Improvement of Artificial Intelligence-Based Software Tools for Early Fire Detection and Rapid Notification in Buildings and Structures

Q. M. Murtazayev; B. B. Choriyev

DSc (Doctor of Technical Sciences), Associate Professor
Head of Department,
Academy of the Ministry of Emergency Situations of the Republic of Uzbekistan

Associate Professor of the Department,
Academy of the Ministry of Emergency Situations of the Republic of Uzbekistan

Publication Date: 2026/09/19

Abstract: This article addresses the use of artificial intelligence-based software tools for the early detection of fires that may occur in buildings and structures and for the rapid notification of the parties concerned. Within the scope of the study, the real-time analysis of images and video streams obtained from video surveillance systems is examined, together with the possibilities of automatically detecting the earliest signs of fire, in particular smoke and flame. To improve the accuracy and speed of the fire-detection process, the advantages of software approaches based on computer vision and deep learning technologies are analyzed. In addition, the automation of the processes for promptly delivering information on a detected fire to responsible persons and for issuing timely notifications is described. The results of the study are aimed at improving artificial intelligence-based software tools that serve to ensure fire safety in buildings and structures, to detect fires at an early stage, and to reduce their adverse consequences.

Keywords: *Fire, Buildings and Structures, Early Detection, Rapid Notification, Artificial Intelligence, Software Tools, Computer Vision, Deep Learning, Video Surveillance, Smoke Detection, Flame Detection, Real-Time Mode, Fire Safety.*

How to Cite: Q. M. Murtazayev; B. B. Choriyev (2026) Improvement of Artificial Intelligence-Based Software Tools for Early Fire Detection and Rapid Notification in Buildings and Structures. *International Journal of Innovative Science and Research Technology*,11(9), 518-526.
https://doi.org/10.38124/ijisrt/26sep274

I. INTRODUCTION

Nowadays, the growing complexity of buildings and structures, the increasing number of people occupying them, and the widespread use of various technological equipment and combustible materials are further increasing the urgency of ensuring fire safety. When a fire occurs, detecting it at an early stage, providing timely warning of the danger, and taking the necessary prompt measures are of great importance for protecting human life and health, as well as for reducing material damage. For this reason, improving reliable, rapid, and automatic fire detection and notification systems is one of the priority directions of modern fire safety.

II. LITERATURE REVIEW

The monitoring of fires that may occur in buildings and structures has been addressed in the studies of foreign researchers such as Li P., Jeon M., Jadon A., Zhang Q., Kumar M., Saeed F., Xu J., Xu L., Guo H., Jiao Z., Zhang Y., Xin J., Mu L., Yi Y., Liu H., Liu D., Dua M., Charan G.S., Paul A., Karthigaikumar P., Nayyar A., Gotthans J., Gotthans T., Marsalek R., Frizzi S., Kaabi R., Bouchouicha M., Ginoux J.M., Moreau E., and Fnaiech [5–13].

The rapid development of information and communication technologies and artificial intelligence is creating new opportunities for solving this problem. In particular, the use of computer vision and deep learning algorithms makes it possible to automatically analyze images and video streams obtained from video surveillance cameras, to detect features characteristic of smoke and flame, and to assess the probability of fire in real time.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26sep274

Recent studies note that convolutional neural networks and algorithms of the YOLO family are being applied as one of the most promising technologies for detecting fire and smoke in images and video.

An important advantage of artificial intelligence-based software tools is that they make it possible not only to detect the presence of a fire, but also to continuously analyze the visual information from the monitored object, classify the signs of danger, and transmit information on the detected condition to rapid-notification systems. Such an approach can serve to detect a fire at its initial stage of development, reduce the number of false alarms, and accelerate the decision-making process. Studies conducted by NIST have also shown that technologies based on artificial intelligence and machine learning hold significant promise for assessing and predicting fire risk in real time and for supporting practical decision-making. [1]

At the same time, in the practical implementation of artificial intelligence-based fire-detection software, issues such as varying lighting conditions, the presence of objects resembling smoke and flame, changes in image quality, the movement of objects, observation from different distances, and ensuring high speed in real time remain relevant. These factors directly affect the accuracy, reliability, and computational efficiency of fire-detection algorithms.

III. RESEARCH METHODOLOGY

The developed software was implemented on the basis of the Python programming language, using the OpenCV library to perform computer-vision tasks and the YOLO model for object detection and classification.

