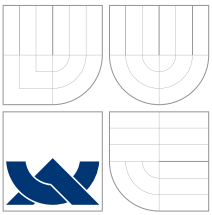
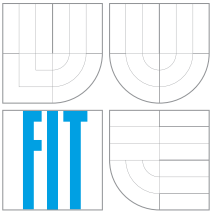


BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

SHARING LOCAL INFORMATION FOR FASTER SCANNING-WINDOW OBJECT DETECTION

SDÍLENÍ LOKÁLNÍ INFORMACE PRO RYCHLEJŠÍ DETEKCI OBJEKTŮ

DOCTORAL THESIS

DISERTAČNÍ PRÁCE

AUTHOR

AUTOR PRÁCE

Ing. MICHAL HRADIŠ

SUPERVISOR

VEDOUCÍ PRÁCE

Prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2014

Curriculum vitae

EXPERIENCE

2013–2014 (14 months) **Cortexica Vision Systems Ltd** – *Researcher in content-based image retrieval*
Responsible for development of novel retrieval and image description methods.

2006–2012 (6 years) **Department of Computer Graphics and Multimedia, Brno University of Technology** – *Researcher in Computer Vision*

- Managed our participation in EU FP7 project TA2 for two years — including supervision of research, development and integration; representation at project meetings; recruitment; and reporting.
- Managed development of a web-based prototype application focused on semi-automatic image/video tagging.
- Led development of a successful image classification system based on local features for TRECVID 2010 and 2011 semantic indexing task, PASCAL VOC 2010 image classification task, and MediaEval 2011 Genre Tagging Task.
- Developed an experimental tool for research on Viola&Jones-like detectors in C++.
- Participated in object detector development for FPGAs and GPUs.
- Led computer vision reading group.

2009 *Part-time developer of computer vision applications*
Worked on a camera-based identity card reader and a camera-based interactive projector

TEACHING

Computer Vision course – Lectures on object detection, AdaBoost, RANSAC, Hough transform, and image classification using local features, individual projects, homeworks.

Classification and Recognition course – lecture on boosting

Thesis supervision – 59 students, mostly working on computer vision or machine learning topics. Examples of topics are: Automatic Image Labeling, Efficient Image Tagging, Deep Learning for Image Recognition, Indexing Events in Video,

Fast Object Detection with Bags of Visual Words, Building deep networks using auto-encoders, Interactive Image Search, Part-Based Models for Object Detection, Video Features for Classification.

Lectures – Computer Vision and Image Processing at PennState Erie, PA, USA; Image Classification at University of Eastern Finland; Support Vector Machines at University of Eastern Finland; Natural User Interfaces at Third Intensive program of Computing Aspects in Computer Game Development summer school

EDUCATION

2007-today	**Doctoral programme** — *Computer Science and Engineering* Brno University of Technology Thesis: Exploiting shared information in object detection
2005–2007	**Master study programme** — *Computer Graphics and Multimedia* Brno University of Technology Thesis: AdaBoost in Computer Vision
2002–2005	**Bachelor study programme** — *Information Technology* Brno University of Technology Thesis: Global illumination of point-based scenes

TEMPORAL STAYS

2010 (4 months)	**University of Surrey, Guildford, UK** *Kristian Mikoljcyzk* - working on better view-point invariance of codebooks for local descriptors
2007 (2 months)	**PennState Erie, PA, USA** Ralph M. Ford

Abstract

This thesis aims to improve existing scanning-window object detectors by exploiting information shared among neighboring image windows. This goal is realized by two novel methods which are build on the ideas of Wald's Sequential Probability Ratio Test and WaldBoost. *Early non-Maxima Suppression* moves non-maxima suppression decisions from a post-processing step to an early classification phase in order to make the decisions as soon as possible and thus avoid normally wasted computations. *Neighborhood suppression* enhances existing detectors with an ability to suppress evaluation at overlapping positions. The proposed methods are applicable to a wide range of detectors. Experiments show that both methods provide significantly better speed-precision trade-off compared to state-of-the-art WaldBoost detectors which process image windows independently. Additionally, the thesis presents results of extensive experiments which evaluate commonly used image features in several detection tasks and scenarios.

Contents

CHAPTER 1

Introduction

Automatic detection of objects in images is an important task with applications ranging from face detection in hand-held cameras and cloud-based photo collections to general scene understanding and human-machine interaction. Development of practical detectors is a scientific and engineering challenge which combines fields of image processing, machine learning, and often hardware acceleration.

The range of methods for object detection is wide. One particular class of methods scans images with a small scanning-window and tries to determine for each of the windows separately if it contains an object of interest or if it contains background. These methods rely on fast classifiers to make the decisions and on efficient features to extract relevant information from the image windows.

Existing scanning-window detectors are fast and precise, able to detect even small objects in Full HD video in real-time. However, computational resources are still not sufficient in some situations and precision of detection has to be sacrificed for speed.

One drawback of many scanning-window detectors is that they process each image window independently even though they overlap and share lot of common information. In this thesis, I propose to make use of the shared information to improve existing detectors.

I explore the idea of sharing local information and I refine it into two novel

practical detection methods. The first method augments existing detectors by an ability to suppress their evaluation at neighboring position in an image. This way, the detector is evaluated fewer times, saving significant computational effort.

The second method relies on the fact that objects cannot occupy the same space in an image. If two objects were too close, a detector would not be able to detect them anyway due to occlusion. This method lets neighboring image positions compete among themselves. It progressively evaluates small parts of a detector at the neighboring positions and gradually reject those positions which will not, with high probability, give the best detection score.

The proposed methods efficiently use the information shared among neighboring image positions, and thus push speed-precision envelope of a range of state-of-the-art detectors. Moreover, the two methods accelerate detection in different parts of an image. The *neighborhood suppression* is effective in background areas while the benefit of letting the detector locally compete improves speed mostly around objects. Because of that, the methods complement each other very well and should provide even greater benefits when combined.

1.1 Summary of Contributions

This thesis contributes to the state-of-the-art of appearance-based object detection methods. It explores an idea that existing *scanning-window detectors* [17] could be improved by exploiting dependencies between neighboring image windows. The idea is refined into two novel, practical, and in certain aspects complementary methods which utilize the shared information to improve detectors. Both methods are demonstrated on specific detectors resulting in two practical detection algorithms.

The methods are general and are not limited to any specific type of detectors. The only requirement is that the detectors have to be decomposable into fragments which provide meaningful discriminative information. Exemplar applications presented in this thesis are based on *soft cascade* [17] detectors which satisfy the requirement very well; however, other detectors, such as *detection cascades* [18], *trees*, and multi-object detectors [11], could be considered as well.

Neighborhood suppression. A detection classifier computed at an image window extracts information relevant to other overlapping windows. The *neighborhood suppression* algorithm (Chapter 4) exploits this fact and trains new classifiers to reject neighboring image windows provided they contain background with high con-

fidence. The new classifiers reuse features of an existing detector changing only the classification function. The *neighborhood suppression* can be realized with minimal computational overhead for *soft cascades* and *domain-partitioning weak classifiers* and it can be directly incorporated in existing detection engines requiring only minor modifications. *Neighborhood suppression* was originally published in [21].

Early non-maxima suppression (EnMS). Scanning-window object detection often includes some kind of *non-maxima suppression* which removes overlapping detections with non-maximal responses of the detection classifier. Such suppression decisions are made only after all the classifiers are fully evaluated. EnMS moves the decision to earlier stages of the classifier in order to stop evaluation of the classifiers which would, with high confidence, be rejected by the ordinary non-maxima suppression. Chapter 5 presents the general idea of EnMS together with a practical version of the algorithm which can be applied to *soft cascades*. EnMS is general and can be applied to a wide range of tasks even outside computer vision – any task which searches for the highest response of a suitable classifier in a group of competing objects. Furthermore, EnMS could be modified to handle multiple classifiers evaluated on a single object. EnMS was originally published in [9].

CHAPTER 2

Sequential analysis in object detection

In object detection using the *sliding-window* technique, the decision at each image position can be regarded as a *statistical hypothesis test* where the *null hypothesis* $\mathcal{H}_0$ states that the image patch does not contain an object of interest [17]. The *alternative hypothesis* $\mathcal{H}_1$ is that the patch contains an object of interest.

The most powerful statistical test [20] for single image window can be defined as

$$\frac{p(\mathbf{x}|\mathcal{H}_1)}{p(\mathbf{x}|\mathcal{H}_0)} \geq k, \tag{2.1}$$

were $\mathbf{x}$ is a multi-dimensional vector of features extracted from a single image position and k is the required confidence. In case the features were independent, the functions $p(\mathbf{x}|H_1)$ could be factorized into products of univariate distributions. Unfortunately, features describing the same object are generally not independent, and should be modeled jointly.

A fully joined model $p(\mathbf{x}|\mathcal{H})$ would be complex, hard to estimate, and computationally expensive. Practical detectors which utilize probabilistic models of background and foreground have to make compromises by omitting some of the dependencies.

Sequential statistical test. A. Wald [20] defined a *sequential test of a statistical hypothesis* as a procedure which, at any stage of an experiment where samples are drawn *independently and identically distributed* from an unknown distribution, gives a specific rule, for making one of the three decisions: (1) to accept the null hypothesis, (2) to reject the null hypothesis, (3) to continue the experiment by making additional observation. A novel idea of the sequential test was that the number of observations needed to make a decision was not predetermined, rather, the number of observations was threated as a random variable. This made it possible to adjust the number of observations to each particular instance of an experiment, and thus reduce the average number of observations while maintaining the same expected error level. As is shown in the following text, the ideas of sequential statistical testing can be adapted in fast detection classifiers which compute and use only so many features at each image position such that a predetermined error rates are achieved.

2.1 Optimal Sequential Decision Strategy

In the following text, the sequential test is formalized in a way which is suited for a *two-class classification task* as opposed to the Wald's definition [20] for independent samples drawn from an unknown distribution. The formulation here follows formulations in [17, 19].

