2022 4th IEEE Middle East and North Africa COMMunications Conference (MENACOMM) | 978-1-6654-9411-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/MENACOMM57252.2022.9998196

A Machine Learning-Based Early Forest Fire Detection System Utilizing Vision and Sensors' Fusion Technologies

Ala' Khalifeh
German Jordanian University
Amman, Jordan
ala.khalifeh@gju.edu.jo

AbdelHamid Nassar
German Jordanian University
Amman, Jordan
a.nassar@gju.edu.jo

Mohammad M. AlAjlouni
Jordan Design and Development Bureau(JODDB)
Amman, Jordan
m.alajlouni@joddb.com

Anas AlNabelsi
German Jordanian University
Amman, Jordan
a.alnabelsi@gju.edu.jo

Zaid Alrawashdeh
German Jordanian University
Amman, Jordan
z.alrawashdeh@gju.edu.jo

Bashar Hejazi
German Jordanian University
Amman, Jordan
b.hejazi@gju.edu.jo

Radi Alwardat
Jordan Design and Development Bureau (JODDB)
Amman, Jordan
ralwardat@joddb.com

Jose Lima
Research Centre in Digitalization and Intelligent Robotics
Instituto Politécnico de Bragança
Bragança, Portugal
jllima@ipb.pt

***Abstract*—The paper aims at utilizing machine learning (ML) towards designing an early warning forest fire detection system. With the aid of the Internet of Things (IoT) and smart edge computing, an embedded system that utilizes sensors' fusion technology, machine vision and ML to early detect forest fire has been proposed. Different from most of the proposed fire detection systems in the literature, which either utilize vision or sensors'-based approaches to detect the fire, the proposed system utilizes both approaches jointly, which in turn will make it more accurate for fire detection. Furthermore, this paper focuses on implementing the proposed system utilizing a smart edge node and discusses the incurred technical challenges and how they have been solved.**

I. Introduction

Early detection of forest fires has become a hot topic recently due to their rate increase yearly. Climate change has eased forest fires incidence and spread; thus, researchers and companies are actively working on algorithms, solutions or systems that detect the fire events immediately. These systems will reduce hazards on human and animal lives, as well as reducing the economic consequences. Nowadays, most of the fire detection systems rely on sensor fusion of wireless sensor networks (WSN). While these algorithms depend on sensors reading levels and raise alarms accordingly, they usually fail to detect early stages of fire events; sensor readings do not change simultaneously or with sensible values until fire starts spreading. Other systems that utilize surveillance cameras either need a human observer or deploy computer vision algorithms to detect smoke or fire. These observations also cannot detect small fires before spreading. The evolution of artificial intelligence (AI) has enabled new capabilities of prediction and early detection in various applications. In this paper, a system that utilizes a fusion of two cutting-edge AI algorithms that processes simultaneously cameras live stream and WSN readings is proposed, where utilizing deep learning, a convolutional neural network (CNN) will be responsible for cameras live stream processing, while a standard artificial neural network (ANN) will analyse the sensors' readings for each sensor node. The algorithm then will fuse the decisions delivered from each network and take the final decision and therefore a corresponding action will be taken. To achieve the expected results, both networks must be designed precisely, deployed to an AI enabled embedded system processing unit (the Jetson Nano [1] in this paper), and a unique dataset shall be collected through out fire rising time to obtain early detection means. This paper presents our preliminary design, results, and implementation challenges for creating an early forest fire detection system, which created an ongoing research pathway that started from designing and implementing a generic WSN that addressed several challenges such as energy optimization, routing, coverage, and connectivity [2]–[4] and will continue until all faced challenges specifically for the current application under consideration, i.e., early forest fire detection are resolved and until the final system is finalized. The rest of the paper is structured as follows: Section II presents the literature review on forest fire detection system while section III addresses the technical background and system design where the algorithms, the design, the data collection and models are presented. Section IV details the experimental setup and performance evaluation with results and comparison between different approaches. Finally, section V concludes the paper and points out the future work direction.

