Extending training dataset for face detector learning

Jiří Kubínek*

Department of Computer Graphics and Multimedia
Faculty of Information Technology
Brno University of Technology

Abstract

Training classifiers for object detection requires large sets of positive and negative samples and usually a large amount of computational time. Obtaining negative samples is not so difficult; it can be almost any pictures not containing the object of interest. Creating the positive training dataset is more challenging. Those pictures must contain the object we want to detect with corresponding annotation (information, where the object actually is). Especially creating the annotation is very time consuming.

This paper describes several ways of extending training dataset used for training face detectors. Several experiments were concluded to evaluate the improvement of resulting classifier trained on datasets extended with the proposed methods.

Web gallery Flickr provides many images suitable for our purposes, we tried to use such images with automatic annotation for training. We also focused on the possibility of using datasets from controlled environment. Finally, random transformations were applied on the annotated data to extend total samples count. We also explore the possibility of reducing training time by sampling the training set with Unique Importance Sampling.

Keywords: Detection, Face Detection, Local Binary Patterns, Viola and Jones, WaldBoost

1 Introduction

Detection of objects in images is an important task of computer vision which has many practical applications in everyday life. Examples of the applications are face detection for human-computer interfaces and for surveillance system and licence plate detection for traffic monitoring systems. Other applications are in robotics, military and machine vision.

Approaches based on scanning images with classifiers proved to be very effective in detecting various classes of objects including faces, cars and pedestrians. The most successful classifiers for real-time classification are variations of cascades of boosted classifiers. Such classifier structure was first proposed for face detection by Viola and Jones [12] in their frontal face detector. Viola and Jones combined simple and extremely fast image features called Haar-like features with AdaBoost algorithm and with cascade structure of the detector. This detector achieved precision of detection which can be used in practical application in real-time.

Although the cascades of boosted classifiers are efficient in detecting objects, they have still limitations. Probably the most significant limitations are the long training time and the need large well annotated training datasets. The requirement for large training datasets increases cost of detectors and the long training time reduces the possibility of tuning parameters of the training process. In this work, we focus on both of these issues. We propose and evaluate several methods to decrease the size of hand-annotated training data and we use unique importance sampling of the training dataset to reduce the training time for large datasets.

In our experiments, we use WaldBoost algorithm [10] to build ensemble classifiers based on Local Binary Pattern [7, 8, 14] image features. In this framework, we evaluate the individual proposed methods.

The document is structured as follows. First, WaldBoost algorithm is introduced in Section 2 and the Local Binary Patterns are presented in Section 3. Then, unique importance sampling is presented in Section 4 and the methods for extending training datasets are proposed in Section 5, in Section 6 and in Section 7. Experiments are described in Section 8 and their results are presented in Section 9. Finally, the paper is concluded in Section 10.

2 WaldBoost

In the original frontal face detector by Viola and Jones [12], a cascade of ensemble classifiers is used. The ensemble classifiers are created by AdaBoost algorithm [2] and their operating point are set to achieve very low false negative rate and moderate false positive rate (background class is negative). All samples classified as negative are classified negative by the whole detector, thus achieving very low overall false positive rate and also low average computational time. However, all information between the individual stages is lost and the lengths and operating points are not set optimally with respect to the speed of the classifier.

Inspired by the Wald's sequential probability ratio test,

*xkubin09@stud.fit.vutbr.cz

Input: $(x_1, y_1), \ldots, (x_n, y_n)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$,
desired false negative rate α and false positive rate β.
Initialize weights distribution $D_1(i) = \frac{1}{n}$
Set $A = \frac{(1-\beta)}{\alpha}$ and $B = \frac{\beta}{(1-\alpha)}$
For $t = 1, \ldots, T$:

- Choose h_t which minimizes $\sum_{i=1}^{n} D_t(i) \exp(-y_i h(x_i))$
- Estimate the likelihood ratio $R_t (\simeq$
- Find thresholds $\Theta_A^{(t)}$ and $\Theta_B^{(t)}$
- Throw away samples from the training set for which $H_t \geq \Theta_B^{(t)}$ or $H_t \leq \Theta_B A(t)$
- Sample new data into the training set:

Output: strong classifier H_T and thresholds $\Theta_A^{(t)}$ and $\Theta_B^{(t)}$

Figure 1: The WaldBoost algorithm. This particular modification is used in the experiments.

which minimizes the number of measurements needed to make an accurate enough decision, Šochman and Matas developed WaldBoost algorithm [10]. This algorithm performs a test after each weak hypothesis which terminates the decision process if the class of the sample is classified with enough confidence.

