Flame and Smoke Detection using Deep Learning

Submitted by:

Muhammad Ubaid

FA19-BCE-008

Saad Anjum

FA19-BCE-018

Ahmad Asim

FA19-BCE-010

Program: BS in Computer Engineering

Supervised by:

Dr. Wasiq Ali

Co-Supervised by:

Dr.Hafeez Anwar

A thesis submitted in partial fulfillment of the requirements for the degree of bachelors of Science in Computer Engineering

Department of Electrical and Computer Engineering

COMSATS University Islamabad, Attock Campus, Pakistan.

Abstract

This project proposes, using deep learning,to detect fires using a combination of closed-circuit television cameras, YOLOv5 object detection, a Raspberry Pi fitted with a Global System for Mobiles (GSM) module, and a cellular network. The cameras would be connected to the Raspberry Pi using a cellular network.To develop the algorithm, we collected numerous datasets containing images taken in various environments and then categorized the photographs showing smoke and fire. Following that, we have annotated particular areas of interest within the images associated with smoke and fire.The system has been constructed to monitor fires in instant time and deliver text messages warnings to the authenticated person. In addition, the system will determine the fire's location using the GPS coordinates obtained from the GSM module. The system then make use of these coordinates at some point. So the proposed system can significantly improved the efficiency of fire detection and response, which, in turn, has the potential to contribute to the preservation of both lives and property.

Keywords: Smoke Detection, Deep Learning Algorithm,CCTV camera.

Contents

List of Figures **v**

List of Tables **vi**

1 Introduction **1**

1.1 Introduction . . . 1
1.2 Background . . . 7
1.3 Motivation . . . 7
1.4 Problem Statement . . . 8
1.5 Research Questions . . . 8
1.6 Scope . . . 9
1.7 UN's Sustainable Goals . . . 9
1.8 Thesis Breakdown . . . 10

2 LiteratureReview **11**

2.1 Literature Review . . . 11
2.2 Conclusion Remarks . . . 20

3 Proposed Methodology **21**

3.1 System Architecture . . . 21
3.2 Data Collection and Pre-processing . . . 22
3.2.1 Data Collection . . . 22
3.3 Data Pre-processing . . . 22
3.4 Data Augmentation . . . 22
3.4.1 Splitting the Dataset . . . 22
3.5 Label smoothing . . . 22
3.6 Data Annotation . . . 23
3.7 Convolutional Neural Network . . . 23
3.7.1 Convolution Layer . . . 23
3.7.2 Hyperparameters . . . 24
3.7.3 Fully Connected Layer . . . 25
3.7.4 Activation Function . . . 26
3.7.5 Loss Function . . . 26
3.7.6 Backpropagation . . . 27
3.7.7 Architecture and Hyperparameters . . . 27
3.8 Working of YOLO . . . 27
3.9 YOLOv5s Implementation . . . 28
3.10 YOLOv5s Configuration Parameters . . . 29
3.10.1 Momentum . . . 29
3.10.2 Weight Decay . . . 30

3.10.3 Learning Rate 30
3.11 OpenCV Implementation 30
3.12 Hardware Implementation 31
3.13 Flow Charts 32
3.14 Pseudo Code 33
3.15 Mathematical Modeling 33
3.16 Component Selection 34
3.17 Hardware And Software Setup 34

4 Results And Simulations **36**
4.1 Simulation Results 36
4.1.1 Precision Recall Curve 36
4.1.2 Precision Confidence Curve 37
4.1.3 Recall Confidence Curve 37
4.1.4 F1 Confidence Curve 38
4.2 Experimental Requirements 39
4.2.1 Visual Based Result 40
4.3 Hardware Circuit 41
4.4 Hardware Based Result 42

5 Conclusions And Future Work **43**
5.1 Conclusions 43
5.2 Future Work 43
5.3 Limitations 44

Bibliography **45**

List of Figures

1.1 Targeted UN's Sustainable Goals 9

3.1 Block Diagram Of Proposed Method 21
3.2 Convolution Layer 24
3.3 Padding 24
3.4 Relu function 25
3.5 Pooling Layer 25
3.6 Architecture of YOLO 27
3.7 Flow Chart Of Proposed Method 32

4.1 Precision Recall Curve 36
4.2 Precision Confidence Curve 37
4.3 Recall Confidence Curve 38
4.4 F1 Confidence Curve 38
4.5 Dataset Image Collection 39
4.6 Software Based Validation Results 40
4.7 Software based Tested Result 40
4.8 Hardware Circuit 41
4.9 Hardware Based Simulation Result 42

List of Tables

2.1 Comparison Of All Approaches Made To detect Fire And Smoke. . . 12

Chapter 1

Introduction

1.1 Introduction

Fires are an unavoidable occurrence, posing risks to both the lives and property of individuals. However, the rate of fire spread at breakneck speed, resulting in fast and accurate fire detection Spot plays a key role in early warning and prevention of severe disasters [1].The traditional flame and smoke detection method of counting the density of particles and smoke dust has limitations such as low accuracy, false alarms, and difficulty in detecting fires in large areas.

While fire detection through digital video and image processing can overcome these limitations and provide a more effective solution. By using machine intelligence, the system can analyze the visual content of the image or video in real-time and detect the presence of fire based on specific features such as color, shape, and movement. This allows for quicker and more accurate fire detection, reducing the chance of false alarms and ensuring that fire incidents are detected in a timely manner [2]. To reduce losses of life and property due to fire breakouts, 24-hour surveillance of business and residential areas is an effective strategy. Yet, it may be challenging to identify fires early on or late on when numerous video feeds are being monitored at once. Investigating fire detection systems that make use of both sensors and surveillance cameras is crucial [3].

A sensor-based fire detection system may transmit warnings to a monitoring centre or emergency services when it detects certain fire characteristics, such as smoke, heat, or flames. On the other hand, current developments in machine learning and artificial intelligence technology allow camera-based fire detection systems to identify flames and smoke in real-time. These devices can quickly notify emergency services after analysing a fire's numerous visual features, such as colour, movement, and heat signature.

The ideal fire detection system may vary depending on the individual requirements of a certain structure or piece of property. To establish the most effective fire detection system for a specific setting, contact with fire safety experts and specialists [3]. Usually the number of fire incidents has been rising after a recent decline, causing property damage of 400 billion since 2013. To reduce the damage caused by fire, it is necessary to develop a fire monitoring system that can quickly detect and respond to fire incidents as it solution is to use computer vision algorithms and cameras to detect fire and alert authorities, which would also integrated with fire suppression systems to provide critical information about the fire [3].

While during the last few years millions of fires occurs in which thousands of fatalities and injuries, fires are a major concern on a worldwide scale. Almost 17,000 people lost their lives and 76,000 others were injured in over 6 million fires that broke out in the United States alone between 2015 and 2019. Around 3 million fires, over 19,000 fatalities, and 68,000 injuries were recorded in 34 countries in 2019 due to flames, according to data. It is crucial to be able to identify fires promptly and correctly so that early warning systems can be used to stop additional damage and loss of life given the potential for such severe property damage and loss of life. Because of this, there have been substantial attempts to construct fire detection systems, and several recent studies have looked into the use of deep learning algorithms to detect fires with high levels of accuracy [4].

However, Deep learning techniques have gained popularity recently in a variety of applications, including fire detection. Researchers have proposed convolutional neural network (CNN) models for video-based identification of fire and smoke and used deep learning models to extract valuable information from picture sequences in order to increase the accuracy of fire detection. Considering the benefits of CNN models, a lot of research has been done on utilising CNN models to detect flames in order to create more precise and effective fire detection systems [5]. These systems enable the early and precise identification of fire occurrences as well as the prompt notification of necessary authorities and people, which may help minimise the damages brought on by flames [6]. Thus Convolutional neural networks (CNNs) are effective for flame detection and other image and video recognition applications. CNNs are very good at tasks like object identification and detection because they can learn and extract pertinent information from photos and videos in a hierarchical manner. CNN models may be used to recognise and categorise the presence of fire in fresh photos and videos after they have been trained [7]. Thus it can extract the important information from images in herrical manner that makes it very effective even in recognition of complicated object identification [8].

So implementing a comprehensive fire monitoring system which can help to reduce the damage caused by fire incidents and improve the safety of communities so usually using deep learning algorithms for fire and smoke detection is a promising approach to increase the speed and accuracy of fire monitoring systems [3].In recent years, the development of advanced machine learning algorithms has trained computer vision models to detect and classify various types of fire and smoke in real time, increasing the speed and reliability of fire detection systems [3]. While Deep learning that is a branch of machine learning that utilizes artificial neural networks. It is an effective technique for identifying patterns and characteristics in complicated data, including pictures and movies. Applications for deep learning include voice recognition, object identification and recognition, natural language processing, and many more. It mostly used to distinguish the visual characteristics of fire and flames in the context of flame and smoke detection smoke however smoke has a very essential role to occur the fire anywhere which could happen any incident in the form fire disaster. However, fire can be controlled by recognizing the place to alert at the time as well as smoke which is collection of airborne solid and liquid particles and gases that are released when something is burned while fire is a rapid chemical reaction between oxygen and a fuel source that release heat, light and gases. So fire can cause property damage and some injuries that are even be deadly in different situation however its very important to take some prevention to reduce such losse and disaster quickly if occurs [9].

Moreover Flame and smoke detection have been handled using deep learning techniques, which is an important area of research. The goal of this method is to develop an intelligent system that can detect a fire and alert people to its presence in a timely and reliable manner. Deep learning is a subset of machine learning that teaches neural networks to comprehend complex patterns and traits that might be utilized to solve prediction issues.

The system is trained on a big dataset of images and videos that include various types of flames and smoke when using deep learning to detect flames and smoke. The neural network learns to recognizes the visual patterns associated with each category after categorizing the videos and photos as flame, smoke, or non-fire.

There are numerous ways to create a deep learning-based system for flame and smoke detection. One approach is to use convolutional neural networks (CNNs), a type of neural network that can identify patterns and attributes in images [5]. In order to extract relevant features, this approach first processes the input data (i.e., the picture or video frames) using a CNN. The output data is then processed using a number of layers of convolution and pooling operations.Before deciding definitively whether there is fire, one or more fully linked layers process the CNN's output.

Overall, using deep learning to detect flame and smoke is a promising area of research that may improve fire safety and save lives. However, the creation of a reliable and accurate system necessitates a sizable amount of training data and rigorous neural network architecture design. A variety of real-world scenarios must be used to validate the system in order to ensure its applicability and durability.

Deep learning algorithms are being used to automate the detection of fires in a range of scenarios, according to a sector that is now in the midst of rapid development called flame and smoke detection using deep learning. The fundamental advantage of using deep learning for flame and smoke detection is that it enables the creation of accurate and reliable automated systems that can operate in real-time without human supervision. Applying deep learning to detect flames and smoke involves several steps. A dataset of images and videos of actual fires is first annotated to indicate whether flames and smoke are present or absent. These images and videos are then used to train a convolutional neural network or another deep learning system (CNN). During training, the deep learning system learns the characteristics that are most important for identifying photos with and without flames and smoke. Among other things, this might have features like colour, texture, and motion.

Once trained, the deep learning technology may be used to detect fires and smoke automatically in new pictures and videos. This can be accomplished by putting more images or video frames through the trained algorithm, which will produce a forecast indicating the propensity for flames and smoke to be present. When the network is trained, it may be used to quickly analyst new video or photo data to detect smoke and fires. The system may also incorporate additional sensors or data sources, such as temperature or motion sensors, to improve the detection's accuracy.

Mostly wide-ranging applications for deep learning-based flame and smoke detection include spotting fires under dangerous conditions in industrial settings, alerting occupants of potential fires in smart homes, and improving fire safety in public spaces. Deep learning-based flame and smoke detection systems have the potential to save lives and reduce property damage by quickly identifying and warning people to flames In Computer vision and machine learning are used to automatically detect the presence of flames and smoke in video or picture data in the new field of flame and smoke detection using deep learning [10]. This technology has a wide range of potential applications, including fire alarms, early warning systems, and fire-fighting robots.

Visual data must first be educated to recognizes patterns that correspond to flame and smoke in order to detect them using deep learning. This is frequently performed by training the network using a big dataset of annotated images and videos of flames and smoke in order to teach the network how to distinguish between these patterns and other kinds of visual input.

To create the training dataset, images from various sources, such as security cameras or publicly available datasets, can be acquired. Preprocessing is typically done on the images to remove any noise or undesirable artefacts that could impede learning. After the training data has been prepared and divided into training and validation sets, the model is trained using a number of deep learning techniques, such as stochastic gradient descent and Adam optimization.

The model updates its biases and weights during the training phase to decrease the error between the predicted and actual labels in the training data. After being taught, the model can be used to detect flames or smoke in brand-new frames of footage or pictures.

