PEOPLE INSTANCE RETRIEVAL FROM HIGHLY CHALLENGING VIDEO SURVEILLANCE REAL-WORLD FOOTAGE

Cătălin Alexandru MITREA[1], Ionuț MIRONICA[2], Bogdan IONESCU[3]

This paper is addressing the task of automated multiple-instance people retrieval from video surveillance footage. Such "real-world" datasets are rising particular issues in terms of low image quality, multiple image perspectives, variable lighting conditions and most distinctly, the lack of training samples. A proposed classification-based method is adapted for experiments on two public datasets. Also, comprehensive state-of-the-art descriptors and decisioners pairs are explored and evaluated in terms of average F2 score. Results denote promising performance while the training frames are reduced consistently to one instance.

Keywords: multiple-instance retrieval, feature extraction, low sampling classification, automated video surveillance

1. Introduction

Consistent threats to public and infrastructure safety and high acceleration of urbanization, are contributing to the exponential increase to the number of video surveillance cameras (e.g., it is estimated there are more than 4 mil [4] CCTV cameras deployed in UK, half a mil. only in London). One major drawback and limitation of these video surveillance systems is the absence of an efficient data processing and object retrieval system. For instance, once a human is labeled as a possible target the existing approaches provides poor tracking capabilities of the subjects during video footage, like finding appearances of a possible burglar on the entire database. This task is conducted manually by human operators, many times being highly time consuming and sometimes inefficient. Video analysis of footage acquired from security (CCTV) video camera sequences is a less well studied field and demands adaptation of current established methods to

[1] PhD, LAPI Lab, University "Politehnica" of Bucharest, 061971, Romania, e-mail: camitrea@gmail.com | catalin.mitrea@uti.eu.com
[2] Assist. Prof., PhD, LAPI Lab, University "Politehnica" of Bucharest, 061971, Romania, e-mail: imironica@imag.pub.ro.
[3] Assoc. Prof., PhD, LAPI Lab, University "Politehnica" of Bucharest, 061971, Romania, e-mail: bionescu@imag.pub.ro.

[4] https://en.wikipedia.org/wiki/Mass_surveillance_in_the_United_Kingdom

accommodate to new challenges that are generated. The quality, lighting levels and perspectives of images from each video camera also varies considerably, and this inconsistency can cause difficulties. Images are often low resolution with poor color clarity and have little discriminative or representative texture and shape definition. On such premises, a performing automated multiple-instance people retrieval from large video surveillance databases becomes essential.

In this article are investigated and evaluated confirmed techniques from the literature for feature extraction. A system for content-based object retrieving task is proposed and evaluated. The novelty of our work is in exploiting these concepts while copying with the lack of training samples and low image quality on which established approaches denote usually lower results.

The experiments are conducted on two standard datasets. We obtained promising results in terms of F2-Score which we believe will contribute to the understanding of specialized algorithms using reduced number of samples and further to alleviate police investigations efforts while coping with offenders and crime prevention.

The rest of the article has the following structure. Section 2 investigates existing methods and advances in literature which are related with our task. Section 3 presents the proposed system architecture to conduct the investigations. Experimental validation and discussion of the results is presented in Section 4 while Section 5 concludes the article.

2. Related Work

Manipulation and automated processing of video surveillance footage remains an open issue that demands innovation and adaptation of current established “academic” algorithms and technologies to cope with main drawbacks and issues (e.g., multiple sources and perspective, noise generated by operating conditions, object small sizes with imperceptible movement, etc). Current research trends and directions include content-based multiple instance object detection and classification (e.g., people, vehicle, and abandoned luggage), behavior and event recognition. For object and pattern recognition tasks, most of methods revolves around low-level feature extraction, a consistent number of feature retrieval algorithms being proposed as, e.g., color [1], texture [2], shape [3], or the popular feature point descriptors which are combined with Bag-of-Words techniques [4-5]. Most of these algorithms' efficiency is driven, at some degree, by robustness to rotation, change of scale, and signal perturbations. Generally, all methods include edge, corner, blob and region detectors and it assume that features detected from sample image (video frame) should remain unchanged under image space transformations, which would enable proper matching of images taken at different views and at different points in time. Other approaches are investigated are addressing the level of effective decision making. At this point, the decisions are usually powered by classifiers. Some of them are investigating methods of automatic pre-processing and refinement of input data in order to leverage classifier accuracy [6] and most of them are focusing on described methods on parameters tuning in order to cope with effective training and data noises [7-9].

