International Journal on Engineering Technology (InJET)

www.kec.edu.np/kec-journal/
Volume 2, issue no. 1, Dec 2024, Page 60-70
Received Date: July 16, 2024
Accepted Date: Sept 24, 2024
ISSN: 3021-940X (Print)
DOI: https://doi.org/10.3126/injet.v2i1.72519

Fire Detection and Extinguishing System

Krish Gurung[1], Sarbagya Ratna Maharjan[2], Shubham Pokhrel[3,*], Yaman Sigdel[4], Bharat Bhatta[5]

[1]*Department of Electronics and Computer Engineering, Sagarmatha Engineering College, Kathmandu, Nepal, thekrishgurung@gmail.com*
[2]*Department of Electronics and Computer Engineering, Sagarmatha Engineering College, Kathmandu, Nepal, sarbagyamaharjan1@gmail.com*
[3]*Department of Electronics and Computer Engineering, Sagarmatha Engineering College, Kathmandu, Nepal, shubhamhtd@gmail.com*
[4]*Department of Electronics and Computer Engineering, Sagarmatha Engineering College, Kathmandu, Nepal, yamansigdel999@gmail.com*
[5]*Department of Electronics and Computer Engineering, Sagarmatha Engineering College, Kathmandu, Nepal, bharat.bhatta@sagarmatha.edu.np*

Abstract

Advanced detection and suppression systems are imperative for early intervention in various environments where fire incidents pose significant risks. A comprehensive fire detection and extinguishing system that uses computer vision and image processing techniques to address important fire safety issues is presented in this project. Using an SSD-MobileNet-v2-FPNLite object detection model to analyze video feeds, the system uses a webcam to detect fires in real-time. The system acts quickly minimizes damage and possible loss by automatically initiating a suppression mechanism upon detecting fire. With automated detection, suppression, and improved situational awareness, integrating advanced technologies facilitates proactive fire management. To ensure the system's usefulness, effectiveness and feasibility analysis considers technical, operational, and financial factors. An incremental development model is used, with a focus on feedback integration and continuous improvement. This emphasizes continuous improvement through the integration of feedback. Metrics from the performance evaluation show how well the system detects fire incidents and launches suppression actions promptly, on time. This offers a significant advancement in automated fire management by demonstrating the potential of using cutting-edge image processing and machine learning techniques to address fire safety issues. This system is a significant advancement in fire safety technology that may shorten response times and lessen the effects of fires in a various context.

Keywords: Deep Learning, Object Detection, Image Processing, CNN, SSD, Mobilenetv2 FPNlite

1. Introduction

Fire incidents pose significant risks in various settings, including homes, offices, factories, and public spaces, often resulting in severe damage and loss of life. Traditional fire detection techniques, such manual firefighting and smoke alarms, have limitations in early detection and response. Smoke alarms typically detect fires only after a significant amount of smoke has accumulated, leading to delayed activation of fire response systems. Additionally, their effectiveness is highly dependent on smoke density, which can result in failures to detect fires promptly, particularly in cases of chemical or electrical fires that produce limited smoke.

Manual firefighting further exacerbates these challenges, relying heavily on the presence of humans to detect and respond to fires. This dependency can lead to potential delays, especially in unoccupied or sparsely occupied areas. The variability in human reaction times, ranging from a few seconds to several minutes, introduces inconsistencies in response, which can be critical in the case of a fire. To address these, this project aims to develop an advanced fire detection and extinguishing system that utilizes modern technologies to enhance early-stage fire response. By integrating sophisticated algorithms and automation, the system leverages computer vision, machine learning, and automated systems to improve accuracy and efficiency. Utilizing components such as Arduino UNO, web cameras, servo motors, and water pumps, the project employs image

*Corresponding Author

processing algorithms like SSD-MobileNetV2-FPN_Lite for real-time fire pattern identification and precise location detection. Upon fire detection, the system autonomously activates a targeted water suppression mechanism, ensuring a rapid and effective response.

Key features of this advanced system include early-stage fire pattern detection, precision nozzle alignment, and real-time environmental monitoring. The Fire Protector aims to significantly enhance fire safety across various environments, addressing the limitations of conventional systems. This project aims to reduce the risks associated with fire and enhance overall safety in a various context by offering a more prompt and precise response.

