Analysis of Machine Learning Methods for Wildfire Security Monitoring with an Unmanned Aerial Vehicles

Dmitriy Alexandrov, Elizaveta Pertseva, Ivan Berman, Igor Pantiukhin, Aleksandr Kapitonov
ITMO University
Saint Petersburg, Russia
mr.alexdmitriy@mail.ru, {pertseva.lizaveta, bermanivan42}@gmail.com, {zevall, kapitonov.aleksandr}@corp.ifmo.ru

***Abstract*—The article is about the methods of machine learning, designed for the detection of wildfires using unmanned aerial vehicles. In the article presented the review of machine learning methods, described the motivation part of machine learning usage and comparison of fire and smoke detection is made. The research was focused on machine learning application for monitoring task with a restrictions according to scenarios of a real monitoring. The results of experiments with demonstration of effectiveness of detection are presented in the conclusion part.**

I. Introduction

The last decade of technological progress characterized by the widespread usage of unmanned aerial vehicles (UAVs): in the US FAA by January 2018 registered 1 million drones for commercial and personal use [1]. Logistic of goods, photo and video filming, agriculture, telecommunications, construction, security and rescue, these areas are needed in UAVs for efficiency increasing [2], [3], [4]. The monitoring of a extended and spread infrastructure is the most useful function of a UAVs [5], [6]. Moreover, the monitoring process can be almost completely automated.

Particularly, UAVs show their workability for the environmental monitoring of forests for fire safety increasing [7]. Right now there are three main ways to detect the wildfire source: fire towers, manned aerial vehicles and satellite monitoring. These methods have significant limitations that reduce the monitoring efficiency. Fire towers have a limited view range, and the accuracy of observation depends from weather conditions and time of day. Piloted vehicles (helicopters and airplanes) needed in the special expensive infrastructure. Also, both of these methods have a human factor and risk to the human life. Satellite observation allows you to automate data collection, but the area of the wildfire for detection is too high. By this time, large forces will be needed to eliminate the fire. Satellite observation give the maximum advantage in strategic preventive actions [8].

UAVs are mitigating all disadvantages of previous methods. The drone fleet much more flexible for territory monitoring with a different weather condition. Distribution of drones around territory can be made in optimal way of price or time [9]. The drone fleet equipped by thermal camera and air quality sensors makes possible to recognize a smoke or wildfire source at the early stage. The main advantage of UAVs is autonomy and low cost against to existing wildfire detection methods [10].

UAVs have been used as an additional monitoring tool since the beginning of the 2010s [11]. Now UAVs are used mainly for data collection: it needs semi-automated processing by 2–4 operators to work during 18–25 days on monitoring tasks [11]. To reduce that costs the automated processing is needed. The task of monitoring in automatic mode leads to the development of a computer vision (CV) system using various detection methods. There are many well-known approaches for detecting fire with photo- and video-processing: starting from simple image processing to machine learning methods (ML) [7], [12]. However, not all of these methods are suitable for monitoring with drones. But usage of drones for fire security regular monitoring needs a small source of wildfire detection. When designing a monitoring system with computer vision, one has to take into account the nuances and limitations that are imposed by field conditions, lack of computing resources and other features. There are a lot of limitation and restriction with external conditions for computer vision, also lack of computing resources and other features. For the widespread usage this system should have an economical advantages.

At the beginning of the 21st century, there was a widespread development of methods for fire detecting, but the practical usage of UAVs for these tasks began only in last years. For example, with NASA support in 2006 was demonstrated a system for a real-time monitoring of wildfires with UAVs [13]. In 2016-2018, the experiments about UAVs fleet in detecting, localizing and measuring squares of wildfires tasks were published [14], [15]. The computer vision applications for that tasks were described in a several papers (e.g. [16], [7], [17]).

But we still have a big gap between the research of a computer methods and its implementation, especially with a machine learning. And the main goal of this work is comparison of machine learning methods applied to the wildfire monitoring tasks. The main focus of the research is on machine learning methods, including deep learning methods cause this class of artificial intelligence methods the most promising for real-time monitoring tasks. The advantages of machine learning methods against classical image processing methods applied to monitoring are mentioned in the sections below.

