I.J. Intelligent Systems and Applications, **2017, 2, 42-48**
Published Online February 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2017.02.06

Video-based Flame Detection using LBP-based Descriptor: Influences of Classifiers Variety on Detection Efficiency

Oleksii Maksymiv
Lviv State University of Life Safety, Lviv, Ukraine
E-mail: aleks.maksymiv@gmail.com

Taras Rak
Lviv State University of Life Safety, Lviv, Ukraine
E-mail: rak.taras74@gmail.com

Dmytro Peleshko
Lviv Polytechnic National University, Lviv, Ukraine
E-mail: dpeleshko@gmail.com

***Abstract*—**Techniques to detect the flame at an early stage are necessary in order to prevent the fire and minimize the damage. The flame detection technique based on the physical sensor has limited disadvantages in detecting the fire early. This paper presents the results of using local binary patterns for solving flames detecting problem and proposes modifications to improve the quality of detector work. Experimentally found that using support vector machines classifier with a kernel based on Gaussian radial basis functions shows the best results compared to other SVM cores or classifier k-nearest neighbors.

***Index Terms*—**Computer vision, fire detection, image analysis, local binary patterns, image segmentation.

I. Introduction

Fire detection from the video stream is considered to be difficult and not fully resolved the problem in computer vision areas research. The majority of fire detection methods in images are based on such features as color, texture or some another temporal characteristic. However, the flame as object of classification has a difficult structure with unstable parameters of movement, wide range of colors (depending on temperature and burning substances), variable contours and other dynamic effects. All these factors greatly increase the number of incorrect classification.

This article describes the process of fire detecting with usage of local binary patterns (LBP) texture descriptor and its modifications, aimed at improving the flame detector efficiency in video stream dissemination. Further, for received features, support vector machine classifier (SVM) applied with different variations of kernels and k-nearest neighbors (k-NN) algorithm. As a result of using this approach, the time required to process the image which is analyzed, average parameters of detection accuracy and recall were calculated.

II. Research Background

To solve the problem of fire detecting was used a variety of approaches. Conditionally, they can be divided into the following groups: color and texture segmentation, motion detection in images, additional flame attributes (flickering, sharp corners, etc.).

There are a lot of research works devoted to color segmentation, which can be divided by type of model used: RGB [1], HSV [2], YCbCr [3], CIE L*a*b [4]. In this paper, we used color model YCbCr, which shows the best quality detection rate - 99% [3]. However, despite the good results, presented by researchers, this method characterizes by the high rate of false positives alarms, that arise from changes in lighting and required parameters setting for each facility separately. Obtained results, in our view, were largely represented by small representative dataset on which the test was performed, which, in particular, taken into account during our experimental studies.

To lower the false alarm rate, in addition to different color models, some researchers also incorporated motion and fire flickering into their work. Rinsurongkawong et al. [4] have proposed to use Lucas Kanade optical flow algorithm for fire moving detect. Toreyin et al. [5] used hybrid background estimation method to detect moving objects. For checks flicker in flames he using 1-D temporal wavelet transform and color variation in fire colored moving regions using 2 –D spatial wavelet transform. Although powerful, these techniques are computationally intensive.

Problems of using various descriptors to identify fire in images/video stream are lighted in works [6] (SIFT descriptor), [7] (HOG descriptor), [8] (SURF descriptor).

Also, there are some studies of using LBP descriptors for the task of flames identify [9, 10]. However, these works are not sufficiently disclosed potential choice classifier, which, according to experimental results can positively effect on developed detection system. In particular, we proposed a number of modifications LBP algorithm, which allow improving the quality of flame detection.

III. Theoretical Basis of LBP Descriptor

As noted above, there are various feature extraction techniques: HOG, Haar Wavelets, LBP etc. For developed system, we have decided to use LBP descriptor. The reason for this choice was caused by number of advantages for fire detection: speed calculation, invariability to brightness change which preserve order $(a > b \rightarrow f(a) > f(b))$, resistance to noise and textures variations.

Classical LBP operator, which was proposed in work [11], forms labels for the image pixels by thresholding the 3 x 3 neighborhood of each pixel with the center value and considering the result as a binary number. Using a circular neighborhood and bilinearly interpolating values at non-integer pixel coordinates allow any radius and number of pixels in the neighborhood. Pixels that have a value greater than the central pixel (or equal), take the value "1", those that are less central, taking the value "0". The results of the comparisons are stored in an 8-bit binary code, which describes the neighborhood pixels. Neighborhood examples used to define a texture and calculate a local binary pattern shown in Fig.1.

