EVENT RECOGNITION ON IMAGES USING SUPPORT VECTOR MACHINE AND MULTI-LEVEL HISTOGRAMS OF LOCAL PATTERNS

Bassel Zeno[1], Dmitry Yudin[1] and Bassel Alkhatib[2]
[1]Belgorod State Technological University named after V.G. Shukhov Russia, Belgorod, Kostukova, Russia
[2]Damask University, Damask, Syria

ABSTRACT

Event recognition on images plays important role in the context of events search and review, archiving and storage of photos, advertising, media. Paper presents a classification system based on texture features (Multi-Level Histogram of Color Multi-Scale Local Binary Pattern and Local Derivative Pattern) and support vector machine (SVM). For classification process authors use several SVM classifiers and compare their results. Three classifiers are tested: "Liblinear", "LibSVM", "Pagasos". Training and test samples are taken from "MediaEval 2013" set of annotating images. The result precision of event recognition varies from 66.75% to 93.75% and the best classifier in terms of AUC metric is "Liblinear" with local binary pattern. Developed image classification system can be used in many applications, for example in internet services, robotics, automated control systems of technological processes where image scenes should be recognized.

Keywords: image recognition, event, machine learning, local binary pattern, local derivative pattern, support vector machine.

1. INTRODUCTION

In recent years, and mainly due to the pervasiveness of digital cameras and camera-phones and rapid development of the sharing platforms (social networks, online albums, etc), there has been an exponential increase in creation, storage and sharing of multimedia data (text, audio, video and photos).

Personal photo collections have also been increasing rapidly in recent times. In contrast to the Web. Where millions of users worldwide capture images and videos to record various events in their lives. According to current estimates, more than 250 billion photos have been uploaded to Facebook and more than 350 million photos are uploaded every day on average [1].

Hence we believe that one important challenge in recent Web-oriented computer vision research is to retrieval photos related to specific event from millions of photo streams. Under the social events we mean events that are planned by people, attended by people and that the media illustrating the events are captured by people. The retrieval of such social events can potentiate a variety of applications [2, 3]. For example, if a family decides to go to a scuba diving trip, they can make a plan by previewing what other people usually do. After the trip, they can also review the similarities and differences of their trip compared to others, and fill missing parts of their photo sets by others' pictures [4]. For users, finding digital content related to social events is challenging, requiring to search large volumes of photo streams, possibly at different sources and sites .Algorithms that can support humans in this task are clearly needed. The proposed task thus consists in developing algorithms that can detect event-related photos and group them by the events they illustrate or are related to (see Figure-1).

Event detection from photos is very challenging, because of the ambiguity of photos across different event classes and because many photos do not convey enough relevant information. Unfortunately, the field still lacks standard evaluation data sets to allow comparison of different approaches. We use data set of photo collections containing more than 4,00 images, annotated with 6 diverse social event classes.

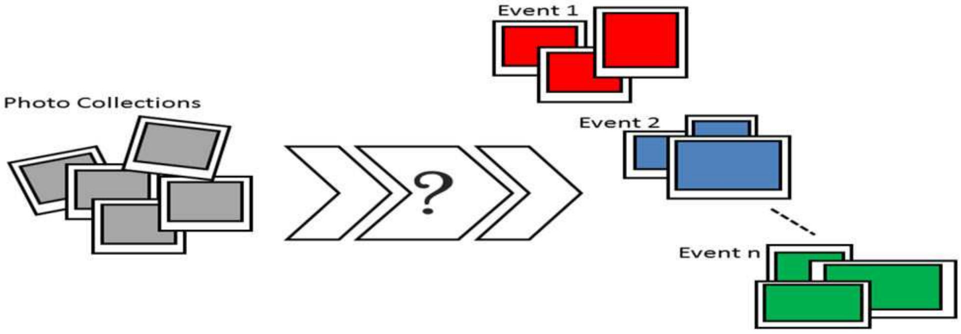

Figure-1. Event detection from photo collection.

2. TASK STATEMENT AND DEFINITION

The problem statement can be summarized as follows: given a large image collections, we aim to classify images based on their image features.

Event Recognition is defined as the process of images classification based on image features, each one of them represents event (e.g., trip, party, meeting, sport, etc). Formally, let the photo collection $\boldsymbol{P}$ be a set of N photos, $\boldsymbol{P} = \{\boldsymbol{p_1}, \boldsymbol{p_2}, \dots \boldsymbol{p_N}\}$. The goal of recognition is to find a subsets of photos$\boldsymbol{E_i}$ ($\boldsymbol{i} = \boldsymbol{1 \dots k}$with$\mathbf{k}$ number of classes), each $\boldsymbol{E_i}$representsan event.