Deep learning in flame and smoke detection may identify fires early in homes, businesses, and public buildings. The system may automatically spot fires, reducing casualties and property damage. It has ability to identify fires and smoke in pictures often uses a neural network model. The network must be trained with a large number of labelled photos. These images may be from online archives or genuine flames and smoke. Convolutional layers identify and extract picture information. One or more fully connected layers recognize flames or smoke from the convolutional layer output. During the training phase, the network develops the capacity to recognize a wide range of patterns and properties related to fires and smoke. Color, texture, and shape are only a few examples of these patterns and characteristics. Once trained, the network can properly recognizes fresh photos as smoke or fire so, after that Deep learning fire detection systems must be tested using real-world data. It enables the system correctly and reliably detect fires in various scenarios and surroundings as compare the sensors that are not suited for some detection as there was some limitation and most costly. There are certain restrictions with conventional fire detection systems that depend on temperature and smoke sensors. These sensors can only detect a limited region hence they have a narrow coverage area. In order to monitor a vast region, several sensors must be used, which may be expensive and time-consuming to install and operate. Moreover, standard fire detection systems do not provide comprehensive details about the detected flames, such as their size and location. Firefighters may not have a thorough grasp of the size and location of a fire as a result, making it difficult for them to react swiftly and efficiently. Nevertheless, more advanced fire detection technologies are being created, such as infrared cameras and video-based fire detection, which can get around some of these drawbacks. These technologies may offer real-time information about the fire, including its position and size, which can help firefighters react more quickly and efficiently [11].By utilizing a deep learning version of YOLO for fire detection could be a promising approach to conside. In comparison to conventional techniques, deep learning-based systems may provide findings for fire detection that are more accurate and dependable. Convolutional neural networks (CNNs), that are one kind of deep learning model, are capable of learning from a massive quantity of data and recognizing intricate patterns in pictures and movies. Because of this, they may be better able to detect flames, particularly in difficult situations or when the fire is tiny or burning slowly. A deep learning version of YOLO for fire detection, however, needs a big and varied dataset of fire photos and videos for training the model, much like any machine learning-based system. The depth learning model's accuracy and efficiency may be directly impacted by the quality and variety of the training data. however, in order to achieve thorough and accurate fire

detection, it is crucial to integrate deep learning-based fire detection with other techniques and technologies, such as temperature and smoke sensors. It is important to note that the development of such a system requires a significant investment in time and resources, as a large annotated dataset of fire images is required, and the model must be fine-tuned for the specific task of fire detection. However, the investment can pay off in the form of increased safety and efficiency in fire detection [2].
To protect the safety of its residents, imagine a large commercial building that needs complete fire and smoke detection. Conventional smoke and temperature sensors-based fire detection systems are already in existence, but they have drawbacks, including a small detection window and the inability to provide precise position and size details.The facilities management of the building choose to employ YOLOv5 deep learning in a video-based fire and smoke detection system to overcome these restrictions [12]. This system consists of a network of high-resolution cameras spread out across the structure that record live video feeds. The YOLOv5 deep learning system is trained on an expansive and varied collection of fire and smoke photos and videos, enabling it to precisely identify flames and smoke.
To provides the thorough details about the location and size of the fire. These sensors depend on environmental physical changes to set off an alert, including temperature variations or the presence of smoke. Due to this, temperature and smoke sensors may not be able to quickly extinguish a fire if it starts too early or if it is to far and away from sensors. Rescue services may take longer to respond as a result, raising the risk of property damage and injuries [13].
The Fire and Smoke Detection project using deep learning aims to develop a system that can automatically detect flames and smoke in real-time using image and video processing techniques [12]. The system is trained using a convolutional neural network (CNN) architecture to analyze input images and identify patterns associated with flames and smoke. The proposed system is expected to achieve high accuracy, speed, and robustness in detecting and alerting about the presence of flames and smoke in a variety of scenarios, such as industrial settings, households, and public spaces. This project has practical applications in fire prevention, safety, and emergency response, and can potentially save lives and resources.
The suggested system typically needs a number of processes, such as data gathering, preprocessing, training, and deployment. The initial stage in creating a prediction model is collecting a large dataset of images and videos depicting fire and smoke in a variety of settings (including private residences, commercial establishments, and public areas). Pre-processing involves labelling, cleaning, and dividing the dataset into training, validation, and test sets so that it is ready for training. The deep learning model is then trained, and this is often accomplished using a convolutional neural network (CNN) architecture.
The Network is trained using the labelled dataset to recognize patterns and traits associated with fire and smoke. The model is optimized during training using a variety of techniques, such as gradient descent and backpropagation, to lower the loss function and improve the model's performance.
Once the CNN has been trained, the system for real-time smoke and flame detection can be put into action. During deployment, the system uses CNN to analyze input pictures and videos in order to detect the presence of flames and smoke. Using GSM and Raspberry Pi, this new way to detect fire and smoke in real time uses computer vision, deep learning, and embedded systems. Users can then be notified via SMS or other ways of communication [14].
The system consists of a camera that is mounted to a Raspberry Pi single-board

computer, which is inexpensive. The Raspberry Pi receives the camera's images and analyses them using a deep learning algorithm that has been trained to discriminate between fire and smoke. The Raspberry Pi will alert a GSM (Global System for Mobile Communications) modem, which will subsequently send a text message to the user in the event of a fire or smoke detection. In addition, the system can be configured to make a phone call to a user instead of sending an SMS or to issue warnings to many users simultaneously.
This system can be used in a wide range of scenarios, such as those requiring the instantaneous detection and notification of fire and smoke in residences, workplaces, industries, and other sites. Because of its low cost, ease of setup, and high accuracy, this system can be very useful for managing disasters and fire safety.
Machine learning algorithms are becoming more and more popular for use in fire detection and monitoring because they can automate the detection process and enable 24/7 monitoring at a bigger scale than human personnel. One such machine learning detector, YOLOv5, has enhanced speed and accuracy over its predecessors, making it a strong option for real-time object recognition applications like fire and smoke detection. The YOLOv5 algorithm, the fifth version of the YOLO algorithm, has been shown to be faster and more accurate than previous versions [1] [11].It is a good choice for object identification applications that require high accuracy as well as rapid processing because it has an upgraded anchor mechanism and a more effective backbone network. Due to the reduced size of its file, it is now appropriate for use on low-resource devices, which is beneficial for some applications and efficient instrument for real-time object recognition in the fire monitoring efforts so if we examine the effectiveness of YOLOv5 and other earlier versions of neural network models for detecting fires, as well as the reliability of these models in the actual world, by comparing different neural network models of each, we may learn more about how effectively these models perform. The potential benefits of machine learning detectors for monitoring and avoiding fires make this an important area for research.
Fire can have devastating consequences on people and property, therefore it's necessary to keep them from becoming out of control. The fact that large flames can frequently develop from minor fires that aren't monitored highlights how important it is to monitor burns in order to prevent fires from starting. In order to monitor and prevent fires, a complete strategy must be in place because it may not be possible to monitor every burn [15]. The purpose of this thesis is to examine the application of machine learning techniques like YOLOv5 for detecting fires and to assess their potential for helping to monitor and prevent fires in high density areas.
In recent years, there has been a growing trend in the field of smoke detection and segmentation as well as computer vision techniques towards the application of image processing and deep learning methodologies. The accuracy and precision of conventional image-processing-based smoke detection methods are limited [11] [1]. Deep learning models, such as YOLOv5, have demonstrated considerable gains in their ability to identify and localize smoke as a result of the availability of more powerful computer resources and larger datasets. Deep learning models that focus on specific regions of interest can produce more precise and trustworthy smoke segmentation results, which can be valuable for early fire prevention and pollution management. Several conventional and deep learning strategies for commercial and industrial and fire smoke segmentation are developed. in addition, in this thesis with the purpose of determining the most efficient method for detecting and localizing smoke in real scenarios. The use of computer vision techniques in a variety of domains, such as highway safety, drug detection, and public health and safety, has been developed by

(a) Fire

(b) Smoke

recent advancements in artificial intelligence and deep learning. Smoke alarms and physical inspections that are two common but ineffective and expensive traditional approaches for detecting smoking behavior. In the context of industrial and fire smoke segmentation, the purpose of this thesis is to examine the efficacy of artificial intelligence and deep learning techniques for the early detection and prevention of smoke-related problems.

1.2 Background

Traditional flame and smoke detection methods often rely on rule-based algorithms or simple image processing techniques. These methods have limitations in terms of accuracy, robustness, and adaptability to different scenarios. Deep learning, a subfield of AI, has emerged as a powerful tool for computer vision tasks, including flame and smoke detection. Deep learning models can automatically learn and extract relevant features from images, leading to more accurate and reliable detection results.

1.3 Motivation

The ultimate goal of this initiative is to bring about a system that is as efficient in terms of both speed and cost as this project is to protect people's valuable possessions in a variety of forms, such as residential and commercial buildings, educational institutions, health-care facilities, places of worship, vehicles, forests, and electrical appliances that are prone to short circuits, such as freezers, washing machines, ovens, and other pricey items. This project's overall objective is to safeguard people's valuable possessions.

For instance, hospitals rely heavily on expensive machinery that is easily damaged by fire, and the textile industry also uses heavy machinery that is susceptible to fire, which results in significant financial losses for the industry. Other examples of industries that rely heavily on machinery that can be easily damaged by fire include: As a result, the purpose of this project is to build a system that is able to identify and prevent the occurrence of potential fire dangers in hospital rooms, thereby protecting the machinery and resources that are kept in those rooms.

In addition, it has a shorter response time, high accuracy, early detection, minimum maintenance, scalability, flexibility, better safety, and environmental friendliness. In comparison, other fire detection systems are more expensive.

1.4 Problem Statement

- The challenge of detecting fires and smoke under various environmental conditions, including low visibility, occlusions, and variations in light and color.
- The need for a robust and reliable system that can minimize false alarms and ensure timely notification to relevant authorities or personnel.
- The use of deep learning algorithms and image processing techniques to preprocess and analyze image or video data for flame and smoke detection.
- The need to optimize the design and architecture of deep learning models to achieve high accuracy and scalability in large scale environments.
- The ethical considerations related to deploying automated flame and smoke detection systems in public spaces, including privacy, bias, and accountability.
- The potential benefits of such a system for public safety, including early detection and response to fires, minimizing property damage, and potentially saving lives.
- Images that are misclassified by classification systems often have comparable attributes. Images with sunsets, fog, mist, or other visual components, for instance, might occasionally be mistakenly identified as the target image.

1.5 Research Questions

- Can deep learning algorithms accurately detect and classify different types of flames (e.g., gas, liquid, solid) and smoke under various environmental conditions?
- How does the accuracy of flame and smoke detection using deep learning compare to traditional fire detection methods e.g., thermal imaging, smoke alarms) in different settings?
- What is the most effective way to preprocess image or video data for flame and smoke detection using deep learning, and how does this affect model performance?
- How can flame and smoke detection systems using deep learning be optimized for use in large-scale environments (e.g., warehouses, factories, forests)?
- What are the ethical considerations related to deploying automated flame and smoke detection systems in public spaces, and how can these be addressed?
- Installation and location of camera?
- How many images should be acquired to achieve an accuracy greater than 90 percent?
- How to extract different color of smoke in an image?
- How to train our CNN model?
- Performance comparison of different CNN models?

1.6 Scope

Flame Detection: Deep learning-based flame detection algorithms can analyze image or video data to identify the presence of flames. These algorithms can detect flames in various environments, including indoor and outdoor settings, and can work with different types of flame sources such as candles, stoves, wildfires, or industrial flames.

Smoke Detection: Deep learning models can be trained to detect smoke patterns in images or video streams. Smoke detection algorithms can identify and track smoke plumes, helping in early detection of fire incidents and enabling timely response.

Fire Detection Systems: Deep learning-based flame and smoke detection systems can be integrated into fire alarm systems for enhanced fire safety. These systems can automatically monitor the surroundings and trigger alerts or activate fire suppression mechanisms when flames or smoke are detected.

Video Surveillance and Monitoring: Deep learning models for flame and smoke detection can be applied in video surveillance systems to monitor critical areas and detect fire incidents in instant-time. This technology can be valuable for applications such as security, industrial safety, and public safety.

IoT Integration: Flame and smoke detection using deep learning can be integrated with Internet of Things (IoT) platforms to enable remote monitoring and control. This allows for real-time alerts, notifications, and even automatic emergency response mechanisms.

1.7 UN's Sustainable Goals

This project can be considered to meet the following aims shown in figure 4.9, which are part of the 17 Sustainable Development Goals to end poverty, safeguard the environment, and ensure that everyone lives in peace and prosperity by 2030.

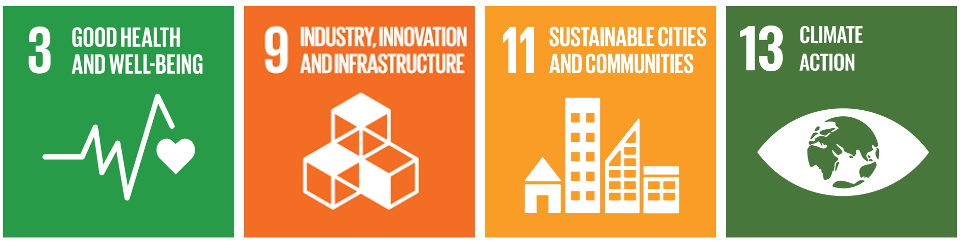

Figure 1.1: Targeted UN's Sustainable Goals

Goal 3: The purpose of this goal in our project to Maintaining Good Health and Well-Being - Encourage safety and minimizing the rate of death or are injured in accidents or natural disasters, such as fires.
Goal 9: The purpose of this goal in our project to Develop and apply technologies, such as deep learning models for fire detection, to increase safety in industry and

infrastructure.
Goal 11: The purpose of this goal in our project to Sustainable Cities and Communities, Improve urban design and management to make cities safe and resilient, including fire prevention and response.
Goal 13: The purpose of this goal in our project to Improve resilience and adaptive ability to hazards and natural disasters, particularly fires that may become more frequent and severe as a result of climate change.

1.8 Thesis Breakdown

In Chapter 1, we look at the notion of flame and smoke detection using deep learning, as well as its rationale and research issues.
In Chapter 2 we have Explore more details, and evaluate literature reviews. possible research techniques that have been used in the past to optimize power
In Chapter 3, we demonstrate the block diagrams, pseudo code, flowchart, mathematical models, and the selecting of project components, hardware and software configuration, as well as all project constraints.
In Chapter 4, we have concluded our all project software and hardware result.
In Chapter 5, we have written our conclusion, future work and limitations of our project.

