Real-time human detection using multi-resolution HOGs

Martin D. Dimitrievski, Zoran A. Ivanovski, *Member, IEEE*

***Abstract* — The paper presents human detection algorithm that works on a general indoor/outdoor and urban/natural user generated video. The algorithm is based on a new multi-resolution histogram of oriented gradients feature for region description. The feature extraction is performed in constant time. Using cascaded approach for the classification of features low processing time is achieved. The final detector is robust to compression artifacts, different lighting conditions and white balance correction.**

***Keywords* — ADAboost, computer vision, cascade of rejectors, histogram of oriented gradients, human detection, integral image, linear SVM.**

I. Introduction

Localisation of upright standing human figures is of special interest in computer vision since it can be applied in many different aplications demanding the exact position and size of a person within a photo or a video sequence.

Two different approaches for human detection have been explored over the last years. The first type of methods represents a generative process where detected parts of the human body are combined according to a prior human model [1]. The second class of methods considers purely statistical analysis that combines a set of low-level features from a detection window to classify the window as containing a person or not, [2],[3] and [5]. The method presented in this paper belongs to the second category.

Using the images from the INRIA person dataset we construct a statistical model for classification of samples into human/other classes. We use only the frames extracted from a single video stream without any additional depth information as explained in [6]. The adopted method for coding the information within an image, first described as a concept in [4], is used in order to gain a huge speed up in performance from previous attempts in [5]. For the classification task we employ a cascade of strong classifiers similar to the one used in [3], which classifies the feature set in a layered approach based on the content of the analyzed region.

Finally we present a simple and fast yet effective way of handling multiple detections of a single object making the final results to look more natural for the observer.

The organization of the paper is as follows: the section II displays the preprocessing, scanning and feature extraction steps, section III gives an overview of the implemented classification technique and in IV we analyze the performance and accuracy of the algorithm.

Martin D. Dimitrievski, Faculty of Electrical Engineering and Information Technologies, Skopje, Karpos bb, 1000 Skopje, Macedonia (phone: 389-2-3099103, e-mail: liquid_mermer@yahoo.com) Zoran A. Ivanovski, Faculty of Electrical Engineering and Information Technologies, Skopje, Karpos bb, 1000 Skopje, Macedonia (phone: 389-2-3099103, e-mail: zoran.ivanovski@feit.ukim.edu.mk).

II. Feature extraction

Simple statistical analysis shows that humans in standing positions have distinguishing characteristics. Discriminatory light intensity information is found in the face/head regions as well as along the legs as highly active vertical regions. Thus, edges capture important cues for discriminating humans from the background. To capture these cues, the low-level features we employ are multi-resolution histograms of oriented gradients or MHOG. The work in [5] shows that simple histogram of oriented gradients (HOG) descriptors yield highest detection rates when computed over an exhaustively overlapping grid of rectangular blocks projecting the input image onto a highly dimensional vector space.

The scanning for potential matches of a human is done in a dense grid by moving the detection window by several pixels in every direction. We scan at 3 sizes of each video frame to detect persons at different distances from the camera. In Fig.1. we observe that during detection, the size of the detection window is constant. Rather than resizing the pattern of the human (detection window) we subsample the input pixels and perform each scan on a smaller frame which in turn produces the same result.

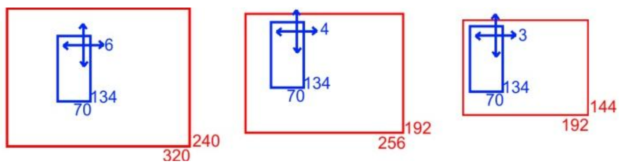


Fig. 1. The different sizes at which we perform scanning for posible matches.