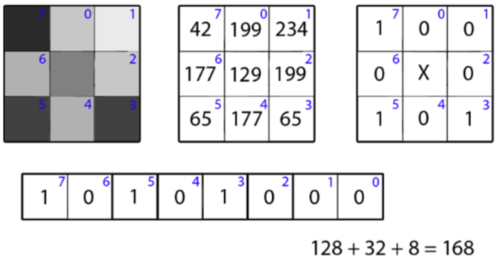

Fig.1. Example of computing LBP

Mathematically, this process can be defined as follows:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s\,(g_p - g_c)2^P \tag{1}$$

where g_c is a center pixel value and g_p $(p = 0, \ldots, P - 1)$ – value of pixels P, which are along a circle of radius R (R > 0), s(x) - sign function, which can be defined as follows:

$$s(x) = \begin{cases} 1 \; if \; x \geq 0 \\ 0 \; if \; x < 0 \end{cases} \tag{2}$$

A number of pixels (P) along a circle with radius R can be chosen arbitrarily. To calculate values in these points for different radius R and points number P, bilinear interpolation was used. Fig. 2 shows pixels set for different neighborhood P and R.

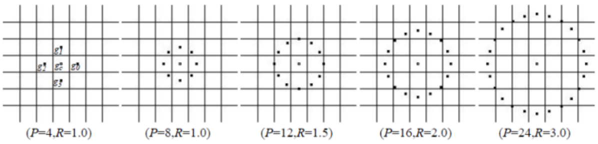

Fig.2. Neighborhood examples used to define a texture and calculate a local binary pattern with different value P and R

It has also been experimentally established, that for improving descriptor performance and efficiency, following set of pixels along a circle: P = 8, R = 1.0 is the most optimal choice for our tasks.

In work [12] was proposed to analyze the images using not all binary patterns, but only those who may contain at most two bitwise transitions from "1" to "0" or vice versa. The number of bitwise transitions, when the binary string is circular, gives a uniformity measure U of the pattern as follows:

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c), if \; U(LBP_{P,R}) \leq 2 \\ P + 1, otherwise \end{cases} \tag{3}$$

$$U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \tag{4}$$

These templates are called uniform pattern and characterized by two advantages. First, we save a memory because the number of uniform LBP patterns is given by $P(P-1)+3$, leading to a much shorter histogram representation. Secondly, they define only important local textures, such as the end of lines, edges, corners, spots. Further LBP operator with uniform patterns we will defined as $LBP_{P,R}^{up}$

In the last step, after the LBP labeled image has been obtained, the LBP histogram (H_i) is computed, which can be defined as:

$$H_i = \sum_{x,y} B(LBP_{P,R}(x,y) = i), i = 0, .., n-1,$$
$$B(k) = \begin{cases} 1, if \; k \; is \; True \\ 0, otherwise \end{cases} \tag{5}$$

In many cases, the LBP histogram may have excessive information, and much of it is often irrelevant to the problem to be solved. Which is why it was decided to use the method proposed in [13]. So, the histogram bins are sorted by descending order. After that, there is only part of the histogram that includes a percentage and patterns which are then submitted for classification.

IV. Proposed Method

Developed flame/non-flame detector must meet the requirements of implicitly high-level object detection and good performance even on slow machines. In this paper, to reduce the time required for verification area, which may cause a fire, we proposed the additional use of color segmentation and some spatial-temporal object characteristics.

The process of detecting area with fire in images based on the five basic principles described below:

1. Color segmentation;
2. Moving objects detecting;
3. Applying LBP to obtained area from the previous steps;
4. Based on the received histograms from the previous method - doing classification;
5. Fit into bounding box regions which qualifier recognize as "flame" and go to next frame.

A. Color segmentation

The performance of the fire detection system critically depends on the effectiveness of fire pixel classifier. A common approach to color segmentation area with fire is using color model YCbCr. For this, on the test dataset doing interactive image segmentation to identify the boundaries of which can be areas with flames. Tyrgay Celik et al. [3, 14] most fully analyzed the depending quality of fire segmentation from the selected color model. As a result, it was found that YCbCr has higher quality in comparison with the RGB model. Following equation describes the color conversion from RGB to YCbCr:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2568 & 0.5041 & 0.0979 \\ -0.1482 & -0.2910 & -0.4392 \\ 0.4392 & -0.3678 & -0.0714 \end{bmatrix} * \\ * \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{6}$$

where Y - luminance component; Cb - the difference between blue component and brightness Y; Cr - the difference between red component and brightness Y.