However, they are far of being effective for real world scenarios (as video surveillance). A consistent drawback of the current approaches (especially the reasoning system) is the reduced generalization power as usually the available samples to train are low or unavailable. The current research focuses mainly on addressing the performance of the classification process by adopting low complexity and fast predictors. With respect to the aforementioned limitations and drawbacks, the main goal of this article is to investigate the performance of the system by adapting new state-of-the-art video descriptors and classifiers while investigating and exploring the limits of the system as the number of training samples are gradually reduced to the unit.

Findings and output of this research are contributed to the performing understanding of the specialized pairs of feature extractors & classification algorithms adapted for automated content-based search in data sets acquired from video surveillance environments.

3. System Architecture

Briefly the system is composed of two layers: Samples selection and training layer, and secondly, the object retrieval and prediction (see Fig. 1). On the first layer the operator selects a region of interest (ROI) of the object/human to be searched (to be found in the database) then the system trains a set of classifiers and saves the models. This step resembles with defining and making the query to the system. Based on the user's "interrogation", on the next layer, the system automatically searches in the entire database all the instances of the object/human to be found. The second layer is composed mainly of three different processing modules: object extraction, feature extraction and prediction module. First module is employed for detection and extraction task of the objects from the video frames and is based on motion information (a background subtraction algorithm powered by a Gaussian-mixture kernel while generated tracks are estimated based on a Kalman filter [24] algorithm). This method has been proven to be robust and suitable for our task. Secondly, a set of state-of-the-art feature extractors are selected and adapted, such as color, texture, shape, and feature points-based. Because of the high amount of key-points which can be generated by latter methods, we have employed PCA to reduce dimension of the input data. Finally, on the last module a set of classifiers (decisioners) are used to automatically label new data and to return the matching results to the operator. This final step represents the output of the queries generated on first layer, which are depicted in the form of images containing framed instances of the object/human obtained. The method can be adapted also for stilled objects but for this particular case the motion detection should be replaced with uniform scanning of the image or with a key-points based detector (these approaches are out the scope of this article).

