Ensemble-based Learning Using Few Training Samples for Video Surveillance Scenarios

C.A. Mitrea[1,2] S. Carata[2], B. Ionescu[1], T. Piatrik[3] and M. Ghenescu[2]
[1] Image Processing and Analysis Lab, UPB, Romania
e-mail: bionescu@imag.pub.ro
[2] Softrust Vision Analytics, UTI Grup, Bucharest, 020492, Romania
e-mail: {catalin.mitrea, serban.carata, marian.ghenescu}@uti.eu.com
[3] Multimedia and Vision Group, QMUL, UK
t.piatrik@qmul.ac.uk

***Abstract*— The article targets the task of content-based multiple-instance people retrieval from video surveillance footage. This task is particularly challenging when applied on such datasets as the available samples to train the decisioning system and formulate the query are insufficient (one image, few frames, or seconds of video recording). To cope with these challenges we investigate three established ensemble-based learning techniques, e.g., boosting, bagging and blending (stacking). Such methods are based on a set of procedures employed to train multiple learning algorithms and combine their outputs, while functioning together as a unified system of decision making. The approach was evaluated on two standard datasets (accounting for 16 people searching scenario on ca. 53000 labeled frames). Performance in terms of F2-Score attained promising results while dealing with our current task.**

***Keywords*— Reduced training samples, Ensemble learning, Classification, Multiple instance retrieval, Video Surveillance.**

I. Introduction

The increasing rate of the population, and implicitly, the acceleration of urbanization process, are boosting the emerging and development of contemporary industrial fields such as automatic video surveillance. Authorities are focusing and aiming consistently their actions at increasing public safety by preventing crimes and protecting properties and infrastructures. As communications and storage technologies have developed tremendously, it became relatively easy to acquire video data from all dispersed security CCTV cameras over large areas. Nevertheless, manipulation and processing of such high amount of footage is still a steady challenge (e.g., only HD video surveillance alone, which is inferior in use comparing to currently standard systems, are expected to reach 859 petabytes generated each day by 2017[1]). These "real-world" data are denoting specific processing difficulties as most of the time the images are noisy or with poor discriminative (texture, color, shape) power for coping with content-based object retrieval. The task is user-driven as is deriving from police forensics investigations practice on dealing with human identifications in video footage. For instance, starting from an object of interest (e.g., a burglar) that is labeled on a given anchor image (or a short footage), the system provides usually low retrieving robustness (e.g., searching of all instances during the video records) in order to confidently identify that specific person. Such issues are usually generated by the lack of enough samples to adequately train the reasoning system (usually powered by classifiers) and also to the high variation of object instances on different footage sources (multiple cameras perspective and object sizes, occlusions in high dense scenes, etc). To cope with such issue, consistent efforts are carried out by police representatives for effective analysis of high volumes of video surveillance data. These premises and assumptions are the basis for conducting the current experiments.

In this article we investigate and test three ensemble learning methods that use confirmed techniques from the literature, e.g., boosting, bagging and blending. The novelty of our work is in exploiting these concepts while copying with the lack of training samples, and further minimizing data variance (e.g., results should be less dependent on peculiarities of single training set) and reducing bias (e.g., a combination of classifier might learn more efficiently a concept class than using a single classifier).

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 training 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 II investigates existing methods and advances in literature which are related with our task and places our main contribution. Section III presents the proposed system architecture to conduct the investigations. Experimental validation and discussion of the results is presented in Section IV while Section V concludes the article.

II. Related Work

Content-based multiple-instance object retrieval is still and open issue, especially in real-world applications types as video surveillance. This is to the fact that usually the available samples from which to extract the region of interest (object to be found during the footage) are low or insufficient (e.g.,

[1] http://defensesystems.com/articles/2013/10/10/hd-surveillance-big-data.aspx

one image, few seconds of recording, which are generating in practice ca. few tens of training instances). Decision making systems are usually powered by classifiers which are denoting low generalization power when there are insufficient training samples. Considerable effort has been focused towards various aspects of learning, including data preprocessing and refinement of input samples in order to leverage classifier precision [1]. For instance, authors in [2] is model the relationship between low-level concepts in a framework based on latent SVM. Most of the efforts of classifiers learning enhancement are focusing to parameter optimization during the training process [3][4]. Other researches are focusing on features selection and class balancing [5] for the training process. Feature selection extracts the most important set of attributes for model training, while class imbalance adjustment occurs when the data set is skewed more toward a class. For object and pattern recognition tasks, most of methods revolves around low and mid-level feature extraction [6][7], and their power to retrieve and quantify knowledge optimally, which further should help decision systems for effective generalization during training process, and finally, to increase prediction accuracy on the new input samples. Another approach who has gained more importance recently is the use of ensemble based training techniques [8][9]. Such methods refers to the procedures employed to train multiple learning machines and combine their outputs, treating them as an integer reasoning system for decision making. The main idea is that individual predictors combined appropriately, should have better overall accuracy, on average, than any separate member of the reasoning system.

