Video Surveillance Classification-based Multiple Instance Object Retrieval: Evaluation and Dataset

Catalin Alexandru Mitrea[1,2], Ionuţ Mironică[1], Bogdan Ionescu[1,3], Radu Dogaru[1]
[1] LAPI & Natural Computing Labs, University "Politehnica" of Bucharest, 061071, Romania
[2] UTI GRUP, Bucharest, 020492, Romania
[3] DISI, University of Trento, 38123 Povo, Italy
Email: *catalin.mitrea@uti.ro*, {*imironica,bionescu*}*@imag.pub.ro*, *radu_d@ieee.org*

***Abstract*—In this paper we propose a classification-based automated surveillance system for multiple-instance object retrieval task, and its main purpose, to track of a list of persons in several video sources, using only few training frames. We discuss the perspective of designing appropriate motion detectors, feature extraction and classification techniques that would enable to attain high categorization accuracy, and low percentage of false negatives. Evaluation is carried out on a new proposed dataset, namely Scouter dataset, which contains approximately 36,000 annotated frames. The proposed dataset contains 10 video sources, with variable lighting conditions and different levels of difficulty. The video database raises several challenges such as noise, low quality image or blurring, increasing the difficulty of its analysis. Also, the contribution of this paper is in the experimental part, several valuable interesting findings are reported that motivate further research on automated surveillance algorithms. The combination and calibration of appropriate motion detectors, feature extractors and classifiers allows to obtain high recall performance.**

I. Introduction

Over the past several decades, automated video surveillance techniques represented an important research domain. Today's public agencies or big companies are faced with a critical need to protect employees or citizens and assets from possible threats. This problem can be solved with a security system that enables rapid response to security breaches and prompt investigation of events.

Technology has reached a stage where mounting cameras to capture video imagery is cheap, but finding available human resources to watch and annotate the video frames is expensive. In this scenario, high performing automated video surveillance becomes essential. However, the main limitation of automated video surveillance remains in the searching capabilities. Once one has identified a possible target event, the system is not able to provide tracking capabilities of the entities causing that event during previous recordings, e.g., back-track a possible burglar, an object or finding the instances of a vehicle. This is actually done manually, by human operators. Considering the fact that a typical video surveillance system, in its simplest form (using only one video source), involves the recording of countless hours of video footage, manually searching the footage is hugely time consuming and at the same time inefficient and often unreliable. In practice, video surveillance systems feature tens of video sources, making the problem even more challenging.

Many algorithms have been proposed for automated surveillance systems. Nevertheless, these approaches are useful for real-time situational awareness; however, they are yet to be tied in with video database management concepts to make this type of analysis possible in a forensic fashion, as it is the case of this paper. A more useful paradigm for video surveillance retrieval applications involves describing the visual content of video scenes and extract list of objects - then allowing a user to create queries about those objects [6].

The primary goal of this paper is to develop a system for providing content-based search capabilities within multi-source video surveillance footage. The proposed system is capable of automatically identify the occurrences of a certain object of interest during video footage.

A second goal is to create a realistic scenario to test the proposed system. In this respect, we create a new dataset that represents a realistic, natural and challenging scenario for video surveillance domain in terms of diversity in scenes. The dataset contains surveillance videos recorded in a real public institution thus addressing a real world scenario, and its purpose is to track a list of persons in several video sources.

The reminder of the paper is organized as follows. Section II discuses several relevant video surveillance approaches and situates our work accordingly. The proposed system is presented in Section III and Section IV presents the proposed dataset and ground truth. Section V reports the experimental results. Finally, Section VI provides a brief summary and concludes the paper.

II. Previous work

There have been a series of research efforts based on the notion of applying content descriptors (quantified features) to large archives of video data [1]. An overview of the major achievements in this filed are presented in [2] [3].

Over the past several decades, the research have focused on two major facets of the problem: content descriptors extraction [2], and intuitive interfaces for video query creation and data mining [4].

