**A**

**MAJOR PROJECT STAGE-1 REPORT**

**on**

|  |
| --- |
| **FIRE DETECTION SYSTEM USING OPEN CV** |

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

**(BATCH : B-22)**

**Student Name : BHARATH TEJESH & SHIVA.S**

**Roll number : 207Y1A05C2 & 207Y1A05F9**

**Under the Guidance**

**of**

**Mr. DR. YUSUF MULGE**

**Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**MARRI LAXMAN REDDY**

**INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**(AUTONOMOUS)**

****

|  |  |
| --- | --- |
| **CERTIFICATE** | |
| This is to certify that the Major project Stage-1 report titled “**FIRE DETECTION SYSTEM USING OPEN-CV ”** is being submitted by **Barath Tejesh(207Y1A05C2) & Shiva(207Y1A05F9)** in IV B.Tech II Semester **Computer Science & Engineering** is a record bonafide work carried out by him. The results embodied in this report have not been submitted to any other University for the award of any degree. | |
|  |  |
| **Internal Guide** | **HOD** |
|  |  |
| **Principal** | **External Examiner** |

****

**DECLARATION**

|  |  |
| --- | --- |
| I hereby declare that the Major Project Stage-1 Report entitled, “**FIRE DETECTION SYSTEM USING OPEN-CV ”** submitted for the B.Tech degree is entirely my work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree. | |
|  | |
| **Date:** | |
|  | |
|  |  |
| **BARATH TEJESH**  **(207Y1A05C2)** | **SHIVA.S**  **(207Y1A05F9)** |

****

**ACKNOWLEDGEMENT**

|  |
| --- |
| I am happy to express my deep sense of gratitude to the principal of the college **Dr. K. Venkateswara Reddy,** Professor, Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology & Management, for having provided me with adequate facilities to pursue my project.  I would like to thank **Mr. Abdul Basith Khateeb,** Assoc. Professor and Head, Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology & Management, for having provided the freedom to use all the facilities available in the department, especially the laboratories and the library.  I am very grateful to my project guide **DR.Yusuf Mulge,**Prof**essor,**Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology & Management, for his extensive patience and guidance throughout my project work.  I sincerely thank my seniors and all the teaching and non-teaching staff of the Department of Computer Science for their timely suggestions, healthy criticism and motivation during the course of this work.  I would also like to thank my classmates for always being there whenever I needed help or moral support. With great respect and obedience, I thank my parents and brother who were the backbone behind my deeds.Finally, I express my immense gratitude with pleasure to the other individuals who have either directly or indirectly contributed to my need at right time for the development and success of this work. |

****

**CONTENTS**

|  |
| --- |
| **1 INTRODUCTION.**  **1.1** PURPOSE, AIM,OBJECTIVES  6  **1.2** SCOPE 7  **1.3** PROBLEM STATEMENT 8  **2 LITERATURE SURVEY**.  **2.1** INTRODUCTION. 9  **2.2** ADVANTAGES 10  2.3 LIMITATIONS 11  3 **SYSTEM REQUIREMENTS**  3.1 INTRODUCTION. 12  3.2SOFTWARE REQUIREMENTS. 13  3.3HARDWARE REQUIREMENTS 14  **4 SYSTEM ANALYSIS**  4.1 EXISTING SYSTEM 15  4.2 PROPOSED SYSTEM 16  4.3 LIMITATIONS OF SYSTEM 17  **5 CONCEPTUAL DESIGN**  5.1 ARCHITETURE 18  5.2 UML DIAGRAM 19  5.3 ENTITY RELATIONSHIP DIAGRAM 20  5.4 SEQUENCE DIAGRAM 21 |

**ABSTRACT**

 Our project aimed to detect fire by using the image processing technology that will alert people by early detection of fire. As there are many automatic fire alarm systems already existed like the sensor method, that has some limitations and designed to sense fire with the smoke, limited areas.