The current research focuses mainly on adapting ensemble based learning and investigate their performance for content-based multiple-instance object retrieval task, while using as predictors different types of establish classifiers, and starting only from few training samples.

Current paper develops further to the work in [10] by implementing new video descriptors and investigating the impact of ensemble based prediction performance. Outcomes of this research are contributing to specialized training techniques which might be adapted for automated content-based search in large video sets acquired from video surveillance, and implicitly police forensics investigations.

III. Proposed System

The system built to conduct the experiments and review the methods is based on two layers (see Fig. 1). First layer deals with samples selection (used for ensemble based training) and to query generation. Second layer drives the object retrieval processing system and consists of the following main steps. First, the system detects and extracts the objects from footage based on motion (powered by an ensemble of background subtraction and Gaussian mixture models, with each track trajectory estimated by Kalman filter). Secondly, a set of well established feature extractors are employed for color, texture, and shape description and quantification. Finally, on the last step, a set of state-of-the-art classifiers are combined into a ensemble based manner for prediction purpose, to label the input data and return task results to the end-user (e.g., police officer).

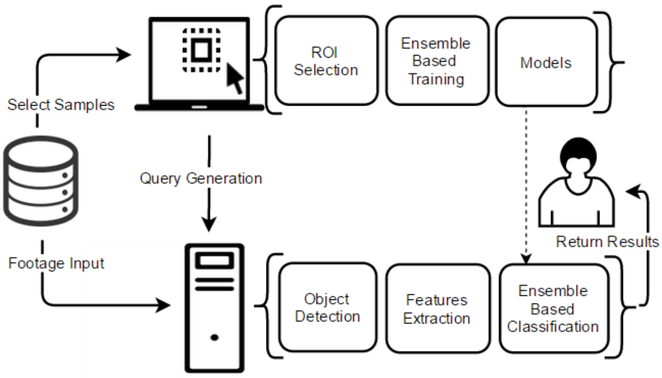

Fig. 1. Proposed System Architecture

Most of the implementation is based on WEKA[2] and other established open source computer vision libraries.

A. Feature Extraction

We have selected a number of nine video descriptors which have been known to obtain competitive results on other surveillance datasets for similar tasks. We use one shape-based descriptor (HOG), two texture-based descriptors (LBP and HK), three color-based descriptors (CM, CN, CSD) and two feature key-points descriptors (SIFT and SURF). 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) [11] is a key points 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). Such key-points are encoded on a 128 vector dimension.

-SURF (Speeded Up Robust Features) [12] is a key points 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, PCA is applied for both SIFT and SURF algorithms.

-HOG (Histogram of Oriented Gradients) [13] the algorithm counts occurrences of gradient orientation in localized portions of an image, computed on a dense grid of uniformly spaced

[2] http://www.cs.waikato.ac.nz/ml/weka/downloading.html

cells which are further composed and combined to generate the final vector of 81 features.

-LBP (Local Binary Patterns) [14] represents a particular case of the texture spectrum model, based of 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) [15] 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 histogram) [16] 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) [17] 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 (Haralick Features) [18] 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 (F) represents a simple summation of all previously presented features. Generated vector is composed by a total number of 898 features.

B. *Classifiers Selection*

It is common knowledge that some classifiers tend to perform better on specific tasks. With respect to that, we have selected a number of nine classifier namely:

-Nearest Neighbor (KNN) [19] 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.

-Random Forest (RF) [20] 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.

-Decision Trees (J48) [21] 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.

-Naive Bayes (NB) represents a classification algorithm based on Bayes rule and assumes that all features from feature descriptor are conditionally independent of one another [22]. Naive Bayes classifier requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification.

-Support Vector Machines (SVM) [23] 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 one, or a set of hyper planes in a high dimensional space. For this approach, we used two types of SVM kernels: a fast linear kernel and a Radial Basis Function (RBF) nonlinear kernel. While linear SVMs are very fast in both training and testing, SVMs with non-liner kernel is more accurate in some classification tasks.

C. *Ensemble Based Training*

We investigate three state-of-the-art ensemble based learning techniques, e.g., boosting, bagging and blending (sometimes named stacking). Main idea is to combine the classifiers selected in previous section for dealing with few samples training, also to minimize data variance (e.g., results should be less dependent on peculiarities of single training set) and to reduce bias (e.g., a combination of classifier might learn a more efficiently a concept class than using a single classifier).

