Pedestrian detection on CAVIAR dataset using a movement feature space

Pablo Negri[1,2], Pablo Lotito[1,3]

[1] CONICET, Av. Rivadavia 1917, Capital Federal, Argentina
[2] Instituto de Tecnologia, UADE, Lima 717, Capital Federal, Argentina
[3] PLADEMA-UNCPBA, Campus Universitario, Tandil, Argentina

Abstract. This work develops a pedestrian detection system using a feature space based on level lines, called Movement Feature Space (MFS). Besides detecting the movement in the scene, this feature space defines the descriptors used by the classifiers to identify pedestrians. Locations hypotheses of pedestrian are performed by a cascade of boosted classifiers. The validation of these regions of interest is carried out by a Support Vector Machine classifier. Results rise to 81 % of good detection rate, having 0.6 false alarms per image on average on the FRONT VIEW CAVIAR dataset.

Keywords: Pedestrian detection, Level Lines, Movement Feature Space.

1 Introduction

CAVIAR dataset [1] is an indoor surveillance video sequence dataset used by the research community to test pedestrian detection methods [10], tracking [14,19,15], as well as detect actions or behaviors [3].

This work aims to detect pedestrians in the CAVIAR video sequences using a pattern recognition system. Our main contribution is the development of the Movement Feature Space (MFS) based on level lines [5]. Other authors used level lines to identify people in video sequences [2,18,9] but they employ them in a different way. In our approach, the MFS has two objectives. First, it generates a descriptor vector of the moving objects in the scene and then, it becomes the input of the detector for classifying between pedestrian and non-pedestrian classes.

Working with the MFS has interesting advantages. For instance, it adapts well to slow changes in the scene while being robust to rapid variations (i.e. illumination changes or weather conditions for outdoor applications). In these situations, the people appearance on the MFS does not change significantly compared to normal conditions and thus, they are easily detected by the classifiers.

The pedestrian detection algorithm has three steps. In the first one the movements on the scene are identified to build up the MFS. Then, a cascade of boosted

classifiers generates several hypotheses. Finally and using a Support Vector Machine (SVM) classifier, these hypotheses are thus confirmed.

The paper is structured as follows. Section 2 gives the guidelines to obtain the MFS in the video sequences. Sections 3 describe the input descriptors and the classifiers used in the pedestrian detection. Experimental results over the CAVIAR dataset are given in section 4. The performance of this approach has been contrasted with the OpenCV Adaboost train cascade library, which works on still images. Finally, we give our final remarks and future extensions in section 5.

2 Movement feature space based on level lines

2.1 Definition of level lines

Let I be a monochromatic image with $h \times w$ pixels, where $I(p)$ is the intensity value at pixel p whose coordinates are (x, y). The (upper) level set X_λ of I for the level λ is the set of pixels $\mathbf{p} \in I$, so that their intensity is greater than or equal to λ, $X_\lambda = \{\mathbf{p}/I(\mathbf{p}) \geq \lambda\}$.

For each λ, the associated level line is the boundary of the corresponding level set X_λ, see [5]. Finally, we consider a family of N level lines $\mathcal{C}$ of the image I obtained from a given set of N thresholds $\Lambda = \{\lambda_1, ..., \lambda_N\}$. From these level lines we compute two arrays S and O of order $h \times w$ defined as follows:

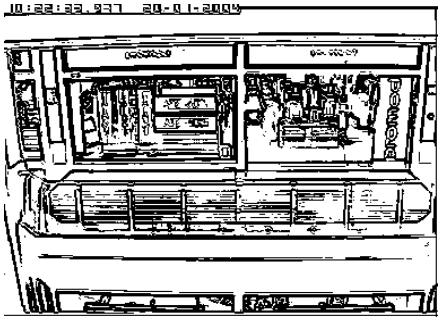
- $S(p)$ is the number of level lines C_λ superimposed at p. When considering all the gray levels, this quantity is highly correlated with the gradient module.
- $O(p)$ is the gradient orientation at p. In this paper, it is computed in the level set X_λ using a Sobel filter of 3×3 pixels, then, orientations are quantized in η values. Here, we do not make difference between a dark-bright transition and a bright-dark one in order to be robust to the high variability in human appearance. For each pixel p, we have a set of $S(p)$ orientations values, one for each level line passing over p. The value assigned to $O(p)$ is the most repeated orientation in the set.

