Proceedings of the Federated Conference on Computer Science and Information Systems pp. 363–372
DOI: 10.15439/2019F291
ISSN 2300-5963 ACSIS, Vol. 18

Object detection in the police surveillance scenario

Artur Wilkowski, Włodzimierz Kasprzak, Maciej Stefańczyk
Insitute of Control and Computation Engineering,
Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
Email: artur.wilkowski@pw.edu.pl
{W.Kasprzak,M.Stefanczyk}@elka.pw.edu.pl

***Abstract*—Police and various security services use video analysis when investigating criminal activity. One typical scenario is the selection of object in image sequence and search for similar objects in other images. Algorithms supporting this scenario must reconcile several seemingly contradicting factors: training and detection speed, detection reliability and learning from sparse data. In the system that we propose a combined SVM/Cascade detector is used for both speed and detection reliability. In addition, object tracking and background-foreground separation algorithm together with sample synthesis is used to collect rich training data. Experiments show that the system is effective, useful and suitable for selected tasks of police surveillance.**

I. Introduction

POLICE and various security services use video analysis when investigating criminal activity. Long surveillance videos are increasingly searched by dedicated image analysis software to detect criminal events, to store them and to initiate proper security actions (see e.g. the P-REACT project [1]). Solutions to automatic analysis of surveillance videos seem already to be mature enough, as the research community is recently also involved in major benchmark initiatives [2], [3]. The computer vision research focus is now shifted to the analysis of video data coming from handheld, body-worn and dashboard cameras and on the integration of such analysis results with police- and public-databases.

In typical object detection scenarios, there are much data to learn from and major objective is to use them in effective manner. In a security-oriented environment the user interaction should be kept as simple as possible and preferably limited only to marking single object in a selected image frame and initiating search to find occurrences of similar objects in other frames of the processed sequence or other sequences. This imposes several constraints on the Machine Vision solution that need to be addressed.

First of all the system should learn on-line or nearly on-line. Secondly - the system must perform per-frame detection quickly and provide approximate results in short time. And thirdly - to system must be able to learn from sparse data.

In this paper, an effective and time-efficient algorithm for instance search and detection in images from handheld video cameras is proposed. The system uses a discriminative approach to differentiate the object from its foreground. In order to do so a combined Haar-Cascade detector and SVM classifier are used. We argue that this provides a very attractive trade-off between detection quality and training/detection times. Both the positive as well as negative samples are extracted only from training images.

Comparable detector solutions based on CNNs provide excellent detection performance [4]. Such solutions, however, rely on off-line training and training/detection speed is still a bottleneck for such systems. This effect is to some extent ameliorated by GPU utilization. Recent developments aim at reduction of detection times e.g. by cascading CNNs [5] or by detecting salient regions first using fuzzy logic [6] but significant reduction of training time is still an open area of research.

One contribution of the paper is the procedure of collecting as much realistic training data as possible providing limited user interaction. Ideally the system should be able to learn from a single ROI selection, all additional examples should be obtained automatically. Such least-user-effort approaches were already discussed e.g. for semi-automatic video annotation and detection systems, such as [7], [8]. In the cited approach, however, the user may be asked to annotated video several time (to decide about samples lying on decision boundary) which is not necessarily acceptable for all end users. An example of another successful detector that works on a single selection is given in [9]. The detector operates on sparse image representation (collection of SIFT descriptors) so it is very fast. Our initial experiments have shown that descriptor-based approaches works the best for highly textured and fairly complex objects.

The procedure of collecting training data given in this paper combines object tracking and background subtraction methods for semi-supervised collection of training windows together with foreground masks. The samples collected during tracking are further synthetically generalized (augmented) to enrich the training set. Scenarios, where tracking results are utilized for the collection of detector's training data, were already covered in literature, especially regarding tracking, with prominent examples [10], [11] or more recent CNN approaches [12], [13]. In such approaches the exact foreground-background separation (which is crucial for effective samples synthesis) is often neglected, since the algorithms typically have enough frames to collect rich training data.

The proposed methods were evaluated on a corpus of

This research was funded by NCBiR Agency, Warsaw, under BIOWIZ project, grant number DOB-BIO7/18/02/2015. The manuscript preparation was supported by statutory funds of the author's home institution (WUT).

TABLE I: Dictionary of abbreviations

Abbreviation	Expansion
CC	Cascade Classifier
CNN	Convolutional Neural Network
CSK	Circulant Structure of Kernels
EER	Equal Error Rate
FPR	False Positive Rate
GPU	Graphic Processing Unit
HD	High Definition
HOG	Histogram of Oriented Gradients
P-REACT	Petty cRiminality diminution through sEarch and Analysis in multi-source video Capturing and archiving plaTform
RBF	Radial Basis Function
RGB-D	Red Green Blue - Depth
ROC	Receiver Operator Characteristics
ROI	Region of Interest
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
TPR	True Positive Rate

surveillance videos and proved that its efficiency is good enough to be effective in supporting a user (police officer or security official) in their common working tasks.

The paper is organized as follows: in section II there is given a technical background and methodology used in our system, section III provides experimental results and IV contains conclusions. For reader's convenience Table I provides a short dictionary of abbreviations used in the paper.

