Annotate. Train. Evaluate. A Unified Tool for the Analysis and Visualization of Workflows in Machine Learning Applied to Object Detection

Michael Storz[1], Marc Ritter[1], Robert Manthey[1], Holger Lietz[2], and Maximilian Eibl[1]

Technische Universität Chemnitz
[1] Chair Media Informatics, Chemnitz, Germany
{michael.storz,marc.ritter,robert.manthey,
maximilian.eibl}@informatik.tu-chemnitz.de
[2] Professorship on Communications Engineering, Chemnitz, Germany
holger.lietz@etit.tu-chemnitz.de

Abstract. The development of classifiers for object detection in images is a complex task that comprises the creation of representative and potentially large datasets from a target object by repetitive and time-consuming intellectual annotations, followed by a sequence of methods to train, evaluate and optimize the generated classifier. This is conventionally achieved by the usage and combination of many different tools. Here, we present a holistic approach to this scenario by providing a unified tool that covers the single development stages in one solution to facilitate the development process. We prove this concept by the example of creating a face detection classifier.

Keywords: Model-driven Annotation, Image Processing, Machine Learning, Object Detection, Workflow Analysis.

1 Introduction

Object recognition has numerous application areas and can be applied in a variety of different fields like image retrieval, driver assistance systems or surveillance technology. In the context of Human Computer Interaction (HCI), object recognition can improve the interaction process leading to more proactive devices, for instance, by enabling technical devices to analyze and recognize properties of potential users like number, position, age or gender of persons. Devices could adapt their interface or their displayed content based on this context prior to the actual usage, thereby potentially leading to a higher usability.

To harness this increased interaction potential, developers need to either use existing technology or create their own object recognition classifiers. If existing technology is either ruled out because of their proprietary nature or because of the limited range of covered object categories, the latter option needs to be explored. State of the art results of the *PASCAL VOC 2012 Challenge* [1] show that object recognition techniques yield reasonable results with an average precision between 57.8% and 97.3 % on 20 different object classes. The chosen object classes are exemplary, so that object

M. Kurosu (Ed.): Human-Computer Interaction, Part V, HCII 2013, LNCS 8008, pp. 196–205, 2013.

recognition techniques can be applied to a far higher amount of object categories, most likely with comparable results.

The open source image database *ImageNet* [2] offers more than 14 million images of different object categories. It is organized according to the *WordNet* hierarchy [3]. Even very specific categories may contain quite large numbers of images. For example, *ImageNet* contains 2,022 images for the object category 'King Penguin' alone. The creation of classifiers for object recognition is a challenging task. One hand the amount of possible object categories is very large and available annotations lack detail, e.g. concerning the alignment and positioning of objects. Two, the training and evaluation processes are very time-consuming and repetitive tasks that require domain specific knowledge. This is mainly caused by large datasets and the high dimensionality of the classification problems. Due to the absence of tools for automated comparison of annotated data with actual classifier results evaluation it is also often a cumbersome process because many evaluations are done by counting results manually. In order to simplify this process, we have built a tool that captures the classifier development from intellectual annotation to training and evaluation in a single holistic environment.

In section 2 we state related work that covers parts of pattern recognition design process. Section 3 describes the developed system. In section 4 we apply our system to the specific task of face detection to proof the systems applicability in to this and related tasks. Finally section 5 outlines future research areas.

2 Related Work

The design of a pattern recognition system usually involves the processes: data collection and pre-processing, data representation, training, and decision making [4]. Our developed tool incorporates these processes and divides them into three stages called annotation, training and evaluation. Throughout this paper, data collection and pre-processing are represented in the annotation stage, which is concerned mostly by annotation of objects in images and by incorporating existing training and test datasets. Furthermore, we represent the algorithm specific parts of the design process, data representation and model creation inside the training stage. The decision making is situated in the evaluation stage and applies the trained model on test data to evaluate classifier performance.