In general, all existing approaches rely on efficient content description of the video information as an intermediate step. Many information sources have been exploited: color [8], texture [9], shape [10], temporal and motion [11], audio [8] and textual information [12]. Other efficient approaches are relying on describing the characteristics of feature points. Their success is due to the high invariance against image perturbations such as change of perspective, change of scale,

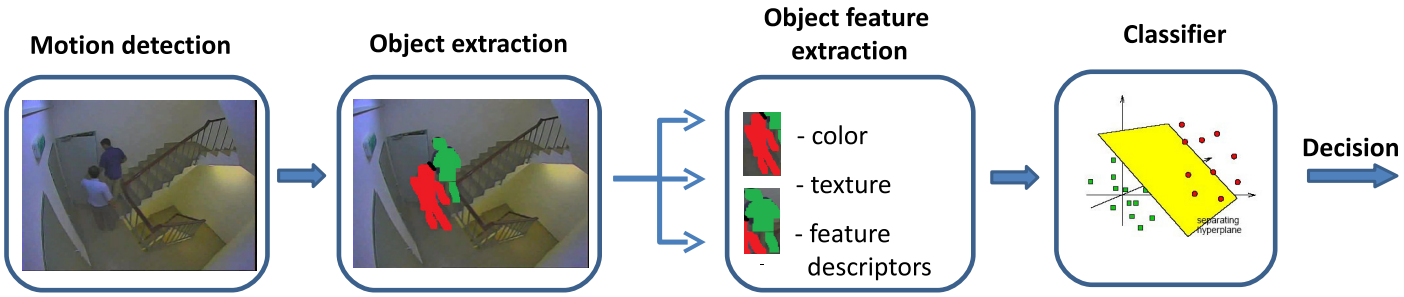

Figure 1: The proposed automated surveillance system.

rotations, translations and illumination changes [13]. More recently, these notions have been extended to capture also the temporal information of a video, e.g., Cuboid detector, Hessian 3D detector or SURF 3D [14]. In spite of their good performance in video indexing tasks, feature descriptors are limited by their computational complexity (e.g., processing a video database may take days) that makes them unsuitable for real-time scenarios. In this respect, current research addresses the development of low complexity algorithms to combine global with local strategies. For instance, in [15] the authors proposed the use of the Fisher kernel to model variation in time for frame-based video features. The method obtained encouraging results on several video scenarios (genre classification, sport and daily activities recognition) with low computational costs.

For automated surveillance algorithms, most of the contribution has been made to find automatic ways of describing video contents with parameters having enough representative power for the retrieval task. The approaches focused on the understanding of video contents using the visual and spatio-temporal information [5]. Many research laboratories produced a number of intelligent video processing algorithms and systems designed specifically around security applications. Also, these technologies are now becoming commercially available through products like ObjectVideo's VEW (Video Early Warning) [7] as a real-time physical security tool.

Most of the instance based classification algorithms have been evaluated on several public datasets, such as the KTH [31], Weizmann [32] and CAVIAR [33] datasets. However, these datasets are created in the context of event detection and action recognition problems, which are unrealistic for a real-world surveillance because they consist of short clips showing one action by one individual. Most of them have been developed for sports action recognition, but, these scene conditions do not apply effectively to provide scenarios for tracking capabilities of the entities. In order to satisfy the more complicated real-life scenarios, there is a need for a new and more complex dataset.

Our major contributions are as follows: (1) we propose a classification-based automated surveillance system for multiple-instance object retrieval task, (2) we introduce a new public camera surveillance video dataset, which provides realistic and diverse event examples (3) this data is accompanied by detailed annotations which include object routes and provide solid basis for quantitative evaluation for automated surveillance tracking.

III. The Proposed System

A. System architecture

The architecture of the proposed system is presented in Figure 1, and it consists of three different layers. First, the cameras collect the video information, which is transmitted to the motion detection layer. This module targets the extraction of moving objects, such as persons or cars. Motion analysis is very important because it optimizes the next stages performance by selecting relevant information, removing the irrelevant and so reducing the computational load.

Afterwards, for each object a descriptor is computed. Feature extraction component addresses the creation of visual patterns for each segmented moving object in the video. Thus, four visual features are calculated, Histogram of Oriented Gradients (HoG), Color Naming (CN) histogram, Color moments (CM) and Local Binary Patterns (LBP). All these features were chosen due to their robustness, compact representation and significance for human perception.

The final layer is represented by the classification algorithm. In order to find the most suitable classifier, we used a wide range of classification algorithms: Naïve Bayes (NB), Nearest Neighbor (KNN), Decision trees, Random forests (RF) and Support vector machines (SVM).

Each of the processing steps is detailed in the following.