II. Methods

A. Detector overview

In the system described in this paper we utilize a classic detection framework, where a sliding window with varying sizes is moved over each frame and for each location the selected image part is evaluated against information gathered from training samples. A crucial part of the detector is formed by a SVM classifier which is responsible for evaluation of each selected image part. A pure SVM classifier when applied to hundred of thousands candidate areas would be too slow to learn and detect, so in our scenario so pre-classification step utilizing HAAR-like features-based cascade classifier is applied to limit the number of candidate windows to about several hundreds. We claim that this simple structure combines good detection rate together with acceptable detection speed (about 10 full-HD frames per second on modest Core i5 computer) as well as fine training speed in typical scenarios (up to few minutes).

In essence the two-stage detector architecture resembles some significant modern CNN approaches, where the detection is divided into region-proposal part and the region recognition part (see: e.g. [14]). In our approach region proposal is performed by cascade classifier, and final classification is done by SVM classifier. Both methods offer reasonable training and detection speeds required for this application.

In our scenario sources of data are naturally sparse. Depending on user decision the detector can be trained either on one or a short sequence of training images. Therefore a critical part of our system are tools aiding user in an effortless collection of training examples from short image sequences as well as methods for artificial synthesis and generalization of training samples to provide the detector with the training data as rich as possible. These tools and methods are discussed in subsequent sections. The overall structure of the training procedure is given in Fig. 1.

B. Collection of positive training samples

Although for some patterns (which include e.g. flat patterns) good detection results can be obtained using only one selected sample that is further generalized and synthesized into a set with larger variability, in most cases detection results highly depend on size and diversity of input training set. In the scenario discussed in this paper these properties of the training set can (at least partially) be achieved by collecting samples from a short sequence of input images. Our scenario is organized as follows: (1) a user select object of interest using rectangular area, (2) the application tracks the object in subsequent frames of the sequence (with optional manual reinitialization), (3) object foreground masks are established using motion information.

1) Object tracking and foreground-background separation: For tracking of rectangular area an optimized version of CSK tracker [15] that utilizes color-names features [16] is used.

As a result of the tracking procedure we obtain a sequence of rectangular areas that encompass the object of interest in subsequent frames. In most cases both object foreground as well as background will be present in the tracked rectangle. However, if the object is moving against moderately static background we can exploit motion information to effectively separate object foreground from background by background subtraction.

Let the tracking results be described by a sequence of rectangular areas $\{R^1, \ldots, R^T\}$ and let us denote coordinates of pixel i as p_i, color attributes for pixel i at time t as c_i^t and a mean of color attributes in the background as

$$\bar{c}_i = \frac{1}{n_i} \sum_{t:p_i \notin R^t} c_i^t \tag{1}$$

where averaging factor n_i is the number of frames where tracking window does not contain pixel i and can be computed as $n_i = |\{t : p_i \notin R^t\}|$.

Now we can specify a background training sequence for each pixel $\{\hat{c}_i^t\}$

$$\hat{c}_i^t = \begin{cases} c_i^t & \text{if } p_i \notin R^t \\ \bar{c}_i & \text{if } p_i \in R^t \end{cases} \tag{2}$$

In accordance with the rule above, only pixels that at given time-step do not belong to the tracked area contribute to the background model computed for the image. Each pixel that always belong to tracked area is conservatively treated as foreground.

The background model adopted here follows algorithms from [17]. In this method scene color is represented independently for all pixels. The color for each pixel (both from

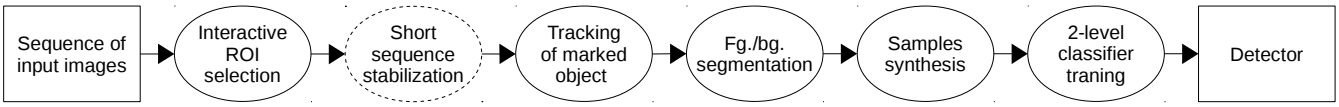


Fig. 1: Structure of the training procedure

Fig. 2: Results of automatic foreground-background separation

Fig. 3: Division into a positive (P) and negative (N1-N4) examples

background and foreground $BG + FG$) given the training sequence C_T, is modelled as:

$$p(c_i|C_T, BG + FG) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(c_i; \hat{\mu}_m, \hat{\sigma}_m) \qquad (3)$$

whereas the background model (BG) is built from the selected number of largest clusters in the color mixture

$$p(c_i|C_T, BG) = \sum_{m=1}^{B} \hat{\pi}_m \mathcal{N}(c_i; \hat{\mu}_m, \hat{\sigma}_m) \qquad (4)$$

where $\hat{\mu}_m, \hat{\sigma}_m$ are estimated means and standard deviation of normal components in the mixture, $\hat{\pi}_m$ are mixing coefficients M is the total number of mixtures and B is the selected number of foreground components. The pixel is decided to belong to the background when

$$p(c_i|BG) > c_{thr} \qquad (5)$$

Threshold c_{thr} can be interactively adjusted by the user. Exact algorithms for updating mixture parameters are given in [17]. Sample result of background subtraction procedure is given in Fig. 2

2) Image stabilization in a short sequence: The foreground-background segmentation procedure works best when stable camera position is available or image sequence is stabilized before segmentation. The system proposed here uses a simple stabilization procedure basing on matching of SURF features [18] and computation of homography transformation between pairs of images. The stabilization works on short subsequences of the original sequence. First frame to stabilize is the frame used for marking the initial region of interest. The procedure than aligns all subsequent frames to the first frame by evaluating homographies relating two images. In order to do so, matching methods from [19] and the Least Median of Squares principle [20] is utilized. To increase stabilization efficiency GPU-accelerated procedures for keypoints/descriptors extraction and matching from OpenCV library are utilized [21].