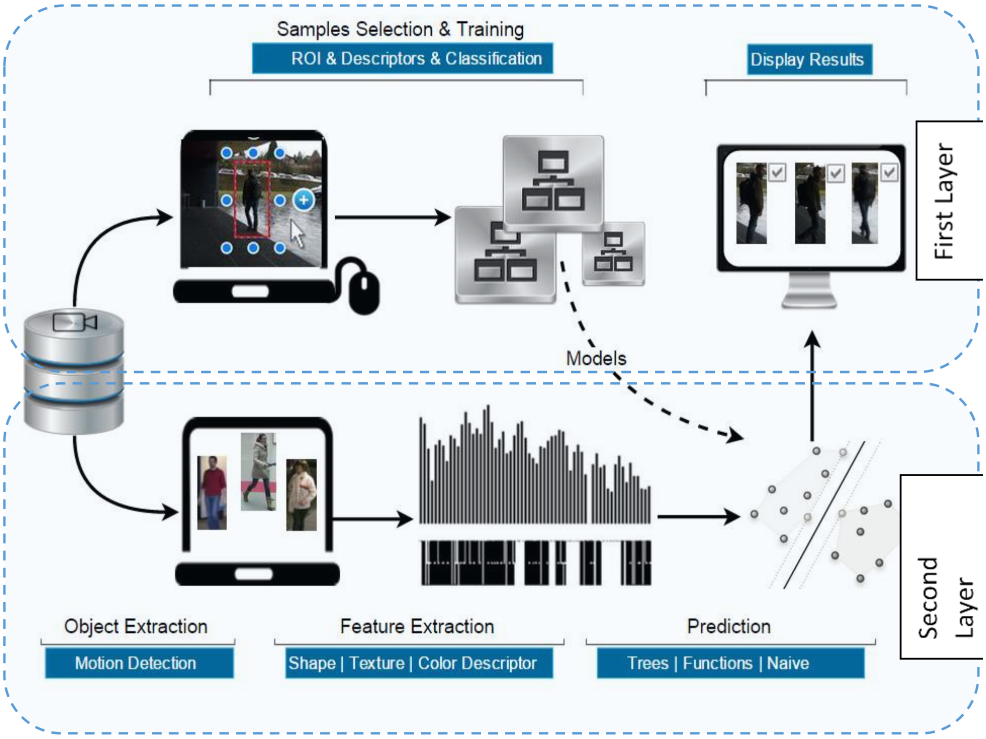

Fig 1. System architecture composed of two layers: Sample Selection & Training layer and secondly, the Object Retrieval & Prediction layer. Classifiers models obtained on first layer are employed for reasoning task on the second layer

3.1 Feature extraction

Competitive results have been obtained using some of these descriptors on other surveillance datasets. Main idea is that some algorithms approaches tend to account for different information, further providing complementary discriminative power to cope with our task. A basic description is made in the following for each feature extraction algorithm:

- *SIFT* (Scale Invariant Feature Transform) [10] – is a key point based feature detection and extraction techniques. Such points are calculated on the premises to encode a distinct pattern which should be "stable" on some degrees on image transformation (translation, rotation and scaling), and less robust to geometrical transformations. Such key-points are encoded on a 128-vector dimension;

- *SURF* (Speeded Up Robust Features) [11] – is a key point based feature detection and extraction techniques. Both as SIFT, SURF interest points are first detected and then descriptors are used for feature vector extraction at each of those interest points. Unlike SIFT which makes use of Difference of Gaussians (DoG) filter, SURF makes use of Hessian-matrix approximation of integral image for localization of interest points thereby reducing computation time. Resulted features vector has a 64 length. To reduce the high number of feature points a PCA is applied for both SIFT and SURF algorithms;
- *HOG* (Histogram of Oriented Gradients) [12] – the algorithm counts occurrences of gradient orientation in localized portions of an image, computed on a dense grid of uniformly spaced cells which are further composed and combined to generate the final vector of 81 features;
- *LBP* (Local Binary Patterns) [13] – represents a particular case of the texture spectrum model, based on a simple texture operator which labels the pixels on an image cell by thresholding the neighborhood of each pixel and outputting the result as a binary number. The resulted feature vector has a 256 length;
- *CM* (Color Moments) [14]– characterize color distribution in an image in the same way that central moments uniquely describe a probability distribution and is based on three central moments of an image’s color distribution: mean, standard deviation and skewness. The image is divided in an *n x n* grid, and a color moment descriptor is computed for each cell. Final outputted vector has 225 features;
- *CN* (Color Naming histograms) [15] – 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”, therefore the resulted vector has 11 dimensions;
- *CSD* (Color Structures Descriptor) [16] – is based on color structure histogram (a generalization of the color histograms) to encode information about the spatial structure of colors in an image as well as their frequency of occurrence. The resulted vector has a 32 length;
- *HK* (Haralik features) [17] – is a texture based descriptor and is powered on a matrix that is defined over an image to be the distribution of co-occurring values at a given offset. The final vector has 11 features;
- *FUSION* – represents an *early* fusion method of all the above features extraction algorithms mentioned. The final vector has a length of 898 values.

3.2 Classification

Following established classifiers are adapted and some tuning parameters are indicated to cope with addressed task, being selected as some of them are obtaining good results in the literature in terms of training speed and generalization power using reduced training samples.