1) Boosting: Represents an ensemble based learning method that starts out by training a classifier, and consequently, other classifiers are gradually added to focus on the instances of the training set that the previous classifier got misclassified. The process of adding classifiers continues until a limit in the number of models or accuracy is reached. The output of the other learning algorithms is combined into a weighted sum that represents the final output of the boosted classifier. In current work we have selected Adaboost [24]. This is an algorithm that combines usually classifiers with poor performance (weak learners), resulting a classifier with much higher performance. AdaBoost is adaptive in the sense that subsequent learners are tunned in favor of those instances misclassified by previous classifiers. AdaBoost is one of the most utilized (boosting) algorithm in the machine learning community.

2) Bagging: Also known as Bootstrap Aggregating technique [25], is an ensemble based learning method that creates distinct samples of the training dataset, for which builds a classifier to be trained for each sample set. The results of these multiple classifiers are then combined (based on majority voting, average, etc.). Main idea is that each training set created should be independent and distinct, giving each classifier during training a different focus and perspective on the class perspective. In order to promote model variance, bagging trains each model in the ensemble using a randomly drawn subset of the training set. Compared with Boosting technique, Bagging should be more noise-tolerant and to estimate better class probability.

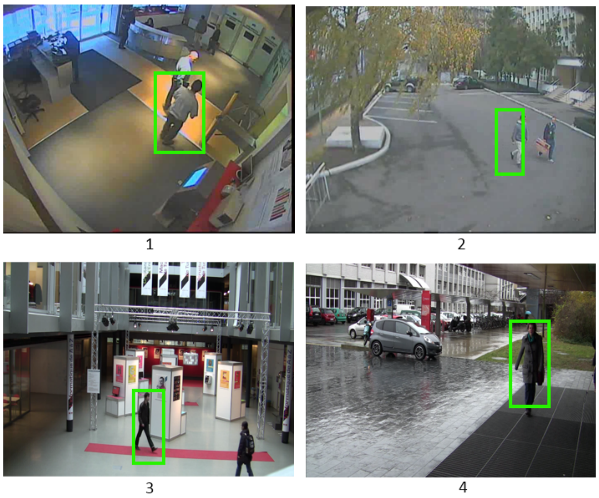

Fig. 2. Samples of SCOUTER (1,2) and PEVID (3,4) databases. Framing depicts correctly retrieved instances, with respect to the GT.

3) Blending: Blending, also known as Stacking [26], represents an ensemble based learning method where multiple different algorithms are prepared on the entire training set and a meta classifier is build to learn how to combine the predictions of each classifier from the set (and further to make accurate predictions on new sets). If an arbitrary combiner meta algorithm is used, then stacking can theoretically represent any of the ensemble techniques described in this article, although in practice, a single-layer logistic regression model is usually used as the combiner. For this work we are adapting as a meta classifier Logistic Regression (refer further with LR acronym) [27] as is a reliable and efficient method to learn how to combine the predictions, being also suited to binary classification tasks.

For the first stacking approach we are investigating one classifier with one meta classifier (logistic regression). This combination generates a total number of 9 pairs (LR and all classifiers from section III-B). For the stacking technique there can be involved more than one classifier. Considering that, the idea is to include into ”blending” process different learning algorithms that presents distinct perspective on the problem and in turn constructs different useful predictions. Regarding their functional and reasoning base structure, for our current work we are using four classifier types: Bayes (1 classifier), Functions (2 classifiers), Lazy (3 classifiers) and Trees (2 classifiers). Using these assumptions and imposing the rule of not pairing two classifier of the same type (e.g., 1KNN combined with 5KNN), it generates for the second stacking approach a total of 23 combinations (see Tabel 1).