1.1. Related Works

The "Fire Extinguishing Robot" project addresses the frequent occurrence of sudden and unintentional fire disasters in various settings such as households, workplaces, and industries. The robots equipped with temperature and gas sensors were employed to detect and extinguish fires automatically. Additionally, the integration of an obstacle avoidance system using ultrasonic sensors with limited range was highlighted. The insights derived from their literature review have been instrumental in shaping the project's direction. However, it was unclear what kind of robots and micro-controllers had been used, which made it difficult for visualization and comprehend (Rashid et al., 2016).

The system can be operated in a various way with the use of GPS, GSM, DTMF, and remote controls. Furthermore, they emphasize that robot can be controlled via DTMF remote control and an Android smartphone, enabling operation in three different modes: autonomous operation, line following and manual control for fire detection. However, the methods by which the system achieves automation and algorithms based on fire situation classification remain unclear (Rashid et al., 2016).

The study also discusses the utilization of an IP Camera module to obtain live images of the field using a mobile camera via Wi-Fi. Any obstacle obstructing the path between the prototype and the affected area is detected by the ultrasonic sensor, prompting the robotic arm to relocate it to another location to clear the path. Nevertheless, details on how the arm handles various weights (loads) and navigates stairs and steps along its path remain ambiguous for understanding and analysis (Memon et al., 2018).

In urban cities and industrial areas, there is a constant need for firefighters' preparedness during emergencies, often resulting in manpower shortages. The researchers suggest that automated robots could assist in such situations. The use of Alex-net for fire detection and Image-net for identifying fire types is proposed, utilizing image processing and object recognition algorithms to enhance project development with substantial evidence. However, the challenge of the camera's inability to identify whether the fire has been extinguished in heavy smoke conditions was not addressed in the literature (Dhiman et al., 2022).

Despite significant technological advancements, Fire Extinguishing Robots are either manually operated or costly for local-level installations. Consequently, one of the major issues in Fire Extinguishing systems, as discussed in various research papers and articles, is the proper installation of the robot. There is confusion regarding where to install and surveil the robot. Therefore, recent projects aim to develop an automated fire extinguishing robot with a static position but dynamic surveillance from a higher vantage point. Additionally, a web camera will be utilized to address the challenge of heavy smoke detection. Thus, the journals reviewed share a common goal of developing an effective automated fire protection system to minimize damage (Saxena et al., 2020; Roboflow, 2024; Saugattimsina, 2024).

2. Methodology

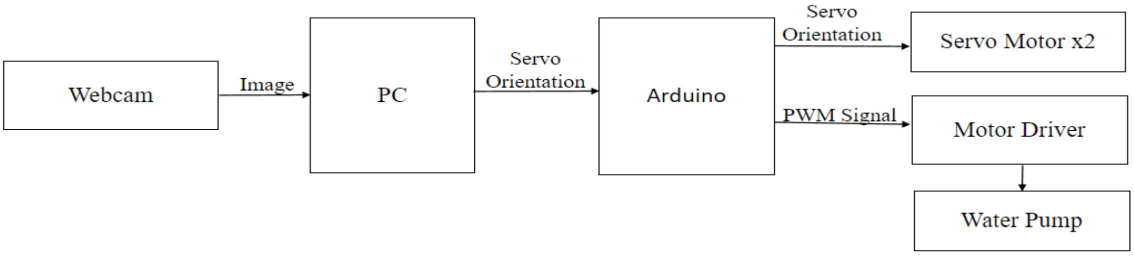

Figure 1. Block Diagram of the system

2.1. Dataset Description

The dataset for this fire detection project was compiled from multiple sources, including Roboflow and other relevant websites. It consisted of 9,410 fire images, each accompanied by a corresponding XML label file, totaling 18,820 data points. These images captured diverse fire scenarios in both indoor and outdoor settings, encompassing various lighting conditions and fire stages. Each XML file contained crucial annotation information for its corresponding image, including class labels such as 'fire,' image width and height, bounding box coordinates with diagonal points (xmin, ymin, xmax, ymax), and the confidence score of detection. The total images were split into 80%, 20%, and 20% for the train, test, and validation folders.