Generally, in the practical implementation, only those pixels for which $S(\mathbf{p})$ is greater than a fixed threshold δ are considered, simplifying the analysis and preserving meaningful contours. In our system, best results where obtained with $N = 48$ and $\delta = 1$.

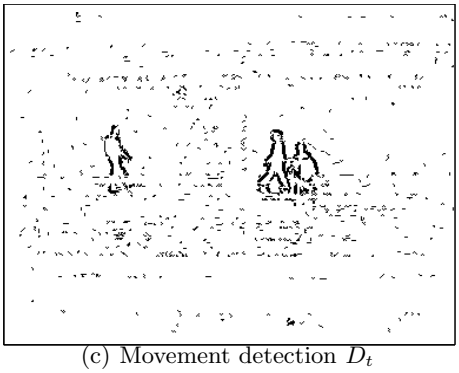
2.2 Movement detection

As described in [4], level lines have many good properties, for example the are invariant to contrast changes. It means that a regular contrast change (monotonic and upper semicontinuous) can either create or remove level lines from a pixel, changing the $S(p)$ quantity, but it could never create a new level line intersecting the original ones [2]. This is crucial because we will use level line intersections to detect movements. The last assertion means that our method will be robust to regular contrast variations.

(a) Frame at t



(b) Reference R_t



(c) Movement detection D_t

Fig. 1. Movement detection of the same sequence, showing the background model reference and the movement detection.

Now, let two consecutive images I_{t-1} and I_t obtained at times $t-1$ and t. When looking for scene changes at pixel p, the variation of $S(p)$ could correspond to a movement but also to a change in contrast. A more reliable indicator is a variation on $O(p)$, i.e., $O_{t-1}(p) \neq O_t(p)$. However, the number of points verifying

that condition between two consecutive images could be very few. Mouchafa and Aubert [11,2] showed that it is better to work with background reference. They defined an adaptative background reference model, composed of the set of pixel p which are stable over an horizon of time, together with the corresponding values S^R and O^R. More precisely, given a horizon of time T we define R_t as the set

$$R_t = \{p \in I_t : O_{t-1}(p) = O_{t-2}(p) = \cdots = O_{t-T}(p)\},$$

together with O_t^R whose value at p is the preserved orientation, i.e., $O_t^R(p) = O_{t-1}(p)$ for any $p \in R_t$. In practice, the equality constraints in the definition of the reference space R_t can be mollified to allow for small variations of orientation due to noise or other perturbations.

Thus, at time t, the set of pixels p that are not in the reference or have an orientation other than the reference: $O_t(p) \neq O_t^R(p)$, are more likely to correspond to moving objects. These pixels will make up the detected set D_t. Figure 1 shows an example of the reference model of the video sequence at time t. Detected set D_t is presented in fig. 1(c).

Below, we will focus the analysis only on pixels in the detected set D_t, and their values of S_t and O_t. This set can be considered as a virtual image with two associated scalar fields, or a kind of feature space referred to as *Movement Feature Space,* or MFS.

3 Pedestrian detector system

Features and descriptors extract information from the image. The choice of a good representation space (feature space), helps the classifier to discriminate whether a person is present inside a region of interest (RoI) or a pattern. Using a learning dataset, positive samples (patterns with a person inside) and negative samples (patterns without persons) are grouped into two classes by the classifier by a decision boundary.

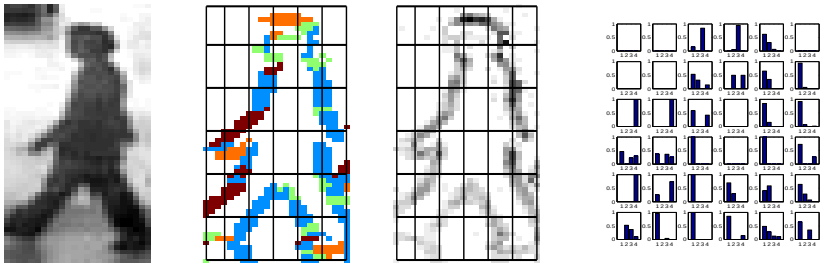
3.1 Feature space

The chosen pattern is a rectangle of 12x36 pixels size. The information inside the pattern generates a set of descriptors called feature space.