Table 1. Classifiers types and combination for second Stacking approach

#	Classifiers — Multi Stacking		Acronym
1	NaiveBayes (Bayes)	Linear SVM (Function)	NB + SVM-Lin
2	NaiveBayes (Bayes)	Non Linear SVM (Function)	NB + SVM-RBF
3	NaiveBayes (Bayes)	1 Nearest Neighbor (Lazy)	NB + 1KNN
4	NaiveBayes (Bayes)	3 Nearest Neighbor (Lazy)	NB + 3KNN
5	NaiveBayes (Bayes)	5 Nearest Neighbor (Lazy)	NB + 5KNN
6	NaiveBayes (Bayes)	Decision Trees (Tree)	NB + J48
7	NaiveBayes (Bayes)	Random Forest (Tree)	NB + RF
8	Linear SVM (Function)	1 Nearest Neighbor (Lazy)	SVM-Lin + 1KNN
9	Linear SVM (Function)	3 Nearest Neighbor (Lazy)	SVM-Lin + 3KNN
10	Linear SVM (Function)	5 Nearest Neighbor (Lazy)	SVM-Lin + 5KNN
11	Linear SVM (Function)	Decision Trees (Tree)	SVM-Lin + J48
12	Linear SVM (Function)	Random Forest (Tree)	SVM-Lin + RF
13	Non Linear SVM (Function)	1 Nearest Neighbor (Lazy)	SVM-RBF + 1KNN
14	Non Linear SVM (Function)	3 Nearest Neighbor (Lazy)	SVM-RBF + 3KNN
15	Non Linear SVM (Function)	5 Nearest Neighbor (Lazy)	SVM-RBF + 5KNN
16	Non Linear SVM (Function)	Decision Trees (Tree)	SVM-RBF + J48
17	Non Linear SVM (Function)	Random Forest (Tree)	SVM-RBF + RF
18	Decision Trees (Tree)	1 Nearest Neighbor (Lazy)	J48 + 1KNN
19	Decision Trees (Tree)	3 Nearest Neighbor (Lazy)	J48 + 3KNN
20	Decision Trees (Tree)	5 Nearest Neighbor (Lazy)	J48 + 5KNN
21	Random Forest (Tree)	1 Nearest Neighbor (Lazy)	RF + 1KNN
22	Random Forest (Tree)	3 Nearest Neighbor (Lazy)	RF + 3KNN
23	Random Forest (Tree)	5 Nearest Neighbor (Lazy)	RF + 5KNN

IV. Experimental Results

Evaluation is made on two standard video surveillance datasets, SCOUTER[3] and PEVID[4]. First dataset contains 3 sets (samples in Fig. 2), each of 10 videos recorded on different locations (a total of 30 videos) and acquired on different camera perspectives, 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. Second dataset consists of 21 recordings and ca. 17.000 manually annotated frames for 14 scenarios (distinctive humans). Selected datasets are rising particular video surveillance challenges due to the diversity of footages - high changing perspectives from one security camera to another (multiple source CCTV cameras), different weather conditions and large variations of the subject to be found (summing a total of 16 scenarios involving people are labeled). For training we have used a total of 120 samples (60 for True class, respectively 60 for False one) while the rest of remaining frames are used for testing (ca. 53000 images). Further we are referring to SCOUTER database using Roman numeral (I), respectively (II) for PEVID dataset.

To assess performance we use the standard F-Score, thus:

$$F_\beta = (1+\beta^2)\frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \qquad (1)$$

where *Precision* is computed as TP/(TP+FP), *Recall* is TP/(TP+FN) and TP are the True Positives (correct detections), FP represents the False Positives and FN are the False Negatives. A high Precision denotes that our system returned substantially more relevant results than irrelevant, while a high Recall means that our system returned most of the relevant results. For an automated surveillance task, the decisioning system should produce no false-negatives and a minimal number of false-positives. For this reason, we consider recall as weighting more than precision when calculating, thus $\beta = 2$.

In the following are shown the results obtained by using the three ensemble based learning techniques.

In Table 2 are presented the results in terms of F2-Score for Boosting technique. Best results are obtained using linear SVM and CSD descriptor (72.1 %) while the lowest results

[3] http://uti.eu.com/pncd-scouter/rezultate-en.html
[4] http://mmspg.epfl.ch/pevid-hd

Table 2. F2-Score performance results using Boosting

F2-Score		HOG	LBP	CM	CN	CSD	HK	SIFT	SURF	F
1KNN	I	39.2	36.8	29.8	30.8	36.4	35.5	34.1	33.7	41.0
	II	66.2	42.2	58.3	69.4	65.5	1.9	62.5	58.5	67.8
3KNN	I	40.9	34.6	30.8	32.7	39.0	31.5	31.3	35.2	41.1
	II	68.0	42.6	58.1	67.2	64.8	1.9	61.6	57.7	67.5
5KNN	I	40.4	32.9	30.7	31.7	38.9	31.5	33.3	33.7	41.0
	II	68.6	40.0	61.7	69.4	52.4	1.9	59.8	53.9	68.2
J48	I	35.5	39.5	34.8	29.1	29.4	31.5	33.5	36.7	37.6
	II	56.9	38.1	59.7	61.5	65.8	5.3	57.2	56.1	55.6
NB	I	32.2	40.2	22.4	31.9	22.0	31.5	30.3	35.5	34.1
	II	51.7	34.4	52.5	57.0	65.9	41.6	56.1	52.9	57.5
RF	I	13.6	38.5	15.5	28.5	29.1	1.9	32.7	35.1	32.0
	II	50.4	39.0	47.9	44.6	70.2	18.6	55.1	56.2	39.9
SVM Lin	I	30.2	24.9	25.0	12.6	40.3	26.5	34.0	33.9	27.4
	II	59.3	37.7	57.5	60.8	**72.1**	46.6	38.2	48.1	41.0
SVM RBF	I	48.0	41.5	30.0	32.9	31.5	26.5	41.5	29.6	41.5
	II	70.6	29.8	48.8	68.2	47.2	46.6	28.9	53.2	40.8