Table 1. Splitation of Total Dataset Images

Dataset			
Total Images	Train	Test	Validation
9410	7528	941	941

2.1.1 Attribute Description

Table 2. Image Attributes

Image Based Attributes	
Standard Deviation	0.01
Mean	0.0
Background pixel encode	0
Activation Function	RELU_6
Batch size	16
Learning Rate	0.08
Label, Class	Fire, 1
Image Size	320
Number of Steps	40000

IOU Threshold	0.6
Color Pattern	RGB

The image-based attributes were used to enhance model performance. Normalizing images ensures consistent data distribution, stabilizing and accelerating training. Background pixel encoding focuses the model on fire regions by minimizing irrelevant information. Activation functions improve the model's learning of complex patterns, while batch size and learning rate balance computational efficiency and performance. Image size retains sufficient detail for accurate fire detection, training steps ensure effective learning without overfitting, and an IOU threshold guarantees precise fire region identification. RGB color patterns leverage distinct fire color features.

Table 3. Annotation Attributes

Annotation Based Attributes	
Xmin, Xmax, Ymin, Ymax	Diagonal Co-ordinates of the bounding box
X1, Y1	Center Co-ordinates of the box
ImW, ImH	Width and Height of the image frame

Annotation-based attributes are crucial for accurate fire region localization. Bounding box coordinates allow the model to learn spatial extents of fire areas, ensuring precise detection. Center coordinates enhance detection accuracy by focusing on the fire's central area, while image dimensions ensure accurate mapping of annotations, aiding in precise fire localization within the image frame. These attributes are fundamental for effective fire detection and response.

2.2. Data Preprocessing

The labels in XML format were required so the dataset was created in XML files format. Also, the images taken from the camera were converted into 320×320 pixels for the input of the model. Auto-orientation of pixel data with EXIF-orientation stripping.

Resizing logic: Original image x scale factor
where, scale factor = 320/max(W, H)

W= Width of original image

H= Height of original image

2.3. Algorithm Description

In our fire detection system, we used a lightweight version of the SSD algorithm, called SSD-MobileNet-v2FPNlite, based on transfer learning. We trained the SSD model with a custom dataset using the TensorFlow 2 Object Detection API.

2.3.1. SSD-MobileNetV2-FPN-Lite

Particularly in resource-constrained environments such as mobile devices or edge devices. This architecture combines MobileNetV2 as the backbone with the Single Shot MultiBox Detector (SSD) for efficient and accurate fire detection. The "Lite" in the name typically signifies optimizations for resource efficiency. The FPN term is feature pyramid network, it addresses the problem of scale variance by constructing a feature pyramid with multiple scales. The term "320" typically refers to the input image size. It specifies the dimensions of the images that the neural network expects for processing.

The Single Shot MultiBox Detector (SSD) is a popular and influential object detection algorithm in the field of computer vision and deep learning. MobileNetV2 is a convolutional neural network architecture designed for optimal performance on mobile devices. FPN is a feature extraction architecture commonly used in detection models. It helps the model to capture fire objects of varying sizes and scales in an image.

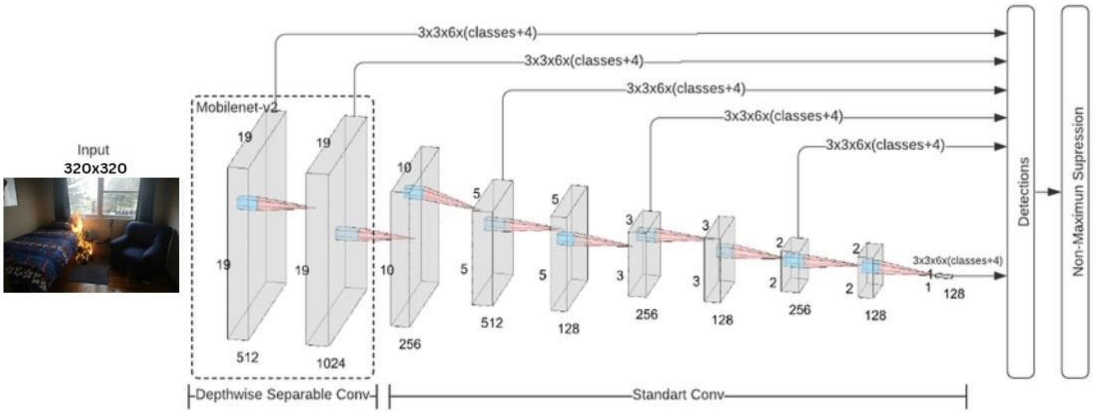

Figure 2. Model Architecture

There were several layers that were used to train our model for detecting fire which are explained below:

INPUT: This was the input layer, responsible for accepting fire images with a resolution of 320×320, or resizing them if they were not in that resolution.