Segmentation area with flames was based on rules:

$$R_1(x,y) = \begin{cases} 1, if\ Y(x,y) \geq Cb(x,y) \cap \\ Cr(x,y) \geq Cb(x,y) \\ 0, otherwise \end{cases} \tag{7}$$

$$R_2(x,y) = \begin{cases} 1, if\ (Cb(x,y) \leq 120) \cap \\ (Cr(x,y) \geq 150) \\ 0, otherwise \end{cases} \tag{8}$$

$$R_3(x,y) = \begin{cases} 1, if\ Y(x,y) \geq Ymean \cap \\ Cb(x,y) \leq Cbmean \cap \\ Cr(x,y) \geq Crmean \\ 0, otherwise \end{cases} \tag{9}$$

$$R_4(x,y) = \begin{cases} 1, if\ Cb(x,y) - Cr(x,y) \geq Th \\ 0, otherwise \end{cases} \tag{10}$$

The mean value for Y, Cb and Cr are calculated as equation:

$$Ymean(x,y) = \frac{1}{M \cdot N} \sum_{x=1}^{M} \sum_{y=1}^{N} Y(x,y) \tag{11}$$

$$Cbmean(x,y) = \frac{1}{M \cdot N} \sum_{x=1}^{M} \sum_{y=1}^{N} Cb(x,y) \tag{12}$$

$$Crmean(x,y) = \frac{1}{M \cdot N} \sum_{x=1}^{M} \sum_{y=1}^{N} Cr(x,y) \tag{13}$$

B. Moving objects detection

An estimate of the background (often called a background model) is computed and evolved frame by frame: moving objects in the scene are detected by the difference between the current frame and the current background model. Initially formed current background model $B_t(x,y)$ over time t, by splitting the input image $I(x,y)$. To update the background model creates a circular buffer with size n = k + 1 elements, in which recorded pixel brightness in k frames, and the brightness of the pixel $I(x,y)$, which contained in the initial background model $B_t(x,y)$. Value of the cyclical buffer arranged in ascending order, and searches the meaning, which is located in the middle of the cyclical buffer. Updating background model $B_t(x,y)$ was doing by replacing its corresponding pixel on the value in middle element of cyclical buffer.

Getting frame difference is due to calculation difference $D(x,y) = |I_t(x,y) - B_t(x,y)|$ current video frame from the updated binary mask and calculating two thresholds cyclical buffer expressions:

$$Th_1(i,j) = \left(\left|P_{\frac{1}{4}k} - P_{\frac{3}{4}k}\right|\right); \\ Th_2(i,j) = |P_1 - P_k| \tag{14}$$

where i, j - number of row and column number which is at the intersection pixel frame, k - the number of elements cyclical buffer, P_i - element circular buffer that is on position i.

In the resulting binary mask $M(x,y)$, a pixel difference $D(x,y)$ is classified as belonging a moving object, if he presents on mask Th_1 and located into three pixels' mask Th_2, otherwise the pixel is considered as background.

For pixels segmentation that can be attributed to micromotions generated by background, moving objects in the resulting binary mask $M(x,y)$ allocated by grouping them all connected pixels and calculates their modules gradient by the expression:

$$G_t = \sqrt{\left\|\frac{\partial I_t(i,j)}{\partial i}\right\|^2 + \left\|\frac{\partial I_t(i,j)}{\partial i}\right\|^2} \qquad (15)$$

where $\left\|\frac{\partial I_t(i,j)}{\partial i}\right\|^2 = I_{t-\Delta t}(i-1,j) - I_t(i+1,j)$, $\left\|\frac{\partial I_t(i,j)}{\partial i}\right\|^2 = I_{t-\Delta t}(i,j-1) - I_t(i,j+1)$ - gradients in horizontal and vertical directions, || || - vector norm.

C. LBP descriptor using

Taking into account the combustion process, we have chosen some modifications of LBP descriptor which will improve the efficiency of our flame detector on the one hand and increase computation speed on the other.

