EJSIT

European Journal of Science, Innovation and Technology

ISSN: 2786-4936

Volume 4 | Number 2 | 2024

Development of a CNN-Based Smoke/Fire Detection System for High-Risk Environments

Nwosu Ifeoma[1], Alagbu Ekene[2*], Okeke Obinna[3], Ikpo Kingsely[4], Onwuamanam Chrysantus[5]

[1, 2, 3]Electronic and Computer Engineering Department, Nnamdi Azikiwe University, Awka, Nigeria

[4]Electrical and Electronics Department, Petroleum Training Institute, Efurun, Delta State, Nigeria

[5]Electrical/Electronics Department, Federal Polytechnic Nekede, Owerri, Nigeria

ABSTRACT

This paper addresses the critical challenge of fire detection in high-risk environments, such as industrial facilities, warehouses, and densely populated areas. These locations face significant fire risks due to inherent processes and flammable materials. While traditional methods like smoke and heat detectors play a role, they have limitations like slow response times, false alarms, and ineffectiveness in large spaces. We propose a novel approach using a Convolutional Neural Network (CNN) for smoke/fire detection in these environments. Our CNN-based system continuously analyses video feeds, enabling faster fire detection compared to traditional methods and easily adapts its detection for scenes with challenging lighting conditions. This paper details the development, training process, and evaluation of the CNN system in simulated high-risk environments. The analysis of the system is performed in terms of accuracy, false alarm rate, and response time. The results demonstrate the potential of CNN technology for improving fire safety and early detection in critical locations, achieving an impressive accuracy of 94.14% and minimal loss of 0.14 during evaluation.

Keywords: fire detection, CNN, deep learning

I: INTRODUCTION

Fire outbreaks pose a significant threat to life, property, and the environment, particularly in high-risk environments. These locations are prone to fire risks due to the nature of essential processes carried out there (such as the operations in oil refineries and chemical plants) and the abundance of flammable materials in the surroundings (Sankarasubramanian et al, 2021). Densely populated buildings, data centers and forest regions are among such locations that require heightened fire safety measures and special monitoring. Fires in these regions can be quite devastating (Avazov et al, 2023). The loss of life is a major worry, with workers, residents, and emergency responders all at danger. Industrial fires may emit dangerous chemicals into the atmosphere and water, causing substantial environmental harm. Furthermore, wildfires may destroy ecosystems and affect the livelihoods of individuals who rely on the land.

The operation of the traditional fire detection methods has been crucial in detection of some fire occurrences but they still present some limitations. Smoke detectors can be slow to detect smoldering fires which produce minimal smoke initially. Also, dust and other airborne particles might trigger false alarms. Heat detectors, which respond to change in temperature, may not be useful in open and high-ceilinged spaces.

* Corresponding Author (Engr Dr Ekene Emmanuel Alagbu, ee.alagbu@unizik.edu.ng)

This paper presents an adaptive solution utilizing a Convolutional Neural Network (CNN) for smoke/fire detection in high-risk environments. CNNs is a deep learning (DL) algorithm that is known for its application and success with image classification and feature extraction tasks (Prince, 2023). A CNN-based system is able to continuously analyze video feeds captured from surveillance cameras, leading to faster fire detection than that being achieved by the traditional systems. The CNN is able to learn the specific visual patterns that identifies smoke and flames, and easily adapts itself for detection in environment with poor lighting conditions and background clutter.

In the following sections of the paper, the development and analysis of the CNN-based smoke/fire detection system will be detailed, examining its potential to ensure fire safety in high-risk environments. Section II provides a detailed literature review of the functionality of the various CNN layers and its application in related research for smoke/fire detection. Section III provides details on the steps involved in the CNN modelling and development. The performance and results can be seen in Section IV. The conclusion and directions for future research is detailed in Section V.

II: LITERATURE REVIEW/CNN OVERVIEW

A CNN is a deep learning (an aspect of AI) algorithm that utilizes neural networks and the convolutional operation for computer vision-based image analysis. They are known by their distinct architecture, made up of convolutional layers, pooling layers and activation functions, which is utilized for automatic extraction of hierarchical representations from visual data. The various layers of the CNN serve various key functions in the operation of the network (Mandal, 2023; Keita, 2023).