In fire-detection tasks, the YOLO model is significant in that it enables the rapid detection of objects such as flame and smoke. According to the model's basic operating principle, the input image is divided into an $n \times n$ grid, and the grid cell in which the center of an object is located is responsible for detecting that object. Each grid cell generates a predetermined number, b, of bounding boxes, for which the coordinates (x, y, w, h), the detection confidence, and the object-class probability (c) are calculated. [5]

In the YOLO model, the processes of object classification and localization are treated as a single regression problem. This approach makes it possible to evaluate the accuracy of each candidate bounding box through a loss function and to iteratively learn the network parameters using the backpropagation algorithm. As a result, in the final output, the fire regions are marked with bounding boxes on the image, and their confidence levels and class-probability values are determined.

The YOLOv8 algorithm was developed in 2023 by the Ultralytics team led by Glenn Jocher as a further improved generation of the YOLO family. This model is regarded as one of the most advanced object-detection algorithms in the field of artificial intelligence and computer vision, offering high accuracy, fast processing, and additional capabilities such as segmentation and classification. Owing to its optimized architecture, the YOLOv8 model enables efficient operation in real time.

Within the scope of this study, the YOLOv8s configuration of the YOLOv8 algorithm was used to detect fire conditions. This version was selected owing to its lightweight architecture, its relatively low computational-resource requirements, and its suitability for real-time operation. These characteristics are of great importance for ensuring speed and stability in fire-detection systems.

This architecture makes it possible to detect objects such as flame and smoke at various distances and sizes during the fire-detection process, thereby increasing the overall accuracy of the system and its efficiency in real time.

The results obtained by uploading images of the fires that occurred into the developed program are presented below (Figure 1).

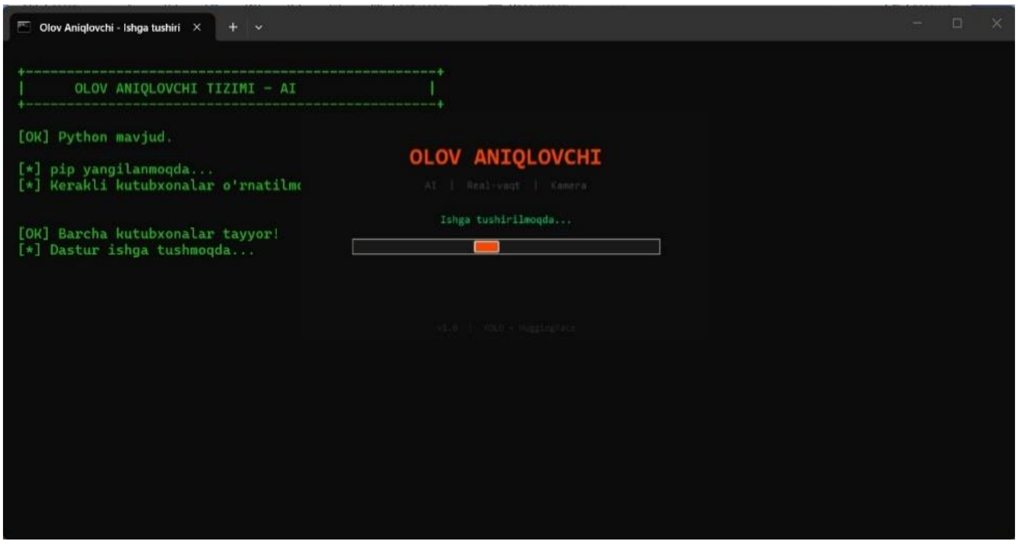

Fig 1. Program Launch Process

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26sep274

To verify how quickly and accurately the program can locate a fire source in real time, an initial experiment was conducted using an artificial flame source (a cigarette lighter) (Figure 2).

