VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018, Karnataka

A PROJECT REPORT
ON

"ECO-HARBOR: AN INTEGRATED APPROACH FOR FOREST FIRE DETECTION, MITIGATION, WILDLIFE RELOCATION, AND TRIBAL ALERT USING DEEP LEARNING AND IOT"

A dissertation work submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE & ENGINEERING

Submitted By

AJAY M	**1EW20CS004**
BHAVANA N	**1EW20CS018**
CHAITHRA B C	**1EW20CS025**
CHANDAN B RAM	**1EW20CS026**

Under the Guidance of
Dr. Achyutha Prasad N
Prof. and Head,
Dept. of CSE, EWIT

EAST WEST INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

No. 63, Off Magadi Road, Vishwaneedam Post Bangalore–560091

2023-2024

East West Institute of Technology

(Affiliated to Visvesvaraya Technological University, Belagavi) Bengaluru-91

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VISION OF THE INSTITUTE

To be an Institute of Academic Excellence in Technical and Management Education on par with global standards to meet changing needs of the society.

MISSION OF THE INSTITUTE

M1:	To impart quality technical education that nurtures the young minds by providing the bestof teaching learning process and state of the art infrastructure.
M2:	To foster technological advancement through research.
M3:	To inculcate holistic personality development through best practices.
M4:	To implant ethical and social commitment that grooms the students to become responsible citizens.

VISION OF THE DEPARTMENT

To be a Department of Excellence in Technical Education through Innovative Learning approach with Social Commitment in the field of Computer Science & Engineering.

MISSION OF THE DEPARTMENT

M1:	To provide a conducive learning ambience to enhance problem solving skills, leadership qualities and team spirit.
M2:	To impart advanced technical knowledge through training programs for the holistic development of students.
M3:	To promote research, innovation and entrepreneur skills in the field of Computer Science and Engineering.
M4:	To inspire the students to become socially responsible technocrats.

Program Educational Objectives (PEOs)

After 4 years of the graduation the CSE graduates shall be able,

PEO1:	To achieve sustainable growth as Computer Science Engineers in reputed organizations.
PEO2:	To accomplish higher education and research with highest degree of professionalism andintegrity.
PEO3:	To extend the services to mankind by exhibiting leadership qualities and ethical values.

EAST WEST INSTITUTE OF TECHNOLOGY

No.63, Off Magadi Road, Vishwaneedam Post Bangalore-560 091

DEPARTMENT OF COMPUTER SCIENCE&ENGINEERING

1968
EAST WEST

CERTIFICATE

This is to certify that the Project work entitled **"ECO-HARBOR: AN INTEGRATED APPROACH FOR FOREST FIRE DETECTION, MITIGATION, WILDLIFE RELOCATION, AND TRIBAL ALERT USING DEEP LEARNING AND IOT"** is a bonafide work carried out **AJAY M (1EW20CS004), BHAVANA N (1EW20CS018) CHAITHRA B C** (**1EW20CS025**) and **CHANDAN B RAM (1EW20CS026)** in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the academic year **2023-2024.** It is certified that all the suggestions/corrections indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect to project work prescribed for the said degree.

---------------------------	----------------------------	-----------------------------
Signature of guide	**Signature of HOD**	**Signature of Principal**
Dr. Achyutha Prasad N	**Dr. Achyutha Prasad N**	**Dr. K Channakeshavalu**
Prof. & Head	**Prof. & Head**	**Principal/Director**
Dept. of CSE, EWIT	**Dept. of CSE, EWIT**	**EWIT**

NAME OF THE EXTERNAL EXAMINERS WITH SIGNATURE

1. ______________________ ______________________
2. ______________________ ______________________

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success.

We express our sincere gratitude to our principal **Dr. K Channakeshavalu**, **Principal,** East West Institute of Technology, Bengaluru, for having permitted to carry out the project work successfully**.**

We also thank our guide and are gratitude to project guide **Dr. Achyutha Prasad N, Professor and Head, Dept. of CSE**, East West Institute of Technology, Bengaluru, for his constant support and guidance.