Table 3. F2-Score performance results using Bagging

F2-Score		HOG	LBP	CM	CN	CSD	HK	SIFT	SURF	F
1KNN	I	44.0	35.2	29.6	27.5	37.7	31.5	30.9	36.7	40.8
	II	66.4	43.2	70.4	71.2	73.6	1.9	63.3	57.7	70.9
3KNN	I	47.3	36.1	31.6	33.0	40.0	31.5	32.0	34.6	42.5
	II	68.3	44.3	72.3	71.5	72.9	1.9	64.4	57.7	70.1
5KNN	I	48.0	36.4	32.4	33.7	41.1	31.5	33.6	33.2	42.5
	II	69.6	45.3	72.7	72.2	**74.1**	1.9	63.6	57.9	68.7
J48	I	31.1	38.2	32.8	23.5	23.7	31.5	31.8	36.7	37.1
	II	57.5	36.5	61.3	62.9	66.2	5.0	57.7	53.7	55.7
NB	I	36.7	38.5	21.7	28.6	24.2	31.5	31.4	34.6	36.0
	II	53.7	37.8	55.1	58.9	68.1	41.6	57.1	53.9	61.3
RF	I	18.9	38.1	16.9	28.0	28.1	1.9	32.3	31.8	33.7
	II	53.8	39.9	46.1	44.6	70.1	18.6	46.4	43.9	42.5
SVM Lin	I	41.9	25.6	26.6	20.4	40.1	1.9	33.6	33.3	27.9
	II	67.0	41.6	60.2	67.4	70.8	39.0	39.1	44.8	42.6
SVM RBF	I	47.7	31.5	35.6	41.3	43.1	1.9	26.5	35.6	31.5
	II	65.7	28.9	51.5	69.8	53.8	39.0	37.9	49.2	37.1

Table 4. F2-Score performance results using simple Stacking

F2-Score		HOG	LBP	CM	CN	CSD	HK	SIFT	SURF	F
1KNN	I	40.2	35.8	29.8	26.8	36.4	31.5	31.1	36.7	40.0
	II	66.2	42.2	62.3	69.4	60.5	1.9	62.5	58.5	67.8
3KNN	I	35.4	30.9	23.8	23.4	31.9	31.5	31.8	34.6	38.6
	II	62.6	31.5	68.9	66.5	69.6	1.9	55.9	54.2	68.2
5KNN	I	31.1	33.5	28.3	31.1	33.8	31.5	32.9	34.9	40.9
	II	63.3	36.0	67.9	68.3	67.4	1.9	57.6	53.7	64.2
J48	I	27.7	37.8	19.6	22.7	17.7	31.5	32.0	37.1	24.4
	II	46.1	30.2	50.7	56.8	55.0	6.4	51.2	50.0	43.0
NB	I	32.5	39.7	15.8	21.6	19.2	31.5	31.2	34.2	34.4
	II	48.6	35.7	46.7	56.0	63.9	41.5	52.8	51.7	52.4
RF	I	19.2	38.1	16.6	26.3	25.8	1.9	32.7	31.7	31.5
	II	54.5	27.3	48.3	43.3	69.7	18.6	48.4	47.2	42.7
SVM Lin	I	41.4	25.0	24.4	24.1	40.1	26.5	33.9	32.7	28.2
	II	65.2	36.2	56.8	65.7	66.5	18.9	36.7	42.5	38.0
SVM RBF	I	47.9	1.7	34.2	41.3	42.5	26.5	41.5	30.4	41.5
	II	**69.4**	48.9	48.3	68.7	63.5	22.8	52.3	48.1	56.5