Main points of article:

- The article considers classical methods of machine learning and deep learning methods: Haar and LBP cascades, Faster R-CNN, SSD, YOLO.

- The methods are compared for aerial detection of wildfires. The comparison parameters are the quality of detection and performance.
- The conclusions about the applicability of the detection of fire or smoke are presented.
- The article is limited to the consideration of the monitoring scenario using the UAV, outside the context of the equipment parameters.

The paper is organized as follows. Section 2 provides an analysis of machine learning methods for wildfire detection. The comparison of smoke and fire detection is described in section 3. In the section 4, the software implementation of the described methods and their testing for fire detection tasks with experimental results description are provided. And the summary in the last section.

II. Analysis of Machine Learning Methods

A. Benefits of Machine Learning

Early fire detection methods were based on classic image processing methods. Classical methods are rather unreliable: for example, methods based on color classification, and methods for detecting fire by analyzing its movement can produce false-positive results in the presence of objects colored in the color of the flame. Classical methods give high reliability: for example, methods based on color classification, and methods for detecting fire by movement analyzing can give false-positive results with objects of flame color. In forest monitoring tasks can be reflections of sunlight on the surfaces of buildings, ponds and rivers. It should be added that the standard recognition methods give a bad result for cases with natural hindrance (fog, dust, etc.).

Machine learning techniques, on the other hand, can overcome these disadvantages. The percentage of false detections for some methods is less than 5%. In addition, deep learning allows you to adapt the same methods for different monitoring scenarios, without the need to rebuild the method itself: only the appropriate data setsare needed for training. Several years ago for the application of neural networks in practical tasks the performance of existing equipment wasnt enough. At the moment, the computing power that the neural network can be deployed on common models of UAVs. Now there are additional modules that can be placed on the UAV for the neural networks usage [18].

B. Problems of the Monitoring Task

As part of the monitoring tasks, two large machine learning tasks are resolved in computer vision: the task of classification and the task of detection. The first task is simple: during the solution, a specific class of objects is detected in the image, without localizing the object. It seems that the use of simple and computationally less costly methods will be more effective for monitoring tasks with the help of UAVs. However, pictures of forests can contain many objects of interest: for example, in the case of several sources of fire or smoke. This will cause the assignment of a class label to become undefined. The second problem is that usually monitoring is carried out at high altitudes (from 200–300 meters) and it becomes impossible to pinpoint the location of the fire source, only its approximate region. For quick action on fire fighting this is critical, especially during dry seasons. And in fact, the scenarios of the processes of classification and detection are interrelated. In the future, only methods for object detection will be considered.

Two factors determine the complexity of the task of object detection. First, it is necessary to process a large number of the proposed locations of the object. Secondly, the proposed locations require clarification in order to obtain accurate localization. Because of this, detection methods have limitations on speed, accuracy, and implementation complexity.

To train the model the labeled data is needed. In the context of object recognition, labeled data is images with bounding boxes (bounding the required object) with coordinates and a class label. At the same time, it is necessary to avoid the phenomenon of overfitting of the model on similar images, since the model should work quite accurately in non-standard situations (other than the training set).

As a rule, object recognition consists of three steps:

1) At the beginning, the model or algorithm is used to generate areas of interest or proposals. Region proposals are a large set of bounding boxes in the original image.
2) Next, the visual parameters for each of the bounding boxes are determined, and the presence of objects of interest is evaluated.
3) At the post-processing stage, non-maximum suppression is performed — the combining of bounding boxes associated with the same object [19].

There are several approaches to the generation of region proposals: a selective search algorithm, the use of complex visual parameters, the direct generation of regions of interest using a sliding window. However, it is important to find a balance between detection accuracy and computational complexity: the more regions are generated, the higher the chance to find an object, and the more expensive processing in the real time. For example, the selective search algorithm, where image pixels are grouped and assumptions based on clusters of pixels are generated, are not applicable in tasks with limited computing resources.

In the beginning, classical methods for detecting objects will be considered. In these methods, the identification of features is carried out by an algorithmic approach. These can be gradient directions or pixel locations in a determined form. Then these features are classified, the training method considers the vectors of features as points in a multidimensional space and searches the boundaries of surfaces in this space. Thus, all objects associated with the same class will be on the same side of boundaries.