C. Collection of negative training samples

Negative samples that are used in detector training are extracted from the same sequence images that positive samples originated from. For each training image one fragment is used to extract positive sample, while the remaining part of the image is divided into at most four sources of negative samples as given in Fig. 3. Thus, an assumption is made that these remaining parts of the training sequence images do not contain positive samples. This assumption is not always valid, but may be strengthened by asking a user to mark **all** positive examples in the training sequence.

D. Positive samples generalization and synthesis

1) Geometric generalization: In this step 3D rotations of patterns and their masks are applied to collected pattern images and their masks. It is assumed that patterns are planar, so this generalization method can be useful only to some extent for non-planar objects. The rotation effect is obtained by an applying a homography transformation, imitating application of three rotation matrices $R_x(\alpha)$, $R_y(\beta)$, $R_z(\gamma)$ to a 3D object. The matrices correspond to rotations around x, y (in-plane rotations) and z (in-plane rotation) axes correspondingly. 3D rotation matrices are defined classically

$$\begin{aligned} R_x(\theta) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \\ R_y(\theta) &= \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \\ R_z(\theta) &= \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned} \qquad (6)$$

To compute the transformation, first a homography matrix is computed using formula

$$H = R - \frac{\mathbf{t}\mathbf{n}^T}{d} \tag{7}$$

where $\mathbf{n}$ is a vector normal to the pattern plane (we set it to $\mathbf{n} = (0, 0, 1)^T$), d is the distance from the virtual camera to the pattern (we set it arbitrarily to $d = 1$, since it only scales 'real-world' units of measurement) and R is the 3D rotation matrix and can be decomposed as

$$R = (R_x(\alpha) \cdot R_y(\beta) \cdot R_z(\gamma))^{-1} \tag{8}$$

In order for the image center (having world coordinates $C = (0, 0, d)^T$) to remain intact during transformation we define 'correcting' translation vector as

$$\mathbf{t} = -RC + C \tag{9}$$

Then we can specify artificial camera matrices as K_1 and K_2

$$K_1 = \begin{pmatrix} f & 0 & c_{in}^x \\ 0 & f & c_{in}^y \\ 0 & 0 & 1 \end{pmatrix}, K_2 = \begin{pmatrix} f & 0 & c_{out}^x \\ 0 & f & c_{out}^y \\ 0 & 0 & 1 \end{pmatrix} \tag{10}$$

where $(c_{in}^x, c_{in}^y)^T$ and $(c_{out}^x, c_{out}^y)^T$ are pixel coordinates of input and output image correspondingly, while f is the artificial camera focal length given in pixels. In this application we set f to be f_{mul} times larger input image dimension. Multiplier f_{mul} decides about the virtual distance of our virtual camera to the object. Smaller values introduce larger perspective distortions of the transformation, larger values introduce smaller distortions. We arbitrarily set f_{mul} to 10 implying only slight perspective distortions.

The final homography transformation applied to the pixels of the input image is given by

$$P = K_2 H K_1^{-1} \tag{11}$$

Rotation angles α, β and γ are selected randomly from the uniform distribution (denoted here as $\mathcal{U}$). The amount of rotation around axes y is twice times the amount of rotation around remaining axes to better reflect dominant rotations in human movement

$$\begin{aligned} \alpha &\sim \mathcal{U}(-1, 1) \cdot \delta_{max} \cdot 0.5, \\ \beta &\sim \mathcal{U}(-1, 1) \cdot \delta_{max}, \\ \gamma &\sim \mathcal{U}(-1, 1) \cdot \delta_{max}. \cdot 0.5 \end{aligned}$$

and δ_{max} is the parameters specifying the maximum extent of allowed rotation.

2) Intensity and contrast synthesis: In the proposed approach image intensity and contrast synthesis is applied in addition to geometric transformations. It is especially important for Haar-like features that lack intensity normalization. A simple linear formula is used here. For each pixel gray value I_{in} we have

$$I_{out} = a * I_{in} + b \tag{12}$$

where

$$a = 1 + c_{dev},\ b = I_{dev} - \mu_I \cdot c_{dev} \tag{13}$$

where μ_I is the average intensity of the sample and contrast deviation c_{dev} as well as intensity deviation I_{dev} are sampled from the uniform distribution $c_{dev} \sim \mathcal{U}(-1, 1) \cdot c_{max}$ and $I_{dev} \sim \mathcal{U}(-1, 1) \cdot I_{max}$. c_{max} is a parameter denoting the maximum allowed contrast change and I_{max} is a parameter denoting the maximum allowed intensity change. Changes in contrast preserve mean intensity of an image. After application of the formula its results are appropriately saturated.

3) Application of blur: Training and test samples may differ in terms of quality of image details due to different factors such deficiencies of optics used or motion blur. In our case we apply a simple Gaussian filter in order to simulate natural blur effects

$$\sigma = \mathcal{U}(0, 1) \cdot \sigma_{max} \cdot \min(I_{width}, I_{height}) \tag{14}$$

where I_{width} and I_{height} are image sample sizes and σ_{max} controls the maximum size of the Gaussian kernel.

4) Merging with the background: Generalized training images are superimposed on background samples extracted from negative examples of size ranging from about 0.25 to 4 times the positive sample size. Gray-level masks are used for seamless incorporation of positive samples into background images.

E. Detector training

Before training all training samples are resampled to a fixed size of 24x24 pixels. The detector training procedure is divided into two steps. In the first step the cascade classifier using HAAR-like features is trained. In our scenario for each cascade stage 300 positive samples and 100 negative samples are utilized. Minimum true positive rate for each cascade level is set to 0.995 and maximum false positive rate is set to 0.5. The classifier is trained for a maximum of 15 stages or until reaching ≈ 0.00003 FPR. The expected TPR is at least $0.995^{15} \approx 0.93$. By using these settings up to about 1000 detections are generated for each Full-HD test image.