We also thank our coordinators **Prof. Rajshekar S A and Prof. Pavithra B S**, **Assistant Professor, Department of Computer Science and Engineering,** for their support and guidance.

Finally, we would like to thank all the Teaching, Technical faculty and supporting staff members of Department of Computer Science and Engineering, East West Institute of Technology, Bengaluru, for their support.

AJAY M (1EW20CS004)
BHAVANA N (1EW20CS018)
CHAITHRA B C (1EW20CS025)
CHANDAN B RAM (1EW20CS026)

DECLARATION

AJAY M (1EW20CS004), BHAVANA N (1EW20CS018) CHAITHRA B C (1EW20CS025) and CHANDAN B RAM (1EW20CS026) students of **Eighth semester Bachelor of Engineering** in the Department of Computer Science and Engineering of East West Institute of Technology, Bengaluru-560091, here by declare that the Project entitled **"ECO-HARBOR: AN INTEGRATED APPROACH FOR FOREST FIRE DETECTION, MITIGATION, WILDLIFE RELOCATION, AND TRIBAL ALERT USING DEEP LEARNING AND IOT"** has been carried out by us under the guidance and supervision of Internal Guide **Dr. Achyutha Prasad N,** Professor and Head of the Department, Department of Computer Science and Engineering, East West Institute of Technology, Bengaluru submitted in the fulfilment of the course requirements for the award of the degree of Bachelor of Engineering of Computer Science and Engineering of Visvesvaraya Technological University during the Academic year **2023-24**.

Place: Bengaluru

Date:

SIGNATURE

AJAY M (1EW20CS004)

CHAITHRA B C (1EW20CS025)

BHAVANA N (1EW20CS018)

CHANDAN B RAM (1EW20CS026)

ABSTRACT

Forest fire is a serious threat to the environment and wildlife. It can also harm the tribal community as well as the people living in close vicinity to forest areas. Early detection and mitigation are crucial to avoid their destructive impact. By using the cameras used to track movement of wildlife we can capture real- time images and videos of the forest area. The data collected by these cameras is transmitted to a control center via wireless communication, where a deep learning algorithm is used for analysis. These algorithms can accurately identify smoke, flames, and fire within the image. Once the fire threat is detected, the system can trigger immediate responses, which includes alerting authorities and deploying firefighting measures. Preventive measures such as sprinkler systems can be used to mitigate the spread of fire. The real-time monitoring of forest allows early detection and rapid response to outbreaks. This approach not only protect forests and the ecosystems but also alert the wildlife and tribal community by high frequency sound waves for resettlement of animals and voice signals for tribal community.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In the face of escalating climate change and its associated risks, wildfires have become an increasingly prevalent and destructive force, threatening ecosystems, wildlife, and human communities globally. Addressing these challenges demands innovative solutions that blend technological advancements with compassionate and effective wildfire management strategies. The proposed Eco- Harbor Forest fire detection system represents a integrated approach designed to mitigate the devastating effects of wildfires while prioritizing the protection of natural habitats and vulnerable species.

By harnessing the power of high-resolution cameras and sophisticated deep learning algorithms, the system enables real-time detection of fire incidents within forested areas. This early detection capability is essential for initiating prompt responses aimed at containing wildfires before they can inflict significant ecological damage. Through the integration of the YOLO V8 deep learning algorithm, the system can accurately identify fire patterns, facilitating swift and targeted mitigation efforts.

Upon the detection of a forest fire, the Eco-Harbor system employs a multifaceted response mechanism to minimize the spread of flames and safeguard wildlife. Utilizing a combination of humane deterrents and automated suppression systems, the system ensures the protection of both flora and fauna within fire-affected areas. The deployment of high-frequency buzzers prompts animals to evacuate, while strategically positioned sprinklers release water to suppress the fire's progression until emergency responders arrive.