3. MAIN PART PROPOSED SYSTEM

The proposed system based on direct features [5]. The system consists of five parts (see figure 2). These parts are respectively:

- Patches Extraction;
- Feature Extraction
- Classifier Module;
- Testing Module;
- Kernel Fusion Module.

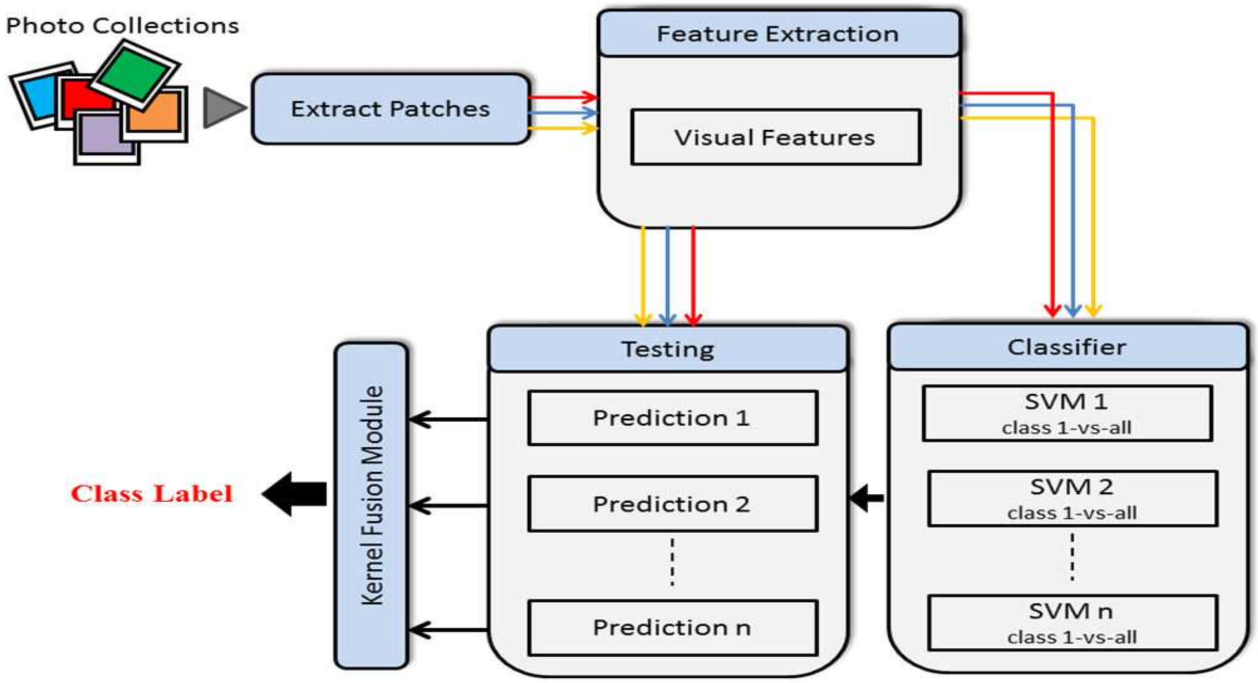

Figure-2. Proposed system structure.

a) Patches extraction

With the images as the input, the outputs of this step are image patches. This process is implemented via sampling local areas of images, usually in a dense (e.g., using fixed grids [6]) or sparse (e.g., using feature extractors [7,8]) manner.

b) Feature extraction

Feature plays a very important role in the area of image processing. Before getting features, various image preprocessing techniques like binarization, thresholding, resizing, normalization etc. are applied on the sampled image. After that, feature extraction techniques are applied to get features that will be useful in classifying and recognition of images. Visual features is defined as the global or local operations applied to an image to generate certain quantitative measurements, which are helpful for solving the computational tasks. According to the types of operations, the visual feature can be categorized into global feature and local feature. Global feature measures the property of the whole image, and local feature measures the property of local regions in the image [9]. We use the following visual features:

Multi-level histogram of color multi-scale local binary pattern (mlhmslbp)