Chapter 2

LiteratureReview

2.1 Literature Review

In the previous chapter, we discussed the significance of flame and smoke detection in saving lives, and how deep learning techniques can be used to achieve accurate and real-time detection. We also highlighted the challenges and crucial parameters, such as motivation and UN sustainable goals, that are integral to this research. In this literature study, we aim to analyze the various approaches taken to optimize flame and smoke detection using deep learning techniques. We will begin with an overview of the current state of the field, and then compare and contrast the different methods used to achieve the best possible results. Our goal is to identify the most effective approaches that can be used to optimize flame and smoke detection systems for public safety, environmental monitoring, and industrial management.
In recent years, the application of deep learning to the detection of fire and smoke has gained significant attention. Convolutional neural networks (CNNs) are a popular deep learning technique that has demonstrated remarkable performance in recognizing fire and smoke in both static images and dynamic video streams. This literature review provides an overview of current research in this field, which highlights the use of deep learning-based methods such as CNN-based fire detection models with multiple spatial scales, CNN-LSTM fire detection models, and two smoke detection systems, all of which are capable of accurately identifying fire and smoke in diverse scenarios. One of the significant advantages of deep learning-based systems is their ability to detect fire and smoke in real-time, making them highly valuable for public safety, environmental monitoring, and industrial management. However, it is essential to develop detection systems that are reliable, resilient, and effective in various environmental conditions. In conclusion, the findings of this literature review underscore the crucial role of deep learning technology in developing advanced fire and smoke detection systems. With continued research and development, these systems can help mitigate the devastating impact of fires and smoke on our communities and the environment [16].

Table 2.1: Comparison Of All Approaches Made To detect Fire And Smoke.

No #	Ref No #	Paper Title	Approach	Technology	Results
1	[17]	Real-time detection of flame and smoke using an improved YOLOv4 network	Real-time detection of flame and smoke	YOLOv4	Light-YOLOv4 can achieve a better balance between performance 85.64 percent mAP and efficiency 71 FPS, which meets flame and smoke detection tasks requirements on the accuracy and real time.
2	[18]	A Smoke Detection Model Based on Improved YOLOv5	Smoke Detection	YOLOv5	identify about 90 percent of the affected area for smoke and flames in different environments
3	[15]	Automatic Early Detection of Wildfire Smoke with Visible Light Cameras Using Deep Learning and Visual Explanation	smoke plume detection	ResNet, EfficientNet,and GradCAM	An Area Under Receiver Operating Characteristic curve (AUROC) of 0.949 obtained with an EfficientNet-B0.
4	[19]	UAV Forest Fire Detection based on RepVGG-YOLOv5	UAV Forest fire detection	RepVGG and YOLOv5	Improve the accuracy and efficiency of forest fire detection and response

Continued On Next Page

Table 2.1 – Continued From Previous Page

No #	Ref No #	Paper Title	Approach	Technology	Results
5	[16]	Wildfire detection in aerial images using deep learning	detection of fire locations monitored by UAV drones	YOLOv5	On the FireNet and FLAME aerial picture datasets, we evaluated the proposed method's performance and achieved the F1-score of 94.44 percent
6	[20]	Deep Convolutional Neural Network for Fire Detection	Fire detection	AlexNet, SqueezeNet and Fire Detection	comparison between CNN models in term of efficiency,execution time and size
7	[21]	An Improvement of the Fire Detection and Classification Method Using YOLOv3 for Surveillance Systems	real-time high-speed fire detection using deep learning	YOLOv3	capable of detecting fires that are 1m long and 0.3m wide at a distance of 50m
8	[2]	Flame Detection Using Deep Learning	Flame Detection	YOLO	the obtained accuracy of our proposed flame detection is up to 76 percent
9	[4]	Real-Time Video Fire Detection via Modified YOLOv5 Network Model	Real-Time Video Fire Detection	Modified YOLOv5	The proposed method can well suppress false detection and missed detection in complex lighting environments and improve the robustness and reliability of fire detection, meet the performance requirements of the video fire detection task

Continued On Next Page

Table 2.1 – Continued From Previous Page

No #	Ref No #	Paper Title	Approach	Technology	Results
10	[3]	Video Based Smoke and Flame Detection Using Convolutional Neural Network	Video Based Smoke and Flame Detection	AlexNet, GoogLeNet and VGG-16	Experimental results showed that all three network models were classifying fire detection at over ninety percent accuracy
11	[22]	Multi-Scale Video Flame Detection for Early Fire Warning Based on Deep Learning	Multi-Scale Video Flame Detection	YOlOv3	Proposed method not only improves the performance of the original algorithm but are also advantageous in comparison with other state-of-the-art object detection networks, and its false positives rate reaches 1.2 percent in the test set.
12	[23]	A Real-Time Fire Detection Method from Video with Multifeature Fusion	Flame Multifeatures Fusion	Video-based, Motion and Color Detection, Feature Extraction, Support Vector Machines	detection rates above 98 percent and false alarm rates below 2 percent but in outdoor fires in forests or industrial sites, fires may not produce visible flames in their early stages, making it difficult to detect them using this approach

Continued On Next Page

Table 2.1 – Continued From Previous Page

No #	Ref No #	Paper Title	Approach	Technology	Results
13	[24]	An Evaluation and Embedded Hardware Implementation of YOLO for Real-Time Wildfire	Evaluate performance of YOLOv5, YOLOv3, and YOLOv3-tiny for wildfire detection, Comparison of different versions of YOLO on Raspberry Pi 4, Detection of wildfires using machine learningrespectively	YOlOv3,YOLOv3-tiny,YOLOv5	YOLOv5 showed high performance and excellent battery life

According to the approach of this paper [17][25], there are many ways to find flames and Smoke. Some methods have been around for a while and use machine learning algorithms, while others are newer and use deep learning-based object detection techniques. In traditional methods, features like color, texture, and shape are collected, and then SVM or BP neural networks are used to train models. People often use these methods. More recent methods like BoWFire and KNN background subtraction have made these strategies more effective because they have made fine-tuning easier and reduced the number of parameters. Deep learning-based techniques for figuring out what an object is are fairly new. For example, a two-stream convolutional neural network looks at an image's spatial and temporal parts to correctly recognize a flame. The improved Faster-RCNN method for identifying flames sets the anchor limit and makes image-wide data to help in the detection process. Color properties are used to do both of these things. A window-based processing strategy is used by the deep convolutional neural network that is used to find wildfires using cameras. It increases the number of fires discovered.Last but not least, the AddNet video-based wildfire detector creates dense and convoluted feed-forwarding passes by using the MF operator to make an operation that looks like a dot product.
In this paper [18], The related literature based on feature extraction and classifier construction, we look at the research on smoke characteristics used to build traditional smoke detection systems. These systems were the basis for finding Smoke. This technique has focused most of its attention on color, texture, and movement as ways to recognize Smoke. Still, deep learning algorithms have greatly affected how well smoke detectors work. Deep convolutional neural networks, which include both two-dimensional and three-dimensional convolutional neural networks, are used by researchers to get information about time and space from Smoke. Traditional convolutional neural networks like AlexNet [?], VGG-16 [26], VGG-19 [27], and ResNet [28] have also been used to compare different collections of smoke pictures in experiments. These comparisons have been made to find out which collections of pictures

of Smoke are the most accurate. Even though smoke detection using deep learning has shown promising results, it is still a hard problem to solve. It is because there isn't a large collection of samples, and smoke scenes change and are hard to predict. Researchers have developed new deep learning algorithms and methods, such as a 3D, parallel, fully convolutional neural network, a deep saliency network, a reconstructed convolutional neural network, and a two-stage smoke detection system, to get around the problems already found. Some examples include a deep saliency network [29], a reconstructed convolutional neural network [30]. It has been shown that using these strategies would make a big difference in how well and how accurately smoke detectors work. Unfortunately, developing new algorithms and methods for deep learning is not enough to solve the problems in smoke detection by itself. Because deep learning is part of the field of machine learning, Also, the industry needs more photos of Smoke that can be used for training and testing. Because of this, the researchers have been putting a lot of effort into making a huge sample database that shows all the different kinds of smoke scenes. As a result, it will be able to make deep learning models for smoke detection that are more accurate and reliable. As a result, deep learning algorithms have greatly improved the accuracy and effectiveness of smoke detectors. Even so, there are still problems in the business, such as the lack of a complete sample library and the complicated structure of smoke scenes. Those are just two examples. In preparation for future research, researchers are emphasizing the need to build a large and varied collection of smoke images while also focusing on making new methods and strategies for deep learning.

In this paper [15], This review of the related literature that the test set that Govil and his colleagues used was not enough. Hohberg's research, on the other hand, did not use a test set at any point. Even though Frizzi et al. and Yin et al. showed good detection and low rates of false positives, the photographs they used differed greatly from those used in this experiment. Renjie and his colleagues used EfficientNet, while other researchers used residual nets for their analyses. Since Wang et al. used GradCAM to find smoke plumes, the literature study stresses the importance of using relevant test sets to evaluate deep learning models for smoke plume detection and shows the importance of using relevant test sets when evaluating deep learning models for detecting smoke plumes. In addition, it shows how important the size of the dataset is and how important the photos used for training and testing are. Both of these are important for getting the correct results. This research suggests ResNet, EfficientNet and GradCAM as ways to find smoke plumes [?]. The results of this review of the relevant literature will be used to build a deep-learning model that can quickly and accurately find smoke plumes in landscapes.

In this paper [19], The review of the related literature Unmanned Aerial Vehicles (UAVs) uses a combination of computer vision and deep learning algorithms to find forest fires. The YOLOv5 object recognition model has recently gotten much attention because it is accurate and can find things quickly. To reach this degree of effectiveness, cutting-edge technologies like the Focus layer, C3 layer, SPP layer, and PANet are used to improve the accuracy of feature extraction and detection. Even though the C3 layer helps the model learn new features better, the Focus layer helps with downsampling so that important information doesn't get lost. By keeping local features the same at different scales, the SPP layer helps improve both resilience and the addition of information at multiple scales. The PANet neck combines high-level and low-level characteristics for more feature expression. Researchers have released a new version of the YOLOv5 model called RepVGG-YOLOv5, which they hope will make it even more effective. This model has a RepVGG block, which uses a multi-

branch structure during training and turns it into a VGG-style basic structure during inference. During training, the model is trained with the help of the structure. This reparameterization makes the model smaller without losing any information and reduces the number of model parameters and the time it takes to make an inference. The most important parts of the reparameterization process are the fusion of the convolutional layer, the batch normalization layer, and the convolutional branch layers. Before the reparameterization process starts, both of these fusions take place. In conclusion, the YOLOv5 model and its modified version, the RepVGG-YOLOv5 [19], are cutting-edge technologies that have proven useful for object recognition in UAV forest fire detection situations. These situations involve using an uncrewed aerial vehicle (UAV) to watch a fire in the woods. These models can find things quickly and accurately, which is important for quickly responding to emergencies like forest fires. In this paper [16], CNNs have quickly become one of the most well-known and widely used deep learning methods for classifying pictures. They can be used to find fires, among other things. CNNs are also called convolutional neural networks. In this review of the relevant research literature, we look at how CNNs have been used to find fires and highlight some of the state-of-the-art models that have been tweaked and improved to achieve high accuracy in fire detection tasks. AlexNet [31], VGG [26], Inception, and ResNet [32] are examples of these models. These models used a wide range of datasets and methods, which led to detection rates that ranged from 86 percent to 99.56 percent. But because these models have so many parameters, it may be hard to run them on smaller, faster machines because they have so many parameters. New multi-channel CNN-based techniques [33] and smaller neural networks, like Fire Net, have been developed so small devices can detect fires with high accuracy. It has been done to try to find a way to fix the problem that has been identified. In conclusion, CNNs are a useful tool for finding fires, and more research on them could improve the accuracy and effectiveness of systems that find fires.
In this paper [20], The researcher discusses how convolutional neural networks, which are used to find fires. CNNs can be used in many fields, such as medicine, emergency response, and security, so finding fires is a popular area of research. This body of research also gives a summary of the many CNN-based methods that have been created to find and classify fires. It also points out the need for models that are both efficient and lightweight so that they can work on mobile devices with limited battery power. Researchers have made a lightweight convolutional neural network (CNN) architecture by studying many convolutional neural networks. This architecture can be used in embedded application environments (CNNs). The recommended convolutional neural network (CNN) used ideas from AlexNet [31] and SqueezeNet [34], and it did a great job of figuring out whether something was fire, Smoke, or neutral. According to the results, CNN architectures with fewer kernels and fewer parameters may be able to match the accuracy of state-of-the-art models while also making the models much smaller. So, the results of this work could have a big effect on the development of CNN-based fire detection systems, especially when there aren't enough computers to go around. The lightweight CNN architecture can be used on small devices like the Raspberry Pi or Jetson Nano. It makes it possible to find and classify fires quickly and accurately in real time. The design can also be used on larger machines, like a supercomputer. Also, the design is flexible enough for gadgets with limited memory. The research described in this work could be used as a starting point for future work on developing effective, reliable, and lightweight CNN-based systems for detecting and classifying fires. It is a possibility because the research could be used as a starting point for further work. It could happen because

the study could be a foundation for future work.
In this paper [21], The review of the related literature are about Traditional computer vision and artificial intelligence systems that use machine learning and deep learning are the two main areas that researchers have focused on throughout the last years to improve fire detection systems. It is because both traditional computer vision systems and systems based on artificial intelligence use machine learning and deep learning. Even though computer vision-based methods are used a lot, they have some problems when recognizing flames that are complicated and always changing. Researchers have looked at deep learning from many different angles as they try to find answers to the problems that have been found. One of these ways is to use convolutional neural networks (CNNs) with dilated convolutions, which helps to increase generalization while reducing the number of false alarms. Researchers have also added spatial and channel-wise attention to CNN and looked into flame recognition algorithms based on how Smoke moves to improve feature representations for scene classification. It was done to improve how scene classification uses feature representations. The goal of this move was to help CNN be more successful. Also, researchers have developed useful methods that can be used to find fires in cities. One of these methods is the static ELASTIC-YOLOv3 model, which has shown good results when finding night-flight fires. Based on the results of these tests, it seems that AI-powered systems can make fire detection more accurate and reliable. If these capabilities were used, it's possible that they could help make public spaces and buildings safer and more secure.
In this paper [2], The review of the related literature are about Identifying flames that is becoming an increasingly vital component of intelligent monitoring. When it comes to detecting fire and flames, we need to be able to train and test using extracted visual characteristics from video frames. A collection of shallow learning models has been developed based on these models based on color, fuzzy logic, motion, and form, amongst others, that have been created specifically to detect fires. Deep learning is an innovative technology that has the potential to be much more effective and precise in flame detection. In this research, we use the YOLO model to perform flame detection and compare it with other shallow learning approaches to identify the most effective flame detection. In this study, we contribute to using the improved version of the YOLO model for flame identification from video frames. We gathered the data and trained the models on it using the TensorFlow platform from Google.Our suggested method for detecting flames has an accuracy of up to 76 percent.
In this paper [4], This review of the related literature aims to make a method that uses deep learning and computer vision to find flames and Smoke. Both of these methods used to figure out what to do. The author suggested improved version of the YOLOv5 algorithm to use bounding boxes to tell the difference between fire and Smoke. Because of this, the problems caused by the model's inability to recognize very small objects and its slow convergence have been solved. Under this method, the dilated convolution technique was added to the SPP module as an operational part [35]. In addition, the activation function GELU was employed instead of SiLU when its counterpart was used. Ultimately, DIoU-NMS was chosen as the best option to replace NMS as the expected bounding box suppression. Since this is the case, it is better for finding very small flame targets and helps the model converge faster. By changing the parameters of YOVOv5, you can get a higher level of refinement. The results of the experiments of modified YOLOv5 works better than the original model.The updated YOLOv5s can still recognize 125 frames per second, the same as before. Also, the technology can recognize flames even when small, reducing the chance of false alarms while making fire detection more accurate. The proposed so-

