_T(\mathbf{x}) = \text{sign}(f_T(\mathbf{x})). \tag{2.2}$$

There is a number of AdaBoost variants for different domains (discrete, real-valued, multi-class, ranking scores), in this work we use the real-valued version. The strong classifier response function f_T is then given by a sum of real-valued responses of weak classifiers $h^{(t)}$. See the AdaBoost training in Algorithm 1.

2.1 Training error upper bound

AdaBoost uses the following theorem to find a weak classifier in each step:

Algorithm 1 Real AdaBoost training

Input: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ where $\mathbf{x}_i \in \mathcal{X}, y_i \in \{-1, +1\}$
Output:
Initialize: sample weight distribution $D_1(i) = \frac{1}{m}$;
for t = 1, ..., T **do**

1. Train weak learner using distribution D_t
2. Get weak hypothesis $h_t : \mathcal{X} \to \{-1, +1\}$ with error
$$\epsilon_t = P_{i\ D_t}(h_t(\mathbf{x}_i) \neq y_i)$$
3. Choose $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
4. Update
$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(\mathbf{x}) = y_i \\ e^{\alpha_t} & \text{if } h_t(\mathbf{x}) \neq y_i \end{cases} = \frac{D_t \exp(-\alpha_t y_i h_t(\mathbf{x}))}{Z_t}$$
where Z_t is a normalization constant

end for
Output: the final hypothesis
$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right)$$

Theorem 1 (Schapire and Singer [12]). *Assuming the re-weighting scheme from Algorithm 1, the following bound holds on the training error of* H_T

$$\frac{1}{m}|\{i : H_T(\mathbf{x}_i) \neq y_i\}| \leq \prod_{t=1}^{T} Z_t. \tag{2.3}$$

Instead of minimizing the training error directly the greed approach is applied to upper bound minimization. In each step t, the weak classifier and its parameters are selected such that

$$Z_t = \sum_{i=1}^{m} w_t(i) \exp(-y_i h^t(\mathbf{x}_i)) \tag{2.4}$$

is minimized. Since $Z_t < 1$ when $\epsilon_t < 0.5$, the upper bound is decreased in each step.

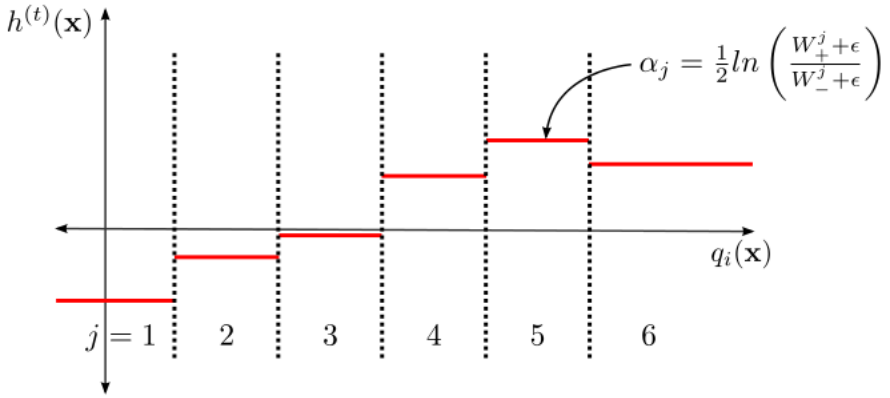


Figure 2.1: The domain-partitioning weak classifier. The response of feature $q(\mathbf{x})$ on object $\mathbf{x}$ is partitioned into bind $j = 1, \ldots, K$. The leftmost and the rightmost bins cover the respective half-spaces. In each bin j, the response of the weak classifier $h(\mathbf{x})$ is computed from the sum of positive (W_+^j) and negative (W_-^j) weights of the training samples falling into the bin. The smoothing constant ϵ is used to avoid numerical problems.

2.2 Domain-Partitioning Weak Classifiers

In this work the weak classifiers class $\mathcal{H}$ return their confidence based on a feature domain partitioning [12]. Weak classifiers $h(\mathbf{x}) \in \mathcal{H}$ are linked to one feature $q(\mathbf{x}) : \mathcal{X} \to \mathbb{R}$. The feature responses are partitioned into disjoint blocks $X_1, \ldots, X_K$ covering the whole domain and output real values for each block and returns one of K numbers α_j according to which block a sample belongs to. Uniform-width interval bins are example of such a partitioning (see Figure 2.1).

The α values are found by minimizing Z_t in equation 2.4. Let us define

$$W_b^j = \sum_{i:\mathbf{x}_i \in X_j \wedge y_i = b} w_t(i); j = 1, \ldots, K; b \in \{+1, -1\} \tag{2.5}$$

a sum of the weights of samples from class b falling into j-th bin X_j. Then we can rewrite equation 2.4 to

$$Z_t = \sum_j \sum_{i:\mathbf{x} \in X_j} w_t(i) e^{-y_i a_j} \tag{2.6}$$

$$= \sum_j (W_+^j e^{-\alpha_j} + W_-^j e^{+\alpha_j}) \tag{2.7}$$

which is minimized when

$$\alpha_j = \frac{1}{2} \log \left(\frac{W_+^j}{W_-^j} \right), \tag{2.8}$$

The stronger the response, the more different are the sums W_+^j and W_-^j. To avoid numerical problems when computing α, a smoothing coefficient ϵ is used

$$\alpha_j = \frac{1}{2} \log \left(\frac{W_+^j + \epsilon}{W_-^j + \epsilon} \right). \tag{2.9}$$

The recommended setting is $\epsilon = 1/m$.

Chapter 3

Sequential analysis in object detection & WaldBoost

3.1 Sequential analysis

Wald [10] developed the sequential decision-making theory as the statistical tool to test sequential hypothesis. He formulated a two-class sequential classification task and proved that evaluation-time-optimal solution is the *sequential probability ratio test*.

3.1.1 Sequential Probability Ratio Test

Wald proposed a Sequential Probability Ratio Test (SPRT). SPRT is a sequential strategy S^*, where

$$S_t(\mathbf{x}) = \begin{cases} +1 & \text{if } R_t(\mathbf{x}) \leq B \\ -1 & \text{if } R_t(\mathbf{x}) \geq A \\ \sharp & \text{if } B < R_t(\mathbf{x}) \leq A \end{cases} \tag{3.1}$$

where $R_t(\mathbf{x})$ is a likelihood-ratio of two hypotheses:

$$R_t(\mathbf{x}) = \frac{p(x_1, \ldots, x_t | y = -1)}{p(x_1, \ldots, x_t | y = +1)}. \tag{3.2}$$

The values A and B constrain the error rates α and β. To find A and B that provide exactly the required α and β is not suitable pro practice, therefore, Wald [11] suggested A and B to be set to their upper and lower bounds

$$A = \frac{1-\beta}{\alpha}, B = \frac{\beta}{1-\alpha}. \tag{3.3}$$

Such a setting of A and B may increase at most one of the resulting probabilities α' and β'. Wald showed the the potential increase is negligible.

3.2 WaldBoost

If SPRT is to be efficient in a classification problem where the measurements are not independent and identically distributed (non-i.i.d.), the decision functions 3.1 have to be evaluated fast. Incorporating the new measurements should be computationally simple and should not depend on the number of measurements taken so far. If the joined class-conditional densities or the likelihood ratios would have to be actually estimated, this would be unfeasible. Also, the order of measurements matters in the non-i.i.d. case (the first measurements taken should be the most informative ones).

Šochman and Matas proposed WaldBoost [6] to avoid the computation of likelihood ratios by projecting the objects to a single scalar value using a discriminatively trained classifier and reformulating the decision functions in a way that that directly thresholds the classifier output.

They use AdaBoost [12] algorithm as the classifier. AdaBoost is very efficient for the task since it chooses and orders the measurements accordingly to their discriminative strength. The classifier is a sum of the weak classifiers and thus enables the inclusion of additional measurements into the classifier's output to be constant and independent on the number of previous measurements.

3.2.1 Decision functions for classification

Let H_t be a real-valued output of a classifier incorporating features $1, \ldots, t$, the likelihood ratio R_t is reformulated as

$$R_t(\mathbf{x}) = \frac{p(H_t(\mathbf{x})|y=-1)}{p(H_t(\mathbf{x})|y=+1)}. \tag{3.4}$$

Assuming the likelihood ratio is a monotonic function of $H_t(\mathbf{x})$, the decision functions can be reformulated such that the classifier output is compared instead of the likelihood ratio:

$$S_t(\mathbf{x}) = \begin{cases} +1 & \text{if } H_t(\mathbf{x}) \geq \theta_B^{(t)} \\ -1 & \text{if } H_t(\mathbf{x}) \leq \theta_A^{(t)} \\ \sharp & \text{if } \theta_A^{(t)} < H_t(\mathbf{x}) \leq \theta_B^{(t)} \end{cases} \tag{3.5}$$

The thresholds $\theta_A^{(t)}$ and $\theta_B^{(t)}$ are estimated from training data such that the conditions are equivalent to the conditions using $R(\mathbf{x})$. Standard procedures

Algorithm 2 WaldBoost classification

Input: $h_t, \theta_A^{(t)}, \theta_B^{(t)}$ and γ for $t \in 1, \ldots, T$
Output: a classified object $\mathbf{x}$
for t = 1, ..., T **do**

1. If $H_t \geq \theta_B^{(t)}$, classify $\mathbf{x}$ as +1 and terminate
2. If $H_t \leq \theta_A^{(t)}$, classify $\mathbf{x}$ as -1 and terminate

end for
If $H_t(\mathbf{x}) > \gamma$, classify $\mathbf{x}$ as +1, -1 otherwise

(e. g. histogram, Gaussian Mixture Model, kernel density estimation) can be used to estimate the class-conditional densities $p(H_t(\mathbf{x})|y = -1)$ and $p(H_t(\mathbf{x})|y = +1)$.

3.2.2 WaldBoost classifier

The classification functions $H_t(\mathbf{x})$ are computed as sums of weak classifiers $h_t(\mathbf{x})$. It is denoted by an ordered set of T weak classifiers $h_t(\mathbf{x})$, the thresholds $\theta_A^{(t)}$ and $\theta_B^{(t)}$ and the threshold γ, that is used for the final response $H_T(\mathbf{x})$ for the object that pass through the whole sequence without being decided - this basically happens very rarely, most samples are decided in earlier stages.

The WaldBoost classification algorithm (see Algorithm 2). The decision functions are applied successively, each functions uses the response of a weak classifier $h_t(x)$ and sums it with the cumulative result of the previous $H_{t-1}(\mathbf{x})$ to evaluate $H_t(\mathbf{x})$. In each stage t, the temporary classifier response is compared to thresholds $\theta_A^{(t)}$ and $\theta_B^{(t)}$ and terminated if the corresponding conditions are met. If the decision is not made, the algorithm continues to the next decision function. If the decision is not after in T stages, the final response is thresholded by γ.