B. Motion detectors

Motion detection algorithms represent the first component of our system. These algorithms have as main purpose to obtain motion information, which further is required for objects's extraction. We use in our experiments three types of motion detectors:

- *Background subtraction motion detectors* represent a technique where the image's foreground is extracted using a set of frame differencing algorithms. For this paper, we used the method presented in [16], where the authors propose the use of an Gaussian probabilistic density function (pdf) on the

most recent n frames. Every pixel is characterized by mean μ_t and variance σ_t^2, and it is classified as object if the following condition is accomplished:

$$\frac{|(I_t - \mu_t)|}{\sigma_t} > th \quad (1)$$

where I_t is the intensity of the current pixel, and the th represent a threshold (usually th = 2.5).

- *Accumulative optical flow method* [17] is based on the integration of accumulative optical flow and double background filtering method (long-term background and short-term background). The biggest advantage of this algorithm is that it does not need to learn the background model from hundreds of images and can handle quick image variations without prior knowledge about the object size and shape.

- *Kalman filter motion detector* [18], also known as linear quadratic estimation (LQE), uses a series of measurements observed over time, containing noise (random variations) and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone. More formally, the Kalman filter operates recursively on streams of noisy input video data to produce a statistically optimal estimate of the underlying system state. Knowledge of the state allows theoretically prediction of the future (and prior) dynamics and outputs of the deterministic system in the absence of noise.

C. Content descriptors

For video descriptors we have used a broad range of visual descriptors including: color, texture, and feature descriptors. Competitive results have been obtained using these descriptors on other surveillance datasets. It is well know that different modalities tend to account for different information providing complementary discriminative power.

In order to describe the visual content, we compute the following features:

- *HoG features (81 values)* - [19] those features exploits local object appearance and shape within an image via the distribution of edge orientations. The image is divided into small connected regions (cells) and for each of them building a pixel-wise histogram of edge orientations is computed. In the end, the combination of these histograms represent the final descriptor.

- *Color Naming histogram (11 dimensions)* describes the global color contents and uses the Color Naming (CN) Histogram proposed in [20]. It maps colors to 11 universal color names: ”black”, ”blue”, ”brown”, ”grey”, ”green”, ”orange”, ”pink”, ”purple”, ”red”, ”white” and ”yellow”. We select this feature, instead of the classic color histogram, because the color naming histogram is designed as a perceptually based color naming metric that is more discriminative and compact.

- *Color moments (225 dimensions)* [21] provide a measurement for color similarity between images. There are three central moments of an image’s color distribution: mean, standard deviation and skewness. The image is divided in a 5x5 grid, and a color moment descriptor is computed for each cell.

- *Local Binary Pattern (256 dimensions)* [22] has become a popular approach in various applications, mainly because of its discriminative power and computational simplicity. Local Binary Pattern (LBP) represents a simple texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

D. Classification algorithms

For classification we use the Weka environment [24] which provides many implementations of the classification algorithms. We have tested the following methods:

- *Naïve Bayes* represents a classification algorithm based on Bayes rule and assumes that all features from vector descriptor are conditionally independent of one another [25]. The value of this assumption is that it dramatically simplifies the representation of P($X_j|Y$), and the problem of estimating it from the training data. Naïve Bayes classifier requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix. In spite of his simplified design and suppositions, Naïve Bayes algorithm had quite good results in many complex real-world situations.

- *Nearest Neighbor* [26] represent a type of instance-based learning (known as lazy learning) where the function is approximated locally, without any training phase. When using a k-nearest neighbor algorithm, the input is classified by taking a majority vote of the k (where k is some user specified constant) closest training records across the dataset.

- *Decision trees* [27] represent one of the most often used classification algorithms in data-mining systems. The attractiveness of decision trees is due to the fact that decision trees represent rules. Rules can readily be expressed so that humans can understand or even directly use them in a database access language like SQL so that records falling into a particular category may be retrieved. Decision trees use a graph approach to compare competing alternatives and assign values to those alternatives by combining uncertainties, costs, and payoffs into specific numerical values.

- *Random forests* [28] consists in an ensemble learning method for classification, created by adding a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees. The leaf nodes of each tree are labeled by estimates of the posterior distribution over the classes, and each internal node contains a test that best splits the space of data to be classified.