Table 5. F2-Score results without using ensemble based learning technique

F2 Score		HOG	LBP	CM	CN	CSD	HK	SIFT	SURF	F
1KNN	I	38.2	34.8	29.5	25.8	36.7	30.5	31.1	36.7	39.0
	II	64.2	41.2	59.3	69.4	53.5	1.9	62.5	58.5	58.8
3KNN	I	46.7	36.9	32.3	32.9	39.0	31.5	31.8	34.4	41.8
	II	67.9	43.1	62.1	63.3	64.8	1.9	63.4	57.7	60.5
5KNN	I	46.0	37.1	34.2	33.6	40.9	31.5	33.5	33.2	42.5
	II	69.2	43.9	62.6	53.7	63.8	1.9	63.5	58.5	69.1
J48	I	32.5	39.7	15.8	21.6	19.2	31.5	31.2	34.2	34.4
	II	48.6	35.7	46.7	56.0	63.9	41.6	52.8	51.7	52.4
NB	I	19.4	38.1	16.6	26.3	25.9	30.2	32.8	31.6	31.5
	II	54.5	39.0	47.2	43.3	57.5	18.6	48.4	47.3	42.6
RF	I	33.6	38.0	35.0	22.3	20.6	31.5	33.3	36.9	37.2
	II	58.3	37.5	58.2	61.1	65.5	4.4	57.4	54.0	57.0
SVM Lin	I	41.4	46.0	24.4	24.1	40.1	31.5	33.9	32.7	38.2
	II	65.2	36.2	56.8	65.7	66.5	8.0	36.1	42.5	38.0
SVM RBF	I	47.9	47.0	34.2	41.3	42.5	31.5	26.5	30.4	47.2
	II	66.2	46.0	48.3	68.7	**69.5**	8.0	37.0	48.1	26.5

is obtained by HK with several classifiers. In general for Boosting technique we have observed that functions and lazy types classifiers performs better compared to the rest.

Table 3 contains the results obtained using ensemble based Bagging technique. Best result is obtained using CSD and 5KNN pair (74.1%) while the lowest result is obtained by HK on SCOUTER dataset (1.9%). Though HK denotes lowest performance overall, in some combinations it obtains comparable results with the rest combinations.

In Table 4 are the results obtained while investigating simple Stacking (one meta classifier and one classifier for learning). In this case best result is obtained using HOG descriptor while paired with nonlinear SVM (69.4%).

In Table 6 are presented the results in terms of F2-Score for the second Stacking technique (one meta classifier and two classifier used for class generalization). Best results here is obtained using SCD and linear SVM-Lin + 1KNN pair classifiers (75.2%). This is the best result overall. We obtained lowest results using HK video descriptor (on some cases few percents, if usually combined with lazy or trees types classifiers).

Table 5 contains the results obtained without ensemble based learning technique (e.g., training classifiers independently). It can be observed that best result is achieved by CSD and nonlinear kernel SVM-RBF pair (69.5%) which is inferior to multi Stacking approach (75.2%).

Key-points approaches are not obtaining the best results. This is to the fact that usually the instance of the object retrieved (humans) can have very low dimensions (50 x 50 pixels) and even low image quality (especially on Scouter dataset). At these dimensions, SIFT descriptor extract on average few tens of key-points which are not enough for proper classification. Also on these conditions, SURF descriptor tends to fail to extract features if object is blurred of the ROI simply doesn't provide enough texture discrimination power.

In general, the results obtained on PEVID dataset are superior to those obtained on SCOUTER (ca. 25% higher). This to the fact that PEVID image quality is higher that SCOUTER, therefore more information is available for feature extractors, which further facilitates classifiers power to generalize better the class (especially the colour descriptors as CSD). Another reason is that on PEVID dataset, scenarios proprieties remains somehow constant (e.g., the clothes colour of people that appear on distinct footages are usually the same, which is not the case of SCOUTER dataset, where the two scenarios have high variance of features proprieties, colour and texture).

For our current scenario and used datasets, the combination of more than two classifiers (second Blending approach) did not improve significantly the performance, lower results are obtained for SCOUTER dataset (compared with simple stacking, bagging and boosting). For second blending approach, generally the results are lower while compared with the first blending approach (or with Boosting and Bagging).