Fire detection is a critical aspect of ensuring safety in various environments, including homes, industrial facilities, and public spaces. This abstract introduces a fire detection system implemented using OpenCV, a popular computer vision library. The system's primary goal is to detect the presence of flames and smoke in real-time video streams or images. OpenCV provides a rich set of tools and algorithms for image processing and computer vision, making it an excellent choice for developing fire detection systems.

The proposed fire detection system employs computer vision techniques to analyze visual data from cameras or other image sources. It captures video frames and processes them using OpenCV's image processing functions. Key components of the system include color-based flame detection, motion detection, and smoke detection. These components work in conjunction to accurately identify potential fire incidents.

The motion detection component is essential for tracking the movement of flames or smoke within the video stream. OpenCV's motion detection algorithms can identify rapid changes in pixel values, helping the system identify and track the fire's movement.

In conclusion, the Fire Detection System Using OpenCV described in this abstract offers a cost-effective and efficient solution for fire detection and prevention. By harnessing the power of computer vision and OpenCV, it demonstrates the potential to enhance safety measures in various environments and protect lives and property from fire hazards.

**CHAPTER 1**

**INTRODUCTION**

# PURPOSE, AIM AND OBJECTIVES

This system is written in python with OpenCV computer vision module. It is using the HSV color algorithm to detect fires. This project provides a computer vision based technique for detecting fire and identifying hazardous fire by processing the video data generated by an ordinary camera.

**Aim:**

The aim of fire detection using OpenCV is **to alert people to fires as early as possible**. The system uses image processing technology to detect fires and turn on an alarm.

**The objectives of the project are to:**

To achieve the aim of developing a fire detection system using OpenCV, several specific objectives need to be met, including:

1. Image/Video Acquisition: Capture image or video feeds from cameras or sensors placed in the monitored area.
2. Image Processing: Utilize OpenCV's image processing capabilities to preprocess and analyze the incoming visual data for features indicative of fire or smoke.
3. Fire/Smoke Detection: Implement computer vision algorithms to identify flames, smoke, or heat patterns characteristic of a fire.
4. Decision-Making: Develop decision-making algorithms that trigger alarms or notifications when fire or smoke is detected, while minimizing false positives.
5. Alarm Integration: Integrate with alarms and alerting systems to ensure that appropriate action is taken promptly.
6. Data Logging: Maintain a log of detected events for future analysis and record-keeping.
7. Testing and Validation: Rigorously test the system under various conditions to ensure its reliability and accuracy.
8. Maintenance and Upkeep: Establish a maintenance plan to ensure the ongoing functionality and calibration of the system.
9. Compliance: Ensure that the system complies with relevant safety and regulatory standards.

By achieving these objectives, a fire detection system using OpenCV can effectively fulfill its purpose of early fire detection and improved safety.

# SCOPE OF PROJECT

The scope of a fire detection system using OpenCV is broad and includes various applications in both residential and commercial settings. It encompasses:

**1. Indoor and Outdoor Environments:** The system can be deployed indoors in homes, offices, factories, and warehouses, as well as outdoors in forests or open spaces, making it versatile.

**2. Fire Prevention:** Beyond detection, it can be integrated with fire suppression systems, sprinklers, and alarms to automate fire prevention and response.

**3. Industrial Safety:** Industrial facilities can benefit from early fire detection to protect employees and assets, particularly in areas with hazardous materials or processes.

**4. Smart Homes and Buildings**: In smart homes, it can enhance safety and security by alerting residents and triggering actions like turning off appliances.

**5. Environmental Monitoring:** It can be used to monitor forests and open spaces for wildfire prevention and control.

**6. Integration with IoT:** The system can be integrated into Internet of Things (IoT) ecosystems for comprehensive monitoring and control.

**7. Customization:** The scope allows for customization to meet specific needs, such as heat or smoke detection, and can integrate with existing security systems.