- *Support vector machines* [29] have become extremely successful in domains as pattern classification or regression. These represent neural networks with two layer architecture that constructs a hyperplane or set of hyperplanes in a high dimensional space, which can be used for classification tasks. For this approach, we used two types of SVM kernels: a fast linear kernel and the RBF nonlinear kernel. While linear SVMs are very fast in both training and testing, SVMs with an polynomial kernel is more accurate in many classification tasks.

IV. The proposed Dataset

In order to evaluate any video analysis algorithm it is necessary to define a methodology. However, none of the existing datasets cannot be applied on our specific problem. For this reason, we decided to develop a general framework for evaluation.

Figure 2: Sample frames from the Scouter dataset.

Figure 3: Example of annotation of frame no. 83 (with the two sets of coordinates - $[x_1, y_1]$ and $[x_2, y_2]$).

The Scouter dataset[1] represents a manual indexed video collection and its main propose is to evaluate algorithms for complex automated surveillance scenarios. It is composed by videos documents, acquired with several video surveillance cameras installed in the headquarter of UTI company[2]. The video content was recorded on three different dates and periods of the day. A variety of camera viewpoints and resolutions were included, and actions are performed by many different people. The video surveillance system consists in 10 analog cameras positioned in various locations (4 indoor cameras and 6 outdoor cameras). Scouter dataset contains broad categories of activities which involve both human and vehicles (see some examples in Figure 2).

The dataset consists of 30 video documents (3 different days x 10 cameras). The videos are recorded at 6 to 10 fps, with 704 x 675 resolution. In total, the collection contains (3 days) x (10 cameras) x (average 120 seconds clip) x (10 frames per second) = approximately 36,000 annotated frames. From these videos, the tracking coordinates from each of the moving objects recorded on the scenes are stored in independent files. Annotations were made with a special developed application (EasyLabel [34]) which allows loading a group of images and drawing a rectangle of the object of interest and assign a name. The four coordinates (an example is presented in Figure 3) containing the upper left corner and bottom right corner (x_1, y_1) and (x_2, y_2) are automatically saved in text files (csv format) together with other related information (the number or the names of the objects). Labeled objects (humans) varies from 50 x 50 pixels to 250 x 350 pixels. Also the object of interest may appear with luggage and/or with a backpack baggage.

The video footage contains variable lighting conditions as well as different levels of difficulty. Also, the video documents includes several challenges such as noise, low quality image or blurring, increasing the difficulty of its analysis. The dataset is divided in two parts: a training set and a test set. The training set contains 180 frames: 60 examples that contains the sought object and 120 frames for negative examples. All these frames come from only one camera. The test dataset contains the other part of the dataset, that is approximately equal to 36,000 annotated frames.

Most of the automated surveillance algorithms have been evaluated on several public datasets, such as the KTH [31] and Weizmann [32] datasets. The KTH and Weizmann datasets contain several human actions performed over homogeneous backgrounds. However, these datasets are not suited for our scenario, but for event detection, which is not the problem that we try to solve. A comparison between our dataset and the KTH and Weizmann datasets is presented in Table I.

[1]the dataset was made publicly available and can be downloaded here: http://uti.eu.com/pncd-scouter/rezultate.html

[2]http://www.uti.eu.com/

Table I: Comparison of characteristics between the Scouter dataset and KTH and Weizmann datasets.

	KTH	Weizmann	Scouter
Max. Resolution (w x h)	160x120	180 x 144	704 x 675
Human Height in Pixels	80 to 100	60 to 70	50 to 350
Human to video height ratio	65 to 85%	42 to 50%	10 to 60%
Scenes Viewpoint Type	Side	Side	Varying
Natural Background Clutter	No	No	Yes
Incidental Objects/Activities	No	No	Yes
Multiple annotations on movers	No	No	Yes

V. Experimental results

A. Evaluation

To assess the retrieval performance, we use several measures. First, we compute the classical precision and recall. Precision represents the proportion of the true positives against all the positive results (measure of false positives) and recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database (measure of false negatives). We also compute the $F_\beta - score$ [35] that combines the precision and the recall:

$$F_\beta = (1+\beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}} \quad (2)$$

where b represent the parameter that allows us to weigh recall more than precision or vice versa. This is an important property of the $F - Score$, and is the primary reason why this measure was chosen. Recall from an automated surveillance system should produce no false-negatives and a minimal number of false-positives. For this reason, recall is weighted twice as much as precision by setting $\beta = 2$ when calculating $F_2 - score$.