Sequential decision strategy. Let $\mathbf{x} \in \mathcal{X}$ be a vector of measurements $x_i \in \mathcal{X}_i$ representing an object. The task is to estimate an unknown class $y \in \{-1, +1\}$ associated with the object based on the values x_i. The sequential test can be formalized as a *sequential decision strategy* $S : \mathcal{X} \to \{-1, +1\}$ which is a sequence of *decision functions* $S = S_1, S_2, \ldots$ Each of the decision functions takes one measurement of the object, and makes its decision based on the previously obtained measurements including the new one – formally $S_t : \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_t \to \{-1, +1, \sharp\}$. The decision strategy terminates when a decision function outputs $+1$ or -1. The symbol '$\sharp$' defers the decision to the following function S_{t+1}.

Strength of a sequential decision strategy S is characterized by its *false negative rate* α_S and its *false positive rate* β_S

$$\alpha_S = P(S(\mathbf{x}) = -1 | y = +1) \quad \text{and} \quad \beta_S = P(S(\mathbf{x}) = +1 | y = -1). \tag{2.2}$$

Second important characteristic of a sequential decision strategy is its *speed* which is expressed as the *number of measurements needed to reach a decision.* This number is a random variable and it will be further denoted as N_S. The average number of measurements $\bar{T}_S = E[N_S]$ depends on the object class. The average number of measurements for the two classes will be denoted as

$$\bar{T}_{S,-1} = E[N_S|y=-1] \quad \text{and} \quad \bar{T}_{S,+1} = E[N_S|y=+1]. \tag{2.3}$$

A sequential decision strategy S^* is considered to be *best* [20] or *evaluation-time-optimal* [19] if it provides the lowest $T_{S*,-1}$ and $T_{S*,+1}$ compared to any other decision strategy of *equal strength* – of those decision strategies that have equal *false negative rate* α_S and *false positive rate* β_S.

Sequential Probability Ratio Test. A. Wald [20] proposed a *Sequential Probability Ratio Test* (SPRT) which he believed was an *evaluation-time-optimal* sequential decision strategy. SPRT is defined as a sequential strategy S^* where

$$S_t^*(\mathbf{x}) = \begin{cases} +1, & \text{if } R_t(\mathbf{x}) \leq B \\ -1, & \text{if } R_t(\mathbf{x}) \geq A \\ \sharp, & \text{if } B < R_t(\mathbf{x}) < A \end{cases} \tag{2.4}$$

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

$$R_t(\mathbf{x}) = \frac{p(x_1, \ldots, x_t|y=-1)}{p(x_1, \ldots, x_t|y=+1)}. \tag{2.5}$$

The constraints A and B determine error rates α and β of the test. Wald [20] suggest A and B to be set to their upper and lower bounds, respectively:

$$A = \frac{1-\beta}{\alpha}, \quad B = \frac{\beta}{1-\alpha}. \tag{2.6}$$

2.2 WaldBoost

In order for SPRT to be efficient in a classification task where the measurements are *not independent and identically distributed* (non-i.i.d.), the decision functions (Equation 2.4) have to be evaluated very fast. Ideally, the decision functions should incorporate the new measurements in a computationally simple way which does not depend on the number of measurements taken so far. Additionally, the *order*

of measurements matters in the non-i.i.d. case. The first measurements taken should be those most informative, as those allow to accumulate enough evidence about the decision problem as early as possible, thus reducing average number of measurements needed.

Šochman and Matas proposed *WaldBoost* [17] which avoids computation of the likelihood ratios by projecting the classified objects to a scalar value using *real AdaBoost* [15] classifier, and by reformulating the decision functions accordingly in a way which directly thresholds output of the classifier.

Decision functions for classification. Let $H_t(\mathbf{x})$ be a real-valued output of a classifier incorporating features $1, \ldots, t$, the likelihood ratio R_t (2.5) is reformulated as

$$R_t(\mathbf{x}) = \frac{p\left(H_t(\mathbf{x})|y=-1\right)}{p\left(H_t(\mathbf{x})|y=+1\right)}. \tag{2.7}$$

Assuming the likelihood ratio is a monotonic function of $H_t(\mathbf{x})$, the decision functions (2.4) can be equivalently redefined such that the decision conditions compare the classifier output instead of the likelihood ratio:

$$S_t^*(\mathbf{x}) = \begin{cases} +1, & \text{if } H_t(\mathbf{x}) \geq \theta_B^{(t)} \\ -1, & \text{if } H_t(\mathbf{x}) \leq \theta_A^{(t)} \\ \sharp, & \text{if } \theta_A^{(t)} < H_t(\mathbf{x}) < \theta_B^{(t)} \end{cases}. \tag{2.8}$$

The *thresholds* $\theta_A^{(t)}$ and $\theta_B^{(t)}$ have to be estimated on a suitable dataset such that the conditions are equivalet to the corresponding conditions using $R_t(\mathbf{x})$ (2.4).

For practical purposes, Šochman [19] suggested to treat $H_t(\mathbf{x})$ as a step function with discontinuities at $\theta_A^{(t)}$ and $\theta_B^{(t)}$. Such change transforms the continues density estimation into a discrete estimation with three bins. As a result, the thresholds should be set as strict as possible while satisfying [19]:

$$p\left(H_t(\mathbf{x}) \leq \theta_A^{(t)}|y=-1\right) \;\geq\; Ap\left(H_t(\mathbf{x}) \leq \theta_A^{(t)}|y=+1\right) \tag{2.9}$$

and

$$p\left(H_t(\mathbf{x}) \geq \theta_B^{(t)}|y=-1\right) \;\geq\; Bp\left(H_t(\mathbf{x}) \geq \theta_B^{(t)}|y=+1\right). \tag{2.10}$$

These constraints are based on the probabilities that a sample of a certain class is from one of the decided regions.

A WaldBoost classifier (shown in Algorithm 1) is defined by an ordered set of T *weak classifiers* $h_t(\mathbf{x})$, by the *corresponding thresholds* $\theta_A^{(t)}$ and $\theta_B^{(t)}$, and by the

final threshold γ which is applied to the full classifier response $H_T(\mathbf{x})$ if a decision is not reached earlier.

Algorithm 1 WaldBoost classification [19]

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

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

end
If $H_t(\mathbf{x}) > \gamma$, classify $\mathbf{x}$ to the class $+1$, -1 otherwise.

WaldBoost learning for object detection. The complete WaldBoost learning algorithm is shown in Figure 2. It accepts as an input a large set of training examples P, desired error rates α and β, and a number of training iterations T. The output is a sequential decision strategy represented by an ordered set of weak classifiers $h_t(x)$, $t \in \{1, \ldots, T\}$ and the corresponding decision thresholds $\theta_A^{(t)}$ and $\theta_B^{(t)}$. The algorithm extends *real AdaBoost* by *bootstrapping* (or sampling of the training set) and by the *decision thresholds*.

A weak classifier is learned in each iteration of WaldBoost as in real AdaBoost. It can be selected on a set of examples $\mathcal{T}$ sampled from P. The sampled set $\mathcal{T}$ changes in each iteration and the weights have to be computed accordingly. The decision thresholds are then set such that they satisfy the constraints from Equation 2.9 and Equation 2.10 on the full training set P which is in turn pruned by the thresholds.

Algorithm 2 WaldBoost learning with bootstrapping. [19]

Input:

- sample pool $\mathcal{P} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}\,; \mathbf{x}_i \in \mathcal{X}, y_i \in \{-1, +1\}$
- desired final false negative rate α and false positive rate β
- the number of iterations T

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

1. Sample training set $\mathcal{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ from $\mathcal{P}$
2. Find $h_t(\mathbf{x})$ by real AdaBoost algorithm on training set $\mathcal{T}$ with weights w_t and compute new weights
3. Find decision thresholds $\theta_A^{(t)}$ and $\theta_B^{(t)}$ such that eq. 2.9 and 2.10 hold
4. Throw away samples from $\mathcal{P}$ for which $H_t(\mathbf{x}) \geq \theta_B^{(t)}$ or $H_t(\mathbf{x}) \leq \theta_A^{(t)}$

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

CHAPTER 3

Information sharing in scanning-window detection

In their basic form, *scanning-window detectors* process image regions independently one by one. An advantage of such design is its simplicity which makes it possible to define the detection task as a *standard binary classification* that can be solved by general learning algorithms without any modifications. However, the independent processing is *sub-optimal in terms of computational cost.*

The ways in which existing scanning-window detectors are optimized with respect to detection window overlap can be divided in two basic groups. Many detectors share some computations across image windows in the form of image preprocessing and in the form of common feature or parts.

The second group includes methods which *make local decisions interdependent* in various ways. These methods include detectors which try to minimize the number of processed image windows by exploiting *smoothness* of a particular detector responses, and some detectors improve speed by assuming *minimum distance between objects* in the same way as non-maxima suppression does.

The rest of this chapter overviews existing detectors which locally share information and discusses how the detectors relate to *neighborhood suppression* and EnMS.

Computation sharing. Most scanning-window detectors do not process image windows completely independently. Notably, Dollár et al. [6] extend the idea of integral images [18] to other types of information with their *integral channel features*. The approach was further improved by Benenson et al. [1].

Sharing of features interlinks neighboring positions even further. Such approach was advocated by Schneiderman [16] as *feature-centric computation* which computes several first features densely across a whole image.

Similarly, most *part-based detectors* share *visual words* or *parts*. Detectors based on visual words [3, 12, 13] compute the words from independently of the detection task as a first step similarly to the *feature-centric computation*.

Some part-based detectors detect the parts first and infer positions of objects from responses of the part detectors. For example, Felzenszwalb et al. [7] detect objects from *response maps* of discriminatively trained part detectors.

