object detection, machine learning,
biometrics, adaboost classifier,
high-resolution images

Jerzy DEMBSKI[1]

MULTISCALED HYBRID FEATURES GENERATION FOR ADABOOST OBJECT DETECTION

This work presents the multiscaled version of modified census features in graphical objects detection with AdaBoost cascade training algorithm. Several experiments with face detector training process demonstrate better performance of such features over ordinal census and Haar-like approaches. The possibilities to join multiscaled census and Haar features in single hybrid cascade of strong classifiers are also elaborated and tested. The high resolution example images were used in detector training process.

1. INTRODUCTION

The object detection task in graphical images is very important in many domains from video surveillance or biometric identification systems to any medical diagnostic tools. During recent 10 years since the famous AdaBoost [9] classifier learning method has been applied to face detection task [12], the machine learning methods have become very popular because of its universality to apply in any object detection tasks, not only face detection. The reason is due to using only sets of training examples to build classifier instead of specific domain knowledge. It means that any improvements and results obtained in face detection task are mostly useful in other object detection tasks.

Apart many variants of AdaBoost algorithm like RealBoost [6], FloatBoost [8], WaldBoost [10] etc., the feature type choosing plays most important role in detector performance improvement. The Haar-like pixel intensities differences in rectangular areas are the first feature type used with AdaBoost training algorithm by Viola and Jones [12]. In last 10 years some other feature types were adapted for AdaBoost object detector training tasks like Modified Census [7], [14] or more general local binary patterns (LBP) [11], [15], [1], [5], locally assembled binary (LAB) [13] and histograms of oriented gradients (HOG) features [2]. There are three main components of feature types rating: detection quality calculated by generalization error (ROC curve), detection speed and detector training speed. The generalization error can by improved by increasing the number of training examples or by increasing a quality of sample images, for instance by using images with high resolution which allows to take into account any important object details not accessible in standard resolution.

[1]Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, ul. Narutowicza 11/12, 80-952 Gdansk, Poland, e-mail: dembski@ue.eti.pg.gda.pl

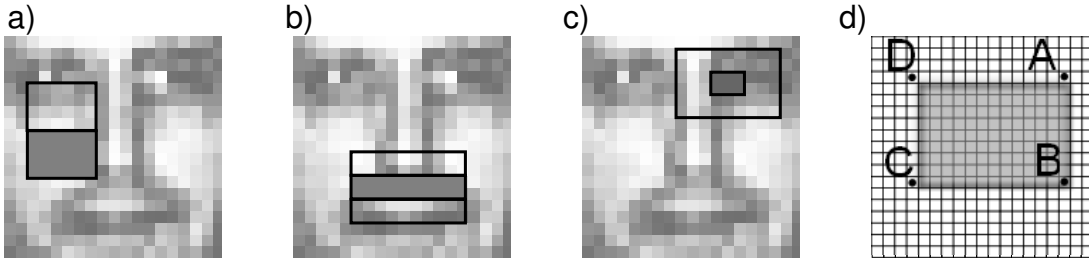

Fig. 1. Features based on intensity differences: a) edge, b) line, c) center-surround, d) calculation of the sum of pixel intensities using integral image.

In the case of Haar-like features the main problem with high resolution images is due to huge number of features which must be checked during the learning process. In [3] three methods of Haar features reduction were described. One method is specific for Haar-like features because it uses rectangular overlapping area similarity measure. Two other methods are universal for any features.

Due to high resolution the special multiscaled census features were implemented in this work and compared with Haar-like features with two methods of feature reduction in AdaBoost detector learning process using 32x32 face images as learning examples.

The second contribution of this work is comparing hybrid feature learning to learning only with Haar-like and only with multiscaled census features. The main questions is how to use multiple feature types in one classifier learning process and if it provides better performance because it can supplement each other is AdaBoost strong classifier assembly.