In the following subsections, we present our experiments. The first experiment (Section V-B) motivates the choice of the best motion detection algorithm that provides the best accuracy. In the second part (Section V-C) we present an in-depth evaluation of several feature extraction and classification techniques that would enable to attain high performance.

B. The evaluation of motion detectors

In this experiment we study the influence of motion detectors algorithms on the systems' performance. We tested three types of motion detectors: background subtraction motion detector [16], accumulative optical flow method [17] and Kalman filter motion detector [18], presented in Section III-B.

The motion detection methods were evaluated using only one video document, which consists of 362 labeled frames (180 for training and 182 for testing). For brevity reasons, we use only the LBP feature and SVM with RBF nonlinear kernel, which obtain good results in preliminary tests. The performance of each motion detector is presented in Table II.

The highest precision is obtained with Kalman filter motion detector, namely 75%. However, the value of recall parameter is very low, which means that the algorithm produces a high number false-negatives. On the other other hand, background subtraction motion detector obtains similar precision performance ($precision = 74\%$), but a high true positive rate ($recall = 86\%$). The smallest increase in performance is obtained with the accumulative optical flow, which has been shown to be very sensitive to its parameters optimization.

Table II: Comparison of system performance between motion detectors algorithms.

Motion detection algorithm	Precision	Recall
Background subtraction motion	74%	86%
Accumulative optical flow method	58%	55%
Kalman filter motion detector	75%	48%

An example of object extraction is presented in Figure 4. The first image represents a frame sample, with three annotated persons. However, it can be observed that only the background subtraction motion algorithm detects all objects that moves.

We conclude that the background subtraction motion detector is more suitable for our task. In all the following experiments we will use only this motion detector.

C. The evaluation of the system

The final experiment consists of comparing several state-of-the-art descriptors and classifiers pairs.

Given the specificity of the task, i.e., automated video surveillance, we tested several visual descriptors which are known to perform well on image retrieval tasks, namely: HoG features, Color Naming histogram, color moments and Local Binary Pattern (see Section (see Section III-C). Also, we combines all the descriptors, using an early fusion strategy, when the features are into one vector [23]. We train our models using a broad category of classifiers: nearest neighbor (using 1, 3 or 5 neighbors), Naïve Bayes, decision trees, random forests, linear SVM and SVM with RBF classifier (see Section III-D).

Figure 5 presents the values of precision and recall scores. Best descriptor precision is obtained by LBP-5KNN, namely 39.36%. Similar performances are performed with HoG and CM features (38.18% and 34.37%). On the other hand, the CN features obtain lower precision rates with 4 to 5 percents. Overall, best precision is obtained using the early fusion strategy paired with 1KNN classifier (46.30%).

In terms of recall, the highest results are obtained by LBP and SVM with RBF kernel, namely 92.17%. Interesting to observe is that except early fusion, all descriptor - classifier pairs denote similar performance in terms of precision (varies between 33.11 to 39.36 %), while it varies substantial in terms of recall (between 49.81% to 92.17%). Even though precision values are lower comparing with recall, we consider that recall measure is more relevant for multiple-instance object retrieval tasks in current scenario. A high precision means that our system returned substantially more relevant results than irrelevant, while high recall means that that our system returned most of the relevant results (most instances of the