D. NRLBP

Classic LBP descriptor characterized by two disadvantages that are described in [16, 23]. First of all, it is sensitive to changes between background and foreground. As can be observed in fig. 3, the binary code of LBP descriptor in two images with flames are different from each other. However, they describe the same spatial characteristics of fire. Another disadvantage of classic LBP descriptor it is high demands on the bins calculation and storage. Thus, the original implementation $LBP_{8,1}$ required 256 histogram bins, in $LBP_{8,1}^{up}$ – only 59 bins, which is also quite costly. To develop flame detection system which can work in real time it is necessary to minimize requirements for their calculation and storage.

Taking into consideration this disadvantages we propose to use Non-Redundant Local Binary Pattern (NRLBP) [17] to obtain visual information about flame presence. From Fig. 3 can be seen that LBP codes of two images complement each other (the sum will amount $2^P - 1$). Mathematically NRLBP can be defined as follows:

$$\text{NRLBP}_{P,R}(x_c, y_c) = min\{\text{LBP}_{P,R}(x_c, y_c), 2^P - 1 - \text{LBP}_{P,R}(x_c, y_c)\} \qquad (16)$$

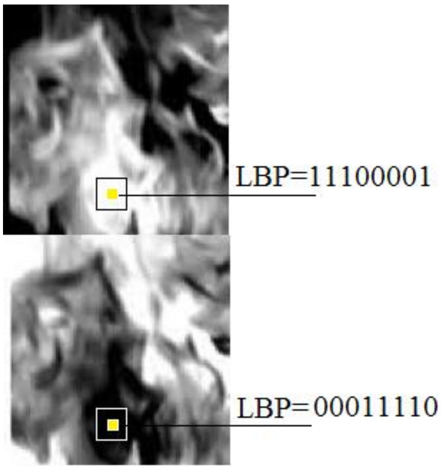

Fig.3. LBP codes for the flame with same structure, but different inversion in background and foreground

Based on the above, binary codes of upper and lower images in fig.3 represent the same texture. This, in turn, will reduce feature vectors dimension at half.

E. VLBP

Despite the fact that discriminatory ability of NRLBP characterized by improved capacity and much lower bins storage requirements, in comparison with classical LBP, it includes only local visual information. To build a quality vision-based fire monitoring system, primarily important information about its temporal characteristics. For this, we decide to use Volume Local Binary Pattern (VLBP), which is calculated on the difference between frames.

In this method, arbitrary pixel frame of video is described as $g_{0,c} = I(x,y,t)$ and its coordinates determined in accordance (x,y) and times moment t. Target (central) pixels has described by equation (17):

$$g_{\text{i},c} = I(x, y, t + i \cdot \Delta t), \quad i = -1, 0, 1, \qquad (17)$$

where Δt – time between sequence.

Neighboring pixels are selected according to equation (18):

$$g_{\text{i},p} = I\big(x + x_p, y + y_p, t + i \cdot \Delta t\big), \quad p = 0, \dots, P - 1, (6) \qquad (18)$$

where P - the number of neighboring pixels.

Dynamic LBP operator, which will depend on the time interval between successive frames Δt , number of neighboring pixels P and the distance between the target and adjacent pixels R.

$$VLBP = \sum_{q=0}^{3P+1} v_q 2^q \qquad (19)$$

where v_q – determines function (2), arguments of which is the difference $(g_{t,p} - g_{0,c})$, whose number is $((3P + 2)$ [18].

As a result, after operators using we obtained a histogram, which, in turn, is fed to the input classifier for further processing.

F. Classification

Selection of qualitative features descriptor undoubtedly plays a key role in developing vision-based fire detection system. However, after receiving features that describe the desired image, the equally important step is the rejection of so-called "extra" features around the set. Therefore, classifier selection is important task during image classification and requires more detailed study.

G. Support Vector Machine

Classification using support vector method is widely used in image recognition thanks to its empirically good performance and moderate calculation complexity. General principles of SVM described in [19, 20]. SVM becomes famous when using pixel maps as input; it gives accuracy comparable to sophisticated neural networks with elaborated features in a handwriting recognition task. It is also being used for many applications, such as hand writing analysis, face analysis and so forth, especially for pattern classification and regression based applications [19].

It should also be noted that the classification function (F) becomes:

$$F(x) = sign\big((w, \varphi(x)\big) + b) \qquad (20)$$

and expression $k(x, \acute{x}) = (\varphi(x), \varphi(\acute{x}))$ called core classifier. Classification accuracy can be improved by proper kernels selection.