The WaldBoost algorithm is shown in Figure 1. It takes as an input a set of labeled samples $(x_1, y_1), \ldots, (x_n, y_n)$, where x_i are the samples and $y_i \in Y = \{-1, +1\}$ are the corresponding labels. Another input is the desired false negative rate α and the desired false positive rate β.

WaldBoost calls a given weak learner in a series of rounds $t = 1, \ldots, T$. In each iteration, the weak learner creates a weak hypothesis $h_t :\rightarrow \mathbb{R}$ minimizing

$$\sum_{i=1}^{n} D_t(i) \exp(-y_i h_t(x_i)) \tag{1}$$

where $D_t(i)$ is a distribution over the training samples. $D_t(i)$ can be interpreted as weights which expresses importance of each sample. The fact that the weak learner minimizes error (eq. 1) implies that it focuses more on samples with higher weight. The selected weak hypothesis h_t is then added to the strong classifier

$$H_t(x) = \sum_{l=1}^{t} (h_l(x)) \tag{2}$$

Next, two early termination thresholds $\Theta_A^{(t)}$ and $\Theta_B^{(t)}$ are chosen on a likelihood ratio

$$R_t(x) = \frac{p(H_t(x) \mid y = -1)}{p(H_t(x) \mid y = +1)} \tag{3}$$

such that

$$\Theta_A^{(t)} = \arg\max_j (R_t(j) \geq A) \quad \Theta_B^{(t)} = \arg\max_j (R_t(j) \leq B), \tag{4}$$

Given: $h_t, \Theta_A^{(t)}, \Theta_B^{(t)}$ (all for $t = 1, \ldots, T$).
Input: a classified object x.
For $t = 1, \ldots, T$:

- If $H_t(x) \geq \Theta_B^{(t)}$ classify x to the class +1 and terminate
- If $H_t(x) \leq \Theta_A^{(t)}$ classify x to the class -1 and terminate

end
If $H_T > 0$, classify x +1. Classify x as -1 otherwise.

Figure 2: The WaldBoost classification algorithm.

where A and B are

$$A = \frac{1-\beta}{\alpha}, \quad B = \frac{\beta}{1-\alpha} \tag{5}$$

The selected thresholds $\Theta_A^{(t)}$ and $\Theta_B^{(t)}$ are then used to throw away samples which are classified with enough confidence to minimize average computational complexity of the classifier while maintaining the required false rates specified by α and β. Samples for which $H_t(x_i) \geq \Theta_B^{(t)}$ or $H_t(x_i) \leq \Theta_a^{(t)}$ are thrown away.

In the later iterations of the WaldBoost algorithm, large fraction of the training samples can be already thrown away and only few samples may remain. Such situation would lead to poor estimates of the optimal weak hypotheses and also of the likelihood ratios R_t. In similar situation, bootstrapping is commonly used to sample more samples in areas where the probability densities have to be estimated very accurately. The same approach is used in WaldBoost. After the weak samples are thrown away, new samples, which pass all the previous thresholds, are sampled to maintain constant size of the training set.

When detecting object in images, the weak learner creates weak hypotheses each based on a single feature (e.g. LBP operator at certain position and scale). Classification of a single image patch then proceeds as shown in Figure 2. The classification algorithm gets as an input the ordered set of weak hypotheses h_t and the ordered sets of thresholds $\Theta_A^{(t)}$ and $\Theta_B^{(t)}$. The classification proceeds in a series of steps $t = 1, \ldots, T$. In each of the steps, h_t is evaluated and H_t is updated. Then, if $H_t(x) \geq \Theta_B^{(t)}$, the sample is classified as class $+1$. If $H_t(x) \leq \Theta_A^{(t)}$, the sample is classified as class -1. In both of the previous cases, the classification process is terminated, otherwise the classification continues with the next step $t + 1$.