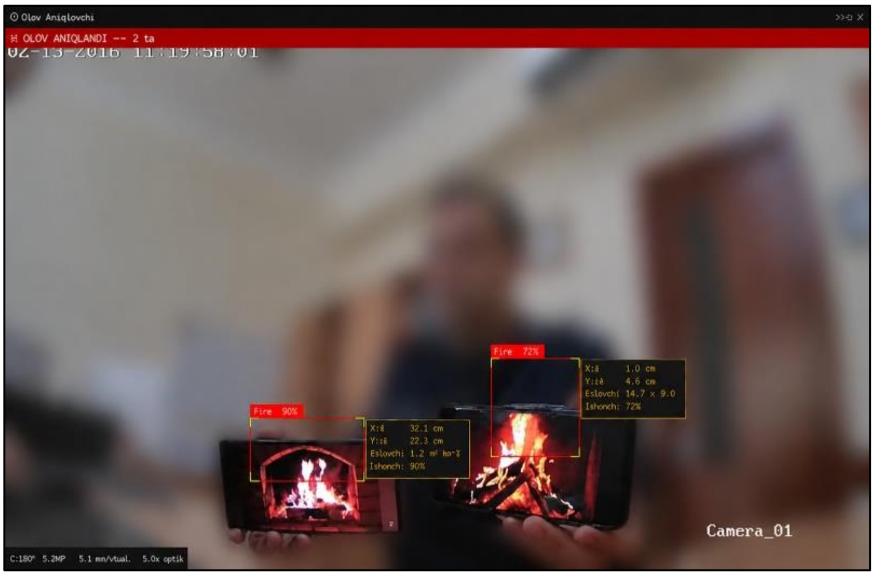

Fig 2. Program Operation Process

When images of flame captured on two separate smartphones were introduced into the camera's field of view, the proposed software system identified each flame source independently of the other. For each flame, individual geometric dimensions (height, width, and area, in mm) as well as confidence percentages were calculated (Figure 3).

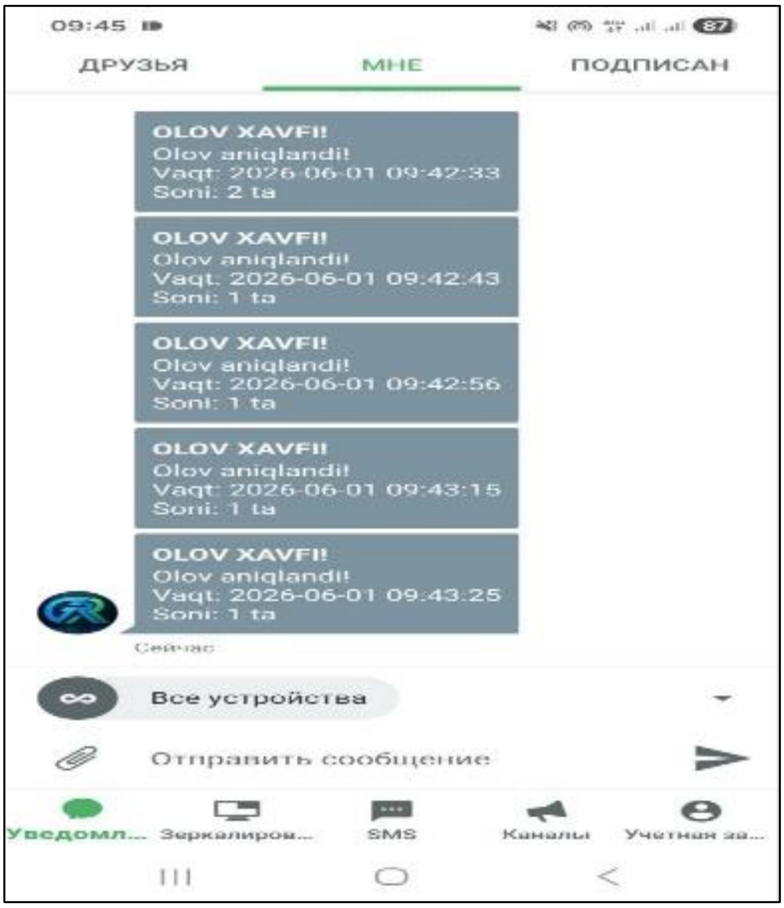

Fig 3. Process of the Program Detecting Multiple Flames and Issuing Notification

The program required 1 second to detect a fire at a distance of up to 2 meters.

IV. RESULTS AND DISCUSSION

➢ *Real-Time Fire Detection Performance at Varying Distances*

The results obtained by testing fire detection at various distances, together with the corresponding table and graph showing the relationship between distance and time, are presented below (Table 1).