II. Literature Review

The research community have been addressing fire detection since a long time ago, as it can be seen from several literature work, as an example, the authors in [5] presented in the 70's decade, a state-of-the-art work in fire detection. It is true that the earlier the fire is detected, and the actors alerted to fight the fire, the easier it is to win this battle. Any fire, no matter how large it may become, begins as a small fire. In its earliest stages, most fires are innocuous and are easily controlled. This justifies the research that has been put in this matter. According to [6], there are several techniques for forest fire detection, namely satellite-based sensing technique, sensor-based technique, camera-based technique, mobile biological sensing technique and unmanned aerial vehicle (UAV)/ Airborne technique. Several researchers focused on the WSN to acquire and transmit data (such as temperature, humidity, radiation and air quality among other variables) in order to detect fires [7], [8]. On the other hand, visual images from the scenes can be obtained and then identify fires. Since a long time ago, this procedure was made with human monitoring those images from watch towers, remotely or using satellite images. These procedures are not efficient due to the high infrastructure costs and the requirement of many trained personnel. On the other hand, sending ground crews to dangerous environments or using different classical techniques have many limitations in terms of cost and efficiency. Instead, UAVs equipped with visual remote sensing technologies were proposed as new and promising technologies that could help for wildfires monitoring and fighting. For this approach, machine learning methods can be used while leaving personnel available to other tasks. Moreover, images can be acquired using unmanned aerial vehicles that are of huge importance in obtaining images from difficult places. UAV-based wildfire detection systems integrate various remote sensing technologies and deep learning-based computer vision techniques, such as [9]. Combining UAVs and deep learning architectures could be very useful to detect fires at their early stages and send valuable information to the actors of firefighting and commands, helping in the decision process. In [9], authors study the use of optical and thermal infrared cameras to detect fire among the deep learning-based computer vision algorithms. In this work [9], several deep learning-based image classification are used and compared while stressing the fire detection with methods such as AlexNet, GoogleNet, ReasNet, Densenet and MobileNet for different approaches. Recently, in [10] a Haar Cascade classifier is proposed to detect fire using a webcam as a source of input for capturing the video feed from the surrounding environment and open CV library for image processing. Authors claim an accuracy of 95.2% for fire identification on images. In a recent study, Chowdary et al. [6] address a forest fire detection system using barrier coverage in wireless sensor networks that are used to circumvent the constraints of existing communication systems while also reducing implementation costs. Barrier coverage utilizes an interconnected network of "sensor nodes" and "base stations" that are equipped with a variety of sensors, including global position sensor (GPS), temperature, light, pressure, magnetic, smoke, acoustic, and camera, which communicate with one another and the base station on an ad-hoc basis. Unlike "area coverage," the barrier coverage network connects the base station to the sensor nodes using the shortest possible path. Chowdary et al. [6] proposed an approach to construct a system for monitoring and detection of forest fire using the barrier coverage networks by continuously sampling the real-time environment data. The data from each sensor is collected and transmitted to the neighboring nodes till the information reaches the base station. Our literature review shows that there is a significant interest from the research community to explore various ways to detect fire as early as possible and there are few papers that focus on the implementation aspects of such a system, which is among the topics that have been addressed in this paper.

III. Technical Background and System Design

This section provides a brief description about the essential background on machine learning needed to realize the proposed system. Furthermore, it describes the proposed forest fire prediction system.

A. Machine learning algorithms

Machine learning is a type of artificial intelligence that allows software applications to become more accurate at predicting outcomes. A neural network (NN) is an example of ML algorithms that utilizes graphs of neurons to model data. These neurons are structured like the human brain, that are designed to recognize and detect patterns in a dataset. It consists of interconnected units or nodes that can send a processed signal to neighboring neurons. [11] They interpret sensory data through a kind of machine perception, labelling raw input. The patterns which can be recognized are numerical, contained in vectors, into which all real-world data, such as images, sound, text, or time series, can be translated into those values. After processing the output of each neuron by using one of many non-linear functions, the network predicts to what category or class does the input belongs to.