The local binary pattern (LBP) is a powerful illumination invariant texture primitive. The histogram of the binary patterns computed over a region is used for texture description. The operator describes each pixel by the relative gray levels of its neighboring pixels. If the gray level of the neighboring pixel is higher or equal, the value is set to one, otherwise to zero. The descriptor describes the result over the neighborhood as a binary number (binary pattern) [10]. After identifying LBP for each pixel, the whole image is represented by building a histogram. Figure 3 shows ad example of a LBP operator[11]:

$$LBP_{R,N}(x,y) = \sum_{i=0}^{N-1} s(n_i - n_c)2^i, \ s(x) = \begin{cases} 1, & x \geq 0, \\ 0, & otherwise. \end{cases}$$

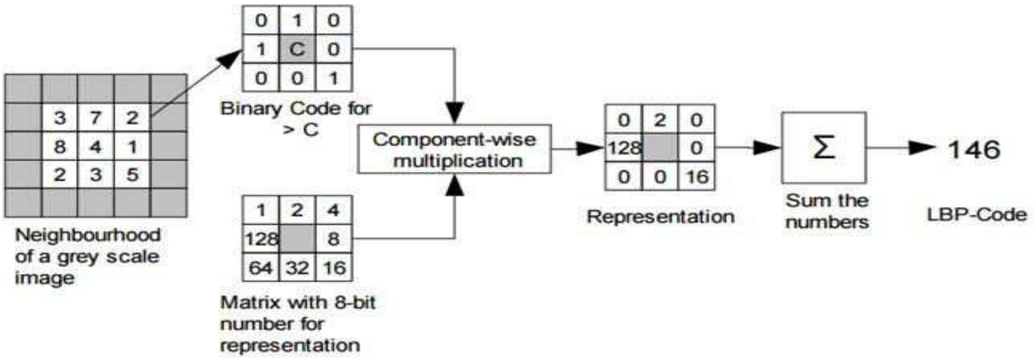

Figure-3. Example of LBP feature extraction.

Multi-level histogram of color multi-scale local derivative pattern (mlhmsldp)

Local Derivative Pattern (LDP) was proposed by Baochang Zhang [12] for face recognition with high order local pattern descriptor. It encodes directional feature pattern based on local derivative variations. It can capture more detailed information than the first order LBP. LDP is a micro pattern representation which can also be modeled by histogram to preserve the information about the distribution of the LDP micro patterns. LBP is always considered first-order local pattern operator, because LBP encodes all-direction first-order derivative binary result whereas LDP encodes the higher-order derivative information. So it contains more discriminative features than LBP. Given an image, the first-order derivatives along 0° , 45° , 90° and 135° directions are denoted as I'_α , where, α =0° , 45° , 90° and 135°. Let V_0 be a point in (V) , and V_i ,i=1,…8 be the neighboring point around V_0 . The four first-order derivatives at $V = V_0$ are given in equations 2, 3, 4 and 5 for 0° , 45° , 90° and 135° respectively [13]:

$$I'_{0^\circ}(V_0) = I(V_0) - I(V_4),$$

$$I'_{45^\circ}(V_0) = I(V_0) - I(V_3),$$

$$I'_{90^\circ}(V_0) = I(V_0) - I(V_2),$$

$$I'_{135^\circ}(V_0) = I(V_0) - I(V_1)$$

In a general formulation, the nth order LDP is a binary string describing gradient trend changes in a local region of directional (*n*-1)th order derivative images $I'_\alpha(V)$ as

$$LDP^n_\alpha(V_0) = \{ f(I_\alpha^{n-1}(V_0), I_\alpha^{n-1}(V_1)), f(I_\alpha^{n-1}(V_0), I_\alpha^{n-1}(V_2)), f(I_\alpha^{n-1}(V_0), I_\alpha^{n-1}(V_7)), f(I_\alpha^{n-1}(V_0), I_\alpha^{n-1}(V_8)) \},$$

Where $I_\alpha^{n-1}(V_0)$ is the (n-1)th order derivative in α direction at $V = V_0$. $f(I_\alpha^{n-1}(V_0) \ , I_\alpha^{n-1}(V_i) \ )$, is defined as

$$f(I_\alpha^{n-1}(V_0), I_\alpha^{n-1}(V_i)) = \begin{cases} 0 \ if \ I_\alpha^{n-1}(V_i).I_\alpha^{n-1}(V_0) > 0, \\ 1 \ if \ I_\alpha^{n-1}(V_i).I_\alpha^{n-1}(V_0) \leq 0. \end{cases} \quad i = 1,2,\ldots 8.$$

This formula encodes the (*n*-1)th -order gradient transitions into binary patterns, providing an extra order pattern information on the local region.

