ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 12, Issue 1, January 2025

Improvement of Software Tools for Monitoring Probable Fires in Buildings and Structures

Q.M.Murtazayev, B.B.Choriyev

Head of the Department, Department of General Professional and General Scientific Sciences, Academy of the Ministry of Emergency Situations, Tashkent, Head of the Department of General Professional and General Scientific Sciences, senior teacher, Academy of the Ministry of Emergency Situations of the Republic of Uzbekistan, Doctor of Technical Sciences, Associate Professor, Tashkent, Uzbekistan

ABSTRACT: This article presents a detection algorithm based on modern technologies and artificial intelligence systems for monitoring potential fires in buildings and structures. Additionally, early warning software has been developed, and image analysis conducted. The proposed system provides information on the rapid detection of fire sources based on video footage and sending alert signals. Unlike other notification systems, it quickly identifies distant fire sources and sends timely alerts.

I. INTRODUCTION

Fire detection at the initial stage effectively prevents the spread of the fire and reduces the damage caused by the fire. Various smoke detectors and fire detectors are widely used in internal environments to report possible fires. However, software sensors have a number of limitations in this process. They require proximity to sources of fire in order to operate in an external environment. In addition, they typically require smoke from a fire to be released and then ignited for a while to trigger the alarm, and cannot provide information about the location and extent of the fire. Hundreds of video cameras are installed in buildings and structures, and there are several observation screens in the control room [1].

We use the “OpenCV” library of the python program to monitor possible fires in buildings and structures. Let's explore this library [2].

The OpenCV program was started by Gary Bradsky at Intel in 1999, and its first version was released in 2000. Vadim Pisarevsky joined Gary Bradsky to lead Intel's OpenCV software team in the Russian Federation. In 2005, the “OpenCV” was used on the Stanley car that won the 2005 DARPA Grand Challenge. Subsequently, its active development continued under the support of Willow Garage, where the project was managed by Gary Bradsky and Vadim Pisarevsky. Currently, “OpenCV” supports many algorithms related to computer vision and machine learning, and it is expanding day by day.

Currently, “OpenCV” supports a variety of programming languages, including C++, Python, Java, and others, and is available on various platforms such as Windows, Linux, OS X, Android, iOS. In addition, CUDA and OpenCL-based interfaces for high-speed GPU operations are also being actively developed.

Compared to other languages such as C/C++, Python runs slightly slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us write large-scale code in C/C++ and create a Python wrapper for them. As a result, it becomes possible to use these packages as a Python module. This provides two advantages: firstly, our code runs as fast as the original C/C++ code (because the original C++ code runs in the background mode), and secondly, coding in Python is very easy. OpenCV Python works in the same way, which is a Python convolution based on the original C++ implementation [3].

Numerous techniques for fire detection based on images have been developed, thanks to various recognition algorithms proposed based on a series of studies. Traditional fire detection methods have been effectively replaced by video-based smoke detection methods. These methods surpass traditional methods due to the following advantages:

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 12, Issue 1, January 2025

- early fire detection;
- quick response;
- absence of spatial boundaries;
- the possibility of receiving information about the development of the fire through live video;
- the visual system is capable of providing evidence during a fire investigation [3, 4].

II. DATA TO BE ADDED TO LAYERS

We explore continuous fire safety monitoring in buildings and structures using a CNN (Convolutional Neural Network) model. The input image captured by cameras is resized to a width of 256 pixels, height of 256 pixels, and depth of 3 pixels, then filtered across six layers. In the first layer, 96 kernels of size 11×11×3 are applied with a filter stride of 4 pixels.

The results of this layer are pooled, significantly reducing data complexity and dimensionality. The second layer applies 256 kernels of size 5×5×64 with a stride of 2 steps, followed by pooling of the outputs. In the third layer, 384 kernels of size 3×3×256 are used, and the outputs are pooled. The remaining layers use filters with 384 and 256 kernels, combining the results accordingly. After the fifth layer, a pooling layer applies a 3×3 filter. Finally, the classification stage is performed using two fully connected layers, each containing 4096 neurons. The output layer consists of two neurons that classify the final output as either “fire” or “no fire” [1].

Max pooling is a layer used in convolutional artificial networks (CNNs) that is used in artificial network models used for image classification or computer vision tasks. This layer is located after the convolutional layer and is used to reduce the spatial dimensions of the feature map emitted by the convolutional layer while maintaining the most important properties.