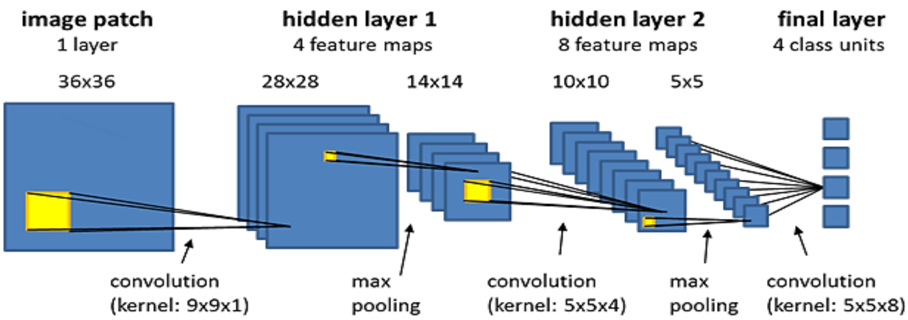

Figure 1. Structure of a 6-layer CNN utilized for classification of an image into any of 4 classes

The convolutional and pooling layers are the fundamental layers of the CNN at which the feature extraction occurs. A filter or kernel is applied on the input image at these layers to develop feature maps which represent the patterns on the image. The major difference between these layers is that the pooling kernel has no weights. The convolution operation at the convolutional layer generates the feature map that corresponds to a value gotten after elementwise multiplication and addition as the kernel passes over the pixel values (Mandal, 2023). The pooling kernel at the pooling layers, usually located between layers, performs dimensionality reduction of the maps to improve the CNN detection efficiency.

Activation functions improve the network's non-linearity and allows it to recognize intricate patterns in the pattern. These functions are incorporated in the various layers to transform the output of the network into a meaningful classification gotten by converting a weighted aggregate of the inputs appropriately. Some common activation functions include ReLU, Sigmoid and tanh functions.

The utilization of CNN is being explored in recent research pertaining to smoke/fire detection as can be seen from published works available. Yavuz Selim et al (2021) utilized transfer learning to perform the comparative analysis of several CNN-based models (Inception V3, SequeezeNet, VGG16 and VGG19) for flame detection. Their work, which presented a 3-stage framework, showed a method for determining the flame mobility by comparing video frames of the fire, thus reducing the damage caused by the fire can be reduced by early detection and timely intervention. Similarly, Lee and Shim (2019), Almoussawi et al (2022), Biswas et al (2023), Satishkumar et al (2023), Khan et al (2019) utilized VGG-16 architecture for early smoke detection. Ryu and Kwak (2022) developed a model for effective detection of smoke and flames. Firstly, their model performs a preprocessing using color conversion and corner detection for flames and dark channel and optical flow for smoke regions. Then, a CNN-based network analyses the region of interest to further certify it as a flame or smoke. Their method ensured an improvement of the detection accuracy by 5.5% for flame and 6% for smoke. The use of CNN-based object detection models such as YOLO and Faster-RCNN was examined in the papers by Xu et al (2022), Wahyono et al (2022), Xu et al (2021) and Avazov et al (2023).

Overall, the research in smoke and fire detection still requires improvements in the design and implementation of a simplified deep learning model for efficient detection and provision of a diverse dataset for detection.

III: CNN DEVELOPMENT AND TRAINING

This section explores the process involved in the implementation of a CNN-based smoke/fire detection system. Python TensorFlow and Keras library is utilized in the creation of the sequential CNN model with the various layers needed for efficient feature extraction and image classification process.

The system is designed such that the image frames captured from surveillance cameras serve as input to the CNN mode (see Figure 2). The detection of the occurrence of fire or smoke in the image triggers an alert that is promptly sent out to emergency personnel for monitoring and assistance. It is recommended to have a human operator monitoring the alerts from a group of cameras in order to ascertain the spread and point of occurrence of an alerted fire, this serves to minimize alerts due to cooking fires or cigarette smoke or other circumstances.