During the second stage of training an SVM classifier is trained to handle samples that passed the first cascade classification. For most experiments the SVM classifier is trained on 300 positive and 300 negative samples. The SVM classifier uses Gaussian RBF kernel.

$$K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma ||\mathbf{x} - \mathbf{y}||^2) \tag{15}$$

The Gaussian kernel size γ and SVM regularization parameter C are adjusted using automatic cross-validation procedure performed on the training data. For SVM classification Histogram of Oriented Gradients features [22] are extracted. For each sample a 9-element histogram in 4x4 cells is created with 16x16 histogram normalization window overlapping by 8 pixels, thus giving $4 * 16 * 9 = 576$ HOG features in total.

Negative samples are extracted from Cascade Classifier decision boundary (containing samples that were positively verified by CC but still negative) if possible. If not - image

fragments used as background images for positive samples or other randomly selected samples are used. In all experiments OpenCV 3.1 [21] Cascade Classifier and SVM implementation are utilized.

Given our test data, the number of resulting support vectors in the SVM classifier varies between 200 and 400. Let us review one specific configuration: 'hat' pattern trained on 55 images with masks and pattern generalization settings $\sigma_{max} = c_{max} = 0$, $\delta_{max} = 0.7$, $I_{max} = 50$. After SVM metaparameter optimization we obtain SVM regularization parameter $C = 2.5$, RBF kernel size $\gamma = 0.5$ and the number of support vectors 233.

F. Detection and post-processing

During detection phase each test image is first processed by the cascade classifier typically returning several hundreds candidate areas. After this, each candidate area is examined by the SVM classifier and a score is assigned to each detection. The score is computed as the signed distance from the separating plane in support vector space with lowest negative scores treated as best matches and high positive scores as worst matches.

For each image only the best score area is considered for further processing. Frames from the test sequence are sampled and processed with increasing density (first,last and middle frame for start and then intermittent frames), to quickly produce some results for the user to review (non minima suppression is used to reduce clutter)

III. Experiments

A. Preliminary experiments

During the first stage of experiments there was selected a single test sequence '00012' with 1776 Full-HD frames. Using this sequence various parameter configurations were evaluated in order to assess basic properties of the solution proposed. Basing on these experiments some answers can be given regarding problems such as impact of utilization of two-layer detector on detection results and detection/training speed, impact of the method of selection of training samples on detection accuracy or impact of values of image synthesis parameters on overall quality. Above questions will be discussed in the following paragraphs. All experiments were performed on Intel Core i5 computer. During the first 3 experiments one sample pattern 'hat' was utilized, in the last experiment 3 other patterns 'logo', 'helmet' and 'shirt' were introduced. Examples of training samples are given in Fig. 4 and samples marked in full-frame image are given in Fig. 5. Filtered detection results for one test sequence presented in the form of a simple GUI are given in Fig. 6.

a) Two-layer detector: In the first experiment there was evaluated a trade-off between detection and training speed for different number of expected cascade stages k (Fig. 8). Identical parameters were used for all k except for the number of SVM training samples. For $k < 15$ there were used 900 positive and negative samples to accommodate for weaker selectivity of the 1-st detection stage. For $k \geq 15$ the default of 300 positive and negative samples were utilized as in all other experiments. The experiment shows that for low k training time is dominated by SVM training, for large k cascade training dominates. A good compromise for our data can be obtained for $k = 15$. Larger k obviously means also faster detection (Fig. 7), but also slightly worse detection results (Fig. 9) (likely due to utilization of more robust HOG features in the second stage).

b) Collection of training samples: In the next experiments there were compared detector performance for different training data collection methods. In the first place the data samples were collected using automatic tracking and foreground-background separation methods given in this paper. In the process 55 data samples from of 'hat' pattern were collected together with their automatically generated masks. The data consisted of images of a hat on top of a head, while the head was making full 180 degrees rotation around central axis. For comparison, a short sequence of training samples representing only 3 extreme head positions (*en-face* and two profiles) was utilized. For both sequences either appropriate foreground-background masks or no masks were used giving 4 different combinations of settings. The detection results are given in Fig. 10.

Not surprisingly the richest possible data source (55 frames with generated masks) gives the best results. It is valuable to note that for our data, application of both object tracking and automatic mask generation is substantial to get optimal results.

c) Synthetic generalization of training data: In these experiments different measures and intensities of samples synthesis were evaluated. The results are given in Fig. 11 and Fig. 12. The results show that moderate geometric as well as contrast and sharpness generalization provides best results. However, the selection of appropriate parameters is object and sequence-specific. E.g. it may be observed that near-flat surfaces e.g. 'logo' benefits from aggressive geometric distortions (i.e. larger rotation angles). In addition, the reduction of sharpness proved to work best for computer-graphics-generated samples.

d) Detection of various patterns: In the last of our preliminary experiments there was evaluated how the detector handles different types of patterns. Therefore, the pattern 'logo' was trained on a single training example with no mask, the pattern 'shirt' was trained on a sequence of 30 samples without a mask and the pattern 'helmet' was trained on 41 samples also without a mask. The result are given in Fig. 13.