III. DIAGRAMMATIC REPRESENTATION

The program's block diagram is as follows (Fig.1):

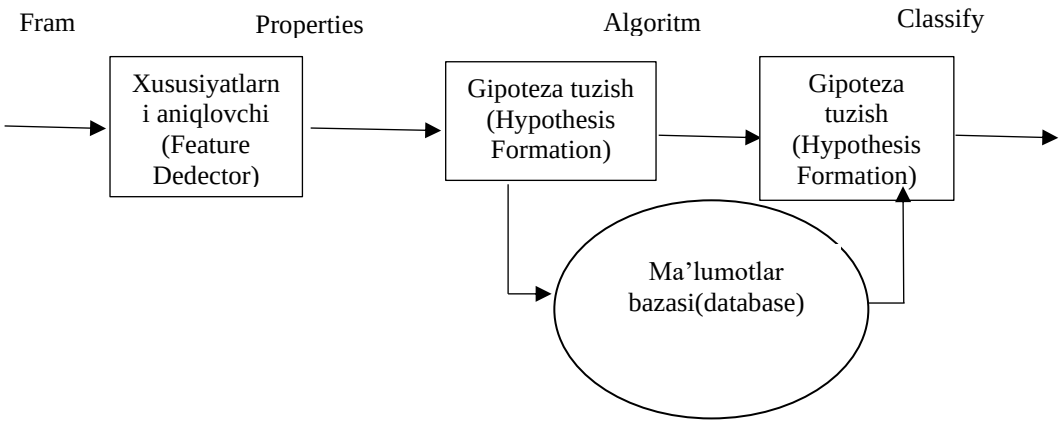


Fig.1 The algorithm of the software system

The camera will be installed where it needs to be monitored. As soon as it is connected to the computer through the network, the frame image is processed using OpenCV. The camera takes the image from the captured video, followed by background separation, segmentation, and other processing. The camera must be positioned correctly so that the entire area of the observed area can be captured [5].

The software operation sequence is as follows (Fig. 2):