2. FEATURES

The Haar-like features presented in Fig. 1 are generated by moving and scaling rectangular area inside sample image template. The feature value is calculated as mean pixel intensity differences between dark and white regions in feature area. This value depends on mean and variance of pixel intensities on an image, so each sample image and each image under detection must be earlier normalized. The calculation of mean of pixel intensities can be strongly accelerated by using an integral image which must be prepared for each image before training or detection. Thanks to that only 3 arithmetic operations is enough to calculate sum of pixel intensities: $B+D-A-C$. In the case of high resolution sample images the number of features grows very rapidly with power of 4 and for instance 19x19 images allows to generate 34200 edge features, 32x32 - 270336 and 64x64 - 4259840 so it needs reduction.

The second type of features uses modify census transform [7] as a special case of local binary patterns (LBP) features. In standard version the number of census features is equal to the number of sample image pixels, so the growth with image resolution is not so rapid as in the case of Haar-like features. Fig. 2 presents the procedure of feature value calculation. Firstly the 3x3 neighbourhood matrix is generated around the chosen pixel. Next, each pixel intensity is compared with mean pixel intensity for neighbourhood matrix. If it is greater the binary value for this matrix pixel is set to 1 else to 0. In the final step the sequence of 9 binary values is changed into single integer number by multiplying binary values by subsequent powers of 2. There are another LBP pattern templates generalized to circles [11], [15]. The main advantage of census features is the independence of pixel intensities mean and variance so that image normalization is needless. The main disadvantage seems to be the great generalization error in the case of high resolution learning samples. The multiscaled version of census features is proposed to resolve this problem. This approach is similar to multiblock LBP [15] but the difference lies in the fact that in multiscaled approach the scale of 3x3 pattern matrix

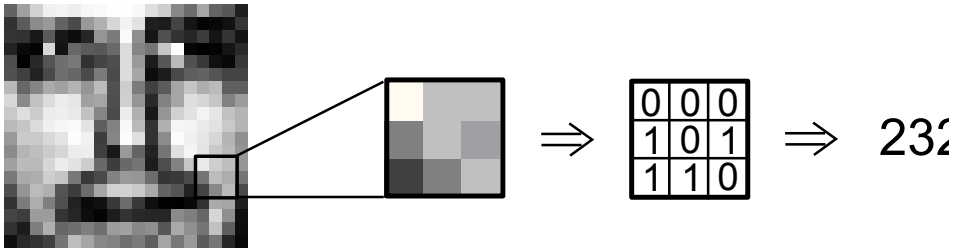

Fig. 2. The example of census feature value calculating.

can be changed smoothly by smooth change of resolution from 32x32 to 3x3. In this work decreasing resolution factor 1.25 is used due to recommended value for detection window scale changing [12]. In practice before training or detection for each image the sequence of decreasing resolution images and its census transforms must be prepared. In my experiments with 32x32 images 11 different sized versions of each sample image were prepared with size 32x32, 26x26, 20x20, 16x16, 13x13, 10x10, 8x8, 7x7, 5x5, 4x4 and 3x3. The whole number of features is equal to 2788. The computational complexity of proposed modification is proportional to the number of features which must be checked during training process. In 32x32 images training process this number is equall 1024 for ordinal census and 2788 for multiscaled census, so that the complexity increase of 172%.

3. THE ADABOOST TRAINING ALGORITHM

The AdaBoost strong classifier learning algorithm is presented in Tab. 1. The idea of boosting [9] is to improve overall strong classifier by adding subsequent weak classifiers which are as simple as possible. The vector of training examples weights helps to pay more attention to wrongly classified examples so far. At each step of while loop the best week classifier from great number of possible weak classifiers is chosen due to current weights. At the last step of while loop the best strong classifier threshold value Θ is calculated using validation or mixed set of examples. The strong classifier training process stops when the false positive error $f <= f_{\max}$ providing that true positive rate is over assumed value $d >= d_{\min}$. The strong classification function $H(\mathbf{x})$ consisted with T weak classifiers as a result of AdaBoost training process is described by the formula:

$$H(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) \geq \Theta \\ \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $\alpha_t = \log(1-\epsilon_t) - \log \epsilon_t$ is a weight of t-th weak classifier, $h_t(\mathbf{x})$ is t-th week classifier output value, $\mathbf{x}$ is an input image pixel intensities vector, Θ is a threshold.