Smoothness of detector responses. Responses of many detectors are smooth due to their *robustness to small shifts* and *other transformations*. Such smoothness can be used to infer responses in local neighborhoods or to reason about a whole group of regions as about a single homogeneous set. The goal of methods which use the smoothness assumption is usually to minimize the number of windows on which the detector is evaluated.

Chum and Zisserman [3] use discriminative features to locate likely object positions which serve as seeds for *discrete gradient ascent search* for a maximal responses of a window classifier. Related is also the *efficient subwindow search* by Lampert et al. [12] which searches the space of all windows in an image guided by an *upper bound* on the classifier response over a set of rectangles. However, the search can be efficient only if the bound is reasonably tight and computationally efficient, which is possible only for relatively simple classifiers which have high invariance to geometrical transformations.

A successful way how to apply the smoothness assumption to fast detectors with attentional structure is to first scan an image relatively sparsely and then re-scan the promising regions more densely. Examples of such approaches are by Butko and Movellan [2] and Gualdi et al. [8].

A promising method was proposed by Dollár et al. [5]. Their *excitatory cascades* realize the sparse scanning idea with soft cascades. The authors suggest an algorithm which sets excitatory thresholds for stages of an existing soft cascade on an unlabeled set of images such that regions containing positive responses of the

original cascade are missed during the sparse scanning phase only with some small and defined probability. However; the authors do not claim that the thresholds are set in optimal way and, in fact, they are clearly sub-optimal.

Non-maxima suppression assumptions. Non-maxima suppression, which is part of most scanning-window detectors [18, 5], is based on the assumption that two objects can overlap only to a limited extent. This assumption is valid for most detectors as they are usually not able to handle severe occlusions anyway. The assumption allows detectors to merge overlapping responses into a single object position, which is usually the window with the highest detector response.

The assumption of non-maxima suppression can be used to accelerate detection. If the final object position is determined only by the window with the highest responses, responses at neighboring positions are not needed and the detector only has to determine that they are to be suppressed. This idea was utilized, for example, by Pedersoli et al. [14] in their *coarse-to-fine detector* which splits an image into a set of neighborhoods that can contain only one object and searches the neighborhoods in greedy recursive coarse-to-fine fashion. First, the object is localized at a coarse resolution, and the position is further refined at higher resolutions.

An interesting application of the non-maxima suppression assumption is the *inhibitory cascade* by Dollár et al. [5]. The inhibitory cascades evaluate neighboring image positions in parallel and terminate computation of those windows which will likely give non-maximal results. The decisions are based on *ratios of partial cascade responses*. The authors proposed an algorithm which sets thresholds on the response ratios for an existing *soft cascade* using unlabeled images. Although the thresholds are set such that the inhibitory cascade introduces a small and defined error, the thresholds are not optimal in terms of decision speed (why *inhibitory cascades* are not optimal and how they relate to EnMS is discussed in Chapter 6).

Relations to EnMS and neighborhood suppression. All methods which accelerate detectors by *sharing computations of features* or by *image pre-processing* are orthogonal to *neighborhood suppression* and EnMS, and could be combined with the proposed methods for even faster detection.

Many of the methods which strongly rely on *smoothness of detector responses* are not applicable to fast detectors with *attentional structures*, which produce discontinuous responses due to the early terminations. The local search methods [3]

and the branch-and-bound search by Lampert et al. [12] target relatively slow detectors which are not the primary focus of *neighborhood suppression* and EnMS.

The *excitatory cascades* by Dollár et al. [5] focus on the same detectors as *neighborhood suppression* and their underlining idea is similar as well. However, the *excitatory cascades* try to select image positions which should be evaluated and *neighborhood suppression*, in contrast, selects image positions which should be skipped.

The coarse-to-fine detector of Pedersoli et al. [14] is in many aspects related to EnMS, which could, in fact, be applied to a multi-stage coarse-to-fine detector in order to create a detector with similar behavior. An advantage of EnMS is that it produces optimal time-to-decision detector for a target localization error.

The *inhibitory cascades* by Dollár et al. [5] are build exactly on the same idea as EnMS and the way they process images is very similar. The methods differ only in the exact form of the conditions which decide when non-maximal windows are to be rejected, and EnMS, unlike inhibitory cascades, finds thresholds for the decisions which optimize detection speed.

CHAPTER 4

Neighborhood suppression

The algorithm proposed in this chapter extends existing appearance-based detectors with an ability to suppress image positions in the neighborhood of the position being currently classified [21]. The proposed method is effective and, at the same time, simple and computationally inexpensive. It learns a new *suppression classifier* which predicts the responses of the original detector at neighboring positions (see Figure 4.1). When the predictions are negative and confident enough, computation of the detector is suppressed at the respective positions.

The suppression is possible because the neighboring positions share information due to overlap of the image windows caused by small horizontal and vertical scanning steps. In order for the *neighborhood suppression* to be efficient, the detector and the suppression classifier have to *share computation*. These reused parts can be image features in the case of Viola & Jones' [18] and similar detectors or possibly other partial computations. The reuse of computation is crucial and, in fact, it is the only reason why faster detection can be achieved this way.

The *neighborhood suppression* creates new suppression classifiers for an existing *soft cascade* using unlabeled images. The new classifiers are trained by WaldBoost [17] and they reuse features of the original soft cascade.

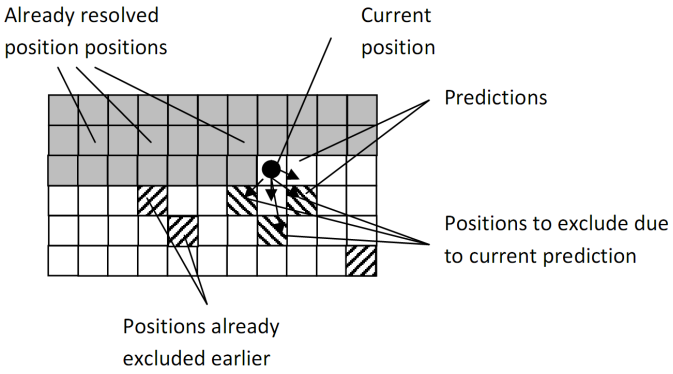


Figure 4.1: Scanning an image in ordinary line-by-line fashion while using *neighborhood suppression*.

4.1 Learning Neighborhood Suppression

A *soft cascade* is a *sequential decision strategy* with decision functions S_t based on a *majority vote* of weak hypotheses $h_t : \mathcal{X} \to \mathbb{R}$:

$$H_T(\mathbf{x}) = \sum_{t=1}^{T} h_t(\mathbf{x}) \tag{4.1}$$

with corresponding decision thresholds (as discussed in Chapter 2).

For *neighborhood suppression*, the three-way decision functions from Equation 2.8) are simplified to:

$$S_t(\mathbf{x}) = \begin{cases} -1, & \text{if } H_t(\mathbf{x}) \leq \theta^{(t)} \\ \sharp, & \text{if } \theta^{(t)} < H_t(\mathbf{x}) \end{cases} . \tag{4.2}$$

Weak hypotheses used in practical detectors [17, 19] are in vast majority of cases *space partitioning weak hypotheses* [15] which internally operate with disjoint partitions of the object space $\mathcal{X}$. The functions partitioning the object space $f : \mathcal{X} \to \mathbb{N}$ will be reffered to in the following text simply as *features*. The space partitioning weak hypotheses are combinations of such features and a *look-up table function* $l : \mathbb{N} \to \mathbb{R}$

$$h_t(\mathbf{x}) = l_t(f_t(\mathbf{x})). \tag{4.3}$$

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

Neighborhood suppression learning algorithm. The task of learning a suppression classifier can be formalized as learning a new soft cascade with a decision strategy S' consisting of hypotheses $h'_t = l'_t(f_t(\mathbf{x}))$, which reuse features f_t of the original detector S, and which only differs in the look-up table functions l'_t and in the rejection thresholds $\theta'^{(t)}$. The goal of the new decision strategy S' is to emulate the original detector at neighboring locations. The whole algorithm for learning suppression classifiers is summarized in Algorithm 3. The learning algorithm is closely related to WaldBoost (see Algorithm 2).

The inputs of the algorithm are target *false negative rate*, existing soft cascade S and a set of unlabeled images.

The target *false negative rate* applies to the binary decision of the suppression classifier. Total change of *false negative rate* of the whole final detector will be lower. This discrepancy is natural and it has two reasons. *Neighborhood suppression* can be performed only within a small neighborhood and, as a consequence, a detector has to be evaluated at many image positions even if all the suppression decisions are successful. Also, the target false negative rate in Algorithm 3 would be reached only if the suppression classifier managed to reject all background positions, which it is not able to do in practice (see Table 4.1).

The *training set* consists of image windows extracted from unlabeled imagesr. The image windows represent positions at which the detector is evaluated, and corresponding labels for the learning task are obtained by evaluating the original soft cascade S at an image position with a particular displacement.

The algorithm proceeds in iterations in which it consecutively creates new weak hypotheses for the suppression classifier – it sets values of the look-up table l'_t and of the early termination threshold $\theta'^{(t)}$ for feature f_t of the original detector S. The look-up table values are set according to *real AdaBoost*. The termination threshold $\theta'^{(t)}$ is set as in WaldBoost (Equation 2.9).

The training set is *pruned* twice in each iteration. First, examples rejected by the new suppression classifier must be removed from the training set. In addition, examples rejected by the original detector S must be removed as well. This corresponds to the behavior during image scanning when only those features which are needed by the original detector to make decision are computed.

Suppression classifiers learned by Algorithm 3 aim to suppress only a *single image position*. However, it can be easily extended to learn such classifiers for suppressing multiple neighboring position. This behavior can be achieved by setting labels of the training samples to -1 only when the original detector rejects all of

Algorithm 3 *Neighborhood suppression* learning algorithm based on WaldBoost as published in [21].