2.1 Datasets

To create classifiers we need to represent the object that we want to classify. This is done by creating datasets made of positive and negative examples. The task of creating a dataset consists of gathering imagery and annotation about the desired object and marking the corresponding image region. The datasets created need to incorporate as much of the possible variation in appearance of an object or concept. Covering all possible variations of an object class, however, is not always possible and as such, can lead to extremely large datasets. The appearance of an object for example, a face, can vary due to intrinsic factors like age and identity and extrinsic factors like orientation and lighting conditions, which can obviously vary quite markedly Covering all

possible variations of an object class can be a near impossible task and oftentimes leads to extensive datasets. Thousands or ten thousands of object instances are not unusual, For example *Schneiderman* [5] collected 2,000 frontal and 2,000 profile faces for developing a face detector, and used a total of 2,000 images of cars for a car detector. Even more images were used in the training of a rotation invariant face detector, which was based on a total of 75,000 face samples [6]. For the prominent case of face detection and face recognition many datasets are available and well annotated. In combination they offer a large number of different faces like *FERET* [7] or *Caltech Webfaces* [8] as well as a large diversity in appearance e.g. *Yale* [9], *PIE* [10] and *Multi PIE* [11]. Prominent examples for datasets containing more than one object categories are *PASCAL VOC* [1], *Caltech 101* [12] and *Caltech 256* [13], *ImageNet* [2] and *LabelMe* [14].

2.2 Annotation Tools

We focus on annotating information that refers to the semantic content of the image to create classifiers that in the future can perform this task automatically. The understanding of the quality of annotation can vary from a simple text label to a comprehensive semantic description of image contents. This range of quality is also what differentiates available annotation tools. The *ESP Game* [15] can be understood as an annotation tool for simple text labels describing image content. In the game two players need to find the same keyword for a specific image in order to get a reward. If a keyword for one image is mentioned more than a certain threshold it is accepted as an annotation. The online accessible annotation tool *LabelMe* [14] allows the annotation of objects in uploaded images by marking the shape and by assigning a text label. The *LHI* annotation tool [16] allows a comparably sophisticated annotation. It includes among others segmentation, sketch, hierarchical scene decomposition and semantic annotation. The annotation tool can create object templates based on annotated data, which can simplify further annotations of the same object class. Unfortunately this tool is not publicly available.

The automatic labeling environment *ALE* [17] uses segmentation techniques (*Graph Cut* [18]) to simplify the annotation of regions. It computes the labels for each image pixel, based on statistics and co-occurrence of sets of labels. The so far presented annotation tools facilitate a very fixed form of annotation, like *text label* [15] or *shape and text label* [14]. The video annotation tool *ViperGT* [19] allows users to create templates that contain all necessary properties of an object. This customization approach allows the adaption to a specific object annotation case.

2.3 Training and Evaluation

Pattern recognition systems in the field of Machine Learning consist of two primary stages [4]. Within the training phase, a model or classifier is trained from given data samples to mathematically group or divide regions in the connected feature space. During the classification stage, which is also referred to as testing and evaluation, an unseen pattern that has not been used in the training stage is mapped by the classifier into the divided feature space where a final output value is usually assigned, e.g. positive or negative example. Naturally, this appears as a mere complex task that often

requires expert knowledge from the application domain and involves critical decision making in the design of specific parts of the algorithms, in data representation as well as in the creation of the model inside the training component.

For the evaluation of the detection performance of a classifier, we use Receiver Operating Characteristic (ROC) curves. The ROC curve is usually defined by the relationship of specificity (false positive rate minus 1) and sensitivity (true positive rate). The analysis of those curves yields comparable results for two or more classifiers working on under different operating points (thresholds). The area under the ROC curve (AUC) provides a normalized overall-performance measure on the quality of the employed classification models. It is a measure to describe how well the classifier separates the two object classes. The higher the AUC value, the better the classifier. If the AUC value is equal to 1.0, the accuracy of the classifier applied on the test set is 100%. [20]

ROC analysis appeared in the context of signal detection theory in the field of psychophysics to describe how well a signal could be distinguished by a receiver from noise [21]. Since then it has grown to become the gold standard in medical data analysis [22] and weather forecasting [23] and is now used as a popular tool for analyzing and visualization of the performance of machine learning algorithms [24].