- *KNN* (Nearest Neighbor) [18] known as lazy learners, are powered by an instance based learning where the kernel function is approximated locally. The input is classified by taking a majority vote of the *K* closest training records across the dataset. In this work, we have selected K = 1, 3, 5. Nearest neighbor search algorithm uses a linear function based on a Euclidean distance while training instances are dropped in a first-in-last-out manner;
- *RandomForest* (Random Forests) [19] relatively very popular, it consists mainly in an ensemble learning method created by adding a multitude of decision trees on training process and outputting the class that is the mode of the classes resulted from 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. For current work, we have selected 10 as the number of trees to be build and one seed for random number generator. The maximum depth of the trees is set for unlimited;
- *DecisionTrees* [20] uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. Decision trees compares competing alternatives and assign values to those alternatives by combining uncertainties. For current work, we have selected the confidence threshold for pruning at 0.25 and a two-minimum number of instances per leaf while performing subtree raising;
- *NaiveBayes* represents a classification algorithm based on Bayes rule and assumes that all features from feature descriptor are conditionally independent of one another [21]. Naive Bayes classifier requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. To process numeric attributes, we use a normal distribution density estimator without supervised discretization;
- *SVM* (Support Vector Machines) [22] are supervised learning models with associated learning algorithms that analyze data and recognize patterns. It is usually represented by a neural networks with few layers architecture that constructs a hyper plane or set of hyper planes in a high dimensional space. For this approach, we used three types of SVM kernels: a fast-linear kernel, a Radial Basis Function (RBF) nonlinear kernel and Sequential Minimal Optimization kernel (SMO) [23]. While linear SVMs are very fast in both training and testing, SVMs with non-liner kernel is more accurate in many classification tasks. SMO approach is chosen usually for fast training and low number of training samples,

being most suitable for real-time systems. In our work, the RFB kernel function is setup to use gamma dimension of being equal to 1 divided by the no. of features.

4. Experimental results

4.1 Datasets

Evaluation is made on two standard video surveillance datasets, SCOUTER[5] and PEVID[6]. First dataset contains 3 sets, each of 10 videos recorded on different locations (a total of 30 videos) and acquired on different camera perspectives, both indoor and outdoor setups and denoting variable lighting conditions. The annotations are made for two distinct scenarios (two people) which appear on all videos. The total number of labeled frames is around 36.000. In addition to the dataset, a second dataset (PEVID) is used to conduct and to validate further the experiments. PEVID is composed of 21 video clips and approximately 17,000 manually annotated frames (recorded at 25fps and 1080p). It is comprised on 14 scenarios (14 different people that appear in some videos). Both datasets selected are rising particular video surveillance challenges due to the changing perspectives from one camera to another, variable lighting conditions and multiple variations of subject to be found (summing a total of 16 scenarios annotated). Some samples from SCOUTER and PEVID databases are depicted in Fig. 2. For training, we have used a max number of 120 sample (60 for True class, respectively 60 for False one) while the rest of remaining frames are used for

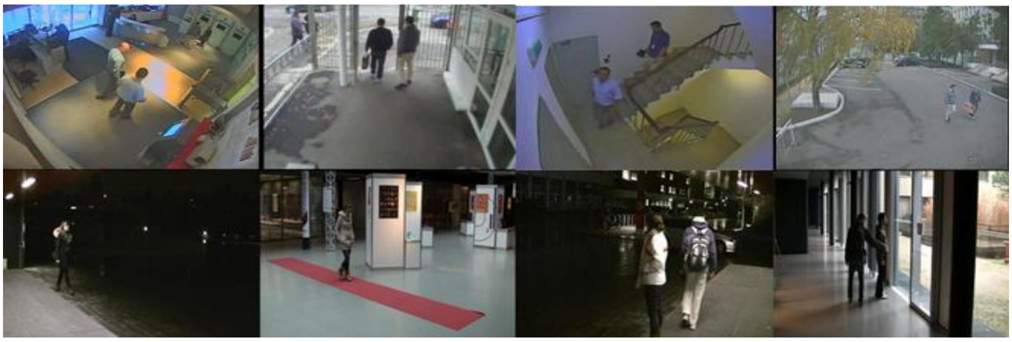

Fig 2. Samples from SCOUTER (first row) and PEVID (second row) video databases. Note the indoor and outdoor setups, different scaling and images quality

testing (ca. 53000 images).