lution would likely be able to meet all of the performance requirements for the role of video fire detection.
In this paper [3], The author devises a way to find fires that use AlexNet, GoogLenet, and VGG-16 and test them. They taught the network to recognize" fire" by using 768 different images of fire (both" smoke" and" flame") as training data, which shows that AlexNet has more than 94.00 percent accuracy, GoogleNet has more than 95.00 percent accuracy, and VGG-16 has more than 98.00 percent accuracy. Images that have been misidentified almost always have parts that are also in images that have been classified, like a sunset, fog, mist, or other things that are similar to these things. Future studies will need to pay close attention to gathering many different photos of fires and images that are similar to each other. It will make it possible to have a more accurate detection rate. Also, it will be essential to finish tests and improve different detection algorithms to reduce the number of false positives. The project aims to improve fire prevention by using surveillance cameras.
In this paper [22], This review of the related literature has current investigation uses an updated Yolov3 model to find tiny flames in early fires. Due to multi-scale convolution, an increasing receptive field, and an FPN structure, a model has been made that tackles both the problem of omission and the problem of false positives head-on. Based on the results of the studies, the suggested model is better than the Yolov3 model, which had been used before, and other models are often used to find objects. Real-world examples are used to test the model's accuracy, and the results show that iterative training is an important part of testing how well the model works in the real world.
In this paper [23], The review of the related literature are about Video-based flame multi-feature fusion combines motion and color detection to detect potential fire regions, extracts the flame's visual features, and uses support vector machine algorithms for increased accuracy and fewer false alarms. Infrared, acoustic, and deep neural networks are other techniques. Deep neural networks can automatically learn complex patterns from photographs or movies, while infrared and acoustic sensors recognize flames by their heat and sound. Based on their characteristics, computer vision techniques can be used to track the movement of flames and Smoke, and mathematical models can be employed to predict the development of fires. According to the literature, video-based systems show promise, but additional research is needed to develop approaches that can handle difficult outdoor environments and detect fires in their earliest stages. Approach Flame Multi-features Fusion Technology: * Video-based, Motion and Color Detection, Feature Extraction, SVMs Detection rates of over 98 percent and false alarm rates of less than 2 percent, but in outdoor fires in forests or industrial sites, fires may not produce visible flames in their early stages, making detection difficult. Accuracy achieves high accuracy rates of around 98 percent on some datasets, but as with any machine learning-based system, the accuracy can be affected by factors such as lighting conditions, camera angles, and the type of fire being detected.
In this paper [24], The review of the related literature are about Wildfires that are a constant threat, and detecting them quickly and accurately is crucial. In this study, the researchers explore the potential of using machine learning to automatically detect wildfires in real-time on embedded systems. They compare the performance of three versions of the You Only Look Once (YOLO) object detection modelYOLOv5, YOLOv3, and YOLOv3-tiny, on the Raspberry Pi 4, and the results are impressive. Yolov5, in particular, demonstrated high detection accuracy and exceptional battery life, making it a promising option for real-world applications. The researchers suggest

that further reliability tests are needed, but the potential of using machine learning to efficiently and accurately detect wildfires is clear..

2.2 Conclusion Remarks

We can see that Fire and Smoke Detection was the main emphasis and discussed objective throughout all of the comparisons. Maximum optimization and cost reduction are our major goals. Other parameters that were studied were:

- Security
- Response time and efficiency.
- Optimization of cost.
- Improved simulation.
- Light-weight model.
- Detection of Faults.

Our Contribution

In recent years, fire incidents have become a major cause of concern worldwide, resulting in significant loss of life and property damage. One of the biggest challenges in combating fires is detecting their location and severity, particularly in large buildings or complex structures. While various methods have been proposed to detect smoke and fire, including traditional sensors and alarms, they have limitations regarding accuracy and reliability. In this project, we contribute to addressing this challenge by proposing a new approach that leverages the power of GSM technology to improve smoke detection and identify the location of fires more precisely. By combining GSM technology with advanced machine learning techniques, we have significantly improved the speed and accuracy of fire detection and response, reducing the potential risks and consequences of fires. Our research could have significant implications for improving fire safety and emergency response systems in various settings, from commercial and industrial buildings to residential homes and public spaces.

Chapter 3

Proposed Methodology

Our proposed system's primary objective is to safeguard human lives and properties by detecting fire incidents and promptly alerting a controller through GSM, enabling identifying the fire location using GPS coordinates. The system comprises Raspberry Pi 4, GSM900, and a USB Camera. Raspberry Pi 4, a compact single-board computer, facilitates real-time flame and smoke detection by processing sensor data or video feed using a deep learning model. It serves as an integrated computing platform adaptable to various applications. By synchronizing Raspberry Pi with a GSM module, notifications or alerts can be sent via SMS when flames or smoke are detected, enhancing system responsiveness and remote monitoring capabilities. The USB Camera captures real-time video for effective fire detection and analysis.
This chapter provides a comprehensive overview of our proposed system architecture. It details the dataset arrangement process and Convolutional Neural Networks (CNNs) setup for flame and smoke detection, specifically designed for real-time Raspberry Pi implementation.

3.1 System Architecture

The proposed structure aims to achieve effective Flame and Smoke Detection. It involves gathering 20k images of flame and smoke, pre-processing through cropping and resizing, and labeling using LabelIMG software. The dataset is further processed and used to train the YOLOv5s model on Google Colab. Finally, the trained model is deployed on a Raspberry Pi to enable real-time fire and smoke detection.

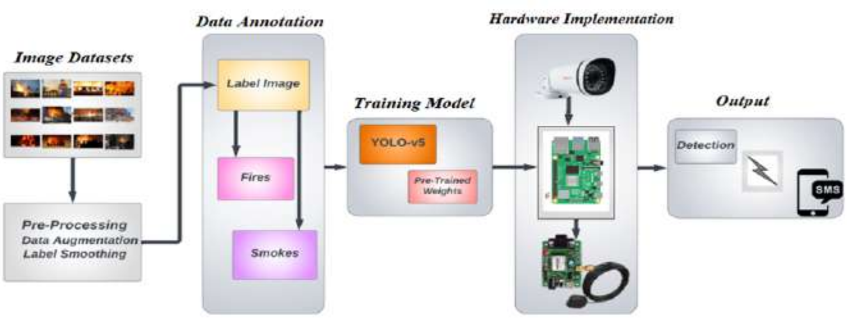

Figure 3.1: Block Diagram Of Proposed Method

Our project used YOLOv5s as the base model for training a custom model to detect fire and smoke. Here is an overview of how the model's functioning typically occurs

3.2 Data Collection and Pre-processing

3.2.1 Data Collection

Data collection is the first and one of the most crucial steps in any deep learning project. It involves gathering suitable datasets that accurately represent the problem at hand.
For our project, we collected images and videos of different types of fires and smoke conditions. This data are sourced from.
Video Feeds Video feeds from YouTube, and other video-sharing platforms are rich data sources. So we downloaded the video and extracted the frames to create a robust dataset.

3.3 Data Pre-processing

After the data has been collected, it needs to be preprocessed to ensure that it is in a suitable format for our deep learning model which involves
Cleaning The data contain irrelevant or misleading information. For instance, some images or videos may not contain any fire or smoke but could still be part of the dataset. So, we identified such data and removed unwanted images.
Labeling For our model to learn effectively, it needs to know what it looks at. Labeling the data as 'fire,' 'smoke,' or 'none.' is a time-consuming process but is crucial for supervised learning models.

3.4 Data Augmentation

We used data augmentation techniques to increase the size and diversity of our dataset. It involves artificially expanding the dataset by creating modified versions of existing images or videos.

3.4.1 Splitting the Dataset

The collected data are split into three sets: training set, validation set, and test set. We trained the model on the training set, optimized it with the validation set, and finally evaluated the performance on the test set.
The outcome of this chapter is a robust, diverse, and well-prepared dataset that is effectively used for training, validating, and testing the deep learning model for flame and smoke detection.

3.5 Label smoothing

When implementing label smoothing in flame and smoke detection, the target labels for training images are adjusted to introduce a level of uncertainty or ambiguity. Rather than assigning strict binary labels of 0 or 1 for "flame" or "non-flame"

and "smoke" or "non-smoke," label smoothing distributes the probability among the classes. Instead of a definite label of 1 for a flame or smoke class and 0 for the non-flame or non-smoke class, a fraction of the probability is allocated to each class.
By incorporating label smoothing, the model becomes less reliant on extreme predictions and develops a greater resilience to variations and uncertainties within the data. This technique helps mitigate overfitting and encourages the model to learn more representative features relevant to flame and smoke detection. Label smoothing enhances the model's generalization ability, especially when the training data may contain noise or ambiguities in labeling flame and smoke instances.

3.6 Data Annotation

Data annotation for flame and smoke detection using deep learning typically involves marking and labeling regions of interest (ROIs) within images or video frames containing flames or smoke. This annotated data serves as the ground truth for training deep learning models.
The annotation process typically includes the following steps.
Image selection Curating a diverse dataset of images or video frames that contain a wide range of flame and smoke instances, including variations in intensity, shape, size, and background.
ROI annotation Manually drawing bounding boxes or polygons around the regions of the image that contain flames or smoke. These annotations indicate the precise location and extent of the flame or smoke within each image.
Class labeling Assigning class labels to the annotated ROIs, such as "flame" or "smoke," to distinguish between the two types of instances. A separate class or label may also be assigned to non-flame/non-smoke regions or background areas.
The annotated data is then used to train deep learning models, such as convolutional neural networks (CNNs), enabling them to learn and recognize patterns associated with flames and smoke. This supervised training process allows the models to generalize and accurately detect flames and smoke in new, unseen images or video frames.

3.7 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a particular class of artificial neural network designed to handle pixel information. CNNs are multi-layer neural networks that excel in extracting informations features.They work beautifully with images and donnot require much pre-processing. We can improve our ability to recognise images by distilling them down to their most basic features using convolutions and pooling. The three layers of a CNN are the convolutional layer, the pooling layer, and the fully connected layer

3.7.1 Convolution Layer

A convolutional layer is a CNNs core component. There are numerous filters (or kernels) in it, and each ones parameters must be learned during training. The filters are typically smaller in size than the source image. Each filter convolves with the image to produce an activation map. The linear process of convolution consists of multiplying an input by a set of weights, where the input is an array of input data and the weights are an array of two-layered array.

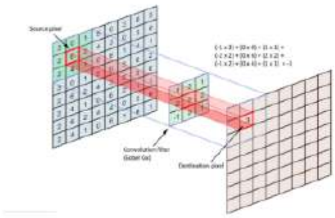

Figure 3.2: Convolution Layer

3.7.2 Hyperparameters

Hyperparameters are variables that have values decided upon before the model training procedure starts. The hyperparameters that make up the CNN structure are the number and size of the kernels for each convolution layer, the step size, and the size of the kernels in the pooling layer

3.7.2.1 Padding

Every time we convolve, the images size decreases. We utilize padding to maintain the same dimension between the output and the input. The method of padding involves equally adding zeros to the input matrix. Padding is a concept that is relevant to convolutional neural networks since it describes how many pixels are added to an image during processing by a CNNs kernel. For instance, if the padding in a CNN is set to 0, any further pixel values will have no value.

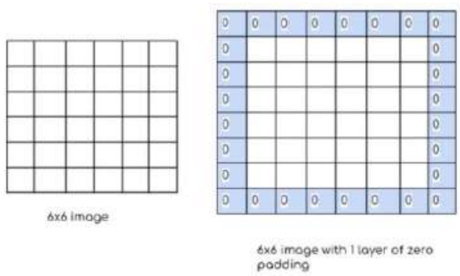

Figure 3.3: Padding

3.7.2.2 ReLU Layer

Each negative value from the filtered photos is removed and replaced with zeros in this layer. To prevent the values from adding up to 0, this is done. Rectified Linear Unit (ReLU) transform operations only commence input nodes that are above and beyond a predetermined threshold. The result is zero while the data is below a particular limit, but increases when the data exceeds a specific limit.