2.4 Are There Other Unified Approaches Yet?

To the best of our knowledge, there are only a few approaches to modeling the complete workflow from dataset creation in combination with annotation over the application of training and evaluation components from the area of machine learning in the field of image processing. *Schreiner et al.* [25] create a semi-automatic annotation tools in the context of driver assistance systems to annotate driving maneuvers. *Meudt et al.* [26] propose a system for multi-modal annotation that can be used to annotate video, audio streams as well as biophysical data supported by active learning environment to reduce annotation costs.

The processes of training and evaluation are directly related to the field of data mining. For instance, the WEKA [27] open source framework includes numerous state-of-the-art methods to analyze and process numerical comma separated data. The *Pattern Recognition Engineering* (PaREn) system [28] provides a holistic approach to evaluation and optimization. A major effort is the shift of the inherent complexity of machine learning methods from the potential academic and scientific end-user into the program itself in order to address the applicability to other programmers, what is achieved by automated application of different algorithms to a data mining problem yielding to adapted parameter sets.

During the last decade, sustainable software development has gained a lot of interest. *PIL-EYE* [29] is a system for visual surveillance that enables the user to build arbitrary processing chains on platform independent image processing modules. Despite the seemingly rich choice of available tools, most of the mentioned tools lack at least a dedicated component for the annotation or creation of datasets, and as such, considerable time and efforts needs to be invested to adapt the settings of a given application for the specific annotation task at hand. Therefore, we follow a more generic approach that enables us to handle more common object patterns that can be easily defined by an open modeling architecture.

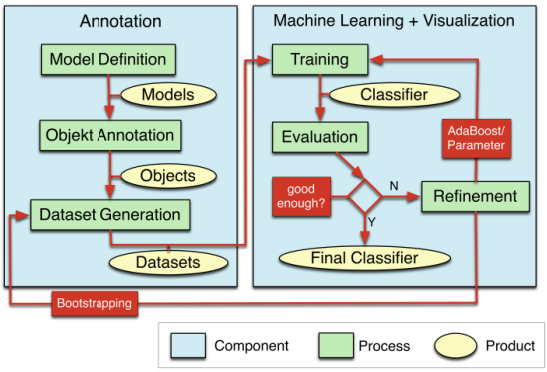

Fig. 1. Tool structure with its components, processes, intermediate products and links between processes that determine the possible sequences of the workflow

3 System Description

The proposed tool implements the design of a pattern recognition algorithm into two connected components: annotation and workflow. The annotation component includes customized annotation modeling, image annotation and dataset creation. The training and evaluation stages are integrated into the workflow component. The integration of processes like *Bootstrapping* [30] and *AdaBoost* [31] can lead to a non linear workflow. *Fig. 1* illustrates the system architecture.

3.1 Annotation Component

Following the example of *ViperGT* [19] our tools allows for the creation of annotation schema called models. They constitute a template for a certain object that needs to be annotated and can consist of an amount of the following elements: Point - coordinate (e.g. center of eye); Bounding Box - a rectangle; Ellipse - an ellipsoid; Polygon - a number of coordinates enclosing an image region; Text - a textual annotation; and Choice - a number of predefined text options. Since our annotation model is customizable, common annotation models e.g. consisting of a polygon and a text label can also be created. Customization is especially advantageous if more than one locatable or several textual labels per object need to be annotated (e.g. body parts).

Additionally annotation guidelines can be added to the annotation model and its contained elements to help unify the annotation process.

