**HUMAN LIFE DETECTION DURING FIRE**

## A PROJECT REPORT

***Submitted by,***

**SUNKU SAI YASWANTH - 20211ISD0007**

**GAJJALA AKHILA - 20211ISD0015**

**K PAVAN KUMAR - 20211ISD0027**

### *Under the guidance of,*

**Ms. MONISHA GUPTA**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION SCIENCE AND ENGINEERING**

**At**



**PRESIDENCY UNIVERSITY**

**BENGALURU**

**JANUARY 2025**

**PRESIDENCY UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE ENGINEERING**

**CERTIFICATE**

This is to certify that the Project report **“HUMAN LIFE DETECTION DURING FIRE”** being submitted by “SUNKU SAI YASWANTH, GAJJALA AKHILA, K PAVAN” bearing roll number(s) “20211ISD0007, 20211ISD0015, 20211ISD0027” in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in **Information Science and Engineering** is a bonafide work carried out under my supervision.

|  |  |
| --- | --- |
| **Ms. Monisha Gupta**  Assistant Professor  School of CSE&IS  Presidency University | **Dr. Pallavi R**  Professor& HoD  School of CSE&IS  Presidency University |

|  |  |  |
| --- | --- | --- |
| **Dr. L. SHAKKEERA**  Associate Dean  School of CSE  Presidency University | **Dr. MYDHILI NAIR**  Associate Dean  School of CSE  Presidency University | **Dr. SAMEERUDDIN KHAN**  Pro-Vc School of Engineering  Dean -School of CSE&IS  Presidency University |

**PRESIDENCY UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE ENGINEERING**

**DECLARATION**

We hereby declare that the work, which is being presented in the project report entitled **HUMAN DETECTION DURING FIRE** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Information Science and Technology**, is a record of our own investigations carried under the guidance of **Ms. Monisha Gupta, Assistant Professor,** **School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

|  |  |  |
| --- | --- | --- |
| **Name(s)** | **Roll No(s)** | **Signature(s)** |
| SUNKU SAI YASWANTH | 20211ISD0007 |  |
| GAJJALA AKHILA | 20211ISD0015 |  |
| K PAVAN | 20211ISD0027 |  |

**ABSTRACT**

This project focuses on the development and evaluation of a robust detection system for identifying humans, fire, and smoke in real-time using advanced deep learning algorithms YOLOv8 and YOLOv9. Leveraging a dataset from Roboflow that contains a variety of images featuring these elements, we aim to implement a model that accurately detects and classifies the presence of humans, fire, and smoke in uploaded images and live camera feeds. The system will be developed in Python using Google Colab as the integrated development environment. By employing the capabilities of YOLOv8 and YOLOv9, we will compare their performance in terms of accuracy, speed, and robustness in various detection scenarios. The model will be capable of processing images from a laptop camera, although we acknowledge potential limitations due to the camera's resolution and clarity, which may affect detection accuracy. The expected outcome is a functional application that provides real-time alerts and visual feedback when fire, smoke, or human presence is detected, enhancing safety and response measures in critical situations. This project aims to contribute to the growing field of computer vision and its applications in safety and security systems.

**ACKNOWLEDGEMENT**

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakkeera L and Dr. Mydhili Nair,** School of Computer Science Engineering & Information Science, Presidency University, and **Dr. Pallavi R**, **Head of the Department**, School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully. We are greatly indebted to our guide **Ms. Monisha Gupta, Assistant Professor** and Reviewer **Mr. Saptasi Sanyal, Assistant Professor**, School of Computer Science Engineering & Information Science, Presidency University for his/her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman,** department Project Coordinators **Mr. Srinivas Mishra** and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

**SUNKU SAI YASWANTH**

**GAJJALA AKHILA**

**K PAVAN**

**LIST OF TABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Table Name** | **Table Caption** | **Page No.** |
| 1 | Table 3.1 | Research Gaps of Existing Methods | 11 |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Figure Name** | **Caption** | **Page No.** |
|  | Figure 6.1 | Use Case Diagram of System Design | 23 |
|  | Figure 9.1 | Precision-Recall Curve yolov8 | 28 |
|  | Figure 9.2 | F1-Confidence Curve yolov8 | 29 |
|  | Figure 9.3 | Training and Validation Metrics yolov8 | 30 |
|  | Figure 9.4 | Precision-Confidence Curve yolov8 | 31 |
|  | Figure 9.5 | Confusion Matrix yolov8 | 32 |
|  | Figure 9.6 | Precision-Recall Curve yolov9 | 33 |
|  | Figure 9.7 | Recall-Confidence Curve yolov9 | 34 |
|  | Figure 9.8 | Training and Validation Metrics yolov9 | 35 |
|  | Figure 9.9 | Precision-Confidence Curve yolov9 | 36 |
|  | Figure 9.10 | Confusion Matrix yolov9 | 37 |
|  |  |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT ACKNOWLEDGMENT**  **LIST OF TABLES**  **LIST OF FIGURES** | **iv**  **v**  **vi**  **vii** |
| **1.** | **INTRODUCTION** | **1** |
|  | 1.1 Objective of the Project | 1 |
|  | 1.2 Problem Statement | 1 |
|  | 1.3 Motivation | 1 |
|  | 1.4 Scope | 2 |
|  | 1.5 Project Introduction | 3 |
| **2.** | **LITERATURE REVIEW** | **6** |
|  | 2.1 Wireless Fire Detection Monitoring System for Fire and Rescue Application | 6 |
|  | 2.2 A Human Detection Approach for Burning Building Sites Using Deep Learning Techniques | 6 |
|  | 2.3 Detection of Human Existence Using Thermal Imaging for Automated Fire Extinguisher | 6 |
|  | 2.4 Analysis of a Real-Time Fire Detection and Intimation System | 7 |
|  | 2.5 A Fire Prevention/Monitoring Smart System | 8 |
|  | 2.6 Research on Fire Detection Based on YOLOv5 | 8 |
|  | 2.7 Forest Fire Detection and Prediction from Image Processing Using RCNN | 9 |
|  | 2.8 Using Hybrid Algorithms of Human Detection Technique for Detecting Indoor Disaster Victims | 9 |
|  | 2.9 Forest Fire and Smoke Detection Using Deep Learning-Based Learning Without Forgetting | 10 |
| **3.** | **RESEARCH GAPS OF EXISTING METHODS** | **11** |
| **4.** | **PROPOSED MOTHODOLOGY** | **13** |
|  | 4.1 YOLOv8 | 13 |
|  | 4.1.1 YOLOv8 Architecture Overview | 13 |
|  | 4.1.2. Architecture Details | 13 |
|  | 4.1.3 Working Flow of YOLOv8 | 15 |
|  | 4.2 YOLOv9 | 16 |
|  | 4.2.1 YOLOv9 Architecture Overview | 16 |
|  | 4.2.2. Architecture Details | 17 |
|  | 4.2.3 Working Flow of YOLOv9 | 18 |
| **5.** | **OBJECTIVES** | 21 |
| **6.** | **SYSTEM DESIGN & IMPLEMENTATION** | 22 |
|  | 6.1 Introduction of Input Design | 22 |
|  | 6.2 Introduction Output Design | 22 |
|  | 6.3 Use Case Diagram Of System Design | 23 |
|  | 6.4 Implementation of System | 23 |
|  | 6.4.1 Data Collection | 23 |
|  | 6.4.2 Data Preprocessing | 24 |
|  | 6.4.3 Model Training | 24 |
|  | 6.4.4 Model Evaluation | 24 |
|  | 6.4.5 Model Saving | 24 |
|  | 6.5 Implementation of User | 25 |
|  | 6.5.1 Register | 25 |
|  | 6.5.2 Login | 25 |
|  | 6.5.3 Input/Upload Image Feed | 25 |
|  | 6.5.4 View Detection Results | 25 |
|  | 6.5.5 Logout | 25 |
| **7.** | **TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)** | **26** |
| **8.** | **OUTCOMES** | **27** |
| **9.** | **RESULTS AND DISCUSSIONS** | **28** |
|  | 9.1 Results for yolov8: | 28 |
|  | 9.1.1 Precision-Recall Curve | 28 |
|  | 9.1.2 F1-Confidence Curve | 29 |
|  | 9.1.3 Training and Validation Metrics | 30 |
|  | 9.1.4 Precision-Confidence Curve | 31 |
|  | 9.1.5 Confusion Matrix | 32 |
|  | 9.2 Results for yolov9: | 33 |
|  | 9.2.1 Precision-Recall Curve | 33 |
|  | 9.2.2 Recall-Confidence Curve | 34 |
|  | 9.2.3 Training and Validation Metrics | 35 |
|  | 9.2.4 Precision-Confidence Curve | 36 |
|  | 9.2.5 Confusion Matrix | 37 |
| **10.** | **CONCLUSION** | **39** |
| **11.** | **REFERENCES** | **40** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**CHAPTER-1**

**INTRODUCTION**

**1.1 OBJECTIVE OF PROJECT:**

The primary objective of this project is to develop a robust detection system capable of accurately identifying humans, fire, and smoke in real-time using advanced deep learning algorithms, specifically YOLOv8 and YOLOv9. The project aims to implement an application that processes both uploaded images and live camera feeds, facilitating immediate alerts in emergency situations. Additionally, it seeks to evaluate and compare the performance of YOLOv8 and YOLOv9 in terms of accuracy and detection speed under varying conditions. Ultimately, the project strives to enhance safety and response measures in environments prone to fire hazards and emergencies.

**1.2 PROBLEM STATEMENT:**

The increasing frequency of fire incidents and the need for immediate response to human presence in hazardous situations necessitate the development of an effective detection system. This project aims to address the challenge of real-time identification of humans, fire, and smoke using advanced deep learning algorithms, YOLOv8 and YOLOv9. We will explore the limitations of live detection via laptop cameras, which may compromise accuracy due to lower image clarity. The goal is to create a reliable detection system that can operate efficiently under varying conditions, enhancing safety measures and response times in critical environments.

**1.3 MOTIVATION:**

This project is driven by the increasing demand for intelligent surveillance systems that can autonomously monitor and identify potential hazards in real time. Fire and smoke pose significant threats to public safety, especially in industrial, residential, and remote areas where rapid detection and response can save lives and minimize damage. Traditional monitoring methods are often resource-intensive, relying on manual observation or sensor-based systems that may not always offer the accuracy or timeliness needed in critical situations.

By harnessing advanced deep learning models like YOLOv8 and YOLOv9, this project aims to leverage the latest in computer vision technology to create a responsive and accurate detection system. This system can be deployed in diverse environments, offering real-time visual alerts for fire, smoke, and unauthorized human presence. Such a solution not only reduces human error but also provides a scalable, adaptable technology that could be integrated into various safety infrastructures. This project also contributes to the development of open-source safety solutions, making it accessible for adaptation and further research in the growing field of AI-driven security and surveillance systems.