Table 6. F2-Score results using proposed multi Stacking approach

F2-Score		HoG	LBP	CM	CN	CSD	HK	SIFT	SURF	F
NB+	I	18.4	25.0	16.4	19.1	22.9	1.9	32.8	32.7	23.6
SVM-Lin	II	56.0	36.2	50.1	45.5	70.0	18.6	49.9	42.6	40.2
NB+	I	19.4	39.7	15.1	26.4	25.6	1.9	32.6	33.0	31.5
SVM-RBF	II	54.5	27.3	51.7	43.3	69.6	18.6	48.4	48.1	42.7
NB+	I	18.1	35.8	29.8	19.6	23.1	31.5	27.7	36.6	29.3
1KNN	II	56.4	42.2	57.9	44.9	68.5	1.9	62.5	58.5	44.0
NB+	I	19.1	34.3	24.8	24.6	28.0	1.9	32.6	34.3	31.5
3KNN	II	54.0	33.3	58.0	47.1	67.3	1.9	58.4	54.2	53.7
NB+	I	19.2	34.2	26.8	24.9	26.7	1.9	33.5	34.6	31.5
5KNN	II	54.4	32.5	60.0	49.2	67.3	1.9	57.8	53.7	55.0
NB+	I	19.1	40.3	14.1	14.7	16.8	1.9	31.9	34.5	31.5
J48	II	47.6	36.2	50.9	42.5	64.0	41.5	52.6	51.7	49.7
NB+	I	21.3	38.1	14.1	23.5	18.4	1.9	30.4	36.5	24.4
RF	II	48.7	29.9	50.3	50.8	57.9	16.4	52.9	51.1	44.5
SVM-Lin+	I	33.2	19.9	29.8	13.7	36.4	31.5	31.1	36.7	27.3
1KNN	II	61.4	26.5	62.8	63.4	**75.2**	1.9	62.5	58.6	37.3
SVM-Lin+	I	35.9	21.0	24.7	14.1	31.9	31.5	32.5	34.7	27.8
3KNN	II	60.0	25.4	63.3	63.9	72.7	1.9	57.9	54.2	36.5
SVM-Lin+	I	34.1	31.1	26.3	18.1	32.7	31.5	33.3	36.3	38.0
5KNN	II	60.3	26.7	64.6	63.0	72.9	1.9	58.1	53.1	37.8
SVM-Lin+	I	36.8	25.0	17.0	20.4	17.0	31.5	31.2	35.1	30.0
J48	II	52.6	33.7	52.5	57.3	63.7	41.5	52.8	51.8	29.7
SVM-Lin+	I	34.3	31.3	21.0	18.5	16.8	31.5	30.6	37.1	23.0
RF	II	48.1	27.3	54.2	59.8	56.4	6.0	52.8	50.7	38.5
SVM-RBF+	I	39.7	35.8	28.2	25.7	36.4	31.5	31.1	36.7	40.0
1KNN	II	65.9	42.2	58.2	68.8	74.6	1.9	62.5	58.4	70.8
SVM-RBF+	I	34.9	31.3	28.8	22.7	31.9	31.5	31.8	34.5	38.6
3KNN	II	62.6	32.9	52.8	66.9	72.4	1.9	56.3	54.3	68.2
SVM-RBF+	I	30.8	28.5	32.3	30.3	33.7	31.5	32.9	33.8	41.6
5KNN	II	62.9	33.7	58.3	68.9	73.2	1.9	57.7	53.7	64.1
SVM-RBF+	I	32.5	39.7	18.1	21.2	19.2	31.5	31.2	34.1	34.4
J48	II	48.3	35.7	48.7	56.1	63.8	41.5	52.8	52.4	52.4
SVM-RBF+	I	30.5	32.4	28.4	22.8	17.7	31.5	32.5	36.9	25.2
RF	II	47.3	29.6	48.7	59.5	57.7	6.0	51.4	51.1	44.3
J48+	I	32.5	32.8	20.6	20.7	36.4	31.5	31.1	37.5	40.0
1KNN	II	53.4	31.9	62.3	59.1	66.0	1.9	59.0	57.7	55.6
J48+	I	30.1	35.3	24.9	24.9	31.9	31.5	32.2	36.6	40.7
3KNN	II	53.7	28.5	62.2	58.4	65.4	1.9	58.5	54.4	54.5
J48+	I	26.3	32.8	23.0	30.1	33.8	31.5	33.7	36.4	39.6
5KNN	II	55.4	31.4	66.0	58.6	67.0	1.9	56.1	53.1	60.3
RF+	I	27.1	34.0	23.0	23.2	18.6	31.5	30.8	37.4	29.3
1KNN	II	49.6	31.0	56.6	60.8	60.2	1.9	56.8	55.3	45.9
RF+	I	26.5	33.7	24.5	20.7	18.5	31.5	31.8	37.1	29.0
3KNN	II	48.9	28.9	56.6	60.9	60.7	1.9	56.4	53.4	45.9
RF	I	28.4	33.3	24.3	22.5	19.0	31.5	33.9	36.9	33.9
5KNN	II	49.0	30.3	56.4	60.8	60.1	1.9	54.7	53.2	47.7

V. Conclusion

We have investigated and review three ensemble based learning techniques, Boosing, Bagging and Blending (Stacking). A comprehensive set of established video descriptors - classifiers pairs and combinations was evaluated while obtained promising results in terms of F2-Score. Comparing with the work done in [10] (see Table 5), we attaining better or similar results, thus making ensemble based learning approach attractive for investigating further system developments and software implementation which is operating in real world scenarios (benefits in terms of processing efficiency).