Input:

- original soft cascade S defined by features f_t, corresponding weak hypotheses $h_t(\mathbf{x})$, and rejection thresholds $\theta^{(t)}$
- training set $P = \{(\mathbf{x}_1, y_1) \ldots, (\mathbf{x}_m, y_m)\}, \mathbf{x}_i \in \mathcal{X}, y_i \in \{-1, +1\}$, where the labels y_i are obtained by evaluating the original soft cascade S at an image position with particular displacement with respect to the position of corresponding $\mathbf{x}_i$ in an respective image
- desired miss rate α

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 using f_t such that

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

2. add l'_t to the suppression classifier

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

3. find optimal threshold $\theta'^{(t)}$ satisfying Equation 2.9
4. remove training set samples for which $H_t(\mathbf{x}) \leq \theta^{(t)}$
5. remove training set samples for which $H'_t(\mathbf{x}) \leq \theta'^{(t)}$
6. update the training set weight distribution

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

the considered positions. In addition, multiple suppression classifiers focusing on different parts of a neighborhood can be combined.

4.2 Neighborhood suppression experiments

I tested the *neighborhood suppression* on *frontal face* detection and *eye* detection tasks. In both tasks, two separate test image sets were used - one with less constrained poses and lower quality images and one with easier poses and good quality images.

Face detection experiments were performed on *MIT+CMU* frontal face dataset and on *GroupPhoto* dataset. From these two, *MIT+CMU* contains lower quality images. *GroupPhoto* contains good quality group shots with close to frontal faces. Eye detection experiments were performed on *XM2VTS* database and on *BioID* database. *XM2VTS* is much easier compared to *BioID* as it contains clutter-free backgrounds. Suppression classifiers were trained on a large set of unannotated images containing faces.

The tests were performed with four types of image features: (1)*Haar* – Haar-like features [18], *LBP* – Local Binary Patterns [22], *LRD* – Local Rank Differences [10], and *LRP* – Local Rank Patterns [10]. The base detectors were learned by WaldBoost [17].

Effect of neighborhood suppression. The first experiment focuses on the effect of *neighborhood suppression* using a single classifier to suppress single positions and using twelve such classifiers to suppress twelve different relative positions in the neighborhood. The effects were measured as relative speed-up of detection and relative change in *average detection rate*. The tests were performed with moderately fast base detectors (4.5 - 6 features per position) and moderate target *false negative rate* of the suppression classifiers ($\alpha = 0.05$).

Results of the experiment are shown in Table 4.1 and Figure 4.2. The results indicate large differences between individual feature types. 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% suppression was achieved for *LRD* and 65% for *Haar*. This can be explained by generally higher descriptive power of LBP and LRP features – it is reasonable to expect that they capture lot of information which is not directly relevant to their primary detection task. In general, the *average detection rate* degraded only slightly – by

dataset	value	Haar single	Haar 12	LBP single	LBP 12	LRD single	LRD 12	LRP single	LRP 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
Group	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

Table 4.1: The effect of *neighborhood suppression*. ROCA(%) is the percentage difference between *average detection rate* without and with *neighborhood suppression*. "Time" represents reduction of computations. "single" and "12" stans for suppressing 1 and 12 position, respectively.

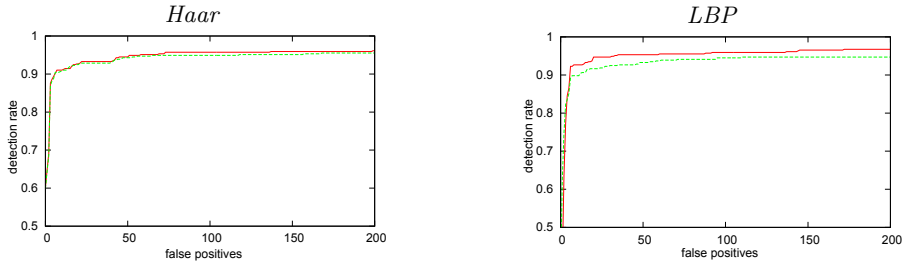


Figure 4.2: The ROC curves on *MIT+CMU* dataset without suppression (full line) and with 12 suppression classifiers (dashed line).

no more than 1% in all cases except for twelve suppressed positions with *LBP* on *MIT+CMU* and *BioID* and with *LRP* on *BioID*.

Suppression distance. This experiment evaluates changes in suppression ability with distance form the evaluated position. Figure 4.3 shows that suppression ability decreases relatively slowly with distance and large neighborhood of radius at least 10 pixels can be suppressed for the tested LBP and LRP classifiers.

Suppressing multiple positions. As mentioned before, single suppression classifier can suppress larger area than just a single position. Relation between speed-up and size of the area suppressed by a single classifier is shown in Figure 4.4. The results show that larger area increases speed compared to suppressing single positions. However, the speedup is not directly proportional to the area size as the suppres-

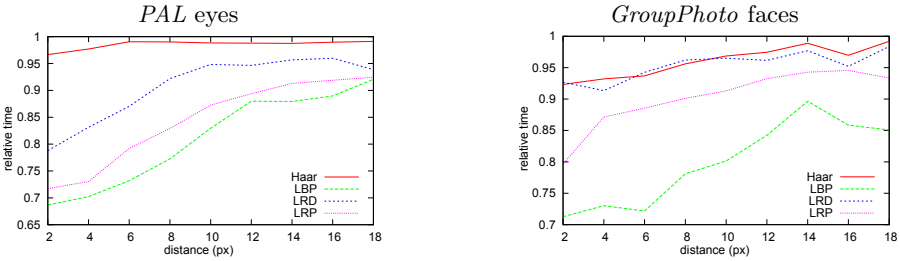


Figure 4.3: Reduction of detection time (y-axis) when suppressing single positions in different horizontal distance from the classified position (x-axis). Target error of the suppression classifiers is 5 %.

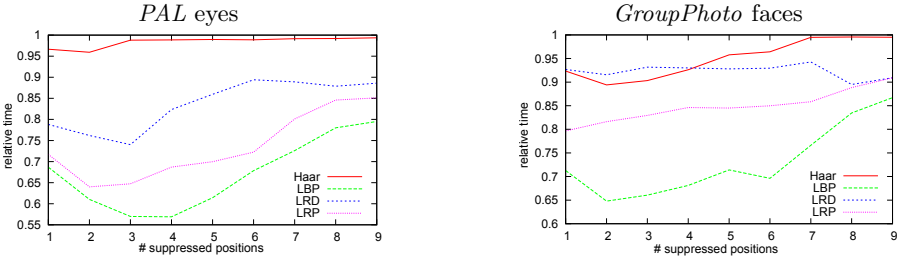


Figure 4.4: Reduction of detection time (y-axis) when suppressing multiple positions by single classifier. x-axis is the number of suppressed positions. Target error of the suppression classifiers is 5 %.

sion task becomes harder with higher number of suppressed positions. Multiple suppression classifiers would always achieve higher speed-up than a single classifier suppressing the same positions. In practical application, the optimal number of suppression classifiers would be determined by the induced computational overhead on the respective platform.

Speed-precision trade-off. If *neighborhood suppression* is to be useful, it has to provide higher speed than the simple detector for the same precision of detection. To validate this, I have trained number of WaldBoost detectors with different speeds (in terms of average number of features computed per position) for each feature type. Then, I learned three suppression classifiers with α set to 0.01, 0.05, and 0.2 for each of the WaldBoost detectors. The corresponding speeds and detection rates of the detectors are shown in Figure 4.5. Even thought only a single suppression classifier of a single position is used in this case for each of the detectors, the results

clearly show that higher speed for the same detection rate can be reached by using *neighborhood suppression*.

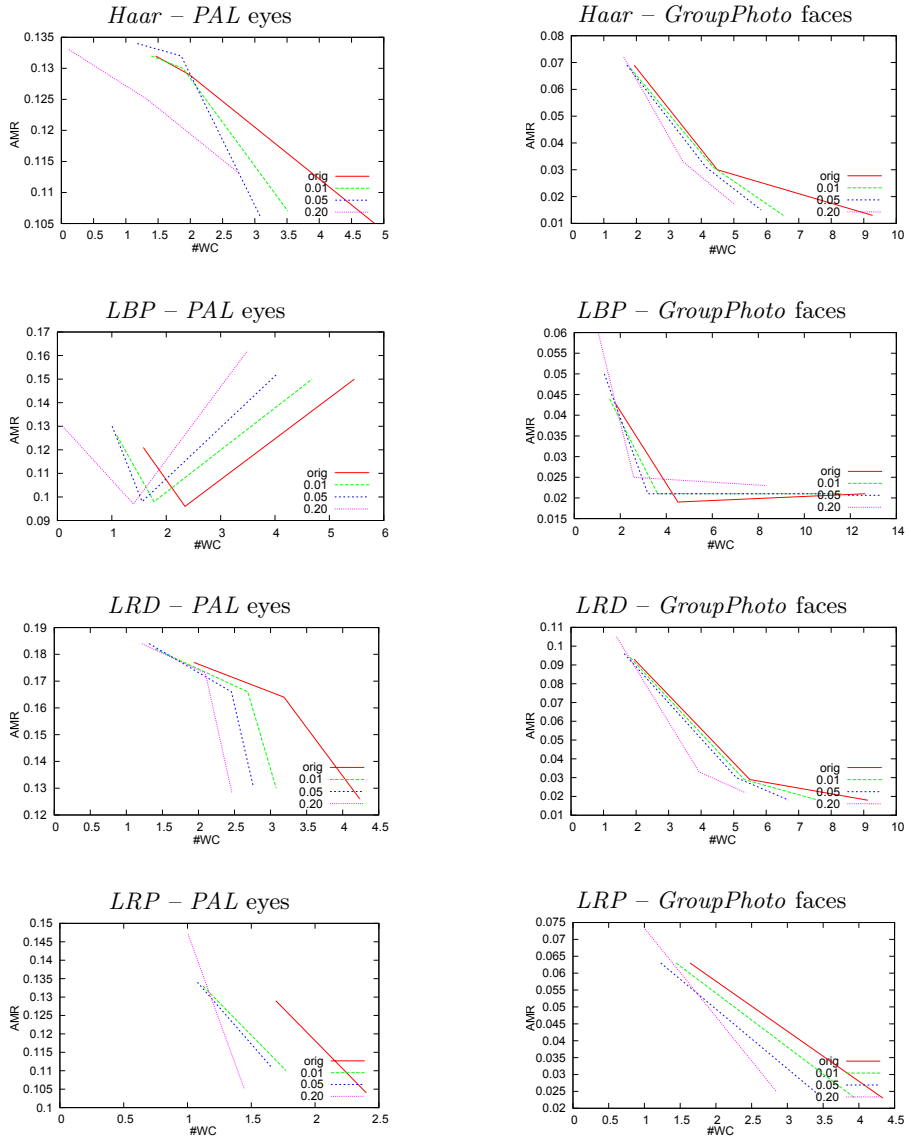


Figure 4.5: Speed-up achieved by suppressing single position for different speeds of the original detector and different target false negative rates α. Neighborhood suppression detectors achieve better speed-precision trade-off. Each line represents results for different α for three original detectors 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 average miss rate (lower is more accurate). Better detectors are closer to the left-bottom corner.