Each possible weak classifier for object detection purpose usually is consisted with one feature and classification parameters, so the weak classifier expression can be enlarged to $h(\mathbf{x}, \mathbf{p}_f, \mathbf{p}_c)$, where $\mathbf{p}_f$ is a vector of feature parameters, $\mathbf{p}_c$ is a vector of classification parameters. The feature choice also determines weak classifier type. The binding of feature, weak classifier and parameters for both is shown in Tab. 2.

Haar-like feature needs to use threshold as classification parameter to determine the border between class 0 and class 1 and polarity parameter to determine the side of class 1 in feature value space. The census and modified census feature needs to inform which class is bind to each binary 3x3 pattern, so the vector of $2^9 = 512$ binary values constitutes classification parameters.

Table 1. Adaboost strong classifier (cascade layer) learning algorithm.

for each training example $\{(\mathbf{x}_1, c_1), (\mathbf{x}_2, c_2) \ldots (\mathbf{x}_K, c_K)\}$
initialize weights $w_i = 1/K_c$, where K_c is a number of examples which belong to the same class c as i-th example
end for
while false positive error $f > f_{\max}$ **or** true positive rate $d < d_{\min}$
1) normalize weights: $w_i \leftarrow \frac{w_i}{\sum_{j=1}^{K} w_j}$
2) select the weak classifier $h_t(\mathbf{x})$, which minimizes the weighted classification error: $\epsilon_t = \min_j \sum_{i=1}^{K} w_i |h_j(\mathbf{x}_i) - c_i|$
3) decrease the weights of properly classified examples: $w_i \leftarrow w_i \frac{\epsilon_t}{1-\epsilon_t}$
4) find the best threshold value Θ for strong classifier with minimum false positive error if false negative error is under expected value for node (cascade layer) $d >= d_{\min}$
end while

Table 2. Weak classifier descriptions.

feature type	classifier type	feature parameters $\mathbf{p}_f$	classification parameters $\mathbf{p}_c$
Haar-like	threshold-like	upper left and bottom down corners coordinates, rectangle orientation (only edge and line subtype)	decision threshold, polarity
census	tabular	neighbourhood center coordinates	the binary vector of classes for all possible patterns
multiscaled census	tabular	neighbourhood center coordinates, scale	the binary vector of class number for all possible patterns

The number of possible threshold weak classifiers with Haar-like features $L = 2N(K-1)$ depends on N–number of features and K–number of examples, which is equal to number of thresholds, because only thresholds between neighbouring examples in respect of current feature value are significant.

In the case of census features the number of possible weak classifiers is theoretically huge: $2^{512}N$ but it doesn't need to check each of them because it's enough to make training set statistics of object or non-object appearing for each of 512 possible patterns for considered feature. Feature selection is based on the error measure. Using symbols from [7], the error measure $\epsilon(\mathbf{x}) = \sum_\gamma \min\{g_t^0(\mathbf{x}, \gamma), g_t^1(\mathbf{x}, \gamma)\}$, where $\mathbf{x}$ - vector of feature parameters (neighbourhood center pixel position + scale parameter in multiscaled version), γ - pattern number (from 0 to 511), $g_t^c(\mathbf{x}, \gamma)$ - weighted sum of training examples from class $c \in \{0, 1\}$ with pattern γ for feature described by $\mathbf{x}$ at step t of strong classifier building process. The weak classifier for the best feature $\mathbf{x}_{\text{best}}$ consists of a vector with 512 binary values. Each value is equall to the class of a pattern which number is related to the element index. The class of pattern γ is evaluated using weighted sums: $c = 0$ if $g_t^0(\mathbf{x}_{\text{best}}, \gamma) < g_t^1(\mathbf{x}_{\text{best}}, \gamma)$ and $c = 1$ else. Due to statistics reliability requirements, the number of training examples must be fitted to the number of patterns. For instance, for 512 different patterns we need proportional number of training examples e.g. several thousands, otherwise there can be too many patterns with few representative examples or even without any examples, so the generalization error can be high.