After identifying LDP for each pixel, the whole image is represented by building a histogram.

c) Classifier module

We use large-locale linear SVM such Liblinear [14]orPagasos [15, 16] or libsvm [17]. They are used to train models since features are almost perfectly linearly separable. The onevs. all classification strategy is selected to train SVM for each class. At the end we will get the *n* training models for each classifier:

n = number of classes · number of features.

d) Testing module

After training classifiers we test them using testing dataset.

e) Kernel fusion module

We used fusion method for combine results from multiple classifiers. We used two type of fusion method (Max, Mean).

4. SYSTEM IMPLEMENTATION AND RESULTS

At the feature level and from 2-D grayscale input images. We use mlhmsldp and mlhmslbp with features with scale = [0.5 , 0.75 , 1], patch size = 16 pixel, nbins = 12 and derivative order = 2. They are extracted using Matlab functions. After feature extraction, we get the raw form of feature descriptors. These descriptors can be directly used in classification. We used the Liblinear, Pagasos SVM, libsvm, as a SVM classifier, they used for large-scale data and design a composite kernel to jointly learn between textual and visual features and heterogeneous features. The one vs. all classification strategy is selected to train SVM. Here, We trained every classifier with one class and then we got a training model. At the end we will get the *n* training model for each classifier, since *n* = number of classes * number of features, then we test these models using selected testing dataset, then we apply fusion (Mean) overall results to get the final results.

a) Dataset

To evaluate the proposed system, data from the MediaEval social event detection challenge 2013 [18] was used. We downloaded 5376 pictures for training set and 4036 pictures for test set. But we chose 300 images for training(61%) and 192 images for testing(39%).We chose four events as following "concert", " conference", " exhibitions", " sport" and we added two types of event, it is called "wedding" and "graduation". Their dimension as follows (width from 336 to 1599 pixel; high from 336 to 1098 pixel). Number of training images equals 50 and number of testing images equals 32 for each class.

b) Algorithm evaluation approach

The ground truth was created by human annotators [18]. The results of event-related media item detection were evaluated using four evaluation measures [19]:

1) $\text{Precision} = \frac{tp}{tp + fp}$;

2) $\text{Recall} = \frac{tp}{tp + fn}$;

3) $\text{F-score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$,

where tp: true positive, eqv. with hit,tn: true negative, eqv.with correct rejection. fp:false positive, eqv.with false alarm. fn:false negative, eqv.with miss. These measures calculated for each class (event), and we used also accuracy measure overall results,4) Area Under Curve (AUC),

AUC is a criteria for comparing learning algorithms. It's recommended when evaluating and comparing classifiers [20]. We have three classifiers: Liblinear, LibSVM, Pagasos.

c) Experiment results

We made6 experiments on notebook computer with processor: Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz, RAM: 8,00 GB, System Type: 64 bit Operating System, Windows Edition: Windows 8.1.Experiments are tested on Matlab.

Results of experiments combined in Table-1. Execution time of each experiments varied from 2.949 seconds to 91.091 seconds.

Table-1 shows than the result precision of event recognition varies from 66.75% to 93.75%. The best precision of the “Concert” event recognition is achieved by Liblinear and Pagasos SVM with local derivative patterns (0.8065). For the “Conference” recognition the best precision is achieved by LibSVMwith local derivative patterns (0. 6129). For the “Exhibition” recognition the best precision is achieved by Pagasos SVM with local binary patterns and local derivative patterns (0. 6875). The best precision of “Graduation” recoginition is achieved by Liblinear with local derivative patterns (0. 7188). For the “Sport” recognition the best precision is achieved by Pagasos SVM with local binary patterns (0. 9375). “Wedding” event have the best recognition precision 0.8438 using Liblinear with local binary patterns. Table 2 shows the best precision results using max function overall experiments.

We compared classifiers performance by applying AUC measure at results and when we applied fusion, we got AUC = 0.9289. Figure 4 shows the comparison between classifiers using AUC and fusion them. We note that the best experiment result is the results of the first experiment.

It means that the best classifier for event recognition task is Liblinear using mlhmslbp (Local binary pattern) feature.

Table-1. Results of experiments.