3.2.3 WaldBoost for object detection

The WaldBoost learning process is shown in Algorithm 3. The input of the algorithm are a large training set P, desired error rates α and β and number of training iterations T. It outputs the strategy, which is represented by an order set of weak classifiers $h_t(\mathbf{x}), t \in \{1, \ldots, T\}$ and the corresponding thresholds $\theta_A^{(t)}$ and $\theta_B^{(t)}$. The algorithm extends real AdaBoost by bootstrapping (sampling of the training set) and by decision thresholds.

In each iteration, a weak classifier is learned as in real AdaBoost. The

Algorithm 3 WaldBoost classification

Input:

- sample pool $\mathcal{P} = \{ (\mathbf{x}_1, y_1), \dots (\mathbf{x}_N, y_N)\}; \mathbf{x}_i \in \chi; y_i \in +1, -1$
- desired final FN rate α and FP rate β
- number of iterations T

Initialize: $A = \frac{1-\beta}{\alpha}$, $B = \frac{\beta}{1-\alpha}$, data weights $w_1(\mathbf{x}_i, y_i) = \frac{1}{N}$

Output: Weak classifiers $h_t(\mathbf{x})$ and the decision thresholds $\theta_A^{(t)}$ and $\theta_B^{(t)}$.

for t = 1, ..., T **do**

1. Sample training set $\mathcal{T} = \{ (\mathbf{x}_1, y_1), \dots (\mathbf{x}_m, y_m)\}$ from $\mathcal{P}$
2. Find $h_t(\mathbf{x})$ by real AdaBoost Algorithm on training set $\mathcal{T}$ with weights w_t and update the weights
3. Find the optimal thresholds $\theta_A^{(t)}$ and $\theta_B^{(t)}$
4. discard the samples from $\mathcal{P}$ for which $H_t(\mathbf{x}) \geq \theta_B^{(t)}$ or $H_t(\mathbf{x}) \leq \theta_A^{(t)}$

end for

training set $\mathcal{T}$ changes every iteration and the weights have to be assigned accordingly. The decision thresholds $\theta_A^{(t)}$ and $\theta_B^{(t)}$ have to be set such that they satisfy the constraints on the full training set P.

The bootstrapping is necessary due to the fact, that the training set is pruned very efficiently a in later iterations only a fraction of the original set remains. In order for the training set to remain representative, the initial number of samples in the original set would have to be unnecessarily large, if the bootstrapping was not used, which would significantly slow down the learning with no measurable impact on the quality of the final classifier.

Chapter 4

Exploiting neighbors for faster scanning window detection

The standard scanning window detectors treat the decisions made about individual windows as independent despite the clear dependence of the signal in overlapping windows. Below, we first briefly overview the method of Hradiš et. al. [2], where he exploits the information shared between spatially overlapping windows. We generalize his method in **Prediction of spatially adjacent windows** and introduce new scanning patterns and predictor types to improve the prediction performance.

Furthermore, we exploit the fact, that the information can also be propagated through the image pyramid levels. We generalize this idea in **Prediction over pyramid**.

Finally we combine both Prediction of spatially adjacent windows and Prediction over pyramid into a single detector **WaldBoost with Crosstalk Prediction**.

4.1 Neighborhood suppression by Hradiš et. al. [2]

Hradiš et. al. proposed to learn a classifier for suppression of the evaluation of the detection classifier in the neighborhood of the currently examined window. He reuses the features used by the reference detector and adds just a single look-up table, so that additional computational cost is almost zero.

The weak hypotheses are a combination of features f and a *look-up table operation* $l : \mathbb{N} \to \mathbb{R}$

$$h_t(\mathbf{x}) = l_t(f_t(\mathbf{x})). \tag{4.1}$$

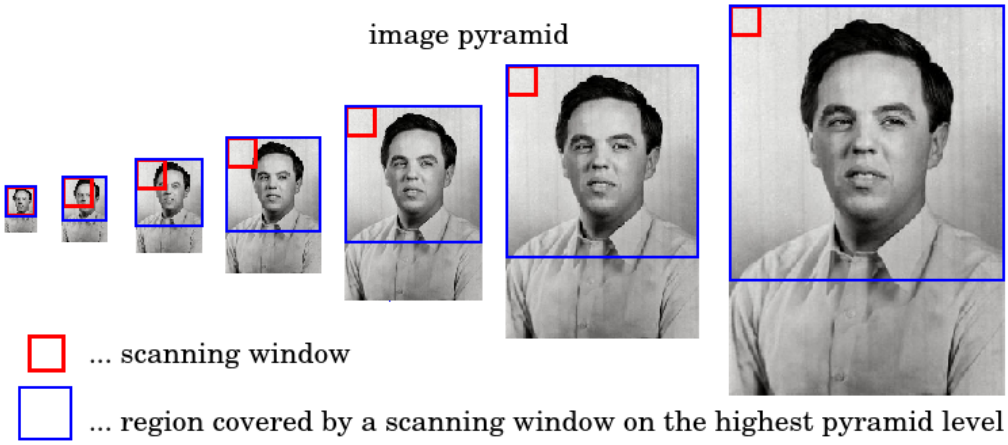


Figure 4.1: Image pyramid and a scanning window. With a decreasing pyramid level the relative detection window size decreases, therefore a single window on level n contains multiple windows on lower levels.

The task of learning the suppressing classifier can then formalized as learning a new soft cascade with a decision strategy S' and hypotheses h'_t, where the weak hypotheses reuse the features f_t from the original classifier, only new lookup-table functions l'_t are learned.

He applies the traditional scanning pattern and uses a single predictor for predicting a single position, therefore a maximum speed-up of 2x is possible. The suppression process is visualized in Figure 4.2.

4.2 WaldBoost with Crosstalk Prediction

WaldBoost with Crosstalk Prediction consists of a reference WaldBoost detector, Prediction of spatially adjacent windows and Prediction over pyramid. Similarly to [2], both predictors reuse the features computed with the original classifier. We use Local Binary Patterns (LBP) features [1] (further discussed in 5) for the classification and AdaBoost for prediction. For the LBP features used, a single additional look-up for the prediction at is about 10 times faster than the feature calculation.

4.2.1 Prediction of spatially adjacent windows

Prediction of spatially adjacent windows generalizes the method of Hradiš [2] in two ways. First, it breaks away from the top-to-bottom, left-to-right scan-

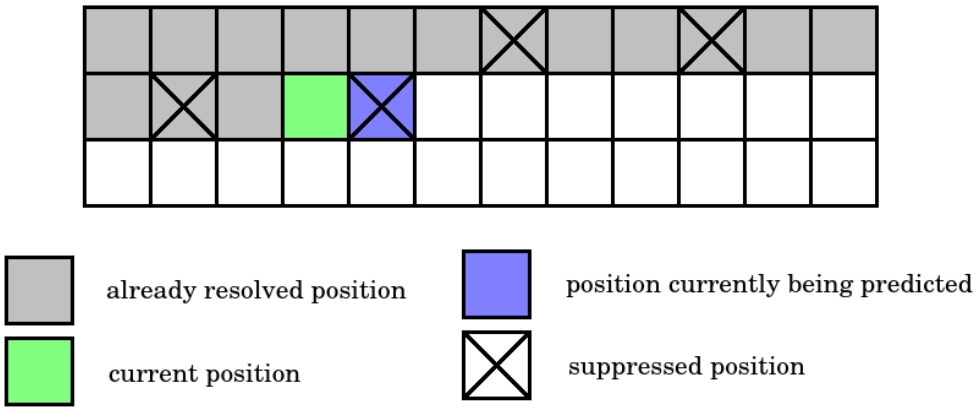


Figure 4.2: Scanning an image in ordinary line-by-line fashion while using neighborhood suppression [2]

ning pattern. Second, it does not use the information for suppression only, but contributes as a weak classifier. The prediction spatially overlapping windows is assessed like zero-length boosted detector and stops evaluation if the confidence is high enough else it is reused as the bias for the detection classifier. The steps of the prediction over neighborhood are following:

Step 1: 2d partitioning of image

Divide set $\mathcal{I}$ of all windows positions in image into 2 disjoint sets $\mathcal{C}$ and $\mathcal{N}$ such that the Minkowski sum $\mathcal{C} \oplus \mathcal{N}$ covers the original domain, i.e. $\mathcal{I} = \mathcal{C} \oplus \mathcal{N}$. $\mathcal{C}$ is set of all center positions, $\mathcal{N}$ is set of all neighboring positions. See examples of the neighborhood types in Figure 4.4. Each element $\mathbf{x} \in \mathcal{C}$ has its corresponding set of neighbors $\mathbf{x}' \in \mathcal{N}$.

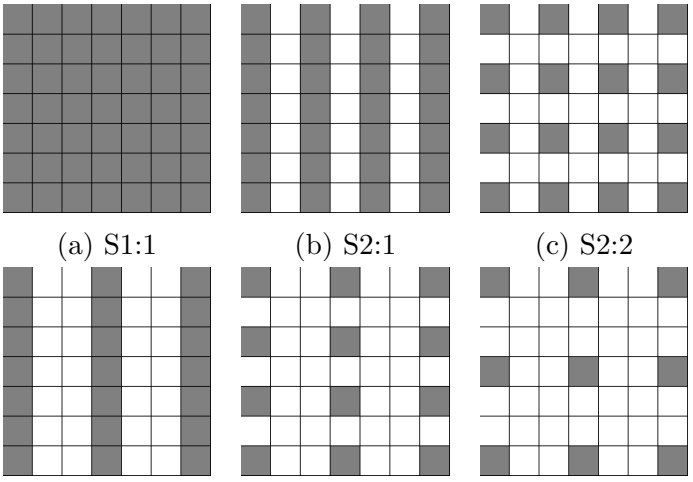
Step 2: Windows classification

1. For each $\mathbf{x} \in \mathcal{C}$ evaluate $H(\mathbf{x})$ and $H^p(\mathbf{x}')$.
2. For each $\mathbf{x}' \in \mathcal{N}$ evaluate

$$H'_t(\mathbf{x}') = H_t(\mathbf{x}') + k \min(H^p(\mathbf{x}'), 0), \tag{4.2}$$

where H is the original classifier, H^p is the predictor, $t = 0, \dots T$ and $H_0(\mathbf{x}') = 0$. Note that the suppression is handled here by enabling $t = 0$ and thus not evaluating the original classifier.