3.1. THE CASCADE OF ADABOOST STRONG CLASSIFIERS

The main reason to use cascade version of AdaBoost classifier is to reduce detection time complexity but there are some another benefits like the possibility of supplement training set

with hard false positive examples during training process. In cascade classification process shown in Fig. 3 an input image is passed through a chain of strong classifiers. If an image is classified as an object in the node (cascade layer), it is passed to the next node, else is rejected. The positive classification needs the image to pass all nodes whereas only one node fail is enough to negative classification. This asymmetry is due to practical consideration that in typical picture the number of non-object windows is usually significantly higher than the number of object windows. Due to time complexity it seems to be better to reject as many false positive windows as possible at early layers and focus on object and object-like images by using more arithmetic operations in consecutive cascade layers. In practice the true positive rate $d_{\min}$ value is near 1.0, while false positive error $f_{\max}$ can't be too small because of the large number of weak classifiers which cause high time complexity during rejecting evident non-object windows. Typical $f_{\max}$ values oscillates between 0.2 and 0.5 and apart that it allows to obtain low overall false positive error which is a product of respective nodes error: $F = \prod_{k=1}^{K} f_k$.

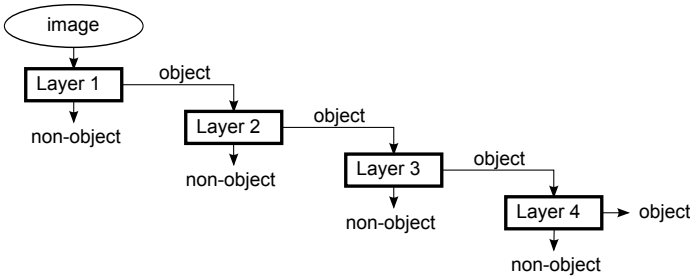

Fig. 3. The AdaBoost cascade object detection process.

The cascade training process schematic diagram is shown in Fig. 4. Bold arrows symbolize data flow, thin arrows - control flow. At the start feature libraries must be prepared for learning. Haar-like feature library for instance needs to prepare integral images for example images. This step is repeated after first and consecutive cascade layers because of new negative examples extracted from non-object photo images repository. The cascade layer construction module relies to strong classifier node learning algorithm described in Tab. 1. After node training process, the new layer is added to overall cascade. Next, the overall false positive error F is recalculated and checked if it is under target value F_{target}. If not, the loop have to be repeated.

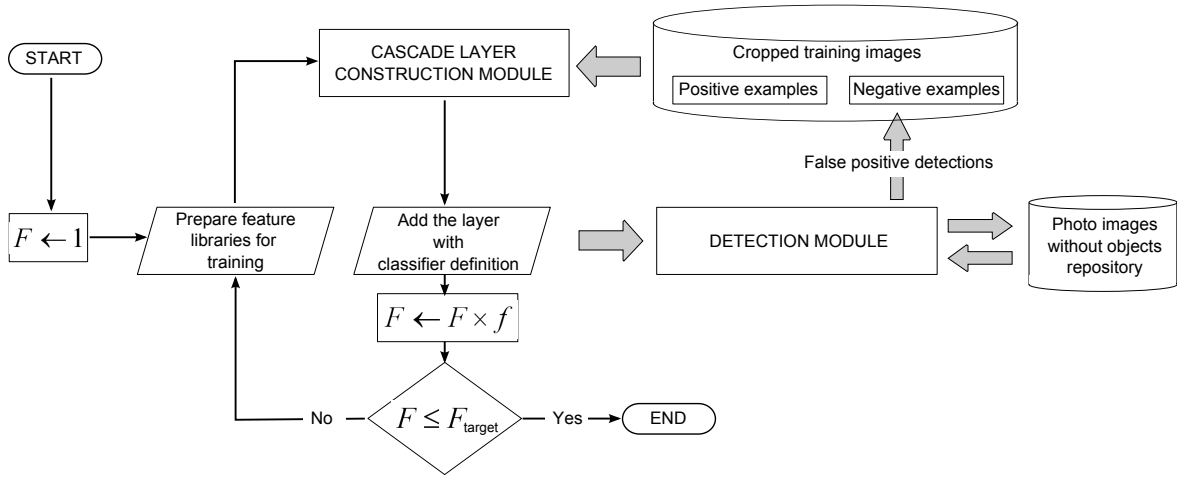

Fig. 4. The AdaBoost cascade training scheme.