Two types of feature spaces are used by the classifiers. First ones are calculated from the MFS and the second ones are computed from the gray scale representation of the image.

Feature space from the MFS Two set of features computed on the MFS. First set is a HOG-like features, called HOG-MFS, calculated on rectangular patches inside the RoI using S_t and O_t information of the D_t pixels (the MFS space).

HOG-MFS results in a concatenated set of Histograms of Oriented Gradients (HOG) [7]. A RoI is subdivided into a grid of rectangular patches. Within each patch, we compute HOG descriptors as follow:



(a) Person (b) O_t grid (c) S_t grid (d) HOG-MFS features

Fig. 2. HOG-MFS calculation on defined grids. In (b), each color correspond to a one of the four directions of the gradient. In (c), darker pixels have highest S_t values.

- a histogram has $\eta = 8$ bins (one for each direction),
- the value for a bin is the sum of the $S(p)$ for the p with this orientation,
- the histogram values are normalized.

Figure 2 shows the 6x6 grid and the HOG-MFS calculated in each grid on figure 2(d). For the sake of simplicity, we consider four gradient directions, where the histogram bin 1 corresponds to the vertical direction, bin 3 corresponds to horizontal direction, and the other two are the diagonals directions.

The second set of features, called MAG-MFS computes the sum of the S_t values inside each patch. This feature helps with the fact that HOG descriptors can make difference between a strong edge in the patch, generating a one in the corresponding bin, and only one pixel of noise, which generates the same histogram after the normalization.

Haar-like filters Rectangular filters or Haar-like features provide information about the gray-level distribution of two adjacent regions in an image. These filters consist of two, three or four rectangles, as proposed by Viola [20]. To compute the output of a filter on a certain region of image, the sum of all pixels values in the gray region is subtracted from the sum of all pixels values in the white one (and normalized by a coefficient in case of a filter with three rectangles). The total number of Haar-like features calculated on the person pattern is 4893.

3.2 CAVIAR Dataset

The CAVIAR dataset is composed of two set of video sequences, one capturing the entrance lobby of the INRIA Labs at Grenoble, France, and the second were recorded in a shopping center at Lisbon. The CAVIAR dataset also provided the ground truth indicating the bounding box around each visible person. We choose the sequence of the view across the hallway of the shopping center, fig. 1(a), named FRONT VIEW, which is composed of more than 20

video clips. From this set, the following sequences were employed for the training: *TwoEnterShop1front*, *ThreePastShop1front*, *ThreePastShop2front*, *OneShopOneWait2front*, *ShopAssistant1front*. The other sequences were employed in the tests.

The choice of the training sequences was not random. The ground truth of these captures has additional information: gaze direction, hand, feet, and shoulder positions and, most important, the head. We need this information to change the bounding box for the training. The new bounding box is centered in the x axis using the position of the head. In fact, the CAVIAR bounding box covers completely a person. And, when it is walking, the head position inside the bounding box can be far from the box center. In our tests, this labeling prevents our classifier to generalize the pedestrian class and to discriminate from the non-pedestrian class.

This new ground truth give us 4773 labeled pedestrians, in near 8000 frames, to train the classifiers. The ground truth of the remaining sequences was left unchanged for the tests. The complete set of test sequences has 15134 labeled persons in more than 15000 captures. There are samples that where discarded because they were partially out of the view (inside the shop, or leaving the view), or if their size where smaller than the pattern ($width$=12, and $height$=36).

3.3 Pedestrian classification

Pedestrian classification means the automatic identification of the people present in the sequence frame. In practice, the system discriminates those regions or windows in the image having a person inside versus empty region (without a person). Previous sections show how to identify the movement on the scene and how to extract the information which can be used to recognize people. Now, it's time to use the MFS to feed a two-step detection algorithm which represents the system response: the pedestrian positions.

Hypothesis generation In the hypothesis generation, the whole image is analyzed using a sliding window approach [7] to identify pedestrians in these windows.

