Platform for Teaching Detection Classifiers

Michal Hradiš Roman Juránek Pavel Zemčík
Graph@FIT
Department of Computer Graphics and Multimedia
Faculty of Information Technology, Brno University of Technology
Brno 61266, Czech Republic
{ihradis,ijuranek,zemcik}@fit.vutbr.cz

Abstract — *Scanning using very fast classifiers is a standard and successful approach to object detection in images. This approach became popular after Viola and Jones introduced their frontal face detector in 2001 which was able to reliably detect faces in unconstrained condition and in real-time. Although detecting objects by these classifiers is taught in many computer vision courses, it is not possible for the students to experiment with the methods because the existing implementations require significant initial effort to be able to train and test a classifier and to interpret the detection results in an intuitive way. In this paper, we describe a web-based application which allows experimenting with detection classifiers with minimal initial effort. The application has an intuitive user interface which allows for simple configuration of experiments and management of results. It also provides pre-prepared experiments and datasets in order to further reduce the initial effort. The application is publicly available so anyone can experiment with the detection classifiers and also train detectors on their own data. The application has a potential to become a useful teaching tool for lecturers of computer vision courses and for other interested people.*

1 INTRODUCTION

Scanning using fast classifiers became a standard and successful approach to object detection in images after Viola and Jones [12] introduced their frontal face detector in 2001. At the present, methods derived from the original detector of Viola and Jones provide state-of-the-art detection rates under real-time constraints for various classes of objects [11, 5, 6, 14]. These methods are part of commercial applications ranging from face detection in consumer cameras to video-surveillance and traffic control. Implementations of these methods are also publicaly available and can be used as building blocks for complex computer vision application with relatively low effort.

Although detecting objects by classifiers is taught in many computer vision courses, it is not possible for the students to experiment with these methods because the existing implementations require significant initial effort to be able to train and test classifier and to interpret the results in intuitive way. Moreover, learning new classifier usually takes a long time, possibly in an order of several CPU-days. Such extensive computation may deter students as it is not practical or even possible to realize it in PC labs or on personal laptop. It certainly can not be done during practical lesson.

To address the mentioned issues, we developed a web-based application which allows experimenting with detection classifiers with minimal initial effort. The application provides state-of-the-art learning algorithms, efficient data sampling, multiple types of image features, it facilitates visualization of detector results and it is able to evaluate detailed characteristics of the created classifiers such as speed and detection accuracy. By using efficient image features and lower amount of training data, the application can provide detectors suitable for fast testing in order of minutes, very accurate detectors in couple of hours and the learning rarely takes more than a day even with the most demanding settings. Such low time needed to create a detector allows performing experiments directly as a part of lecture or during a practical lesson. Moreover, the learning runs on remote server and as such it does not require the students to aquire suitable computational resources.

Integral part of the presented application are step-by-step tutorials which allow students to conduct meaningful experiments in under 30 minutes. The students can mutually compare results achieved by their detectors in a simple way. Such possibility of direct comparison increases competitiveness among students and leads in deeper interest in the studied topic. The application allows sharing datasets, configurations, created classifiers and results of the classifiers among its users.

The application includes pre-prepared data for learning detectors of frontal faces, cars and pedestrians together with standard test sets for these types of objects. Other datasets can be simply uploaded by the users together with ground truth annotation and if needed the annotation can be cre-

ated and altered directly in the application.

The web-based application is build on top of Framework for research of detection classifiers [4]. This framework allows training of standard boosted cascades as in the approach of Viola and Jones [12] and it also offers state-of-the-art soft-cascade algorithm WaldBoost [11, 10]. The image features which can be used by the detectors include Haar-like features [12], Local Rank Patterns [5], sparse granular features [6], Multi-Block LBP [13], Extended Histograms of Oriented Gradients [3] and others.

To our knowledge, the only publicly available and useful tool for training of detection classifiers is the implementation of cascade of boosted classifiers with extended set of Haar-like features [8] which is similar to the original algorithm used by Viola and Jones [12] in their frontal face detector. This implementation is available as a part of the OpenCV library. Unfortunately, this tool was not intended for teaching and is not suited for this purpose because of steep learning curve and other reasons stated in the previous text.