The impact of the number of samples on classifiers accuracy is also investigated. In current work, we evaluate the power of classifier to generalize while the set is reduced progressively to 50, 25, 15, 10, 5, 3, 2, and 1 sample. These values are

[5] http://uti.eu.com/pncd-scouter/rezultate-en.html

[6] http://mmspg.epfl.ch/pevid-hd

selected as in common surveillance systems are usually associated with the typical recording frame rate. Most frequently, for bandwidth and storage considerations, frame rate is selected low (less than 15 fps). For current task, this setup should offer to the user at least one or two seconds (or one frame) from which to identity the object to be found. Considering databases proprieties (image quality & scenarios difficulties), by reducing the training frames to unit we aim to identify the *limit* at which the algorithms are unable to perform effective, results resembling more of random behavior.

4.2 System evaluation

To evaluate the system, we took into consideration some automated video surveillance specific tasks requirements. Usually in a surveillance-based object retrieval task it is more important to retrieve most of the relevant instances than to return more relevant results than irrelevant. Therefore, we consider *Recall* (accounts for the non-detections) as weighting higher than *Precision* (accounts for the number of false positives), i.e., we are interested in retrieving all of the existing instances while the number of false positives has a lower impact. F_2 - Score is computed as:

$$F_2 = 5\frac{PrecisionRecall}{4Precision+Recall} \tag{1}$$

where *Precision* is computed as *TP / (TP+FP)*, *Recall* is *TP / (TP+FN)* and TP are the True Positives (correct detections), FP counts for False Positives and FN represents False Negatives. While the number of training samples is reduced progressively, both *True* and *False* class sets are randomly selected (from the max. 60 samples available), therefore it is more properly to average the *F2 Score* on a number of "*n*" runs. This should assure that most of the diversity from the available samples are took in consideration on the final result (in this work we have selected *n = 10*). On these assumptions, the final *F2 Average Score* is represented by the equation:

$$Ave\ F_2 = \frac{\sum_{k=1}^{n}(F_2)_k}{n} \tag{2}$$

where *n* is the number of runs on each sample scenario (a total no. of 16 scenarios).

In Fig. 3 are depicted the results on selected video descriptors – classifiers pairs obtained on SCOUTER and PEVID datasets, while the number of training frames are reduced progressively to unit. Some interesting findings are:

- On average results obtained on PEVID dataset are generally higher than the results obtained on SCOUTER dataset with an average around 9%. One reason would be the image quality and resolution of PEVID which is superior to the SCOUTER one.

- While the number of training samples drops down to unit results on SCOUTER are decreasing steadily and close relating to the number of training samples. This is not the case of PEVID where the performance is decreasing almost insignificant between 25 and 5 training samples cases.
- At *one* single training sample, results obtain on PEVID dataset are higher than 25% while in SCOUTER case are not crossing this threshold (with an average around 20%).