It can be noted the relatively worse performance for the 'shirt' pattern, mainly due to numerous occlusions. Even in the case of the 'shirt' pattern we still have about 90% of successful hits for recall rates of 0.3. For best patterns such as 'helmet' we have 70% of positive examples with still 0 false positives!

In the course of experiments, it was observed that motion blur (inherent or originating from de-interlacing) is the most destructive type of noise regarding both training and detection phase. In addition, due to quite severe subsampling of the pattern (down to 24×24), the detector may suffer from problems in distinguishing between patterns differing only in

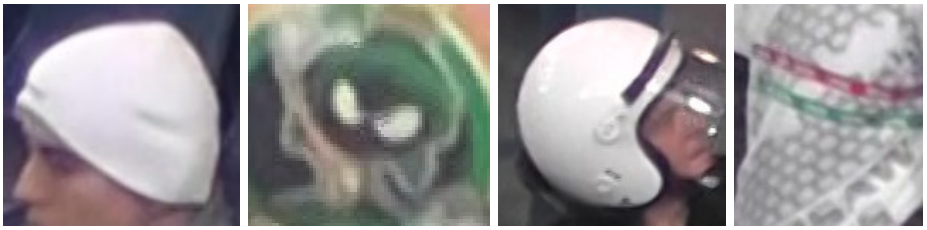


Fig. 4: Example training samples of 'hat', 'logo', 'helmet' and 'shirt'

Fig. 5: Frame with marked 'hat', 'logo', 'helmet' and 'shirt' samples

Fig. 6: Detection results filtered by minimum distance (25 frames) between hits

small details. On the other hand, due to this property, the detector should well handle also small patterns - only slightly bigger than the nominal 24×24 pattern size.

B. Large-scale experiments

Tests of the presented algorithm were conducted on a dataset containing 11 recordings, with nearly 30 thousand frames in total, with full HD resolution. Three patterns were created

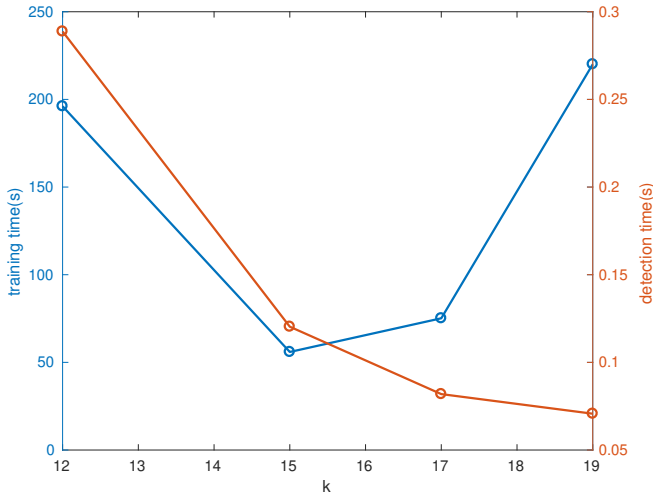


Fig. 7: Training/detection time vs. the number of cascades (k).

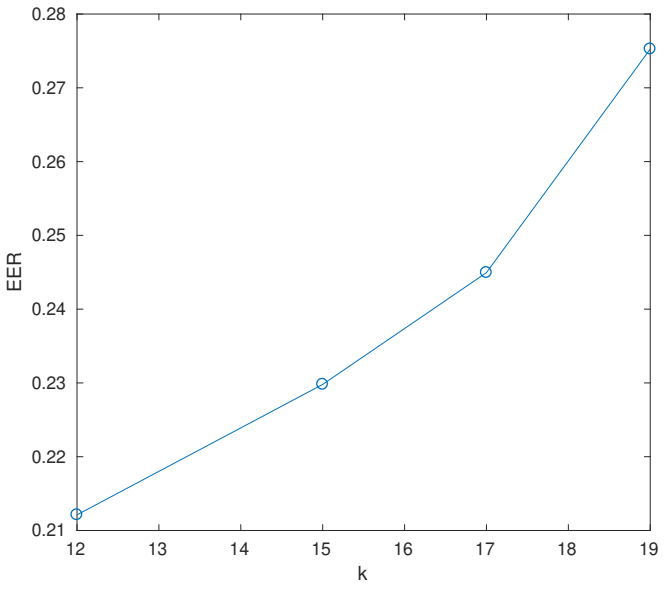


Fig. 8: 'hat' detection EER vs. the number of cascades (k)