Next will be considered several methods of deep learning based neural networks. Classical ML methods provide lower detection accuracy against deep learning, but require less computational power, so we add to the comparison.

C. Haar Cascades

Haar cascades — this is a classic mathematical model used to detect objects using machine learning [20]. The idea of the method is to identify and classify the characteristics of the image. 3 types of features are defined: two-rectangle, three-rectangle and four-rectangle features. The characteristic value of two rectangles is equal to the difference between the number of pixels in two rectangular areas. These areas can be horizontal or vertical. The feature of the image of the three rectangles is calculated as the sum of the pixels in the two side rectangles minus pixels in the central rectangle. And finally, for the four rectangles — this is the difference between the rectangles along the diagonals.

After definition of features from positive and negative images, the stage of training the classifier by the method AdaBoost begins. The idea is to create a ”strong” classifier by combining ”weak” classifiers. In this case, a ”weak” classifier is created on the basis of a small number of features. Further, the most important attributes are selected for classification using AdaBoost.

The detection process is done by classifying sliding windows. The classification is making with the decision tree method or the cascade method. The window must move through a sequence of weak classifiers and be detected. If the window is rejected by one of the classifiers, the next classifier is not activated, and the window is not detected.

D. Cascades of Local Binary Patterns

Local binary patterns (LBP) is a kind of visual descriptor used for classification tasks [21]. LBP operator marks every pixel of the image as a digit from 0 to 9. Next, the pixels are compared with their neighbors by subtracting the center pixel in the cell. Negative values are replaced by 0, positive ones — by 1. Next, the cell is recorded as a binary clockwise sequence, starting from the upper left corner. According to the features from positive and negative images, the process of learning and detection is making as in the previous paragraph.

E. Faster Region-based Convolutional Network

Faster R-CNN (Faster Region-based Convolutional Network) implements object detection through deep learning [22]. The model is a modification of the Fast R-CNN model (and Fast R-CNN is modification of R-CNN). There are three main tasks for the original R-CNN method:

1) Search for regions with objects.
2) Image feature extraction using convolutional neural network (CNN).
3) Using the support vector machine (SVM) for classifying objects and clarifying the positions of regions of interest.

The main problem for UAV is the high computational resources on a board. For example, R-CNN uses the method of selective search of regions, training takes up to 80 hours, and the work time is tens of seconds. Fast R-CNN was able to improve performance up to a few seconds by optimizing the time with pooling proposals region. This became possible due to a change in the architecture: first, the original image is now fed to the input of a convolutional neural network; a convolutional map of features is formed at the output. Second, the method of selective region search uses a convolutional feature map, and with the help of the RoI pooling layer, the dimension of the feature vector is changed, and the vector is transferred to the softmax layer to classify and refine the position.

Faster R-CNN completely abandoned selective search. The process of regions search is fully performed by the convolutional neural network RPN (Region Proposal Network). The resulting regions are further classified in the same way as in Fast R-CNN.

RPN uses a neural network as a sliding window, the input of a neural network is a feature map of intermediate layers. The window moves along the map, and at the output transmits a feature vector associated with two fully-connected layers for the boxes — the box regression layer and the box classification layer. For each window position, the K maximum of regions of interest is predicted: the regression layer has $4K$ outputs for coding the coordinates of the regions, and the classification layer outputs $2K$ the probability of finding an object in the region. K predicted number of regions of interest are called anchors. Anchors are centered in the window and have a different rectangular shape and size. For every anchor the neural network samples the probability value, and keeps the probabilities greater than the threshold value. As a result, the selected anchors and feature maps are transferred to the Fast R-CNN model.

F. Single Shot Multibox Detector

The SSD (Single Shot Multibox Detector) model is an object recognition model that combines proposal regiones and image parameters obtained by one deep neural network [23]. In this case, all pixels in the image are not resampled, and the accuracy is not inferior to previous models.

The model accepts an image at the entrance and passes it through several convolutional layers with different filter sizes (10×10, 5×5, 3×3). The neural network predicts bounding boxes based on a map of features found in different positions. Boxes are found by a special convolutional layer called an extra feature layer with a 3×3 filter. The result is a set of restrictive boxes that perform the same function as anchors in the Fast R-CNN model. Box has 4 parameters: two coordinates of the center (x, y), width and height.