Images are annotated by first importing them into the tool and then associating them with a specific model. Thereafter, the windows *Annotation Scheduler* and *Annotation Visualizer* (see *Fig. 2*) guide the annotation process. The scheduler allows the navigation through available annotation tasks - so called 'jobs'. A job is a single instance of a predefined model and refers to the annotation of a single object. The values of the model elements are displayed in the scheduler and can be modified. The visualizer displays the locatable annotations (point, bounding box, and so on) of the current selected job and

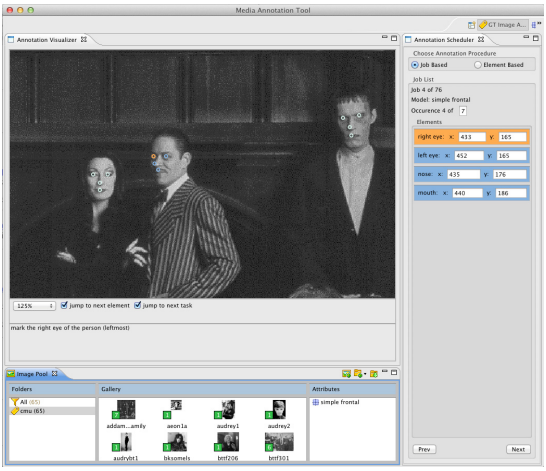

Fig. 2. Annotation of several faces with a simple model using the *Scheduler* (right) and the *Visualizer* (top left)

others sharing the same model. In order to accelerate annotation the list of elements can be traversed automatically if an annotation is finished. Additionally the annotator can choose to annotate either all jobs subsequently or focus on the annotation of a single element in all jobs. To generate training and testing datasets, annotations can be stored in XML-format including or excluding images.

3.2 Machine Learning Workflow and Visualization

A workflow is made up by a number of separate processes that are sequentially executed and form a processing chain. In our workflow and visualization component we visualize all processes of the processing chain separately. This enables us to manipulate process parameters, visualize and store intermediate results and allow manual execution of processes. Making computations across a large number of images, as is common in the field of pattern recognition usually takes a lot of time. The visualization of intermediate results can help verify if parameter settings and also if underlying code is correct and thereby presumably avoid rerunning time-consuming computations. We allow the storage of the processing chain with its associated parameters and already computed intermediate results. This infrastructure makes it easy to pickup past computations without retraining and reconfiguring. The decomposition of a training algorithm into separate processes can lead to a high re-usability. Common algorithms encapsulated in such processes can be integrated into different processing chains.

Our tool visualizes the processing chain (see *Figure 3*) in form of a stack of square boxes that signify all involved processes. The boxes are labeled with the name of the process and can contain input fields for its parameters and a set of buttons to plot or preview intermediate results or to start associated computations. For convenience, the processing chain can either be composed using the GUI or by manipulating an XML configuration file. Evaluation results in like ROC curves can be visualized with the plot function.

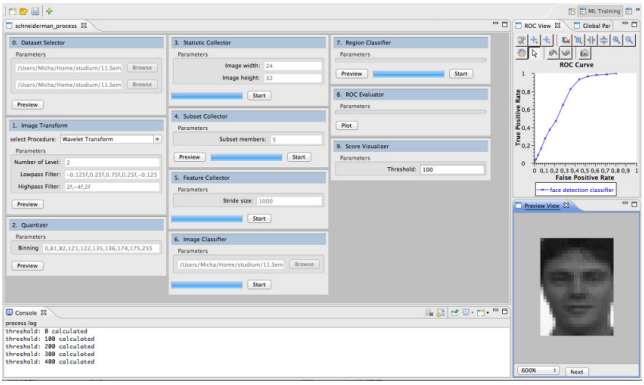

Fig. 3. Visualization of a processing chain: The GUI enables the composition of the processing chain (middle), the display of intermediate results (bottom right) and plotting of ROC curves (top right)

4 Proof of Concept at the Example of Face Detection

As a proof of concept we trained a face detection classifier using a simplified version of *Schneidermans* well known face detection method [32]. Annotation, creation of datasets, training and evaluation was done using the proposed tool.

The training process requires aligned faces. We created a simple face model containing coordinates for eyes, nose and mouth. Later on the alignment can be performed on these facial features. To create a training dataset we took faces from the *FERET* dataset [7] and created synthetic variations to increase the number of training samples. The first training dataset consisted of 50,000 positive and 50,000 negative examples. The *CMU test dataset* [33] was used for the evaluation; the images were re-annotated for that purpose.