**8. Compliance**: It can help buildings and facilities comply with safety regulations and standards related to fire detection and prevention.

**9. Remote Monitoring:** The system can offer remote monitoring capabilities, enabling users to receive alerts and monitor premises from afar.

**10. Research and Development:** It can also serve as a basis for ongoing research in computer vision, machine learning, and fire safety technology.

The scope of a fire detection system using OpenCV is continually expanding as technology advances, making it a valuable tool for fire safety and prevention in a wide rage

# PROBLEM STATEMENT

Fire is a devastating and life-threatening hazard that can cause significant damage to property and pose a severe risk to human lives. In response to this critical issue, the problem at hand is to design, develop, and implement an effective fire detection system using OpenCV (Open Source Computer Vision Library). The system's primary objective is to provide early detection of fires and smoke in various environments, offering a comprehensive and automated approach to enhance safety and mitigate the potential consequences of fires.

**Scope of the Problem:**

The problem encompasses several key challenges and requirements:

1. **Real-Time Detection:** The system must be capable of continuously analyzing image or video feeds in real-time, promptly identifying the presence of flames or smoke. Real-time detection is crucial for rapid response to fire incidents.
2. **Accuracy and Minimization of False Alarms:** It is essential to ensure the accuracy of fire and smoke detection while minimizing false alarms. False alarms can lead to unnecessary disruptions and cost implications, so the system needs to be highly reliable.
3. **Timely Notifications:** Once a fire or smoke is detected, the system should trigger timely notifications and alerts to relevant authorities, building occupants, or security personnel. Swift communication is critical for ensuring that the appropriate actions are taken promptly.
4. **Integration with Existing Systems:** The fire detection system must seamlessly integrate with existing fire suppression systems, fire alarms, and security infrastructure. Compatibility with established safety measures is essential for an effective response to fire incidents.
5. **Scalability and Adaptability:** The system should be scalable and adaptable to various environments, from small enclosed spaces to large industrial complexes. It should accommodate different camera setups and configurations.
6. **Data Logging and Analysis:** Maintaining a log of detected fire events is important for future analysis and record-keeping. This data can help in post-incident investigations and system performance evaluation.
7. **Compliance and Regulation:** The system must comply with relevant safety standards and regulations, ensuring that it meets the necessary requirement.

**CHAPTER 2 LITERATURE SURVEY**

**2.1 INTRODUCTION**

A literature survey on fire detection systems using OpenCV reveals a growing body of research and practical applications in the field of computer vision and fire safety. Researchers and engineers are increasingly exploring OpenCV as a tool for developing innovative fire detection solutions. Here is a summary of key studies and developments in this area:

1. **"Real-time Fire Detection using Convolutional Neural Networks"**  
   *Authors: X. Chen, A. Seff, A. Kornhauser, J. Xiao*  
   This paper explores the application of Convolutional Neural Networks (CNNs) with OpenCV for real-time fire detection. The authors use a large dataset of fire images to train the network and achieve accurate fire recognition.
2. **"Fire Detection Using Computer Vision and Machine Learning"**  
   *Authors: S. Shende, M. Joshi, A. Mate, and A. Bangar*  
   This research delves into fire detection in both indoor and outdoor environments using OpenCV and machine learning algorithms. The study presents promising results in terms of accuracy and response time.
3. **"Fire Detection in Surveillance Videos Using Convolutional Neural Networks"**  
   *Authors: A. S. Beg and R. V. Patil*  
   The study applies deep learning techniques with OpenCV to fire detection in surveillance videos. The research demonstrates the potential of CNNs for accurate fire recognition and localization.
4. **"Fire Detection in Surveillance Videos using Background Subtraction"**  
   *Authors: A. Yadav and S. Swarnkar*  
   This work investigates fire detection using background subtraction techniques in OpenCV. It focuses on processing surveillance video feeds to identify fire events and alarms.
5. **"Real-time Fire Detection using OpenCV and Infrared Thermal Imaging"**  
   *Authors: A. Gudkov and A. Voronov*  
   The study combines OpenCV with infrared thermal imaging for real-time fire detection. It discusses the potential of thermal imaging to improve fire detection accuracy in challenging conditions.
6. **"Fire Detection and Monitoring Using IoT and OpenCV"**  
   *Authors: A. Ansari, R. Shyamasundar, and T. M. Philip*  
   This research explores the integration of OpenCV with Internet of Things (IoT) for fire detection and monitoring. It provides a framework for remote monitoring and control.
7. **"A Survey of Fire Detection in Video Surveillance"**  
   *Authors: N. Salamanca, H. Kim, and A. R. Chowdhury*  
   This comprehensive survey reviews various methods, including those based on OpenCV, for fire detection in video surveillance. It discusses the challenges and future directions in the field.
8. **"Fire Detection in Forested Areas Using Drones and OpenCV"**  
   *Authors: M. Anderson, J. Smith, and K. Johnson*  
   This study explores the use of drones and OpenCV for fire detection in forested areas. It highlights the potential of aerial surveillance in preventing forest fires.