Neural networks are constructed from numerous layers which are input layers, hidden layers, and output layers. Also, neural networks can be categorized into three main categories: artificial neural networks, convolutional neural networks, and recurrent neural networks. These neural networks regardless of their category, must be trained on a labelled dataset, which should be customized of what input will the model receive. As the network is training, the weights of neurons which forms the neural network will be updated to achieve a prediction of what category the input belongs to. After successful training the neural network becomes a trained model and can classify its inputs to what output they belong to. The artificial neural network, specifically deals with numerical data and in a feed forward fashion, meaning the data will move from the input layer through its neurons and move only forward until it reaches the final layer which is the output layer. The recurrent neural networks, is very similar to the ANN as well, as it deals with numerical data, but the recurrent, unlike the ANN that is a feed forward, has either a number of loops or one loop that is integrated inside the layers, meaning it is not just feeding forward, which adds to its complexity and needs time series logic to make use of the past values or acknowledgment of time. The convolutional neural networks are very complex neural networks that consists of a huge number of neurons, the neural network which specializes in processing data that has a grid-like topology, such as an image which consists of a series of pixels arranged in a grid-like fashion that contains pixel values, that denotes how bright and what colour each pixel should be. CNN can be used for image classification and object detection applications, where a huge number of labelled images are required to create a trained model. A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer, and while the network receives an image, processing begins as it goes through the layers. These layers are arranged in such a way so that they detect simpler patterns first, such as lines, curves, and edges, then it moves on to more complex patterns such as faces and objects. After the network has finished with feature extraction, it can classify to which class the image belongs to and will display this output either by a text if using image classification, or by framing the object detected in the image with a labelled box when using object detection. Training a CNN can be a tedious task, as it needs a lot of computational power and time, but instead of creating the model from scratch, a method called transfer learning is used, as it takes the knowledge from a previous model in a relatively similar task and gets transferred to a new custom model that has its own layers to specialize it to a specific task. Furthermore, the Jetson Nano is an edge computing device manufactured by NVIDIA company, which is specialized for artificial intelligence applications, and it is used to run small to medium sized artificial intelligence and Internet of Things (IoT) applications [12]. The Jetson Nano runs on Ubuntu-Linux system, which is an open-source software mainly used for edge and cloud computing applications. In this work, we utilize the Jetson Nano to analyze both the captured sensors' data and the captured images, process them and run the ML models to make the final prediction decision pertaining the forest fire early identification and detection.

B. Fire detection system design

The proposed fire detection system consists of two phases, the sensing, and the vision stages. Both stages have a pre-trained ML model that have been trained to detect fire either using the sensors or the images. This model is stored in the Jetson Nano edge computing node, which will make the prediction on each stage. The final decision of fire detection is generated as follows: first, the sensor nodes send the captured measurements to the Jetson Nano which runs the sensors' pre-trained ML model and give a preliminary decision on whether fire is detected or not, then if the first decision detects fire, the Jetson Nano will activate a camera, take a picture and use it as an input for the image-based pre-trained ML model, if the second model detects a fire, then the final decision will be announced as fire is detected, if at any stage, a no-fire decision is made, then fire will not be detected. In what follows, a more detailed description for each stage is provided. At the sensing stage, a group of sensor nodes equipped with a set of sensors designed to get the required measurements in the early fire detection process are utilized, where some basic measurements are recorded related to fire detection such as: the temperature, humidity, light intensity, carbon monoxide and (CO) concentration. After a careful search and considerations for the suitable sensors, the following sensors have been adopted:

1) The flame sensor: this sensor is mainly designed for detecting, as well as, responding to the occurrence of a fire or flame. This sensor detects flames with light emissions that have a wavelength is in the range of 760 nm – 1100 nm.
2) BME680: is an environmental sensor incorporating high-accuracy, high-linearity temperature, humidity, atmospheric/barometric pressure, and volatile organic compounds (VOCs) gas sensors.

It is noticed that the utilized sensors' readings were very accurate in capturing the values of the measurements related to fire, which are needed for the ANN fire-detection learning process. Having accurate measurements is essential to give meaningful predictions in the trained model that will run inside the Jetson Nano edge computing node. Particularly, the flame sensor has a high accuracy which is around 96%, and as for the BME680, the accuracy was 90%. Furthermore, since Deep learning process is utilized, a huge dataset is needed to increase the model's ability to learn and perform well. It was estimated that at least 20 thousand of rows of data are needed to have a good, trained models. To achieve that, a data collection process is performed where a real fire has been ignited and the sensors' measurements have been recorded on different time periods during the day, such that the captured data reflect the morning, noon, afternoon, and night periods of the day.