MobileNet-v2: MobileNet-v2 is a lightweight CNN architecture used to extract features from input images, enabling the model to learn flame shapes, colors, and textures from fire-related patterns. It effectively reduces computational complexity while maintaining high accuracy.

Convolutional Layers: Several convolutional layers were used to extract and transform features from the input images. These layers include:

- 19×19×512: This convolutional layer, with a kernel size of 19×19 and 512 filters/kernels, was used to capture the spread and intensity of flames.
- 19×19×1024: This convolutional layer, with a kernel size of 19×19 and 1024 filters/kernels, was used to refine more intricate details from the previous layers.
- 3×3×6 x (classes+4): These depthwise separable convolution layers, with a kernel size of 3×3 and 6 filters, followed by (classes+4) output channels, were used for classification purposes, helping the model distinguish fire from non-fire objects, such as lighting effects or reflections.
- 1×1×256: This pointwise convolutional layer, with a kernel size of 1×1 and 256 filters/kernels, was used to adjust the number of channels and dimensions, optimizing computational efficiency without losing critical information.
- 1×1×512: Another pointwise convolutional layer with a kernel size of 1×1 and 512 filters/kernels was used to further refine the previous layers, discarding irrelevant information.

2.3.2. Standard Convo

A traditional convolutional layer with a kernel size of 3×3, used to provide additional processing that enhances the model's feature extraction capabilities, thereby improving its accuracy in detecting fire.

2.3.3. Detections

This layer represents the final output layer, where the network provides the bounding box coordinates and confidence levels of the detected fire regions in the image, enabling the effective identification and localization of fire.

2.3.5. Non-Maximum Suppression

It was a post-processing step to eliminate redundant bounding boxes with high overlap. It helped to refine the final output by selecting the most probable fire instances.

Overall, our model for detecting fire consisted of an input layer, a backbone network (MobileNet-v2), several convolutional layers for feature extraction and transformation and a detection layer for outputting fire instances. Non-Maximum Suppression was used for post-processing to refine the final output.

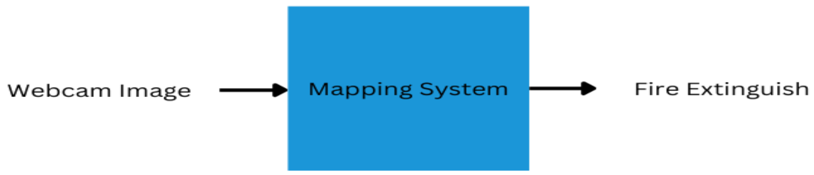

Figure 3. Input Output of the system

2.4. Inferencing

In this context, Google Colab was used for training and inferencing. The inferencing process was started when the frame was read from web-cam. The frame was acquired and resized, converted the BGR color pattern to RGB color pattern and provided the axis. After the actual detection, the output parameters like Bounding box coordinates, class present in the image and confidence score were extracted. The rectangular bounding box was drawn if the confidence score was greater than minimum threshold value. We get, Identifying the top-left corner pixels coordinates:

ymin = Convert to integer(Max among(1, box_y * image_height))
xmin = Convert to integer(Max among(1, box_x * image_width))
Identifying the bottom-right corner pixels coordinates:
ymax = Convert to integer(Min among(image_height, box_height * image_height))
xmax = Convert to integer(Min among(image_width, box_width * image_width))

Those were the diagonal coordinates of the bounding box. Also, the frame rate (fps) was drawn on one corner of the frame which is about 30.

2.5. Mapping into Real World Coordinates

A mapping function was made inside the inference code. This function takes the 2 coordinates of bounding box and calculates the midpoint of the box as:

X1=(Xmax-Xmin)/2
Y1=(Ymax-Ymin)/2

Here, (X1, Y1) was the midpoint of the bounding box. The resolution of web cam or the frame image from where the Midpoint of frame was identified as:

X2=ImW/2
Y2=ImH/2

Here, (X2, Y2) was the midpoint of the frame. Now, the difference was calculated in these points to know the distance by:

dx=x2-x1
dy=y2-y1

The initial position of servo motors was made random along X-axis and Y-axis. One Servo motor was for vertical displacement mapping and another for horizontal displacement mapping. The range of the servo motors was approximately 23 degrees to 100 degrees for x displacement and 50 degrees to 102 degrees for y displacement. After the dx and dy values were calculated, they were mapped within the limits of the servo motors using a mapping function from NumPy, which linearly maps the values to the specified target range. The mapping function is given as: mapped_value = Linear interpolation (value_to_be_mapped, (min_input, max_input), (min_target, max_target)) Here, the values to be mapped were dx and dy obtained above. The degree of rotation for servo motors are listed as below:

Table 4. Output of Mapping Function

X Coordinates	**Y Coordinates**	**Servo X degree**	**Servo Y degree**
329	202	80	64
534	978	67	102
543	768	67	102
345	768	79	102
357	435	78	81

2.6. Servo Control Mechanism (nozzle alignment)

The Servo.h library was used to connect and control the servo motors from the Arduino. First of all the python script running fire detection model sends angles for servo motors to Arduino using USB port through serial communication. The message passed was in the format of 'Xdata1Ydata2' (data1 indicating x alignment of servo and data2 for y). This format of message was decoded and data1 and data2 were separated by Arduino which was then used as inputs for servo motors.

2.7. Water Pump Activation

Once the nozzle was set in the direction of projection, the Arduino sent a command signal to the motor driver to turn on the water pump, which was powered by the motor driver. The water pump can outflow at the rate of 10L/min to the area of projection with a power of 8W. The spraying was done for 4 seconds which was more than enough for extinguishing fire. After the spraying was completed, the fire detection program resumed to detect the presence of fire.

3. Results

Analyzing the graphs, it's evident that the loss decreases consistently as the model progresses through training. With a final accuracy of 83.5%, the model improves with increasing checkpoints, enhancing its ability to predict the actual label (fire) in images and accurately localize objects with confident bounding boxes. Additionally, the model adapts to regularization penalties, learning more straightforward representations over time to strike a balance between fitting the training data and avoiding over fitting. The decreasing total loss indicates overall improvement in the model's capacity to classify objects accurately, localize them precisely and maintain simplicity. Moreover, the learning rate graph shows a decreasing trend as training progresses, facilitating slower and more careful convergence early on and faster convergence towards optimal solutions later, thus enhancing the model's performance in fire detection tasks.