Through this project, we hope to make a tangible impact by providing a reliable, efficient tool that can significantly improve safety measures in high-risk areas, ultimately contributing to a safer and more secure environment for communities worldwide.

**1.4 SCOPE:**

The scope of this project encompasses the following key areas:

**Algorithm Implementation:** Utilize YOLOv8 and YOLOv9 algorithms to create a detection system for identifying humans, fire, and smoke in images and live camera feeds.

**Dataset Utilization:** Leverage a comprehensive dataset from Roboflow, containing diverse images of humans, fire, and smoke, to train and evaluate the detection models.

**Real-Time Detection:** Implement live detection capabilities using a laptop camera, while addressing the challenges posed by lower resolution and image clarity.

**Performance Comparison:** Analyze and compare the effectiveness of YOLOv8 and YOLOv9 in terms of detection accuracy, speed, and robustness in various scenarios.

**Safety Applications:** Explore practical applications of the detection system in enhancing safety measures in environments susceptible to fire hazards, such as buildings, industrial sites, and public spaces.

**User Interface Development:** Create an intuitive user interface that allows users to upload images and initiate live detection seamlessly.

**Limitations and Future Work:** Acknowledge the limitations of the current system and outline potential improvements for future iterations, such as integrating higher-quality cameras or refining detection algorithms.

**1.5 PROJECT INTRODUCTION:**

The rapid development of technology in recent years has transformed how we approach safety and surveillance, creating new opportunities for monitoring environments prone to critical hazards. In this context, the proposed project aims to develop a real-time detection system that uses advanced deep learning algorithms, YOLOv8 and YOLOv9, to identify humans, fire, and smoke in various settings. This system is built to provide immediate visual and auditory alerts, making it a valuable tool for enhancing safety in environments where fire and unauthorized human presence could lead to severe risks. The system uses a dataset from Roboflow, rich in diverse images of fire, smoke, and human elements, to train the models and ensure high accuracy across varying conditions. This project is developed in Python, with Google Colab as the primary integrated development environment, enabling a streamlined approach to training and testing the models.

Traditional fire and smoke detection methods rely on smoke detectors, thermal sensors, and human surveillance, each of which has limitations. Smoke detectors, for example, are often limited by sensitivity issues and can generate false alarms due to dust or steam. Meanwhile, human monitoring can be inconsistent and costly, requiring continuous attention and resources. The proposed detection system addresses these challenges by harnessing YOLO (You Only Look Once), a state-of-the-art object detection framework that has evolved significantly with each new version, offering faster and more accurate detection capabilities. YOLOv8 and YOLOv9 are the latest iterations, designed with enhanced architecture and optimization techniques that allow for improved performance in real-time applications. By implementing these models, the system can quickly analyze visual data and detect the presence of fire, smoke, and humans, even in low-resolution or cluttered backgrounds, making it highly adaptable for use with a laptop camera or other standard video feeds.

The core objective of this project is to build a responsive and accurate detection model that can process both live feeds and uploaded images. By doing so, the system becomes versatile, capable of functioning in diverse environments, including industrial sites, residential complexes, forests, and public spaces. This versatility is achieved through a comprehensive approach to training, wherein the dataset is meticulously curated and augmented to expose the models to various scenarios. The Roboflow dataset used for this project includes images with different lighting conditions, angles, and scales, allowing the models to learn and generalize from a broad range of situations. Augmenting the dataset with images that contain multiple instances of fire, smoke, or humans further strengthens the model's ability to accurately differentiate between these elements in complex scenes.

One of the significant advantages of using YOLOv8 and YOLOv9 in this project is their capability for real-time processing, which is essential in safety-critical applications where delays can have severe consequences. YOLOv8 and YOLOv9 use a unified deep learning architecture optimized for speed and accuracy, making them well-suited for scenarios where decisions must be made instantaneously. YOLOv9, in particular, introduces advanced features such as an enhanced path aggregation network, attention mechanisms, and adaptive image resizing, which further improve its detection capabilities. These features allow the system to accurately identify small, fast-moving objects, such as flames or humans, in the field of view, making it a highly reliable tool for rapid detection and response.

Implementing the detection system in Google Colab provides additional advantages, as it allows for the use of powerful GPUs, reducing the training time and enabling rapid prototyping. With Colab’s collaborative environment, the project can be easily shared and updated, facilitating further improvements and experimentation. The system is built using Python, a versatile programming language that supports a range of libraries essential for computer vision, including OpenCV, TensorFlow, and PyTorch. By leveraging these resources, the project not only achieves efficient model training but also makes the system accessible for future expansion, such as integrating additional detection capabilities or adapting the model to different hardware.

In practical terms, the detection system offers real-time alerts, which are crucial for applications requiring instant response. For instance, in a factory setting, the system can alert personnel of any unusual presence of fire or unauthorized human entry, prompting immediate action to prevent accidents. In outdoor environments, such as forest reserves, the system can detect early signs of smoke, enabling proactive firefighting efforts and potentially saving vast areas from damage. Additionally, this project contributes to the field of computer vision and deep learning by providing insights into the comparative performance of YOLOv8 and YOLOv9, especially in a high-stakes application like fire and human detection.

The anticipated outcome of this project is a fully functional application that serves as a reliable safety tool, enhancing situational awareness and response capabilities. This project underscores the potential of computer vision in addressing real-world challenges, particularly in areas of public safety and disaster management. By accurately detecting fire, smoke, and human presence, the system supports quick, data-driven decisions, ultimately helping to protect lives and reduce risks in critical environments. The project also sets a foundation for future work, such as extending the detection system to identify additional hazards, improving its adaptability to different environments, or integrating it with drones and IoT devices for remote monitoring in inaccessible areas.

In conclusion, this project demonstrates the practical applications of deep learning in creating solutions that address pressing safety needs. The development of this detection system not only advances the capabilities of fire, smoke, and human detection but also contributes to the broader goal of making environments safer and more secure through intelligent automation. By leveraging YOLOv8 and YOLOv9, this project exemplifies the impact that state-of-the-art technology can have on improving emergency response and disaster prevention efforts.

**CHAPTER-2**

**LITERATURE SURVEY**

**2.1 Wireless Fire Detection Monitoring System for Fire and Rescue Application (2015)**

"This study introduces a wireless fire detection monitoring system designed to enhance fire response and rescue operations. The system utilizes wireless communication and sensor networks to deliver real-time fire status updates to emergency services. By facilitating remote monitoring and rapid information relay, the research demonstrates how wireless technology can significantly improve the efficiency and effectiveness of fire rescue operations."

**Advantages**:

Real-time updates, remote monitoring capabilities, and faster communication enhance response efficiency and coordination in fire rescue operations.

**Limitations**:

Dependence on wireless network reliability, potential interference, and high setup costs for deploying sensor networks across large areas.

### 2.2 A Human Detection Approach for Burning Building Sites Using Deep LearningTechniques(2018) This study explores the application of deep learning techniques for human detection in burning building sites. The research focuses on using advanced algorithms to identify and locate individuals in hazardous environments, such as buildings engulfed in flames. By leveraging deep learning models, the study aims to improve rescue operations by accurately detecting human presence in low-visibility or dangerous conditions, enabling faster and more efficient responses.

**Advantages:**

accurate human detection, improving the safety of rescue teams. It allows for quick identification of individuals in dangerous situations, potentially saving lives by locating victims faster.

**Limitations:**

This system requires large datasets for training, high processing power for real-time use, and may struggle in extreme conditions such as thick smoke or flames. Additionally, it can be challenging to integrate with existing fire rescue systems.

### 2.3 Detection of Human Existence Using Thermal Imaging for Automated Fire Extinguisher (2020)

This study developed a thermal imaging-based system for detecting human presence in fire scenarios. The system is integrated with an automated fire extinguishing mechanism, aiming to enhance safety by detecting individuals in danger and triggering fire suppression actions accordingly.

**Advantages:**

The system provides high accuracy, rapid detection, and adaptability to various environments. It is also suitable for real-time applications, all of which enhance the overall effectiveness of fire monitoring and response.

**Limitations:**

The system is limited by the accuracy of thermal detection, which may face challenges in environments with fluctuating temperatures or when distinguishing human heat from other heat sources.

**2.4 Analysis of a Real-Time Fire Detection and Intimation System (2020)**

"This research focuses on the development of a real-time fire detection and alerting system designed to reduce response times to fire incidents. By combining image processing techniques with sensor data, the system detects fires and immediately notifies relevant personnel. The study highlights the importance of real-time systems in mitigating fire hazards through instant alerts, enabling early intervention and control."

**Advantages**:

Rapid detection and notification, enhanced response times, and integration of image processing for accurate fire identification.

**Limitations**:

Dependence on sensor and image processing accuracy, potential false alarms, and the need for robust infrastructure to support real-time operations.

**2.5 A Fire Prevention/Monitoring Smart System (2021)**

"This study proposed a smart fire prevention and monitoring system designed to detect potential fire hazards before they escalate. Leveraging IoT devices and sensors, the system continuously monitors environmental conditions, providing real-time data for fire prevention and early detection. By integrating intelligent technologies, the study highlights the potential to enhance response times and reduce fire-related risks, offering a proactive approach to fire safety."

**Advantages**:

Real-time data monitoring, early detection, automation, and scalability enhance fire prevention efficiency and reduce response times.

**Limitations**:

High initial costs, reliance on network connectivity, maintenance needs, and potential data security concerns may limit system adoption.

**2.6 Research on Fire Detection Based on YOLOv5 (2021)**

"This study examines the use of the YOLOv5 deep learning model for fire detection, showcasing its effectiveness in accurately identifying fires across diverse environmental conditions. With its high-speed processing and robust object detection capabilities, YOLOv5 proves to be well-suited for real-time fire detection. The research underscores the potential of AI-based solutions in delivering fast and reliable fire monitoring systems."

**Advantages**:

High accuracy, rapid detection, adaptability to various environments, and suitability for real-time applications enhance fire monitoring effectiveness.

**Limitations**:

Requires substantial computational resources, dependency on high-quality training data, and potential challenges in detecting small or obscured fires.

### 2.7 Forest Fire Detection and Prediction from Image Processing Using RCNN (2022)

This study developed an RCNN-based system for detecting and predicting forest fires. The system uses image processing techniques to identify early signs of fire in forested areas, enabling timely detection and intervention to prevent larger-scale fires.

**Advantages:**

The system provides accurate detection of forest fires through advanced image processing, improving response times and potentially reducing the damage caused by fires.

**Limitations:**

The system's effectiveness depends on the quality and clarity of the images, and it may face challenges in detecting fires in areas with poor visibility or under adverse weather conditions.