The most common kernels are [24]:

- polynomial (homogeneous):$k(x, \acute{x}) = (x \cdot \acute{x})^d$;
- polynomial (inhomogeneous): $k(x, \acute{x}) = (x \cdot \acute{x} + 1)^d$;
- radial basis function (RBF): $k(x, \acute{x}) = exp(-\gamma\|x - \acute{x}\|)^2, \text{для } \gamma > 0$;
- Gaussian radial basis function (RBF-G): $k(x, \acute{x}) = \exp(-\frac{\|x-\acute{x}\|^2}{2\sigma^2})$
- sigmoid function: $k(x, \acute{x}) = tanh(\kappa x \cdot \acute{x} + c)$, almost for all k>0 i c<0

To analyze the performance of the flame classifier we analyzed this kernels in work, results of which are described in Section V.

H. k-nearest neighbors algorithm

k-nearest neighbors algorithm is a non-parametric method based on a similarity measure (e.g., distance functions). If K = 1, then the case is simply assigned to the class of its nearest neighbor.

A main advantage of the k-NN algorithm is that it performs well with multi-modal classes because the basis of its decision is based on a small neighborhood of similar objects. Therefore, even if the target class is multi-modal, the algorithm can still lead to good accuracy. However, a major disadvantage of the k-NN algorithm is that it uses all the features equally in computing for similarities [21, 25].

It should be noted about some disadvantages of k-NN: model can not be interpreted, it is computationally expensive to find the k nearest neighbors when the dataset is very large, performance depends on the number of dimensions that we have.

V. Results

A. Dataset

For test classifications quality we formed the own dataset of video, which is divided into two categories: the flame and without flame. A number of copies used for the category "flame" - 22, for categories in which there are no flames - 34. In particular, the last category includes objects which may look like a flame (bright clothing, lights, garlands, etc.).

To train the classifier we used image dataset that was formed during the previous study [22]. It includes 1876 images with fire and the 4634 images without fire.

B. Evaluation method

Results of fire classification evaluated based on two well-known techniques - precision and recall. High precision relates to a low false positive rate, and (20)high recall relates to a low false negative rate. High scores for both show that the classifier is returning accurate results (high precision), as well as returning a majority of all positive results (high recall).

$$Precision = \frac{TP}{TP+FP} \qquad (21)$$

$$Recall = \frac{TP}{TP+FN} \qquad (22)$$

where TP - the number of true positives (correct system inference about area where were flames); FP - false-positive operation (system result agrees with the presence of fire whereas ground truth agrees with the absence of fire); FN - false-negative operation (system result agree of absence of fire whereas ground truth agree of the presence of fire).

C. Experimental results

The results figures of above methods are presented in Table I. As we can see, using SVM classifier with a kernel based on radial basis function showed the best precision result. SVM classifier with Gaussian radial basis function showed the best recall result in fire detection task. SVM with polynomial kernel showed the best time information processing. k-nearest neighbors algorithm shows the worse performance of time and classification efficiency.

Table 1. Results of using different classification methods

Method	Precision	Recall	Time
SVM- polynomial kernel	81,3 %	93,5 %	**0,009**
SVM-RBF kernel	**81,9 %**	94,2 %	0,02
SVM-RBFG kernel	81,6 %	**94,3 %**	0,03
SVM- sigmoid kernel	81,3 %	93,7 %	0,03
k-NN	80,7 %	92, 6 %	0.01

VI. Conclusions

For developed video-based flame detector, the most important parameter is the recall of classification. Using SVM classifier with Gaussian RBF kernel shows the best recall and relatively high performance of accuracy and performance time. So, it's the most better choice for solving the problem of flame detection in video.

More future research should be done to verify and improve the results of this paper. In future work, we will endeavor to incorporate fire reflection detection algorithms into our current algorithm.

Some results of false positive and true positive/negative detection shown in Fig. 4, 5, 6.