C. ANN and CNN utilized models

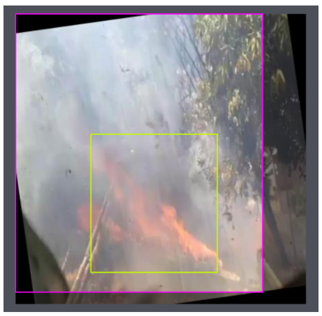
In what follows, a brief description of the utilized ML models used for fire detection is provided.

1) The ANN model: For the ANN model, it is designed to make the first stage fire detection utilizing the captured readings from the sensors, i.e., the temperature, humidity, gas concentration and light intensity. In fact, a recurrent neural network (RNN) model can be considered the best with its integrated loops that will connect the current readings with the previous readings. However due to its complexity, opting for a normal feed forward network with efficient alternative solutions was decided. Such as the previous result is added to the next reading to act as an input.

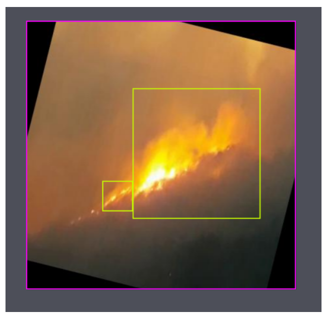
2) The CNN model: The CNN model was used for the second image-based fire detection stage. In this model, transfer learning was utilized. Transfer learning is widely used when the knowledge from a previous model in a relatively similar task is transferred to a new model and may be built upon. Large CNN models like the ones that will be discussed later, were trained on millions of images for thousands of hours. Thus, with transfer learning, it is possible to use the large pre-trained model and add to it the fire-images to make it more specialized to the fire-detection process. There are two types of CNN models that fit our application as listed below:

- Object detection models
- Image classification models

In the object detection model, the model is responsible for detecting the fire, dealing with it as an object, and then drawing a box around its location in the image.



(a)



(b)

Fig. 1. Sample of the Object Detection data set

The Fig.1 represents the dataset that is used for this type of models with a green box drawn around the fire, and the output. The main advantage of this approach is the ability to pinpoint the fire location. However, to achieve that, the dataset must be prepared while following a strict standard. For example, the following must be met:

- Labelling consistency
- Labelling accuracy

However, it can be seen from the differences between Fig.1a and Fig.1b that the labelling was inconsistent; the fire was enclosed in one box in one picture, and in multiple boxes in the other. Furthermore, Fig.1a has a small part of the fire extending beyond the box enclosing it. Thus, these differences and inaccuracies would cause the model not to be able to learn accurately. The solution is to redraw the boxes in the data set and use the marked new images for the training process. However, that is a very time-consuming task especially with the large number of images used for the training purpose. The alternative was to use the image classification approach, which is on the other hand, receives an image as an input and gives an output of plain text either ”Fire” or ”No Fire” without the need of boxes. However, the lost advantage of detecting the fire location can be reclaimed by the sensor nodes that send their measurements, where a GPS sensor can be added with the sensors to accurately determine their locations, thus realizing the detected fire location as well. In this paper, 11,829 images were used for the training process which are divided into 5,187 images of no fire and 6,642 images of fire.

TABLE I: ANN results

Validation Accuracy	Validation Loss	Learning rate	Batch size	Hidden neurons	Optimizer	Loss function
0.8607	0.2185	0.0001	100	6	Adam	Binary Crossentropy
0.9526	0.2508	0.001	100	6	Adam	Binary Crossentropy
0.9890	0.0606	0.001	32	6	Adam	Binary Crossentropy
0.9478	0.1719	0.001	32	3	Adam	Binary Crossentropy
0.8648	0.2169	0.001	32	4	Adam	Binary Crossentropy
0.9774	0.1361	0.001	32	6	RMSProp	Binary Crossentropy
0.7117	0.7899	0.001	32	6	Adam	Hinge

IV. Performance Evaluation

In what follows, the performance proposed fire-detection system utilizing the two stages model is discussed. In particular, the first subsection discusses the performance of the ANN model that has been used to predict the fire based on the sensors' reading, while the second subsection focuses on the CNN model that has been used to detect the fire based on analyzing the captured images.