To determine boxes at the end of the SSD model, non-maximum suppression is used. After that, the Hard Negative Mining (HNM) method is used to reduce false positives. HNM allows you to reduce the number of negative boxes required for learning by adding images that have a false positive to the set of negative examples. Boxes are ranked by their probability, and boxes with a highest probability are selected.

G. You Only Look Once

The YOLO model samples a image once, just like the SSD model [24]. Next, the image processed modification GoogLeNet or VGG. Several convolutional layers with ReLU activation feature process a feature map. Full connected layers and dimensional changes are the last stage of processing is used to obtain a tensor.

Every vector from the grid contains information about:

- box center coordinates in the grid,

- box size to the size of the initial image,
- box confidence indicator about choosing right object,
- coefficients indicating the probability of finding a particular class in a particular box.

After calculating the tensor, vector of classes for every box is compiled by multiplying the original vector by confidence coefficient of a specific box. Next, for every class a comparison with a threshold value is applied, and if class values are less than the threshold, then the class is assigned a zero. The boxes are sorted with respect to class, then non-maximum suppression is performed. The result of detection for every box is the maximum non-zero class.

Thanks to this architecture, YOLO has the ability to be modified by shaping the number of convolutional layers. In addition, the model is trained on general representations of objects and allows usage in different situations.

H. Reasons for Methods Selection

First, these methods showed high detection probability. Secondly, these methods have not been practically tested for use in wildfire monitoring with help of UAVs. All the presented methods are capable of detecting fire and smoke [25], [26], [27], [28], [29].

Comparison of classic ML methods shows that the accuracy of Haar and LBP cascades can be similar (about 80%) [30]. However, in the case of Haar cascades, it varies from 50% to 70%, while the accuracy of LBP cascades is stable.

Faster R-CNN and SSD models on the same datasets are demonstrated that the difference in accuracy is small (mean average precision for Faster R-CNN — 75.9%, for SSD — 80.0%) and speed of computation potentially allows the usage of methods in real time (for Faster R-CNN — 20 FPS, for different SSD versions — from 20 to 40 FPS) [23].

The reason for analysis of YOLO for solving the fire detection problem is high performance and speed of object detection. YOLO has mAP indicators similar to other methods and allows to detect objects in real time. Also, there is a light YOLO model — YOLO-tiny and its architecture contains fewer layers than the original version. Due to this, performance increases significantly and according to the developers of the Darknet framework, it reaches a value of 220 FPS. However, the detection accuracy of YOLO-tiny is lower.

III. Difference in Fire and Smoke Detection

The source of fire in the forest can be detected by flame or smoke. To train machine learning models, it is necessary to choose a set of classes of objects of interest, one class is defined in the simplest case.

Flame and smoke detection is characterized by the following features:

- Regardless of the choice of the object, fire source will be detected in the recognition process.
- The smoke covers more than the flame area in the frame of video from UAV.
- Smoke can cover the flame, making it impossible for visual detection.
- Because of the foliage on the trees, small fires can be difficult to detect. On the other hand, smoke goes up and can be detected.
- In some cases, there is only smoke without visible fire.
- At night, detection of smoke is difficult.

Based on this, it can be concluded that the optimal object for detection in terms of speed and visibility of detection (during daylight hours) is smoke. However, it is preferable to use hybrid methods, depending on the nature of the monitoring environment. Further detection will be focused on the search for smoke.

IV. Implementation and Comparison of Methods

Testing equipment:

- OS: Ubuntu 18.04.1 LTS,
- RAM: 32 GiB (for learning),
- processor: Intel Xeon CPU E5-2630 v4 @ 2.20GHz.

Method comparison indicators:

1) the number of positive and negative detection results with and without error: TP — True Positive, TN — True Negative, FP — False Positive, FN — False Negative;
2) precision:
$$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}};$$
3) recall:
$$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}};$$
4) accuracy:
$$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}};$$
5) FPS — frames per second;
6) mAP — mean average precision;
7) IoU (intersection over union) — the relationship between the intersection area of a real box and a model box and the total area of a real box and a model box.