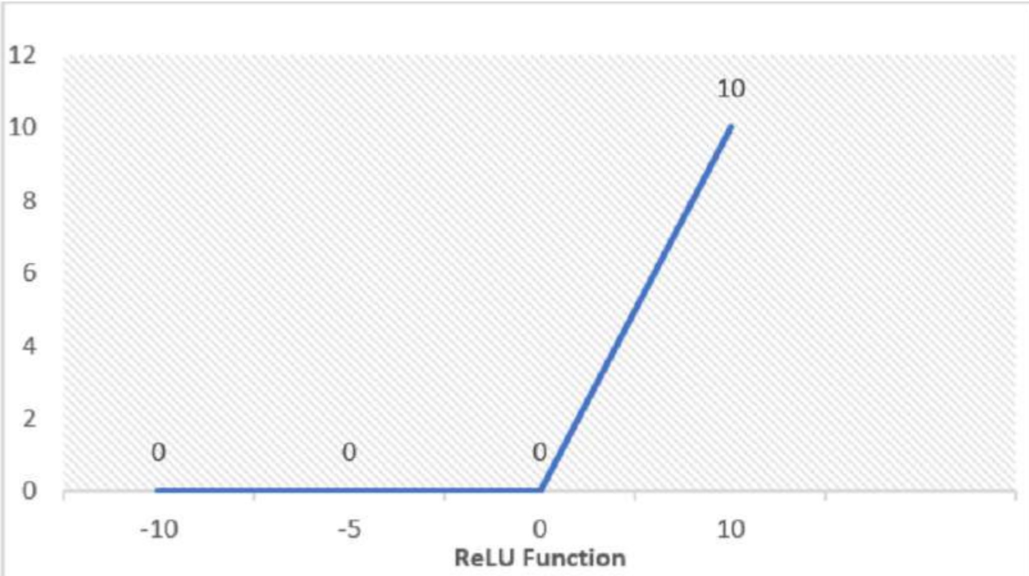

Figure 3.4: Relu function

3.7.2.3 Pooling Layer

The spatial scale of the input image is reduced by the pooling layer, which reduces the number of calculations required by the network. By reducing the size, pooling accomplishes down-sampling and provides only the important information to CNNs subsequent layers. The following are the three different kinds of pooling operations:
Maximum Pooling: The batchs highest pixel value is selected.
Min Pooling: The batchs minimum pixel value is picked.

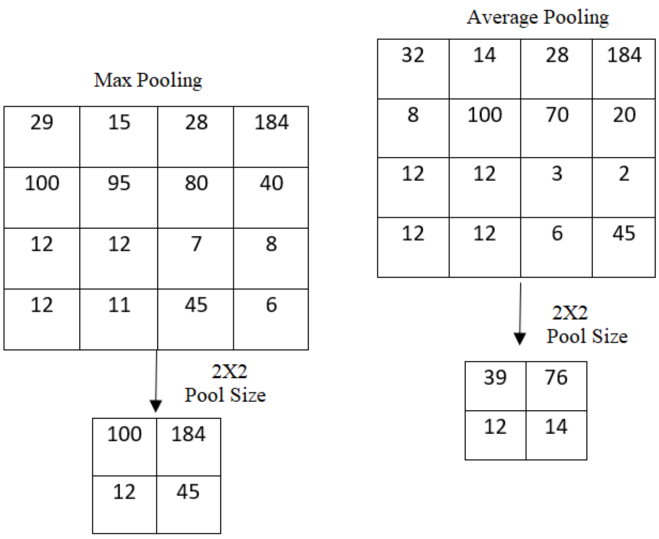

Figure 3.5: Pooling Layer

3.7.3 Fully Connected Layer

The output of convolutional and pooling layers is flattened and passed through fully connected layers. These layers learn high-level features and make predictions. Flame

and smoke detection using deep learning leverages Convolutional Neural Networks (CNNs) to analyze visual data effectively. CNNs are specially designed for structured grid-like data, such as images, and have proven to be powerful models for this task.
One of the key components of a CNN is the convolutional layer. In the context of flame and smoke detection, the convolutional layer performs convolutions on the input images using learnable filters. These filters capture local patterns and features that are relevant to the detection of flames and smoke.
An important concept in CNNs is the activation function, which introduces non-linearity to the output of the convolutional layer. Activation functions like Rectified Linear Units (ReLU) help the network learn complex patterns by applying non-linear transformations to the intermediate feature maps. Pooling layers are utilized in CNNs to downsample the feature maps and aggregate information from local regions. In the case of flame and smoke detection, pooling layers assist in reducing the spatial dimensions of the feature maps while retaining important information. Max pooling is a commonly used operation that selects the maximum value within a pooling window.
The output from the convolutional and pooling layers is flattened and passed through fully connected layers. These layers learn high-level features and make predictions based on the extracted features. For flame and smoke detection, the fully connected layers are crucial in analyzing the learned representations and classifying the presence of flames and smoke.
During training, CNNs employ a loss function to measure the discrepancy between the predicted output and the true labels of the training data. Common loss functions used in flame and smoke detection include cross-entropy and softmax loss. The network's parameters are updated using optimization techniques like stochastic gradient descent (SGD), which leverages backpropagation to calculate gradients and adjust the parameters accordingly.
The architecture and hyperparameters of a CNN, such as the number of layers, filter size, stride, pooling window size, and learning rate, play a crucial role in achieving optimal performance. These factors are determined through experimentation and hyperparameter tuning to ensure the CNN effectively detects flames and smoke in the input images.
In conclusion, using deep learning, flame, and smoke detection relies on the powerful capabilities of Convolutional Neural Networks (CNNs). These networks employ convolutional, pooling, and fully connected layers to learn hierarchical features from input images, enabling accurate detection of flames and smoke. The architecture, hyperparameters, and optimization techniques are carefully selected to train the CNN and achieve optimal performance for this specific detection task.

3.7.4 Activation Function

An activation function introduces non-linearity to the output of the convolutional layer, helping the network learn complex patterns. Rectified Linear Units (ReLU) are a common activation function.

3.7.5 Loss Function

CNNs are trained in a supervised manner using labeled data. The loss function measures the discrepancy between predicted output and true labels. Common loss functions include cross-entropy loss and softmax loss.

3.7.6 Backpropagation

CNNs are trained using backpropagation, which calculates gradients of parameters concerning the loss function. These gradients update the parameters using optimization techniques like stochastic gradient descent (SGD).

3.7.7 Architecture and Hyperparameters

CNNs have a specific architecture, including the number of layers. Hyperparameters like filter size, stride, pooling window size, and learning rate need careful selection. They are determined through experimentation and tuning.
By stacking multiple convolutional, pooling, and fully connected layers, CNNs learn hierarchical features from input data. Earlier layers capture low-level features, while deeper layers capture high-level features. CNNs have revolutionized computer vision tasks and excel in analyzing visual information. Their automatic feature learning capability makes them powerful tools in artificial intelligence research.

3.8 Working of YOLO

YOLO (You Only Look Once) is an object detection algorithm widely used in computer vision tasks. It operates by dividing the input image into a grid and making predictions directly on this grid. The network architecture of YOLO can be summarized as follows.

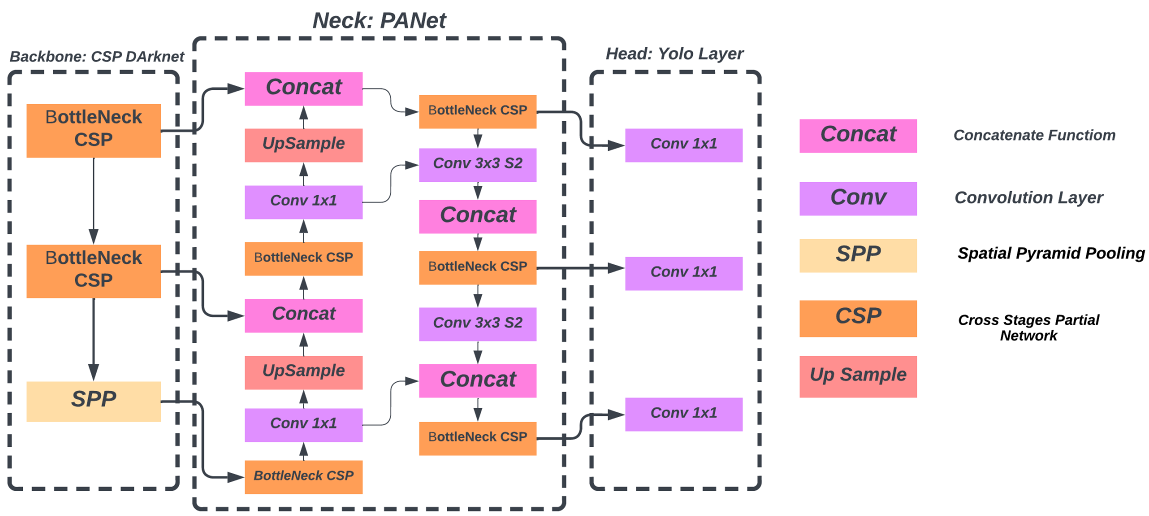

Figure 3.6: Architecture of YOLO

Input: Layer The YOLO algorithm takes an image as input.
Backbone: The backbone is the initial part of the network, responsible for extracting meaningful features from the input image. It typically consists of convolutional layers and is designed to capture low-level to high-level visual features.To extract useful and instructive information from an input image in YOLOv5, CSP (Cross Stage Partial Networks) serve as the foundational framework or framework. CSP networks are being used as the backbone with the intention of capturing and utilising important

properties and details in the image. This enhances the network's ability to interpret the content and execute tasks like object detection more effectively.
Neck: The neck component follows the backbone and further refines the features extracted by the backbone. It often includes additional convolutional or pooling layers and incorporates techniques like feature pyramid networks or skip connections to capture features at different scales and improve object detection performance.In YOLOv5, PANet works as a neck to obtain feature pyramids.
Head: The head is the final part of the network responsible for generating object detection predictions. It typically consists of fully connected layers and convolutional layers. The head takes the refined features from the neck and performs computations to predict bounding box coordinates, confidence scores for object presence, and class probabilities for different object categories.
To achieve this, YOLO divides the image into a grid of cells, such as 7x7 or 13x13, depending on the specific YOLO version. For each cell, YOLO predicts multiple bounding boxes and assigns confidence scores and class probabilities to those boxes. To refine the predictions, YOLO utilizes anchor boxes, which are predefined bounding box shapes of various sizes and aspect ratios. The algorithm predicts bounding box coordinates relative to each grid cell and adjusts them based on the anchor boxes. After the predictions, a process called non-maximum suppression (NMS) is applied to remove redundant or overlapping bounding box predictions based on their confidence scores. The remaining boxes with their associated class labels and confidence scores constitute the final object detection output.
The YOLO architecture is known for its real-time performance, as it processes the entire image in a single pass through the network. By considering the global context of the image and utilizing a unified architecture which are effectively detect the objects in real-time scenarios.

3.9 YOLOv5s Implementation

In our project flame and smoke detection using deep learning, we implemented YOLOv5 by installing essential dependencies such as Python, PyTorch, NumPy, and OpenCV. These libraries were necessary for implementing YOLOv5 and working with deep learning models. We then cloned the YOLOv5 repository from GitHub to access the required information and resources.
Afterward, we obtained pre-trained weights for YOLOv5 from either the official YOLOv5 repository. These weights served as a starting point for our implementation. We prepared a custom fire and smoke detection dataset by labelling fire and smoke in the images and creating bounding box annotations in the YOLO format.
We modified YOLOv5's configuration files, such as Coco. yaml, to meet our specific requirements. These files let us specify model parameters, dataset paths, augmentation settings, and other necessary configurations.
We initiated the training process by executing the script with the appropriate command, configuration files, and dataset paths. The script loaded the pre-trained weights and fine-tuned the model to improve its fire and smoke detection capabilities. Periodically, we saved the trained weights.
To evaluate the model's performance, we used validation or test datasets. By running the validation script with the path to the trained weights and the evaluation dataset, we generated metrics like precision, recall, and mean Average Precision (mAP) to assess the accuracy of the fire and smoke detection predictions.

We employed the trained YOLOv5s model for inference and deployment on new images or real-time applications. We loaded the trained weights into the model, processed input images through the YOLOv5 pipeline, and post-processed the predictions to obtain accurate fire and smoke detection results. These results were crucial for deploying the model in various applications requiring fire and smoke detection capabilities.
In summary, following these steps, we successfully implemented YOLOv5 for our FYP project. We configured and trained the model on our custom dataset, evaluated its performance, and deployed it for accurate fire and smoke detection. The efficient architecture and feature extraction capabilities of YOLOv5 make it a powerful tool for object detection tasks.

3.10 YOLOv5s Configuration Parameters

The YOLOv5s configuration parameters control various aspects of the YOLOv5s model, allowing you to customize its behavior and performance. Here's a brief explanation of some key concepts related to these configuration parameters
Model Configuration These parameters define the general settings of the model, such as the number of classes to detect and the scaling factors for adjusting the model's depth and width.
Input Configuration These parameters relate to the input data and preprocessing. The imgage size parameter specifies the size of input images the model expects. The augment parameter enables or disables data augmentation techniques during training, which can help improve model generalization. The mosaic parameter controls mosaic augmentation, a technique combining multiple images during training.
Model Architecture These parameters define the architecture of the YOLOv5s model. The backbone parameter specifies the backbone network, which extracts features from the input images. The neck parameter determines the architecture of the neck, which further refines the features. The head parameter defines the architecture of the detection head, which performs object detection and localization.
Training Configuration These parameters are related to the training process. The batch size determines the number of images processed in each training iteration. The epochs parameter specifies the number of times the entire training dataset is passed through the model during training. The momentum and weight decay parameters control the optimization process to prevent overfitting. The learning rate determines the initial learning rate used for training. The scheduler parameter defines the learning rate scheduler, which adjusts the learning rate during training to improve convergence.

3.10.1 Momentum

In our project on flame and smoke detection using deep learning, we employed different optimization techniques to improve the performance of our model. One such technique is momentum, which acts as the inertia of the optimization process. By accumulating updates from previous iterations, momentum helps to speed up convergence. This is particularly useful when dealing with flame and smoke detection due to noisy gradients or sparse data. Higher momentum values result in smoother and faster convergence, enhancing the overall performance of our model. However, it is crucial to avoid setting momentum too high as it can lead to overshooting or instability during training, which may affect the accuracy of our flame and smoke detection system.

3.10.2 Weight Decay

In our deep learning model, we utilized weight decay, and L2 regularization, to prevent overfitting. This technique involves adding a penalty term to the loss function during training. The goal of weight decay is to discourage the model from having large weight values, which helps reduce the complexity of our flame and smoke detection model. By encouraging the model to learn smaller weight values, weight decay prevents overreliance on specific features or memorization of the training data. This regularization technique enhances the generalization ability of our model, allowing it to detect flame and smoke patterns in various scenarios effectively.