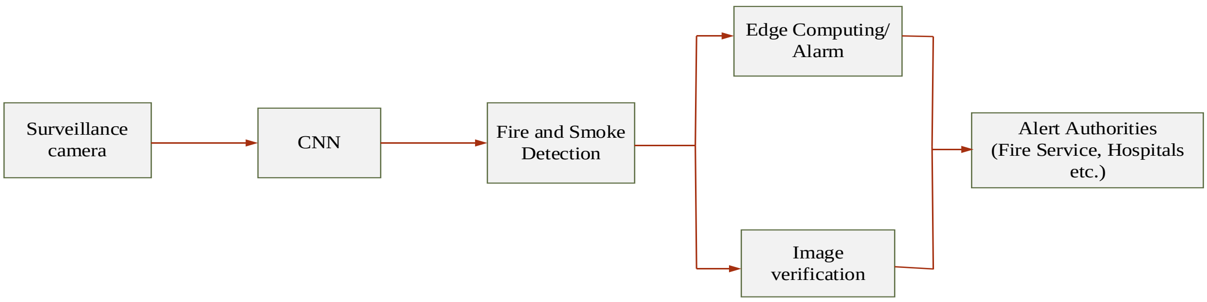

Figure 2. System model for the Smoke and Fire Detection System

Data Collection and Preparation

The data needed for this work required a sufficient number of images that contained fire and/or smoke and those that had neither. This is to ensure that the trained model has no bias towards any of the features from the training data. A custom image dataset was curated from related images sourced from the Internet, Kaggle and other open-sourced datasets. It includes images of fires from diverse settings such as house fires, car fires, industrial fires etc. 1,506 images were gotten of which 1,206 were used for training the model while 300 of them were reserved for validating the model. The dataset partitioning was done in such a way such that about 80% is for training and the remaining 20% was used for testing. The training data was further split into four classes of smoke; fire; smoke and fire; and non-fire and non-smoke images and each class had about 300 images in its category.

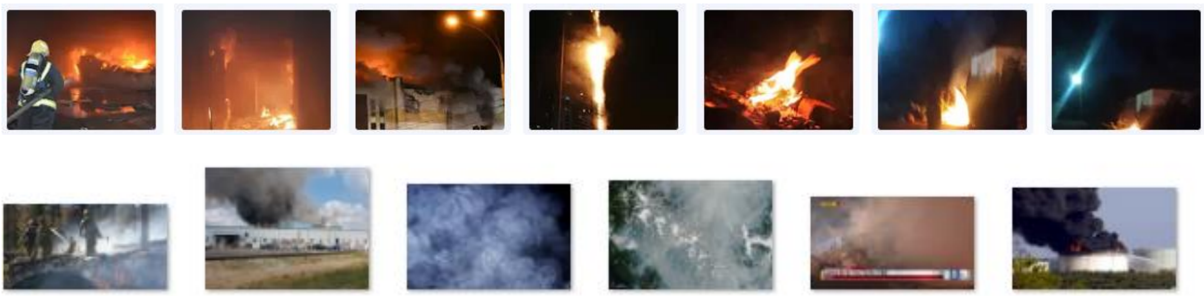

Figure 3. Sample images from the dataset

CNN Training Process

The CNN model processes image frames of $64 \times 64 \times 3$ pixels. This is done to ensure uniformity in the processed data. The input image is subjected to 32 filters with a size of 3×3, thus, producing 32 feature maps in the first convolution layer. The features are extracted during the convolution process using the filter. The feature map, F is represented by the convolution operation between the input image, M and the filter, T as shown in equation (1).

$$F[i,j] = (M * T)_{[i,j]} \tag{1}$$

Max pooling layers with a pool size of 2×2 pixels is used to choose the maximum activations of these 32 feature maps. Finally, the dense layer was used to make the classification of the detected features of the image into any of the four pre-defined classes. The ReLU activation function is used for all the layers except the last dense layer which uses a Softmax activation function. The activation functions serve to introduce bias values to the output of the layers in order to aid the classification process. The algorithmic steps for the developed CNN are as follows:

i. *Import necessary libraries and modules: This includes Keras, sklearn, os, numpy, and cv2.*
ii. *Initialize necessary lists: This includes a list to store models, histories, test predictions, and test labels.*
iii. *Load and pre-process the data:*
    a. *Get the list of all images and their corresponding labels from the data directory.*
    b. *Convert labels to integers and then one-hot encode them.*
    c. *Convert one-hot encoded labels to single labels.*
iv. *Initialize a Sequential model and add layers to it. This includes Conv2D, MaxPooling2D, Flatten, and Dense layers.*
v. *Compile the model with Adam optimizer, categorical cross-entropy loss, and accuracy metrics.*
vi. *Fit the model on the pre-processed data and validate it on the test data.*
vii. *Use the model to predict the test set and store the predictions and actual labels in the lists.*
viii. *Save the model in the list of models.*
ix. *Calculate the average accuracy and loss at each epoch.*