For each of the 3 sizes we perform the same scheme for feature extraction. Detection windows are broken into many overlapping rectangular regions (cells) which are described by a single feature vector each. When extracting a feature vector from a cell we break the cell into four equal sub-blocks and perform the same procedure on each of the blocks separately. Each sub-block is coded using HOG into a 9-dimensional vector, then subsampled by a factor of 2 and coded again. The resulting histograms are concatenated into a single vector using L-1 normalization. This procedure equalizes the contrast of the cell:

$$f = [L1(f_1), L1(f_2)] = \left[\frac{f_1}{(|f_1|_1+\varepsilon)}, \frac{f_2}{(|f_2|_1+\varepsilon)}\right], \quad (1)$$

where f is the contrast normalized feature vector (length 72), f_1 and f_2 are the feature vectors for each of the two spatial resolutions, $|f|_1$ is the L1 norm of a vector and ε is a small constant to avoid division by zero. Thus 4 sub-blocks in 2 spatial resolutions produce a descriptor vector with 72 dimensions, which carries the information of the original region or cell.

Computation of each HOG is speeded up by using integral images similar to the method proposed in [4]. Firstly we compute the horizontal and vertical gradient of each pixel using convolution with masks [-1 0 1] and $[-1\ 0\ 1]^T$, respectively. Then, the L-1-norm magnitude and direction of the gradient is calculated. The direction is calculated via a look-up table which speeds up the computation time for the otherwise slow to compute transcendental inverse tangent function.

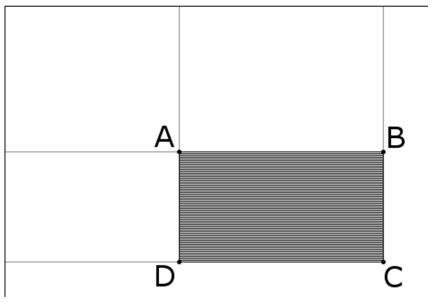


Fig. 2. A rectangular region or cell from which we extract a feature vector using integral images

The quantization of the orientations of gradients into orientation bins is performed by uniformly grouping the angles. Secondly, we make a weighted voting for each pixel's gradient into the correct orientation bin of the histogram. Voting is done proportional to the gradient's magnitude or L-1 norm of the gradient vector. These operations can be performed in constant time independently of the size of the analyzed cell. Only the computation of the necessary integral images is done in linear time, but these computations take up marginal time in the final detector when compared to the number of times a MHOG is computed. If we are analyzing a cell or region as the one enclosed by points A, B, C and D in Fig.2. one can easily calculate the area or the sum of values inside the rectangle by using a pre-calculated integral image of the region.

The sum of gradient magnitudes within a single orientation bin is computed by only 4 operations:

$$X_1(R) = IntIm(R_A) + IntIm(R_C) - IntIm(R_B) - IntIm(R_D), \tag{4}$$

where *IntIm* is the respective integral image and $R_{A,B,C,D}$ are the respective boundaries of region in the original image and the integral image. To compute an entire descriptor vector, for each of the 9 orientation bins, we need to compute 9 different integral images. Computation of these images is done in one pass knowing that the value of the current pixel depends on the values of the previously computed pixels:

$$IntIm(x,y) = a(x,y) + IntIm(x-1,y) + IntIm(x,y-1) - IntIm(x-1,y-1)\,, \tag{5}$$

where x and y are pixel locations and a is the original image. When computing the features for the sub sampled image region we cannot simply subsample the integral images. Therefore another 9 integral images are computed using sub sampled pixel values, accumulating to a total of 18 integral image computations.

III. Classification

We apply the already proven technique of layered or cascaded binary classifier first presented in [3] and later used in [2] and many more rapid object detection systems. The cascaded classifier consists of 30 stages, [3].