Table 1. Values of Time Change Depending on Distance

Distance (m)	1–2	3–4	5–10	10–15	15–30	30–40	40–60
Time (s)	1	1.5	2	2.5	3	3.5	3.5–4.5

The graph of the change in time is presented below (Figure 4):

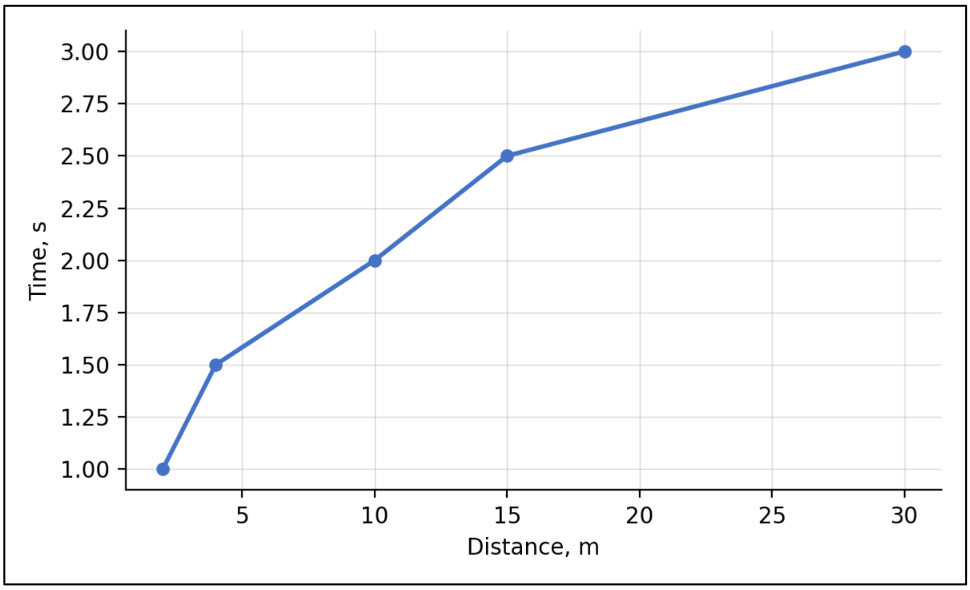

Fig 4. Graph of the Change in Detection Time Depending on Distance

A comparative table and graph of the relationship between distance and time for modern smoke detectors and the model we developed are presented below (Table 2, Figure 5):

Table 2. Comparison of Detection Time Between Modern Smoke Detectors and the Developed Program, Depending on Distance

Distance (m)	1–2	3–4	5–10	10–15	15–30	30–40	40–60
Our model (s)	1	1.5	2	3	4	5	5–7
Smoke detectors (s)	60	90	180	180	420	–	–

As can be seen from the table above, the fire-detection times of modern smoke detectors and the program we developed differ sharply from one another depending on distance.

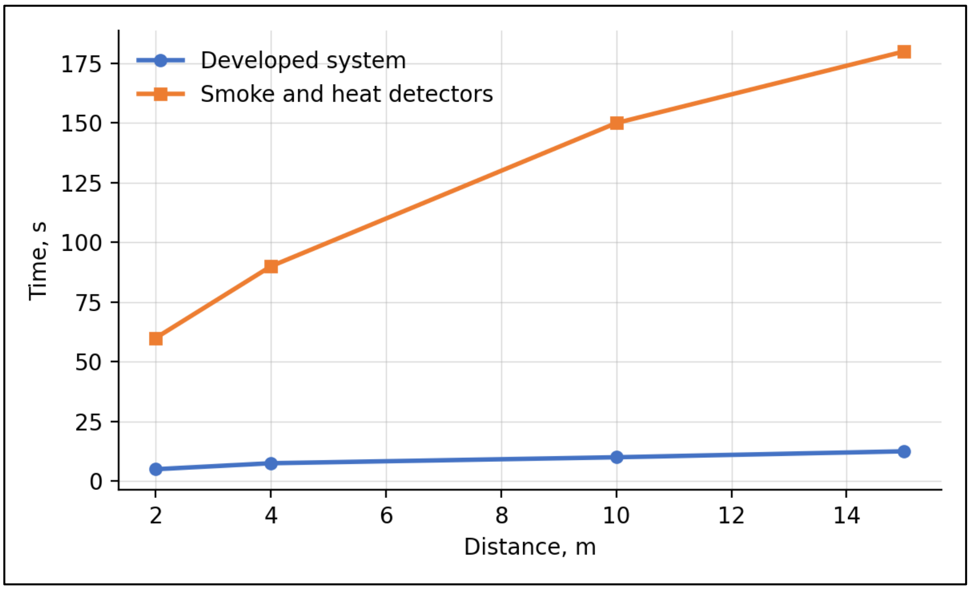

Fig 5. Fire-Detection Times of Modern Smoke Detectors and the Developed Program as a Function of Distance