### 2.8 Using Hybrid Algorithms of Human Detection Technique for Detecting Indoor Disaster Victims (2022)

This study developed a method for detecting disaster victims indoors using hybrid algorithms. By combining multiple detection techniques, the system aims to improve the identification of human presence in disaster-stricken indoor environments, ensuring quicker and more accurate rescue operations.

**Advantages:**

The hybrid algorithm enhances detection accuracy and robustness, improving the ability to locate disaster victims in complex indoor environments. It enables faster and more reliable responses during rescue operations.

**Limitations:**

The system's performance may be affected by environmental conditions such as smoke, debris, or poor visibility, and it may require extensive training data for optimal performance.

### 2.9 Forest Fire and Smoke Detection Using Deep Learning-Based Learning Without Forgetting (2023)

This study developed a system for detecting forest fires and smoke using deep learning techniques, specifically utilizing a learning method that avoids forgetting previously learned information. The system aims to improve detection accuracy and adaptability by continuously learning from new data while retaining the ability to recognize past patterns, making it well-suited for dynamic environments like forests.

**Advantages:**

The system enhances forest fire detection by using deep learning techniques that improve accuracy over time. Its learning without forgetting method allows it to adapt to new data, making it more effective in changing conditions.

**Limitations:**

The system may require large amounts of data for training and may face challenges in real-time processing in highly dynamic forest environments. Additionally, performance could be impacted by environmental factors such as smoke or varying visibility conditions.

**CHAPTER-3**

**RESEARCH GAPS OF EXISTING METHODS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Authors** | **Title** | **Limitations (Research Gaps)** |
| 1 | Muhammad Salihin Ahmad Azmil, M. S., Ya'acob, N., Tahar, K. N., & Sarnin, S. S. | Wireless fire detection monitoring system for fire and rescue application (2015) | Limited range of wireless communication, require frequent maintenance and updates |
| 2 | F. Jaradat and D. Valles | A Human Detection Approach for Burning Building Sites Using Deep Learning Techniques (2018) | Requires extensive training data ,Struggles in extreme conditions like dense smoke or high heat. |
| 3 | Seetharaman, R., Sreeja, R. R., Dakshin, S. Vidhul, Nivetha, N., Gowsigan, S., & Barath, M. | Analysis of a Real Time Fire Detection and Intimation System (2020) | May face false alarms under certain conditions, Depends on network connectivity. |
| 4 | Aathithya, S., Kavya, S., Malavika, J., Raveena, R., Durga, E. | Detection of Human Existence Using Thermal Imaging for Automated Fire Extinguisher (2020) | Limited to thermal detection accuracy, Struggles in environments with varying temperatures. |
| 5 | Zaher, A., Al-Faqsh, A., Abdulredha, H., Al-Qudaihi, H., & Toaube, M. | A Fire Prevention/Monitoring Smart System (2021) | Dependency on sensors, potential high cost, Potential high initial cost. |
| 6 | Luo, W. | Research on fire detection based on YOLOv5 (2021) | Requires high computational resources, May face challenges in real-time processing. |
| 7 | Jaeseung Baek, Taha J. Alhindi, Young-Seon Jeong, Myong K. Jeong, Seongho Seo, Jongseok Kang, We Shim, Yoseob Heo | Real-time fire detection system based on dynamic time warping of multichannel sensor networks (2021) | Complex sensor network integration, Vulnerable to environmental interferences. |
| 8 | Lee H-W, Lee K-O, Bae J-H, Kim S-Y, Park Y-Y | Using Hybrid Algorithms of Human Detection Technique for Detecting Indoor Disaster Victims (2022) | Complexity in algorithm implementation, Limited by environmental conditions like smoke. |
| 9 | Chopde, Abhay et al. | Forest Fire Detection and Prediction from image processing using RCNN (2022) | Dependency on large datasets for training, Relies on clear image quality. |
| 10 | Sathishkumar, V.E., Cho, J., Subramanian, M., et al. | Forest fire and smoke detection using deep learning-based learning without forgetting (2023) | High computational demand for training, Needs continuous retraining with new data. |

**Table 3.1 Research Gaps of Existing Methods**

**CHAPTER-4**

**PROPOSED MOTHODOLOGY**

**4.1 YOLOv8:**

The YOLOv8 (You Only Look Once, Version 8) architecture represents a cutting-edge advancement in object detection, known for its high speed and accuracy. It is designed to detect objects in real time, making it highly suitable for applications where quick responses are essential, such as fire and smoke detection, security, and surveillance. Building on the strengths of previous YOLO versions, YOLOv8 incorporates several enhancements to improve processing efficiency and detection precision. This methodology delves into the architecture and working flow of YOLOv8, explaining each component’s purpose and functionality in the overall detection process.

**4.1.1 YOLOv8 Architecture Overview**

YOLOv8 is a single-stage object detection model designed to process images in a single forward pass, ensuring quick predictions with minimal computational resources. Its architecture combines several elements introduced in previous YOLO iterations while incorporating new features for better accuracy and speed. The key components in the YOLOv8 architecture are the Backbone, Neck, Detection Head, and Loss Functions.

**4.1.2 Architecture Details**

Each of these components plays a unique role in processing images and extracting critical information for accurate object detection. Here’s a breakdown of each:

a) Backbone – CSPDarknet

The backbone of YOLOv8 is a customized version of CSPDarknet, which functions as a feature extractor. This backbone leverages Cross Stage Partial Networks (CSPNet) technology to enhance feature extraction efficiency while reducing the model's computational load. In CSPDarknet, the input image undergoes a series of convolutional, batch normalization, and activation layers, which progressively distill image features, capturing low-level details (like edges) in early layers and more complex, high-level features (such as shapes and textures) in later layers.

The CSPNet structure is vital because it divides feature maps into two parts, with only one part processed through residual blocks before merging the two. This split reduces redundant computations, making the model both efficient and effective in handling large feature maps. YOLOv8’s backbone is thus capable of processing input images with greater computational efficiency while maintaining strong feature representation.

b) Neck – PANet (Path Aggregation Network)

The neck of YOLOv8 is a Path Aggregation Network (PANet), designed to enhance feature fusion across multiple scales. The PANet structure integrates both top-down and bottom-up pathways, enabling features from different depths in the backbone to be combined. This design enriches the model’s ability to detect objects of varying sizes, ensuring that small details are not overlooked while capturing the larger context of the scene.

PANet helps strengthen both object localization and classification by aggregating fine-grained details with broader contextual information. This multiscale feature fusion is crucial for applications that require precise detection of small objects, like a distant human in a safety surveillance scenario or small flames and smoke in fire detection.

c) Detection Head – Anchor-Free YOLO Head

The detection head in YOLOv8 is responsible for generating bounding boxes, class probabilities, and confidence scores for each detected object. Unlike traditional YOLO heads, YOLOv8 uses an anchor-free design, which simplifies the model and reduces computational requirements. This anchor-free approach predicts the location, size, and classification of objects directly without relying on predefined anchor boxes, making YOLOv8 easier to configure and more versatile across different use cases.

The detection head operates at three different scales, allowing YOLOv8 to accurately capture objects of various sizes. This multiscale approach is crucial for scenarios where the objects range from small (e.g., flames or tiny smoke clouds) to large (e.g., a person). Each scale outputs a matrix that includes coordinates, class labels, and confidence scores for each detected object.

d) Loss Function

YOLOv8 employs a combination of loss functions to optimize model performance:

* CIoU Loss (Complete Intersection over Union): Enhances the alignment between predicted and ground truth bounding boxes by considering overlap, distance, and aspect ratio. This function helps improve localization accuracy.
* Classification Loss: Penalizes errors in the classification of detected objects, ensuring high accuracy in object identification.
* Objectness Loss: Focuses the model’s attention on regions that likely contain objects, reducing false positives and making predictions more reliable.

**4.1.3 Working Flow of YOLOv8**

The detection process in YOLOv8 can be broken down into several stages, each designed to process images efficiently and accurately. Here’s an in-depth look at how YOLOv8 detects objects in real time:

a) Input Image Preprocessing

The input image is resized to a fixed dimension, such as 640x640 pixels, ensuring that the model processes images consistently. YOLOv8 also applies data augmentation techniques like scaling, flipping, and color adjustments during training, which improves model robustness and helps it generalize better across varied real-world scenarios.

b) Feature Extraction (Backbone)

Once the image is processed, it passes through the backbone (CSPDarknet), where convolutional layers extract feature maps at different scales. These maps capture essential details and textures, from basic edges to more complex patterns, providing a rich representation of the image’s content.

c) Feature Fusion (Neck)

The extracted features are passed through the neck (PANet), which aggregates information across different scales. This fusion enables the model to detect both small and large objects in the same scene by combining fine details with high-level contextual information. The PANet structure ensures that features from earlier and later layers contribute equally to the detection output.

d) Object Detection (Head)

The detection head generates predictions by directly locating object centers and dimensions through an anchor-free approach. By avoiding anchor boxes, YOLOv8 reduces computational complexity, allowing faster processing. Predictions are made at three different scales, ensuring objects of varying sizes are accurately detected in the output.

e) Post-Processing

YOLOv8 applies non-maximum suppression (NMS) to eliminate redundant bounding boxes, retaining only the most confident predictions. NMS ensures that only the most accurate bounding boxes remain, which improves the clarity and reliability of detection results.

The YOLOv8 architecture and working flow make it an exceptional tool for real-time object detection. Its CSPDarknet backbone, PANet neck, and anchor-free detection head enable efficient and precise object localization and classification across various scales. With a combination of CIoU Loss, Classification Loss, and Objectness Loss, YOLOv8 optimizes both detection accuracy and speed. This makes it a valuable asset in safety-critical applications like fire, smoke, and human detection, where rapid and reliable responses can significantly enhance safety and security measures. The advancements in YOLOv8 exemplify the potential of deep learning in practical, real-world applications, especially those requiring continuous monitoring and real-time feedback.

**4.2 YOLOv9:**

YOLOv9 builds upon the advancements of previous YOLO versions, integrating additional architectural innovations and improvements to provide even faster, more accurate, and robust object detection. Designed for high-stakes real-time applications, YOLOv9 enhances feature extraction, multiscale detection, and computational efficiency. This methodology explains the architecture and working flow of YOLOv9, detailing how each component functions to deliver high-speed, precise detections in complex environments, such as fire, smoke, and human presence detection.

**4.2.1 YOLOv9 Architecture Overview**

YOLOv9 is a single-stage object detector optimized for speed and accuracy. It introduces modifications to the backbone, neck, and detection head, leveraging recent advancements in deep learning, including enhanced attention mechanisms, path aggregation, and adaptive image scaling. The primary components of YOLOv9’s architecture are the Backbone, Neck, Detection Head, and Loss Functions.