3 Local Binary Patterns

The Local Binary Pattern (LBP) texture operator was first introduced as a complementary measure to the local image contrast to be used in texture recognition. The first incarnation of the operator [7] worked with the eight-neighbours of a pixel, using the value of the center pixel as

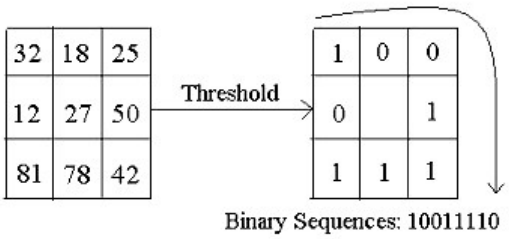

Figure 3: Illustration of Local Binary Patterns. A 3x3 local neighbourhood is thresholded by the middle value and resulting circular binary code is outputted. Taken from [5]

a threshold. An LBP code for a neighbourhood was produced by linearizing the thresholded values (Fig. 1). The LBP operator is invariant to monotonic changes in greyscale and possibly to rotations. The invariance to rotations can be achieved by merging appropriate code values. Rotation invariance can be further improved by distinguishing only uniform patterns [8] – patterns with at most two transitions between 0 and 1 in the corresponding binary code. The LBP operator was used in many practical applications mostly tightly connected to static texture analysis.

The LBP was successfully used for face detection by Zhang et al. [14], who proposed a variation of the original 3x3 LBP which they call Multi-Block LBP. They allow resizing of the neighbourhood and use sum of pixels in rectangular areas instead of sampling individual pixel values. This variation of LBP is used in our experiments. We use a set of LBP with all possible sizes and positions in the sample. The result of the LBP is directly the linearized binary code without rotational invariance. Rotational invariance is not needed, it would be possibly even counterproductive when detecting faces in image. This particular variation of LBP has 256 possible output values.

4 Training set sampling

With the information available on the Internet, it is possible to acquire training data sets (pools) of virtually unlimited size. However, due to computational limitations and training complexity of the boosting algorithms, it is very rare to process the entire dataset at once. Methods able to process large pools effectively are thus very important. One way to reduce training time is to sample a subset of training samples in each iteration of the boosting algorithm. The importance of each sample is determinated by its weight $D_t(i)$. At the start of training, the sample's weights are usually equal. According to [4], three sampling strategies are commonly used:

Trimming selects N samples with the highest weights. Weights of the selected samples are then normalized, so that their sum is equal to 1. N is set so that a predefined fraction of the total weight mass is used for training.

Unique uniform sampling (UUS) selects N unique samples with probability of selecting sample x_i equal to $P(i) = 1/n$ (n is the total number of samples). Weights of the selected samples are then normalized, so that their sum is equal to 1. This method is often used in practice for its simplicity, but the high probability of missing important samples is disadvantage of this strategy.

Importance sampling (IS) selects N samples with replacement from the pool. The probability of selecting samples is equal to their weight $D_t(i)$. Then, the weighs of the selected samples are normalized to $D_t(i) = 1/N$.

As shown in [4], the choice of the sampling significantly influences performance of the resulting classifiers especially when the number of selected samples N is relatively small. A good choice of the sampling algorithm may allow using only small fraction of the overall training set for selection the weak hypothesis in iterations of the boosting algorithm.

In the concluded experiments, we have used a variation of importance sampling which produces only unique samples. This sampling method, which we call *Unique Importance Sampling* (UIS), selects the samples exactly in the same way as IS does, but the selected samples are consequently aggregated, such that the sampled set contains only unique samples. The weights of the samples are set to

$$D_t(i) = \frac{count_i}{M}, \tag{6}$$

where $count_i$ is the number of repetitions of sample x_i in the sampled set and M is the total size of the sampled set. UIS is terminated when the number of unique samples reaches N.

5 Random Transformations

When detecting object by scanning images with a detection classifier, the distance between neighbouring scanned sub-windows in position and scale is not infinitely small. Because of that, the classifier has to be invariant to small changes in position and scale. Also the manual annotations of the objects of interest used for training are not perfectly aligned. These facts suggest, that it should be possible to apply small random geometrical transformations to the annotated training data without any loss of detection performance. On the other hand, such transformations could produce better approximation of the real probability density of the object class.