By performing processing operations on the images extracted from the continuously incoming video stream, the developed software confirms whether a detected suspicious event is indeed a fire, generates information on the main parameters of the fire, and produces an automatic warning signal indicating that a fire hazard has arisen for the monitored object. [2]

➢ *Software System Implementation and Field Testing*

Introducing early fire-hazard detection and rapid-notification systems in buildings and structures is of great importance for protecting human life and health, preserving material assets, and minimizing the consequences of fire. In modern fire safety systems, rapid-notification and warning software makes it possible to detect hazards at the earliest stages of a fire, to promptly inform the relevant services, and to organize evacuation processes effectively.

The process of rapid fire notification and warning consists of a set of organizational, engineering-technical, and software measures aimed at detecting the probability or onset of a fire, transmitting information in real time, and promptly informing fire-rescue units and persons present at the site. This process serves to carry out timely rapid-tactical actions for extinguishing fires, to coordinate emergency-rescue operations, and to ensure the safety of people.

The developed software system is designed for the rapid detection and notification of fires that may occur in buildings and structures, residential areas, entertainment venues, and various other facilities. As soon as the initial signs of fire are detected, this software makes it possible to notify the relevant services and responsible persons in real time.

To evaluate the practical effectiveness of the software system, it was tested under real conditions, that is, in environments where fire-related situations may arise. In this process, a "software tool for the rapid notification of possible fires in buildings, structures, residential areas, and public and administrative buildings" was first developed and put into practice.

To assess the functional capabilities and accuracy of the proposed model, trial tests were carried out at the Academy of the Ministry of Emergency Situations of the Republic of Uzbekistan and at Fire-Rescue Unit No. 22 of the Yangihayot district of Tashkent city. During the experiment, a set of video materials reflecting fire situations under various environments and conditions was used.

This video dataset contains real footage of various fire scenarios, reflecting different sizes, types, and intensity levels of flame. Each video sequence is accompanied by ground-truth annotations indicating the presence of fire. This made it possible to train, test, and comprehensively evaluate the effectiveness of the software model designed for fire detection.

➢ *Dataset, Model Training, and Hardware Configuration*

The proposed convolutional neural network (CNN) model was implemented in the Python programming language, and the OpenCV, TensorFlow, and Keras libraries were extensively used in its development and training. These libraries make it possible to process image and video streams, build deep-learning models, and evaluate their effectiveness.

The processes of training and testing the model were carried out on a personal computer equipped with modern computing resources. Specifically, the computing environment was configured on a technical platform equipped with an Intel® Core™ i5-13420H central processor (2.10 GHz base and 2.80 GHz maximum frequency), the Windows operating system, and an NVIDIA GTX 1080 graphics accelerator. The use of a graphics processor made it possible to accelerate the neural-network training process and to efficiently process large volumes of video data.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26sep274

The main characteristics of the software and hardware tools used in the study are summarized in Table 3 in the form of system parameters.

Table 3. System Parameters of the Software and Hardware Tools Used in the Study

No.	Parameter	Proposed Model	FireNet Model	EfficientNet Model	Deep FireNet Model	Deep FireNet V2 Model
1	Operating system	Windows 10	Windows 10	Linux 20.04 LTS (64-bit) or Windows 10/11 (64-bit)	Linux 20.04 LTS (64-bit) or Windows 10/11 (64-bit)	Linux 20.04 LTS (64-bit) or Windows 10/11 (64-bit)
2	Processor	Intel® Core i5-13420H @ 2.10–2.80 GHz	Intel® Core i5/i7-13420H @ 2.5–2.80 GHz	Intel® Core i7 2.6 GHz and above	Intel® Core i7 2.6–3.2 GHz and above	Intel® Core i7 or equivalent, 2.8–3.4 GHz and above
3	Graphics device (GPU)	Integrated with processor, 2–4 GB	At least 4 GB	CUDA/cuDNN-capable, at least 6–8 GB VRAM	CUDA/cuDNN-capable, at least 8 GB VRAM	CUDA/cuDNN-capable, at least 8–12 GB VRAM
4	RAM	8 GB	8–16 GB	16 GB	16–32 GB	16–32 GB
5	Software	Python 3.9	Python 3.9 and above	Python 3.7 and above	Python 3.8 and above	Python 3.8 and above
6	Libraries	OpenCV, TensorFlow, Keras	TensorFlow, PyTorch, OpenCV, NumPy	TensorFlow, PyTorch, OpenCV, NumPy, Matplotlib	TensorFlow, PyTorch, OpenCV, NumPy, Matplotlib	TensorFlow, PyTorch, OpenCV, NumPy, Matplotlib
7	Approximate price	8,000,000–9,500,000 UZS	9,000,000–11,000,000 UZS	14,000,000–16,000,000 UZS	16,000,000–18,000,000 UZS	18,000,000–22,000,000 UZS