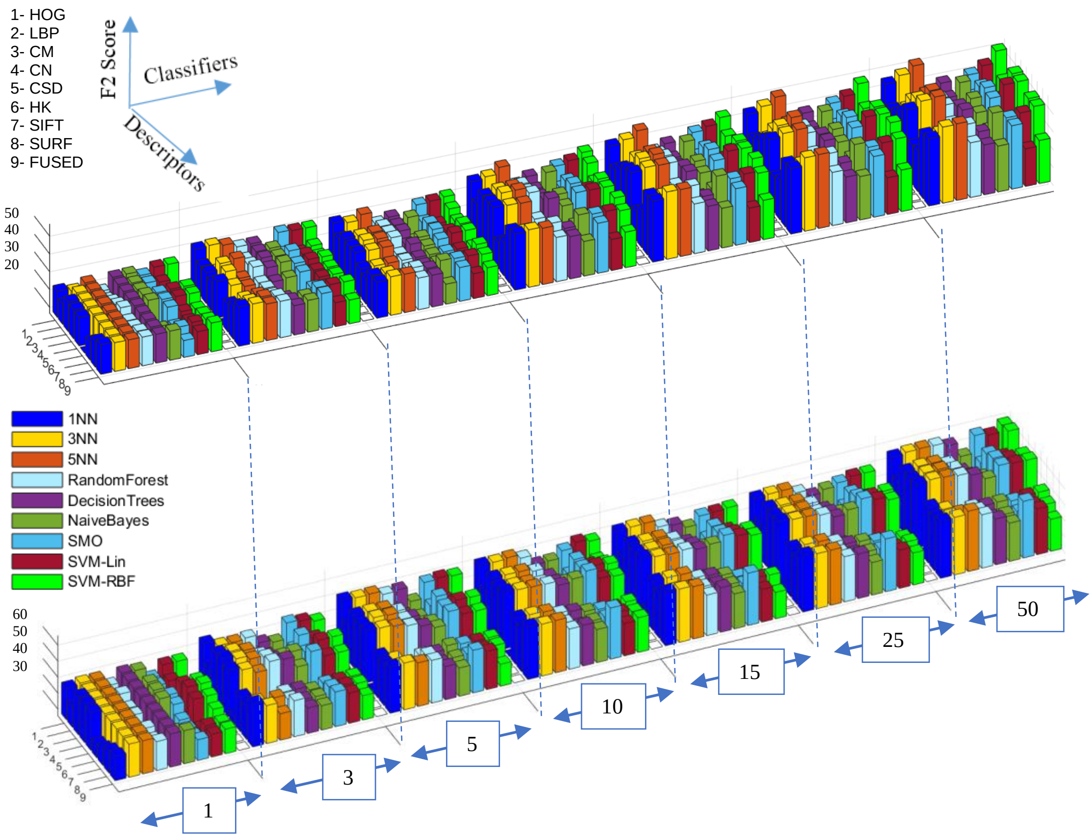

Fig 3. Average F2 Score (%) for each classifier & descriptor pair performed at each training set (staring from 1, 3, 5, 10, 15, 25 to 50 samples). On top are the results on Scouter while on bottom the ones for PEVID dataset

- Overall best results are obtained in terms of *F2 Average Score* by LBP and HOG descriptors combined with non-linear SMV and KNN clsifiers while the lowest results are obtained by *Haralick* features & *NaïveBayes* pairs.
- If in SCOUTER dataset color descriptors performs low in PEVID dataset it denotes better performance (the resolution and image quality are higher).

In Table 1 and Table 2 are the results obtained for SCOUTER and PEVID datasets while using 60 samples used for training *True* class and 60 samples for *False* class (with bold the best results).

Table 1

F2 Average Score on SCOUTER dataset

F2 Score	1NN	3NN	5NN	Random Forest	Decision Trees	Naïve Bayes	SMO	SVM Lin	SVM RBF
HOG	40.249	46.689	46.033	33.623	32.495	19.371	35.627	41.387	**47.913**
LBP	35.81	36.879	37.128	38.017	39.739	38.061	32.647	46.045	46.995
CM	29.823	32.291	34.189	34.987	15.834	16.643	31.891	24.444	34.195
CN	26.751	32.896	33.563	22.327	21.625	26.292	28.665	24.099	41.325
CSD	36.422	39.023	40.94	20.602	19.234	25.932	33.901	40.074	42.546
HK	31.463	31.463	31.463	31.463	31.463	30.205	31.463	31.469	31.469
SIFT	31.068	31.794	33.456	33.331	31.171	32.842	34.289	33.88	26.482
SURF	36.668	34.424	33.238	36.866	34.23	31.646	33.416	32.698	30.384
FUSED	39.954	41.797	42.479	37.171	34.409	31.522	39.504	38.244	47.241