**4.2.2 Architecture Details:**

Each component in YOLOv9’s architecture contributes to efficient processing and accurate object detection. Here’s a breakdown of each:

a) Backbone – CSPDarknet++ with Enhanced Attention Mechanisms

YOLOv9 uses an improved version of CSPDarknet called CSPDarknet++. This backbone incorporates an advanced attention mechanism, which allows the model to focus on significant parts of the image, reducing interference from background noise and enhancing the model’s ability to differentiate between objects. The CSP (Cross Stage Partial) structure, central to CSPDarknet++, splits feature maps into two streams: one is processed through residual blocks, while the other bypasses these blocks and is combined afterward. This structure reduces redundant computations and enhances feature extraction, making the model both efficient and accurate.

The backbone also includes a Self-Attention Module that learns to highlight important areas in an image, making YOLOv9 especially effective in detecting objects in cluttered scenes, such as fire and smoke within complex backgrounds. Additionally, CSPDarknet++ uses a combination of convolutional and activation layers (typically Mish or Swish) to extract deep, high-level features essential for object classification.

b) Neck – Enhanced PANet with Adaptive Feature Pyramid Network (AFPN)

The neck in YOLOv9 is an enhanced PANet with an added Adaptive Feature Pyramid Network (AFPN) component. This network improves multiscale feature fusion, allowing YOLOv9 to handle objects at varying scales more effectively. The AFPN structure uses a top-down and bottom-up approach, combining features across multiple layers. AFPN adjusts feature maps dynamically to ensure better detection for objects of different sizes by weighing the importance of specific layers based on object scale.

The multiscale fusion provided by this neck enables the model to accurately detect both large and small objects, making it particularly suitable for detecting distant flames, small patches of smoke, or people in real-time safety applications. This flexibility enhances the model's adaptability, allowing it to perform well even when objects occupy only a small part of the image.

c) Detection Head – Dual-Path YOLO Head with Attention

YOLOv9’s detection head is more advanced than its predecessors, featuring a dual-path architecture with integrated attention mechanisms. This dual-path design allows the model to predict bounding boxes and classify objects through two parallel paths, improving detection speed and accuracy. The addition of Spatial Attention Modules ensures that the model focuses on relevant areas of the image, enhancing its ability to detect objects in challenging conditions, such as low-light environments or low-resolution video feeds.

This anchor-free detection head also uses three scales to predict object locations, sizes, and classes, covering small, medium, and large objects in a scene. The anchor-free design simplifies the model, allowing it to achieve faster processing speeds while maintaining high accuracy across different object sizes and types.

d) Loss Function

YOLOv9 optimizes object detection using a sophisticated combination of loss functions, which include:

CIoU Loss (Complete Intersection over Union): A robust metric that accounts for overlap, distance, and aspect ratio between predicted and ground truth boxes, optimizing localization.

Classification Loss: Penalizes classification errors, encouraging precise labeling of objects.

Objectness Loss: Guides the model to focus on regions containing objects, reducing false positives in object-free areas.

**4.2.3 Working Flow of YOLOv9:**

The detection process in YOLOv9 involves multiple stages to ensure high-speed and high-accuracy processing of input images. Here’s a detailed breakdown of each step in YOLOv9’s workflow:

a) Input Image Preprocessing

The input image is resized to a fixed size (e.g., 640x640 pixels) for uniformity, allowing YOLOv9 to process it consistently. YOLOv9 also employs data augmentation techniques during training, including random scaling, flipping, and color changes. This step improves the model’s ability to generalize across different scenarios, enhancing robustness against varying lighting conditions and complex backgrounds.

b) Feature Extraction (Backbone)

The preprocessed image passes through the backbone (CSPDarknet++), where convolutional layers extract a series of feature maps at different resolutions. The enhanced attention mechanisms in CSPDarknet++ allow the model to focus on important regions, improving detection accuracy. The resulting feature maps contain detailed information about both the shape and context of objects in the scene, which are critical for accurate classification and localization.

c) Feature Fusion (Neck)

The feature maps are processed through the neck (Enhanced PANet with AFPN), which aggregates information from different scales. This feature fusion process combines high-resolution, fine-grained details with broader contextual information, enabling YOLOv9 to detect objects ranging from small flames and smoke clouds to large, complex structures. The AFPN’s adaptive approach adjusts the importance of features based on the scale of the objects in the image, optimizing detection across various object sizes.

d) Object Detection (Dual-Path Head with Attention)

The detection head generates predictions using its dual-path architecture, which allows YOLOv9 to classify and locate objects simultaneously along two parallel paths. By using spatial attention modules, the head prioritizes essential image regions, ensuring that detections are accurate even in crowded or visually noisy environments. The anchor-free design of the detection head simplifies the model’s configuration, improving processing speed and making YOLOv9 capable of real-time detection.

Predictions are made at three scales, covering a range of object sizes. Each prediction includes bounding box coordinates, class probabilities, and confidence scores, ensuring accurate identification across a wide variety of objects and contexts.

e) Post-Processing

YOLOv9 applies non-maximum suppression (NMS) to filter out redundant bounding boxes, keeping only the most confident predictions. This post-processing step ensures that the final output is free from duplicate detections, providing cleaner and more reliable results.

YOLOv9 introduces several architectural enhancements over its predecessors, with a CSPDarknet++ backbone, an improved PANet with AFPN for multiscale feature fusion, and a dual-path detection head with spatial attention modules. These elements work together to deliver high-speed, accurate object detection suitable for applications in challenging environments, such as real-time detection of fire, smoke, and human presence in safety and surveillance contexts.

The anchor-free, dual-path design of YOLOv9’s detection head, combined with advanced attention mechanisms, enables the model to focus on significant areas within images, making it more resilient to background noise and complex scenes. The robust combination of CIoU Loss, Classification Loss, and Objectness Loss ensures accurate localization and classification. YOLOv9’s architecture and workflow provide a versatile, high-performance solution for safety-critical applications, advancing the state of deep learning in real-time object detection and contributing to improved public safety and security systems. This methodology highlights YOLOv9’s potential to be a valuable tool in applications where speed and accuracy are essential, setting a new standard for object detection in computer vision.

**CHAPTER 5**

**OBJECTIVES**

1. Develop a Real-Time Detection System: Build a robust detection system that can accurately identify humans, fire, and smoke in real-time using advanced deep learning models, specifically YOLOv8 and YOLOv9.

2. Data Preparation and Utilization: Utilize a dataset from Roboflow containing diverse images of humans, fire, and smoke to train and validate the detection models.

3. Model Implementation and Optimization: Implement YOLOv8 and YOLOv9 models in Python using Google Colab as the integrated development environment. Optimize the models for high accuracy, speed, and efficiency in detecting and classifying objects.

4. Performance Comparison of YOLOv8 and YOLOv9: Analyze and compare the performance of YOLOv8 and YOLOv9 in terms of accuracy, speed, robustness, and their ability to handle various scenarios.

5. Real-Time Processing Capability: Develop the ability to process images and video feeds from a laptop camera, ensuring the system can function with real-time inputs despite potential limitations in resolution and clarity.

6. Visual Feedback System: Implement a system that provides visual feedback when the presence of fire, smoke, or humans is detected, thereby enhancing safety measures and response time in emergency situations.

**CHAPTER-6**

**SYSTEM DESIGN & IMPLEMENTATION**

**6.1 Introduction of Input Design:**

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Welldesigned input forms and screens have following properties −

* It should serve specific purpose effectively such as storing, recording, and retrieving the information.
* It ensures proper completion with accuracy.
* It should be easy to fill and straightforward.
* It should focus on user’s attention, consistency, and simplicity.
* All these objectives are obtained using the knowledge of basic design principles regarding −
  + What are the inputs needed for the system?
  + How end users respond to different elements of forms and screens.

**Objectives for Input Design:**

The objectives of input design are −

* To design data entry and input procedures
* To reduce input volume
* To design source documents for data capture or devise other data capture methods
* To design input data records, data entry screens, user interface screens, etc.
* To use validation checks and develop effective input controls.

**6.2 Introduction Output Design:**

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

**Objectives of Output Design:**

The objectives of input design are:

* To develop output design that serves the intended purpose and eliminates the production of unwanted output.
* To develop the output design that meets the end user’s requirements.
* To deliver the appropriate quantity of output.
* To form the output in appropriate format and direct it to the right person.
* To make the output available on time for making good decisions.

**6.3 Use Case Diagram of System Design:**

This system is designed for users to interact with a Open CV model built using YOLOv8 or YOLOv6. The system collects and preprocesses data, builds the model, and allows users to register, log in, and provide input data. Based on the input, the system generates predictions and provides results, after which users can log out. It streamlines the entire workflow from data preparation to prediction delivery.

#### 

**Figure 6.1 Use Case Diagram of System Design**

#### 6.4 **Implementation of System**

**6.4.1 Data Collection**

* **Objective**: Collect a comprehensive dataset of images and videos containing instances of fire, smoke, and human presence for training detection models.
* **Details**: Gather diverse image and video data from sources like open datasets, surveillance footage, and simulated environments to ensure robustness across different settings, lighting conditions, and backgrounds.
* **Dataset Split**: Divide the data into training (70%), validation (15%), and testing (15%) subsets to facilitate effective model training, evaluation, and fine-tuning.

**6.4.2 Data Preprocessing**

* **Image Processing**: Clean and normalize image data by adjusting brightness, contrast, and color balance. Apply augmentation techniques like rotation, scaling, and flipping to increase dataset variety.
* **Video Processing**: Perform frame extraction, noise reduction, and segmentation on video data. Convert video frames into a consistent format and resolution suitable for model training.

**6.4.3 Model Training**

* **Detection Models**: Train YOLOv8 and YOLOv9 models using the processed image and video datasets to ensure accurate detection of fire, smoke, and human presence.
* **Data Augmentation**: Apply data augmentation techniques during training to improve model robustness against variations in lighting, angles, and object scales.

**6.4.4 Model Evaluation**

* **Performance Metrics**: Assess detection models using metrics such as Intersection over Union (IoU), precision, recall, and F1-score to evaluate model accuracy and effectiveness in identifying fire, smoke, and humans.
* **Validation**: Use validation datasets to fine-tune models and ensure they generalize well to unseen data, refining them to reduce false positives and negatives in critical scenarios.

**6.4.5 Model Saving**

* **Model Serialization**: Save trained models in suitable formats (e.g., .pt for PyTorch) for deployment. Store model checkpoints to allow easy reloading and adjustments based on future data or requirements.

#### **6.5 Implementation of User**

**6.5.1 Register**

* **Objective**: Users create an account to access the fire detection and alert system.
* **Details**: Registration includes providing credentials, contact information, and alert preferences to receive timely notifications in case of fire or smoke detection.