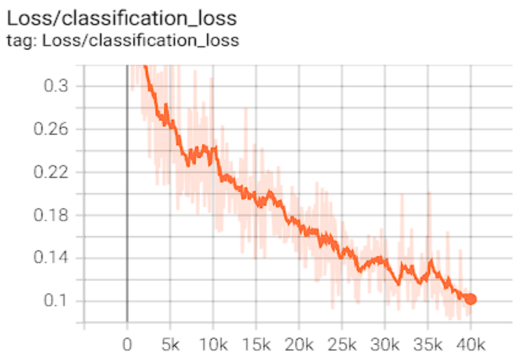

Figure 4. Classification Loss

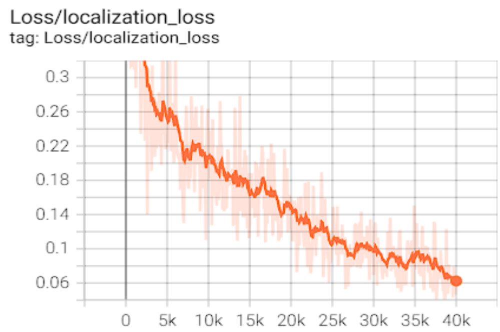

Figure 5. Localization Loss

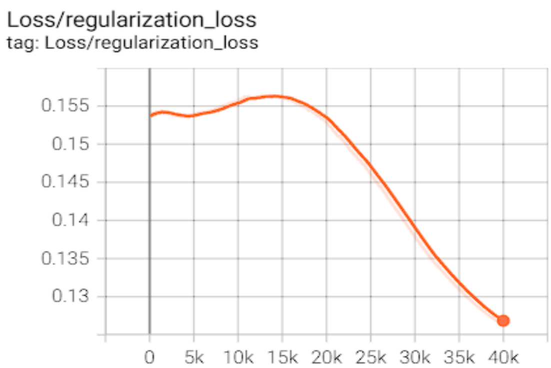

Figure 6. Regularization Loss

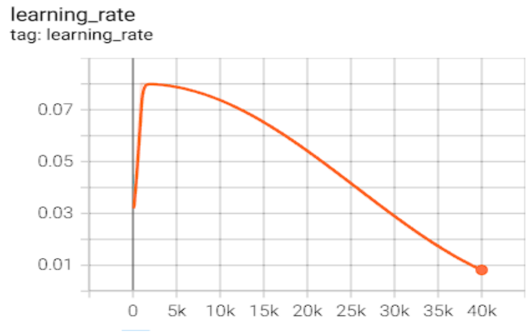

Figure 7: Learning Rate

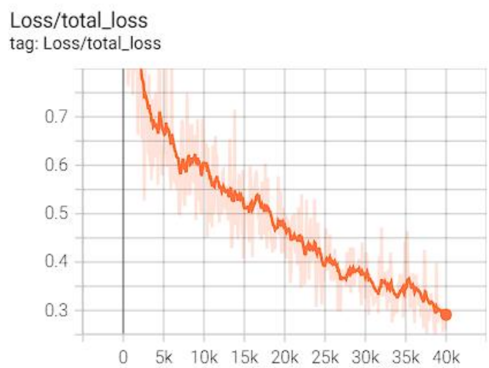

Figure 8. Learning Rate

3.1. Evaluation Metrics

The performance of our model was evaluated using key metrics, including Precision, Recall, F1-score, confusion matrix and mean Average Precision (mAP). Our model achieved an mAP score of 0.7590 at a 0.5 Intersection over Union (IoU) threshold for 'n' classes. Since our model has only one label, 'fire' (i.e., n=1), the mAP score is equivalent to the average precision value for the same parameters. Table 3 summarizes the performance metrics.

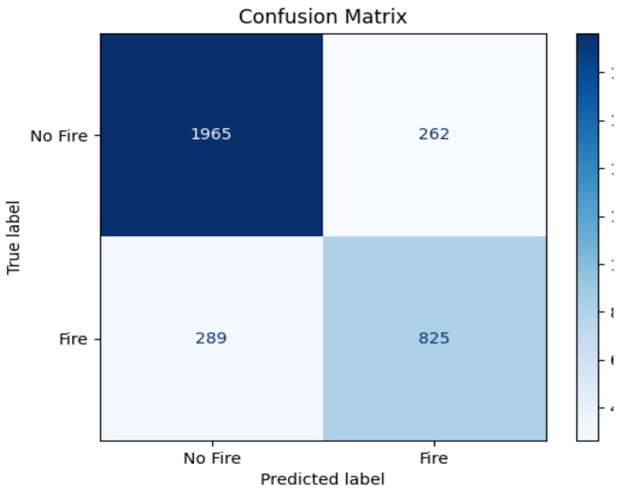

Figure 10. Actual Setup

For the confusion matrix, 2227 of non-fire images and 1114 of fire images were used. From the confusion matrix, it is clear that the model accurately determined as “No Fire,” 1965 cases, while it accurately determined 825 cases of “Fire” incident. However, it predicted 262 samples having “No Fire” reading as “Fire” while 289 samples with a “Fire” reading were identified as “No Fire”.

The following table shows the evaluation results of SSD-Mobilenet-v-2FPN-lite when set IoU threshold at 0.50. The model received an Accuracy of 83.51%, Precision of 75.90% and this indicates the proportion of true positive predictions (correctly detected objects) among all positive predictions made by the model. Recall of 74.06 % and this reflects the proportion of true positives out of all actual objects present in the dataset., and F1-Score of 74.97 % that reveal its efficiency in object detection designs.

Table 5. Metrics Table

		Evaluation	
Algorithm	**Metrics**	**Intersection over Union (IoU)**	**Score**
SSD-Mobilenet-v2FPNlite	Accuracy	0.50	0.8351
	Precision	0.50	0.7590
	Recall	0.50	0.7406
	F1-Score	0.50	0.7497