CNN Performance Metrics

A deep learning classification model can be accessed based on certain criteria (metrics). The metrics relevant to this study includes the Accuracy, Recall and False Positive Rate (FPR) and these are calculated using four values gotten from the confusion matrix computed at the end of the training and validation of the model. This matrix has values labelled as True Positive

(TP), False positive (FP), False negative (FN) and True negative (TN) and it is used to compute the following metrics:

$$Accuracy = \frac{(TP+TN)}{TP+TN+FP+FN} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$False\ Positive\ rate = \frac{FP}{FP+TN} \quad (4)$$

The accuracy computes the number of correct predictions out of the sample data. Recall or True Positive Rate (TPR) computes the ratio of the positive predictions that correctly classified as positive out of the positive sample data. The FPR computes the ratio of the negative predictions that incorrectly classified as positive out of the negative sample data.

IV: RESULTS

An investigation of the accuracy of the model detailed a value of 94.14% for the validation accuracy at the training while the training accuracy reached a maximum of 95% (Figure 4). This means that the model is able to make 94 correct predictions out of every 100 images. The absence of a plateau in the accuracy values over the 200 epochs training period indicates an absence of overfitting in the model.

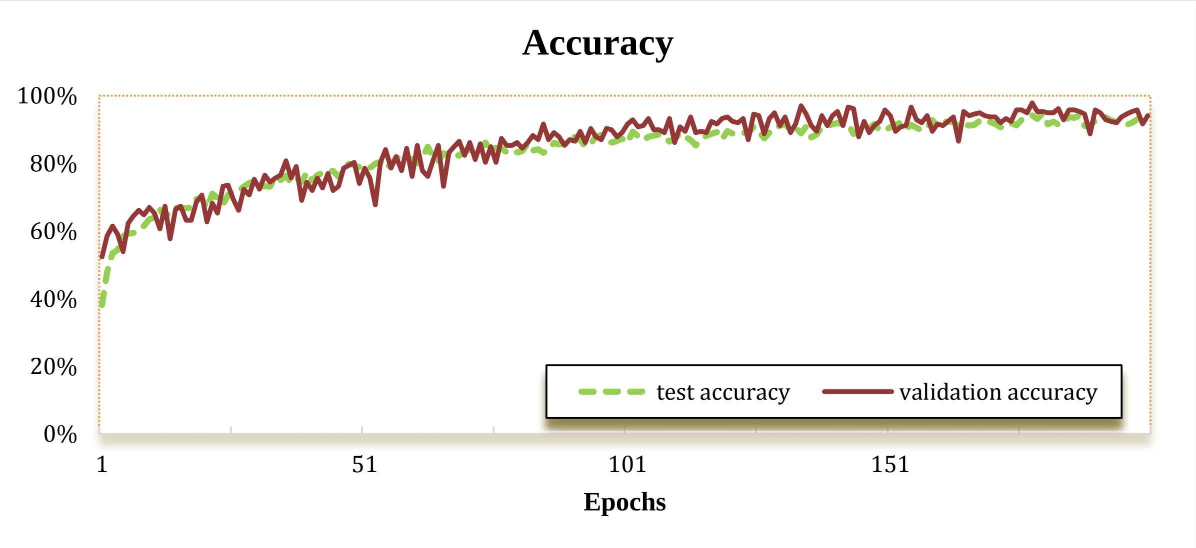

Figure 4. The graphical representation of the developed model's accuracy on the test data and validation data