Next section of this paper introduces the basic concepts of detection classifiers together with the detector framework which is used as basis for the web-application. The following section describes the web-application itself. Finally, the the paper is concluded in the last section.

2 BACKGROUND

The first practically applicable general-purpose detector of rigid 2D patterns which provided useful detection rates even for uncostrained capturing conditions and was at the same time able to process a video stream in real time was the frontal face detector by Viola and Jones [12]. They achieved amazing performance by combining features based on computationally efficient image filters with a powerful learning algorithm, attentional structure of the classifier, good training dataset and a large amount of computational time. This frontal face detector as well as other similar detectors scan all positions of an image in several resolutions and possibly also several rotations and for each position the classifier tries to estimate if the position contains an object of interest (e.g. face) or whether it contains background. Scanning of an image by a classifier is demonstrated in Figure 1. Considering that to detect an object in any position, scale and orientation requires to scan possibly millions of positions, the classifier has to be extremely fast. The individual parts from which the Viola and Jones detector consists, together reduce average computations per

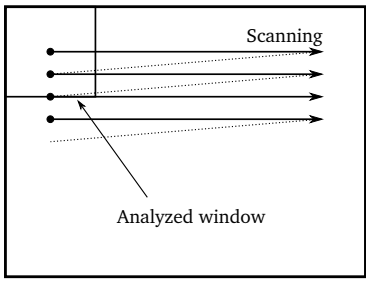

Figure 1: Scanning of an image by floating window.

Figure 2: Different types of Haar-like features used in [12].

single image position to the order of hundereds of instructions.

On the lowest level, Viola and Jones used Haar-like features to extract information from images. These features are 2D linear filters which consist of small number of adjacent rectangular axis-aligned areas. The sum of pixels in these areas is either subtracted or added to the response of the feature. Some prototypes of Haar-like features are shown in Figure 2. These features can be computed very fast and in constant time regardless the size of the feature by utilizing an intermediate data structure called integral image. The integral image stores in each pixel sum of image pixels in a upper-left rectangle defined by this pixel. A sum of arbitrary axis-aligned rectangle can be than computed using only four values in an integral image. To make the classifier more robust to lighting changes the responses of Haar-like features are scaled by reciprocal of standard deviation of pixel values in the scanned image window.

The number of all possible Haar-like features which fit even a small-resolution image is very large. For example, the set of features used by Viola and Jones numbers 180,000 individual features for samples of size 24×24 pixels. To create a compact classifier Viola and Jones used AdaBoost [2]. The AdaBoost algorithm combines simple classifiers into a single powerful classifier. By restricting the simple classifiers to single feature, AdaBoost effectively select a set of most informative features and creates a compact classifier at the same time.

The boosted classifier based on Haar-like features computed on an integral image is itself relatively fast; however, it is not yet suitable for real-time detection. To reduce the detection time further, Viola and Jones utilized an attentional structure which

they call detection cascade and which has form of degenerated decision tree. The cascade consists of several classifiers. Each of these classifiers is trained to reject a portion of positions for which it is certain that they do not contain an object of interest and the rest of position is passed to following classifier in the cascade which makes similar decision. By chaining several such classifiers, a very low false positive rate can be achieved while keeping the detection time low. The reason that the detection time stay low is that most image positions do not contain object of interest and are rejected early in the cascade. Moreover, the decision problem which needs to be solved in the first stages of the cascade is very simple and the first classifier can be very simple (e.g. with only two Haar-like features).

The original approach by Viola and Jones was subsequently extended by many researches. The extensions focused on all aspects of the detector. Some of the more important contributions improve the original cascade structure of the classifier which is not optimal. It discards all information gathered by a stage even though this information is still relevant for the following stages and also the lengths of the stages and their operating points are not optimal. Probably the most important is the work of Šochman and Matas [10]. Their WaldBoost algorithm creates single classifier with rejection threshold after each weak classifier. These thresholds have the same function as the thresholds after stages in cascade, however, the advantage is that no information is lost between the stages. Also, the thresholds are optimal in the sense that they provide the fastest possible decision for requested false negative rate (on the training set). The same detection structure was also proposed by Bourdev and Brandt [1] under the name soft-cascade; however, their method for selecting thresholds is not optimal and lead to much slower detectors.