In our approach, for each sample from the annotated dataset, more samples are generated by applying small changes in scale and position. The possible transformations are restricted by a requirement for minimum overlap. To compute the relative overlap with the original annotation, radius and center of circle inscribed in both of the

image sub-windows are required. The overlap o is computed as [11]

$$o = \frac{r}{R}\left(1 - \frac{d_c}{r+R}\right), \tag{7}$$

where r is the radius of the smaller circle, R of the bigger circle and d_c is the distance between the centers of the two circles. The equation 7 is basically an approximation of the real overlap by linearly interpolating between two extreme cases. One of the extreme cases occurs, when the circle centers are equal. Then the overlap is approximated as r/R. The second extreme case occurs, when the circles have only one point in common (d=r+R). The overlap is 0 in this case.

In our approach, the random transformations are generated with uniform probability density in the space of translation and scale which is bounded by the minimum overlap o. The minimum overlap was set to 0.95 in our experiments.

6 Active Learning

Active learning is a semi-automatic annotation approach, which is often used to annotate large positive training data sets with affordable amount of human assistance. In this approach, a classifier is created on a limited sub-set of the all available training data. A human is then presented with samples for which the classifier is not confident enough. The final classifier is trained on the full dataset. However, such an approach could still be a problem for data sets containing millions of samples, which is the case of the face detection task. Our thought was to use fully automatic annotation based on a classifier which is trained on manually annotated data. Further images are then scanned across many positions and scales, getting classifier's response for each scanning window. Then, using these responses, annotation is created by non-maxima suppression without any user attention. Such approach will not directly improve the detection rates of the classifier. However, it should provide better approximation of the face probability density which could consequently lead to improved classifiers.

As mentioned above, we used already existing classifier to get responses over all scanning windows. In most cases, several detections appear around the objects of interest. To avoid multiple detections, non-maxima suppression was used to create single annotation and ignore other near detections.

To acquire new data, web gallery Flickr.com was chosen as it provides interface for image operations and the data at the gallery is organized in thematic groups. For our purposes, group "Portraiture" which contains 150,000 images was downloaded. Most of the images from this group contain single dominant face.

7 Controlled environment datasets

Currently, many annotated databases of faces are available. However, most of the face databases contain images which were acquired under controlled conditions. The head pose and facial expression is usually restricted and the images are taken in a indoor environment with controlled uniform lighting. Such data is very different to the images on which are the resulting face detectors tested in our case. However, adding this kind of data to the training data set can still be beneficial for the detectors.

We have used three frontal face databases to extend our manually annotated dataset. These databases are the BioID frontal face database[1], XM2VTS database[2] and Productive Aging Laboratory face database[3]. Annotated positions of eyes are available for all three of these databases. The left-top corner of the face annotation $\vec{c_1}$ and the right-bottom corer $\vec{c_2}$ were set as

$$\vec{c_1} = \frac{\vec{le}+\vec{re}}{2} + \begin{bmatrix} -1.25 \\ 0.45-1.25 \end{bmatrix} \|\vec{le}-\vec{re}\|, \tag{8}$$

$$c_2 = \frac{\vec{le}+\vec{re}}{2} + \begin{bmatrix} 1.25 \\ 0.45+1.25 \end{bmatrix} \|\vec{le}-\vec{re}\|, \tag{9}$$

where $\vec{le}$ respective $\vec{re}$ is the coordinate of the left eye respective of the right eye.

8 Experiments and Data

The original frontal face training dataset, which has been used for training face detectors at our organization contains images uploaded by user of a web-based face detection demonstration application which were then hand-annotated. This dataset is referred to as the "Schneiderman" data set in the following text. In all the images downloaded from Flickr, our classifier found more then 72 000 faces. The controlled environment databases contain total 4654 frontal faces. Negative samples were provided by images downloaded from web. For testing our classifiers, we used standard frontal face testing data set CMU+MIT[4] and also Matej Smid dataset which contains good quality photos of groups of people. The training data sets used in our experiments together with testing data sets are summarized in Table 1. For all our experiments, samples were resized to 24x24 pixels.