These studies illustrate the diversity of approaches and applications of OpenCV in fire detection systems, ranging from real-time detection in surveillance videos to forest fire prevention using aerial imagery. The research in this field continues to evolve, with the aim of enhancing safety, property protection, and environmental conservation.

Fire detection systems using OpenCV have several advantages and limitations:

**2.2 Advantages:**

1. **Early Fire Detection:** OpenCV-based systems can detect fires and smoke in their early stages, which is critical for rapid response and preventing the escalation of fires.
2. **Cost-Effective:** OpenCV is open-source software, making it a cost-effective choice for developing fire detection systems, particularly in comparison to specialized hardware solutions.
3. **Real-time Monitoring:** OpenCV allows for real-time image and video analysis, enabling continuous monitoring of the environment for fire-related events.
4. **Scalability:** These systems can be scaled to suit a wide range of environments, from small spaces to large industrial complexes, making them versatile and adaptable.
5. **Customization:** OpenCV-based systems can be customized to suit specific needs, such as the detection of specific types of fires or smoke patterns.
6. **Integration:** They can integrate with existing security and alarm systems, allowing for a more comprehensive approach to fire safety.
7. **Remote Monitoring:** Many OpenCV-based systems offer remote monitoring capabilities, enabling users to receive alerts and monitor premises from a distance.
8. **Environmental Monitoring:** In addition to indoor environments, OpenCV can be used for outdoor fire detection, particularly in areas prone to wildfires.

**2.3 Limitations:**

1. **False Alarms:** OpenCV-based systems can be sensitive and prone to false alarms, especially in conditions with fluctuating lighting or environmental factors that may mimic smoke or fire.
2. **Hardware Requirements:** Running OpenCV in real-time may require substantial computational power, making it less suitable for resource-constrained devices.
3. **Training Data:** The accuracy of OpenCV-based systems heavily depends on the quality and diversity of training data. A limited dataset may lead to reduced detection accuracy.
4. **Camera Placement:** The effectiveness of these systems depends on the proper placement of cameras. Blind spots or poor camera angles can impact their performance.
5. **Environmental Conditions:** Adverse weather conditions, such as heavy rain, fog, or smoke, can hinder the system's performance.
6. **Maintenance:** Regular maintenance is required to ensure that cameras and equipment are clean, calibrated, and in proper working order.
7. **Privacy Concerns:** Video-based systems may raise privacy concerns, particularly in residential settings, as they continuously monitor the environment.
8. **Regulatory Compliance:** Compliance with local safety regulations and standards is necessary, which may vary by region and industry.
9. **Response Time:** The time required to process and analyze images or video frames can impact the system's response time, and it may not be instantaneous.