➢ *Computational Resource and Cost Comparison*

Based on the data in the table above, it can be noted that the computer resources required to implement the proposed model are relatively lower compared with other models. In particular, an Intel® Core i5 processor, 8 GB of RAM, and a graphics device with 2–4 GB of video memory, either integrated with the processor or discrete, are sufficient for the proposed model — a configuration available even on modern mid-range computers.

The FireNet, EfficientNet, and Deep FireNet models, in contrast, require processors with high computing power, 16–32 GB of RAM, and graphics processors with 6–12 GB of video memory supporting CUDA and cuDNN technologies. Such hardware requirements significantly increase the overall cost of the system.

According to approximate estimates, the cost of the computer system required to run the proposed model is on average about 8–9.5 million UZS, whereas the hardware required for the FireNet model costs 9–11 million UZS, for the EfficientNet model 14–16 million UZS, and for the Deep FireNet models 16–22 million UZS.

Thus, compared with the other models considered, the proposed system not only requires fewer computing resources, but the technical means needed for its implementation are also considerably more economical and efficient. This broadens the possibilities for putting the system into practice and helps reduce additional financial costs.

➢ *Evaluation Metrics and Model Performance*

In evaluating the effectiveness of the CNN-based system, a number of key performance metrics widely used in machine learning were applied. These included the accuracy rate (the ratio of all instances correctly classified by the model to the total number of samples), the precision rate (the proportion of instances labeled as positive by the model that are actually positive), and the recall rate (the proportion of actual positive instances that were identified by the model). The accuracy metric expresses the overall correctness of the model’s predictions in comparison with the ground-truth data; the precision rate reflects how many of the instances evaluated as positive by the model were in fact correctly identified, while the recall rate shows what proportion of the actual positive instances were detected by the model.

The accuracy, precision, and recall metrics were calculated using the following mathematical expressions (Formulas 1, 2, and 3).

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

$$Precision = TP / (TP + FP) \quad (2)$$

$$Recall = TP / (TP + FN) \quad (3)$$

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26sep274

where:
TP (true positive) — a fire is present, and the developed model also indicates that a fire is present;
TN (true negative) — no fire is present, and the developed model also indicates that no fire is present;
FP (false positive) — no fire is present, but the model indicates a fire (a false alarm);
FN (false negative) — a fire is present, but the model fails to detect it.

In the course of the study, a dataset of more than 1,200 images in total (650 images containing an actual fire and 550 images without fire; for our model, TP = 620, TN = 470, FP = 80, FN = 30) was used. Using these images, our CNN-based model was trained and tested, and according to the experimental results, the model achieved a positive fire-detection rate of 91.6%. This indicator confirms that the proposed software system effectively performs the tasks of rapid fire detection and notification. [3]

To test the performed research work, a dataset consisting of flame images was used. This dataset includes images both containing and not containing flame. In the course of the study, 90.89% of the data was allocated for training the model and validating the results, while the remaining 9.1% was allocated for evaluating the model's effectiveness. [3]

The proposed model was implemented in the Python programming language, and the TensorFlow and Keras libraries were used in its development and training. A stepwise gradient-descent algorithm was applied in training the model, using the "Leaky ReLU" activation function and the "Adam" optimizer. This approach served to ensure fast and stable convergence of the model.

Before training the model, data-preprocessing procedures were carried out. Specifically, the initially obtained images were first resized to match the input dimensions of the developed software system. In the subsequent data-processing stage, various filters and techniques were applied, including normalization, filtering of recurrent noise, removal of insignificant data, and image cleaning. These procedures served to increase the accuracy of the model and to reduce the number of false detections.