Num. of experi-ment	Features type	Classifier	Quality measures	Event type					
				Concert	Conference	Exhibition	Graduation	Sport	Wedding
1	Local binary patternsLBP	Liblinear	Precision	0.7419	0.5484	0.6250	0.6250	0.8125	**0.8438**
			Recall	0.7931	0.6800	0.8000	0.6667	0.6842	0.6279
			F-Score	0.7667	0.6071	0.7018	0.6452	0.7429	0.7200
2		LibSVM	Precision	0.7097	0.4194	0.6250	0.6563	0.7813	0.6563
			Recall	0.6875	0.5200	0.8000	0.6364	0.6250	0.6000
			F-Score	0.6984	0.4643	0.7018	0.6462	0.6944	0.6269
3		Pagasos SVM	Precision	0.6452	0.3226	**0.6875**	0.6563	**0.9375**	0.5000
			Recall	0.6897	0.6667	0.7097	0.6563	0.5172	0.6400
			F-Score	0.6667	0.4348	0.6984	0.6563	0.6667	0.5614
4	Local derivative patternsLDP	Liblinear	Precision	**0.8065**	0.4516	0.5938	**0.7188**	0.8125	0.8125
			Recall	0.6757	0.6667	0.7037	0.6970	0.7222	0.7222
			F-Score	0.7353	0.5385	0.6441	0.7077	0.7647	0.7647
5		LibSVM	Precision	0.7097	**0.6129**	0.5625	0.6250	0.7188	0.7813
			Recall	0.6471	0.6129	0.6667	0.7143	0.7188	0.6579
			F-Score	0.6769	0.6129	0.6102	0.6667	0.7188	0.7143
6		Pagasos SVM	Precision	**0.8065**	0.4194	**0.6875**	0.5625	0.8438	0.6875
			Recall	0.6098	0.7222	0.7097	0.7826	0.6000	0.6875
			F-Score	0.6944	0.5306	0.6984	0.6545	0.7013	0.6875

Table 2. The best precision results from all experiments.

	Concert	Conference	Exhibition	Graduation	Sport	Wedding
Precision	0.8065	0.6129	0.6875	0.7188	0.9375	0.8438

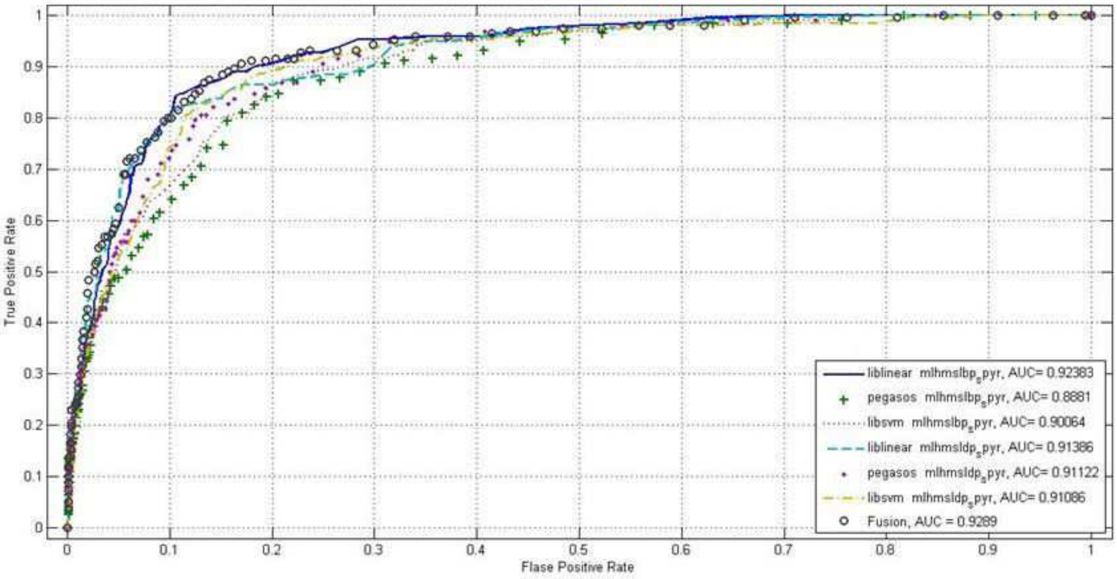

Figure-4. Comparison between classifiers.

5. CONCLUSIONS

In this paper we use technique "Extract Direct Features" and "Fusion" in testing process for recognizing six types of events. We extract 2 types of features (local binary and local derivative patterns), then we train the extracted features with 3 types of classifiers (Liblinear, LibSVM, PagasosSVM). The result precision of event recognition varies from 66.75% to 93.75% and the best classifier for described task in terms of AUC metric is "liblinear" with local binary pattern. The results give us the highest performance in class "sport". This technique very fast. Although we have used dataset with 492 photos, but in future we will test it on large dataset. Also in future we can use technique "Bag of words" and other features types for classifiers training. Developed image classification system can be used in many applications, for example in internet services, in robotics, in automated systems of technological processes control where image scenes should be recognized.