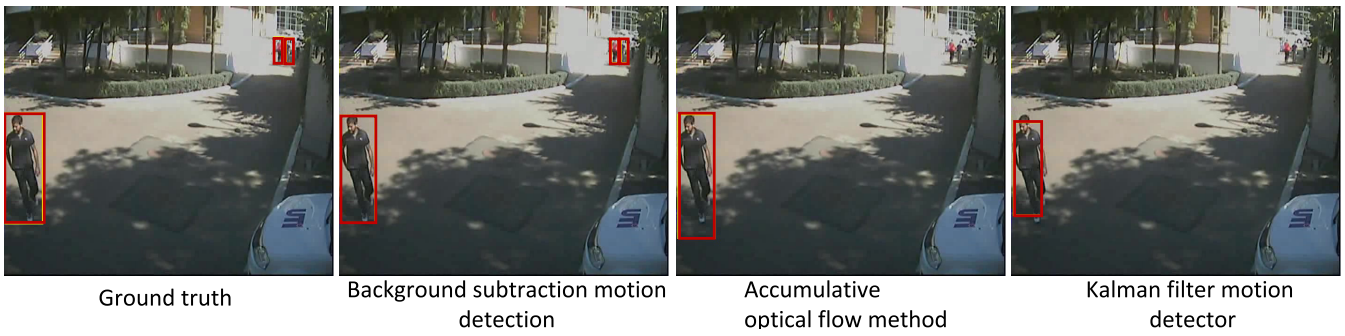

Figure 4: Examples of the motion detection results on a sample frame. First image represents the ground truth (three moving objects), the second image shows results of the background subtraction motion, the third present the performance of accumulative optical flow method, and the fourth image present an example for Kalman filter motion detector.

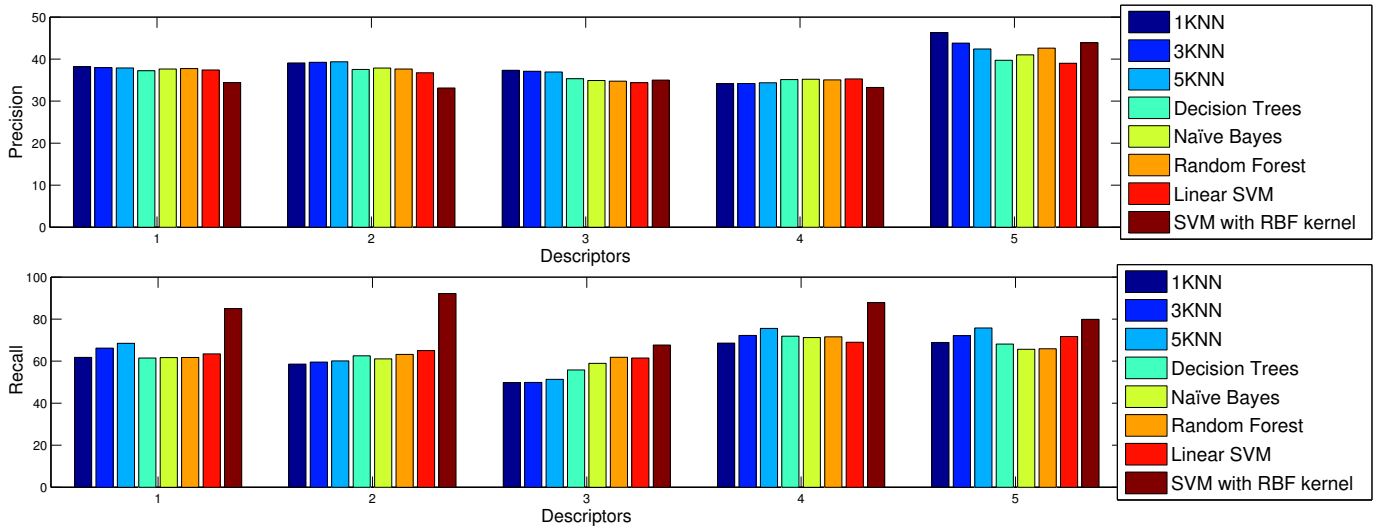

Figure 5: Precision and recall values using various descriptor - classifier combinations (1 - HoG, 2 - LBP, 3 - CM, 4 - CN, 5 - Early fusion of all features).

Table III: Comparison of system performance using different feature / classifier combinations ($F_2 - score$ values).

Classification Algorithm	HoG	LBP	CM	CN3x3	Early Fusion
1 KNN	54.98%	53.28%	46.69%	57.05%	62.75%
3 KNN	57.62%	53.97%	46.67%	59.09%	63.88%
5 KNN	58.97%	54.39%	47.61%	60.95%	65.46%
Decision Trees	54.38%	55.18%	50.01%	59.43%	59.60%
Naïve Bayes	54.69%	54.41%	51.80%	59.10%	58.61%
Random Forest	54.77%	55.63%	53.48%	59.23%	59.39%
Linear SVM	55.69%	56.35%	53.11%	57.93%	61.42%
SVM with RBF kernel	65.70%	67.94%	57.00%	66.12%	**68.65%**

objects to be found was returned), which is more relevant to video surveillance tasks.