Furthermore, the system incorporates a communication module facilitated by a Node MCU microcontroller, enabling real-time notifications and updates to be disseminated to relevant stakeholders. This feature ensures swift coordination among forestry officials, emergency services, and conservationists, enhancing overall response effectiveness.

In summary, the Eco- Harbor Forest fire detection system represents a holistic and compassionate approach to wildfire management. By integrating advanced technology with wildlife-friendly response mechanisms, the system aims to minimize ecological damage and protect biodiversity in forested regions, ultimately contributing to the sustainable preservation of natural environments for future generations.

1.1 Internet of Things [IoT]

The Internet of Things (IoT) describes the network of physical objects "things" that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools. With more than 7 billion connected IoT devices today, experts are expecting this number to grow to 10 billion by 2020 and 22 billion by 2025.

Over the past few years, IoT has become one of the most important technologies of the 21st century. Now that we can connect everyday objects kitchen appliances, cars, thermostats, baby monitors to the internet via embedded devices, seamless communication is possible between people, processes, and things. By means of low-cost computing, the cloud, big data analytics, and mobile technologies, physical things can share and collect data with minimal human intervention. In this hyperconnected world, digital systems can record, monitor, and adjust each interaction between connected things. The physical world meets the digital world and they cooperate.

1.2 APPLICATIONS OF IoT

IoT Intelligent Applications are prebuilt software-as-a-service (SaaS) applications that can analyze and present captured IoT sensor data to business users via dashboards. We have a full set of IoT Intelligent Applications.

1.2.1 Wearables

Wearable technology is the hallmark of IoT applications and one of the earliest industries to deploy IoT. We have fit bits, heart rate monitors and smartwatches these days. Guardian glucose monitoring device has been developed to help people with diabetes. It detectsglucose levels in our body, uses a small electrode called the glucose sensor under the skin, andrelates it to a radiofrequency monitoring device.

1.2.2 Smart Home Applications

The smart home is probably the first thing when we talk about the IoT application. The example we see the AI home automation is employed by **Mark Zuckerberg.**

1.2.3 Health care

IoT applications can transform reactive medical-based systems into active wellness-based systems. Resources that are used in current medical research lack important real-world information. It uses controlled environments, leftover data, and volunteers for clinical trials.

Fig 1.2.3.1 IoT in Health care

1.2.4 Smart Cities

Most of you have heard about the term smart city. Smart city uses technology to provide services. The smart city includes improving transportation and social services, promoting stability and giving voice to their citizens. The problems faced by Mumbai are very different from Delhi. Even global issues, such as clean drinking water, declining air quality, and increasing urban density, occur in varying intensity cities. Therefore, they affect every city.

1.2.5 Agriculture

By the year **2050**, the world's growing population is estimated to have reached about 10 billion. To feed such a large population, agriculture needs to marry technology and get the best results. There are many possibilities in this area. One of them is Smart Greenhouse. Farming techniques grow crops by environmental parameters. However, manual handling results in production losses, energy losses and labor costs, making it less effective.