Though we obtained best result using second Stacking approach (Logical Regression as meta classifier and two classifiers for class learning generalization), if averaging the results on both datasets, best result is obtained using Bagging approach. One reason is that during training process the method generates new samples based on available distribution set. Also Bagging approach tends to be the fastest which makes it attractive for real-time tasks implementation. Future work will investigate co-training technique, which is adapted when there are only small amounts of labeled data and large sets of unlabeled data.

Acknowledgment

This work is supported by the ExcelDOC POSDRU/159/1.5/S/132397, also by projects as SCOUTER (28DPST/30.08.2013) and THz-DETECT (PCCA290/2014).

References

[1] P.V. Gehler and S. Nowozin, Let the kernel figure it out; Principled learning of pre-processing for kernel classifiers. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2836-2843, 2009.

[2] I. Hamid and S. Mubarak, Recognizing Complex Events Using Large Margin Joint Low-Level Event Model. *Lecture Notes in Computer Science Volume 7575*, pp. 430-444, 2012.

[3] P. Gaspar, J. Carbonell and J.L. Oliveira, On the parameter optimization of Support Vector Machines for binary classification. *Journal of Integrative Bioinformatics*, 9(3):201, 2012.

[4] O. Chapell, V. Sindhwani and S. S. Keerthi, Let the kernel figure it out; Principled learning of pre-processing for kernel classifiers. *Journal of Machine Learning Research*, pp. 203-233, 2008.

[5] H. Liu and L. Yu, Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4), pp. 491-502, 2005

[6] A. Vailaya, M.A.T. Figueiredo, A.K. Jain, and H.J. Zhang, Image classification for content-based indexing. *IEEE Transactions on Image Processing*, pp. 117 - 130, 2001.

[7] Y.L. Boureau, F. Bach, Y. LeCun, and J. Ponce, Learning mid-level features for recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2559-2566, 2010.

[8] R. Polikar, Ensemble based systems in decision making, *IEEE Circuits and Systems Magazine*, pp. 21-45, 2006.

[9] A. Knudby, Z.G. Acevedo, P. Chow, L. Hammar, L. Eggertsen, and M. Gullstrom, Multi-image ensemble classification, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2014.

[10] C.A. Mitrea,I. Mironica, B. Ionescu and R. Dogaru, Multiple Instance-based Object Retrieval in Video Surveillance: Dataset and Evaluation, *IEEE International Conference on Intelligent Computer Communication and Processing*, pp. 171-179, 2014.

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

[12] H. Bay, T. Tuytelaars, and L. Van Gool, Speeded Up Robust Features, *ETH Zurich, Katholieke Universiteit Leuven*.

[13] L. Hu, W. Liu, B. Li and W. Xing, Robust motion detection using histogram of oriented gradients for illumination variations, *International Conference on Industrial Mechatronics and Automation (ICIMA)*,2010.

[14] 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.

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

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

[17] 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.

[18] Robert M. Haralick, Statistical and structural approaches to texture, *Proc. IEEE*, vol. 67, no. 5, pp. 786-804, 1979.

[19] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, *The American Statistician*, 46(3), pp.175-185, 1992.

[20] Breiman Leo, Random Forests, *Machine Learning*, 45(1), 2001.

[21] J. R. Quinlan, Simplifying decision trees, *International Journal of Man-Machine Studies*, 27(3), 1987.

[22] Russell Stuart, Artificial Intelligence: A Modern Approach (2nd ed.), Prentice Hall. ISBN 978-0137903955, 1995.

[23] C. Cortes and V. Vapnik, Support-vector networks, *Machine Learning*, 20(3), 1995.

[24] P. Viola, and M. Jones, Rapid object detection using a boosted cascade of simple features, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.1, pp. 511-518, 2001.

[25] Leo Breiman, Bagging predictors, *Machine Learning*, 24(2),1996.

[26] J. Sill, G Takacs, L. Mackey and D. Lin, Feature-Weighted Linear Stacking, arXiv:0911.0460, 2009.

[27] David A. Freedman, Statistical Models: Theory and Practice. Cambridge University Press,p. 128, 2009.