As we consider the recall measure more relevant for our scenario, in Table III are represented $F_2 score$ results. Best results are obtained using the LBP with SVM-RBF pair while overall best score is obtained using early fusion technique and SVM-RBF as classifier (68.65%). Lowest results (46.67%) are obtained by CM and 3KNN pair. In this experiments we have obtained best results by combining all four descriptors (early fusion) while separately best results using RBF descriptor combined with SVM-RBF classifier. This is to the fact that RBF descriptor is a powerful feature for texture classification while SVM with RBF kernel can efficiently perform a non-linear classification which is suitable for our current scenario.

Figure 6 presents several system responses, when we use the best system configuration (SVM with RBF kernel and early fusion): first three lines represent the true positives (TP) examples in which the object found by the system are correctly identified according to the ground truth (note the scenario difficulty, different fields of view, object dimension, different object color, illumination, camera noise and other objects around the object of interest). Anyway there are also false negatives situations (NT line four) in which the system is unable to classify correctly (according to ground truth) the object detected due to the signal noise, illumination conditions (insufficient, over exposed), partial object view (out of frame, junction with another object) or dimension too low.

VI. Conclusion

In this paper we addressed the problem of automatic video surveillance categorization. We studied the contribution of various modalities and the role of the early fusion mechanisms in increasing the percentage of true positives, while decreasing the false negatives. The design of appropriate descriptors allows to achieve a recall up to 92.17%, that represent an initial promising result. The study was carried out on the proposed Scouter dataset, which provides a realistic scenario for the evaluation of an user tracking algorithm. The classification-based approach seems a suitable perspective to solve multi-instances object retrieval (search) within several video surveillance flows, being capable of learning from very few examples. Good results are achieved in terms of recall measure using selected descriptors or their combination (fusion). However the major drawback is in the power of the classifier to generalize starting from a few training samples. The performance of the system is closely related to the number of frames and the diversity of training sample (different perspective, object size, the quality of the images) and the method tends to fails when too fewer samples are used for training.

A future research direction is to improve the recall even more by adopting (developing) new classifiers or by investigating the benefits of the co-training techniques which are adapted to the situation when very few training samples are available.

Acknowledgment

This work has been supported under ExcelDOC POSDRU/159/1.5/S/132397 (2014-2015) and SCOUTER PN-II-IN-DPST-2012-1-0034 (co-funded by UEFISCDI and UTI Grup Romania, 2013-2015).

References

[1] M. E. Donderler, O. Ulusoy, U. Gudukbay, *Rule-Based Spatio Temporal Query Processing for Video Databases*, VLDB Journal, vol.13, 2004.

[2] C.G.M. Snoek, A.W.M. Smeulders, *Video Search Engines*, IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[3] M. Worring, *Multimedia Analytics: Exploration of Large Multimedia Collections*, on International Workshop on Content-Based Multimedia Indexing, 2012.

[4] M. Shah, O. Javed, K. Shafique, *Automated visual surveillance in realistic scenarios*, IEEE MultiMedia, vol. 14(1), pp. 30-39, 2007.

[5] B. Benfold, I. Reid, *Stable multi-target tracking in real-time surveillance video*, In Computer Vision and Pattern Recognition (CVPR), pp. 3457-3464, 2011.

[6] V. Reilly, H. Idrees, M. Shah, *Detection and tracking of large number of targets in wide area surveillance*, in Computer Vision ECCV, pp. 186-199, 2010.

[7] J.-C. Tai, S.-T. Tseng, C.-P. Lin, K.-T. Song, *Real-Time Image Tracking for Automatic Traffic Monitoring and Enforcement Applications*, in Image and Vision Computing, 22(6), pp. 485-501, 2003.

[8] B. Ionescu, C. Rasche, C. Vertan, P. Lambert, *A Contour-Color-Action Approach to Automatic Classification of Several Common Video Genres*, in Springer-Verlag LNCS - Lecture Notes in Computer Science, Eds. M. Detyniecki, P. Knees, A. Nurnberger, M. Schedl and S. Stober, vol. 6817, pag. 74-88, 2011.

[9] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, *Content-Based Image Retrieval at the End of the Early Years*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, nr. 12, pag. 1349-1380, 2000.

[10] Y. Mingqiang, K. Kidiyo, R. Joseph, *A Survey of Shape Feature Extraction Techniques*, in Pattern Recognition, pag. 43-90, 2008.