Table 2

F2 Average Score on PEVID dataset

F2 Score	1NN	3NN	5NN	Random Forest	Decision Trees	Naïve Bayes	SMO	SVM-Lin	SVM-RBF
HOG	66.191	67.906	69.165	58.316	48.641	54.546	60.976	65.227	66.205
LBP	42.198	43.139	43.918	37.544	35.685	39.003	36.407	36.222	45.961
CM	60.332	62.121	62.555	58.207	46.67	47.198	65.137	56.827	48.319
CN	69.445	63.27	53.681	61.071	56.041	43.285	68.767	65.712	68.736
CSD	54.458	64.773	63.836	65.494	63.902	57.493	66.012	66.484	**69.452**
HK	1.9181	1.9181	1.9181	4.4223	41.626	18.595	1.9181	7.9886	7.9886
SIFT	62.474	63.439	63.5	57.438	52.843	48.403	41.977	36.063	36.978
SURF	58.514	57.652	58.485	54.003	51.722	47.293	43.568	42.454	48.057
FUSED	59.767	60.454	69.136	57.015	52.419	42.61	64.15	38.022	26.5

5. Conclusion

In the current work, we have investigating different pairs of algorithms of feature extraction and classification performance on two public databases for the task of multiple-instance people retrieval. The results are evaluated in terms of average F2 score. Also, the impact of training samples on classifiers performance was carried out systematically as the number is reduced consistently to the unit.

In traditional classification-based problems it is assumed that the number of training samples are consistent and available in sufficient number in order to assure that the classifier is able to generalize with good performance each class. The training samples are often insufficient in the case of video surveillance (real-world databases) tasks. Usually the particular object (person) to be found is obtained from just a few frames or a few seconds of footage. Starting from these few examples, to find all instances of the object (human) in the entire database is a very difficult and tricky task.

Thanks to findings of this work, the minimum number of samples required at training is higher for processing tasks involving low image quality datasets (SCOUTER) in comparison to datasets with clear and high image quality (PEVID). SCOUTER scenarios are more complex and difficult to analyze and process due to object obstructions, diversity of objects in scenes (cars and large number of other people) and high changing of human-instance parameters from one footage set to another (e.g. clothes/colors varies considerably from one recording day to another, high shifting of lighting, scaling, etc). On the other hand, PEVID database has lower objects and obstructions in scenes and the human-instances parameters are varying less (even if the total number of scenarios/humans is much higher). These main differences determine performance on PEVID dataset to be higher in general than counterpart SCOUTER dataset.

Future work will address the relevance feedback (RF) based learning techniques which might offer extra help for classifier on dealing with low training sampling.

Acknowledgment

Part of this work was funded under ExcelDOC POSDRU/159/1.5/S/132397 and research grant PN-III-P2-2.1-PED-2016-1065 (agreement 30PED/2017, project SPOTTER).

R E F E R E N C E S

[1]. *R. Khan, J. Weijer, F. Khan, D. Muselet, C. Ducottet and C. Barat*, "Discriminative Color Descriptors", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2866- 2873, 2013.

[2]. *N. Vatani, M. De Deuge, B. Douillard, and S.B. Williams*, "An analysis of monochrome conversions and normalizations on the Local Binary Patterns texture descriptors", IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 831 – 837, 2013.

[3]. *X. Bai, C. Rao and X. Wang,* "Shape Vocabulary: A Robust and Efficient Shape Representation for Shape Matching", IEEE Transactions on Image Processing, pp. 3935 - 3949, 2014.