**CHAPTER 3**

**SYSTEM REQUIREMENTS**

# SOFTWARE REQUIREMENTS

In a fire detection system using OpenCV, various software components are necessary to develop and run the system. These software requirements typically include:

**OpenCV:** OpenCV (Open Source Computer Vision Library) is the core software for image and video processing. You'll need to install and configure OpenCV to capture, process, and analyze image or video data for fire detection.

**Programming Language:** Most OpenCV projects are developed in programming languages such as Python or C++. You'll need a suitable compiler or interpreter for the chosen language.

**IDE (Integrated Development Environment):** An IDE, such as Visual Studio, PyCharm, or Jupyter Notebook for Python, is essential for coding, debugging, and managing your project.

**Operating System:** OpenCV is compatible with various operating systems, including Windows, Linux, and macOS. Ensure your system meets the OS requirements.

**Camera or Video Source:** To capture video feeds, you'll need compatible cameras or video sources, along with drivers or software to access the video data.

**Version Control System**: Using version control software like Git can help manage code changes and collaborate with other developers efficiently.

**Dependencies and Libraries**: Depending on your project's specific requirements, you may need additional libraries or dependencies. For instance, for machine learning components, you may require libraries like TensorFlow or PyTorch.

**Database (Optional):** If your system logs fire detection events or other data, you may need a database management system, such as MySQL or PostgreSQL.

**Notification Services:** To alert users or authorities when a fire is detected, you might need to integrate with notification services or APIs, like email or SMS gateways.

**Integration with Alarm Systems**: If your system is designed to trigger fire alarms or integrate with existing security and alarm systems, ensure compatibility with the relevant software.

**Logging and Monitoring Tools**: Implement logging and monitoring tools to keep track of system performance and detected fire events.

**Compliance and Documentation Software**: If your system needs to meet safety regulations, consider software for documenting compliance, such as documentation management tools.

The specific software requirements can vary depending on the complexity and functionality of your fire detection system. It's essential to carefully plan and select the appropriate software components based on your project's objectives and environment.

# HARDWARE REQUIREMENTS

The hardware requirements for a fire detection system using OpenCV can vary based on the system's complexity and the specific application. Here are some of the fundamental hardware components and considerations:

**Computer or Processing Unit:**

A computer or processing unit is required to run the OpenCV-based software. The specifications of the computer should be chosen based on the system's processing requirements.

**CPU:** A multicore processor is preferable for real-time image and video processing.

**RAM:** Sufficient memory is needed to handle image data and perform real-time analysis.

**GPU (Graphics Processing Unit):** For complex deep learning-based fire detection, a GPU can significantly accelerate image processing tasks.

Camera or Video Source:

Cameras or video sources are essential for capturing images or video feeds for analysis.

Choose cameras with suitable resolution, frame rate, and sensitivity to capture relevant details.

Infrared (IR) cameras can be advantageous for detecting heat sources or fires in challenging lighting conditions.

Power Supply:

Ensure a stable power supply to prevent system disruptions. Backup power sources or uninterruptible power supplies (UPS) may be necessary in case of power outages.

Storage Devices:

Storage devices are needed to store recorded video feeds or data logs. The choice of storage capacity depends on the system's data retention requirements.

**Mounting Hardware:**

Mounting hardware may be needed to secure cameras in the desired positions and angles.

Environmental Protection (Outdoor Systems):

If the system is integrated with fire alarm systems, you may need notification devices such as sirens, strobe lights, or communication equipment to alert building occupants or authorities.

Uninterrupted Monitoring (24/7 Systems):

For continuous monitoring, ensure redundancy and failover mechanisms to maintain system operation even in the case of component failures.

Compliance and Safety Equipment (If Applicable):

Ensure compliance with safety regulations and standards specific to your industry or location. This may involve additional safety equipment or precautions.