The detector is an Adaboost Cascade of 20 boosted classifiers [20] discriminating pedestrian and non-pedestrian hypothesis. The implemented methodology is analogous to the one presented in [13,12]. This classifier has a good performance, it evaluates about 49.000 RoIs in some milliseconds and deliver to the next detection step only most probable hypothesis.

The input descriptors of the cascade will be: MAG-MFS, HOG-MFS as a generative function (calculated using a pedestrian model, see [13]), HOG-MFS as a discriminant function, and Haar-like features. At each iteration of the learning process of a strong classifier, the learner chooses between one of those sets of features and their weak classification function associated.

The HOG-MFS and the MAG-MFS descriptors are computed on a dense grid of 3707 overlapped square and rectangular patches with different sizes on the pattern.

As was mention earlier, positive training dataset is composed of those pedestrian patches, in the new label. Negative training dataset, which train a specific stage of the cascade, is composed of the patches in the sequence validated by precedent stages, and not containing a pedestrian. In fact, patches which "touch" a pedestrian, i.e. the overlapping criteria (see section 4.1) is lower than 0.1, are employed as negatives. If we don't do that, the classifier only concentrates to discriminate motion objects from noise. Then, any patch on the image which have motion features (HOG-MFS and MAG-MFS) will be validated for the classifier.

It is important to note that in the learning process, initially the boosted classifiers of the cascade are chosen among HOG-MFS and MAG-MFS descriptors, which discriminate the movement in the scene. Then, at later steps, classifiers choose Haar-like features, they are highly discriminant and help to identify pedestrians from noise or others moving objects.

Hypothesis validation Once the Adaboost classifiers have finished their work, remaining hypothesis are those which the detector could not discriminate between pedestrian and non-pedestrian class (they can be considered as harder samples), as show fig. 4(a).

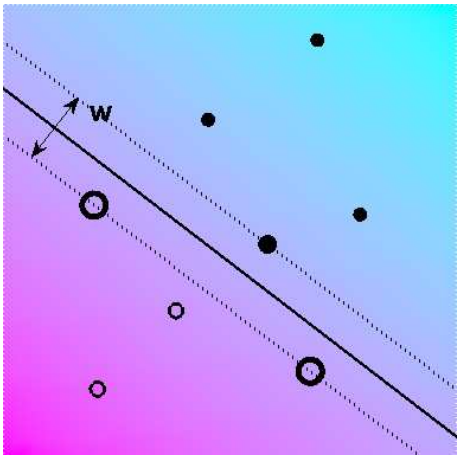


Fig. 3. Hyperplane for a linear separable problem. Dotted lines represent the margin defined by the three support vectors: two O's and one filled point.

The hypothesis validation is carried out by an hyperplane learning algorithm called Support Vector Machine (SVM) [17]. For linearly separable problems, among all the hyperplanes separating the training data on the feature space (person versus non-person classes), there will exists a unique optimal hyperplane which maximize the separation margin $\mathbf{w}$, as show fig. 3 [16]. This hyperplane is defined by the training samples called support vectors. Those samples are equally close to the hyperplane, also defining the dotted line in fig. 3. A test

sample is evaluated on the feature space using all the support vectors to found on which side of the hyperplane is projected (classified).

When the problem is not linearly separable (our problem), can be found a classifier which allow the possibility of miss-classified samples. To solve this new optimization problem, slack variables and new constrains are introduced, in order to found a soft margin classifier.

The SVM classifier has better discriminant properties than the Adaboost cascade, but it is very time consuming because each classification depend on the number of support vectors and the size of the feature space. To speed it up, we use a limited set of HOG-MFS. The RoI is divided in a set of 2x2, 4x4 and 6x6 non-overlapped patches (as show fig. 2(b)). In addition, we have calculated three more grids overlapping the others of 1 patch size (overlapping the grid of 2x2), 3x3 patches, and 5x5 patches. This set of 91 HOG-MFS features is the input of the SVM classifier. Positive samples are the labeled pedestrians of the training datasets, and negative samples are those RoIs validated by the Adaboost cascade. To train the classifier we use the LIBSVM [6] and their default parameters.

Hypothesis filtering Validated RoIs, as shown in fig. 4(b), are then grouped using a Mean Shift Clustering method [8]. This is an iterative algorithm that chooses a mean position from the position of neighbouring RoIs. Returned clusters (RoI positions) are considered the system response (see figure 4(c)). Finally, the system output, i.e., the estimated pedestrian positions, is given by the position of those resulting RoIs.