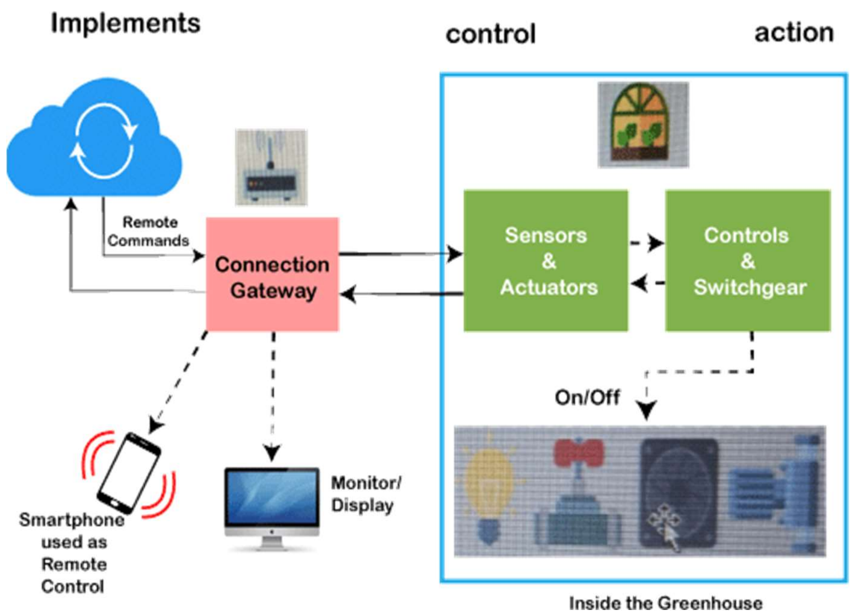

Fig 1.2.5.1 IoT in Agricultural field

1.2.6 Industrial Automation

It is one of the areas where the quality of products is an essential factor for a more significant investment return. Anyone can **re-engineer** products and their packaging to provide superior performance in **cost** and **customer experience** with IoT applications. IoT will prove as a game-changer. In industrial automation, IoT is used in the following areas:

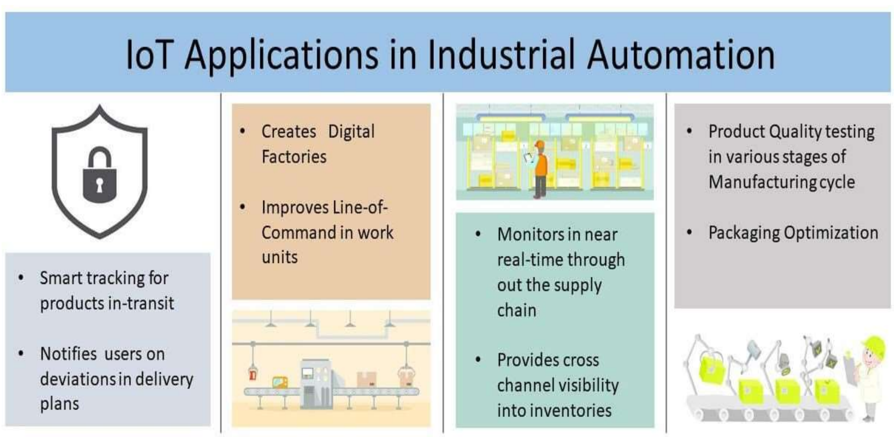

Fig 1.2.6.1 IoT applications in Industrial automation

1.3 ADVANTAGES OF IoT

1.3.1 Efficiency through automation
1.3.2 Cost reduction via optimization
1.3.3 Data-driven decision making
1.3.4 Enhanced customer experience
1.3.5 Improved safety and security
1.3.6 Enablement of new business models
1.3.7 Environmental sustainability
1.3.8 Remote monitoring and control
1.3.9 Predictive maintenance
1.3.10 Health and wellness monitoring

1.4 DISADVANTAGES OF IoT

1.4.1 Risk of hacking and data theft
1.4.2 Dependency on internet connectivity
1.4.3 Increased potential for system failures
1.4.4 Loss of control and reliance on technology
1.4.5 Potential decrease in physical activity and intelligence
1.4.6 Job displacement for unskilled workers
1.4.7 Complexity and difficulty in implementation

1.5 INTRODUCTION TO DEEP LEARNING

Deep learning is a subset of machine learning that involves the use of neural networks with multiple layers to learn complex patterns in large amounts of data. Unlike traditional machine learning approaches, deep learning algorithms can automatically learn representations of data from the data itself, without relying on handcrafted features. Deep learning has gained significant attention and popularity in recent years due to its remarkable success in various domains, including computer vision, natural language processing, speech recognition, and autonomous driving.