CHAPTER 5

Early non-Maxima Suppression

Non-maxima suppression is an important part of most scanning-window detectors. It aggregates *per-window responses* of a detector into probable object positions, and it suppresses multiple detections of the same object. Non-maxima suppression usually operates locally in a small neighborhood defined by a range of positions, scales, rotations, aspect ratios, and possibly other transformations. In such neighborhood, only the highest response of the classifier which is above a specific threshold is kept and all lower responses are suppressed. As the suppressed responses have no influence on the final detections, there is no need to compute them, and it should be possible to terminate computation of the detector at such positions as soon as it is certain they will, in fact, be suppressed. Such *early terminations* would improve speed without any changes to detection results.

The main idea of *Early non-Maxima Suppression* (EnMS) is to merge existing focus-of-attention approaches with non-maxima suppression, and take the non-maxima suppression decisions from the post-processing step to the classification phase itself. Such shift of the non-maxima suppression decisions could reduce unnecessary computations with only low overhead and could significantly increase detection speed.

The EnMS algorithm proposed in this chapter is formalized as a *sequential decision strategy* and it builds upon the *Sequential Probability Ratio Test* [20] and

WaldBoost [17] which optimize time-to-decision for a certain target error level. It creates a new sequential decision strategy based on an existing *soft cascade* detector by replacing all its rejection thresholds with variable thresholds which depend on tentative results of the detector in neighboring positions. The proposed algorithm only needs an existing detector and a set of unlabeled images.

Although EnMS was primarily motivated by object detection, it is applicable in various other pattern classification tasks where the magnitude of classifier response is significant and the classifier can be divided into separate steps.

5.1 Conditioned SPRT and EnMS

EnMS can be formalized as a *two-class sequential decision problem* where the first class contains samples $\mathbf{x}_{\text{best}}$ which get the highest response of the whole classifier, and the second class contains all the other samples. When formalized in this way, the task is to create an *optimal strategy* which would decide at each stage of the sequential classifier for each sample from a competing set: (1) whether to reject it, (2) whether to accept it as the best sample, (3) or if this problem cannot be decided yet with high enough confidence and further information is needed. Such strategy would compute one stage of the classifier at a time and make the decision simultaneously for each of the competing samples. The following EnMS algorithm is an extension of SPRT (see Chapter 2) and it utilizes the WaldBoost's projection trick for dependent measurements.

Conditioned SPRT. The classification task in the case of EnMS is specific in that the goal is to use information from a set of competing samples to guide the decisions about any of the individual samples. Unfortunately, the original SPRT cannot accommodate such sharing of information and has to be extended. The resulting *Conditioned Sequential Probability Ratio Test* (CSPRT) allows the decision to be conditioned by an arbitrary function over additional data. In CSPRT, the decision functions when combined with the projection trick of WaldBoost (see Equation 2.8) become:

$$S_t^*(x, z_t) = \begin{cases} +1, & \text{if } H_t(x) > \theta_B^{(t)}(z_t) \\ -1, & \text{if } H_t(x) < \theta_A^{(t)}(z_t) \\ \sharp, & \text{if } \theta_A^{(t)}(z_t) \leq H_t(x) \leq \theta_B^{(t)}(z_t) \end{cases} \tag{5.1}$$

where $z_t \in \mathcal{Z}$ is some additional conditioning data and the thresholds on the classifier response $\theta_B^{(t)} : \mathcal{Z} \to \mathbb{R}$ and $\theta_A^{(t)} : \mathcal{Z} \to \mathbb{R}$ are now functions of this additional data. The likelihood-ratio from Equation 2.7, which is used to estimate optimal $\theta_B^{(t)}(z_t)$ and $\theta_A^{(t)}(z_t)$, becomes

$$R_t = \frac{p\left(H_t(x)|z_t, y=-1\right)}{p\left(H_t(x)|z_t, y=+1\right)}. \tag{5.2}$$

CSPRT for EnMS. The goal of EnMS is to find the sample $\mathbf{x}_{\text{best}}$ with maximal response of a classifier $H(\mathbf{x})$ (the champion) among a set of competing samples $\mathcal{X}$ based on the intermediate result of the classifier $H_t(\mathbf{x})$. Whether a classifier response for a sample is maximal or not depends highly on the other competing samples. Considering this, it is reasonable to make z_t a function of $\mathcal{X}$.

$H_t(\mathbf{x})$ becomes very good indicator of the final value of $H(\mathbf{x})$ with increasing t. A good choice for z_t which indicates the probable highest value of $H(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}$ is:

$$z_t = \max_{\mathbf{x} \in \mathcal{X}_{t-1}} (H_t(\mathbf{x})), \tag{5.3}$$

where $\mathcal{X}_{t-1}$ is a set of samples still not decided by the previous decision function S_{t-1}.

Similarly to WaldBoost, it is not practical for EnMS to make positive decisions – it should only reject samples by setting $\theta_B^{(t)}(z_t) = +\infty$. A reasonable form of the negative threshold $\theta_A^{(t)}(z_t)$ is

$$\theta_A^{(t)}(z_t) = z_t - \lambda_t, \tag{5.4}$$

where λ_t can be interpreted as a *handicap* of the leading sample. With this choice of $\theta_A^{(t)}(z_t)$, the condition for rejecting samples as losers from Equation 5.1 becomes

$$H_t(\mathbf{x}) < z_t - \lambda_t. \tag{5.5}$$

Learning Early non-Maxima Suppression. The process of learning an EnMS strategy is depicted in Algorithm 4.

The inputs of the algorithm are the *training sets* of samples $\left\{\mathcal{X}_0^{(k)}\right\}_{k=1}^{N}$, the *target false negative rate* α of the strategy, and a classifier $H(x)$ for which the EnMS strategy should be created. The classifier must provide real-valued responses and must be evaluated in stages $H_t(x)$ where each subsequent stage gives better

Algorithm 4 Learn algorithm for EnMS strategy as published in [9].

Input: classifier $H(\mathbf{x})$ consisting of T stages $H_t(\mathbf{x})$; training sets of samples $\left\{\mathcal{X}_0^{(k)}\right\}_{k=1}^{N}$; target false negative rate α

Output: EnMS handicaps $\{\lambda_t\}_{t=1}^{t_{\max}}$

1: find champions $\mathbf{x}_{\text{best}}^{(k)} = \underset{\mathbf{x}\in\mathcal{X}_0^{(k)}}{\arg\max}\left(H(\mathbf{x})\right)$

2: count losers $L_{\text{all}} = \sum_k \left\| \mathcal{X}_0^{(k)} \setminus \left\{\mathbf{x}_{\text{best}}^{(k)}\right\} \right\|$

3: **for** each stage $t = 1$ to T **do**

4: find all $z_t^{(k)} = \underset{\mathbf{x}\in\mathcal{X}_{t-1}^{(k)}}{\max}\left(H_t(\mathbf{x})\right)$

5: $\lambda_t = \min \tilde{\lambda}_t$, such that $\alpha\frac{L_{\text{killed}}(\tilde{\lambda}_t)}{L_{\text{all}}} > \frac{C_{\text{killed}}(\tilde{\lambda}_t)}{N}$,
where the number of killed losers $L_{\text{killed}}(\tilde{\lambda}_t) =$
$L_{\text{all}} - \sum_{k=1}^{N}\left\|\left\{\mathbf{x}\,\middle|\,H_t(\mathbf{x}) > z_t - \tilde{\lambda}_t, \mathbf{x}\in\mathcal{X}_{t-1}^{(k)} \setminus \left\{\mathbf{x}_{\text{best}}^{(k)}\right\}\right\}\right\|$
and where the number of killed champions $C_{\text{killed}}(\tilde{\lambda}_t) =$
$N - \sum_{k=1}^{N}\left\|\left\{\mathbf{x}\,\middle|\,H_t(\mathbf{x}) > z_t - \tilde{\lambda}_t, \mathbf{x}\in\mathcal{X}_{t-1}^{(k)} \cap \left\{\mathbf{x}_{\text{best}}^{(k)}\right\}\right\}\right\|$

6: prune sample sets
$\mathcal{X}_t^{(k)} = \mathcal{X}_{t-1}^{(k)} \setminus \left\{\mathbf{x}\,\middle|\,H_t(\mathbf{x}) < z_t^{(k)} - \lambda_t, \mathbf{x}\in\mathcal{X}_{t-1}^{(k)}\right\}$

7: **end for**

estimate of the final decision. The individual training sets $\mathcal{X}_0^{(k)}$ each represent one competing set of samples (e.g. local image neighborhood in object detection).