3.10.3 Learning Rate

The learning rate is vital in optimizing our flame and smoke detection model. It determines the step size of the optimization algorithm during training. A higher learning rate enables larger updates in each iteration, accelerating our model's convergence. This is advantageous as it helps our model quickly adapt to flame and smoke patterns. However, we need to exercise caution when setting the learning rate. If the learning rate rises, it may lead to overshooting or instability during the optimization process, negatively impacting the performance of our flame and smoke detection system. On the other hand, using a lower learning rate may result in slower convergence or getting stuck in suboptimal solutions. Therefore, finding an appropriate learning rate is crucial for achieving real flame and smoke detection performance in our deep learning model. By incorporating momentum, weight decay, and an optimal learning rate, our flame and smoke detection model can effectively detect and classify flame and smoke patterns in various environments. These optimization techniques enhance our model's performance, robustness, and generalization ability, making it reliable for real-time flame and smoke detection applications.

3.11 OpenCV Implementation

OpenCV (Open Source Computer Vision) is a powerful library written in C++ that provides functionalities for machine learning, deep learning, computer vision, and image processing tasks. It supports the real-time implementation of deep learning models and offers various capabilities such as image and video processing, feature detection, and more. With the introduction of OpenCV 3.3, the library introduced the deep learning DNN (Deep Neural Network) module, which allows the integration of different deep learning frameworks such as TensorFlow, PyTorch, Caffe, and Darknet. For our specific task of flame and smoke detection using YOLOv5, we can leverage the OpenCV library and its DNN module. While OpenCV's DNN module doesn't provide training capabilities for deep learning models, it allows us to utilize our pre-trained models within OpenCV scripts. We saved the trained weights after training custom YOLOv5 flame and smoke detection models.

To develop a complete pipeline for flame and smoke detection, we utilize OpenCV and its DNN module. The process begins with capturing an image or video frame

using a camera. We define a reference rectangle within the image to specify the region where we want to detect the presence of flames and smoke. This reference rectangle helps in avoiding multiple detections of flames and smoke.
The input image is then passed through the trained YOLOv5 model, specifically designed for flame and smoke detection (YOLOv5-Flame-Smoke), which predicts the bounding boxes and class labels for the detected instances. We crop the figure to include only the bounding box that overlaps the reference rectangle to obtain a single detected instance of flames or smoke.
Next, we apply additional processing steps, such as flame and smoke analysis or classification, to refine the detection results. It may involve analyzing the intensity or shape of the detected regions, applying filtering techniques, or using additional machine learning algorithms to make more accurate predictions.
OpenCV's DNN module workflow for flame and smoke detection using the YOLOv5 model. The process involves loading the image using OpenCV, resizing it to a specific dimension (e.g., 416 x 416), loading the YOLOv5 model configuration and weights, and obtaining the output layers responsible for predicting the objects of interest. The image is passed through the network, and bounding box predictions are obtained. To eliminate unnecessary bounding boxes, Non-Maximum Suppression (NMS) is applied. Finally, the resultant image with recognized flames and smoke instances and their corresponding bounding boxes is obtained.
By leveraging the power of OpenCV and its DNN module, we developed an effective and efficient flame and smoke detection system using the trained YOLOv5 model.

3.12 Hardware Implementation

In setting up a flame and smoke detection system using a Raspberry Pi 4 B, a webcam, GSM 900 module, and NEO-6M GPS module,we go through following hardware connections.
Raspberry Pi B: Connect the Raspberry Pi to a 5v 3A power source and a remote desktop for initial setup. You will also need a keyboard and mouse for configuration.
Webcam: Connected the webcam to one of the USB ports on the Raspberry Pi. The webcam should be recognized as a video device by the Raspberry Pi.
GSM 900 Module: The GSM module usually communicates with the Raspberry Pi via a serial connection. Connect the TX transmit pin of the GSM module to the RX receive pin of the Raspberry Pi and the RX pin of the GSM module to the TX pin of the Raspberry Pi. Additionally, connect the module's power and ground pins to the respective 5V and GND pins on the Raspberry Pi.
NEO6M GPS Module: The GPS module also communicates with the Raspberry Pi via a serial connection. Connect the TX pin of the GPS module to the RX pin of the Raspberry Pi and the RX pin of the GPS module to the TX pin of the Raspberry Pi. Similar to the GSM module, connect the power and ground pins of the GPS module to the 5V and GND pins on the Raspberry Pi.

3.13 Flow Charts

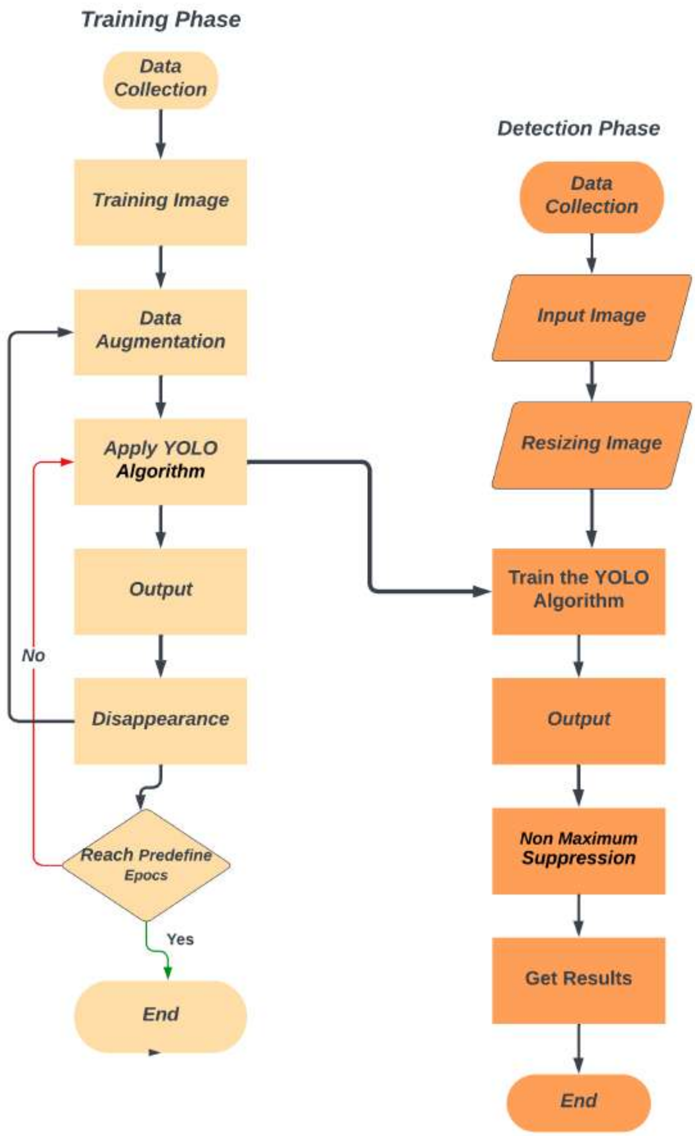

Figure 3.7: Flow Chart Of Proposed Method

According to the Fig. 3.7, In the process of flame and smoke detection using deep learning, there are several important steps. To begin, a dataset containing images of flames and smoke is gathered. These images serve as the training data for a deep learning model that learns to identify and categorize flames and smoke. To improve the model's performance and its ability to handle different scenarios, data augmentation techniques are applied. These techniques involve making modifications to the training images, such as rotating or scaling them.

Once the model is trained, the YOLO (You Only Look Once) algorithm is employed for detecting flames and smoke. This algorithm can locate and recognize multiple instances of flames and smoke within an image. It provides detailed information about the specific locations and classifications of the flames and smoke it detects.

During the training phase, special attention is given to cases where flames or smoke

might disappear or become partially obscured. Techniques like analyzing and filtering the training data are employed to effectively handle these situations and ensure accurate detection.
The training phase involves repeatedly applying the YOLO algorithm for a specific number of epochs or iterations. This iterative process helps the algorithm learn and improve its performance over time. Once the training phase is completed, the detection phase begins.
In the detection phase, an input image is provided to the trained YOLO algorithm. Before processing the image, it may need to be resized or scaled to match the algorithm's expected input size. The YOLO algorithm continues to be refined or updated based on new data or feedback through a feedback loop mechanism.
The output of the detection phase is the result of the input image being processed by the YOLO algorithm. This output includes detection results that indicate the presence and location of flames and smoke within the image. To ensure accurate and reliable results, a non-maximum suppression step is performed. This step eliminates redundant or overlapping detections, ensuring that each flame or smoke instance is detected only once.
Finally, after applying non-maximum suppression, the detection results are obtained as the final outcome of the process. This signifies the completion of the detection phase.

3.14 Pseudo Code

Algorithm 1 Pseudo Code For Proposed Method

Input: start Respberry Pi.
Input: Initialize YOLOv5s model.
Input: Initialize GSM900 module.
Input: Initialize GPS module.
Input: Initialize GPS module.
Output: Best possible Energy to run load.

```
Take images from camera
Preprocess images
perform object detection using yolov5s on the image
if flame and Smoke detected then
    Get Current GPS Coordinates from GPS module
    Send Sms Alert using GSM900 module with Flame or Smoke detected At GPS Coordinats
else
    Continue Monitoring for Flame And Smoke
```

3.15 Mathematical Modeling

The flame and smoke detection using deep learning with YOLOv5s, we employ mathematical modeling to evaluate the model's performance. The following formulas are used to calculate the metrics
Intersection over Union (IoU)

$$IoU(A, B) = \frac{A \cap B}{A \cup B} \tag{3.1}$$

Here, A represents the predicted bounding box, and B represents the ground truth bounding box. The intersection of A and B is divided by their union to obtain the IoU value. If IoU is greater than a threshold (usually 0.5), it indicates that the object is present inside the bounding box.
Average Precision (AP)

$$AP = \frac{\text{True Positive}(c)}{\text{True Positive}(c) + \text{False Positive}(c)} \tag{3.2}$$

AP represents the average precision for a specific class (c). True Positive (c) refers to the number of correctly detected instances of class c, while False Positive (c) refers to the number of incorrectly detected instances of class c.
Mean Average Precision (mAP):

$$mAP = \frac{1}{\text{number of classes}} \sum_{c} AP(c) \tag{3.3}$$

mAP calculates the average precision across all classes. It is obtained by summing up the AP values for each class (c) and dividing it by the total number of classes.
By applying these formulas, We assessed the performance of the YOLOv5s model in flame and smoke detection

3.16 Component Selection

In our proposed methodology for flame and smoke detection, we have carefully selected specific components to ensure efficient and accurate results. Firstly, we opted for the Raspberry Pi 4B model as our main processing unit due to its powerful capabilities and flexibility. Its advanced computing power and ample memory make it suitable for handling complex algorithms and real-time image processing. To capture the visual data, we have chosen a USB camera known for its high-resolution and reliable performance, allowing us to obtain clear and detailed images for analysis. Additionally, we integrated a GSM900 module to enable immediate alert notifications in case of fire emergencies. This module ensures that relevant authorities and stakeholders are promptly informed, enhancing the safety response. Furthermore, we included a GPS module to track the location of the device and provide accurate information about the incident's geographical coordinates. The selection of these components showcases a well-rounded approach to our project, enabling us to develop an effective flame and smoke detection system.

3.17 Hardware And Software Setup

- Processor Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz, 2000 Mhz, 2 Core(s), 4 Logical Processor(s) RAM 12GB/DDR3 Ram/CPU core2/frequency 2GHZ
- Window 11,10
- Python 3.9

- LabelImg
- Notepad text Editor
- Respeberry Pi 4B 4gb
- Gps NEO-6M-0-001
- Webcam 640*480/30fps
- Putty
- Vnc Viewer
- Google Colab Tesla T4,25GB RAM,16gb gpu

Chapter 4

Results And Simulations

4.1 Simulation Results

4.1.1 Precision Recall Curve

These two metrics provide complementary insights into the model's performance. Precision focuses on the accuracy of positive predictions, while recall emphasizes the model's ability to identify all positive instances.the precision and recall have an inverse relationship. As the model becomes more cautious in making positive predictions, precision typically increases while recall may decrease, and vice versa. The balance between precision and recall depends on the specific requirements of the problem and can be adjusted by changing the decision threshold of the model's classification output.

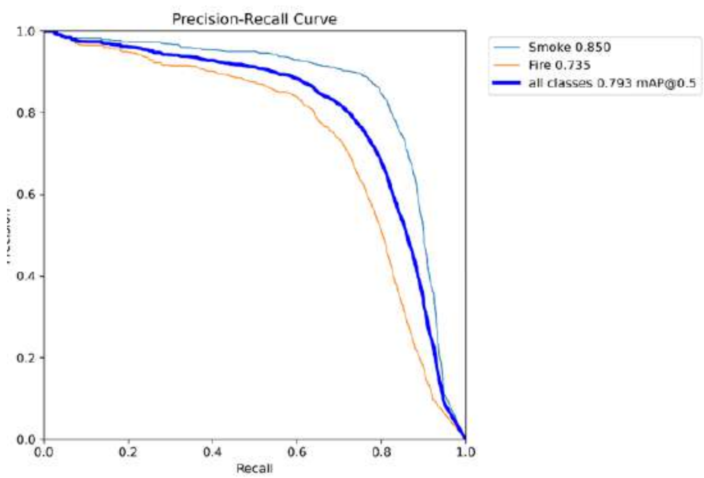

Figure 4.1: Precision Recall Curve

According to Fig. 4.1, The precision-recall curve graphically depicts the balance between precision and recall at different thresholds for classification. It gives us a deeper understanding of the model's performance by examining various levels of precision and recall.

In the present case, the precision-recall curve, the model's performance of "smoke" and "fire." which is 0.850 for smoke and 0.735 for fire signifying the accuracy of pos-

itive predictions. These scores demonstrate how well the model correctly identifies instances belonging to each class.
Furthermore, the average precision (AP) of 0.793 is across all classes when the threshold is set at 0.5. This metric provides an overall evaluation of the model's performance by averaging the precision values across the different classes.