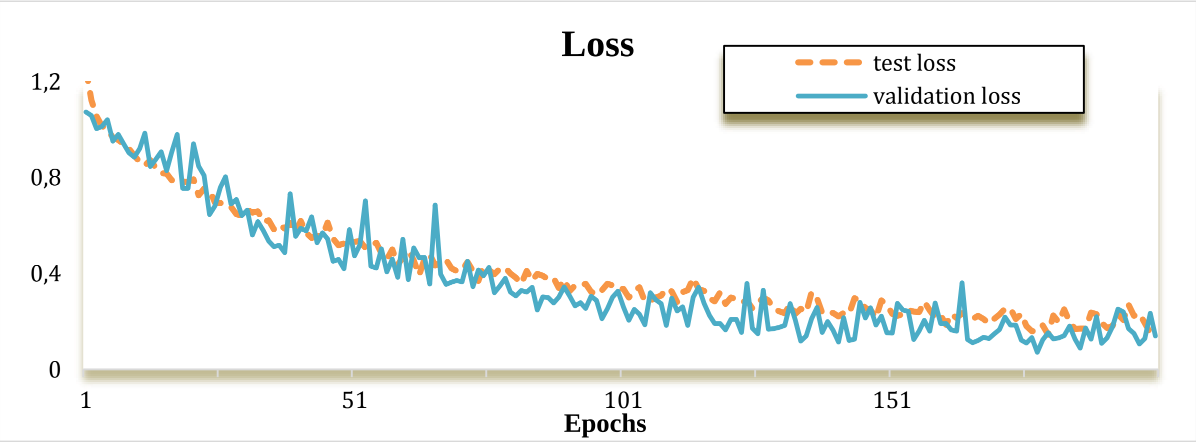

Figure 5. The graphical representation of the developed model's loss performance of the test data against validation data

An evaluation of the loss of the model is also performed (Figure 5). It indicates the difference between the predicted labels and the actual labels. The analysis of the loss of the model is also represented. The model produces a minimal loss of 0.14 after a gradual decrease over the training period.