Deep learning models consist of multiple layers of interconnected nodes, known as neurons, organized into a hierarchical structure. Each layer extracts increasingly abstract features from the input data, with deeper layers capturing high-level representations. The process of training a deep learning model involves feeding it with labeled data and adjusting the parameters of the network to minimize the difference between the predicted output and the actual labels. This is typically done using optimization algorithms such as stochastic gradient descent.

Deep learning has revolutionized several fields, enabling breakthroughs in image recognition, language translation, medical diagnosis, and many others. Its ability to automatically learn complex patterns and representations from raw data has made it a powerful tool in tackling real-world problems.

1.6 TYPES OF DEEP LEARNING

Deep learning encompasses various architectures and techniques tailored to different types of tasks and data. Some of the prominent types of deep learning include:

1.6.1 Convolutional Neural Networks (CNNs): CNNs are primarily used for tasks involving images and video data. They consist of convolutional layers that apply filters to input images to extract features such as edges, textures, and shapes. CNNs have achieved remarkable success in image classification, object detection, and image segmentation tasks.

1.6.2 Recurrent Neural Networks (RNNs): RNNs are designed to handle sequential data, such as text, speech, and time-series data. They have connections between neurons that form directed cycles, allowing them to capture temporal dependencies in the data. RNNs are widely used in natural language processing tasks like language modeling, machine translation, and sentiment analysis.

1.6.3 Generative Adversarial Networks (GANs): GANs consist of two neural networks, a generator and a discriminator, that are trained simultaneously through adversarial learning. The generator learns to generate synthetic data samples that are indistinguishable from real data, while the discriminator learns to differentiate between real and fake samples. GANs have been used for generating realistic images, creating art, and data augmentation.

1.6.4 Deep Reinforcement Learning: Deep reinforcement learning combines deep learning with reinforcement learning, a type of machine learning focused on training agents to make sequential decisions in an environment to maximize cumulative rewards. Deep reinforcement learning has achieved significant advancements in playing games, robotics, and autonomous navigation.

1.7 APPLICATIONS OF DEEP LEARNING

1.7.1 Healthcare: Deep learning algorithms are transforming healthcare by enabling them to analyse medical images like X-rays, CT scans, and mammograms with high accuracy. This allows for earlier detection of abnormalities and diseases like cancer, leading to improved treatment outcomes. Deep learning is also being explored in drug discovery, where it can analyse vast amounts of molecular data to accelerate the development of new life-saving medications.

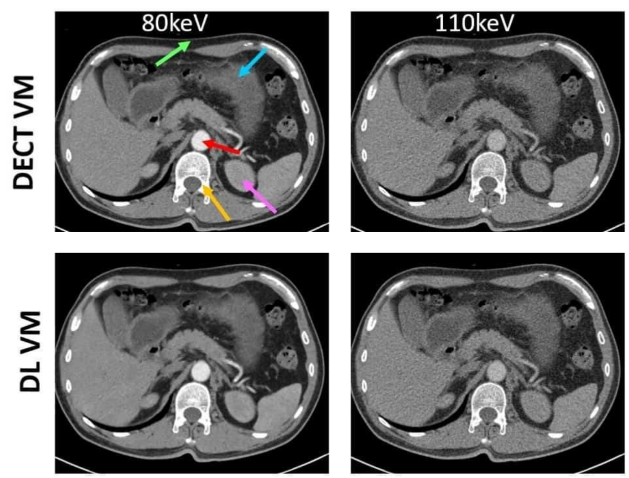

Fig 1.7.1.1 Deep Learning applications in healthcare

1.7.2 Transportation: Deep learning is essential for self-driving cars to navigate safely and autonomously. By analyzing camera images in real-time, deep learning algorithms can identify objects like pedestrians, vehicles, and traffic lights. This allows self-driving cars to make critical decisions, such as braking, steering, and following traffic rules, for safe navigation on the road.