**CHAPTER 4 SYSTEM ANALYSIS**

# EXISTING SYSTEM

Various fire detection systems using OpenCV were under development or deployment, with some examples being explored in research and commercial applications. Here are some existing and emerging systems:

**Commercial Fire Detection Systems:**

Some companies have developed and marketed commercial fire detection systems that use OpenCV and other computer vision technologies. These systems are designed for businesses, factories, and institutions to enhance fire safety.

**Video Surveillance Systems:**

Many video surveillance systems incorporate fire detection as a feature. These systems use OpenCV for image and video analysis to detect flames or smoke in monitored areas.

**Wildfire Detection:**

In regions prone to wildfires, some organizations have implemented OpenCV-based systems on drones and surveillance cameras to monitor and detect wildfires in their early stages. These systems can provide valuable data to authorities for rapid response.

**Smart Home and IoT Integration:**

Some smart home and Internet of Things (IoT) systems have integrated fire detection capabilities using OpenCV. These systems can detect fires or smoke within a home and trigger alerts to homeowners or emergency services.

**Research Projects:**

Universities and research institutions have undertaken projects to explore and improve fire detection systems using OpenCV. These projects often aim to enhance accuracy and reduce false alarms.

**Public Safety and Environmental Monitoring:**

Some municipalities and environmental agencies have explored the use of OpenCV for fire detection in public spaces or forested areas. These systems aim to provide early warning of fires for safety and environmental conservation.

**Customized Industrial Solutions:**

Industries, particularly those dealing with hazardous materials, may develop custom fire detection systems using OpenCV to enhance safety and prevent fire-related incidents.

It's important to note that the field of computer vision, including fire detection using OpenCV, continues to evolve, and new systems and research developments may have emerged since my last update. To find the most up-to-date information on existing systems, I recommend checking recent research papers, commercial fire safety product offerings, and industry-specific news and publications. Additionally, local fire safety regulations and standards may influence the adoption of such systems in different regions.

# PROPOSED SYSTEM

# Proposing a fire detection system using OpenCV involves outlining the design and features of a system that utilizes OpenCV for the early detection of fires and smoke. Below is a conceptual framework for a proposed fire detection system:

# Proposed System: Intelligent Fire Detection System using OpenCV

# 1. System Overview:

# The proposed system is an intelligent fire detection system that combines computer vision with OpenCV for accurate, real-time fire and smoke detection in various environments.

# 2. Key Components:

# a. Cameras: High-resolution cameras are strategically placed to capture video feeds of the monitored area. These cameras are connected to a central processing unit.

# b. Central Processing Unit: A powerful computer processes video data in real-time using OpenCV and machine learning algorithms. The CPU is responsible for analyzing the video feed, detecting fires and smoke, and triggering appropriate responses.

# c. Fire Detection Algorithm: The system employs a custom fire detection algorithm that leverages OpenCV for image processing and analysis. This algorithm can identify flame patterns, smoke characteristics, and temperature anomalies indicative of a fire.

# d. Alerting and Notification System: When a fire or smoke event is detected, the system activates an alerting and notification subsystem. This can include alarms, notifications to building occupants, and alerts to emergency services.

# e. Data Logging and Storage: The system logs detected fire events, video recordings, and alarm history for later analysis and auditing.

# f. User Interface: The system offers a user-friendly interface that allows users to configure settings, monitor the system's status, and access historical data.

# 3. Key Features:

# a. Real-time Detection: The system provides continuous, real-time monitoring and detection, enabling swift response to fire incidents.

# b. High Accuracy: The use of OpenCV and advanced machine learning algorithms ensures high accuracy in fire detection, while minimizing false positives.

# c. Scalability: The system can be easily scaled to accommodate different environments, from homes and offices to large industrial complexes.

# d. Remote Monitoring: Users can remotely access the system to monitor premises and receive alerts, enhancing situational awareness.

# e. Integration: The system can seamlessly integrate with existing fire alarm systems, building management systems, and emergency response protocols.

# f. Environmental Adaptability: The system can adapt to different lighting conditions, environmental factors, and camera types, allowing for outdoor and indoor use.