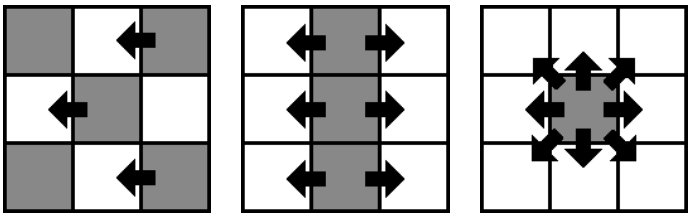
The algorithm for learning the predictor is described in Algorithm 4



(a) S1:1 (b) S2:1 (c) S2:2

(d) S3:1 (e) S3:2 (f) S3:3

Figure 4.3: Example of scanning patterns



(a) PL (b) PLR (c) P8

Figure 4.4: Types of predictors: PL (prediction left), PLR (prediction left & right), P8 (prediction for all 8 surrounding positions). Gray color corresponds to center windows $\mathcal{C}$, white to neighboring windows $\mathcal{N}$.

Algorithm 4 Training predictor H^p

Input:

- original soft cascade $H_T(\mathbf{x}) = \sum_{t=1}^{T} h_t(\mathbf{x})$, its termination thresholds $\theta^{(t)}$ and its features f_t
- training set $\{(\mathbf{x}_1, y_1) \ldots, (\mathbf{x}_m, y_m)\}, \mathbf{x} \in \chi, y \in \{-1, +1\}$, where the labels y_i are manually obtained.

Output:

- look-up table functions l_t^p of the new predictor H^p

Initialize: sample weight distribution $D_1(i) = \frac{1}{m}$
for t = 1, ..., T **do**

1. estimate new l_t^p such that its
$$c_t^{(j)} = -\frac{1}{2} \ln \left(\frac{P_{i \sim D}(f_t(\mathbf{x}_i)=j | y_j=+1)}{P_{i \sim D}(f_t(\mathbf{x}_i)=j | y_j=-1)} \right)$$
2. add l_t^p to predictor
$$H_t^p(\mathbf{x}) = \sum_{r=1}^{t} l_r^p(f_r(\mathbf{x}))$$
3. remove from the training set samples for which $H_t(\mathbf{x}) \leq \theta^{(t)}$
4. update the sample weight distribution
$$D_{t+1}(i) \propto \exp(-y_i H_t^p(\mathbf{x}_i))$$

end for

4.2.2 Prediction over pyramid

Prediction over pyramid is a novel approach. It exploits the fact that a single window on image pyramid level n contains multiple windows on levels $l' < l$ (see Figure 4.1). With common detector settings (step size = 2px, scale = 1.2, window size = 24x24px) each window in pyramid level l contains approx. 4 windows on pyramid level $l-1$, 9 windows on level $l-2$, 16 windows on level $l-3$ etc., thus offering a potential for a great speed-up if ale the sub-windows are suppressed at once.

As in Prediction of spatially adjacent windows, we reuse the features computed by the original detector and use just different look-up table, thus keeping the computational overhead close to zero.

The parameters of classification with prediction over pyramid are l^{PYR} and θ^{PYR}. If the prediction response for window $\mathbf{x}$ on pyramid level $l \leq l^{PYR}$ is lower than θ^{PYR}, then all windows $\mathbf{x}'$ that are fully overlapped by $\mathbf{x}$ are

Algorithm 5 WaldBoost with Crosstalk Prediction classification of center windows

Input: $h_t, \theta_A^{(t)}, \theta_B^{(t)}, \gamma, h_t^{SPA}, h_t^{PYR}$, for $t \in 0, \ldots, T$, where $f_t = f_t^{SPA} = f_t^{PYR}$
Output: a classified object $\mathbf{x}, H^{SPA}(\mathbf{x}), H^{PYR}(\mathbf{x})$
for t = 1, ..., T **do**

1. evaluate $H_t^{SPA}(\mathbf{x})$ and $H_t^{PYR}(\mathbf{x})$
2. If $H_t \geq \theta_B^{(t)}$, classify $\mathbf{x}$ as +1, $r = t$, terminate
3. If $H_t \leq \theta_A^{(t)}$, classify $\mathbf{x}$ as -1, $r = t$, terminate

end for
If $H_t(\mathbf{x}) > \gamma$, classify $\mathbf{x}$ as +1, -1 otherwise

classified as -1. Note that windows $\mathbf{x}'$ can only be on pyramid levels $l' < l$. The higher the value of l^{PYR} is set, the higher speed-up is expected, however, with an increased risk of missing a target objects. The same holds for value θ^{PYR}.

We learn a predictor for suppressing such windows which don't include a positive sample in any corresponding sub-window on lower pyramid levels. The same learning algorithm 4 as for Prediction of spatially adjacent windows is used.

4.2.3 WaldBoost with Crosstalk Prediction

WaldBoost with Crosstalk Prediction combines the reference detector H_t, prediction of spatially adjacent windows H^{SPA} and prediction over pyramid H^{PYR} in a straightforward manner. The additional computational cost is almost zero since the weak hypotheses of predictors are ordered identically as the weak hypotheses of the reference detector, therefore only two additional look-up tables are needed.

The classification of a center windows and spatially neighboring windows is summarized in Algorithm 5 and 6 respectively. Algorithm for object detection in image with WaldBoost with Crosstalk Prediction is summarized in Algorithm 7. It accepts an image on the input and outputs windows corresponding to detected objects. If the neighborhood is selected such a single neighbor position is spatially predicted from multiple center windows, the mean value of these predictions is used.

Algorithm 6 WaldBoost with Crosstalk Prediction classification of spatially neighboring windows

Input: $k, h_t, \theta_A^{(t)}, \theta_B^{(t)}, \gamma, h_t^{SPA}, h_t^{PYR},$ for $t \in 0, \ldots, T$, where $f_t = f_t^{SPA} = f_t^{PYR}$

Output: a classified object $\mathbf{x}, H'^{SPA}(\mathbf{x}), H^{PYR}(\mathbf{x})$

for t = 0, ..., T **do**

1. evaluate $H_t^{SPA}(\mathbf{x})$ and $H_t^{PYR}(\mathbf{x})$
2. evaluate $H'_t(\mathbf{x}) = H_t(\mathbf{x}) + k \min(H^{SPA}(\mathbf{x}), 0)$
3. If $H'_t \geq \theta_B^{(t)}$, classify $\mathbf{x}$ as +1, $r = t$, terminate
4. If $H'_t \leq \theta_A^{(t)}$, classify $\mathbf{x}$ as -1, $r = t$, terminate

end for

If $H'_t(\mathbf{x}) > \gamma$, classify $\mathbf{x}$ as +1, -1 otherwise

Algorithm 7 WaldBoost with Crosstalk Prediction for object detection in image

Input:

- image I
- neighborhood type P and scanning pattern S
- constants $k, \theta^{SPA}, \theta^{PYR}, l^{PYR}$
- functions H_t, H'_t, H_t^{SPA} and H_t^{PYR}
- function $p(\mathbf{x})$ that returns a set of all windows $\mathbf{x}''$ that are fully overlapped by $\mathbf{x}$

Output: set $\mathcal{D}$ of windows classified as $+1$

Initialize:

- initialize set $\mathcal{X}$ of all windows $\mathbf{x}$ to be classified in image I by building an image pyramid
- divide all $\mathbf{x} \in \mathcal{X}$ into disjoint sets $\mathcal{L}_1, \dots \mathcal{L}_L$, where $\mathcal{L}_l$ consists of all windows on image pyramid level l
- let $\mathbf{x}'$ be the spatial neighbors of $\mathbf{x}$ according to neighborhood type P
- $\mathcal{D} = \emptyset$

for $l = L, \dots, 1$ **do** ▷ for each image pyramid level
  $\mathcal{C} = \emptyset, \mathcal{N} = \emptyset$
  add all center windows $\mathbf{x} \in \mathcal{L}_l$ into $\mathcal{C}$ accordingly to S
  add all neighboring windows $\mathbf{x} \in \mathcal{L}_l$ into $\mathcal{N}$ accordingly to P

  for each $\mathbf{x} \in \mathcal{C}$ **do** ▷ for each center window
    evaluate $H(\mathbf{x}), H^{SPA}(\mathbf{x}')$ and $H^{PYR}(\mathbf{x})$
    if $H(\mathbf{x}) = +1$ **then** add $\mathbf{x}$ to $\mathcal{D}$
    end if
    if $l \leq l^{PYR}$ and $H^{PYR}(\mathbf{x}) \leq \theta^{PYR}$ **then**
      remove all $\mathbf{x}'' \in p(\mathbf{x})$ from their corresponding set $\mathcal{L}$
    end if
  end for

  for each $\mathbf{x} \in \mathcal{N}$ **do** ▷ for each neighbor window
    evaluate $H'(\mathbf{x})$ and $H^{PYR}(\mathbf{x})$
    if $H'(\mathbf{x}) = +1$ **then** add $\mathbf{x}$ to $\mathcal{D}$
    end if
    if $l \leq l^{PYR}$ and $H^{PYR}(\mathbf{x}) \leq \theta^{PYR}$ **then**
      remove all $\mathbf{x}'' \in p(\mathbf{x})$ from their corresponding set $\mathcal{L}$
    end if
  end for

end for
return $\mathcal{D}$ ▷ Output set of windows classified as $+1$

Chapter 5

Implementation details

5.1 Training data

The predictors were trained on the set consisting of 20000 positive and 70000 negative samples (patches with resolution 24x24 pixels).

The positive training samples for Prediction of spatially adjacent windows were generated such that each position containing a face in ground truth was shifted by 2 pixels to all 8 directions (right, left, top, bottom, right-top, right-bottom, left-top, left-bottom). See Figure 5.1 for examples of positive training data.

The positive training samples for Prediction over pyramid were generated such that each window fully overlapping an annotated face, no matter on what image pyramid level, was considered as positive. See Figure 5.2.

The negative samples were generated by random sampling of human annotated images that did not include any face.

5.2 Features

The purpose of features is to extract useful information from the given data with a low computational cost. Features can express the prior knowledge of the object class and can make learning much easier and faster than using directly the raw image data. In general, different features are suitable in different tasks, however, Local Binary Patterns (LBP) are proven to be efficient for number of different classes. In this work we used Extended Set of Local Binary Patterns [1] by Trefný and Matas (see Figure 5.3).

Figure 5.1: Positive training samples for Prediction of spatially adjacent windows

Figure 5.2: Positive training samples for Prediction over pyramid

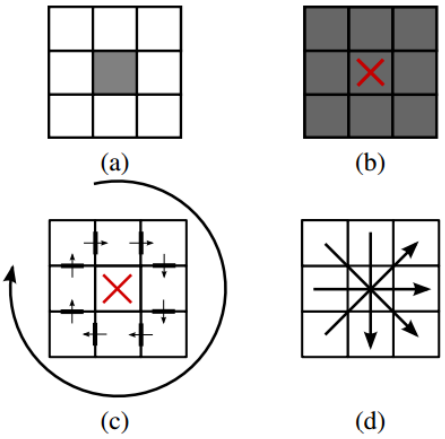


Figure 5.3: Extended set of LBPs [1]: (a) conventional LBP thresholded by center pixel value; (b) 8-bit coded modified LBP (mLBP) thresholded by pixels mean value; (c) transition coded LBP (tLBP); (d) direction coded LBP

5.2.1 What is measured

The local binary pattern operator [13] is a non-parametric descriptor on gray-scale space invariant to monotonic transformations of the intensity function. The basic LBP pattern measures a 3x3 pixel square.