Figure 10. Actual Setup

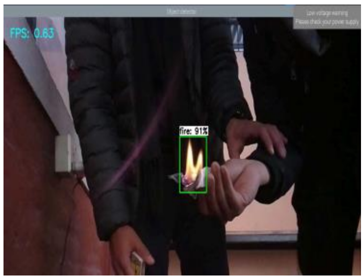

Figure 11. Visual image of a fire detection

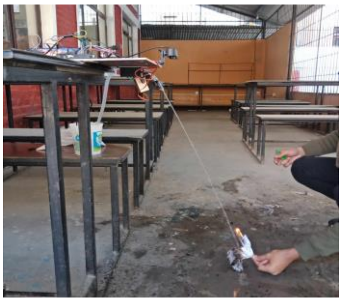

Figure 12. Practical operation of system

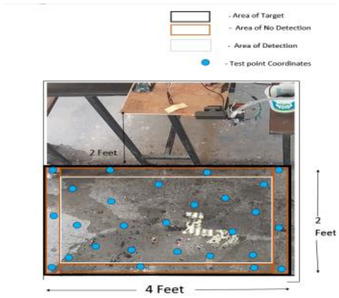

Figure 13. Coverage of System

System with multiple tests at different coordinates within the target area. The blue dots represent test points and a white flaming paper was used as a fire sample. The system detected fire effectively in the white zones but failed to detect fire in the orange zones, even though they were within the target area. The system is designed to detect fire only within the black target zone, with detection failures occurring in regions beyond 0.48 feet on the length and 0.24 feet on the breadth. The area of non-detection is calculated as follows:

Area of Non-Detection = Total Area – Detected Area

= ((4 * 2) - (3.04 * 1.52)) square feet = 3.3792 square feet

3.2. Comparison with Traditional System

1. The "Fire Extinguishing Robot" by Rashid et al. (2016), which relied on temperature and gas sensors, this model employs advanced computer vision (SSD-MobileNet-v2-FPN-lite) for fire pattern identification. This allows for more precise and real-time fire detection without depending on physical environmental changes like temperature or gas levels.

2. Image-based detection (using IP cameras) (Memon et al., 2018), this system leverages deep learning model capable of object detection, offering higher accuracy and adaptability, even in challenging scenarios like low smoke density.

4. Conclusions and Future Enhancements

The Fire Protector is clear example of merging the image processing with automated water suppression resulting in efficient fire protection solution. This makes its usage in industrial and public sectors as an example of its use in fire risk reduction and enhancement of the response to emergencies.

4.1. Limitations

The system today has limitations, lack of camera coverage resulting in blind areas, possible water damage due to system failure, and low recognition capability due to high density smoke or during very bright conditions. Limited effectiveness is due to dependence on a consistent power supply and high webcam resolution.

4.2. Future Enhancements

Subsequent versions should consider enlarging the area monitored by the cameras to avoid blind zones, and introduction of backup power sources to keep the cameras running at all times, and besides implementing a smart lock that prevents fires from spreading to other rooms installing inert gases to extinguish the fires. On board the platforms such as Raspberry Pi with installed Pi cameras and carrying out tests on the actual flame will enhance the reliability and accuracy.

Acknowledgments

This project has been supported by the Department of Electronics and Computer Engineering at Sagarmatha Engineering College (Lalitpur, Nepal). We extend our heartfelt gratitude to Er. Bharat Bhatta (project supervisor and head of department, Department of Electronics and Computer Engineering) and Er. Bipin Thapa Magar (project coordinator, Department of Electronics and Computer Engineering), for their suggestions and continuous guidance throughout the project. We also appreciate the invaluable support of Er. Baikuntha Kumar Acharya (deputy head of department, Department of Electronics and Computer Engineering).Also, we thank Sofa Hub Kathmandu Pvt. Ltd. for their cooperation and expertise. Special thanks to their staff for providing valuable insights and resources that significantly shaped our work. Additionally, we acknowledge the contribution of individuals who shared their experiences and insights on fire incidents, ensuring the relevance and effectiveness of our research.

References

Dhiman, A., Shah, N., Adhikari, P., Kumbhar, S., Dhanjal, I.S. and Mehendale, N., 2022. Firefighting robot with deep learning and machine vision. *Neural Computing and Applications*, pp. 1–9.

Memon, A.W., Du, J., Abro, A.H., Shah, S.I. and Bhutto, M.A., 2018. Design and implementation of fire extinguisher robot with robotic arm. In: *MATEC Web of Conferences*, vol. 160, p. 06008. EDP Sciences.

Rashid, H., Ahmed, I.U., Ullah, A., Newaz, M.F., Robin, M.S.R. and Reza, S.M.T., 2016. Multiple sensors based fire extinguisher robot based on dtmf, bluetooth and gsm technology with multiple mode of operation. In: *2016 International Workshop on Computational Intelligence (IWCI)*, pp. 41–46. IEEE.

Roboflow. Available at: https://universe.roboflow.com/search?q=fire [Accessed 28 June 2024].

Saugattimsina. Saugattimsina/nepali-numberplate-dataset.

Saxena, A., Singh, R., Khatun, S., Singh, S. and S iddiqui, K.M., 2020. A review on fire extinguishing robot. *Journal on Electronics Engineering*, 10(3).