4.1.2 Precision Confidence Curve

The precision-confidence curve showing how precision changes with confidence thresholds. Higher thresholds lead to higher precision, indicating cautious positive predictions. Lower thresholds result in decreased precision.

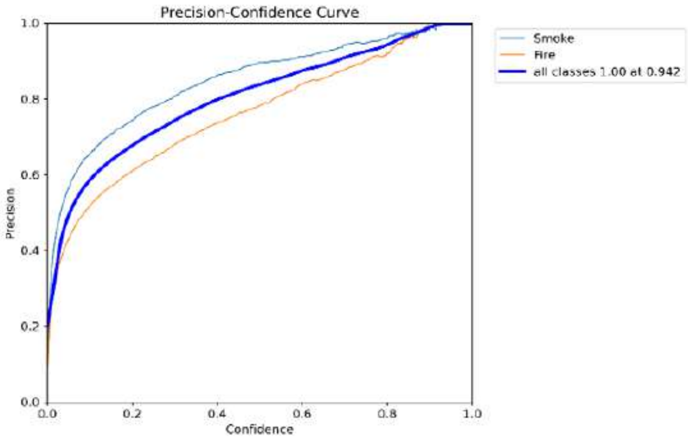

Figure 4.2: Precision Confidence Curve

According to Fig. 4.2, The precision-confidence curve obtained from our project on flame and smoke detection using deep learning reveals an impressive precision of 0.942 across all classes. This indicates the model's exceptional accuracy in distinguishing instances as either flame or smoke. Notably, as we vary the confidence thresholds, the precision consistently remains high. This consistency underscores the model's reliability and resilience in its performance. The precision-confidence curve effectively showcases the model's capacity to generate confident and accurate predictions, furnishing valuable insights that can assist in decision-making for flame and smoke classification tasks.

4.1.3 Recall Confidence Curve

The recall-confidence curve depicts the connection between recall and confidence levels in a binary classification model. By adjusting the confidence threshold, we can observe how the recall changes. At higher thresholds, the model becomes more cautious, resulting in a lower recall. Conversely, at lower thresholds, the model becomes more lenient, leading to a higher recall. This curve assists in determining the best confidence threshold to maximize recall, offering valuable insights into the model's ability to capture positive instances accurately.

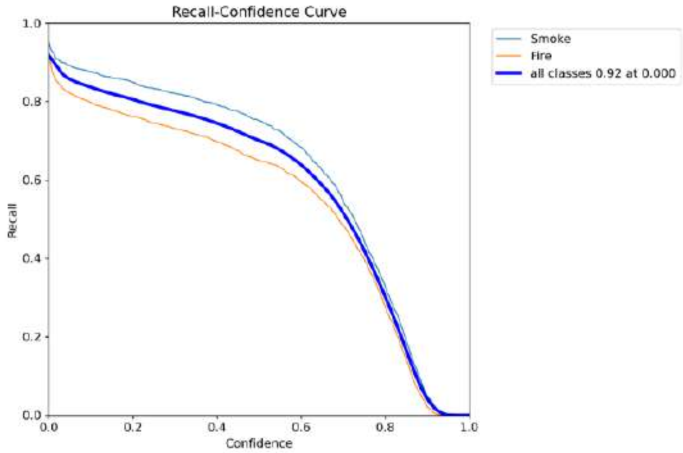

Figure 4.3: Recall Confidence Curve

According to Fig. 4.3, As we get the recall-confidence curve of our flame and smoke detection project using deep learning which are demonstrating a recall of 0.92 across all classes at a confidence threshold of 0.000. This signifies that our model successfully identifies and captures a high proportion of positive instances, encompassing both flame and smoke, even at an extremely low confidence threshold.
As we decrease the confidence threshold, the curve consistently maintains a high recall value. This implies that the model effectively detects the majority of positive instances, including both true positives and some false positives, even when the confidence in those predictions is exceptionally low.

4.1.4 F1 Confidence Curve

The F1-confidence curve represents the relationship between the F1 score, which considers both precision and recall, and confidence levels in a deep learning model for flame and smoke detection. It shows how the F1 score varies as the confidence threshold is adjusted, indicating the model's performance in balancing precision and recall at different confidence levels.

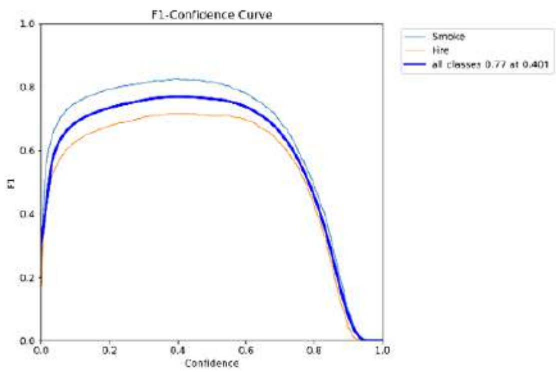

Figure 4.4: F1 Confidence Curve

According to Fig. 4.4, As we get the F1-confidence curve which demonstrating an F1

score of 0.777 across all classes. When the confidence threshold is set at 0.401. The F1 score is a significant metric as it considers both precision and recall, providing a comprehensive assessment of the model's ability to balance these aspects.
The curve visually depicts how the F1 score changes as we adjust the confidence threshold. Higher confidence thresholds generally result in higher F1 scores, indicating a better equilibrium between precision and recall. This implies that when the model is more confident in its predictions, it achieves a higher F1 score by accurately identifying true positives while minimizing the occurrence of false positives and false negatives.

4.2 Experimental Requirements

As Shown in Fig. 4.5, we collected a total of 20,887 images in our flame and smoke detection project using YOLOv5s. These images were divided into two classes: Class 1 represents fire with 11,887 images, and Class 0 represents smoke with 9,000 images.

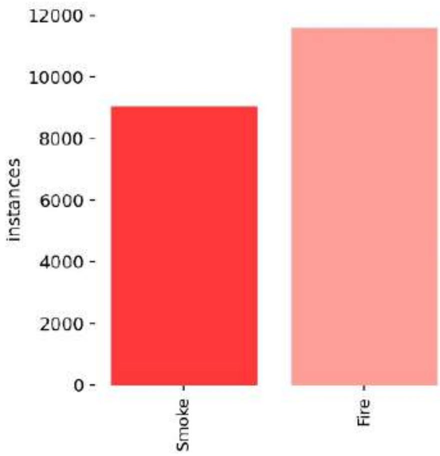

Figure 4.5: Dataset Image Collection

To train our model, we split the dataset into a training set and a testing set. The training set consisted of 80 percent of the images, while the testing set contained 20 percent of the images.
For training, we utilized the YOLOv5s network, which is an advanced version of YOLO (You Only Look Once). The training process implemented using the YOLOv5s framework in Python 3.9, running on Windows 11.
The testing set was crucial in evaluating the performance of our trained YOLOv5s model in terms of accurately detecting flame and smoke. Our primary objective in this project is to develop a reliable and efficient solution for real-time flame and smoke detection using YOLOv5s.
The selection of hyperparameters in our flame and smoke detection project using YOLOv5s plays a crucial role in determining the values of these parameters, which are independently adjusted by the learning algorithm during training.
The YOLOv5s model consists of three detection layers: 82, 94, and 106. Each of these layers is responsible for detecting objects. During the training process, we conducted 150 iterations and utilized 32 subdivisions with 16 batches. The training images were

resized to a dimension of 640 x 640 pixels. We incorporated leaky ReLU activation functions in our model architecture.
The training loss in our model is determined by three main components: object loss, classification loss, and coordinate loss. These components contribute to optimizing the model's ability to accurately detect and classify flame and smoke instances.

4.2.1 Visual Based Result

We employ the trained model to detect explosives in various scenes and display the results of the detection. The model network can identify approximately 91% of the afflicted area for smoke and flames in various environments, as depicted in the figure 4.6 and 4.7.

Figure 4.6: Software Based Validation Results

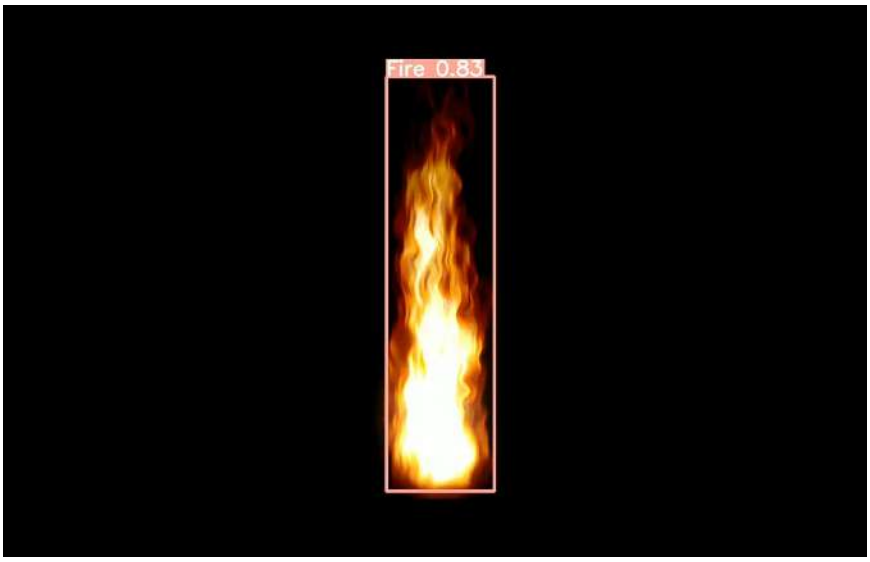

Figure 4.7: Software based Tested Result

4.3 Hardware Circuit

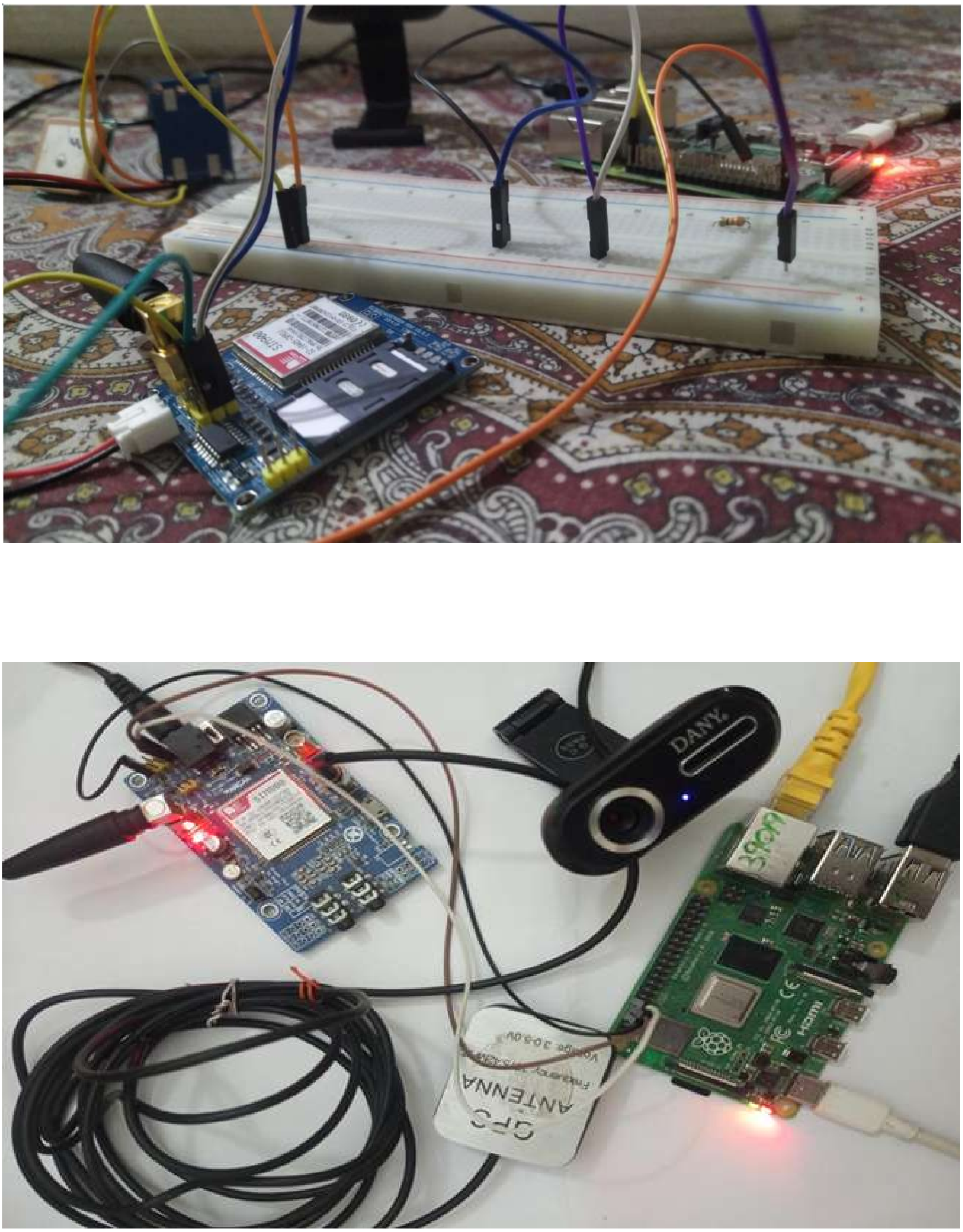

Figure 4.8: Hardware Circuit

4.4 Hardware Based Result

This is hardware result of our project flame and smoke during real time Detection.