Results of the three steps applied to identify pedestrians on the sequences are showed in figure 4.

4 Experimental results

4.1 Detection validation

Test sequence captures are analyzed using a sliding window approach to detect pedestrians. Sliding windows are rectangles with size multiple of the pattern size ($width = 12$, $height = 36$), multiplied by a scale factor of 1.15.

To validate detections, we use the Pascale Challenge[4] overlapping criteria. To consider a detection correct, we calculate the overlapped area between the test RoI, RoI_t, and the ground truth bounding box B_{gt}:

$$A = \frac{RoI_t \cap B_{gt}}{RoI_t \cup B_{gt}}$$

If $A > 0.5$, RoI_t is considered a correct detection. Otherwise, it will be a false positive.

[4] `http://pascallin.ecs.soton.ac.uk/challenges/VOC/`

(a) Hypothesis generation

(b) Hypothesis validated

(c) Final RoIs

Fig. 4. The three steps of the MFS detection algorithm.

4.2 OpenCV object detector

The proposed approach, MFS-Detector, is compared against a widely used detector on still images: the OpenCV cascade of boosted classifiers.

(a) OpenCV-Det20 detection (b) OpenCV-Det25 detection

Fig. 5. Example of the behavior of both OpenCV detectors.

OpenCV provides a set of functions for the training of a cascade of boosted classifiers. Some functions are already compiled and allow to: select training images, train the cascade, and test the cascade.

The positive training dataset is the same employed by the MFS-Detector classifiers. The negative training dataset is composed of the 3353 captures of the training sequences without persons. The negative samples of the INRIA Person, 1219 images without persons of the training folder, and 454 images from the test folder, where also added to the negative training dataset.

The parameters employed on the training by the *opencv_training* application where the following:

- 4000 is the number of positive and negative samples to train each stage.
- the minimal desired hit rate for each stage classifier is 0.995.
- the maximal desired false alarm rate for each stage classifier is 0.5.
- the type of Haar features set is the full set, composed of 100871 features for the pattern size.
- 12 is the pattern sample with.
- 36 is the pattern sample height.
- for the other parameters we used default values.

Finally, two classifiers where trained with different number of stages. We called OpenCV-Det20 the pedestrian detector with 20 stages, and OpenCV-Det25 the pedestrian detector with 25 stages. Increasing the number of stages, leads to reduce the number of false alarms detected by the classifier, which can be seen in figure 5. But, at the same time, the detection rate can decrease.

4.3 Pedestrian detection results

Table 1 shows the results of the proposed approach, MFS-Detector, and the OpenCV object detectors.

As can be seen from the table, the successive steps of the MFS-Detector eliminate a great number of false alarms without decreasing the detection rate,

Test Sequence	Captures	Persons	MFS-Detector		OpenCV-Det20		OpenCV-Det25	
			% Det	# FA	% Det	# FA	% Det	# FA
TwoEnterShop2front	1750	2528	76.8	**1313**	**78.4**	7006	73.5	4417
TwoEnterShop3front	1474	1673	**61.4**	**481**	61.1	6199	54.8	4210
TwoLeaveShop1front	409	618	**81.2**	**269**	73.3	3477	68.0	2186
TwoLeaveShop2front	284	432	**79.4**	**126**	78.0	1499	69.7	983
EnterExitCrossingPaths1front	193	343	86.7	**209**	**88.0**	1499	82.2	1011
EnterExitCrossingPaths2front	367	639	66.6	**418**	**71.0**	3187	67.6	2105
OneLeaveShop1front	95	157	**87.6**	**70**	79.6	1099	70.7	694
OneLeaveShop2front	137	144	**91.2**	**48**	80.6	1017	69.4	647
OneLeaveShopReenter2front	405	586	**88.4**	**147**	86.2	4348	83.3	2139
OneStopEnter1front	740	1092	**84.5**	**459**	83.2	6022	80.4	3904
OneStopMoveEnter1front	761	3107	**73.3**	**967**	73.0	8815	67.4	5427
OneStopMoveNoEnter2front	207	908	**93.3**	**115**	84.8	5904	76.6	3737
OneStopNoEnter1front	688	325	81.2	**136**	**91.1**	2203	83.4	1380
OneStopNoEnter2front	1136	990	**86.7**	**183**	84.7	5939	80.1	3756
OneShopOneWait1front	1377	1136	**89.0**	**398**	84.7	5939	77.1	3931

Table 1. Comparison between MFS-Detector and OpenCV classifiers results.

which is not the case of the OpenCV object detector. In order to minimize the false alarms, the OpenCV detector need to add more stages to the cascade, but the detection rate also decreases.