While the set of positive examples (object images) is unchanged in cascade learning process, the negative subset should be supplemented after each subsequent layer training process. First reason is due to rejection of negative examples with rate $1 - f_k$ so that all properly classified

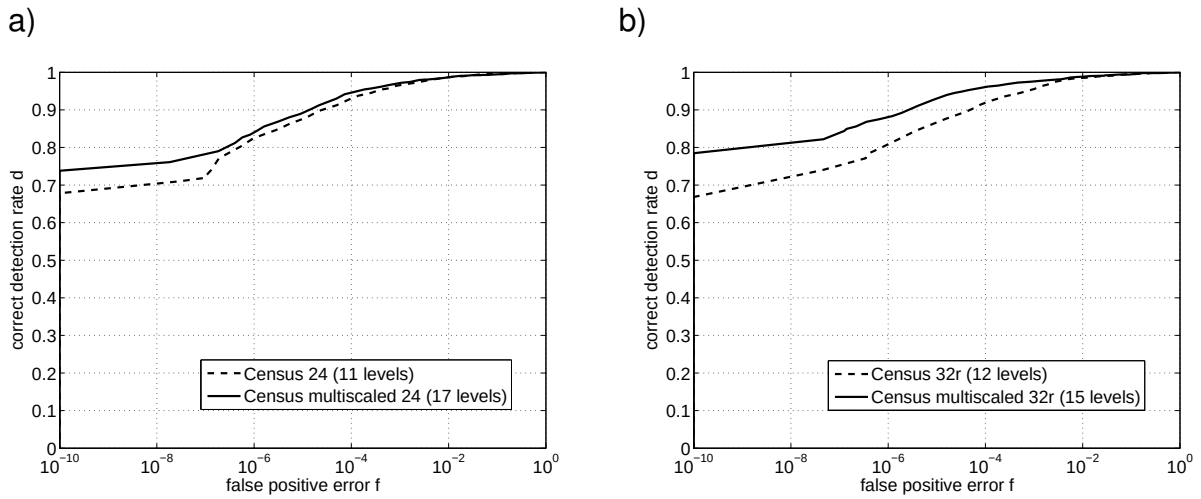

Fig. 5. ROC curves of AdaBoost classifiers trained with a set of a) 24x24, b) 32x32 images. Comparison of census and multiscaled census features.

non-object images are not valuable for higher levels training and should be rejected. Second reason is related to the need of providing proper number of non-trivial negatives for learning. The good idea described firstly in [12] is to use the huge repository of photographs shown at right side of Fig. 4 which do not contain any considered objects. The random set of negative examples is used at the start of learning but later negative examples are obtained as false positives by scanning photographs from repository using partially built classifier. Thanks to this negative examples are not trivial.

4. EXPERIMENTS AND RESULTS

For experimental purposes sets of different sized positive examples were generated using own face image acquisition system which requires only pointing eyes and a center of mouth. Negative examples for training are obtained from repository which contains over 1000 photographs from human environment like flats, offices, streets, forests but without human faces. A separate 100-image repository was prepared for classifier tests.