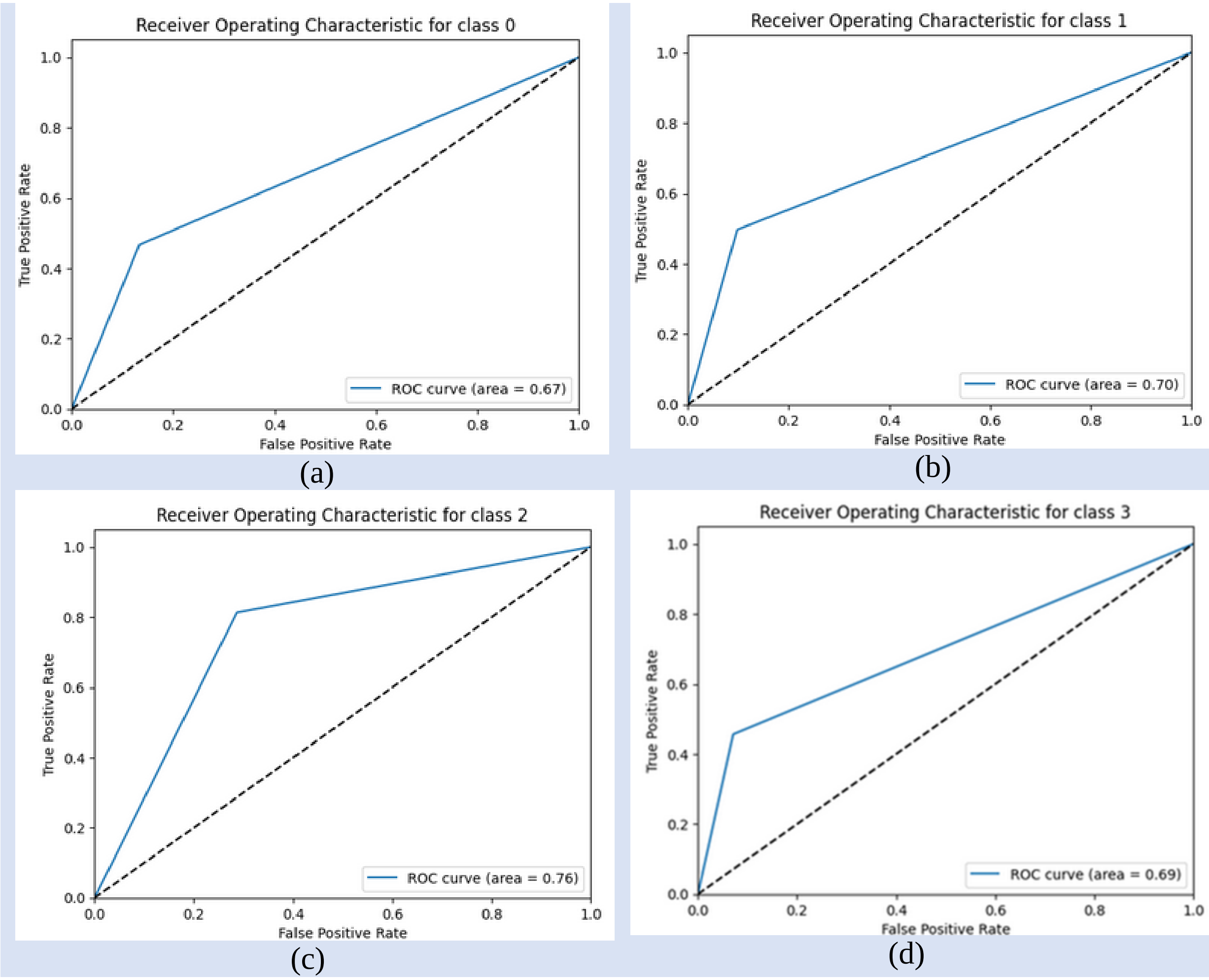

Figure 6. ROC curve of the CNN smoke/fire detection system with specified AUC values for its classification with regards to (a) Class 0 (b) Class 1 (c) Class 2 (d) Class 3

The ROC curve shows a graphical comparison of the TPR and FPR. The different AUC values of the classes defined in the code are displayed in the model's ROC curves. As a result, classes 0, 1, 2, and 3 are represented by the AUC values of 0.67, 0.70, 0.76, and 0.69 respectively. This suggests that there is a 67% chance that the model will correctly identify photos in class 0, 70% for class 1, 76% for class 2, and 69% for class 3.

Comparative Performance

The outcome of this investigation was contrasted with other deep learning-based models to determine the efficacy of the smoke and fire detection system. The selection criteria were predicated on the use of CNN-based models and the classification technique (smoke and fire) of the current approaches. Table 1 presents a performance comparison of the created approach versus the existing methods. In their work, Sheng et al. (2021) introduced an AlexNet-based model that achieved 88.42% accuracy by changing the activation function to a tanh function in the last layer. Mukhopadhyay et al (2019) model stated an accuracy of 73.27% with a false alert rate of less than 5%. In contrast, out of all the techniques taken into consideration, the constructed model had the greatest accuracy detection rate of 94.14%. This demonstrates why the strategy used in this study was beneficial.

Table 1: Comparison with other smoke and fire algorithms

S/N	Algorithms	Accuracy values
1.	Mukhopadhyay et al (2019)	73.27%
2.	Khan et al (2019)	84.85%
3.	Sheng et al (2021)	88.42%
4.	Ryu et al (2022)	93.00%
5.	Muhammad et al (2019)	90.06%
6.	**The developed model**	**94.14%**

Although there are a lot of false positives and false negatives when it comes to smoke occurrence, our model is able to identify flames in both indoor and outdoor environments with a high accuracy rate. A demo version of the program was tested on a PC to evaluate the detection system (see Figure 7). The playsound module of the OpenCV library was used to create a simulated alarm system configuration. The application was given access to the webcam and the "Alarm Sound.mp3" mp3 audio file was connected to the deep learning model. An alarm sound is produced and the user is made aware of the potential hazard as soon as smoke or a possible fire is identified through the acquired image frames.

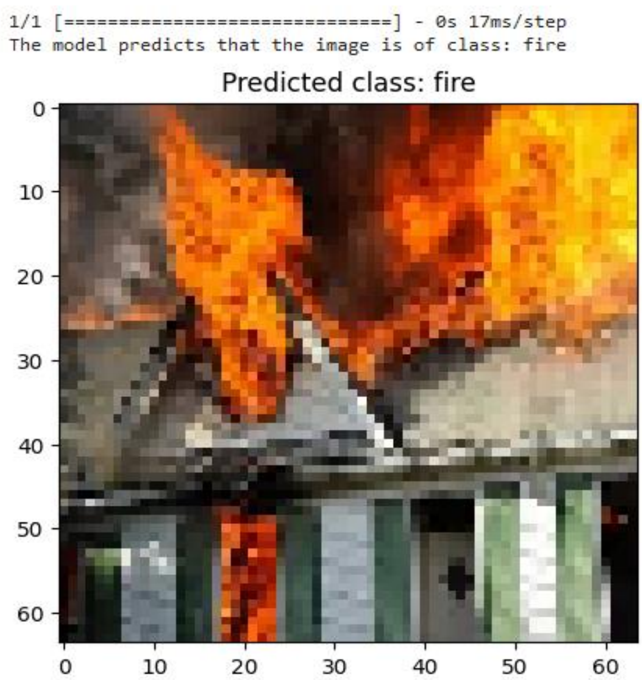

Figure 7. Prediction result of the CNN model on a test image of a fire occurrence