True Positive — real smoke was detected in the frame, False Positive — there is no smoke, but it is detected, False Negative — real smoke is not detected in the frame, True Negative — there is no smoke, and it is not detected.

A. Datasets

The following datasets were used for training:

1) The Real smoke + Forest background dataset — dataset with 12 000 images of real smoke on the background of forests [27].
2) The Simulative smoke + Forest background dataset — dataset with 12 000 simulated images of smoke [27].
3) Own dataset, developed on the basis of video recordings of wildfires made from UAVs. 6600 positive images, 15600 negative images.

TABLE I. MODEL PARAMETERS BASED ON THE HAAR AND LBP CASCADES

		Haar	LBP
N_{pos}	positive samples num	5000	2500
N_{neg}	negative samples num	11368	10000
N_{stg}	learning stages num	15	10
R_{brk}	acceptance ratio break value	0.001	0.001
W	sample width, px	24	24
H	sample height, px	24	24
V_h^{min}	hit rate, $V_h^{min} = 1 - FN$	0.99	0.99
V_a^{max}	false alarm detections num before the new training stage	0.5	0.5
R_{tr}	weight trim rate	0.95	0.95
V_{wk}^{max}	num of weak decision trees for learning at each stage	100	100

TABLE II. TEST RESULTS OF THE HAAR AND LBP CASCADES

	Frms	TP	TN	FP	FN	Prec	Recall	**Acc**	**FPS**
Haar	1780	1556	0	224	0	0.874	1	**0.874**	**14.62**
LBP	1780	1447	0	283	0	0.813	1	**0.813**	**22.40**

B. Testing Classic Machine Learning Methods

Comparison of Haar cascades and LBP cascades (Fig. 1). Training done with own dataset. The minimum size of a detection object is 100×100 pixels.

The parameters of the tested models are shown in Table I.

The test results are shown in Table II.

C. Testing Deep Learning Methods

Characteristics of training models of Faster R-CNN, SSD and YOLO and the results of their verification on test data (smoke images, Fig. 2) are summarized in Table III.

Since the Faster R-CNN model showed itself better in terms of total error, its performance was tested on video records with fires to check the dependence of FPS on video quality (Table IV).

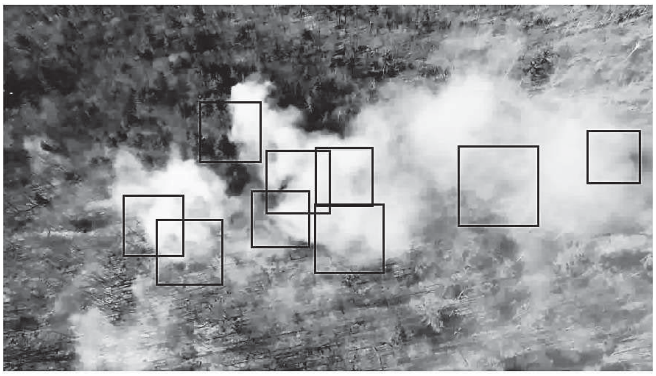

Fig. 1. Detection result of Haar cascades

TABLE III. COMPARISON OF TRAINING MODELS FASTER R-CNN, SSD AND YOLO

	Faster R-CNN	SSD	Yolo_v2
dataset	RS+SS	RS+SS	RS+SS + Custom
batch size — samples num per iteration of training	1	100	16
num of steps	1000	100	2500
time of training, min	433	135	5645
mAP at IoU > 0.5	0.6916	0.1447	—
object classification err	0.133	6.400	—
object localization err	0.1403	3.052	—
total err	0.3611	9.698	1.354

TABLE IV. TESTING THE FASTER R-CNN MODEL ON VIDEO

Dimensions of videos, px	Duration, sec	Time for detecting, sec	FPS
240×240	40	175	5.71
640×480	40	211	4.74
1440×1080	40	244	4.10
1920×1080	60	376	3.83

V. CONCLUSION

The results of classical methods and methods of deep learning are summarized in Table V. The main findings are presented below.