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 12, Issue 1, January 2025

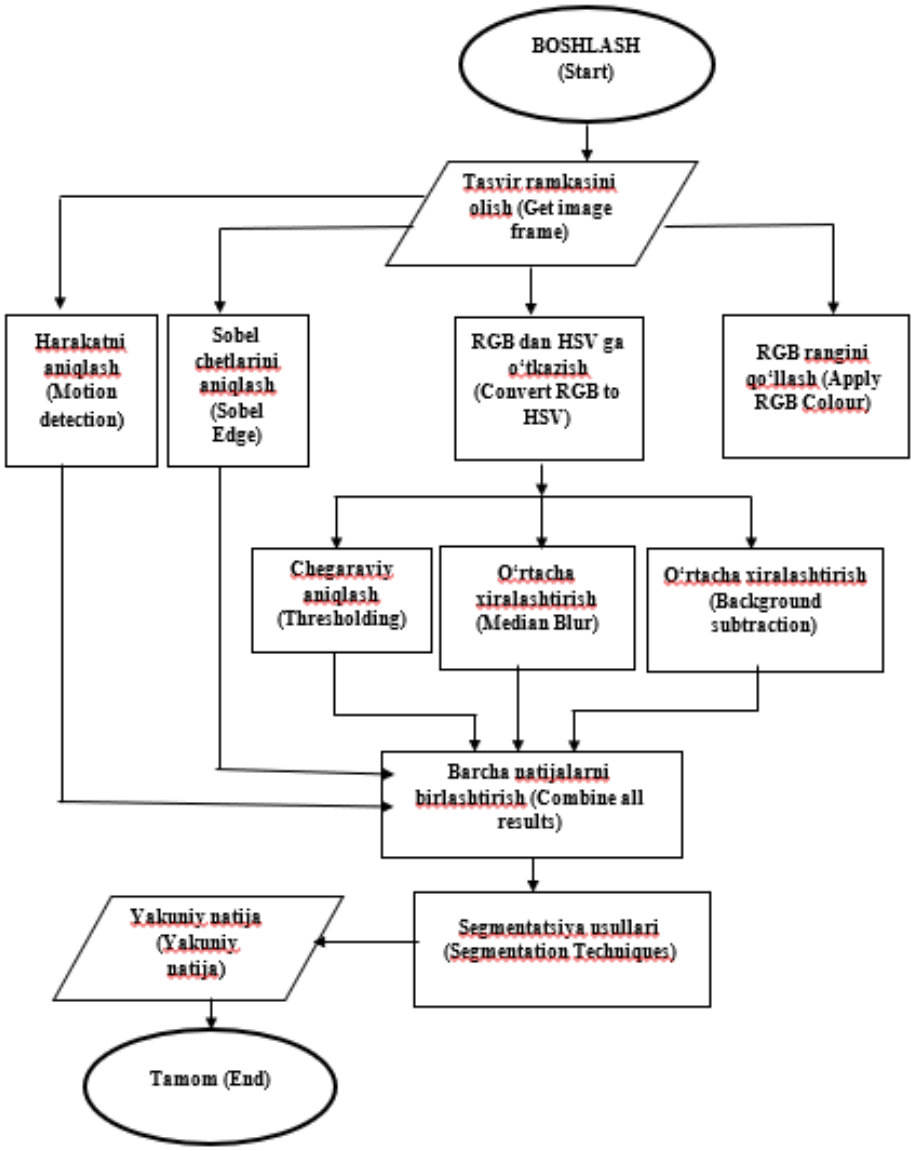


Fig. 2 The basic algorithm of the software system

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 12, Issue 1, January 2025

IV. INPUT DESIGN

This approach is a fire detection algorithm that is free from sensors found in typical fire detection systems. The goal of the dissertation is to create a system that can detect a fire as early as possible from a live video stream. The proposed system is designed to detect a fire when it is small and has not reached a large volume. Also, hardware is minimal and uses pre-existing equipment, which saves costs. It also reduces costs by avoiding temperature and heat sensors. According to the results obtained, the system demonstrated its effectiveness in fire detection. This system consists of a combination of various fire detection algorithms [3, 5, 6].

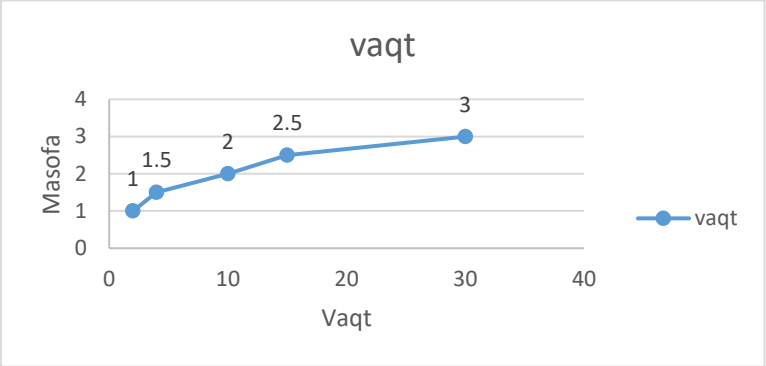
V. OUTPUT DESIGN

The table below shows a table of test results for detecting fires occurring at different distances, depending on the distance and time, and a graph of their change (Table 1).

Table 1

distance (meter)	1-2	3-4	5-10	10-15	15-30	30-40	40-60
Time (second)	1	1,5	2	2.5	3	3.5	3-4

The data graph shown in the table above is:



VI. TESTING OF SYSTEM

To evaluate the proposed model, an experiment was conducted at the Academy of the Ministry of Emergency Situations and the 22nd fire and rescue unit of the "Yangikhayot" district of Tashkent. In this experiment, we used a collection of video clips depicting different fire scenarios in different environments. This data set provides a realistic representation of various fire events, including different sizes, types, and intensities of fire. Each video sequence is annotated with key real markers indicating the presence of fire, which allows for the training and evaluation of fire detection models [2].
The developed program, as a result of processing images obtained from continuous video images, confirms that the suspected event is a fire, provides information about fire parameters and a fire warning signal at the observed object.

VII. DEVICE DESCRIPTION

The proposed CNN model was implemented using OpenCV, TensorFlow, and Keras libraries in the Python programming language. The machine features used to train the model are: Intel® Core i5-13420H CPU @ 2.10GHz or 2.80 GHz, Windows operating system and GTX 1080 graphics card. Table 2 presents the system's features in tabular form.