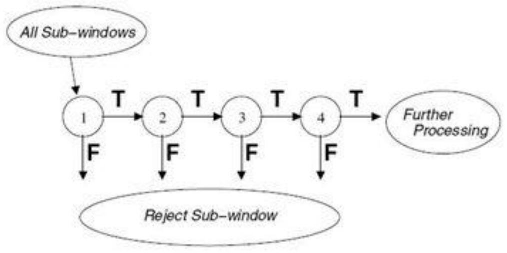


Fig. 3. An overview of the cascade classifier

Each stage is a strong ADAboost classifier trained on the positive samples from the INRIA person dataset and negative samples from a personal collection of images manually annotated of containing no human. The training sets are highly asymmetrical because of the different accuracy needed for the different classes. For any object detection problem a higher accuracy is needed when classifying negative samples in contrast to the true positive rate; hence, we use 2400 positive samples and 10^6 negative samples.

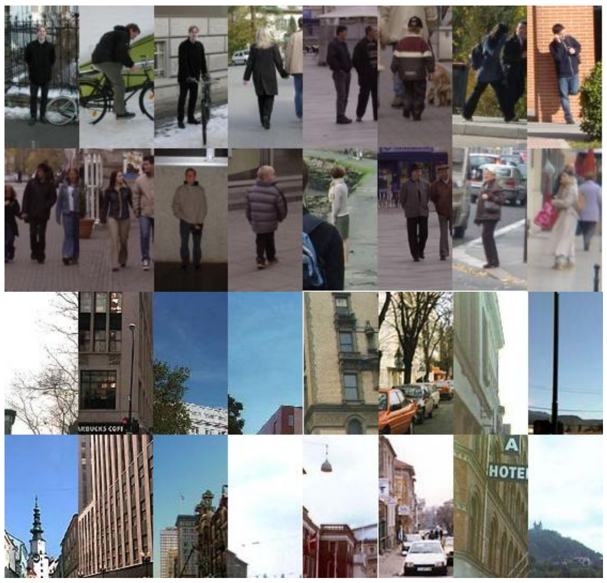


Fig. 4. First two rows – positive samples from the INRIA person dataset, last two rows – negative samples extracted from a personal image database

Fig.4. gives a better illustration of the training samples used in the training process. During the training of the cascade we create a set of 2306 overlapping cells with different positions, sizes and aspect ratios, Fig.5., from which each ADAboost classifier chooses the most informative ones to create a weak learner (linear SVM) based on the features extracted from each cell. In the process of training a single ADAboost classifier we train 500 weak learners each on a different cell chosen randomly from the original set.

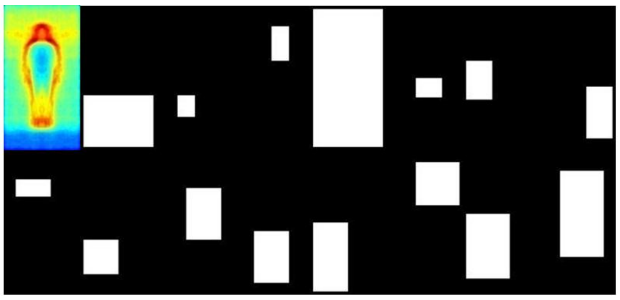


Fig. 5. Some of the possible cell locations and sizes. Top-left : the average gradient magnitude of a human computed over 2400 positive training samples.

The training of each weak learner is performed on a symmetrical set consisting of all of the positive samples and a subset of 2400 negative samples randomly chosen from the entire collection. For every next stage we perform testing on the entire training set with the current cascade to remove the correctly classified negative samples and continue to train the next strong classifier with the incorrectly classified or "difficult" samples. Each ADAboost stage chooses the best weak learners and weights their votes into a final strong discriminant function H(x):

$$H(x) = thresh\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right), \qquad (6)$$

where x is the training sample or descriptor vector of a cell, h is the weak learner and α is a weight factor. Weak learners are added to the final strong classifier until the predefined precisions are met. The strong classifier is adjusted to correctly classify at least 99.5% of the positive samples and at least 30% of the negative by setting a negative threshold value in (6). The entire training process is explained by this pseudo code:

Input: F_{target}: target overall false positive rate
f_{max}: maximum acceptable false positive rate per cascade level (0.3)
d_{min}: minimum acceptable detection per cascade level (0.995)
Pos: set of positive samples
Neg: set of negative samples

```
initialize: $i = 0, D_i = 1.0, F_i = 1.0$
loop $F_i > F_{target}$
    $i = i + 1$
    $f_i = 1.0$
    loop $f_i > f_{max}$
        1) train 500 (20% at random) linear SVMs using Pos and Neg samples
        2) add the best SVM into the strong classifier, update the weight in AdaBoost manner
        3) evaluate Pos and Neg by current strong classifier
        4) decrease threshold until $d_{min}$ holds
        5) compute $f_i$ under this threshold
    loop end
    $F_{i+1} = F_i * f_i$
    $D_{i+1} = D_i * d_{min}$
    Empty set Neg
    if $F_i > F_{target}$ then evaluate the current cascaded detector on the negative,i.e. non-human,images and add misclassified samples into set Neg.
loop end
Output: A 30-levels cascade
each level has a boosted classifier of SVMs
Final training accuracy: $F_i$ and $D_i$
```

After 30 strong ADAboost classifiers the expected true positives accuracy is 0.995^{30} or 86% and the expected false positive rate is 0.7^{30} or 0.002% which implies that most of the humans will be correctly classified and there will be almost no false positives at all. However, by setting such high precision rates, the later stages of the cascade need increasingly larger number of weak learners to achieve the local criterions for accuracy, which directly impacts the number of feature vector calculations and processing times.

IV. Results and discussion

As expected, the majority of negative samples are correctly classified only by the first 5 stages using minimal number of features. On Fig.6. we present the number of weak learners used by the first 15 stages of the cascade.

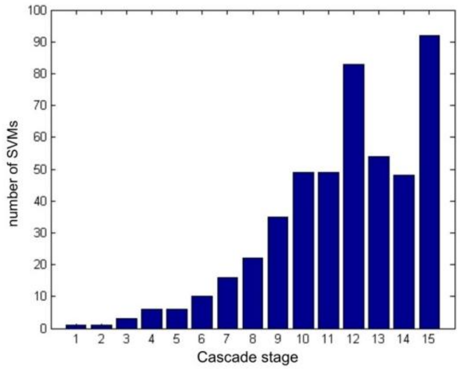


Fig. 6. - Number weak learners (SVMs) of used per cascade stage

We observe that if a more complex region has to be classified then more stages of cascade are needed to make a decision. Every single next stage needs exponentially increasing number of features and SVMs to be computed and classified. As the title of this paper implies, we focused on classification speed sacrificing accuracy and using only the first 15 stages of the cascade.

On Fig.7. we observe two ROC curves, the blue one for the theoretical model of the expected cascade classifier and the red for the actual trained 15 stage cascade. Because of the processing speed needed for the real-time implementation we can compare our accuracy only at 10^{-3} FPPW (false positives per detection window) to some of the current approaches. The cascade classifier in [3] shows 81% true positive rate at 10^{-3} FPPW, the approach in [5] has a higher accuracy of 95% same as the cascade in [2].

We improve the true positive rate at 10^{-3} FPPW to 96.1% using only 15 stages of the cascade classifier.

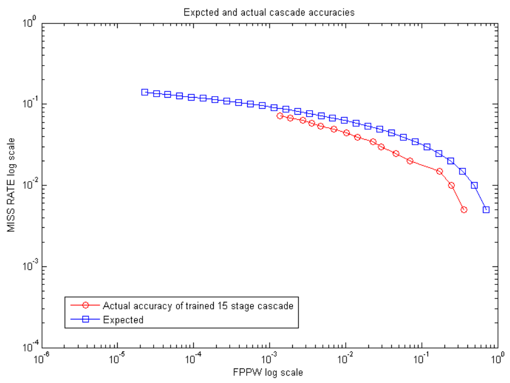
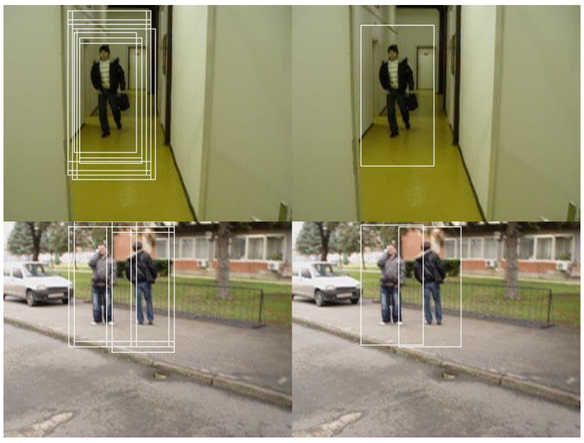