- The results show that the best performance is achieved for classical methods of machine learning, however, their accuracy is lower than Faster R-CNN and YOLO models.
- The SSD model showed the worst performance results and similar accuracy results with classical methods.
- The poor accuracy results of classical methods are explained by the fact that detection methods with cascades are more applicable to the recognition of objects that have a constant shape and color [21]. Smoke (and flame) does not satisfy this condition. However, Haar and LBP cascades are applicable in situations, where the frame has a large amount of smoke.

Fig. 2. Detection result of the Faster R-CNN model

TABLE V. FINAL RESULTS

Methods	Precision	Recall	**Accuracy**	**FPS**
LBP	0.813	1	**0.813**	**22.40**
Haar	0.874	1	**0.874**	**14.62**
YOLO_v2	1	0.983	**0.983**	**5.78**
Faster R-CNN	1	0.959	**0.959**	**4.10**
SSD	0.884	0.907	**0.811**	**1.33**

- The results of smoke detection using Faster R-CNN show that method average performance is 4 FPS, at that only smoke with a light color shade is detected.
- The YOLO model demonstrated the best accuracy among all considered models and was the fastest among deep learning models. Like Faster R-CNN, YOLO is more suitable for detecting fires at an early stage. Therefore, this model is optimal for solving monitoring problems.

The application of machine learning methods requires compromise between performance and accuracy. Moreover, the performance can be compensated by changing the flight mode of the UAV — more drones, longer flights, etc. For example, this will allow quite successfully apply Faster R-CNN.

On the other hand, deep learning methods for detection tasks improving permanently, therefore, in the following papers, perspective modifications of You Only Look Once model will be investigated. Also, in the future research, the theoretical base described here will be tested on UAVs in the real time.

ACKNOWLEDGMENT

This work was financially supported by ITMO University: grant num. 618275 ”Analysis of heterogeneous data sources from unmanned aerial vehicles for monitoring of nature protection zones and restricted access areas”.

REFERENCES

[1] US Department of Transportation website, ”FAA drone registry tops one million”, Web: https://www.transportation.gov/briefing-room/faa-drone-registry-tops-one-million.

[2] H. Zhao, H. Wang, W. Wu, and J. Wei, ”Deployment algorithms for uav airborne networks toward on-demand coverage”, *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, 2018, pp. 2015–2031.

[3] J. Howard, V. Murashov and C.M. Branche, ”Unmanned aerial vehicles in construction and worker safety”, *American Journal of Industrial Medicine*, vol. 61, no. 1, 2018, pp. 3–10.

[4] M. Silvagni, A. Tonoli, E. Zenerino, and M. Chiaberge, ”Multipurpose UAV for search and rescue operations in mountain avalanche events”, *Geomatics, Natural Hazards and Risk*, vol. 8, no. 1, 2018, pp. 18–33.

[5] G. Lindner, K. Schraml, R. Mansberger and J. Hubl, ”UAV monitoring and documentation of a large landslide”, *Applied Geomatics*, vol. 8, no. 1, 2016, pp. 1–11.

[6] Y. Zhang, Z. Su, W. Shen, R. Jia and J. Luan, ”Remote monitoring of heading rice growing and nitrogen content based on UAV images”, *International Journal of Smart Home*, vol. 10, no. 7, 2016, pp. 103–114.

[7] C. Yuan, Z. Liu and Y. Zhang, ”UAV-based forest fire detection and tracking using image processing techniques”, *Proceedings of 2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 639–643.

[8] L. Yu, N. Wang and X. Meng, ”Real-time forest fire detection withwireless sensor networks”, *Proceedings of 2005 International Conference on Wireless Communications, Networking and Mobile Computing*, vol. 2, 2005, pp. 1214–1217.

[9] Y. Huang et al., ”Energy-Optimal Path Planning for Solar-Powered UAVs Monitoring Stationary Target”, *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Safety and Resilience*, 2018, p. 7.

[10] B. R. Christensen, ”Use of UAV or remotely piloted aircraft and forward-looking infrared in forest, rural and wildland fire management: evaluation using simple economic analysis”, *New Zealand Journal of Forestry Science*, vol. 45, no. 1, 2015, p. 16.

[11] F. Ofli, P. Meier, M. Imran, C. Castillo, D. Tuia, ”Combining human computing and machine learning to make sense of big (aerial) data for disaster response”, *Big Data*, vol. 4, no. 1, 2016, pp. 47–59.