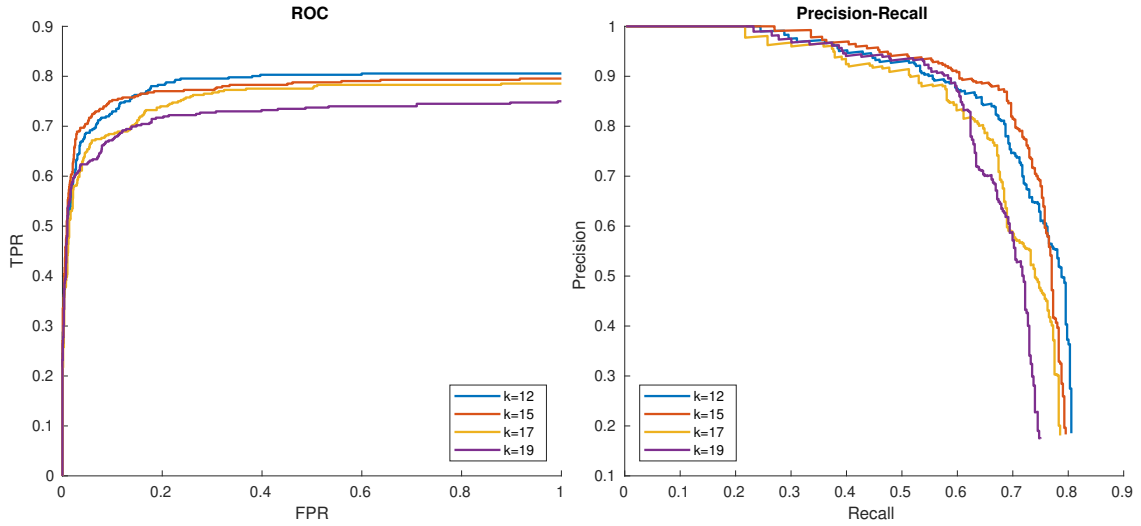


Fig. 9: 'hat' in '00012' detection results with respect to number of the requested cascade stages.

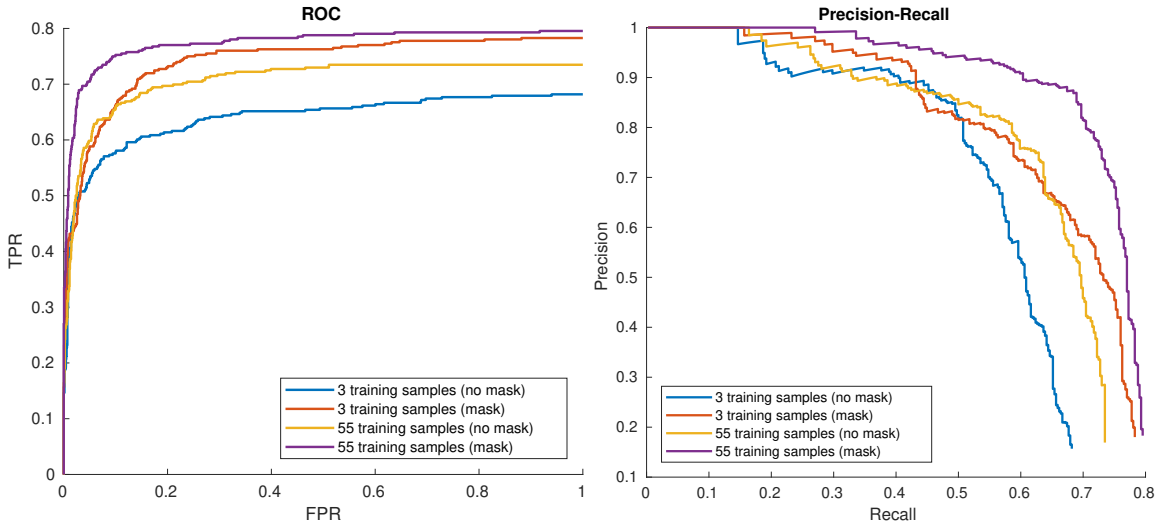


Fig. 10: 'hat' in '00012' detection results for different training data collection methods

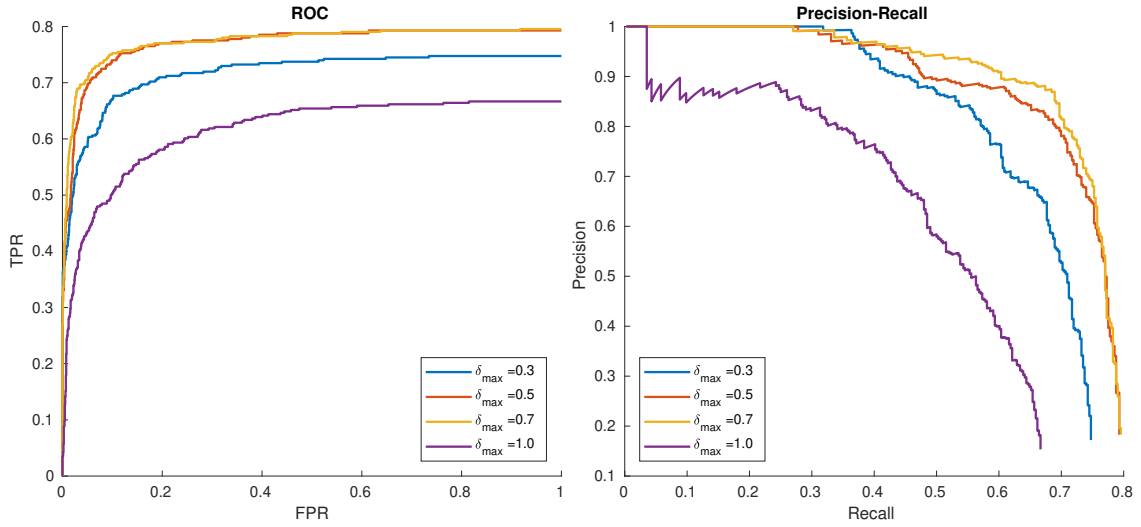


Fig. 11: 'hat' in '00012' detection results for different levels of geometric synthesis