In the first sequence of experiments two classifiers are compared in respect of feature types. For each resolution of training images the multiscaled census feature type is better than simple version which is depicted by the ROC curves in Fig. 5. The differences grow with training images resolution which is probably due to that multiscaled census features can be more proof to single pixels noise in high resolution images. The experiments lasted several days till the classifier finds single false positives in the training photograph repository. The main parameters of learning $d_{\min} = 0.9995$ and $f_{\max} = 0.1$ are the same for each cascade layer. After overall learning process two parameters remain unfixed: an optimal cascade overall threshold multiplier and an optimal number of layers. First parameter is a ROC curve parameter and allows to trade off between high false positive and low false negative error and otherwise. The better approach is to find separate threshold value for each layer but complex optimization technique like simulated annealing or genetic algorithms [4] needs to be used. Second parameter is related to overfitting phenomena and is difficult to optimize in respect to ROC curve trade-off because clear case appears only when the whole curve dominates under another one. In unclear cases mean error in practical false positive error interval: $f >= 10^{-8}, f <= 10^{-5}$ is used as an optimality measure. These values depend on the scanning parameters. In this work the simple scanning method was used by checking all image windows shifted in vertical and horizontal

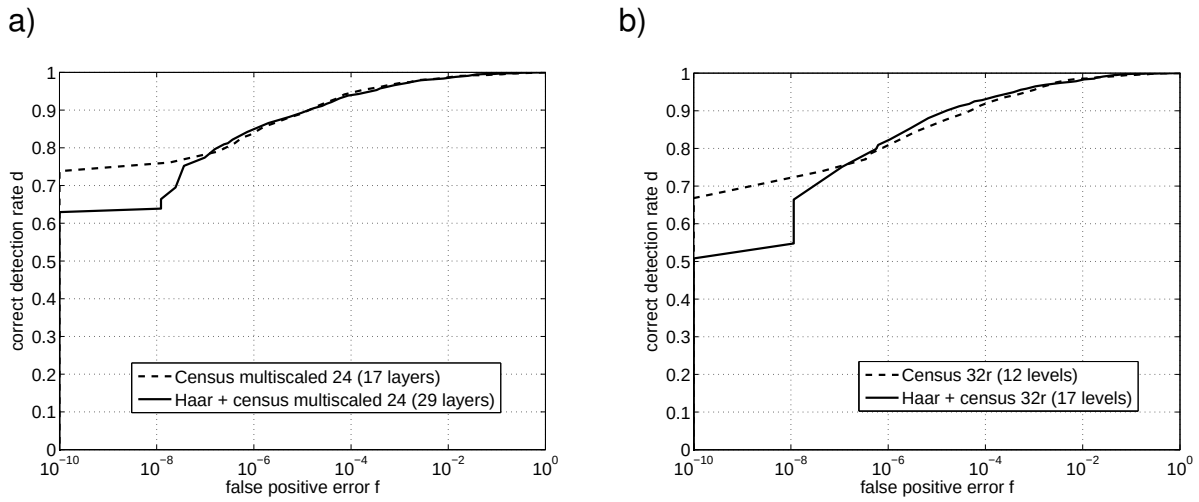

Fig. 6. ROC curves of AdaBoost classifiers trained a) with a set of 24x24 images using multiscaled census with or without Haar features, b) with a set of 32x32 images using ordinal census with or without Haar features.

directions by a step $size(\text{window})/size(\text{training image})$ rounded to less or equal integer number of pixels. The window size starts from training images size to short edge length of a scanning image with the magnification factor 1.25. The choice of threshold multiplier which also means false positive error choosing should depend on a scanned image size. HD image (1920x1080) for instance includes about 5 million windows, so the expected number of false detections is equal about $1/2$ for $f = 10^{-7}$ whereas 50 for $f = 10^{-5}$, so that to avoid false detections with good correct detection rate it's enough to choose $f = 10^{-7}$. For VGA (640x480) images with 650000 windows the proper false detection error seem to be equall $f = 10^{-6}$. False positive error is calculated during scanning test repository photographs as the number of positive classified windows divided by the number of all scanned windows.