The strategy of *Schneidermans* method [32] is to find face regions that have a high covariance and therefore are highly dependent on each other. To reduce dimensionality and to cover different resolutions and orientations a wavelet transform and a quantization are performed. Afterwards co-occurrence statistics for all wavelet coefficient pairs in all training data are gathered. The statistics are used to create subsets of coefficients that resemble highly dependent face regions. The values of wavelet coefficients in subsets form a feature value. Again these subsets are applied to the training data and all occurring feature values are gathered. To apply feature values in classification the feature values of an image region (e.g. from test dataset) need to be computed. The retrieved values can be compared with the occurrence inside the test dataset. If a specific feature value occurred more often in face samples than in non face samples the region is more likely to be a face. The classification score is computed by evaluating several feature values equal to the number of subsets.

For the training and evaluation process, we implemented the described method by constituting a processing chain of nine separate processes (see *Fig. 3*). For the optimization of the classifier, we used the bootstrapping strategy which adds false positives to the training dataset of the next iteration. *Fig. 4* shows the ROC curves for all three training iterations. Bootstrapping led to a significant increase in performance.

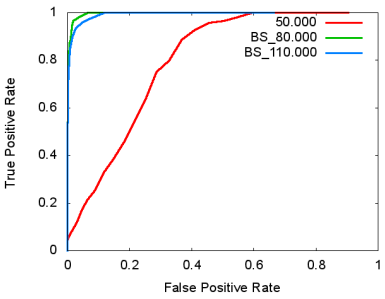

Fig. 4. Comparison of classifier performance of first iteration (50,000 training samples per class - red) with two bootstrapping iterations (additional 30,000 training samples per iteration - green and blue). The bootstrapped classifier with 80,000 has the best AUC value.

5 Summary and Future Work

We presented a generic approach to an all-in-one solution that covers the different development stages from dataset creation and model-driven annotation over training to evaluation and optimization in order to create more reliable classifiers for object detection. The proof of concept was demonstrated at the example of creating a face detection classifier. Future work will focus on predefined models of pedestrians, cars and other frequently emerging objects to provide some adaptability to customized data sets. Two final avenues we will explore in future are: an application to video annotation and analysis, and the incorporation of audiovisual models and features for speech and noise analysis.

Acknowledgments. This work was partially accomplished within the project ValidAX – Validation of the AMOPA and XTRIEVAL framework (Project VIP0044), funded by the Federal Ministry of Education and Research (Bundesministerium für Wissenschaft und Forschung), Germany and the Research Training Group CrossWorlds – Connecting Virtual and Real Social Worlds (Project GRK1780), funded by the DFG (Deutsche Forschungsgesellschaft), Germany.

References

1. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes (VOC) Challenge. International Journal of Computer Vision 88(2), 303–338 (2010)
2. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: IEEE International Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
3. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM 38(11), 39–41 (1995)
4. Jain, A.K., Duin, R.P.W., Gregory, R.L. (eds.): The Oxford Companion to the Mind, 2nd edn., pp. 698–703. Oxford University Press, Oxford (2004)