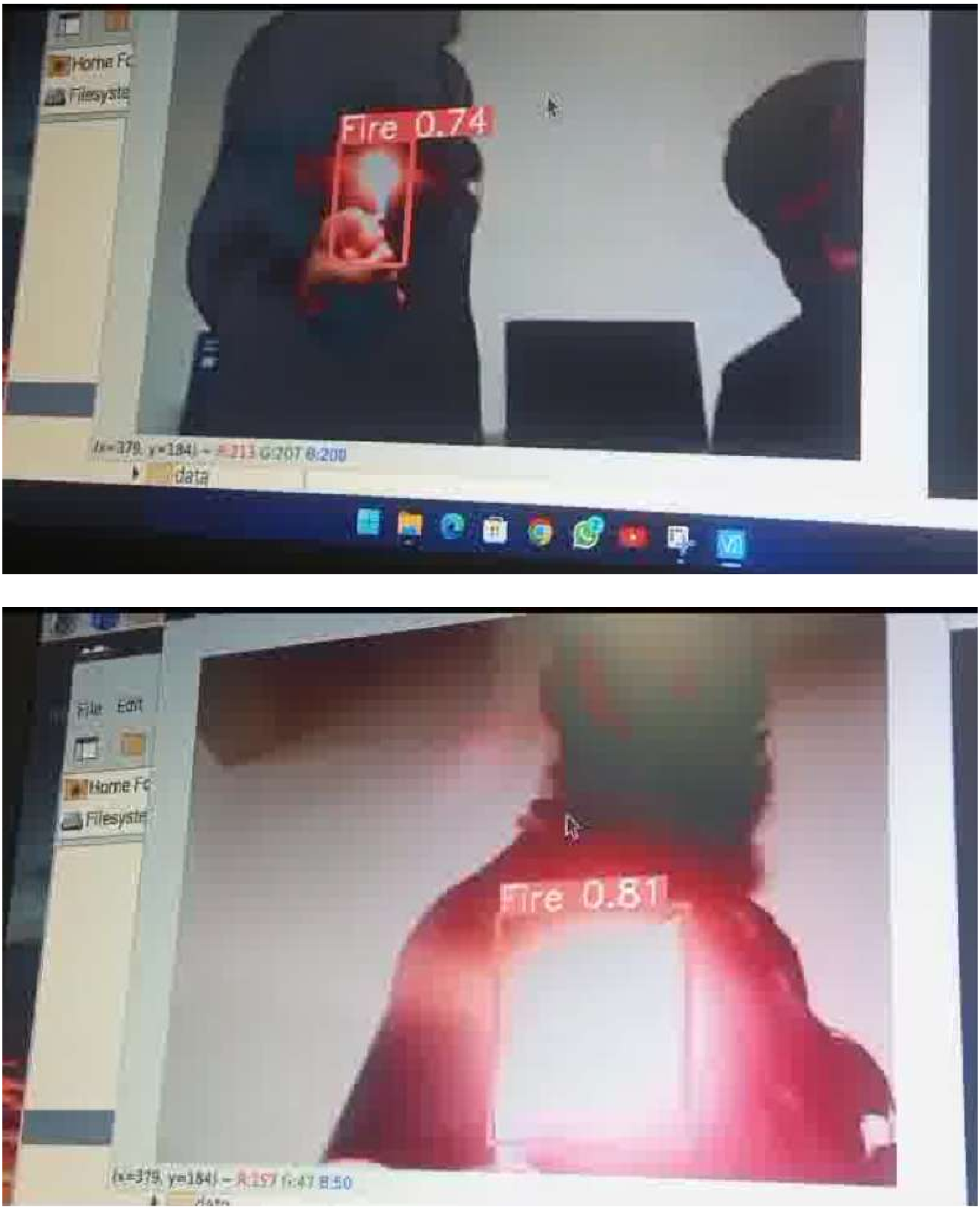

Figure 4.9: Hardware Based Simulation Result

Chapter 5

Conclusions And Future Work

5.1 Conclusions

This project's objective was to create a real-time combustion and smoke detection system using deep learning techniques, a Raspberry Pi 4, a Pi/USB camera, and a GSM 900 module. We achieved our objective through meticulous planning, data collection, model training, hardware configuration, and system integration.
Our deep learning model was effectively trained to recognize flames and smoke under a variety of visual conditions. It was then deployed on a Raspberry Pi 4 B+ with 4GB of RAM, operating in tandem with a Pi/USB Camera to process live video streams and detect fire incidents as quickly as possible. Once a fire is detected, the system immediately transmits an alert through the GSM 900 module, providing both immediate notification and the precise GPS location of the fire. This allows for swift and effective firefighting measures.
Extensive testing has confirmed the system's dependability in detecting flames and smoke and delivering alerts in a timely manner. The system establishes an optimal balance between precision and speed, making it a useful tool for fire detection in a variety of environments.

5.2 Future Work

Future enhancements can optimize this undertaking even further. System capabilities can be enhanced by implementing the most recent version of YOLO (You Only Look Once), employing the most recent model of Raspberry Pi, utilizing an upgraded GSM module, and relying on Google APIs instead of a GPS module. Incorporating a larger and more diverse dataset of fire and smoke scenarios can also improve the model's robustness and generalizability.

This project demonstrates the successful development of a real-time fire and smoke detection system, and future enhancements have the potential to enhance its performance, accuracy, and integration with the latest technologies.

5.3 Limitations

When we working on a project "Flame and Smoke Detection Using YOLOv5s with Raspberry Pi, GSM900, GPS Module, USB Camera," several limitations can arise. Here are some potential limitations to consider.
Accuracy of Detection: In real-world scenarios, the accuracy of flame and smoke detection using YOLOv5s in our project may be affected by a number of factors, such as the quality and resolution of the USB camera. It should be noted, however, that the camera's efficacy is compromised due to its low pixel count, resulting in a reduction in precision. In contrast, during virtual simulations, where resolution limitations are not a factor, the detection process is extraordinarily efficient, yielding precise results in nanoseconds.

Hardware Limitations: It is essential to consider the strengths and weaknesses of various components, such as the Raspberry Pi, GSM900, GPS module, and USB camera. These components have limitations, such as restricted processing power, storage capacity, or communication range, among others. Because of these constraints, the system's real-time performance, data storage capacity, and capacity for remote monitoring are all affected.

Detection Range and Coverage: In our project the detection range and coverage area of the system have limitations due to the characteristics of the local USB camera.

Cost and Scalability: The cost associated with the hardware components, especially when considering higher-end devices like the latest version of the Raspberry Pi with increased storage, processing power, and RAM, can be significantly high. Additionally, the inclusion of a high-resolution camera adds to the expenses. At this stage, we are unable to afford such expensive components. Furthermore, deploying the system on a larger scale, particularly in extensive surveillance networks, would necessitate additional resources and financial considerations.

Environmental Factors: It is possible that environmental elements, such as extreme weather conditions (for example, rain or fog), obstacles in the camera's field of view, or variations in ambient illumination, could have an effect on the efficiency of the system. It is possible that the presence of these elements will make it more difficult to accurately identify smoke and flames.

Bibliography

[1] T. Yang, S. Xu, W. Li, H. Wang, G. Shen, and Q. Wang, "A smoke and flame detection method using an improved yolov5 algorithm," in *2022 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 366–371, IEEE, 2022.

[2] D. Shen, X. Chen, M. Nguyen, and W. Q. Yan, "Flame detection using deep learning," in *2018 4th International conference on control, automation and robotics (ICCAR)*, pp. 416–420, IEEE, 2018.

[3] G. Son, J.-S. Park, B.-W. Yoon, and J.-G. Song, "Video based smoke and flame detection using convolutional neural network," in *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 365–368, IEEE, 2018.

[4] Z. Wu, R. Xue, and H. Li, "Real-time video fire detection via modified yolov5 network model," *Fire Technology*, vol. 58, no. 4, pp. 2377–2403, 2022.

[5] A. Mozo, B. Ordozgoiti, and S. Gomez-Canaval, "Forecasting short-term data center network traffic load with convolutional neural networks," *PloS one*, vol. 13, no. 2, p. e0191939, 2018.

[6] W. Wang and T. Li, "Fire video image detection based on a convolutional neural network," in *Journal of Physics: Conference Series*, vol. 1453, p. 012161, IOP Publishing, 2020.

[7] X. Zheng, F. Chen, L. Lou, P. Cheng, and Y. Huang, "Real-time detection of full-scale forest fire smoke based on deep convolution neural network," *Remote Sensing*, vol. 14, no. 3, p. 536, 2022.

[8] P.-F. Tsai, C.-H. Liao, and S.-M. Yuan, "Using deep learning with thermal imaging for human detection in heavy smoke scenarios," *Sensors*, vol. 22, no. 14, p. 5351, 2022.

[9] S. Chaturvedi, P. Khanna, and A. Ojha, "Comparative analysis of traditional and deep learning techniques for industrial and wildfire smoke segmentation," in *2021 Sixth International Conference on Image Information Processing (ICIIP)*, vol. 6, pp. 326–331, IEEE, 2021.

[10] B. Wei and W. Gao, "Image processing of casting defects based on convolutional neural network," in *Journal of Physics: Conference Series*, vol. 2137, p. 012059, IOP Publishing, 2021.

[11] B. Zhang, L. Sun, Y. Song, W. Shao, Y. Guo, and F. Yuan, "Deepfirenet: A real-time video fire detection method based on multi-feature fusion," *Mathematical biosciences and engineering*, vol. 17, no. 6, pp. 7804–7818, 2020.

[12] A. Gagliardi, F. de Gioia, and S. Saponara, "A real-time video smoke detection algorithm based on kalman filter and cnn," *Journal of Real-Time Image Processing*, pp. 1–11, 2021.

[13] W. Yuanbin and M. Xianmin, "Early fire detection for high space based on video-image processing," in *2014 International Symposium on Computer, Consumer and Control*, pp. 785–788, IEEE, 2014.

[14] C. Bahhar, A. Ksibi, M. Ayadi, M. M. Jamjoom, Z. Ullah, B. O. Soufiene, and H. Sakli, "Wildfire and smoke detection using staged yolo model and ensemble cnn," *Electronics*, vol. 12, no. 1, p. 228, 2023.

[15] A. M. Fernandes, A. B. Utkin, and P. Chaves, "Automatic early detection of wildfire smoke with visible light cameras using deep learning and visual explanation," *IEEE Access*, vol. 10, pp. 12814–12828, 2022.

[16] H. U. A. Tahir, A. Waqar, S. Khalid, and S. M. Usman, "Wildfire detection in aerial images using deep learning," in *2022 2nd International Conference on Digital Futures and Transformative Technologies (ICoDT2)*, pp. 1–7, IEEE, 2022.

[17] Y. Wang, C. Hua, W. Ding, and R. Wu, "Real-time detection of flame and smoke using an improved yolov4 network," *Signal, Image and Video Processing*, vol. 16, no. 4, pp. 1109–1116, 2022.

[18] Z. Wang, L. Wu, T. Li, and P. Shi, "A smoke detection model based on improved yolov5," *Mathematics*, vol. 10, no. 7, p. 1190, 2022.

[19] K. Song, Y. Zhang, B. Lu, W. Chi, and L. Sun, "Uav forest fire detection based on repvgg-yolov5," in *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1277–1282, IEEE, 2022.

[20] J. Gotthans, T. Gotthans, and R. Marsalek, "Deep convolutional neural network for fire detection," in *2020 30th international conference radioelektronika (RADIOELEKTRONIKA)*, pp. 1–6, IEEE, 2020.

[21] A. Abdusalomov, N. Baratov, A. Kutlimuratov, and T. K. Whangbo, "An improvement of the fire detection and classification method using yolov3 for surveillance systems," *Sensors*, vol. 21, no. 19, p. 6519, 2021.

[22] P. Dai, Q. Zhang, G. Lin, M. Masoon, Y. Huo, R. Tu, and Y. Zhang, "Multi-scale video flame detection for early fire warning based on deep learning," *Frontiers in Energy Research*, p. 228, 2022.

[23] F. Gong, C. Li, W. Gong, X. Li, X. Yuan, Y. Ma, and T. Song, "A real-time fire detection method from video with multifeature fusion," *Computational intelligence and neuroscience*, vol. 2019, 2019.

[24] J. Johnston, K. Zeng, and N. Wu, "An evaluation and embedded hardware implementation of yolo for real-time wildfire detection," in *2022 IEEE World AI IoT Congress (AIIoT)*, pp. 138–144, IEEE, 2022.

[25] M. Hashemzadeh, N. Farajzadeh, and M. Heydari, “Smoke detection in video using convolutional neural networks and efficient spatio-temporal features,” *Applied Soft Computing*, vol. 128, p. 109496, 2022.

[26] S. Tammina, “Transfer learning using vgg-16 with deep convolutional neural network for classifying images,” *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, pp. 143–150, 2019.

[27] M. S. M. Khan, M. Ahmed, R. Z. Rasel, and M. M. Khan, “Cataract detection using convolutional neural network with vgg-19 model,” in *2021 IEEE World AI IoT Congress (AIIoT)*, pp. 0209–0212, IEEE, 2021.

[28] M. F. Haque, H.-Y. Lim, and D.-S. Kang, “Object detection based on vgg with resnet network,” in *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–3, IEEE, 2019.

[29] D. Zhang, J. Han, Y. Zhang, and D. Xu, “Synthesizing supervision for learning deep saliency network without human annotation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 7, pp. 1755–1769, 2019.

[30] J. Schlemper, J. Caballero, J. V. Hajnal, A. N. Price, and D. Rueckert, “A deep cascade of convolutional neural networks for dynamic mr image reconstruction,” *IEEE transactions on Medical Imaging*, vol. 37, no. 2, pp. 491–503, 2017.

[31] N. Hamid, R. A. Razali, and Z. Ibrahim, “Comparing bags of features, conventional convolutional neural network and alexnet for fruit recognition,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 1, p. 333, 2019.

[32] P. Li and W. Zhao, “Image fire detection algorithms based on convolutional neural networks,” *Case Studies in Thermal Engineering*, vol. 19, p. 100625, 2020.

[33] S. Liu and Z. Liu, “Multi-channel cnn-based object detection for enhanced situation awareness,” *arXiv preprint arXiv:1712.00075*, 2017.

[34] M. Hassanpour and H. Malek, “Learning document image features with squeezenet convolutional neural network,” *International Journal of Engineering*, vol. 33, no. 7, pp. 1201–1207, 2020.

[35] I. Khalfaoui-Hassani, T. Pellegrini, and T. Masquelier, “Dilated convolution with learnable spacings,” *arXiv preprint arXiv:2112.03740*, 2021.