Beside the Haar-like features, many image features were proposed for object detection in the past. Features like Local Rank Patterns [5], sparse granular features [6], Multi-Block LBP [13] are often used. For the purpose of detection more articulated objects and objects which are defined by distinct edges, various types of Histogram of Oriented Gradients provide good results. Extended Histograms of Oriented Gradients [3] were specifically designed for boosted classifiers and real-time detection.

Different boosting algorithms can be used to construct the classifiers. The most usefull is the real AdaBoost [9] which allows real-valued weak classifier. The real value given by weak classifier is used to convey their confidence. This modification leads in faster convergence of the learning classifier and to more compact and faster detectors.

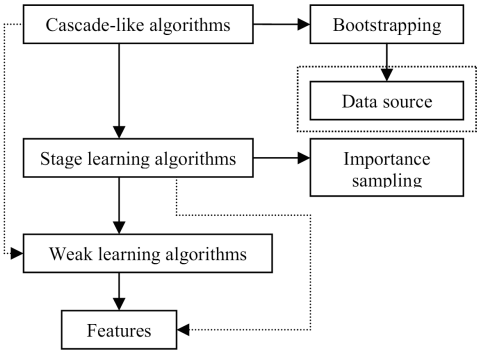

Figure 3: Schematic view of the training framework.

Several researchers investigated ways how to speed-up learning of the detectors. The most efficient way to do so is to sample training data in individual iteration of the boosting algorithm. Kálal et al. [7] discus several data sampling strategies.

The web-application is built on top of a framework originally intended research. The framework supports training and testing of detection classifiers. For detailed description of the framework please refer to the [4]. The framework is written in C/C++ programming language and it is command-line application configured by rather complex XML files. Algorithms contained in the framework are designed for multi-core systems so when multiple CPUs are present the training runs multi-threaded using OpenMP interface.

The main learning algorithm supported by the framewrok is WaldBoost [10]. However, it also supports the original cascade with AdaBoost [12]. The tool aslo offers all image feature mentioned in the previous text, several weak classifiers and also advanced data sampling methods. Together, the possibilities for training classifiers are extensive. The configuration of this tool, however, is very complicated as the number of basic configurable parameters reaches order of tens and the user needs extensive knowledge in order to follow all dependencies (e.g. between types of features and weak classifiers). Until now, no simple interface for generating the XML configurations existed. Basic structure of function blocks of the framework are shown in Figure 3.

3 THE WEB INTERFACE

The interface supports several entities. These entities are images, object types, annotations, image sets, datasets, image features, configurations, detectors, and detector results.

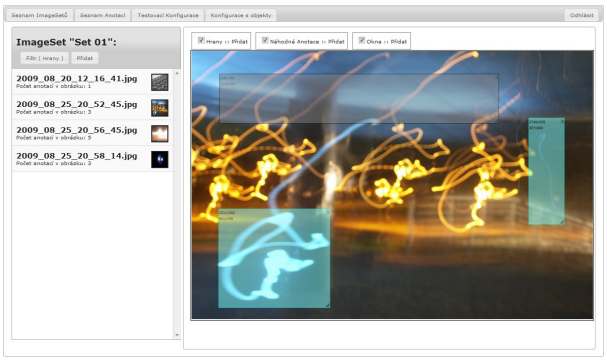

Figure 4: Interface of the web application.

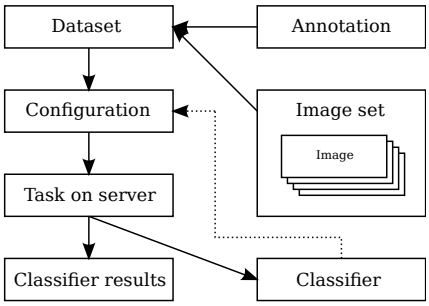

Figure 5: Block diagram of the web application.

Image sets are unstructured collections of images. Images can be freely added to image sets by uploading new images to the server and they can be also deleted. Image sets are the smalest units of data and by selecting an annotated object type and defining how samples should be extracted from images, an image set becomes a dataset which can be already used for training or testing detectors.