Worst results in detection rate where reported on those sequences with a great number of pedestrian partially occluded by other pedestrians. It happens because the low resolution in the image captures. The pedestrians' colors are very similar and there are no visible edges to differentiate them with our feature families and the Haar-like features of the OpenCV detector.

5 Conclusions and future work

This article presented a pedestrian detector system that obtains promising results on the CAVIAR dataset. The proposed Movement Feature Space helped to detect movements and to generate descriptors to identify pedestrians in the scene. The algorithm works on this space, which is based on level lines, all along the three detection steps: movement detection, RoI generation, and RoI validation. The MFS-Detector obtained better results than the OpenCV object detector.

These are preliminary results and there is a work in progress aiming to improve the detection. This detection system will be employed later to analyze the behavior of pedestrians crossing the street in an intersection and their interaction with moving vehicles.

References

1. `http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1`
2. Aubert, D., Guichard, F., Bouchafa, S.: Time-scale change detection applied to real-time abnormal stationarity monitoring. Real-Time Imaging 10, 9–22 (2004)
3. Botchen, R., Bachthaler, S., Schick, F., Weiskopf, D., Ertl, T.: Action-based multi-field video visualization. IEEE Transactions on Visualization and Computer Graphics 14(4), 885–899 (2008)

4. Cao, F., Musse, P., Sur, F.: Extracting meaningful curves from images. Journal of Mathematical Imaging and Vision 22, 1519–181 (2005)
5. Caselles, V., Col, l.B., Morel, J.: Topographic maps and local contrast changes in natural images. International Journal on Computer Vision 33, 5–27 (1999)
6. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines, `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, accessed november 2011
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition. pp. 886–893 (2005)
8. Finkston, B.: `http://www.mathworks.com/matlabcentral/fileexchange/10161-mean-shift-clustering`, accessed on march 2012.
9. Gouiffes, M., Bouchafa, S., Zavidovique, B.: Segments of color lines - a comparison through a tracking procedure. In: ICINCO-RA. pp. 433–438 (2009)
10. Liu, C., P.C.Yuen, G.Q.: Object motion detection using information theoretic spatio-temporal saliency. Pattern Recognition 42, 2897–2906 (2009)
11. Mouchafa, S.: Motion detection invariant to contrast changes. Application to detection abnormal motion in subway corridors. Ph.D. thesis, UPMC Paris VI (1998)
12. Negri, P.: `http://pablonegri.free.fr/Downloads/RealAdaboost_PANKit.htm`, accessed march 2012.
13. Negri, P., Clady, X., Hanif, S., Prevost, L.: A cascade of boosted generative and discriminative classifiers for vehicle detection. EURASIP JASP 2008, 1–12 (2008)
14. O'Callaghan, R., Haga, T.: Robust change-detection by normalised gradient-correlation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2007)
15. SanMiguel, J., Cavallaro, A., Martinez:, J.: Adaptive online performance evaluation of video trackers. IEEE Transactions on Image Processing 21(5), 2812–2823 (2012)
16. Schölkopf, B., Smola, A.: Learning with Kernels. Suppor Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA (2002)
17. Vapnik, V.: The nature of Statistical Learning Theory. Springer (1995)
18. Veit, T., Cao, F., Bouthemy, P.: An a contrario decision framework for region-based motion detection. International Journal on Computer Vision 68(2) (2006)
19. Vijverberg, J., Koeleman, C., de With, P.: Tracking rectangular targets in surveillance videos with the gm-phd filter. In: Symposium on Information Theory in the Benelux. pp. 177–184. Eindhoven (2009)
20. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Converference on Computer Vision and Pattern Recognition. vol. 1, pp. 511–518 (2001)