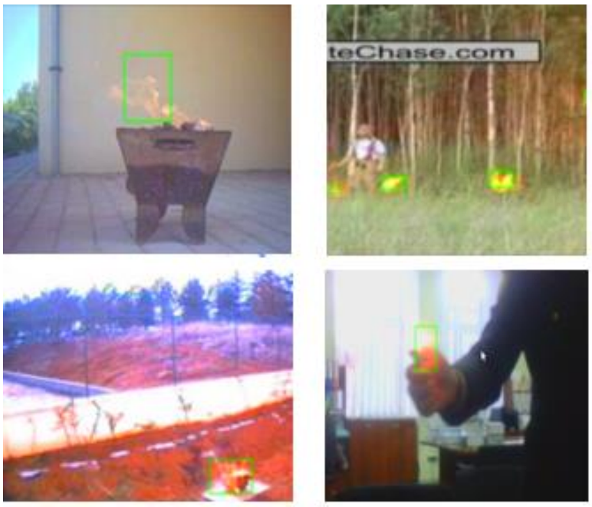

Fig.4. Examples of correct detection (True positive)

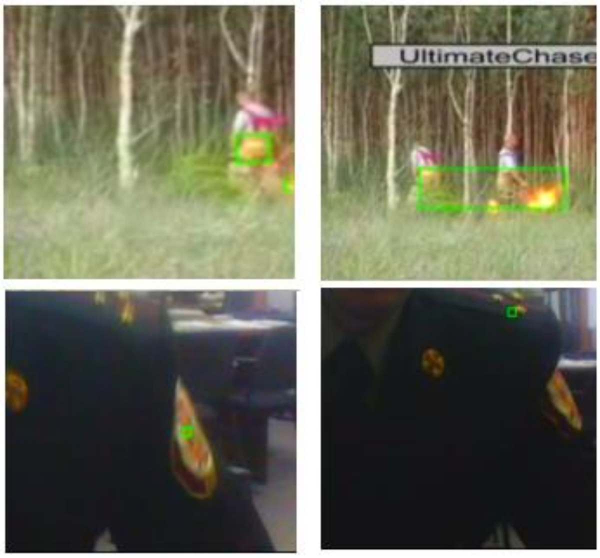

Fig.5. Examples of false detecting (True negative)

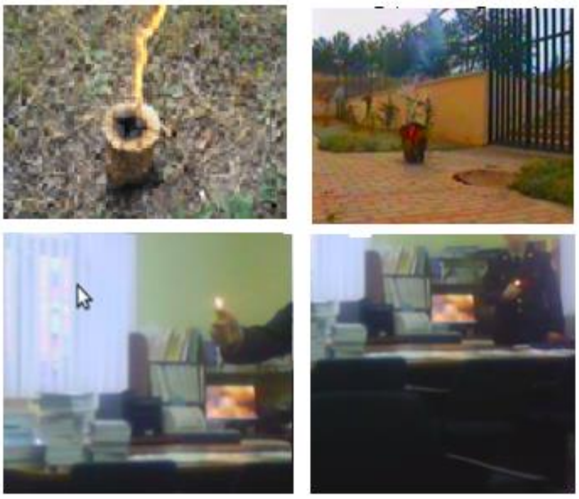

Fig.6. Examples of falsely detector work (False positive)

REFERENCES

[1] T. Chen, P. Wu, and Y. Chiou, “An early fire-detection method based on image processing,” ICIP ’04, pp. 1707–1710, 2004.

[2] X. Qi, J. Ebert “A Computer Vision-Based Method for Fire Detection in Color Videos,” International journal of imaging, no. 9, pp. 22–34, 2009.

[3] Celik T., H. Demirel “Fire detection in video sequences using a generic color model,” Fire Safety Journal, no.2, pp. 147–158, 2009.

[4] S. Rinsurongkawong, M. Ekpanyapong, and M. N. Dailey, “Fire detection for early fire alarm based on optical flow video processing,” in 9th Int. Conf. on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology Phetchaburi, pp. 1-4, 2012.

[5] B.U. Toreyin, Y. Dedeogln, U. Gudukbay, and A.E. Cetin , “Computer vision based method for real time fire and flame detection,” Pattern Recognition Letters, vol. 27, no. 4, pp. 49-58, 2006.

[6] L.Ying, W. Hui-qin, B. Zhen-Bo and X. Fei “Research on Large Space Building Fire Positioning Technology Using Video Surveillance Images,”pp. 1218–1222, 2013.

[7] P. Gomes, P. Santana, J. Barata “Vision-based Approach to Fire Detection,” International Journal of Advanced Robotic Systems, no. 11, pp. 1–12, 2014.

[8] Jiang, B., Lu, Y., Li, X. and Lin, L. “Towards a solid solution of realtime fire and flame detection,” Multimedia Tools and Applications., 2014.

[9] F. Sthevanie, H. Nugroho, F. Arif Yulianto “Visual-Based Fire Detection Using Local Binary Pattern-Three Orthogonal Planes,” CYBERNETICSCOM, 2013, pp. 155–159.