In the first step of the algorithm, the champions $\mathbf{x}_{\text{best}}^{(k)}$ (there is one champion in each set of competing samples, all the rest of the samples are losers) are found in each set of competing samples $\mathcal{X}_0^{(k)}$.

After the initial steps, the algorithm proceeds in iterations $t = 1 \ldots T$. In each of the iterations, a single decision function is estimated starting from the first stage of the classifier. The iterations consist of three steps. In Step 4, the conditioning parameters $z_t^{(k)}$ (see Eq. (5.3)), which are the "best responses so far", are found for all the training sets of competing samples. Then, the only parameter of the stage decision function λ_t (from Equation 5.4) is estimated and, finally, the individual sets of competing samples are pruned by the newly estimated EnMS decision function (from Equation 5.5). Note that the gradual pruning of the sets of samples significantly reduces computational time of later iterations.

The parameter λ_t should be set such that the condition imposed by the threshold (Equation 5.5) rejects only samples for which the likelihood-ratio R_t (Equation 5.2)

Algorithm 5 Execute EnMS

Input: classifier $H(\mathbf{x})$ consisting of t_T stages $H_t(\mathbf{x})$ with corresponding handicaps λ_t, and a set of competing samples $\mathcal{X}_0$

Output: $\mathcal{X}_T$

1: **for** each stage $t = 1$ to T **do**

2: $z_t^{(k)} = \max_{\mathbf{x} \in \mathcal{X}_{t-1}} (H_t(\mathbf{x}))$

3: prune sample sets

$\mathcal{X}_t = \mathcal{X}_{t-1} \setminus \left\{ \mathbf{x} \middle| H_t(\mathbf{x}) < z_t^{(k)} - \lambda_t, \mathbf{x} \in \mathcal{X}_{t-1} \right\}$

4: **end for**

satisfies

$$\begin{aligned} & R_t(\mathbf{x}, z_t) \geq \frac{1}{\alpha}, \text{i.e.} \\ & \alpha p\left(H_t(\mathbf{x}) \,|z_t, y = -1\right) \;>\; p\left(H_t(\mathbf{x}) \,|z_t, y = +1\right), \end{aligned} \tag{5.6}$$

which comes Equation 2.6 when $\beta = 0$. The condition is equivalent to

$$\alpha p\left(H_t(\mathbf{x}), z_t \,|y = -1\right) \;>\; p\left(H_t(\mathbf{x}), z_t \,|y = +1\right). \tag{5.7}$$

The condition divides examples into two disjoint sets, which can be used to reformulate the constraint from Equation 5.7 in terms of these two sets as is done in WaldBoost (see Equation 2.9):

$$\alpha p\left(H_t(x) < z_t - \lambda_t \,|y = -1\right) \;>\; p\left(H_t(\mathbf{x}) < z_t - \lambda_t \,|y = +1\right), \tag{5.8}$$

which is already expressed in terms of λ_t. The handicap λ_t should be set as low as possible while still satisfying this constraint.

EnMS decision algorithm. The algorithm of applying EnMS strategy on a set of samples is described in Algorithm 5. An important feature of EnMS is that it diverges very little from the standard classifier runtime: only the "*best so far*" response must be found after each stage of the classifier and then each instance's response is compared to a calculated threshold (as in the case of most other focus-of-attention strategies). Although this kind of synchronization could be undesirable on some parallel architectures, it requires only minimal additional computation and modern parallel architectures (e.g. CUDA) support constant-time voting operations, such as finding the maximal value among concurrent threads.

5.2 EnMS in face localization

The following text presents EnMS experiments on a *face localization* task. The experiments aim to asses how effective EnMS is compared to attentional detectors which process image windows independently.

The input classifier used in the experiments was a *monolithic real AdaBoost* face detector composed of 1000 weak classifiers based on *LRD* features.

EnMS strategies were learned on a separate training set of unlabelled images (described further) and the strategies were then applied to a separate testing set. Several error rates were measured and are reported in the tables of results:

- "=X" – rate of images where the EnMS strategy rejected the ultimate champion $\mathbf{x}_{\text{best}}$, i.e. $\mathbf{x}_{\text{best}} \notin \mathcal{X}_T$,
- ">X–2" – rate of images where the found best sample's score was different from $H(\mathbf{x}_{\text{best}})$ by more than 2, i.e. $\max_{\mathbf{x} \in \mathcal{X}_T} H(\mathbf{x}) < H(\mathbf{x}_{\text{best}}) - 2$,
- ">X–6" – similarly, $\max_{\mathbf{x} \in \mathcal{X}_T} H(\mathbf{x}) < H(\mathbf{x}_{\text{best}}) - 6$,
- ">2" – rate of images where the reported best sample's score was below 2, i.e. the reported maximum was not a face.

"=X" is the true error of the sequential decision strategy and it should ideally correspond to the target *false negative rate* α. For the classifier used as input, image windows well aligned on objects give $H(\mathbf{x})$ around 40–60, so decision errors which comply the ">X–2" or ">X–6" condition are still well usable for most applications.

WaldBoost detector as a baseline reference. The main question concerning the proposed EnMS approach is what is the real benefit of the additional information shared by the competing samples compared to traditional focus-of-attention mechanisms which do not share such information. To estimate this, we compared the EnMS to WaldBoost [17] face detector with the same properties as the monolithic classifier.

Although the WaldBoost classifier does not directly aim to emulate the monolithic classifier, its task is the same. Also the experiments show that for small target *false negative rate* α, the WaldBoost classifier achieves minimal error rate with respect to the monolithic classifier. Moreover, this or similar approach would probably be used today when optimizing object localization for speed.

EnMS results The dataset was created from images from group "*portraits*" (training set) and "*just_faces*" (test set) from server flicker.com. Training set contained 84,251 images and the test set 6704. The images were then rescaled so that the size of faces was 50-by-50 pixels. Further, the images were cut to a defined size with the faces centered in the middle. The size of training images was 100-by-100 pixels. The test images were cut to 70-by-70, 85-by-85, 100-by-100, 120-by-120, and 150-by-150 pixels.

Results of EnMS with 100-by-100 testing images are given in Table 5.1 and graphically in Figure 5.1 – the figure contains results of the WaldBoost baseline as well. The performance of EnMS is approximately twice as good as the WaldBoost baseline.

target % error	average speed-up	% error "=X"	% error ">X–2"	% error ">X–6"	% error ">2"
1.00	103.3	3.07	1.48	0.31	0.03
2.00	119.4	4.77	2.31	0.60	0.06
5.00	165.0	8.86	5.06	1.67	0.09
10.00	236.3	17.99	11.99	4.79	0.39
16.00	330.6	29.52	22.24	10.99	1.18

Table 5.1: EnMS results on *Dataset B* with image size 100-by-100 pixels.

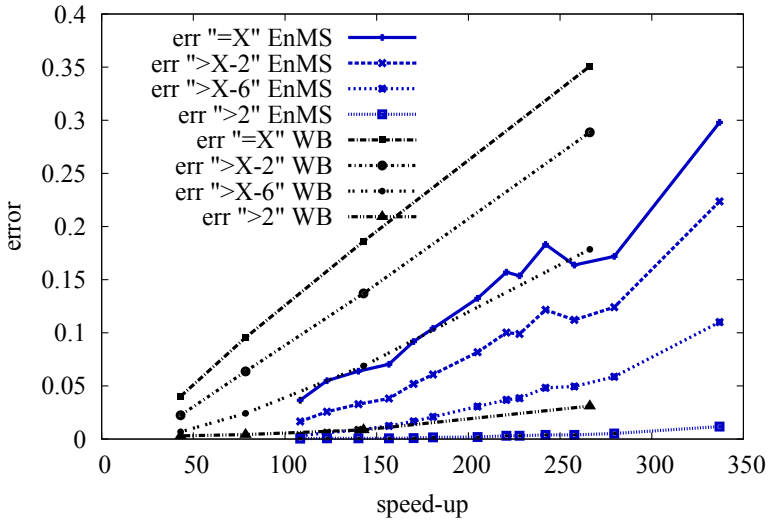


Figure 5.1: Comparison of EnMS and WaldBoost baseline on *Dataset B* with image size 100-by-100 pixels.

Effect of neighborhood size. EnMS should be more effective on larger images as the larger images contain more competing windows. To asses this relation, EnMS strategy learned on the training set (all samples 100-by-100) was executed on the five resolutions of the test sets (see Figure 5.2 for results). Note that the average speed-up of EnMS increases with the number of competing samples. The speed-up is roughly 2× higher on 150-by-150 images than on 70-by-70 images.

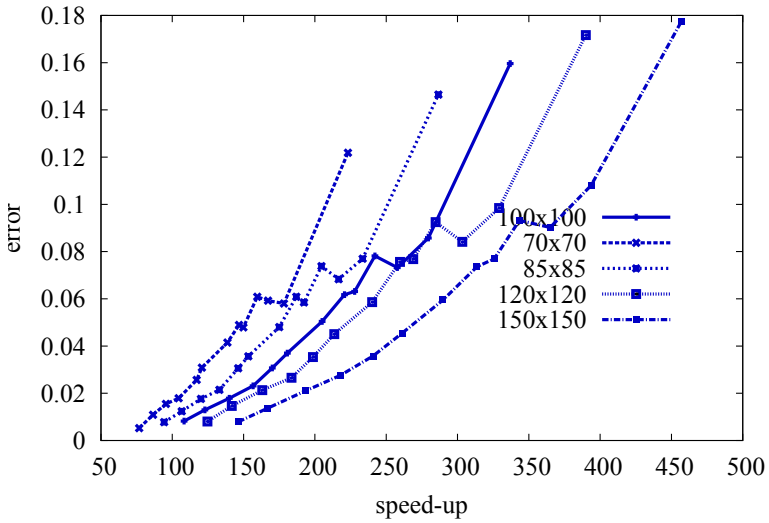


Figure 5.2: Performance of EnMS on test datasets with samples of different dimensions.