The "Leaky ReLU" activation function was used in the hidden layers of both the CNN model developed in this study and the existing fire-detection models selected for comparison. The use of this activation function helps to reduce the "dead neuron" problem, ensure stable gradient propagation, and improve the generalization capability of the model. [3]

➢ *Comparative Evaluation of Fire Detection Models*

The proposed rapid fire-notification software system was comparatively analyzed against existing advanced fire-detection and warning systems. According to the results of the analysis, the architecture of the developed model is relatively shallow, and the small number of trainable parameters it employs is one of its key advantages. This approach reduces the demand for computing resources and expands the possibilities for applying the system in real time.

It was also found that the developed model is efficient in terms of memory footprint, occupying only 6.47 MB of storage space. This indicator makes it possible to apply the system even on devices with limited hardware resources. [3]

It should be noted that other highly effective solutions designed for fire detection also exist within the scope of scientific research. However, the effectiveness of a particular method largely depends on the algorithmic approaches used, the technical tools employed, and the volume and quality of the dataset. For this reason, the advantages of the proposed model are determined specifically by its speed, resource efficiency, and practical applicability under real conditions.

Although most previously developed fire-detection models are capable of processing an average of 4–5 frames per second on simple and relatively inexpensive hardware platforms, they are characterized by high memory-resource requirements. This limits the possibilities of applying these models widely in real time.

The software package developed on the basis of the proposed CNN model, in turn, provides the ability to detect fire in real time at up to 24 frames per second. This indicator is close to the visual-perception speed of the human visual system, enabling continuous and stable monitoring of fire conditions. As a result, the system significantly improves the timeliness of fire-sign detection and the efficiency of the rapid-notification process.

Thus, the combination of a high detection speed with efficient use of computing resources increases the overall effectiveness of the proposed model and characterizes it as a promising solution for practical application.

A comparison of fire-detection models in terms of performance indicators is presented below (Table 4).

Table 4. Comparison of Fire Detection Models by Performance Indicators

Model	Accuracy	Precision	Recall
FireNet	91.56%	0.80	0.84
EfficientNet	91.67%	0.84	0.89
Deep FireNet	91.80%	0.81	0.86

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26sep274

Model	Accuracy	Precision	Recall
Deep FireNet V2	90.46%	0.82	0.89
Proposed system	**90.89%**	**0.88**	**0.95**

As can be seen from Table 4, our model recorded the highest accuracy (90.89%) and the highest recall (0.96) among all the evaluated models. The results show that, compared with the other models evaluated in the study, our model is more effective at detecting fires. [3]

- *Relationship Between Fire Detection Time and Material Damage*

One of the important indicators in evaluating the effectiveness of fire safety systems is the time spent on detecting and extinguishing a fire. Practical experience and scientific research show that the shorter the fire-detection and extinguishing time, the lower the amount of material damage caused to the facility. Conversely, if a fire is not monitored for a long time, its rate of development increases, and this leads to a sharp rise in economic losses.

The process of fire development generally consists of several stages: the incipient stage, the growth stage, the fully developed fire stage, and the decay stage. In the incipient stage, the fire arises in a small area and its rate of heat release is relatively low. At this stage, the probability of quickly detecting and extinguishing the fire is high. However, as time passes, the rate of fire development increases and it transitions to the growth stage; during this process, the heat-release rate rises sharply, and the fire may spread over a large area.

According to research in this field, the process of fire development is often expressed on the basis of a quadratic relationship (as an empirical model of fire growth). According to this relationship, the fire's power increases in proportion to the square of time. Mathematically, this is expressed as follows (Formula 4):

$$Q(t) = \alpha \cdot t^2 \quad (4)$$

where Q(t) is the time-dependent fire power (kW), t is time (s), and α is a coefficient expressing the rate of fire development.

This model shows that as time increases, the power of the fire rises very rapidly. As a result, an uncontrolled fire can engulf a large area within a short period. For this reason, systems for the early detection and rapid extinguishing of fires are of great importance for ensuring fire safety. [3]

The time factor is of particular importance in estimating the amount of material damage caused by a fire. In practical calculations, the amount of damage can be expressed as a function of time using the following simplified mathematical model (Formula 5):

$$D(t) = k \cdot t^n \quad (5)$$

where D(t) is the time-dependent amount of damage, k is a coefficient depending on the value of the facility and the level of hazard, t is the time elapsed until the fire is extinguished, and n is an empirical exponent, usually ranging between 2 and 3.

This model shows that an increase in the time required to extinguish a fire leads to a sharp rise in the amount of material damage. For example, if the fire-extinguishing time doubles, the amount of damage may increase fourfold or more. This demonstrates the economic importance of early fire detection and prompt response measures.