In the second sequence of experiments the hybrid feature training is compared with single feature approach. The results for two cases of example image resolution are shown in Fig. 6. In practical false positive error interval: $f \in \langle 10^{-8}, 10^{-5} \rangle$ the correct positive rate in a part of this interval is better for hybrid approach. The feature and week classifier type choice in each week classifier learning step depends only on weighted classification error. Such little differences between charts emerge from general worse performance of single Haar features in comparision with census features which cause that very few Haar features were chosen in cascade building process. In experiment shown in Fig. 6a) only the overall cascade contained only 6 Haar-like week classifiers in comparision with 313 multiscaled census. In experiment shown in Fig. 6b) it was 11 Haar and 215 census week classifiers but in experiment with Haar and multiscaled census features with 32x32 learning images none of Haar-like classifier is chosen which indicate quite higher class separabilty of modified census compared with Haar-like and ordinal census.

5. CONCLUSIONS

Taking into account face detection experimental results the generalization is better in the case of using the multiscaled census features than ordinal census or Haar-like features. The experiments with hybrid approach in which each strong classifier in AdaBoost cascade can contain multiscaled census and Haar-like features shows a bit better results only in a part of practical interval of ROC curve. The reason is related to a small number of Haar-like based week classifiers in overall cascade. It can be explaned by its higher weighted error compared with multiscaled census based week classifiers.

In my opinion the lower generalization error then presented in article can be reached by using higher resolution training images and several training algorithm modifications. The most important topics listed below will be taken into account in future works:

- experiments with higher resolution training images,
- cascade layers thresholds value evaluation by genetic algorithm,
- comparing locally assembled binary (LAB) and histograms of oriented gradients (HOG) with Haar-like and census features in pure and hybrid approaches.

The preliminary experiments with higher resolution (64x64 and 96x96) training images show better generalization of cascade classifiers than with lower resolution images but plausible results needs more comprehensive experiments.

The strong classifiers threshold values Θ (see Eq. 1) are usually evaluated in greedy fashion after each layer week classifiers assembling process. The global optimization method like simulated annealing or genetic algorithm and validation data set will be used to improve generalization error.

The hybrid approach with more than two feature types will be elaborated and experimentally tested.

BIBLIOGRAPHY

[1] AHONEN T., HADID A., PIETIKINEN M. Face recognition with local binary patterns. In *In Proc. of 9th Euro15 We*. pp. 469–481.

[2] DALAL N., TRIGGS B. Histograms of oriented gradients for human detection. 2005. pp. 886–893.

[3] DEMBSKI J. Feature reduction using similarity measure in object detector learning with haar-like features. In *Image Processing and Communications Challenges*, R. S. Choraś, Ed., 2015, Vol. 7. pp. 47–54.

[4] GOLDBERG D. Genetic algorithms in search, optimization and machine learning. 1989. Addison-Wesley Inc.

[5] HADID A., MEMBER S. Face description with local binary patterns: Application to face recognition. 2006.

[6] HUANG C., MEMBER S., AI H., LI Y., LAO S. Fast rotation invariant multi-view face detection based. In *on Real AdaBoost, Proc. Sixth Intl Conf. Automatic Face and Gesture Recognition*, 2004. pp. 79–84.

[7] KÜBLBECK C., ERNST A. Face detection and tracking in video sequences using the modified census transformation. 2006, Vol. 24(6). pp. 564–572.

[8] LI S. Z., ZHANG Z. Floatboost learning and statistical face detection. 2004, Vol. 26. p. 2004.

[9] SCHAPIRE R. E., FREUND Y. Boosting the margin: A new explanation for the effectiveness of voting methods. 1998, Vol. 26(5). pp. 137–154.

[10] ŠOCHMAN J., MATAS J. Waldboost-learning for time constrained sequential detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, Vol. 2. pp. 150–156.

[11] TREFNY J., MATAS J. Extended set of local binary patterns for rapid object detection. 2010. pp. 37–43.

[12] VIOLA P., JONES M. Robust real-time face detection. 2004, Vol. 57(2). pp. 137–154.

[13] YAN S., SHAN S., CHEN X., GAO W. Locally assembled binary (lab) feature with feature-centric cascade for fast and accurate face detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008. pp. 1–7.

[14] ZABIH R., WOODFILL J. A non-parametric approach to visual correspondence. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1996.

[15] ZHANG L., CHU R., XIANG S., LIAO S., LI S. Face detection based on multi-block lbp representation. 2007.