[11] M. S. Ryoo, J. K. Aggarwal, *Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities*, in Computer Vision International Conference, 2009.

[12] P. Knees, T. Pohle, M. Schedl, D. Schnitzer, K. Seyerlehner, G. Widmer, *Augmenting Text-Based Music Retrieval with Audio Similarity*, in International Computer Music Conference, pp. 591-598, 2002.

[13] D. G. Lowe, *Distinctive Image Features from Scale-In variant Keypoints*, in International Journal of Computer Vision, vol 60(2), pp. 91-110, 2004.

[14] J. Stottinger, B. Tudor Goras, N. Sebe, A. Hanbury, *Behavior and properties of spatio-temporal local features under visual transformations*, 2010.

[15] I. Mironica, J. Uijlings, N. Rostamzadeh, B. Ionescu, N. Sebe, *Time matters: capturing variation in time in video using fisher kernels*, in ACM international conference on Multimedia, pp. 701-704, 2013.

[16] C.R. Wren, A. Azarbayejani, T. Darrell, A.P. Pentland, *Pfinder: real-time tracking of the human body*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19 (7), pp. 780785, 1997.

[17] N. Lu, J. Wang, L. Yang, Q. H. Wu, *Motion Detection Based On Accumulative Optical Flow and Double Background Filtering*, in World Congress on Engineering, pp. 602-607, 2007.

[18] A. Bovik, *The essential guide to video processing*, Elsevier Inc, 2009.

[19] O. Ludwig, D. Delgado, V. Goncalves, U. Nunes, *Trainable Classifier-Fusion Schemes: An Application To Pedestrian Detection*, IEEE Int. Conf. On Intelligent Transportation Systems, vol. 1, pp. 432-437, 2009.

[20] J. Van De Weijer, C. Schmid, J. Verbeek, *Learning color names from real-world images*, in Computer Vision and Pattern Recognition, 2007.

[21] M. Stricker, M. Orengo, *Similarity of color images*, in SPIE Conference on Storage and Retrieval for Image and Video Databases, 1995.

[22] T. Ojala, M. Pietikinen, D. Harwood, *Performance evaluation of texture measures with classification based on Kullback discrimination of distributions*, in IAPR International Conference on Pattern Recognition, vol. 1, pp. 582 - 585, 1994.

[23] C. G. M. Snoek, M. Worring, A. W. M. Smeulders *Early versus late fusion in semantic video analysis*, ACM Multimedia, 2005.

[24] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, *The WEKA Data Mining Software: An Update*, in SIGKDD Explorations, vol. 11(1), 2009.

[25] Harry Zhang, *The Optimality of Naive Bayes*, in AAAI Press, 2004.

[26] K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft, *When Is Nearest Neighbor Meaningful?*, in Database Theory ICDT Lecture Notes in Computer Science, 1999.

[27] J.R. Quinlan, *Induction of Decision Trees*, in Machine Learning, vol. 1, pp. 81-106, 1986.

[28] L. Breiman, *Random forests*, in Machine learning, vol. 45(1), pp. 5-32, 2001.

[29] V.N. Vapnik, *Statistical Learning Theory*, New York: John Wiley & Sons, 1998.

[30] O. Chapelle, *Training a Support Vector Machine in the Primal*, in Neural Computation, MIT Press, pp. 1155-1178, 2007.

[31] C. Schuldt, I. Laptev, B. Caputo, *Recognizing human actions: A local SVM approach*, in ICPR, 2004.

[32] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, *Actions as Space-Time Shapes* in PAMI, vol. 29(12), pp. 22472253, 2007.

Figure 6: Examples of system classification responses: first three lines represent examples in which the object found by the system are correctly identified according to the ground truth, and the fourth line provides false negative examples.

[33] R. B. Fisher. The PETS04 Surveillance Ground-Truth Data Sets. 2004.

[34] B. Ionescu, A.-L. Radu, M. Menndez, H. Mller, A. Popescu, B. Loni, *Div400: A Social Image Retrieval Result Diversification Dataset*, in ACM Multimedia Systems - MMSys2014, 2014.

[35] G. Hripcsak, A. Rothschild, *Agreement, the f-measure, and reliability in information retrieval*, in Journal of the American Medical Informatics Association, vol. 12(3), pp. 296298, 2005.