CHAPTER 6

Discussion

The experimental results of *neighborhood suppression* (Section 4.2) and of EnMS (Section 5.2) indicate that both methods effectively share information between neighboring image positions, and that they are both able to improve detectors which process image windows independently. EnMS achieved roughly 2× speed-up at same error level compared to WaldBoost in *face localization* on small images. *Neighborhood suppression* improved speed of face detectors up to 3× at the expense of only minor reduction of detection rates (average detection rate was reduced in most cases no more than by 2%). The results show that both *neighborhood suppression* and EnMS provides better *speed-precision trade-off* compaterd to WaldBoost baseline.

The improvements in speed are impressive considering that the baseline WaldBoost detectors are already very fast – the fastest ones compute as few as two features per image window. *Neighborhood suppression* was able to reduce the average number of computed features per window down to single feature in some of the experiments.

From the nature of EnMS, its performance should not depend on specific properties of the detector it is based on, such as which features it uses, as long as the detector conforms to the basic requirements of the method. On the contrary, behavior of *neighborhood suppression* depends strongly on the type of features(see

Table 4.1, Figure 4.3, and Figure 4.4).

Neighborhood suppression. Although *neighborhood suppression* aims to improve speed of an existing detector by sacrificing precision in a controlled way, it, in fact, provides better speed-precision trade-off as does EnMS (as shown in Figure 4.5). Combination of *neighborhood suppression* with a slow and more precise detector achieves on average better detection accuracy compared to WaldBoost detector with the same speed.

A downside of the *neighborhood suppression* as described in this thesis is that it does not provide a mechanism how to create a detector from scratch with specific error or speed. The two stage process which improves existing detectors has its benefits, but the only way an optimal *neighborhood suppression* detector with a specific error rate can be created this way is to try to learn multiple *neighborhood suppressions* for multiple detectors and select the best combination. Such approach would be tedious and time-consuming. Alternatively, *neighborhood suppression* could update rejection thresholds of the original detector while learning the suppression classifiers. Such approach would be similar to the combination of a *soft cascade* and *excitatory cascade* of Dollár et al. [5] and may provide good compromise with respect to complexity of training.

Neighborhood suppression effectively utilizes information shared in neighborhoods for rejection. The approach should be extended to use the evidence extracted from neighboring locations as a starting point for decision at current image location.

Early non-Maxima Suppression The task that EnMS solves is the same as the one addressed by *efficient subwindow search* by Lampert et al. [12] and by the *inhibitory cascades* of Dollár et al. [5]. It is also similar to the taks of *recursive coarse-to-fine localization* by Pedersoli et al. [14]. The problem which these methods solve is to find an image window with the highest response of a detector in a set of windows.

Unlike EnMS, the *efficient subwindow search* is guaranteed to always find the optimal window, but it can only be applied to simple detectors for which an efficient upper bound on detector response exists. Although EnMS is, in a way, constrained with respect to what classifiers it can be applied to as well, it can support classifiers of arbitrary complexity and strength.

The recursive coarse-to-fine localization is an ad-hoc process which, unlike EnMS, does not provide any indications of what is the error caused by the coarse-

to-fine structure of the detector. In fact, EnMS could be applied to a multi-stage coarse-to-fine detector which would result in a detector with similar behavior, but with controlled error and optimal computational complexity for the target error and detector structure.

The closest competitors of EnMS are the *inhibitory cascades* of Dollár et al. [5] which let positions with strong tentative detector responses suppress other positions in a local neighborhood in almost exactly the same way as EnMS. Both methods enhance existing detectors, have similar requirements on the detectors, and require only unlabeled images as training data. *Inhibitory cascades* and EnMS differ in two aspects: (1) exact functional form of suppression conditions, (2) method for choosing suppression thresholds. As the following text argues, the choices made by Dollár et al. in *inhibitory cascades* are not optimal in contrast to EnMS. Considering that both methods have the same computational overhed, EnMS should be considered superior.

Inhibitory cascades base their decisions on the *ratio* of tentative results – a window $\mathbf{x}$ with competing neighbors $\mathcal{X}$ gets suppressed if

$$\frac{H_t(\mathbf{x})}{H_t(\mathbf{x}_{\max})} \;<\; \theta_t, \tag{6.1}$$

where $\mathbf{x}_{\max} = \arg\max_{\mathbf{x}\in\mathcal{X}} H_t(\mathbf{x})$. Although this condition makes certain intuitive sense at the first sight, it becomes less reasonable when the underlying meaning of $H_t(\mathbf{x})$ is considered.

The value of $H_t(\mathbf{x})$ can be directly linked to *log likelihood ratio* [17]:

$$\lim_{T\to\infty} H_T(\mathbf{x}) \;=\; -\frac{1}{2}\log\frac{p(\mathbf{x}|y=-1)}{p(\mathbf{x}|y=+1)} + \frac{1}{2}\log\frac{P(+1)}{P(-1)} \tag{6.2}$$

Even though the limit is defined for infinitely long detectors, it can be safely used for certain reasoning about shorter detectors as well.

The limit can be substituted into the condition used by *inhibitory cascades* (Equation 6.1), resulting in:

$$\frac{p(\mathbf{x}_{\max}|y=-1)}{p(\mathbf{x}_{\max}|y=+1)}\sqrt{\frac{p(\mathbf{x}|y=-1)}{p(\mathbf{x}|y=+1)}} \;>\; e^{\theta_t}. \tag{6.3}$$

Seeing the condition in this form makes it clear that it does not have any clear or meaningful interpretation.

On the other hand, the condition used by EnMS (Equation 5.5), which can be rewritten as

$$H_t(\mathbf{x}) - H_t(\mathbf{x}_{\max}) < \lambda_t, \tag{6.4}$$

can be similarly expressed by substituting the limit from Equation 6.2 as

$$\log \frac{p(\mathbf{x}|y=+1)}{p(\mathbf{x}|y=-1)} - \log \frac{p(\mathbf{x}_{\max}|y=+1)}{p(\mathbf{x}_{\max}|y=-1)} \;<\; \frac{1}{2}\lambda_t. \tag{6.5}$$

As the *logarithmic likelihood ratios* can be interpreted as *certainty levels*, the EnMS condition can be said to be true if $\mathbf{x}_{\max}$ is at least by $\frac{1}{2}\lambda_t$ more likely to contain an object than $\mathbf{x}$ is. The condition does not depend on the certainty of the individual windows, only on the difference. This is a necessary property for the condition to work the same way in regions which certainly contain an object, as well as, in regions which are ambiguous.

The second difference between EnMS and *inhibitory cascades* is that Dollár at al. set the thresholds such that the decisions in all stages induce the same constant error. Such approach does not take into account that the computational savings by rejections in early stages are much greater compared to rejections in late stages. EnMS takes these differences into account and produces optimal time-to-decision detector for the target error.

Comparison of EnMS and neighborhood suppression Although EnMS and *neighborhood suppression* were demonstrated on simple boosted (or WaldBoost) detectors, the approaches can be directly applied to other detectors with similar structure which are composed of stages. These include all detectors with attentional structure [18, 17]. Monolithic detectors [4] would have to be split into meaningful parts first.

Neighborhood suppression and EnMS are, in a certain sense, complementary and most powerful in different situations. *Neighborhood suppression* does not assume anything beyond what is required for existing scanning-window detectors, it behaves as a standard scanning-window detector and it can process image positions sequentially. In essence, it just extends existing attentional structures by an early rejection stage which extracts information from neighboring positions and which is very cheap as it relies on features which would be computed anyway by the neighboring classifiers. This implies that the suppression can not help at regions which are likely to contain objects of interest.

EnMS, on the other hand, diverges from the standard scanning window pro-

cedure and assumes that only the locally maximal position are of interest. It is inherently parallel and requires all image positions to be evaluated concurrently. EnMS should remain effective even in regions which are likely to contain an object of interest as it adapts to the image content. The only requirement for EnMS to be effective is that the competing regions differ in how likely they contain an object. In fact, it is reasonable to expect that EnMS would improve a detector with fixed rejection thresholds mostly in ambiguous regions where the fixed thresholds are not effective.

Neighborhood suppression is closely linked with object detection as it explicitly relies on topological relations. On the other hand, EnMS can be directly applied on any task, even outside computer vision, which uses classifiers and which is interested in finding the highest response in a set of candidates.

CHAPTER 7

Conclusions

This thesis studied scanning-window detectors and, especially, how such detectors can be improved by sharing local information and by interlinking decisions at neighboring positions. This general idea resulted in two novel methods, *neighborhood suppression* and *Early non-Maxima Suppression*, which improve existing scanning-window detectors by utilizing the information shared between neighboring image positions. The methods provide higher speed (up to 2× faster in experiments) at the same detection rates or conversely better detection rates at the same speed compared to detectors which process image windows independently.

Both methods were developed into practical algorithms which can be used in real world applications with minimum changes to existing detection engines on various platforms including highly parallel environments, such as FPGA and GPU. Especially, EnMS matches the nature of highly parallel platforms well, as it requires a high number of competing hypotheses to be computed concurrently in parallel. The novel methods have potential to improve object detectors in a wide range of applications from embedded devices and smart cameras to high-throughput GPU clusters in cloud-based photo galleries and surveillance systems.

The novel algorithms are build upon *Sequential Probability Ratio Test* [20] and *WaldBoost* [17] which optimize time-to-decision for a certain target error level. These ideas were directly used in *neighborhood suppression* and extended into

Conditioned Sequential Probability Ratio Test for EnMS.