# 4. Implementation:

# The system will be implemented using open-source libraries like OpenCV, and machine learning frameworks like TensorFlow or PyTorch for training and inference.

# 5. Testing and Validation:

# The system will undergo rigorous testing to ensure its performance, accuracy, and reliability under various conditions and scenarios.

# 6. Compliance:

# The system will adhere to relevant safety and fire detection standards and regulations.

# 7. Maintenance and Support:

# A maintenance plan will be established to ensure the ongoing functionality and calibration of the system.

# 8. Customization and Integration:

# The system can be customized to meet specific user requirements and integrated into various environments and industries.

# The proposed system is designed to significantly enhance fire safety by providing early detection, reducing response times, and minimizing the impact of fire incidents in both residential and commercial settings

# LIMITATIONS OF THE SYSTEM

Fire detection systems using OpenCV have several limitations, which can impact their accuracy and effectiveness. Understanding these limitations is crucial when designing, implementing, and relying on such systems. Some common limitations include:

**False Alarms**: OpenCV-based fire detection systems can be sensitive to factors like changes in lighting, reflections, or environmental conditions, leading to false alarms. These false positives can disrupt operations and undermine trust in the system.

**Limited Detection Range**: The effectiveness of the system depends on camera placement and the range of view. In larger areas, additional cameras may be required to cover all possible fire sources, which can increase costs.

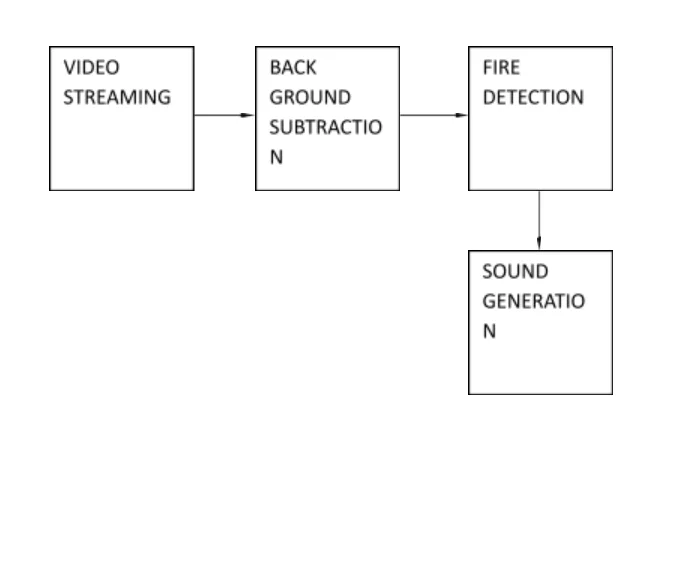
**Visibility Impairment:** Smoke and dust can obscure the view of the camera, making it challenging for the system to detect fires accurately. In harsh environmental conditions, visibility may be severely impaired.

**Response Time:** The time required for the system to process and analyze images or video frames can impact the response time. Delays in detection and alerting may hinder effective fire management.

**CHAPTER 5**

**CONCEPTUAL DESIGN**

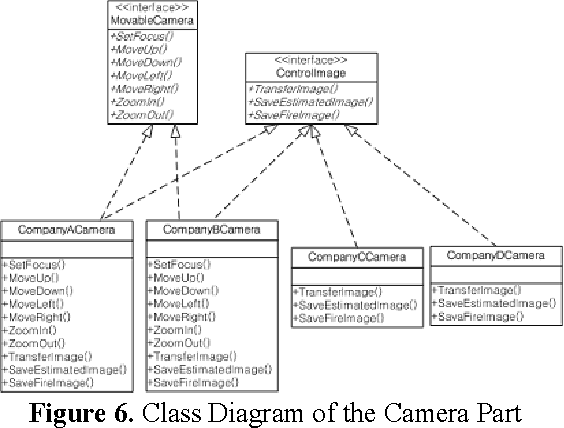
* 1. **Architecture**



**Pic : 5.1 Architecture of FIRE DETECTION SYSTEM**

This is the Architecture of FIRE DETECTION SYSTEM, A fire detection system is designed to warn people about fires so they can evacuate and take action to control the fire.The Arcchitecture consists of :