V: CONCLUSION

This study introduced a CNN-based smoke and fire detection system. The suggested solution seeks to address smoke and fire detection through automatic feature extraction: The acquired image is first pre-processed to bring attention to the important details. The Python programming language was utilized to implement the OpenCV library in order to accomplish this. Subsequently, the processed image is fed into the CNN model, this trains it to accurately identify images into an appropriate category of fire, smoke or neither occurrence. Keras, a deep learning framework, was used to ensure implementation simplicity.The performance result of the system indicated an accuracy of 94.14% on the training data and 95% on the validation data indicating the capability of the classification process. The system, which operates in tandem with surveillance cameras, is advised for use in high-risk buildings or areas to minimize the damages due to smoke and fire outbreaks. Further investigation on the detection system can be carried out on the application of recent object detection models like YOLOv8 (You Only Look Once v8) and the utilization of 3D-CNN models for improved generalizations.

REFERENCES

Almoussawi, Z. A., Khalid, R., Obaid, Z. S., Al Mashhadani, Z. I., Al-Majdi, K., Alsaddon, R. E., & Abed, H. M. (2022). Fire Detection and Verification using Convolutional Neural Networks, Masked Autoencoder and Transfer Learning. *Majlesi Journal of Electrical Engineering*, *16*(4), 159-166.

Avazov, K., Hyun, A. E., Sami S, A. A., Khaitov, A., Abdusalomov, A. B., & Cho, Y. I. (2023). Forest fire detection and notification method based on AI and IoT approaches. *Future Internet*, *15*(2), 61. https://doi.org/10.3390/fi15020061

Biswas, A., Ghosh, S. K., & Ghosh, A. (2023). Early fire detection and alert system using modified inception-v3 under deep learning framework. *Procedia Computer Science*, *218*, 2243–2252. https://doi.org/10.1016/j.procs.2023.01.200

Keita, Z. (2023). *Convolutional Neural Networks (CNN) with TensorFlow Tutorial: Learn how to construct and implement Convolutional Neural Networks (CNNs) in Python with Tensorflow Framework 2*. Datacamp. https://www.datacamp.com/tutorial/cnn-tensorflow-python

Khan, S., Muhammad, K., Mumtaz, S., Baik, S. W., & de Albuquerque, V. H. C. (2019). Energy-efficient deep CNN for smoke detection in foggy IoT environment. *IEEE Internet of Things Journal*, *6*(6), 9237-9245.

Lee, Y., & Shim, J. (2019). False positive decremented research for fire and smoke detection in surveillance camera using spatial and temporal features based on deep learning. *Electronics*, *8*(10), 1167. https://doi.org/10.3390/electronics8101167

Mandal, M. (2023). *Introduction to Convolutional Neural Networks (CNN)*. Analytics Vidya. https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/

Ryu, J., & Kwak, D. (2022). A study on a complex flame and smoke detection method using computer vision detection and convolutional neural network. *Fire*, *5*(4), 108. https://doi.org/10.3390/fire5040108192

Sankarasubramanian, P., & Ganesh, E. N. (2021). Artificial Intelligence-Based Detection System for Hazardous Liquid Metal Fire. In *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 1-6). IEEE, 2021.

Sathishkumar, V. E., Cho, J., Subramanian, M., & Naren, O. S. (2023). Forest fire and smoke detection using deep learning-based learning without forgetting. *Fire Ecology*, *19*(1). https://doi.org/10.1186/s42408-022-00165-0

Wahyono, Harjoko, A., Dharmawan, A., Adhinata, F., Kosala, G., & Jo, K.-H. (2022). Real-time forest fire detection framework based on artificial intelligence using color probability model and motion feature analysis. *Fire*, *5*(1), 23. https://doi.org/10.3390/fire5010023

Yavuz Selim, T., Koklu, M., & Altin, M. (2021). Fire Detection in Images Using Framework Based on Image Processing, Motion Detection and Convolutional Neural Network. *International Journal of Intelligent Systems and Applications in Engineering (IJISAE)*, *9*(4), 171-177.