[12] G. N. Rao, P. J. Rao and R. Duvvuru, ”A drone remote sensing for virtual reality simulation system for forest fires: semantic neural network approach”, *Proceedings of IOP Conference Series: Materials Science and Engineering*, vol. 149, no. 1, 2016, p. 12011.

[13] V. G. Ambrosia, S. Wegener, T. Zajkowski, D. V. Sullivan, S. Buechel, ”The Ikhana unmanned airborne system (UAS) western states fire imaging missions: from concept to reality (20062010)”, *Geocarto International*, vol. 26, no. 2, 2011, pp. 85–101.

[14] V. Sherstjuk, M. Zharikova, and I. Sokol, ”Forest Fire-Fighting Monitoring System Based on UAV team and Remote Sensing”, *Proceedings of 2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO)*, 2018, pp. 663–668.

[15] K. A. Ghamry, M. A. Kamel and Y. Zhang, ”Cooperative forest monitoring and fire detection using a team of UAVs-UGVs”, *Proceedings of 2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016, pp.1206–1211.

[16] C. Yuan, Z. Liu and Y. Zhang, ”Fire detection using infrared images for uav-based forest fire surveillance”, *Proceedings of IEEE 2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 567–572.

[17] Y. Zhao, J. Ma, X. Li, and J. Zhang, ”Saliency detection and deep learning-based wildfire identification in UAV imagery”, *Sensors*, vol. 18, no. 3, 2018, p. 712.

[18] A. Vijayendra, S. K. Saksena, R. M. Vishwanath and S. N. Omkar, ”A Performance Study of 14-Channel and 5-Channel EEG Systems for Real-Time Control of Unmanned Aerial Vehicles (UAVs)”, *Proceedings of Second IEEE International Conference on Robotic Computing (IRC)*, 2018, pp. 183–188.

[19] J. Hosang, R. Benenson and B. Schiele, ”Learning non-maximum suppression”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2017.

[20] A. Filonenko and K. H. Jo, ”Fast fire flame detection on videos using AdaBoost and parallel processing”, *Proceedings of the 2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, 2018, pp. 645–650.

[21] F. Yuan, Z. Fang, S. Wu, Y. Yang and Y. Fang, ”Real-time image smoke detection using staircase searching-based dual threshold AdaBoost and dynamic analysis”, *IET Image Processing*, vol. 9, no. 10, 2015, pp. 849–856.

[22] S. Ren, K. He, R. Girshick and J. Sun, ”Faster R-CNN: towards real-time object detection with region proposal networks”, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, 2017 pp. 1137–1149.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, ”SSD: Single shot multibox detector”, *Proceedings of European Conference on Computer Vision*, 2016, pp. 21–37.

[24] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, ”You Only Look Once: Unified, real-time object detection”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[25] M. Grega, A. Matiolanski, P. Guzik and M. Leszczuk, ”Automated detection of firearms and knives in a CCTV image”, *Sensors*, vol. 16, 2016, no. 1, p. 47.

[26] S. Guennouni, A. Ahaitouf and A. Mansouri, ”A comparative study of multiple object detection using Haar-like feature selection and local

binary patterns in several platforms”, *Modelling and Simulation in Engineering*, 2015, p. 17.

[27] Q. Zhang, G. Lin, Y. Zhang, G. Xu and J. Wang, ”Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images”, *Procedia Engineering*, vol. 211, 2018, pp. 441–446.

[28] H. Liau, Y. Nimmagadda and Y. Wong, ”Fire SSD: Wide fire modules based single shot detector on edge device”, Web: http://arxiv.org/abs/1806.05363.

[29] D. Shen, X. Chen, M. Nguyen and W. Q. Yan, ”Flame detection using deep learning”, *Proceedings of the IEEE 4th International Conference on Control, Automation and Robotics (ICCAR)*, 2018, pp. 416–420.

[30] J. Cruz, E. Shiguemori and L. Guimaraes, ”A comparison of Haar-like, LBP and HOG approaches to concrete and asphalt runway detection in high resolution imagery”, *Journal of Computational Interdisciplinary Science*, vol. 5, 2015, no. 4, pp. 121–136.