ACKNOWLEDGEMENTS

The work was performed under the Strategic development program of BSTU n.a. V.G. Shukhov 2012-2016 (# 2011-PR-146) and grant of Russian Foundation for Basic Research # 14-41-08016.

REFERENCES

[1] A Focus on Efficiency, A whitepaper from Facebook, Ericsson and Qualcomm, pp. 6, September 16, 2013.

[2] Kozelkov, O.A., 2014. Upravleniyegruppovympredpochteniyem v sotsial'noysfererazlichnykh form kollektivnoydeyatel'nosti [Group preferences management of the social sphere of various forms of collective activity]. Bulletin of BSTU named after V.G. Shukhov, 2: 166-169.

[3] Yudin,D.A., G.G. Postolsky, A.S. Kizhuk and V.Z.Magergut,2013. Mobile Robot Navigation Based on Artificial Landmarks with Machine Vision System. World Applied Sciences Journal, 24 (11):1467-1472. (DOI: 10.5829/idosi.wasj.2013.24.11.7010)

[4] Gunhee Kim, 2013. Reconstruction and Applications of Collective Storylines from Web Photos Collections. Ph.D. Thesis, CMU-CS-13-125, Computer Science Department, Carnegie, pp: 5.

[5] Yudin, D.A. and V.Z. Magergut, 2014. Sistemytekhnicheskogozreniyadlyamonitoringaprotse ssaobzhigavovrashchayushchikhsyapechakh [Machine vision systems for monitoring the firing process in a rotary kilns]. Belgorod, BSTU, pp: 108.

[6] Marszalek, M.,C. Schmid, H. Harzallah, and J. van de Weijer, 2007. Learning representations for visual object class recognition. ICCV Worshop on the PASCAL VOC Challenge.

[7] Harris, C. and M. Stephens, 1988. A combined corner and edge detector. The Fourth Alvey Vision Conference, pp: 147 - 151.

[8] Matas, J., O. Chum, M. Urban and T. Pajdla, 2004. Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing, 22 (10):761–767.

[9] Weisi, Lin, Tao Dacheng, JanuszKacprzyk, Li Zhu, EbroulIzquierdo, and Haohong Wang (Eds.), 2011. Multimedia Analysis, Processing and Communications, pp: 681-686.

[10] Heikkila, Marko, PietikainenMatti and SchmidCordelia, 2009. Description of interest regions with local binary patterns. Pattern Recognition, 42 (3): 425 - 436.

[11] Tobias Lindahl, 2007. Study of Local Binary Pattern.

[12] Baochang, Zhang and YongshengGao, 2010. Local Derivative Pattern versus Local Binary Pattern: Face Recognition with High Order Local Pattern Descriptor, IEEE Transactions on Image Processing, 19 (2): 533-544.

[13] Raju,U. S. N.,S. Kumar, B. Mahesh, E. Reddy,2010. Texture classification with high order local pattern descriptor local derivative pattern. Global Journal of Computer Science and Technology, 10 (8):72-76.

[14] Liblinear library. Date Views 01.06.2015. Available at: http://www.csie.ntu.edu.tw/~cjlin/liblinear/

[15] Pegasos library. Date Views 01.06.2015. Available at: http://www.cs.huji.ac.il/~shais/code/

[16] Zhimo,Shen, 2009. Primal Estimated sub-Gradient Solver for SVM. University of California.

[17] Libsvm library. Date Views 01.06.2015. Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[18] Petkos,G., S. Papadopoulos, V.Mezaris, R.Troncy, P.Cimiano, T. Reuter and Y.Kompatsiaris, 2014. Social event detection at MediaEval: a three-year retrospect of tasks and results. Proc. ACM ICMR 2014 Workshop on Social Events in Web Multimedia (SEWM), pp: 6.

[19] Precision and recall. Date Views 01.06.2015.Available at:: http://en.wikipedia.org/wiki/Precision_and_recall

[20] Ling, X., Jin Huang and Harry Zhang, 2003. AUC: a Better Measure than Accuracy in Comparing Learning Algorithms. Advances in Artificial Intelligence of the series Lecture Notes in Computer Science, 2671: 329-341.