[10] Y. Zhao, T. Guizhong “Fire video recognition based on flame and smoke characteristics”, Systems and Informatics (ICSAI), 2014, pp. 113 – 118.

[11] T. Ojala, M. Pietikainen, and T. Maenpaa “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7):971–987, 2002.

[12] S. Nigam, A. Khare “Integration of moment invariants and uniform local binary patterns for human activity recognition in video sequences,” Multimed Tools Appl, 2015, pp. 1–30.

[13] S. Liao, M. W. “Dominant Local Binary Patterns for Texture Classification,” Transactions on Image Processing, Vol 18, 2009, No 5

[14] T. Celik “Fast and efficient method for fire detection using image processing,” ETRI journal, 2010, №6, pp. 881–890.

[15] T. Celik, H. Ozkaramanli, H. Demirel “Fire pixel classification using fuzzy logic and statistical color model,” Acoustics, Speech and Signal Processing, 2007, pp. 1205–1208.

[16] H. Tian, L. Wanqing, P. Ogunbona and others “Smoke Detection in Videos Using Non-Redundant Local Binary Pattern-Based Features,” Multimedia Signal Processing, 2010, pp. 1–4.

[17] D. T. Nguyen, Z. Zong, P. Ogunbona, and W. Li, “Object detection using non-redundant local binary patterns,” in Proc. IEEE International Conference on Image Processing, 2010, pp. 4609–4612.

[18] G. Zhao and M. Pietikäinen, "Dynamic Texture Recognition Using Volume Local Binary Patterns," Infotech Oulu and Department of Electrical and Information Engineering University of Oulu, Oulu, 2007.

[19] J. C. Burges. “A tutorial on support vector machines for pattern recognition,” Data Mining and Knowledge Discovery, vol. 2(2), pp. 121-167, 1998.

[20] U. Krebel “Pairwise classification and support vector Machines. In Advances in Kernel Methods: Support Vector Learnings,” MIT Press, Cambridge, MA, pp 255-268, 1999.

[21] Jinho Kim, Byeong-soo Kim and Silvio Savarese, “Comparison Image Classification Methods: K-Nearest Neighbor and Support-Vector-Machines”, Proceedings of the 6th WSEAS International Conference on Circuits, Systems, Signal and Telecommunications”, Cambridge, USA, pp. 133-138, Jan. 25-27, 2012.

[22] O. Maksymiv, T. Rak, O. Menshikova, "Deep convolutional network for detecting probable emergency situations", 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP), Lviv, Ukraine, pp. 199-202, 2016.

[23] M. Pietikäinen, A. Hadid, G. Zhao and T. Ahonen, “Computer Vision Using Local Binary Patterns,” Oulu: Springer, 2011

[24] B. Scholkopf and A. J. Smola. “Learning with Kernels,” MIT Press, 2002.

[25] Ye zhiwei, Yang Juan, Zhang Xu, Hu Zhengbing,"Remote Sensing Textual Image Classification based on Ensemble Learning", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.8, No.12, pp.21-29, 2016.DOI: 10.5815/ijigsp.2016.12.03.

Authors’ Profiles

Oleksii Maksymiv was born in Drohobych, Ukraine (1992), now he lives in Lviv, Ukraine. He obtained B.Sc. (2013) in fire safety and M.Sc. (2014) in project management. Currently, he is a Ph.D. student in computer science at the Lviv State University of Life Safety. His research interests focus on the computer vision, intelligent systems, image analysis and processing.

Rak Taras, Dr. Sc., an Associate Professor and vice-rector on scientific and research work at Lviv State University of Life Safety. He received a Doctor of Sciences degree in Information Technology. He has published more than 100 papers in international and national scientific issues. The main research interests include information technology, decision-making system, control systems for emergencies.

Dmytro Peleshko, Dr. Sc., Professor at Lviv Polytechnic National University, Lviv, Ukraine. He has published more than 100 papers in international and national scientific issues and journals and he is the author of several monographs. His scientific interests are image and video processing for the system of artificial intelligence

How to cite this paper: Oleksii Maksymiv, Taras Rak, Dmytro Peleshko,"Video-based Flame Detection using LBP-based Descriptor: Influences of Classifiers Variety on Detection Efficiency", International Journal of Intelligent Systems and Applications(IJISA), Vol.9, No.2, pp.42-48, 2017. DOI: 10.5815/ijisa.2017.02.06