The output of LBP is a binary code, which is computed by thresholding the eight neighborhood pixel values by the value of the center pixel, see Figure 5.4 (a). The operator was further extended to rotation symmetric and multi scale version [14], see Figure 5.4 (b). This LBP pattern is parametrized by the neighborhood size P and the radius R and is denoted as:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{5.1}$$

where

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}, \tag{5.2}$$

g_p correspond to gray values regularly spaced on circle and g_c is the gray center value. Gray values at non integer positions are obtained by interpolation.

LBPs are commonly used in classification of distributions (histograms) of semi-local neighborhoods. In the approaches exploiting the spatial appearance, single LBP measurements tend to be unstable and sensitive to localization and noise. This was addressed by citezhang, who introduced a

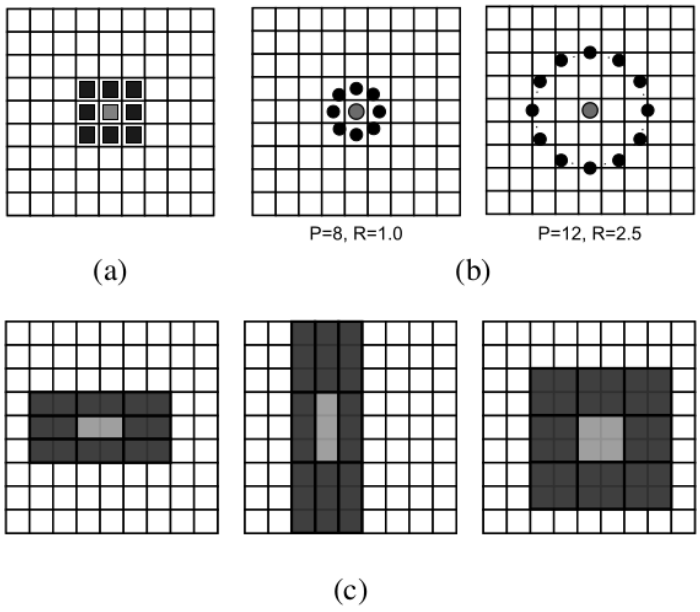


Figure 5.4: LBP comparison values: (a) original LBP, (b) rotation symmetric and multiscale $LBP_{P,R}$, (c) Examples of multi-block local binary pattern (MB-LBP)

Multi-Block LBP (MB-LBP) [15]. He compares mean values of 3x3 adjacent rectangular blocks instead of comparing the pixel values, which can be done in constant time using the integral image [3]. MP-LBPs enable generating large sets of operators with different aspect ratios and scales (see Figure 5.4), however, it's not invariant to monotonic intensity transformations, as the original LBPs, it only preserves the invariance to affine intensity changes.

5.2.2 Encoding methods

Motivated by spatial appearance classification models, Trefný and Matas [1] proposed novel encoding methods: *Transition Local Binary Patterns* and *Direction coded Local Binary Patterns.*

Transition Local Binary Patterns (tLBP) - The LBP thresholds tje neighboring gray pixel values by the center value. This provides a rough information of the relation of neighbors to the center pixels, however, the relations between pixels with the same binary value are lost. Binary code of tLBP is composed of neighbor pixel comparisons in clockwise direction, which enables to encode the information between neighboring pixels.

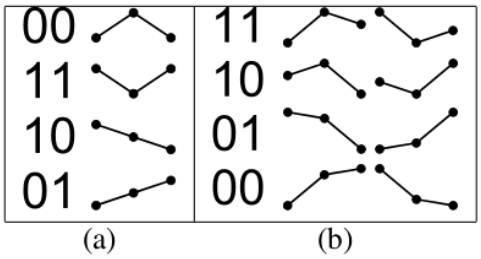


Figure 5.5: Examples of generated codes and schemes of possible pixel intensity values for a given pixel sequence: (a) LBP encoding rule, (d) dLBP encoding rule

Formally, it can be denoted as

$$tLBP_{P,R} = s(g_0 - g_{P-1}) + \sum_{p=1}^{P-1} s(g_p - g_{p-1})2^p, \tag{5.3}$$

where g_p is a gray value p-th neighbor of center pixel. tLBP is gray-scale invariant.

Direction coded Local Binary Patterns. Motivation behind dLBP is to provide a better information in sense of a direction function. Using a standard LBP operator, there are 4 basic directions through the center pixel. Motivated by spatial appearance classification models, Trefný and Matas encode intensity variation along these directions into 2 bits so that the resulting binary word is of the same length as the original LBP. The first bit encodes, whether the difference of edge pixels grows or falls due to the center one. See figure 5.5 for comparison of LBP and dLBP rules for a given direction. Both LBP and dLBP rules encode the binary information of the center pixel being an extrema. Unlike the LBP rule, dLBP does not encode it as maximum or minimum, but instead it encodes, whether the sign of first and second differential is the same. Using this property the dLBP is not inly gray-scale intensity invariant, but also has the intensity inversion invariance property.

Formally written:

$$dLBP_{P,R} = \sum_{p'=0}^{P'-1} \left(s(g_{p'} - g_c)(g_{p'+P'} - g_c)2^{2p'} + s(|g_{p'} - g_c| - |g_{p'+P'} - g_c|)2^{2p'+1} \right) \tag{5.4}$$

Chapter 6

Experiments

We evaluated the performance of our method on FDDB dataset [16] while using the reference detector as a baseline. In fact, we used 2 reference detectors: FRONTAL (frontal-view) and MULTI (multi-view) WaldBoost face detector consisting of 1000 weak classifiers.

FRONTAL face detector is a classic WaldBoost detector using one lookup-table per a feature. It is trained for recognizing faces from a frontal view. The average number of weak classifiers per a single window on FDDB dataset was 2.19.

MULTI detector is a WaldBoost detector that uses 5 look-up tables for each feature. Each table is trained on different angle of rotation in order to detect such faces that are not in vertical position. MULTI detector is basically an improved version of FRONTAL detector, it results in about 3% better detection performance on FDDB dataset. The average number of weak classifiers evaluated per a window is 6.22.

The parameter settings of the detectors were following: step size = 2px, scanning window size = 24px, scaling constant = 1.2.

The following two metrics were evaluated: (a) detection performance / speed, (b) geometric accuracy / speed. Speed is denoted as relative average number of evaluated weak classifiers per image window comparing to the reference detector.

To evaluate (a), the average value of TP rate in range of FP $\in [10, 1000]$ (in logarithmic space) is taken (see 6.1). This interval represents the "most informative" part of ROC curve. With lower values of FP, only easily detectable target objects are considered as TP and these don't vary significantly among state-of-the-art detectors. With higher values of FP, the number of misclassified background objects is too high for detector to be usable in real applications. Therefore, the classifier with the best performance within the specified interval can be considered as the best classifier overall. Since the

ROC in this interval in logarithmic space has a shape close to straight line, taking the average of it is an efficient way to enumerate the ROC curve by a single number.

To evaluate (b), the average successful detection bounding box overlap to the ground truth is used. Each detected bounding box with overlap ≥ 0.5 to a ground truth box is considered a successful detection bounding box (overlap of two boxes is computed conventionally as an intersection over union).

6.1 Prediction of spatially adjacent windows

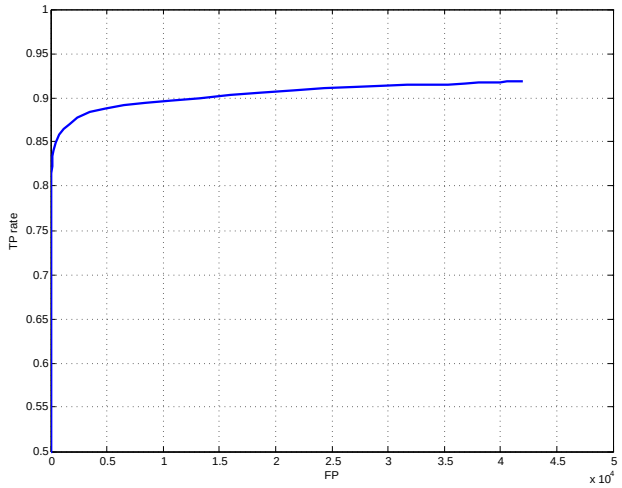
MULTI. The evaluation of number of neighborhood types combined with different scanning patterns is shown in Figure 6.2. The best overall performance was reached by predictor for neighborhood P8 with scanning pattern 2:2. This predictor was capable of gaining 2x speed-up with no impact on detection rate and 3x speed-up with losing less than 0.3% of detection rate.

We also included the predictor HRADIS in the evaluation. HRADIS corresponds to suppression classifier from [2] with the only difference, which is using only a single threshold for final predictor response instead of using a WaldBoost like set of thresholds. We argue this does not have a significant impact on the performance. See that HRADIS does quite well in the geometric accuracy, but is one of the weak ones in the detection performance.

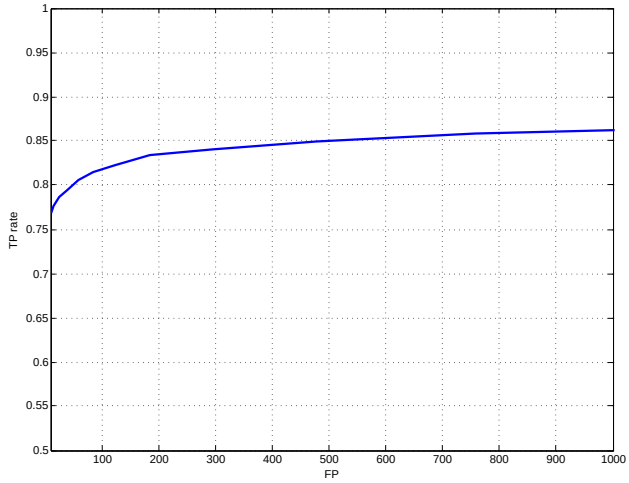
In general, neighborhood type P8 proved to be the best one of tested neighborhoods, on the other hand, neighborhood type PL is the weakest one. This supports the idea, that predicting multiple positions at once is a good approach.

Evaluation of using the predictor response as a starting point for selected predictors (setting $k = 1$) is shown in Figure 6.3. As one can see, using the prediction as a starting point of the original detector for the original classifier doesn't have as significant impact on the performance as the suppression itself, in fact, in most cases cases it resulted in a slightly worse detection performance.