**6.5.2 Login**

* **Objective**: Registered users log in to access system features.
* **Details**: Manages user authentication and session control, ensuring secure access to personalized alert settings and detection data.

**6.5.3 Input/Upload Image Feed**

* **Objective**: Users upload video feeds or connect live camera feeds for real-time monitoring and detection.
* **Details**: Supports video or camera feed integration to monitor areas for fire, smoke, and human presence. Handles receiving and validating input data for processing by detection models.

**6.5.4 View Detection Results**

* **Objective**: Users view detection results for any identified risks.
* **Details**: Displays detected instances of fire, smoke, and human presence in real-time.

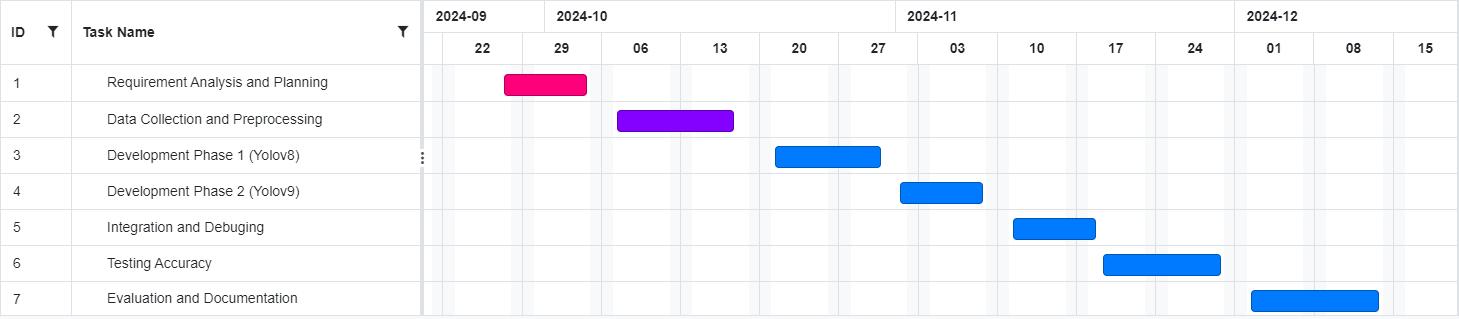
**6.5.5 Logout**

* **Objective**: Users log out to secure their session and personal data.
* **Details**: Manages session termination, ensuring that user data.

**CHAPTER-7**

**TIMELINE FOR EXECUTION OF PROJECT**

**(GANTT CHART)**

****

**CHAPTER-8**

**OUTCOMES**

1.Functional Real-Time Detection System: A fully functional application capable of

detecting humans, fire, and smoke in real-time from uploaded images and live video

feeds. The system will provide reliable detection and classification of these elements,

allowing for practical applications in safety monitoring.

2. High-Accuracy Detection Models: Successfully trained and validated YOLOv8 and

YOLOv9 models, optimized to accurately detect humans, fire, and smoke under various

conditions. The models will demonstrate high accuracy and minimal false positives or

negatives in the detection tasks.

3. Performance Evaluation and Comparison: A comprehensive comparison of YOLOv8

and YOLOv9 in terms of accuracy, speed, and robustness. This evaluation will highlight

the strengths and weaknesses of each model, offering insights into their suitability for

different real-time detection scenarios.

4. Real-Time Processing with Camera Feeds: Implementation of a system capable of

processing images from a laptop camera, enabling real-time detection despite

potential limitations in camera resolution and quality. This will demonstrate the

system's practicality in everyday safety applications.

5. Alert and Visual Feedback System: A responsive alert system that provides

notifications and visual cues when fire, smoke, or human presence is detected. This

feature aims to enhance response times in critical situations, contributing to improved

safety and situational awareness.

**CHAPTER-9**

**RESULTS AND DISCUSSIONS**

**9.1 Results for yolov8:**

**9.1.1 Precision-Recall Curve yolov8**

A graph of a graph

Description automatically generated

**Figure 9.1 Precision-Recall Curve yoloV8**

The Precision-Recall Curve shows the trade-off between precision and recall for each class ("fire," "human," "smoke") and provides an overview of the model's performance.

Fire: This class achieves a high precision-recall balance with an average precision (AP) score of 0.982. The curve is close to the top right, indicating excellent performance in detecting "fire" with high precision and recall across different thresholds.

Smoke: The model also performs well on the "smoke" class, with an AP of 0.849. While slightly less than "fire," the curve is still fairly close to the top right, showing good balance between precision and recall.

Human: The "human" class has a lower AP of 0.626, with a curve that drops more quickly as recall increases. This suggests the model struggles to maintain high precision as it tries to capture more instances, indicating room for improvement in detecting "human" accurately.

All Classes: The overall mean Average Precision (mAP@0.5) for all classes is 0.819, representing a good overall performance. However, there is a noticeable gap between "human" and the other classes, indicating a need for further tuning or more data to improve accuracy for "human" detection.

In summary, the model performs very well for "fire" and "smoke" but needs enhancement in detecting "human" to achieve better overall precision and recall.

**9.1.2 F1-Confidence Curve yolov8**

**A graph of a graph showing a curve

Description automatically generated with medium confidence**

**Figure 9.2 F1-Confidence Curve yolov8**

The F1-Confidence Curve illustrates the relationship between confidence thresholds and F1 scores for different classes ("fire," "human," "smoke") and for all classes combined:

Fire and Smoke: These classes maintain high F1 scores (close to 0.9) across a broad range of confidence thresholds, indicating strong model performance and consistency in detecting these classes.

Human: The F1 score for "human" is lower and peaks around 0.6. This curve declines more rapidly at higher confidence levels, suggesting that the model struggles with human detection compared to fire and smoke.

All Classes: The thick blue curve represents the F1 score across all classes, peaking at 0.82 with an optimal confidence threshold around 0.323. This indicates the best balance of precision and recall for the entire model at that threshold.

In summary, the model performs well for "fire" and "smoke" but less accurately for "human," which might need further model tuning or data adjustments for improvement.

**9.1.3 Training and Validation Metrics yolov8**

**A graph of a graph

Description automatically generated with medium confidence**

**Figure 9.3 Training and Validation Metrics yolov8**

The plots indicate the model’s performance across various training metrics over epochs:

Training Loss: The box, classification, and DFL (Distribution Focal Loss) losses all decrease steadily, showing effective learning and reduction in errors during training.

Validation Loss: Validation losses (box, classification, DFL) also decline, indicating that the model is generalizing well to unseen data.

Precision and Recall: Both precision and recall metrics steadily improve, reflecting the model's increasing accuracy in correctly identifying and classifying objects.mAP Metrics: The mAP@0.5 and mAP@0.5-0.95 (mean Average Precision) metrics rise consistently, demonstrating enhanced overall detection performance across different confidence thresholds and object scales.Overall, these metrics suggest that the model is successfully learning and improving with each epoch.

**9.1.4 Precision-Confidence Curve yolov8**

A graph of different colored lines

Description automatically generated

**Figure 9.4 Precision-Confidence Curve yolov8**

Fire and Smoke: These classes have high precision, especially at higher confidence thresholds. Precision for both classes approaches 1.0 as the confidence threshold increases, indicating that the model is very accurate in detecting "fire" and "smoke" at high confidence levels.

Human: Precision for the "human" class improves more gradually, starting low and only reaching higher values at high confidence thresholds. This suggests the model is less accurate in detecting "human" compared to "fire" and "smoke," especially at lower confidence levels.

All Classes: The combined precision for all classes (thick blue line) reaches 1.0 at a confidence level of 1.0, meaning that the model is very reliable in its predictions when it has high confidence.

Overall, the model is highly precise for "fire" and "smoke" across various thresholds but needs improvement in precision for the "human" class, especially at lower confidence levels.

**9.1.5 Confusion Matrix yolov8**

A screenshot of a graph

Description automatically generated

**Figure 9.5 Confusion Matrix yolov8**

This normalized confusion matrix shows the performance of a multi-class classifier for detecting "fire," "human," "smoke," and "background."

Fire: Detected with high accuracy (0.97), with minor misclassifications to "background" (0.24).

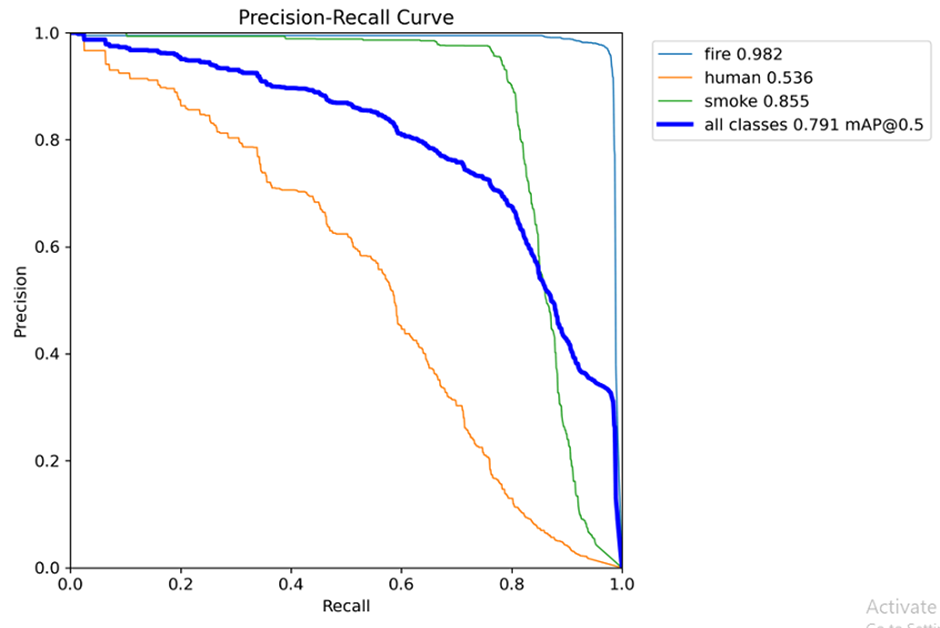
Human: Moderate accuracy (0.65), with notable misclassifications to "background" (0.61).

Smoke: Fairly high accuracy (0.79), but some confusion with "background" (0.15).

Background: Often misclassified, especially as "human" (0.34) and "smoke" (0.20).The model generally performs well on "fire" and "smoke" but struggles with separating "background" and "human," which may require further model refinement.