The application allows only rectangular axis-aligned annotation. The annotations can be imported for an image set from a simple text file with format:

```
imagename left top right bottom left top right bottom...
imagename left top right bottom...
...
```

Annotations can be also viewed, modified and added directly in a simple image browser (see Figure 4) which is an integral part of the application.

The application distinguishes three types of datasets. The first type is a positive dataset which extracts samples overlapping object annotations. Positive dataset is used in training to extract prototypical samples of object of interest from images. Directly opposite to the positive dataset is negative dataset which extracts samples not overlapping annotated regions. This type of dataset is used to extract background samples for training. The final type of datasets is scanning dataset which is used for testing of classifiers.

In training configuration, the user selects learning algorithm, possibly several types of image features, weak learners, data sampling method and datasets for training and testing. After the configuration is prepares, the user can submit the task to the server where it is added to a queue of waiting jobs. Every waiting task is processed when enough computing resources become available. The user can monitor state of all of his scheduled tasks. When a task is completed, user gets notified by e-mail. The result of a task is usually a new classifier and results of the classifier on possibly several test sets in a form of receiver operating characteristics and average precision. The user can also view responses of classifier in a form of graphical representation and download classifier responses in a form of CSV file. An existing classifier can be tested by simply assigning the classifier to task configuration in which case the provided classifier is used instead of training a new one.

All entities in the application are by default private, but can be made public by the user who created them.

The application contains public annotated datases for training and testing dectors of frontal faces, traffic signs and cars. Very important are also the detailed tutorials which explain training algorithms, data sampling, image features and other aspect of detection classifiers. These tutorials come with prepared configurations which a student can modify and experiment with.

4 CONCLUSIONS

The presented application is publicly available and anyone can experiment with detection classifiers or train detectors on their own data. The application provides simple to use user interface and can be used for very fast experiments as well as for training state-of-the-art detectors. The application has a potential to become a useful teaching tool for lecturers of computer vision courses and for other interested people.

In the future work, we will focus on enhancing of the user interface, which is not perfect yet, and extending it with new functionality to provide better user experience. As the training back-end is still under development, we will constantly add new features to the interface as they become available.

Acknowledgments

The work presented in this paper was funded in part by the FRVŠ, Ministry of Education, Youth

and Sports, FR2786/2010G1 "Support for teaching of detection classifiers for computer vision" and the BUT FIT grant FIT-10-S-2.

References

[1] Lubomir Bourdev and Jonathan Brandt. Robust object detection via soft cascade. In *CVPR*, 2005.

[2] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[3] C. Hou, H.Z. Ai, and S.H. Lao. Multiview pedestrian detection based on vector boosting. In *ACCV07*, Lecture Notes in Computer Science, pages I: 210–219, 2007.

[4] Michal Hradiš. Framework for research on detection classifiers. In *Proceedings of Spring Conference on Computer Graphics*, pages 171–177, 2008.

[5] Michal Hradiš, Adam Herout, and Pavel Zemčík. Local rank patterns - novel features for rapid object detection. In *Proceedings of International Conference on Computer Vision and Graphics 2008*, number 12 in Lecture Notes in Computer Science, pages 1–12. Springer Verlag, 2008.

[6] C. Huang, H.Z. Ai, Y. Li, and S.H. Lao. High-performance rotation invariant multiview face detection. *PAMI*, 29(4):671–686, April 2007.

[7] Z. Kalal, J.G. Matas, and K. Mikolajczyk. Weighted sampling for large-scale boosting. In *BMVC08*, pages xx–yy, 2008.

[8] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *IEEE ICIP 2002*, pages 900–903, 2002.

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

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

[11] Jan Sochman and Jiri Matas. Learning fast emulators of binary decision processes. *International Journal of Computer Vision*, 83(2):149–163, June 2009.

[12] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:511, 2001.

[13] Lun Zhang, Rufeng Chu, Shiming Xiang, ShengCai Liao, and Stan Z. Li. Face detection based on multi-block lbp representation. In *ICB*, pages 11–18, 2007.

[14] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and Shai Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1491–1498, Washington, DC, USA, 2006. IEEE Computer Society.