A. The ANN model performance evaluation

The dataset utilized for this model was obtained experimentally as explained before, by making an artificial fire, the collected data comprised of 7,285 rows of data. Before initiating the training process, the parameters that need to be adjusted and tweaked were researched to find a small and optimal range to choose from and test. The parameters were as follows:

- Layer activation functions
- Loss functions
- Optimizers
- Number of layers and number of nodes within each layer
- Number of batch size and epochs

After a clear understanding of each parameter, many tests were conducted and the results after tweaking one parameter at a time were observed. As can be observed from the highlighted row in table I, it achieved the best results using that combination of parameters. This result proves the ANN model can learn and perform well in such a task if given a larger data set. However, the accuracy for the ANN model can be further improved if the dataset was enlarged. Moreover, different sensors may give better results as well, which is an ongoing research that we are still pursuing.

B. The CNN model performance evaluation

After choosing the image classification approach, a thorough search was conducted to find the popular models that also were used previously for fire applications. Consequently, this paper [13] was found very suitable for our problem. Thus, we imported a pre-trained **Resnet50** model that had been trained on ImageNet dataset previously. After that, three layers were added; one to flatten, one to do learning for the fire-detection task, and a final output layer to divide it into two classes ("Fire", "No Fire"). The metrics that were used to evaluate the proposed model's performance are: the validation accuracy and the validation loss. The training was performed on Google Colab which offered many advantages over training the model locally. Especially when dealing with large neural networks, and a huge dataset which was about 5.5 Gb. This would require days to train on our office personal computers. While it took merely a few hours on the Google Colab environment. In addition, Google Colab allowed a direct access to Google Drive, thus the dataset was uploaded just once to the Google drive and each time the training took place, the Google Colab environment directly called it from the drive. Thus, the whole process was done through the cloud without stress on our devices. Once the Resnet50 model completed its training, it achieved great results that will be shown later. However, when it was time to deploy the model to the Jetson Nano, it was discovered that it was too large for the Jetson Nano because of the Jetson's limited computational resources, thus. an error occurs stating there is not enough memory. Thus, it was time to search for a smaller yet still quite good model to use. Therefore, the Keras applications website is used to compare all the pre-trained models it offered, and the criteria for choosing the next model was based on the size and the accuracy of the model reached while training on the ImageNet dataset. Therefore, the next best option was **EfficientNetB4**. However, this model failed again to run consistently on the Jetson Nano when deployed and did not always work. Thus, it was unreliable. After this model, the last chosen model was the **EfficientNetV2B3**. This model managed to run smoothly on the Jetson Nano and consistently. Finally, Table II provides a comparison between the different tested models, which shows the difference in sizes between all the tested models. It can be observed that the sizes of 98 MB and 75 MB were too large for the Jetson Nano to handle. Note that the top-1 and top-5 represent the best accuracy the model has achieved while training on the ImageNet dataset after one and five batches, respectively. It can also be observed that the EfficientNet models offered better performance than the original Resnet50 while still having a smaller size. Thus, it was the best choice for the Jetson Nano's accurate and efficient operation.

TABLE II: General comparison between the CNN models

Model	Size (mb)	Top-1	Top-5	Parameters
Resnet50	98	74.9%	92.1%	25.6M
EfficientNetB4	75	82.9%	96.4%	19.5M
EfficientNetV2B3	59	82.0%	95.8%	14.5M

Furthermore, table III depicts the results obtained after training on the utilized dataset for the fire-detection application under consideration. The EfficientNetB4 did not have results to report because it was tested for compatibility with the Jetson Nano first before committing the time to train it. Moreover, the final EfficientNetV2B3 model as can be seen from the table managed to reach a very high accuracy of 99.56% while running smoothly on the Jetson Nano.