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 12, Issue 1, January 2025

Table 1

T/r	Name	Experimental environment parameters
1.	Operation system	Windows 10
2.	Processor	Intel® Core i5-13420H CPU @ 2.10GHz yoki 2.80 GHz
3.	Video device	Protsessor bilan birga 1 GB
4.	RAM	8 GB
5.	Software	Python 3.9
6.	Libraries	OpenCV, TensorFlow, Keras

VIII. TESTING OF SYSTEM

A total of 1,200 images were used in our research, with a positive result of 89.7% of the images.

We compared our proposed fire alarm system with existing advanced fire alarm systems. The level of our system is shallow (shallow network), and the number of parameters used in it is small, which is one of its main advantages. This model takes up only 6.47MB of disk space. It should be noted that other highly effective fire detection solutions can be found among scientific works. How effective one method is than another depends more on the tools used and the data set used.

Various previously developed models are capable of detecting 4-5 frames per second on simple, inexpensive hardware platforms, but they require more hard disk space. The software package, created based on our CNN model, can detect a fire up to 24 frames per second in real time, providing speed close to a person's visual abilities. This strong combination increases the model's efficiency.

IX. RESULTS

In our study based on the CNN model, a number of indicators are used for evaluation, such as accuracy (aniq), level of accuracy (adarajasi), and return (qaytar). Accuracy shows how accurate predictions were made in comparison with actual data, while adaraja reflects the proportion of accurate predictions in total predictions. adaraj and return are calculated using the following equations. (Formula 1)

$$A = \frac{aniq}{aniq + qaytar} \qquad 1$$

A set of fire images was used to test the completed research work. The data set includes fire and non-fire images. We used 89.7% of the data to confirm the research and 10.3% to evaluate it. The “TensorFlow” and “Keras” libraries were used in the Python programming language.

X. CONCLUSION

In nature, more severe weather conditions may occur. Like rainy days, snowy days, or stormy weather. Due to the limited data set, the research in this article is limited to fires. In the future, we will study even more harsh weather conditions. Experimental results have shown that the fire image purification network can improve the quality of fire images and enhance the accuracy of fire detection models, which is of significant practical importance for the safe performance of the tasks of the ultra-high-voltage converter station.

REFERENCES

[1]. Q.M.Murtazayev, B.B.Choriyev bino va inshootlarda sodir bo‘lgan yong‘inlarni zamonaviy xabarlash tizimlari asosida aniqlashning usul va vositalarini takomillashtirish. yong‘in-portlash xavfsizligi jurnali 2023-yil dekabr.

ISSN: 2350-0328

International Journal of AdvancedResearch in Science, Engineering and Technology

Vol. 12, Issue 1, January 2025

[2]. Q.M.Murtazayev, B.B.Choriyev Improvement of methods and means of rapid notification of possible fires in buildings and constructions. Science and innovation international scientific journal volume 3 issue 12 december 2024, 37-43

[3]. Amit Hatekar, Saurabh Manwani, Gaurav Patil, Akshat Parekh. Fire Detection on a Surveillance System using Image Processing.

[4]. Bay, H., Ess, A., Tuytelaars, T., & Gool, L. V. (2008). Speeded-up robust features (SURF). Computer Vision and Image Understanding, 110 (3), 346–359. doi: 10.1016/ j.cviu.2007.09.014.

[5]. Sonali K. M., Poojashree M., Shilpashree A., Sindhu N., Rekha K. S.,"Automated Fire Detection Surveillance System", CSE, NIE, Mysuru,International Journal of Current Trends in Engineering & Research (IJCTER) e-ISSN 2455– 1392 Volume 2 Issue 6, June 2016 pp. 49 – 52.

[6]. T. 3elik, H. Цzkaramanlı and H. Demirel, "Fire and smoke detection without sensors: Image procecessing based approach," in *Proc. of 15th European Signal Processing Conf.*, Poznan, Poland, pp. 1794–1798, 2007.

[7]. Saeed, F., Paul, A., Karthigaikumar, P., & Nayyar, A. (2020). Convolutional neural network-based early fire detection. *Multimedia Tools and Applications*, *79*(13), 9083-9099.

[8]. Gotthans, J., Gotthans, T., & Marsalek, R. (2020, April). Deep convolutional neural network for fire detection. In *2020 30th International Conference Radioelektronika (RADIOELEKTRONIKA)* (pp. 1-6). IEEE.

[9]. Frizzi, S., Kaabi, R., Bouchouicha, M., Ginoux, J. M., Moreau, E., & Fnaiech, F. (2016, October). Convolutional neural network for video fire and smoke detection. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society* (pp. 877-882). IEEE.