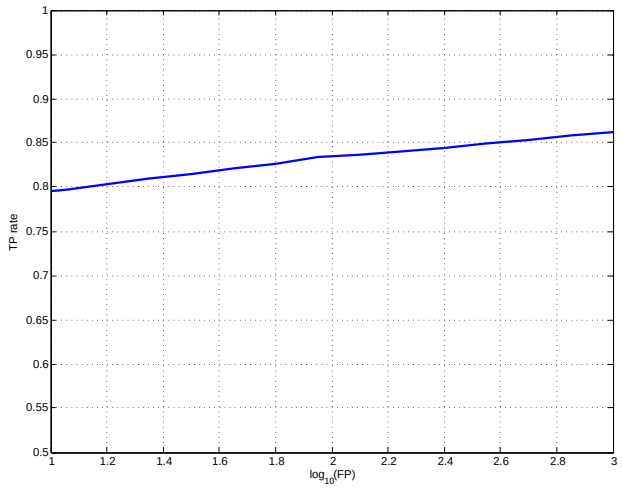
FRONTAL. The evaluation of number of different neighborhood with and without using the predictor response as the starting point of the reference detector classifier is shown in 6.4. Similarly to MULTI, the neighborhood type P8 with scanning pattern s2:2 resulted in the best performance in recognition/speed metric outperforming PLR s3:1 and HRADIS. Difference between setting k=0 or 1 turned out to be similarly negligible as with MULTI, while providing a little worse performance overall.



(a) Original ROC curve



(b) ROC curve within FP interval $[10, 1000]$



(c) ROC curve within FP interval $[10, 1000]$ in logarithmic space

Figure 6.1: ROC curve in logarithmic space

In the figures, each point represents a single ROC curve. ROC curves were obtained by shifting the zeroth threshold of $\theta_A^{(0)}$.

6.2 Prediction over pyramid

MULTI. The evaluation of prediction over pyramid is shown in Figure 6.5. The ROC curves here were generated by shifting the value of θ^{PYR}, whereas the bottom most point of each line corresponds to $\theta^{PYR} = \infty$, which is equivalent to leaving out all the pyramid levels below given l^{PYR}.

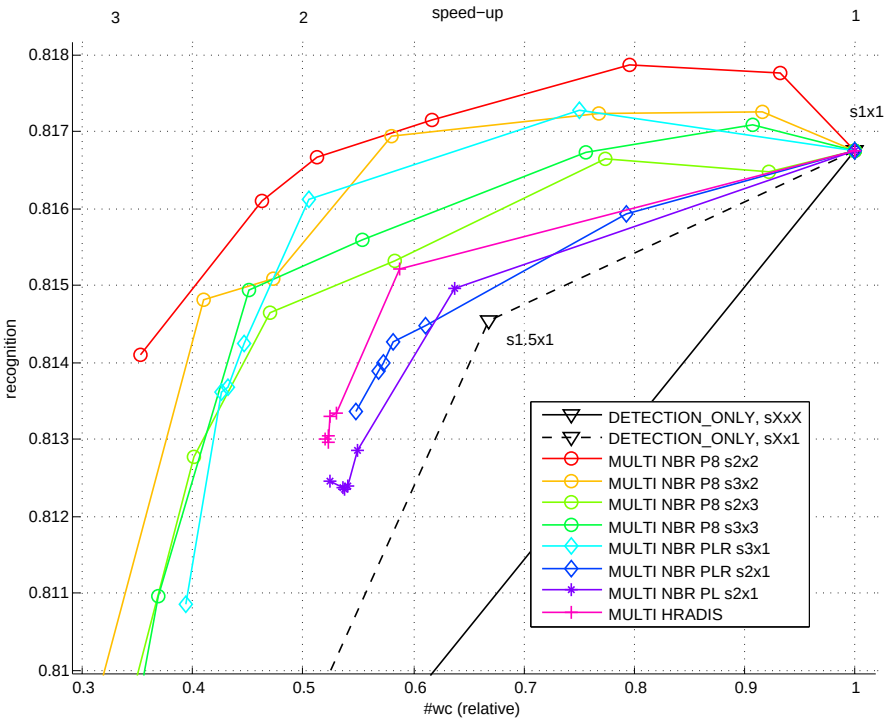
Note that a curve representing $l^{PYR} = 4$ gets to about 0.2% better detection rate than the reference detector while speeding-up almost 1.5x. See when setting l^{PYR}=Inf, the performance gets significantly worse. The most interesting results from these are probably the curves corresponding to $l^{PYR} = 4$ and 6. This means, that the predictor can still well predict the responses of windows that are scaled down by factor of 1.2^4 and 1.2^6.

See examples of successful face detection with prediction over pyramid in Figure 6.7. To generate these, l^{PYR} was set to ∞ and θ^{PYR} was set as high as possible with keeping the results identical to the output of the detector without prediction. Note that the best result is equivalent to 33x speedup of the original classifier. The images with lower speed-up (higher number of evaluated weak classifiers) generally correspond to images containing faces on relatively low levels (small faces in large images). This is understandable, since the predictions made on high levels tend to be very imprecise for windows on low levels and since these images contain small faces, the threshold θ^{PYR} must be set very low not to result in missing these faces.

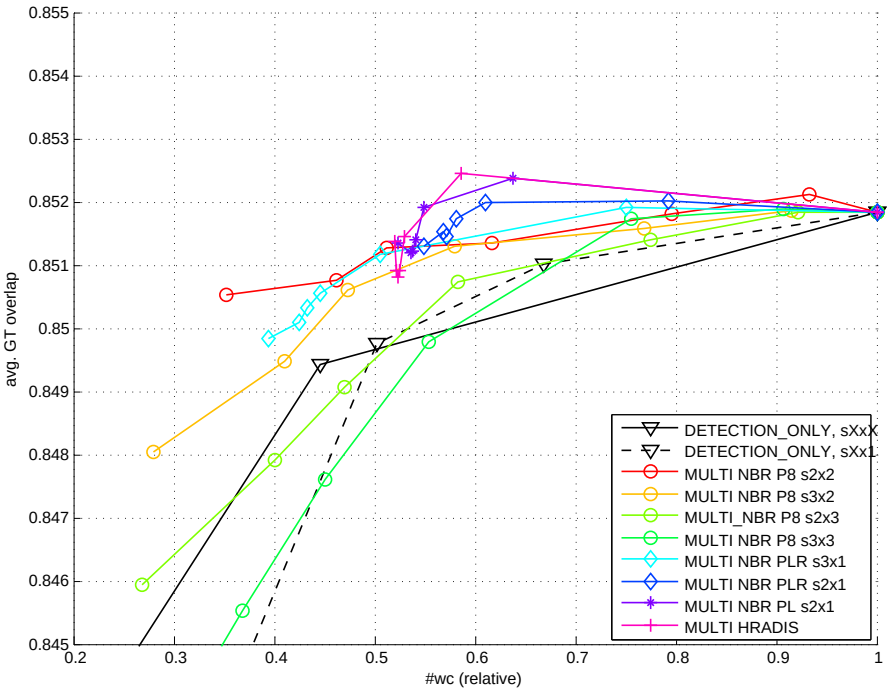
Examples of failure are shown in Figure 6.8. Here, boxes containing faces are marked as negatives with the predictor. Surprisingly, the boxes are not as large compared to faces as one would expect. Also, two of the missed men are wearing a hat, which could be the reason for the failure. These failures are matter of future investigation, possible solution cloud be extending the training dataset.

FRONTAL. See results for evaluation of prediction over pyramid in Figure 6.6. The prediction over pyramid turned out to be very sensitive for a θ^{PYR} setting. However, from these results it seems the prediction over pyramid results in rather worse performance than the original detector, since none of the points resulting from the experiments ended up being significantly over the reference curve.

In the figures, each point represents a single ROC curve. ROC curves

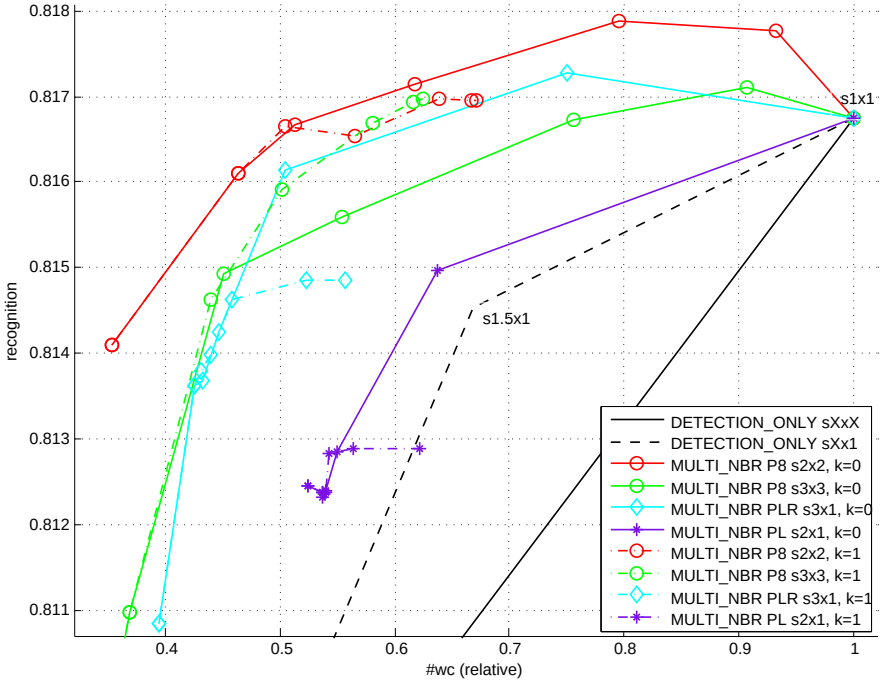


(a) recognition / relative number of evaluated weak classifiers per window

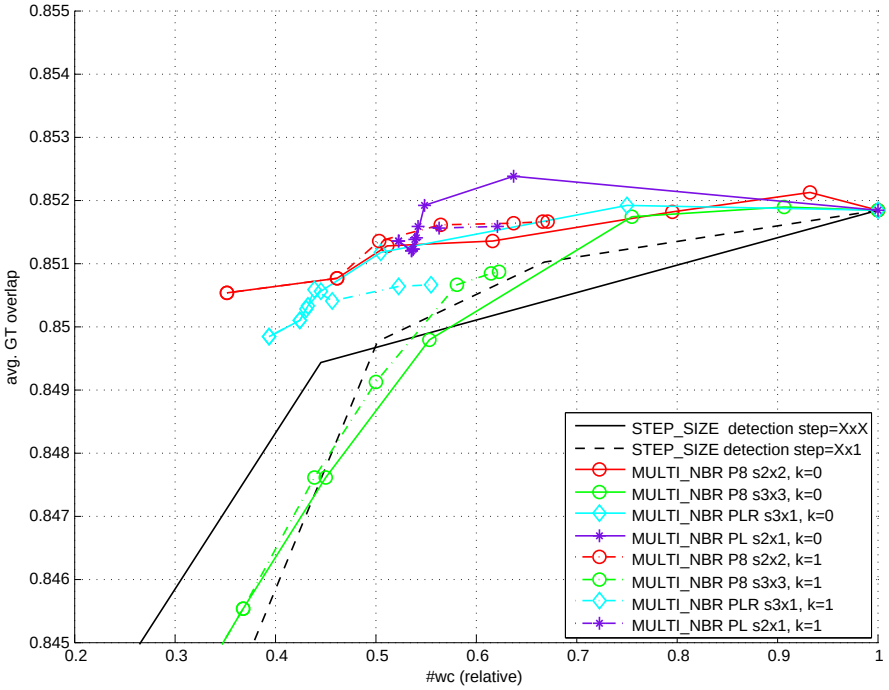


(b) geometric accuracy / relative number of evaluated weak classifiers per window

Figure 6.2: Prediction over neighborhood: multi-view detector.