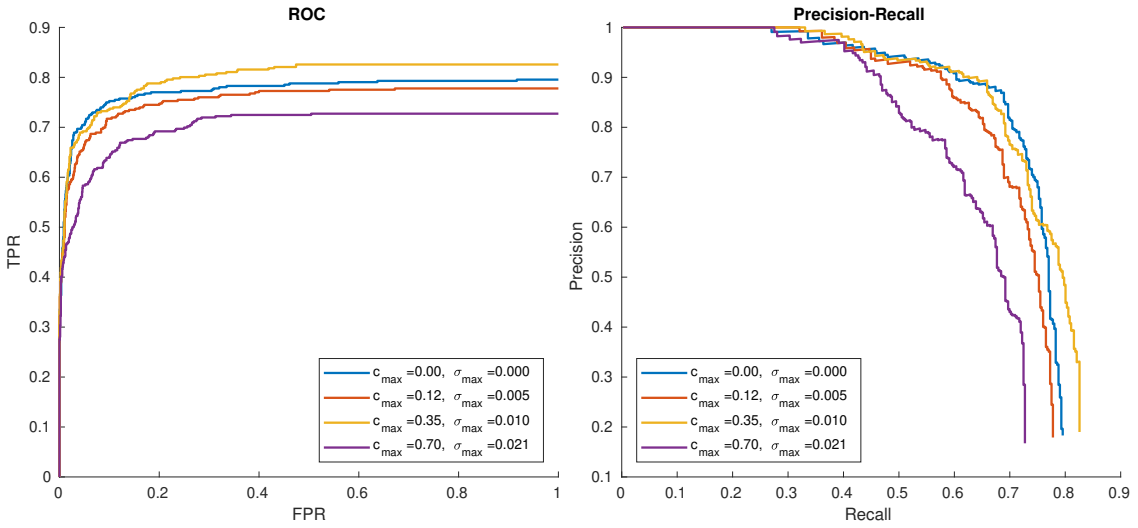


Fig. 12: 'hat' in '00012' detection results for different contrast and sharpness synthesis levels

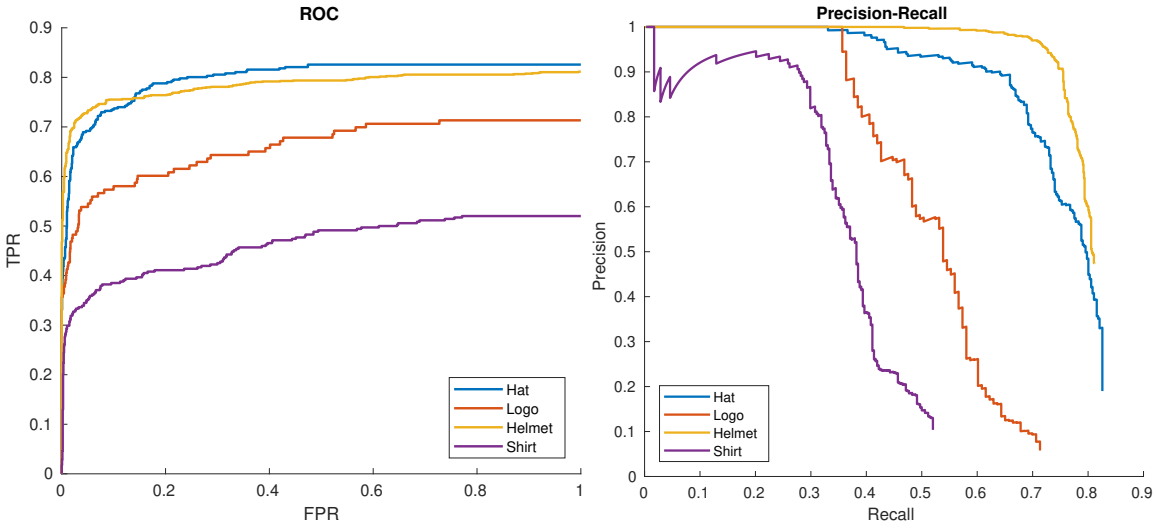


Fig. 13: ROC and PR curves of 'hat','logo','helmet' and 'shirt' detections in '00012' sequence

(Fig. 14), and all sequences were carefully labeled by hand to create ground-truth data. All patterns were created based on a single frame (one positive sample). As a training data, high quality still picture was used, with resolution scaled down to full HD.

Results of the experiments (ROC curve) for the selected pattern A is presented on Fig. 15a. EER is similar for all patterns A,B,C, and is equal to 25.3%, 28.3% and 28.0% for each pattern respectively. Accumulated EER equals to 27.4%. Obtained results resemble those from small dataset. Even though the training sample and query images were taken with different devices and had different quality, the algorithm gave satisfactory results.

Final addition to the testing scenario was the utilization of short sequences. For every short sequence, from all the results only the one with the best response was taken as a final detection and passed to further processing. Accumulated results for the sequences of length 5 is presented on Fig. 15b (remaining charts are given in supplemental materials). EER for them are, respectively: 27.4%. 15.1% and 14.3%. It was observed that the longer the sequence the smaller is the quality gain.

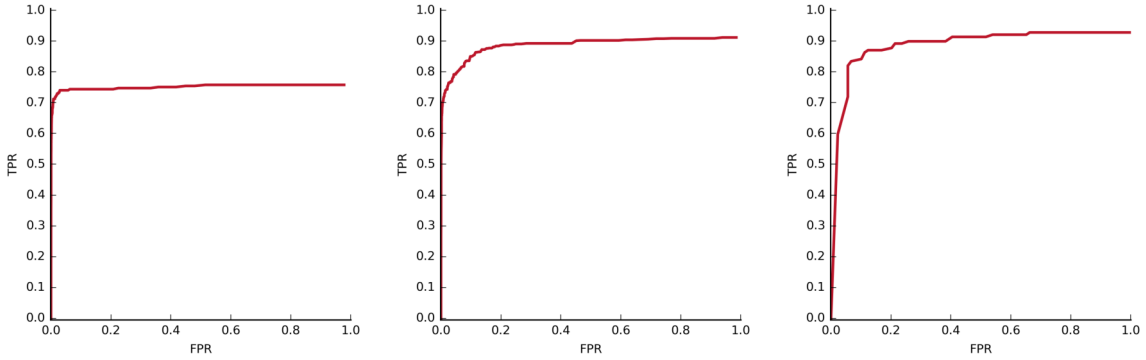
More tests were also conducted using one of the widely used dataset – RGB-D Object Dataset [23]. It contains multiple everyday objects, along with masks, that can be used to create models and short sequences of scenes with multiple objects. Fig. 15c presents sample results obtained for the cereal_1 object in desk_3 sequence. Model was created using only 7 views of the object in this case.