**9.2 Results for yolov9:**

**9.2.1 Precision-Recall Curve yolov9**

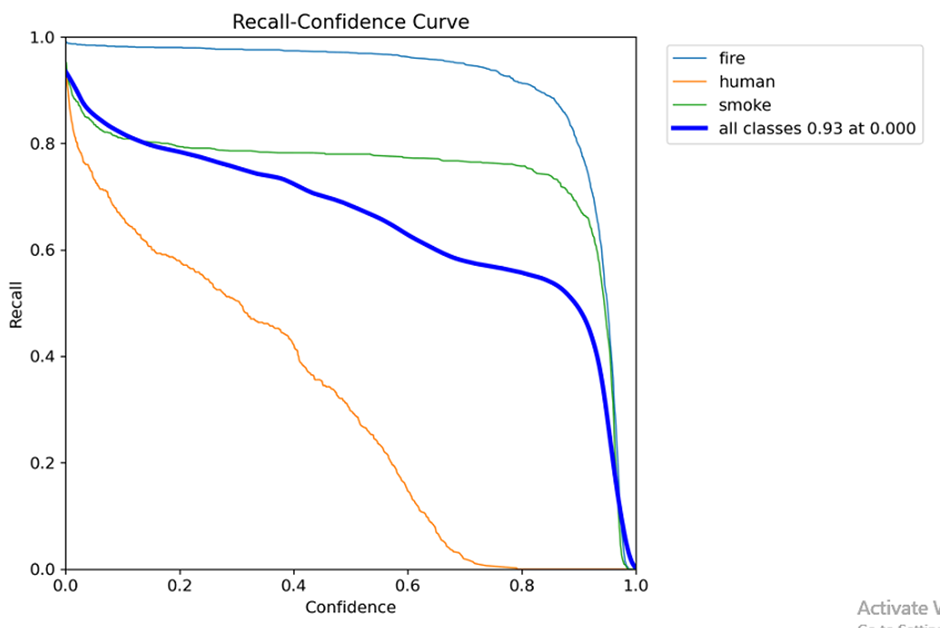


**Figure 9.6 Precision-Recall Curve yolov9**

**Precision-RecallCurve:**  
 The precision-recall curve provides a comprehensive view of the model's trade-offs between precision and recall for all classes.

* **Fire (0.982)**: This class has an almost perfect precision-recall balance, meaning the model is both accurate and exhaustive in identifying fire instances.
* **Human (0.536)**: The lowest-performing class, with precision and recall decreasing significantly as recall increases. This indicates the model struggles to capture all human instances, likely due to insufficient features or overlap with the background.
* **Smoke (0.855)**: Shows a strong balance between precision and recall, although there is room for improvement, particularly in reducing false negatives.
* **Mean Average Precision (**[**mAP@0.5**](mailto:mAP@0.5) **= 0.791)**: While overall performance is commendable, the relatively low mAP indicates that the model struggles to generalize well for all classes, particularly humans. Enhancing the dataset, using data augmentation, or fine-tuning the model could improve this score further.

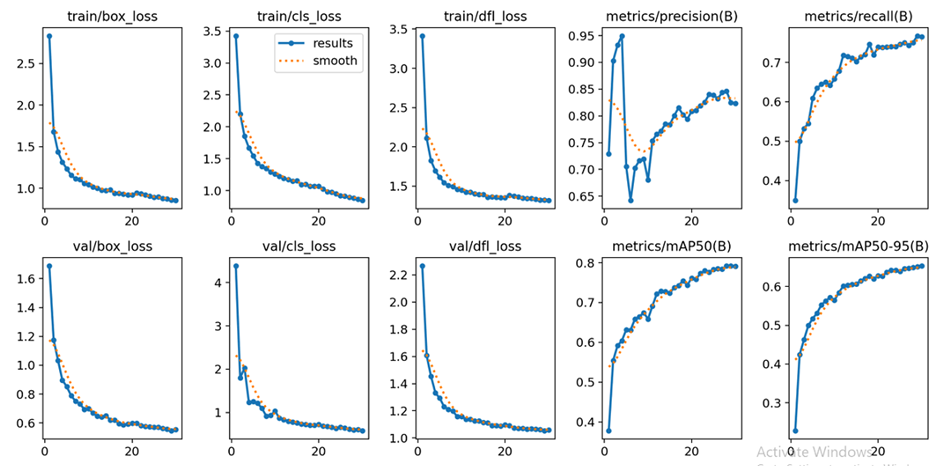
**9.2.2 Recall -Confidence Curve yolov9**



**Figure 9.7 Recall -Confidence Curve yolov9**

* **Fire**: High recall across all confidence levels, meaning fire detection is very reliable.
* **Smoke**: Moderate recall drop at higher confidence levels.
* **Human**: Steep drop, indicating the model struggles with human detection at higher confidence thresholds.

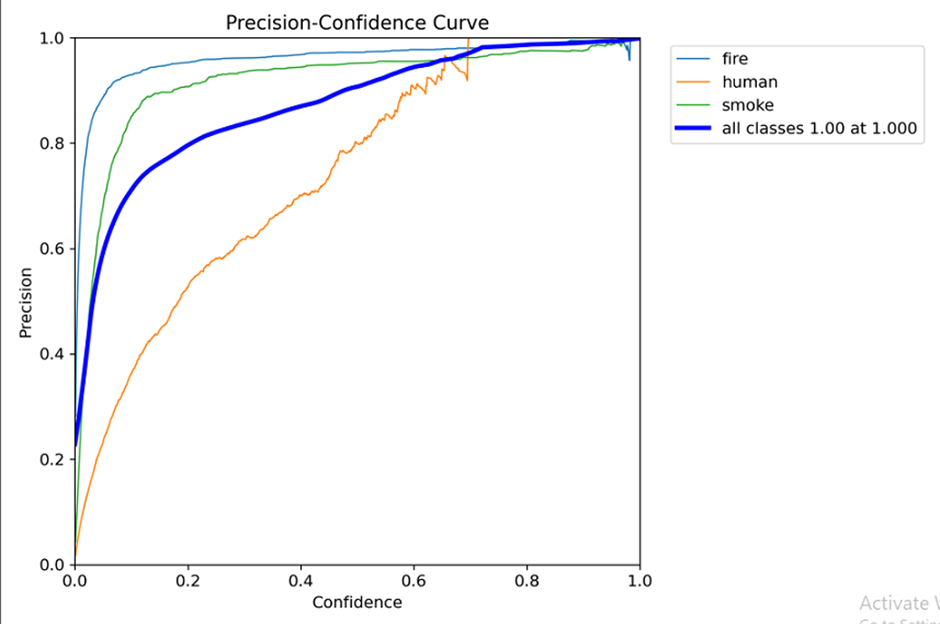
**9.2.3 Training and Validation Metrics yolov9**



**Figure 9.8 Training and Validation Metrics yolov9**

* **Training Losses**: The losses for bounding boxes, classification, and localization (top row) decrease steadily, indicating the model is learning effectively.
* **Validation Losses**: Similar trends for the validation set (bottom row) show that the model generalizes well.
* **Metrics**:
* **Precision & Recall**: Both improve during training, showing the model’s ability to make accurate predictions.
* **mAP** (Mean Average Precision): Gradual improvement reflects better overall detection performance.

**9.2.4 Precision-Confidence Curve yolov9**

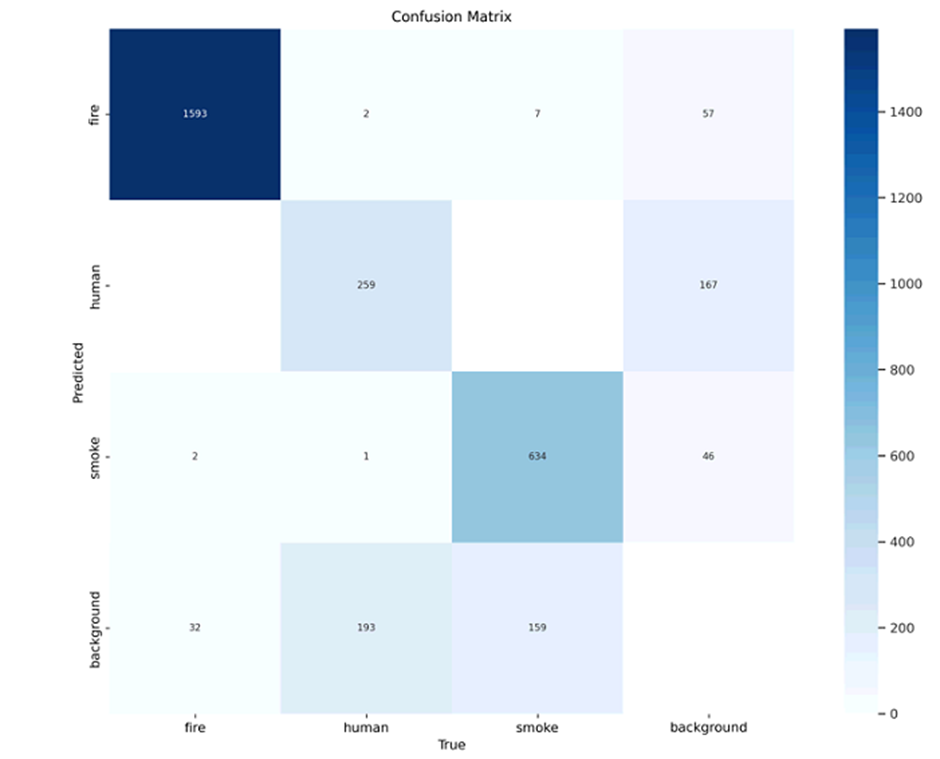


**Figure 9.9 Precision-Confidence Curve yolov9**

The precision-confidence curve visualizes the relationship between confidence scores and precision for each class:

* **Fire**: Maintains high precision across all confidence levels, showing that the model is highly reliable in detecting fire even at lower thresholds.
* **Human**: Precision drops significantly at lower confidence levels, meaning false positives are common for this class. This indicates either overlapping features with other classes or insufficient training samples for human detection.
* **Smoke**: Performs better than "human" but less consistently than "fire." Precision improves significantly with higher confidence, suggesting that strict thresholds can reduce false positives.
* **Overall Performance**: The combined "all classes" curve confirms that the model achieves high overall precision at high confidence levels but struggles to balance precision and recall for certain classes like humans.

**9.2.5 Confusion Matrix yolov9**



**Figure 9.10 Confusion Matrix yolov9**

The confusion matrix showcases the performance of the YOLOv9 model in distinguishing between the four classes: fire, human, smoke, and background.

* **Fire**: The model correctly classified 1593 instances as fire, with minimal misclassifications. Only 57 instances were misclassified as background, and 7 as smoke, indicating excellent accuracy for this class.
* **Human**: The model detected 259 humans correctly, but 167 instances were misclassified as smoke and 193 as background. This highlights the model's difficulty in distinguishing humans from other objects, particularly smoke.
* **Smoke**: With 634 correct predictions, the model performs well here, but some instances were classified as fire (2) or background (46).
* **Background**: Background detection is mixed; 159 instances were confused with smoke, and 32 with fire, which might indicate overlapping visual features. Improving feature extraction or class separation could address these issues.