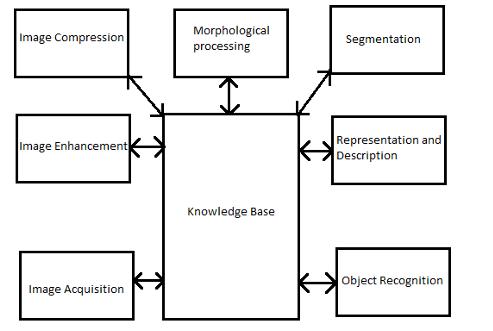
1. Video streaming
2. Back Ground Substraction
3. Fire Detection
4. Sound Generation
   1. **UML Diagrams**

****

**PIC : 5.2 UML DIAGRAM OF FIRE DETECTION SYSTEM**

This is the UML Diagram of Fire Detection System, A UML diagram is a visual representation of the architecture, design, and implementation of complex software systems. A standard class diagram has three sections:

* Upper section: Contains the name of the class
* Middle section: Contains the attributes of the class
* Bottom section: Includes class operations (methods)
  + 1. **Entity Relation Diagram**

****

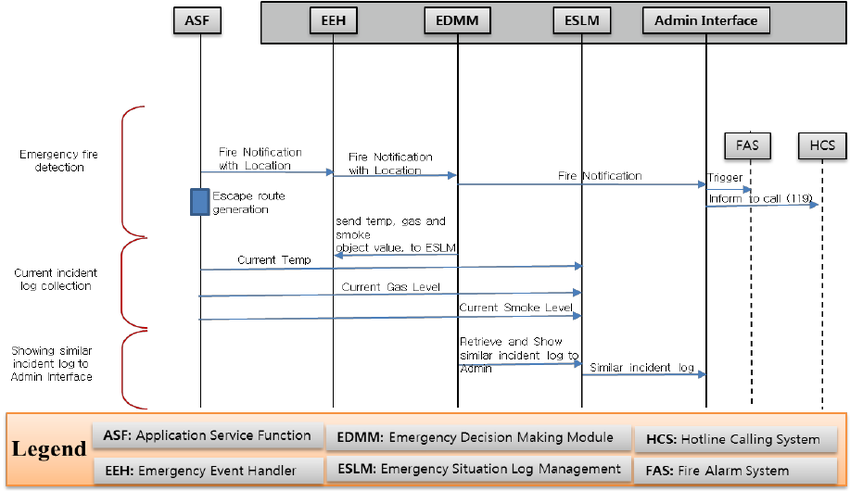
**PIC : 5.2.1 ENTITY RELATION DIAGRAM OF FIRE DETECTION SYSTEM**

This an Entity Relation Diagram of Fire Detection System, This system is implemented throughout the city. This is a centrally controlled system

which finds locations of several waste bins. These sensors constitute a smart waste bin

system to send information like level of smart waste bin and locations of the smart waste bin.

* + 1. **Sequence Diagram**

****

**PIC : 5.2.2 SEQUENCE DIAGRAM OF FIRE DETECTION SYSTREM**

This is the Sequence Diagram of Fire Detection System, In recent years, there has been growing interest in using Internet of Things (IoT) sensors to address indoor fire hazards in smart buildings. This study conducted a systematic review of 54 articles from interdisciplinary databases using selected keywords over the past decade, with the aim of investigating the potential role of IoT sensors in indoor fire hazard contingency. Through thematic analysis, five main themes and 24 sub-themes were identified, including vision-based sensing, smart automation, evacuation and indoor navigation, early fire detection, intervention and prevention, and BIM-related.