5. Schneiderman, H.A.: Statistical method for 3D object detection applied to faces and cars. PhD Thesis, Carnegie Mellon University (2000)
6. Huang, C., Ai, H., Li, Y., Lao, S.: High-Performance Rotation Invariant Multiview Face Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(4), 671–686 (2007)
7. Phillips, P., Moon, H., Rizvi, S., Rauss, P.: The FERET evaluation methodology for face-recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(10), 1090–1104 (2000)
8. Angelova, A., Abu-Mostafam, Y., Perona, P.: Pruning training sets for learning of object categories. In: IEEE International Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, pp. 494–501 (2005)
9. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(6), 643–660 (2001)
10. Sim, T., Baker, S., Bsat, M.: The CMU Pose, Illumination, and Expression (PIE) database. In: Fifth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 46–51 (2002)
11. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-PIE. Journal Image and Vision Computing 28(5), 807–813 (2010)
12. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. Journal Computer Vision and Image Understanding 106(1), 59–70 (2007)
13. Griffin, G., Holub, A., Perona, P.: Caltech-256 Object Category Dataset. California Institute of Technology. Technical Report 7694 (2007)
14. Russell, B., Torralba, A., Murphy, K., Freeman, W.: LabelMe: a database and web-based tool for image annotation. International Journal of Computer Vision 77(1), 157–173 (2008)
15. Ahn, L., von, D.L.: Labeling images with a computer game. In: Proceedings of the 2004 Conference on Human Factors in Computing Systems, pp. 319–326 (2004)
16. Yao, B., Yang, X., Zhu, S.-C.: Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks. In: Yuille, A.L., Zhu, S.-C., Cremers, D., Wang, Y. (eds.) EMMCVPR 2007. LNCS, vol. 4679, pp. 169–183. Springer, Heidelberg (2007)
17. Ladicky, L., Russell, C., Kohli, P., Torr, P.H.S.: Graph cut based inference with co-occurrence statistics. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 239–253. Springer, Heidelberg (2010)
18. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(11), 1222–1239 (2001)
19. Doermann, D., Mihalcik, D.: Tools and Techniques for Video Performance Evaluation. In: Proc. 15th International Conference on Pattern Recognition, vol. 4, pp. 167–170 (2000)
20. Lachiche, N., Flach, P.A.: Improving Accuracy and Cost of Two-class and Multi-class Probabilistic Classifiers Using ROC Curves. In: 20th International Conference on Machine Learning, pp. 416–423 (2003)
21. Tanner, W.P.J.R., Swets, J.A., Welch, H.W.: A New Theory of Visual Detection. Defense Technical Information Center, Electronic Defense Group, University of Michigan. Technical Reports, p. 42 (1953)
22. Metz, C.E.: Receiver operating characteristic analysis: A tool for the quantitative evaluation of observer performance and imaging systems. Journal of the American College Radiology 3(6), 413–422 (2006)

23. World Meteorological Organization (Eds.): Manual on the Global Data Processing System, part II, Attachments II.7 and II.8. 2010, Updated in 2012. Switzerland, p. 193 (2012)
24. Provost, F.J., Fawcett, T.: Robust Classification for Imprecise Environments. Machine Learning 42(3), 203–231 (2001)
25. Schreiner, C., Zhang, H., Guerrero, C., Torkkola, K., Zhang, K.: A Semi-Automatic Data Annotation Tool for Driving Simulator Data Reduction. In: Driving Simulation Conference, North America, p. 9 (2007)
26. Meudt, S., Bigalke, L., Schwenker, F.: Atlas Annotation tool using partially supervised learning and multi-view co-learning in human-computer-interaction scenarios. In: 11th International Conference on Information Science, Signal Processing and their Applications, pp. 1309–1312 (2012)
27. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1), 10–18 (2009)
28. Shafait, F., Reif, M., Kofler, C., Breuel, T.: Pattern Recognition Engineering. In: Rapid-Miner Community Meeting and Conference, Dortmund, Germany (2010)
29. Chang, H.J., Yi, K.M., Yin, S., Kim, S.W., Baek, Y.M., Ahn, H.S., Choi, J.Y.: PIL-EYE: Integrated System for Sustainable Development of Intelligent Visual Surveillance Algorithms. In: IEEE Digital Image Computing: Techniques and Applications, pp. 231–236 (2011)
30. Sung, K.K., Poggio, T.: Example-based learning for view-based human face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(1), 39–51 (1998)
31. Schapire, R., Freund, Y.: A decision theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
32. Schneiderman, H.: Learning statistical structure for object detection. In: Petkov, N., Westenberg, M.A. (eds.) CAIP 2003. LNCS, vol. 2756, pp. 434–441. Springer, Heidelberg (2003)
33. Rowley, H., Baluja, S., Kanade, T.: Neural network-based face detection. In: International Conference on Computer Vision and Pattern Recognition, pp. 203–208 (1996)