**CHAPTER-10**

**CONCLUSION**

In conclusion, the training and evaluation metrics indicate that the model has been effectively trained to detect and classify objects, specifically "fire," "smoke," and "human," with increasing accuracy. The steady decline in training and validation losses suggests that the model has successfully minimized errors and learned the patterns in the data. The rising precision and recall metrics, along with improved mAP scores ([mAP@0.5](mailto:mAP@0.5) and [mAP@0.5-0.95](mailto:mAP@0.5-0.95)), demonstrate that the model is becoming more proficient at identifying objects across varying scales and confidence levels.

While the model performs well in detecting "fire" and "smoke," there is room for improvement in the "human" category, as reflected by slightly lower precision and recall scores for this class. Enhancing the dataset or refining model parameters specific to "human" detection could help boost performance in this area. Overall, the model is well-positioned for deployment in real-time safety applications, with promising accuracy and reliability for detecting critical objects like fire and smoke, aiding in prompt response and enhanced safety.

YOLOv9 demonstrates outstanding performance in detecting critical classes like "fire" (precision-recall 0.982) and "smoke" (0.855), with minimal misclassifications, making it highly suitable for real-time safety applications such as fire and smoke monitoring. However, it struggles with "human" detection (precision-recall 0.536), leading to misclassifications with other classes like background and smoke. Its overall [mAP@0.5](mailto:mAP@0.5) (0.791) is solid but slightly lower than YOLOv8, indicating room for improvement in generalizing across all classes.

YOLOv8, on the other hand, achieves higher mAP scores and delivers more balanced performance across all classes, including better detection of humans. The model shows consistent learning with steadily declining training and validation losses, making it versatile for diverse object detection tasks. While YOLOv8 is better for general-purpose applications, YOLOv9 excels in specific, high-precision scenarios like fire and smoke detection. The choice depends on whether the focus is on specific safety-critical tasks or broader object detection needs.

**CHAPTER-11**

**REFERENCES**

[1]. Aathithya, S., Kavya, S., Malavika, J., Raveena, R., & Durga, E. (2020) ‘Detection of Human Existence Using Thermal Imaging for Automated Fire Extinguisher’, Developed a thermal imaging-based system for human detection. Useful for automated fire extinguishing. Limited to thermal detection accuracy.

[2]. Chopde, A., et al. (2022) ‘Forest Fire Detection and Prediction from Image Processing Using RCNN’, Developed an RCNN-based system for forest fire detection. High accuracy in image-based detection. Dependency on large datasets for training.

[3]. Jaradat, F., & Valles, D. (2018) ‘A Human Detection Approach for Burning Building Sites Using Deep Learning Techniques’, Developed a human detection approach using deep learning. Effective in complex environments. Requires extensive training data.

[4]. Jaeseung Baek, Alhindi, T. J., Jeong, Y. S., Jeong, M. K., Seo, S., Kang, J., Shim, W., & Heo, Y. (2021) ‘Real-time fire detection system based on dynamic time warping of multichannel sensor networks’, Developed a real-time fire detection system. High efficiency with multichannel sensor networks. Complexity in sensor network integration.

[5]. Lee, H.-W., Lee, K.-O., Bae, J.-H., Kim, S.-Y., & Park, Y. Y. (2022) ‘Using Hybrid Algorithms of Human Detection Technique for Detecting Indoor Disaster Victims’, Developed a method for detecting disaster victims using hybrid algorithms. Improved detection accuracy in indoor settings. Complexity in algorithm implementation.

[6]. Luo, W. (2021) ‘Research on fire detection based on YOLOv5’, Implemented fire detection using YOLOv5. High detection accuracy. Requires high computational resources.

[7]. Muhammad Salihin Ahmad Azmil, M. S., Ya'acob, N., Tahar, K. N., & Sarnin, S. S. (2015) ‘Wireless fire detection monitoring system for fire and rescue application’, Developed a wireless monitoring system. Wireless, portable, suitable for rescue operations. Limited range of wireless communication.

[8]. Seetharaman, R., Sreeja, R. R., Dakshin, S., Vidhul, N., Nivetha, N., Gowsigan, S., & Barath, M. (2020) ‘Analysis of a Real Time Fire Detection and Intimation System’, Analyzed real-time fire detection and alert systems. Quick response time, efficient alerting. May face false alarms under certain conditions.

[9]. Sathishkumar, V. E., Cho, J., Subramanian, M., et al. (2023) ‘Forest fire and smoke detection using deep learning-based learning without forgetting’, Developed a system for detecting forest fires using deep learning. Adaptive learning capabilities. High computational demand for training.

[10]. Zaher, A., Al Faqsh, A., Abdulredha, H., Al-Qudaihi, H., & Toaube, M. (2021) ‘A Fire Prevention/Monitoring Smart System’, Developed a smart system for fire prevention and monitoring. Real-time monitoring and prevention. Dependency on sensors, high cost.

**APPENDIX-A**

**PSUEDOCODE**

**BACKEND:**

**YOLOV8:**

1. Install necessary libraries: ultralytics, roboflow, diagrams.
2. Initialize Roboflow API with API key.
3. Access Roboflow project and dataset.
4. Download the dataset for YOLOv8.
5. Display random sample images from test set.
6. Initialize YOLOv8 model.
7. Train the YOLOv8 model:
   * Specify training data (data.yaml).
   * Set hyperparameters (epochs, image size, batch size).
   * Define project and experiment name.
   * Enable caching.
8. Display training result plots:
   * Iterate over plot files.
   * Check if plot file exists.
   * Display plots using Matplotlib.
9. Load the trained YOLOv8 model.
10. Perform inference on a test image.
11. Visualize predictions:
    * Draw bounding boxes on the image with class labels and colors.
    * Display the image with predictions.

**YOLOV9:**

1. Install roboflow
2. Install ultralytics
3. Initialize Roboflow API
4. Get project from workspace
5. Get dataset version
6. Download dataset in YOLOv9 format
7. Define content of data.yaml
8. Write data.yaml to file
9. Run YOLOv9 training on the downloaded dataset with specified configurations
10. Display training results image
11. Display confusion matrix
12. Run YOLOv9 prediction on test images
13. Loop through the predicted images
14. Display each prediction image

**FRONTEND:**

# Load the necessary libraries and modules

LOAD required libraries (cv2, mysql.connector, YOLO, tkinter, PIL)

# Initialize YOLO model with a pre-trained weights file

INITIALIZE YOLO model with "best.pt"

DEFINE class names and their corresponding bounding box colors

# Initialize camera feed

INITIALIZE camera object for capturing video

# Create main GUI window using Tkinter

CREATE Tkinter root window

SET window properties (title, fullscreen mode, resize settings)

APPLY modern look and feel using styles

# Define MySQL database connection function

FUNCTION connect\_db():

CONNECT to MySQL database with specified parameters

RETURN database connection object

# Create sidebar and content area frames

CREATE sidebar frame for navigation

CREATE content frame for main content display

# Define global variables (e.g., stop flag for live feed)

DECLARE global variables for controlling program flow

# Define function to display live camera feed

FUNCTION show\_live\_feed():

CAPTURE a frame from the camera

PERFORM object detection using YOLO model

DRAW bounding boxes and labels on detected objects

DISPLAY the processed frame in Tkinter window

IF stop flag is set, stop the live feed

OTHERWISE, RECALL the function after a delay

# Define function to stop live feed

FUNCTION stop\_feed():

SET stop flag to true

RELEASE camera resources

CLEAR displayed images and reset UI

# Define function to start live feed

FUNCTION start\_live\_feed():

RESET stop flag

INITIALIZE camera if not already active

START displaying live feed

SHOW a stop button in the UI

# Define function to upload an image for object detection

FUNCTION upload\_image():

OPEN file dialog to select an image

PERFORM object detection on the uploaded image

DRAW bounding boxes and labels on detected objects

DISPLAY the processed image in Tkinter window

# Define function to display main buttons in the sidebar

FUNCTION display\_buttons():

CREATE and DISPLAY buttons for starting live feed, uploading image, and navigating home

# Define main page layout and navigation functions

FUNCTION index\_page():

DISPLAY a welcome message and main navigation buttons

FUNCTION register\_user():

DISPLAY user registration form

VALIDATE user input

INSERT new user data into database if valid

FUNCTION login\_user():

DISPLAY login form

VALIDATE user credentials

IF valid, NAVIGATE to user home page

FUNCTION user\_home\_page():

DISPLAY user-specific options (start detection, upload image, logout)

# Start with the index page

CALL index\_page()

# Start Tkinter event loop to keep the GUI running

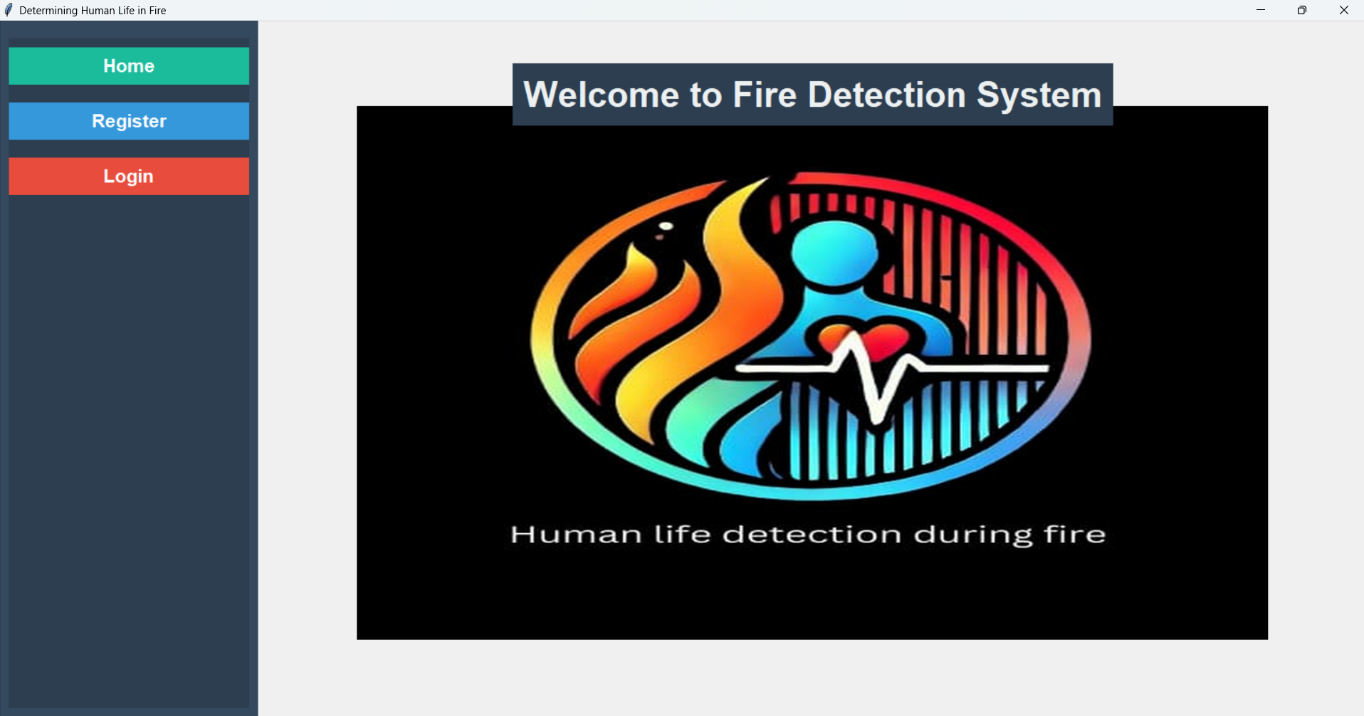
START Tkinter main loop

**APPENDIX-B**

**SCREENSHOTS**

**Home Page:**

The page allows you to either start a live video feed for human detection or upload an image for analysis. The goal is to identify human figures in real-time or static images.