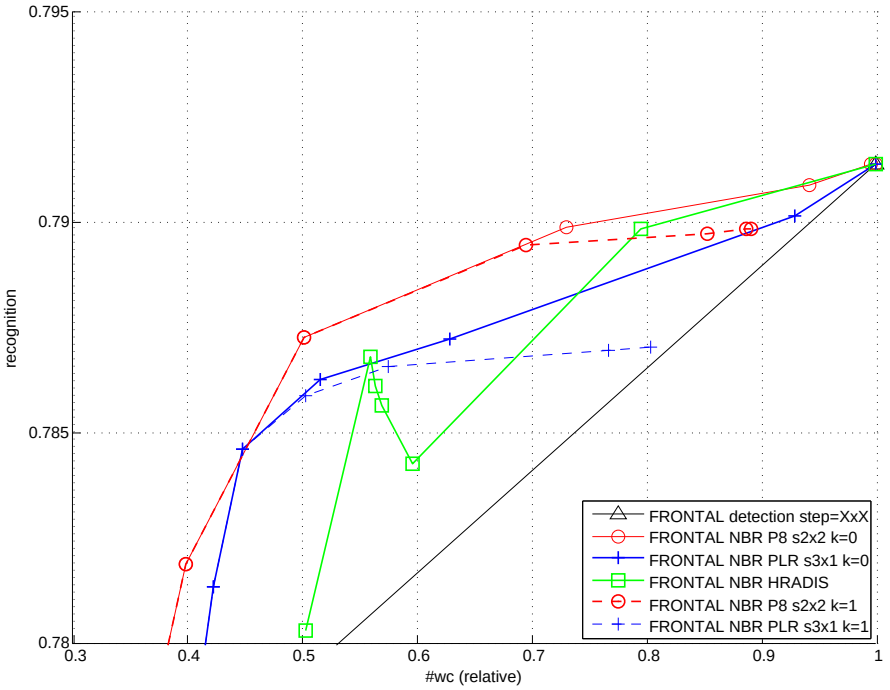


(a) recognition / relative number of evaluated weak classifiers per window

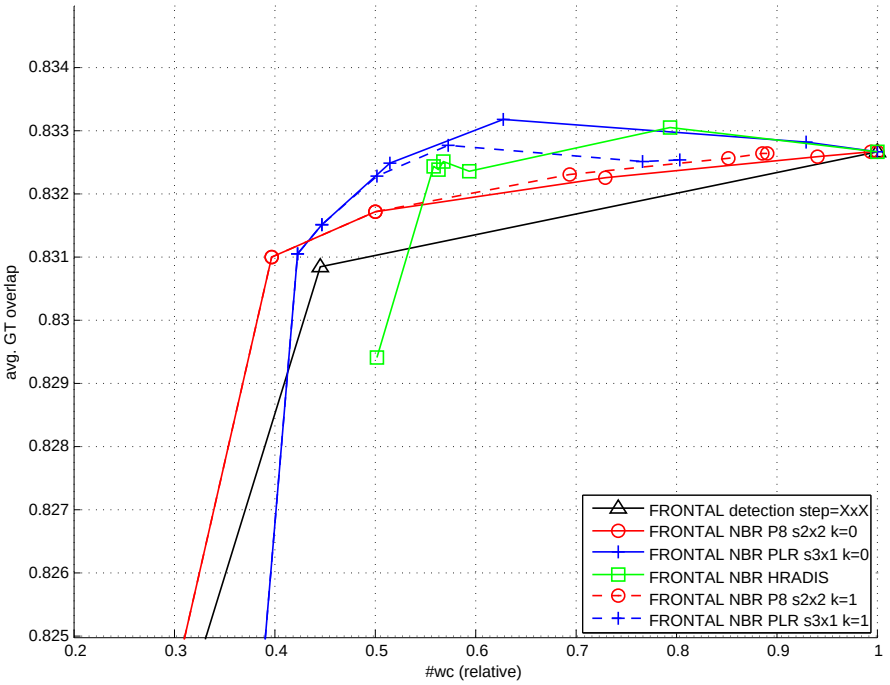


(b) geometric accuracy / relative number of evaluated weak classifiers per window

Figure 6.3: Prediction over neighborhood: multiview detector. Using predictor response as the starting point for the original detector.



(a) recognition / relative number of evaluated weak classifiers per window



(b) geometric accuracy / relative number of evaluated weak classifiers per window

Figure 6.4: Prediction over neighborhood: frontal detector.

were obtained by shifting the value of θ^{PYR}.

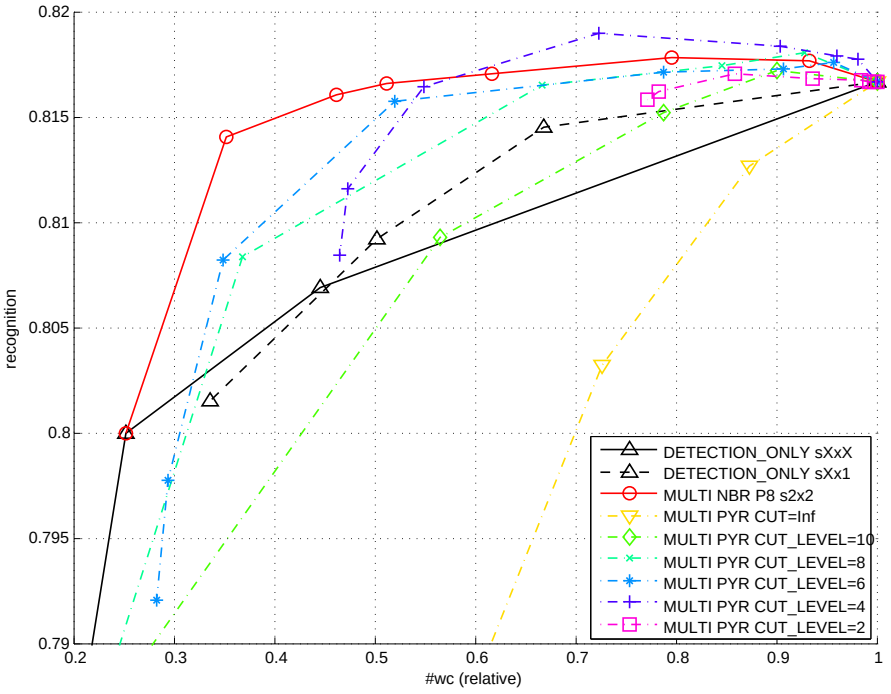
6.3 WaldBoost with Crosstalk Prediction

MULTI. The results of WaldBoost with Crosstalk Prediction are in Figure 6.9. We used the best performing predictor of spatially adjacent windows: P8 predictor with s2:2 scanning pattern and we tested number of different parameters setup of the predictor over pyramid. See that we were capable of reaching 3x, 4x, 5x and 6x speed-up with losing about 0, 0.5, 0.7 and 1.5% of the detection rate respectively with no significant lose of the geometric accuracy. Such a speed-up would be impossible with predicting only a single position as in [2]. Interestingly, 5x speed-up with losing 0.7 % of detection rate result in 1.23 evaluated weak classifiers per window in average while still a keeping state-of-the-art performance. The geometric accuracy also remains almost unchanged up to 5x speed-up.

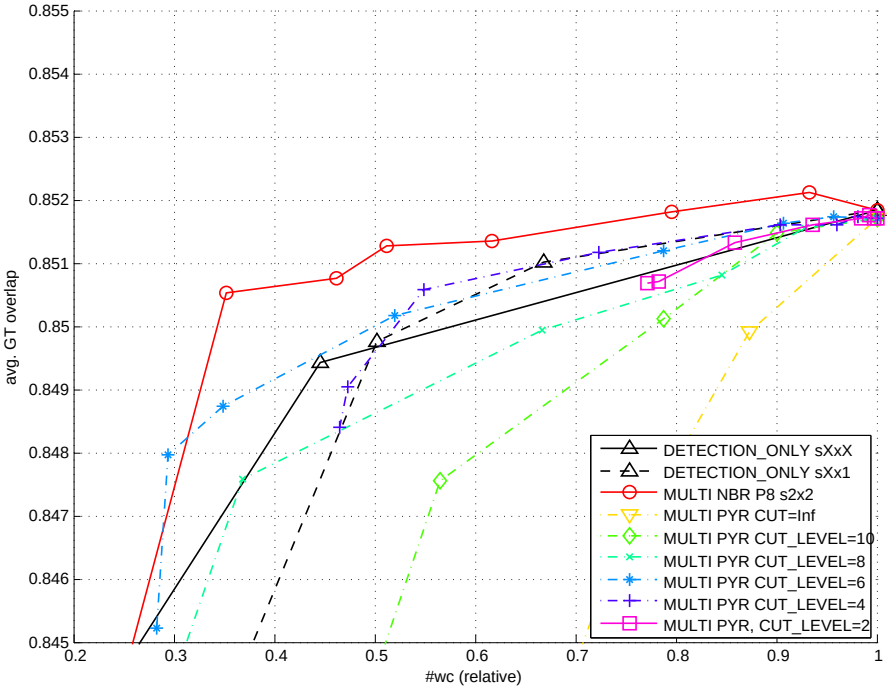
See examples of detections including the relative number of evaluated weak classifiers per window in 6.11.

FRONTAL. See Figure 6.10. Identically to MULTI, we used P8 s2:2 as the predictor of spatially adjacent windows. We reached 1.5x, 3x, 4x and 5x speed-up with losing about 0, 0.7, 1.5 and 2.7 % of the detection rate with almost no loss of the geometric accuracy. The loss in recognition is a bit more significant here than in case of MULTI, but considering the average number evaluated weak classifiers per a single window, we get 1.46, 0.73, 0.54 and 0.4, therefore less than one evaluated classifier per 2 windows, which is very decent when considering not the top-notch, but still very high detection rate. Furthermore, the quality of geometric accuracy remains almost the same as the reference detector.