Although both *neiborhood suppression* and EnMS were tested on boosted detectors with simple image features and *soft-cascade* attentional structure, they are not in any way limited to these detectors. *Neighborhood suppression* can be directly applied to any detector which can be decomposed into smaller predictive functions (such as features in boosted classifiers). EnMS requires the original detector to be composed of stages which give progressively more confident predictions of the final decision. Also, EnMS, being inspired by non-maxima suppression, finds only the region with the highest response of the detector in a local group of competing regions. Although the requirements of EnMS are stricter, it can be applied to wider range of tasks even outside computer vision – any task which searches for the highest response of a suitable classifier in a group of competing objects.

Although *neighborhood suppression* is able to use information from neighboring positions effectively to suppress evaluation of a detector, the same information could be potentially used even more effectively as initial evidence by the detector. Such tight integration should be further explored as it could lead to significant speed-up without any degradation of detection quality.

EnMS as presented in this thesis becomes less effective on small neighborhoods, such as those used by non-maxima suppression in face detection. To ensure competitiveness of EnMS in such situations, it should be extended by adding WaldBoost-style fixed rejection thresholds. Adding such thresholds does not presents any difficulties; however, an algorithm which sets both types of thresholds in a unified way such that the speed is optimized for specific target error rate should be developed.

Ideally, EnMS should be combined with neighborhood suppression or with a method similar to the *excitatory cascade* of Dollár et al. [5]. Such combination would benefit from the complementary strengths of the methods and it could result in very fast detectors.

Author's selected publications

- Zemčík, P., Juránek, R., Musil M., Musil, P., Hradiš, M.: High Performance Architecture for Object Detection in Streamed Videos, In: Proceedings of FPL 2013, Porto: IEEE Circuits and Systems Society, 2013, pp. 1-4. ISBN 978-1-4799-0004-6.
- Bednařík, R., Vrzáková, H., Hradiš, M.: What you want to do next: A novel approach for intent prediction in gaze-based interaction, In: ETRA '12 Proceedings of the Symposium on Eye Tracking Research and Applications, Santa Barbara, US, ACM, 2012, s. 83-90, ISBN 978-1-4503-1221-9
- Herout, A., Hradiš, M., Zemčík, P.: EnMS: Early non-Maxima Suppression, In: Pattern Analysis and Applications, roč. 2012, č. 2, DE, s. 121-132, ISSN 1433-7541
- Hradiš, M., Eivazi, S., Bednařík, R.: Voice activity detection in video mediated communication from gaze, In: ETRA '12 Proceedings of the Symposium on Eye Tracking Research and Applications, Santa Barbara, US, ACM, 2012, s. 329-332, ISBN 978-1-4503-1221-9
- Hradiš, M., Kolář, M., Král, J., Láník, A., Zemčík, P., Smrž, P.: Annotating images with suggestions - user study of a tagging system, In: Advanced Concepts for Intelligent Vision Systems, Brno, CZ, Springer, 2012, s. 155-166, ISBN 978-3-642-33139-8, ISSN 0302-9743
- Hradiš, M., Řezníček, I., Behúň, K.: Semantic Class Detectors in Video Genre Recognition, In: Proceedings of VISAPP 2012, Rome, IT, SciTePress, 2012, s. 640-646, ISBN 978-989-8565-03-7

- Juránek, R., Hradiš, M., Zemčík, P.: Real-time Algorithms of Object Detection using Classifiers, Real-Time System, Rijeka, HR, InTech, 2012, s. 1-22, ISBN 978-953-510-510-7
- Herout, A., Jošth, R., Juránek, R., Havel, J., Hradiš, M., Zemčík, P.: Real-time object detection on CUDA, In: Journal of Real-Time Image Processing , roč. 2011, č. 3, DE, s. 159-170, ISSN 1861-8200
- Hradiš, M., Řezníček, I., Behúň, K.: Brno University of Technology at MediaEval 2011 Genre Tagging Task, In: Working Notes Proceedings of the MediaEval 2011 Workshop, Pisa, Italy, IT, 2011, s. 2, ISSN 1613-0073
- Herout, A., Zemčík, P., Hradiš, M., Juránek, R., Havel, J., Jošth, R., Žádník, M.: Low-Level Image Features for Real-Time Object Detection, Pattern Recognition, Recent Advances, Vienna, AT, IN-TECH, 2010, s. 111-136, ISBN 978-953-7619-90-9
- Zemčík, P., Hradiš, M., Herout, A.: Exploiting neighbors for faster scanning window detection in images, In: ACIVS 2010, Sydney, AU, Springer, 2010, s. 12, ISBN 978-3-642-17690-6
- Beran, V., Herout, A., Hradiš, M., Řezníček, I., Zemčík, P.: Video Summarization at Brno University of Technology, In: ACM Multimedia, New Yourk, US, ACM, 2008, s. 4, ISBN 978-1-60558-303-7
- Herout, A., Jošth, R., Zemčík, P., Hradiš, M.: GP-GPU Implementation of the "Local Rank Differences" Image Feature, In: Proceedings of International Conference on Computer Vision and Graphics 2008,
- Heidelberg, DE, Springer, 2008, s. 1-11, ISBN 978-3-642-02344-6
- Herout, A., Zemčík, P., Juránek, R., Hradiš, M.: Implementation of the "Local Rank Differences" Image Feature Using SIMD Instructions of CPU, In: Proceedings of Sixth Indian Conference on Computer Vision, Graphics and Image Processing, Bhubaneswar, IN, IEEE CS, 2008, s. 9, ISBN 978-0-7695-3476-3
- Hradiš, M., Herout, A., Zemčík, P.: Local Rank Patterns - Novel Features for Rapid Object Detection, In: Proceedings of International Conference on Computer Vision and Graphics 2008, Heidelberg, DE, Springer, 2008, s. 1-12, ISSN 0302-9743
- Hradiš, M.: Framework for Research on Detection Classifiers, In: Proceedings of Spring Conference on Computer Graphics, Budmerice, SK, UNIBA, 2008, s. 171-177, ISBN 978-80-89186-30-3

- Polok, L., Herout, A., Zemčík, P., Hradiš, M., Juránek, R., Jošth, R.: "Local Rank Differences" Image Feature Implemented on GPU, In: Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems, Berlin, Heidelberg, DE, Springer, 2008, s. 170-181, ISBN 978-3-540-88457-6
- Granát, J., Herout, A., Hradiš, M., Zemčík, P.: Hardware Acceleration of AdaBoost Classifier, In: Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI), Brno, CZ, 2007, s. 1-12
- Šilhavá, J., Beran, V., Chmelař, P., Herout, A., Hradiš, M., Juránek, R., Zemčík, P.: Platform for Evaluation of Image Classifiers, In: Spring Conference on Computer Graphics, Budměřice, SK, UNIBA, 2007, s. 103-109, ISBN 978-80-223-2292-8
- Šilhavá, J., Beran, V., Chmelař, P., Herout, A., Hradiš, M., Juránek, R., Zemčík, P.: Testbench for Evaluation of Image Classifiers, In: Computer Graphics & Geometry, roč. 2007, č. 9, RU, s. 31-47, ISSN 1811-8992

Bibliography

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

[2] N.J. Butko and J.R. Movellan. Optimal scanning for faster object detection. In *CVPR*, pages 2751–2758. IEEE, June 2009.

[3] O. Chum and A. Zisserman. An Exemplar Model for Learning Object Classes. In *CVPR*, 2007.

[4] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, pages 886–893, IEEE, 2005.

[5] P. Dollár, R. Appel, and W. Kienzle. Crosstalk cascades for frame-rate pedestrian detection. In ECCV'12, Springer-Verlag, October 2012.

[6] P. Dollar, S. Belongie, and P. Perona. The Fastest Pedestrian Detector in the West. In BMVC, pages 68.1–68.11. BMVA, 2010.

[7] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–45, September 2010.

[8] G. Gualdi, A. Prati, and R. Cucchiara. A multi-stage pedestrian detection using monolithic classifiers. In *AVSS*, pages 267–272. IEEE, August 2011.

[9] A. Herout, M. Hradis, and P. Zemcik. EnMS: Early non-Maxima Suppression. *Pattern Analysis and Applications*, 2011(1111):10, 2011.

[10] M. Hradiš, A .Herout, and P. Zemčík. Local Rank Patterns - Novel Features for Rapid Object Detection. In *ICCVG*, pages 1–12, 2008.

[11] C. Huang, H.Z. Ai, Y. Li, and S.H. Lao. High-Performance Rotation Invariant Multiview Face Detection. *PAMI*, 29(4):671–686, 2007.

[12] C.H. Lampert, M.B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, pages 1–8. IEEE, June 2008.

[13] J. Li and Y. Zhang. Learning SURF Cascade for Fast and Accurate Object Detection. In *CVPR*, pages 3468–3475. IEEE, June 2013.

[14] M. Pedersoli, J. Gonzàlez, A.D. Bagdanov, and J.J. Villanueva. Recursive coarse-to-fine localization for fast object detection. In *ECCV 2010*, pages 280–293. Springer-Verlag, September 2010.

[15] R.E. Schapire and Y. Singer. Improved Boosting Algorithms Using Confidence-rated Predictions. *Mach. Learn.*, 37(3):297–336, 1999.

[16] H. Schneiderman. Feature-Centric Evaluation for Efficient Cascaded Object Detection. *CVPR*, 2:29–36, 2004.

[17] Jan Sochman and Jiri Matas. WaldBoost - Learning for Time Constrained Sequential Detection. In *CVPR*, pages 150–156, IEEE, 2005.

[18] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. *CVPR*, 1:511, 2001.

[19] Jan Šochman. *Learning for Sequential Classification.* PhD thesis, Czech Technical University in Prague, 2009.

[20] A. Wald. Sequential Tests of Statistical Hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, June 1945.

[21] P. Zemcik, M. Hradis, and A. Herout. Exploiting neighbors for faster scanning window detection in images. In *ACIVS 2010*, Springer Verlag, 2010.

[22] L. Zhang, R. Chu, S. Xiang, S. Liao, and S.Z. Li. Face Detection Based on Multi-Block LBP Representation. In *ICB*, pages 11–18, 2007.