Fig. 7. – The expected and actual ROC curves for our detector

On an average user generated video the detector uses only 3.86 features and linear SVMs to classify a detection window as positive or negateive. Early estimates of the run times on a modern PC for a single threaded aplication with an optimized version of the detector suggest that it can run in real-time with more than 3fps for a QVGA video. All of the optimizations are towards a less computationaly complex algorithm and mathematical approach, and not using platform speciffic instructions. We use only simple arithmetic operations which can be easily implemented on embeded platforms or anywhere where the computing performance is limited.

In total more than 1500 detection windows are analyzed per frame. Fig.8 shows some of the results from the algorithm, on the left side are the frames with the original decisions. A clearly visible artifact of the dense scanning method is the multiple detection of a single object. This result is not natural for the observer and the raw data stream of positive detection windows locations and sizes can be misleading. To counter this we perform a very simple clustering of the data output from the cascade classifier. We use k-means clustering on the coordinates of the center pixel of each positive detection window as well as it's relative size to the frame.

On the right side of Fig. 8. are the detection windows clustered with k-means clustering algorithm. The clustering is performed with a low number of iterations beacause of the quick convergance of the few data instances and doesn't impact performance of the overall algorithm, but makes a clear difference to the observer.



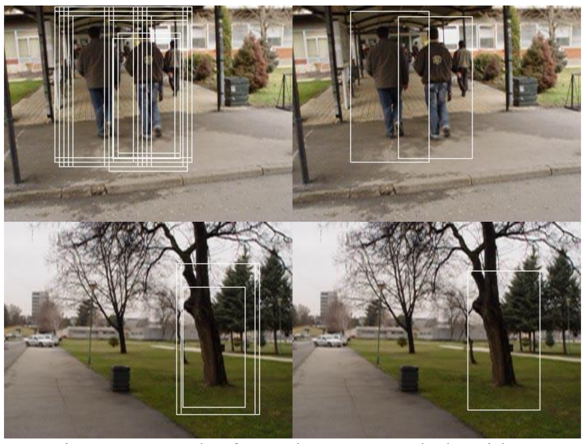


Fig. 8. – Results from the proposed algorithm

V. Conclusion

This paper presents a real-time human detection algorithm resistant to image degradation occuring from lossy video compression. The algorithm is based on multi-resolution HOGs for feature extraction uses a cascade of strong ADAboost classifiers for classification of samples.

The achieved accuracy at 10^{-3} FPPW is higher than the currently fastest human detectors and on pair with the algorithm proposed in [5]. The computational complexity of the proposed algorithm enables real-time aplication at 3 frames per second on a QVGA video.

References

[1] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *ECCV 2004*, volume I, pages 69–81, 2004.

[2] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients.In CVPR 2006, pages 1491–1498, 2006.

[3] Jones, P. V. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Vol. 1 (15 April 2001), pp. I-511-I-518.

[4] Porikli, F. (2005). Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1 (2005), pp. 829-836.

[5] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In CVPR 2005, 2005.

[6] W. Abd-Almageed, M. Hussein, M. Abdelkader, L. Davis. Real-time human detection and tracking from mobile vehicles. IEEE Intelligent Transportation Systems Conference ITSC 2007. pp. 149 - 154