A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

**Upload Page:**

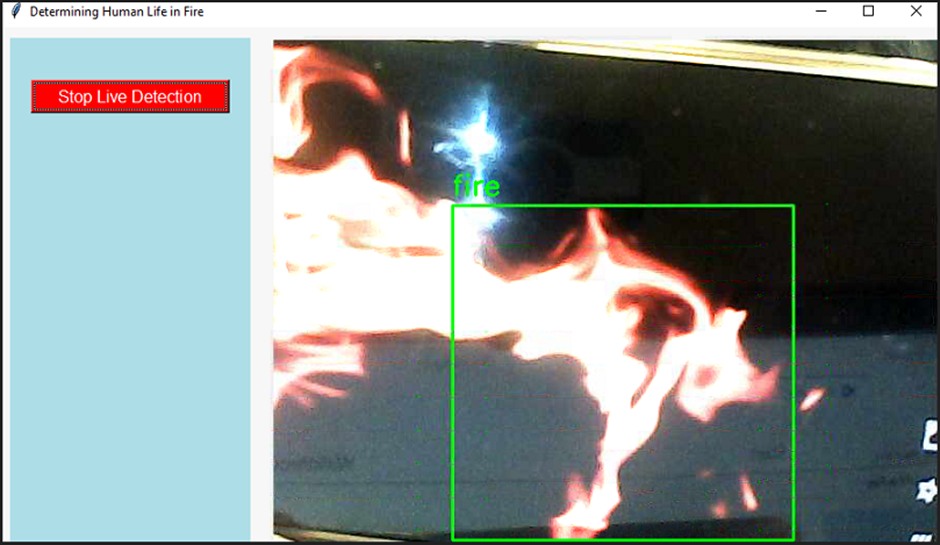
Upload an image of a fire scene. Our tool will analyze the image and detect any humans present within the fire, providing visual cues for potential rescue efforts.

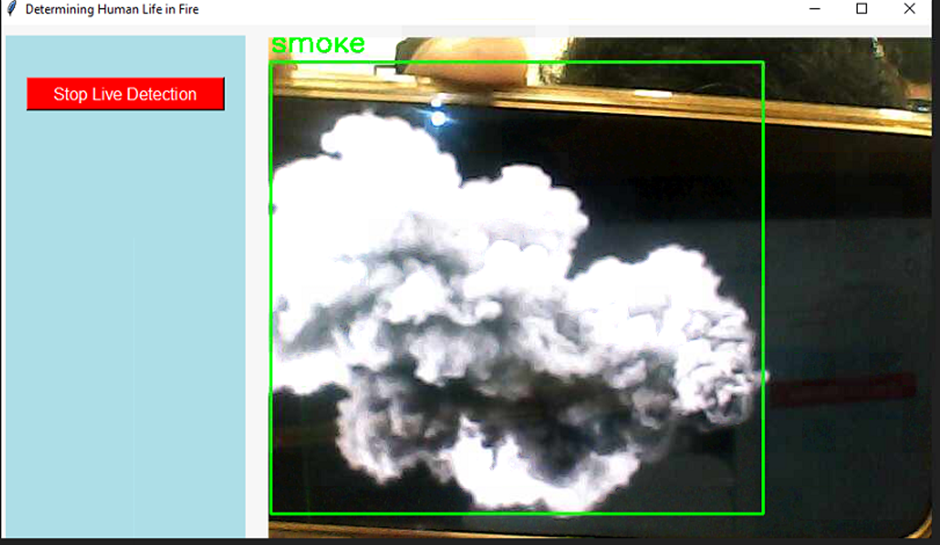
A firefighter putting out a fire

Description automatically generated

**Live Detection Page:**

The page is likely part of a fire detection or surveillance system. It displays a live video feed with a green rectangle highlighting a potential human figure within a fire and smoke.

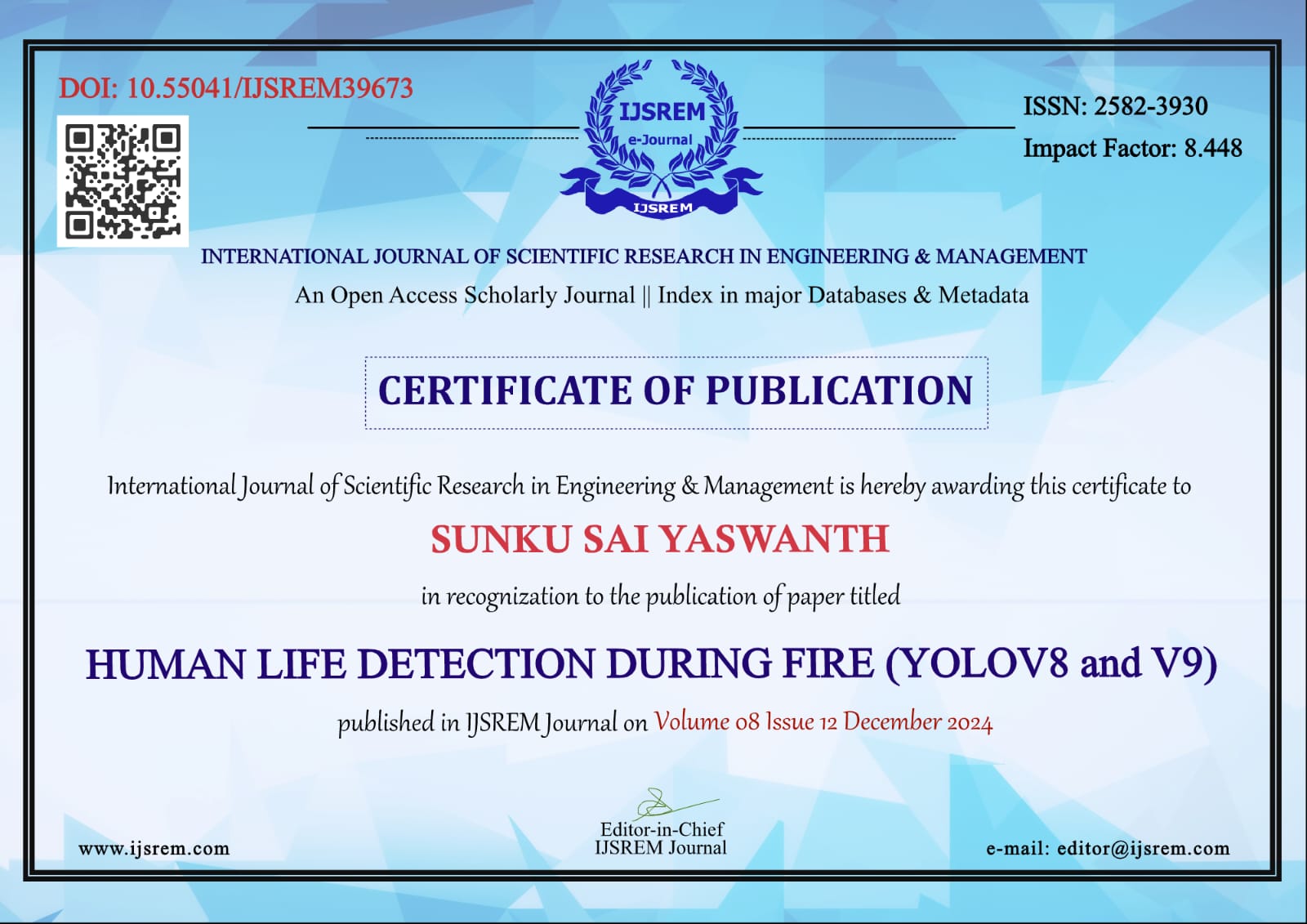




**APPENDIX-C**

**ENCLOSURES**

**Journal publication/Conference Paper Certificates**

****

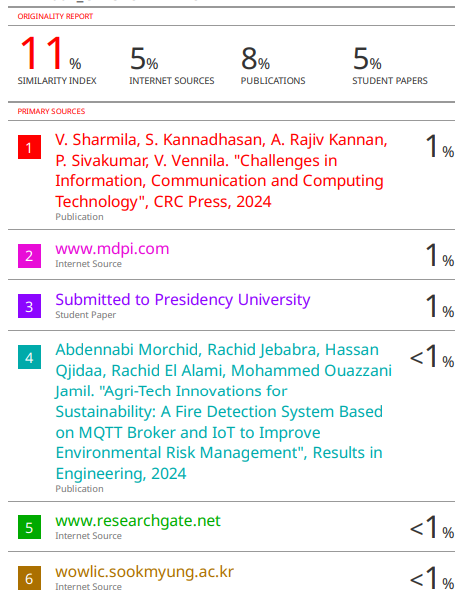
**A certificate of publication

Description automatically generated**

**A certificate of publication

Description automatically generated**

**Plagiarism Report**

****

**Sustainable Development Goals (SDGs).**

A chart of goals for a sustainable development

Description automatically generated

Detecting human life during a fire aligns most closely with SDG 9 (Industry, Innovation, and Infrastructure) and SDG 11 (Sustainable Cities and Communities).

**SDG 9: Industry, Innovation, and Infrastructure**

Leveraging innovative technologies to develop systems (e.g., thermal imaging, AI-based sensors, or drones) for detecting human life during emergencies promotes safer, smarter, and more resilient infrastructure.

Advances in technology can improve emergency response capabilities, enhancing the safety of industrial and urban environments.

**SDG 11: Sustainable Cities and Communities**

Technologies for detecting human life during fires contribute to making cities safer, more resilient, and better equipped to handle disasters.

They enhance urban safety systems, ensuring preparedness and protection for inhabitants in residential and commercial spaces.

Both SDGs emphasize the importance of safety, innovation, and sustainability in addressing challenges like fire emergencies.