IV. Conclusions

In this paper, we presented a solution that can support work of police officers in surveillance tasks. The system proved to positively address difficult task requirements concerning sparse

(A) (B) (C)

Fig. 14: Selected test patterns



(a) (b) (c)

Fig. 15: (a) ROC curve for the pattern A. (b) Accumulated ROC curve for 5-elements sequence analysis. (c) ROC curve for cereal_1 object in desk_3 sequence

training data, quick learning and fast and reliable detection. An attractive training/detection speed and recognition rate trade-off was obtained by the application of 2-layer cascade/SVM classifier. The system proposed can learn from a single training sample, but also can collect samples from short image sequences with only small user supervision in order to obtain rich training data. Performance of the system vary depending on the type and quality of training/test data, but we argue that on average results are satisfactory and even not-the-best results provide sufficient information to be useful in practical surveillance scenario.

REFERENCES

[1] J. Arraiza, N. Aginako, G. Kioumourtzis, G. Leventakis, G. Stavropoulos, D. Tzovaras, N. Zotos, A. Sideris, E. Charalambous, and N. Koutras, "Fighting volume crime: an intelligent, scalable, and low cost approach," *9th Summer Safety & Reliability Seminars, SSARS 2015, June 21- 27, 2015, Gdansk/Sopot, Poland*, 2015.

[2] S. Blunsden and R. Fisher, "The behave video dataset: Ground truthed video for multi-person behavior classification," *Annals of the BMVA*, vol. 2010, no. 4, pp. 1–11, 2010.

[3] G. Awad, C. G. M. Snoek, A. F. Smeaton, and G. Quénot, "Trecvid semantic indexing of video: A 6-year retrospective," *ITE Transactions on Media Technology and Applications*, vol. 4, no. 3, pp. 187–208, 2016. doi: 10.3169/mta.4.187 Invited paper.

[4] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[5] D. Zeng, F. Zhao, S. Ge, and W. Shen, "Fast cascade face detection with pyramid network," *Pattern Recognition Letters*, vol. 119, pp. 180 – 186, 2019. doi: https://doi.org/10.1016/j.patrec.2018.05.024 Deep Learning for Pattern Recognition. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865518302125

[6] M. Woźniak and D. Połap, "Object detection and recognition via clustered features," *Neurocomputing*, vol. 320, pp. 76 – 84, 2018. doi: https://doi.org/10.1016/j.neucom.2018.09.003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231218310634

[7] Y. Abramson and Y. Freund, "Active learning for visual object detection," UCSD, Tech. Rep., 01 2006.

[8] ——, "SEmi-automatic VIsual LEarning (SEVILLE): Tutorial on active learning for visual object recognition," *Proc. CVPR*, 2005.

[9] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ser. ICCV '03. Washington, DC, USA: IEEE Computer Society, 2003. ISBN 0-7695-1950-4 pp. 1470–. [Online]. Available: http://dl.acm.org/citation.cfm?id=946247.946751

[10] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, July 2012. doi: 10.1109/TPAMI.2011.239

[11] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, June 2008. doi: 10.1109/CVPR.2008.4587583. ISSN 1063-6919 pp. 1–8.

[12] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017. doi: 10.1109/ICCV.2017.330 pp. 3057–3065. [Online]. Available: https://doi.org/10.1109/ICCV.2017.330

[13] K. Kang, W. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutional neural networks," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 817–825, 2016.

[14] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[15] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-33765-9_50. ISBN 978-3-642-33765-9 pp. 702–715. [Online]. Available: https://doi.org/10.1007/978-3-642-33765-9_50

[16] M. Danelljan, F. S. Khan, M. Felsberg, and J. v. d. Weijer, "Adaptive color attributes for real-time visual tracking," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014. doi: 10.1109/CVPR.2014.143. ISSN 1063-6919 pp. 1090–1097.

[17] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2, Aug 2004. doi: 10.1109/ICPR.2004.1333992. ISSN 1051-4651 pp. 28–31 Vol.2.

[18] H. Bay, T. Tuytelaars, and L. Van Gool, *SURF: Speeded Up Robust Features*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417. ISBN 978-3-540-33833-8. [Online]. Available: https://doi.org/10.1007/11744023_32

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov 2004. doi: 10.1023/B:VISI.0000029664.99615.94. [Online]. Available: https://doi.org/10.1023/B:VISI.0000029664.99615.94

[20] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. John Wiley & Sons, Inc., 2005, ch. Algorithms, pp. 197–215. ISBN 9780471725381. [Online]. Available: http://dx.doi.org/10.1002/0471725382.ch5

[21] Itseez, "Open source computer vision library," https://github.com/itseez/opencv, 2015.

[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, June 2005. doi: 10.1109/CVPR.2005.177. ISSN 1063-6919 pp. 886–893 vol. 1.

[23] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1817–1824.