TABLE III: CNN results

The used pre-trained model	Validation Accuracy	Validation loss
Resnet50	0.9985	0.0082
EfficientNetB4	-	-
EfficientNetV2B3	0.9956	0.0135

V. Conclusion and Future Work

In this paper, a machine learning based system to detect early-fire is proposed. The proposed system consists of two stages, the first one utilizes a set of sensors to detect a fire, while the second stage utilizes images for the same purpose. To achieve that, two ML pre-trained models were used in the Jetson Nano edge computing device, an ANN model was used for the first stage, while a CNN model was used for the second stage. The Jetson Nano node will make the final decision about the fire detection process if it detects fire on both stages. One challenge faced on this work was to find out the best ML model to deploy on the Jetson Nano that has both high accuracy and low memory consumption, especially for the CNN model, which was found to be the EfficientNet model. As a future work, there are several directions for extending this project as explained below:

- Generating a larger dataset for the fire/no fire test under different weather conditions.
- Exploring more sensors that have more accurate results
- Connecting the sensor nodes to the cluster head using wireless connectivity as currently the nodes are directly connected to the cluster head (the Jetson Nano) using a wire.
- Optimizing the node's energy consumption and utilize solar energy to extend the battery life-time.

Acknowledgement

This work has been in part supported by King Abdullah II Fund for development. Grant No.2022/6 and Jordan Design and Development Bureau(JODDB).

References

[1] "Nvidia inc. site," https://developer.nvidia.com/embedded/jetson-nano-developer-kit, accessed: 2020-9-30.

[2] K. A. Darabkh, M. Al-Akhras *et al.*, "Improving routing protocol for low-power and lossy networks over iot enviroment," in *2021 30th Wireless and Optical Communications Conference (WOCC)*. IEEE, 2021, pp. 31–35.

[3] M. Nuimat, A. Gharaibeh, and A. Khalifeh, "Joint routing and channel assignment for throughput maximization in mobile wireless sensor networks," in *2020 International Conference on UK-China Emerging Technologies (UCET)*. IEEE, 2020, pp. 1–4.

[4] A. Khalifeh, N. Bartolini, S. Silvestri, G. Bongiovanni, A. Al-Assaf, R. Alwardat, and S. Alhaj-Ali, "Hybrid wireless sensor networks: A prototype," in *IFIP Conference on Human-Computer Interaction*. Springer, 2019, pp. 549–553.

[5] R. L. P. Custer and R. G. Bright, "Fire detection: The state-of-the-art," in *National Bureau of Standards*, 1974.

[6] V. Chowdary, D. Deogharia, S. Sowrabh, and S. Dubey, "Forest fire detection system using barrier coverage in wireless sensor networks," *Materials Today: Proceedings*, vol. 64, pp. 1322–1327, 2022, innovative Technologies in Mechanical Engineering-2021 (ITME2021). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214785322023410

[7] B. F. Azevedo, T. Brito, J. Lima, and A. I. Pereira, "Optimum sensors allocation for a forest fires monitoring system," *Forests*, vol. 12, no. 4, 2021. [Online]. Available: https://www.mdpi.com/1999-4907/12/4/453
[8] N. Varela, D.-M. Jorge L, A. Ospino, and N. A. Lizardo Zelaya, "Wireless sensor network for forest fire detection," *Procedia Computer Science*, vol. 175, pp. 435–440, 2020, the 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC),The 15th International Conference on Future Networks and Communications (FNC),The 10th International Conference on Sustainable Energy Information Technology. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920317427
[9] A. Bouguettaya, H. Zarzour, A. M. Taberkit, and A. Kechida, "A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms," *Signal Processing*, vol. 190, p. 108309, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168421003467
[10] K. S. Kumar, K. B. Varma, L. Sujihelen, S. Jancy, R. Aishwarya, and R. Yogitha, "Computer vision-based early fire detection using machine learning," in *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, 2022, pp. 1–5.
[11] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 53, Mar 2021. [Online]. Available: https://doi.org/10.1186/s40537-021-00444-8
[12] L. Barba-Guaman, J. Eugenio Naranjo, and A. Ortiz, "Deep learning framework for vehicle and pedestrian detection in rural roads on an embedded gpu," *Electronics*, vol. 9, no. 4, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/4/589
[13] S. Dogan, P. Datta Barua, H. Kutlu, M. Baygin, H. Fujita, T. Tuncer, and U. Acharya, "Automated accurate fire detection system using ensemble pretrained residual network," *Expert Systems with Applications*, vol. 203, p. 117407, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422007497