Our experiments with new data should confirm or decline, if extending training data set improves classifier's performance. We ran three experiments on data from Flickr, with different ratio between original and new data. For training, we used only new data, new data with small percentage of original data and new data with majority of original data. Results from these experiments were

[1] http://www.bioid.com/downloads/facedb/index.php
[2] http://www.ee.surrey.ac.uk/CVSSP/xm2vtsdb/
[3] https://pal.utdallas.edu/facedb/
[4] http://vasc.ri.cmu.edu/idb/html/face/frontal images

Dataset	#images	#objects
Flickr	150 684	72 670
Controlled environment data	4 654	4 654
Schneiderman	5 170	5 396
Matej Smid	89	1 618
CMU+MIT	113	491

Table 1: Data sets used in our experiments. Matej Smid and CMU+MIT are testing data sets.

Experiment	#positive	#negative	sampled
ExtendedBoth	200 000	200 000	1 500
ExtendedPositive	200 000	10 000	1 500
ExtendedNegative	10 000	200 000	1 500
Original	10 000	10 000	1 500
Sampled6000	200 000	200 000	6 000

Table 2: Summary of experiments evaluating random transformations and Unique Importance Sampling. The first column contains the name which is used in the text to refer to the particular experiment. #positive respective #negative is the amount of positive respective negative samples bootstrapped in to the training set in the WaldBoost algorithm. 10 000 means that random transformations were not applied. The last column contains the size of the set sampled by Unique Importance Sampling.

matched against the original classifier on our testing data sets. The same experiments were run with controlled environment data.

Another part of our work was dedicated to experiments with extending positive data set by random transformations, negative data set and both of them. We also experiment with the unique importance sampling. Results from these experiments were again matched against original classifier on our testing data sets. Informations about these experiments are summarized in Table 2.

9 Results

We use two ways of visualizing performance of the classifiers. Receiver operating characteristic, or simply ROC curve illustrates the relation between true positives rate (y-axis) and false positives (x-axis). Our goal is to get the highest possible true positive rate with as few false positives as possible. The best results are near the left-top corner.

The second type of graphs illustrates the relation between performance of the classifier (area above ROC, higher value means worse result) and its speed. As we want fast classifier with high performance, the best results are near the left-bottom corner.

According to Figure 4 and Figure 5, it appears to be more effective extending positive training dataset then the negative one. Furthermore, there is almost no difference

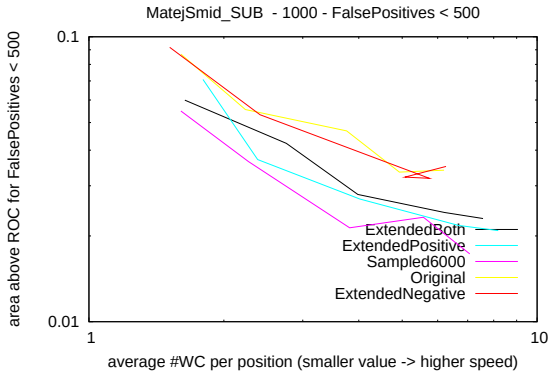

Figure 4: The relation between area above ROC curve and the speed of the classifier on Matej Smid data set is shown. The area above the ROC curve is integrated over interval [0,500] false positives. The best results are in the left-bottom corner.

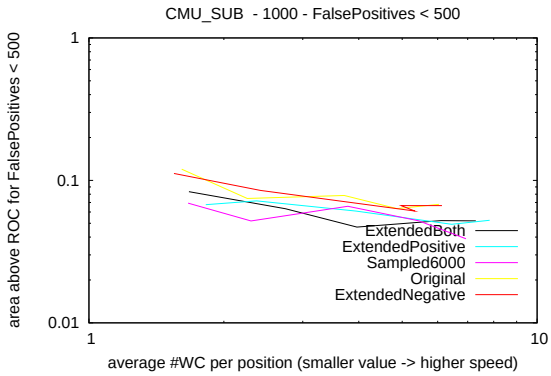

Figure 5: The relation between area above ROC curve and the speed of the classifier on MIT+CMU data set is shown. The area above the ROC curve is integrated over interval [0,500] false positives. The best results are in the left-bottom corner.

between the original classifier and the one trained on extended negative data set. According to our experiment, it is enough to train on negative data set containing 10 000 samples.