Practical calculations show that extending the extinguishing time at facilities where a fire has occurred from 5 minutes to 15 minutes leads to a significant increase in the amount of damage. The table below presents the approximate effect of fire-extinguishing time on the amount of material damage (Table 5).

Table 5. Damage as a Function of Fire-Extinguishing Time

No.	Extinguishing time	Estimated damage share (%)	Damage level
1	5 minutes	10–20%	Low damage
2	7 minutes	25–35%	Moderate damage
3	10 minutes	50–60%	High damage
4	15 minutes	70–80%	Very high damage

As can be seen from the table, the amount of damage increases rapidly as the fire-extinguishing time increases. In particular, once a fire reaches the fully developed stage at a certain point in its progression, an entire room or structure may become fully engulfed in flame within a short period. In this case, the amount of material damage rises sharply. [3]

For this reason, modern fire safety systems place special emphasis on the early detection of fire. Modern technologies, including video-monitoring systems, artificial intelligence algorithms, and automatic alarm systems, make it possible to detect fires at their earliest stages. This, in turn,

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26sep274

makes it possible to eliminate a fire within a short time and to prevent large amounts of material damage.

V. CONCLUSION

In conclusion, reducing fire detection and extinguishing time is one of the key factors in improving the effectiveness of fire safety systems. Scientific studies show that eliminating a fire within the first 5 minutes can reduce the amount of material damage to a facility by several times. Therefore, the implementation of modern technologies aimed at the early detection of fire is of significant scientific and practical importance for ensuring fire safety.

REFERENCES

[1]. Murtazayev, Q.M., & Choriyev, B.B. (2023). Bino va inshootlarda sodir bo‘lgan yong‘inlarni zamonaviy xabarlash tizimlari asosida aniqlashning usul va vositalarini takomillashtirish [Improvement of methods and means for detecting fires in buildings and structures based on modern notification systems]. Yong‘in-portlash xavfsizligi jurnali [Journal of Fire and Explosion Safety], December 2023.

[2]. Murtazayev, Q.M., & Choriyev, B.B. (2024). Improvement of methods and means of rapid notification of possible fires in buildings and constructions. Science and Innovation International Scientific Journal, 3(12), 37–43.

[3]. Choriyev, B.B. (2026). Bino va inshootlarda sodir bo‘lishi mumkin bo‘lgan yong‘inlarni tezkor xabarlash dasturiy ta’minoti joriy etish va samaradorligini baholash [Implementation and effectiveness evaluation of software for the rapid notification of possible fires in buildings and structures]. Proceedings of the International Scientific-Practical Conference “30 Years of Emergency Situations Authorities: Path of Development, Reforms and Prospects,” 2026, pp. 409–413.

[4]. Wu, H., Wu, D., & Zhao, J. An intelligent fire detection approach through cameras based on computer vision methods.

[5]. Kai-Kuang, M. (2008). Computer vision based fire detection in color images. 2008 IEEE Conference on Soft Computing in Industrial Applications, 258–263.

[6]. Chen, T.H., Wu, P.H., & Chiou, Y.C. (2004). An early fire-detection method based on image processing. ICIP Proceedings — International Conference on Image Processing, 1707–1710.

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

[8]. Luo, P., Tian, Y., Wang, X., & Tang, X. (2014). Switchable deep network for pedestrian detection. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (pp. 899–905). Columbus, OH, USA.

[9]. Vadim, K., Maxime, O., Minsu, C., & Laptev, I. (2016). ContextLocNet: Context-aware deep network models for weakly supervised localization. In European Conference on Computer Vision (Vol. 1, No. 2, pp. 350–365).

[10]. Wang, Z., & Liu, J. (2017). A review of object detection based on convolutional neural network. In 36th Chinese Control Conference (pp. 11104–11109). Dalian, China.

[11]. Çelik, T., Özkaramanlı, H., & Demirel, H. (2007). Fire and smoke detection without sensors: Image processing based approach. In Proceedings of the 15th European Signal Processing Conference (pp. 1794–1798). Poznan, Poland.

[12]. Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84–90.

[13]. Foggia, P., Saggese, A., & Vento, M. (2015). Real-time fire detection for video-surveillance applications using a combination of experts based on colour, shape, and motion. IEEE Transactions on Circuits and Systems for Video Technology, 25(9), 1545–1556.