[4]. *G. Khan, N.Y. McCane and B. Wyvill,* "SIFT and SURF Performance Evaluation against Various Image Deformations on Benchmark Dataset", International Conference on Digital Image Computing Techniques and Applications (DICTA), pp. 501 – 506, 2011.

[5]. *J.R.R. Uijlings, A.W.M. Smeulders and R.J.H. Scha*, "Real-Time Visual Concept Classification", In IEEE Transactions on Multimedia, vol. 12, pp. 665 -681, 2010.

[6]. *T.R. Tavares, G.G. Cabral and S.S Mattos,* "Preprocessing unbalanced data using weighted support vector machines for prediction of heart disease in children", International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2013.

[7]. *O. Chapell, V. Sindhwani and S.S. Keerthi,* "Optimization Techniques for Semi-Supervised Support Vector Machines", Journal of Machine Learning Research, pp. 203-233, 2008.

[8]. *H. Zheng, X. Zhang and H. Guo,* “Using SVM to Learn the Efficient Set in Multiple Objective Discrete Optimization, Sixth International Conference on Fuzzy Systems and Knowledge Discovery, pp. 489 - 493, 2009.
[9]. *C.A. Mitrea, I. Mironica, B. Ionescu and R. Dogaru,* “Multiple instance-based object retrieval in video surveillance: Dataset and evaluation”, Int. Conf. on Intelligent Computer Communication and Processing, ISBN 978-1-4799-6568-7, pp. 171-179.
[10]. *D. G. Lowe,* “Distinctive Image Features from Scale Invariant Keypoints”, International Journal of Computer Vision, vol 60(2), pp. 91-110, 2004.
[11]. *H. Bay, T. Tuytelaars, and L. Van Gool,* "Speeded Up Robust Features", ETH Zurich, Katholieke Universiteit Leuven.
[12]. *L. Hu, W. Liu, B. Li and W. Xing*, “Robust motion detection using histogram of oriented gradients for illumination variations”, Industrial Mechatronics and Automation (ICIMA), 2010 2nd International Conference on , vol. 2, pp. 443 - 447, 2010.
[13]. *D.C. He and L. Wang,* “Texture Unit, Texture Spectrum, and Texture Analysis”, IEEE Transactions on Geoscience and Remote Sensing, vol. 28, pp. 509-512, 1990.
[14]. *M. Stricker and M. Orengo,* “Similarity of color images”, SPIE Conference on Storage and Retrieval for Image and Video Databases, 1995.
[15]. *J. Van De Weijer, C. Schmid and J. Verbeek,* “Learning color names from real-world images”, Computer Vision and Pattern Recognition, 2007.
[16]. *B. S. Manjunath, J.R. Ohm, V.V. Vasudevan and A. Yamada*, “Color and Texture Descriptor, IEEE Transactions on Circuits and Systems for Video Technology, pp. 703-715, 2001.
[17] *Robert M. Haralick*, “Statistical and structural approaches to texture”, Proc. IEEE, vol. 67, no. 5, pp. 786-804, 1979.
[18]. *N. S. Altman,* "An introduction to kernel and nearest-neighbor nonparametric regression". The American Statistician 46 (3), pp.175–185, 1992.
[19]. *Breiman Leo,* "Random Forests". Machine Learning 45 (1), pp. 5–32, 2001.
[20]. *J. R. Quinlan,* "Simplifying decision trees". International Journal of Man-Machine Studies 27 (3), 1987.
[21]. *Russell Stuart*, Artificial Intelligence: A Modern Approach (2nd ed.), Prentice Hall. ISBN 978-0137903955, 1995.
[22]. *C. Cortes and V. Vapnik,* "Support-vector networks", Machine Learning 20 (3), 1995.
[23]. *James Platt*, “Fast Training of Support Vector Machines using Sequential Minimal Optimization”. In Advances in Kernel Methods - Support Vector Learning, 1998.
[24]. *B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla*, Kalman filtering with intermittent observations, IEEE Transactions on Automatic Control, vol: 49, Issue: 9, pp. 1453-1464, 2004.