With the unique importance sampling, using just 1500 samples instead of 6000 decreases performance just a little. However, the selection of weak hypothesis, the most time consuming step in training procedure, is 4x faster.

Classifier trained on new data taken from Flickr provides nearly identical performance as our original classifier. As we can see in Figure 6 and Figure 7, no matter the ratio between original data and the new data, the little differences can be attributed to the random evaluation error. Acquiring new data from Flickr did not improve classifier's performance, one reason for that could be, that our random transformations do the same job.

Training on controlled environment data exposed significant decrease of performance. As can be seen on Figure

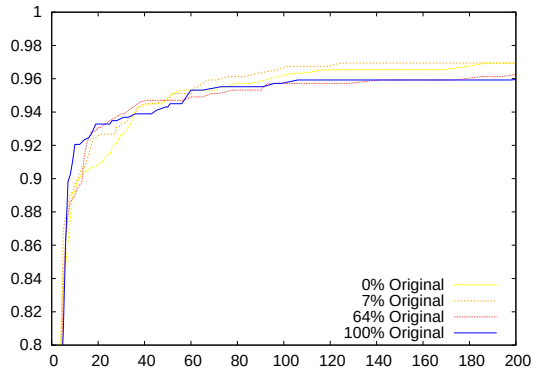

Figure 6: ROC curve comparing new classifiers trained on data from Flickr with the original one on the Matej Smid testing data set. The performance is not significantly different.

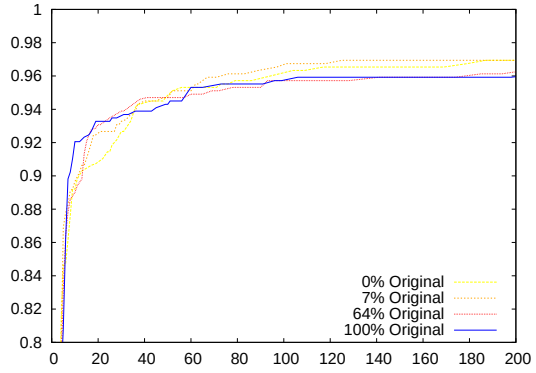

Figure 7: ROC curve comparing new classifiers trained on data from Flickr with the original one on the MIT+CMU testing data set. The performance is not significantly different.

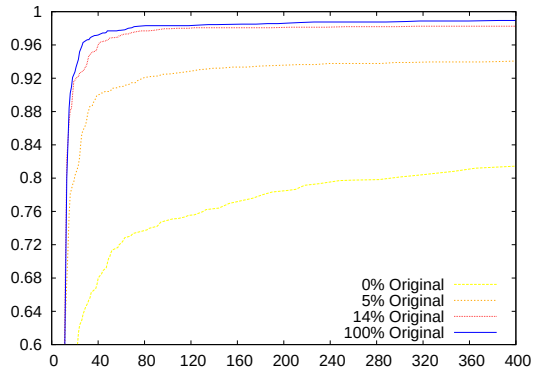

Figure 8: ROC curve comparing new classifiers trained on controlled environment data with the original one on the Matej Smid testing data set. With the lower percentage of original data, the performance is decreasing.

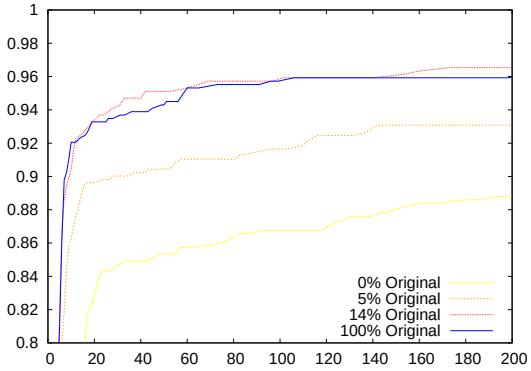

Figure 9: ROC curve comparing new classifiers trained on controlled environment data with the original one on the MIT+CMU testing data set. With the lower percentage of original data, the performance is decreasing.