In the figures, each point represents a single ROC curve. ROC curves were obtained by shifting the value of θ^{PYR}.

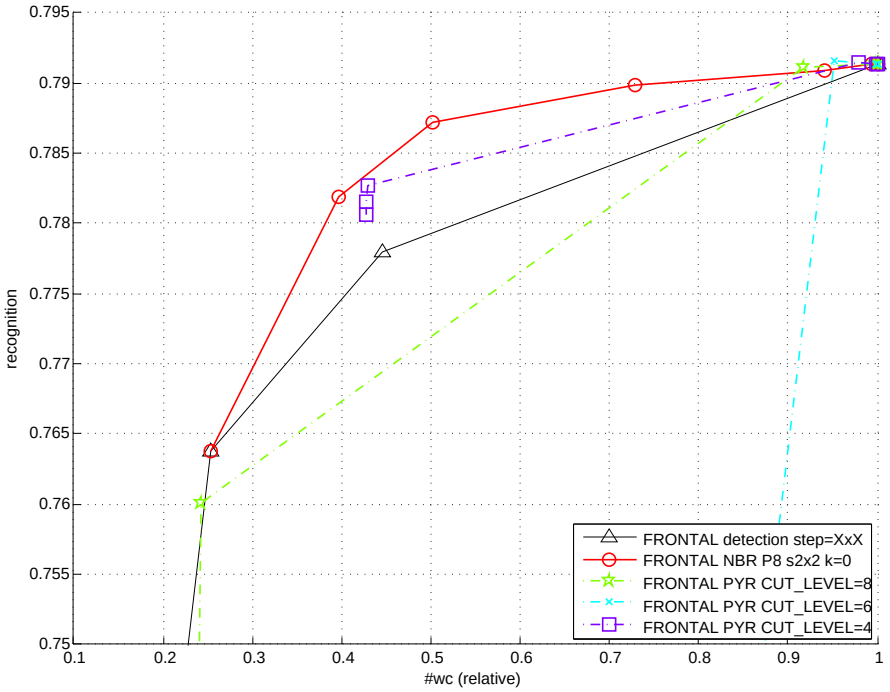


(a) recognition/ relative number of evaluated weak classifiers per window

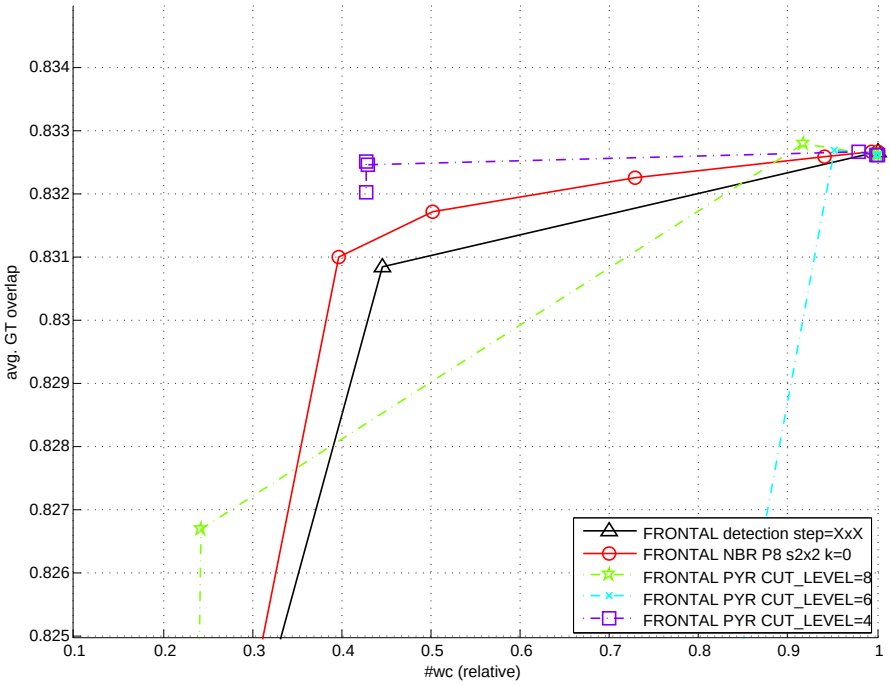


(b) geometric accuracy / relative number of evaluated weak classifiers per window

Figure 6.5: Prediction over pyramid: multi-view detector.



(a) recognition / relative number of evaluated weak classifiers per window



(b) geometric accuracy / relative number of evaluated weak classifiers per window

Figure 6.6: Prediction over pyramid: frontal detector.

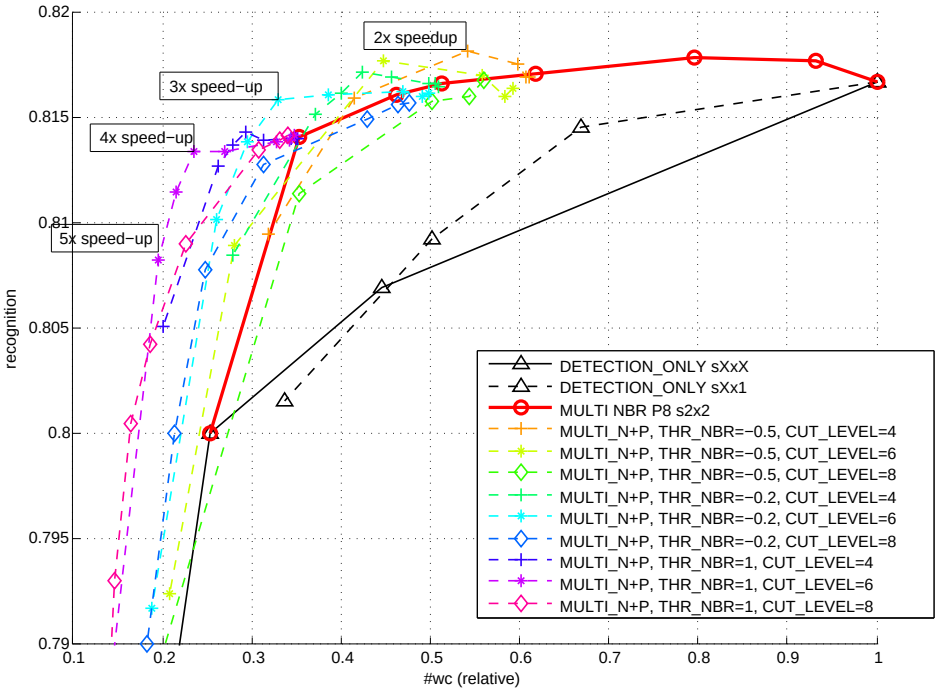
(a) 0.03 (b) 0.06 (c) 0.16

(d) 0.21 (e) 0.24 (f) 0.28

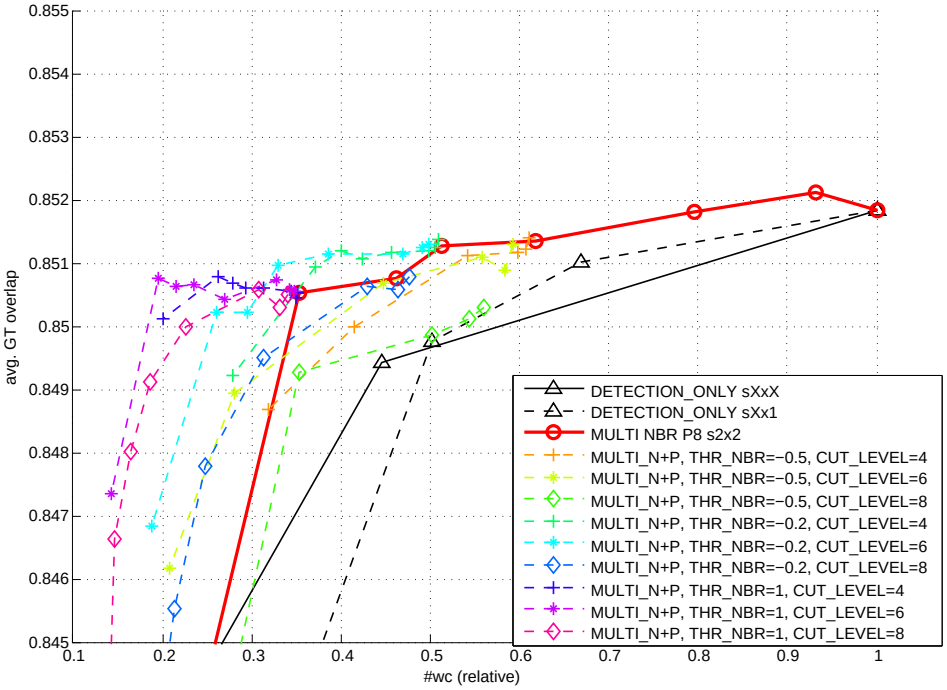
(g) 0.41 (h) 0.44 (i) 0.77

Figure 6.7: Prediction over pyramid: successful detections. Numbers in captions represent a relative number of evaluated weak classifiers compared to the reference detector without prediction.

Figure 6.8: Prediction over pyramid: failures. Red boxes correspond to windows, that were discarded due to a low predictor response, although the reference detector would detected the face on lower image pyramid levels. Predictor responses for these were -1.6, -2.5 and -3.5 respectively.