detector / FP	6	10	21	46	50	65
Viola-Jones	–	0.76	0.88	–	0.91	0.92
Schneiderman	0.90	–	–	0.96	–	–
Lienhart et al.	–	0.82	–	–	0.90	–
Brubaker et al.	0.89	0.90	0.93	0.94	0.94	0.94
Our best	0.85	0.91	0.94	0.94	0.94	0.95

Table 3: Comparison of various frontal face detectors on the MIT+CMU dataset. For each detector, detection rates for multiple numbers of false detections (FP) are shown. The table contains result of Viola and Jones 2004 [13], Schneiderman 2004 [9], Lienhart et al. 2003[6] and Brubaker et al. 2008 [1]. The results were taken from [1].

8 and Figure 9, the results are worse then original with exception of the case with 14% of original data. In this case, the performance is nearly the same as original. The possible reason of this bad results is the different type of controlled environment data and the data in testing data sets. Another reason can be in little variability among the new data.

Table 3 shows the detection rates of our best classifier compared to other published results on the MIT+CMU dataset. Our results are very close to the best published results on this dataset. Moreover, the classifier needs less then 5 weak hypotheses per image position which is less than in any of the other approaches.

10 Conclusion and future work

In our work, we focused on extending training data sets for classifier training with hope of improving its performance. With data provided by Flickr, we achieve no improvement, but also no decrease in performance. It means, that these data annotated by our classifier are comparable with out manually annotated data.

Experiments made on controlled environment data show, that these data are not suitable for our training. Due

to minor variability among these data, the performance of resulting classifier decreases with lowering the percentage of original data. Another reason for this behavior can be caused by difference between controlled environment data and our testing data.

Basically, larger active training sets improve the results. Furthermore, extending positive training dataset with random transformations appears to be more effective then extending the negative set. Using just 1 500 samples instead of 6 000 with the unique importance sampling decreases the performance just a little with significant reduce of the training time.

The next possible step in our work includes acquiring training and testing data sets of different classes of objects and experiment with them to verify the results. We could also evaluate the effect of changing the threshold for automatic annotation with classifier's responses. Trying more samples with lower threshold or less samples, but with more confidence could also improve our results.

Acknowledgements

The experiments reported in this contribution were concluded using "Framework for Research on Detection Classifiers" [3].

The research on which this (report) is based acknowledges the use of the Extended Multimodal Face Database and associated documentation. Further details of this software can be found in; K. Messer, J. Matas, J. Kittler, J. Luettin and G. Maitre; "XM2VTSbd: The Extended M2VTS Database, Proceedings 2nd Conference on Audio and Video-base Biometric Personal Verification (AVBPA99)" Springer Verlag, New York, 1999. CVSSP URL: http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb

References

[1] S. Charles Brubaker, Jianxin Wu, Jie Sun, Matthew D. Mullin, and James M. Rehg. On the design of cascades of boosted ensembles for face detection. *Int. J. Comput. Vision*, 77(1-3):65–86, 2008.

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

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

[5] S. Liao, W. Fan, A.C.S. Chung, and D.Y. Yeung. Facial expression recognition using advanced local binary patterns, tsallis entropies and global appearance features. pages 665–668, 2006.

[6] Rainer Lienhart, Er Kuranov, and Vadim Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *In DAGM 25th Pattern Recognition Symposium*, pages 297–304, 2003.

[7] Timo Ojala and Matti Pietikäinen. Unsupervised texture segmentation using feature distributions. In *ICIAP '97: Proceedings of the 9th International Conference on Image Analysis and Processing-Volume I*, pages 311–318, London, UK, 1997. Springer-Verlag.

[8] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part I*, pages 404–420, London, UK, 2000. Springer-Verlag.

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

[10] Jan Šochman and Jiří 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 Šochman and Jiří Matas. Learning a fast emulator of a binary decision process. In Yasushi Yagi, Sing Bing Kang, In So Kweon, and Hongbin Zha, editors, *ACCV*, volume II of *LNSC*, pages 236–245, Berlin Heidelberg, 2007. Springer.

[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] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, 2004.

[14] 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.