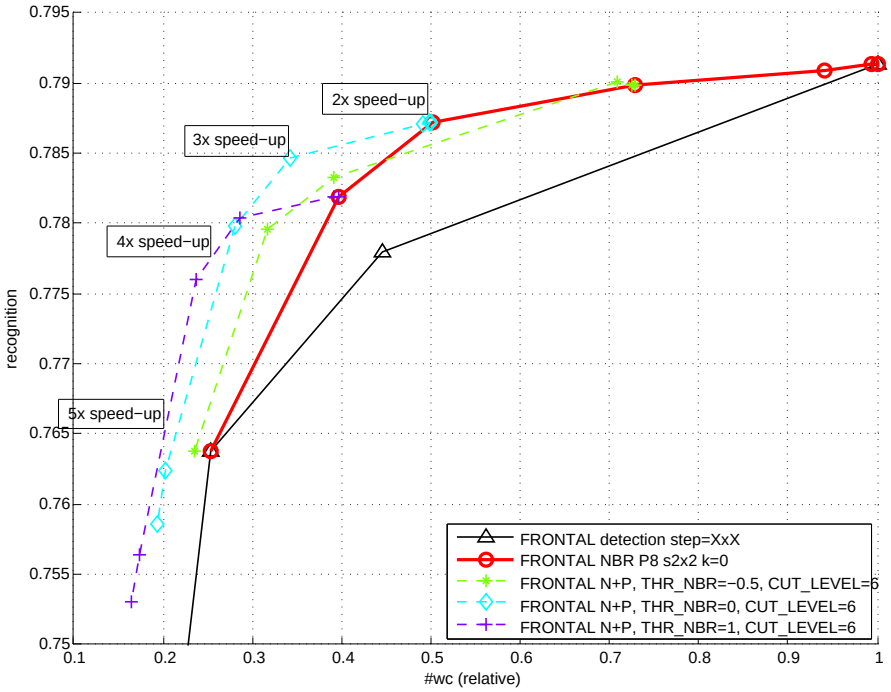


(a) recognition / relative number of evaluated weak classifiers per window

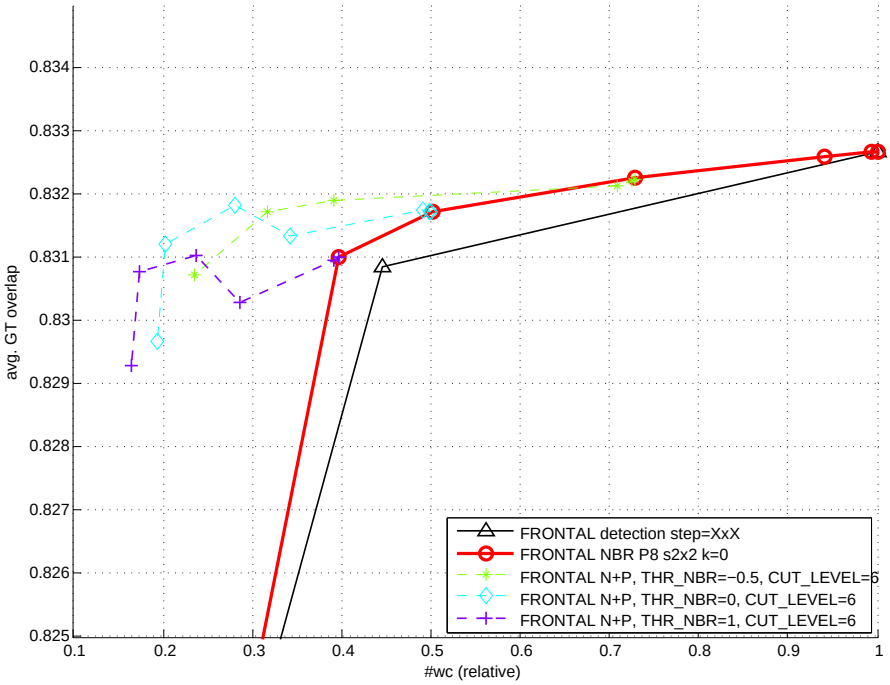


(b) geometric accuracy / relative number of evaluated weak classifiers per window

Figure 6.9: WaldBoost with Crosstalk Prediction: multi-view detector

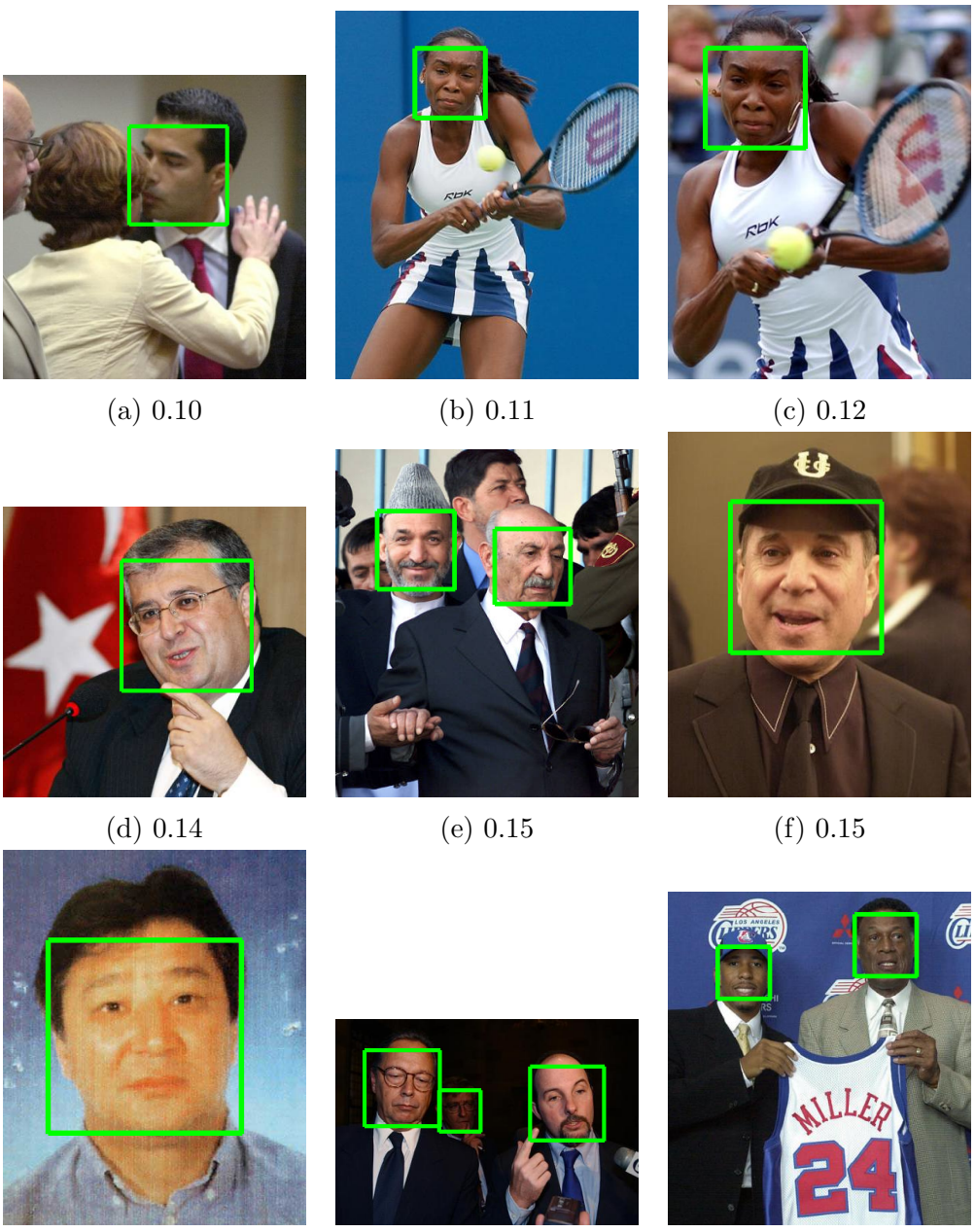


(a) recognition / relative number of evaluated weak classifiers per window



(b) geometric accuracy / relative number of evaluated weak classifiers per window

Figure 6.10: WaldBoost with Crosstalk Prediction: frontal detector.



(a) 0.10 (b) 0.11 (c) 0.12

(d) 0.14 (e) 0.15 (f) 0.15

(g) 0.16 (h) 0.18 (i) 0.19

Figure 6.11: WaldBoost with Crosstalk Prediction detection results. Numbers in captions represent a relative number of evaluated weak classifiers compared to the reference detector without prediction.

Chapter 7

Conclusion

The scanning strategy and the selection of predictors is a significant factor in quality of the prediction algorithm. In this work, we generalized the idea of exploiting information from spatially neighboring windows to multiple scales and we proposed new predictors for spatially adjacent windows and evaluated their performance. We proposed a novel WaldBoost with Crosstalk Prediction, which uses the information shared between windows that overlap spatially or over image pyramid.

Inspired by work of Hradiš et. al. [2], the prediction is computed on the same features as the detection, therefore no additional computational cost is required (adding one additional look-up table results in 1.1 times longer processing time). We evaluated the detection performance with the prediction on state-of-the-art dataset for face detection, when the detection rate, geometric accuracy and speed were measured. We used 2 reference WaldBoost detectors: frontal-view and multi-view detector. For both detectors, experiments showed that a significant speed-up can be achieved with no or a little loss of detection rate and geometric accuracy, outperforming the reference method of Hradiš et. al.

Testing with a multi-view detector, which computes about six weak classifiers per evaluated window, the final detector using the best performing of the predictors was 3 times as fast as the reference detector without prediction with no loss of the detection rate and up to 5 or 6 times as fast with losing only 0.7% or 1.5.% of the detection rate respectively.

Testing with a frontal-view detector, which computes about 2 weak classifiers per evaluated window, the final detector was about 3 times faster when losing less then 1% and more than 4 times faster when losing 1.5% of the detection rate, which is equal to less than one evaluated weak classifiers per 2 windows.

Evaluation of proposed method on other dataset for face detection and

other object classes (pedestrians, cars) is one of topics for future work. New scanning patterns and possibly new neighborhood types can be tested to improve the prediction of spatially adjacent windows. Optimizing the scale parameter for the prediction over pyramid could also be a topic for investigation. The prediction over pyramid proved to have a potential for a great speed-up, therefore it is worth considering, whether it shouldn't be treated as an additional standalone classifier instead of training it to reuse the features of the reference detector.

Bibliography

[1] J. Trefny and J. Matas. Extended set of local binary patterns for rapid object detection. In *Computer Vision Winter Workshop, Czech Republic*, 2010.

[2] Pavel Zemčík, Michal Hradiš, and Adam Herout. Exploiting neighbors for faster scanning window detection in images. In *ACIVS 2010*, LNCS 6475, page 12. Springer Verlag, 2010.

[3] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. pages 511–518, 2001.

[4] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *ECCV*, 2014.

[5] *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*. IEEE Computer Society, 2012.

[6] Jan Sochman and Jiri Matas. Waldboost " learning for time constrained sequential detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 150–156, Washington, DC, USA, 2005. IEEE Computer Society.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, page 2012.

[8] Henry Schneiderman. Feature-centric evaluation for efficient cascaded object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2004.

[9] P. Dollár, R. Appel, and W. Kienzle. Crosstalk cascades for frame-rate pedestrian detection. In *ECCV*, 2012.

[10] Abraham Wald. *Sequential Analysis*. John Wiley and Sons, 1st edition, 1947.

[11] A. Wald. Sequential tests of statistical hypotheses. *Ann. Math. Statist.*, 16(2):117–186, 06 1945.

[12] Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.*, 37(3):297–336, December 1999.

[13] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51 – 59, 1996.

[14] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, July 2002.

[15] Lun Zhang, Rufeng Chu, Shiming Xiang, Shengcai Liao, and Stan Z. Li. Face detection based on multi-block lbp representation. In *Proceedings of the 2007 International Conference on Advances in Biometrics*, ICB'07, pages 11–18, Berlin, Heidelberg, 2007. Springer-Verlag.

[16] Vidit Jain and Erik Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.