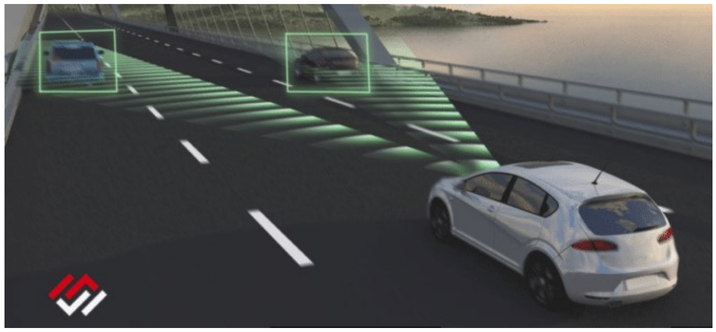

Fig 1.7.2.1 Deep Learning applications in Transportation

1.7.3 Security and Surveillance: Deep learning plays a crucial role in security and surveillance systems. Facial recognition systems powered by deep learning can identify individuals in images and videos, aiding in criminal investigations and access control. Additionally, deep learning can analyze security camera footage to detect anomalies or suspicious activities, potentially preventing crimes and improving public safety.

1.7.4 Retail: Deep learning is revolutionizing the retail industry by personalizing the shopping experience and increasing sales. Deep learning algorithms can analyse customer purchase history and browsing behavior to recommend relevant products. This targeted approach can lead to higher customer satisfaction and increased sales for businesses. Deep learning also powers self-checkout systems, where image recognition technology can automatically identify items, reducing checkout times and improving customer convenience.

1.7.5 Entertainment: Deep learning is making waves in the entertainment industry by enabling the creation of realistic images, videos, and even music. This technology is used for special effects in movies, generating video game content, and composing music pieces that mimic a certain style or artist. However, the use of deepfakes, which are realistic-looking manipulated videos, requires careful ethical considerations to avoid misuse.

1.7.6 Natural Language Processing (NLP): Deep learning is a driving force behind advancements in Natural Language Processing (NLP). Machine translation powered by deep learning is becoming more accurate and nuanced, breaking down communication barriers and fostering global collaboration. Deep learning is also being used to develop chatbots and virtual assistants that can understand natural language and respond in a helpful and informative way. These AI-powered tools are transforming customer service experiences by providing 24/7 assistance and answering customer queries efficiently.

1.7.7 Science: Deep learning is proving to be a valuable tool in scientific research. In climate modelling, deep learning algorithms can analyse complex climate data to improve climate models and predict future weather patterns with greater accuracy. This can help us prepare for the effects of climate change, such as extreme weather events and rising sea levels. Deep learning is also being applied in astronomy, where it is used to analyse astronomical images from telescopes. This allows scientists to identify distant galaxies, classify celestial objects, and unlock the secrets of the universe.

1.8 ADVANTAGES OF DEEP LEARNING

1.8.1 Feature Learning.

1.8.2 High Accuracy.

1.8.3 Versatility.

1.8.4 Wide Applications.

1.8.5 Scalability.

1.9 DISADVANTAGES OF DEEP LEARNING

1.9.1 Data Dependency.

1.9.2 Computational Complexity.

1.9.3 Interpretability.

1.9.4 Overfitting.

1.10 PROBLEM STATEMENT

The increasing frequency and intensity of forest fires pose a significant threat to both the environment and human communities. To address this critical issue, there is a need to develop an integrated system for forest fire detection and mitigation leveraging Internet of Things (IoT) and deep learning technologies.

Key Challenges:

- Early Detection: Developing a system that can detect forest fires at their early stages is crucial to minimizing their impact. The challenge is to create a reliable and timely detection mechanism that can differentiate between natural phenomena and actual fires.

- Real-time Monitoring: Forests cover vast areas, making it challenging to monitor every potential fire hotspot in real-time. The system needs to provide continuous monitoring and instant alerts over a large geographical area.

- Data Collection: Collecting data from remote forest locations and transmitting it to a central processing unit is a complex task. It requires efficient and robust data acquisition methods i.e., High resolution cameras with built in wireless communication tools.

- Deep Learning Models: Designing and training deep learning models capable of accurately identifying fires while minimizing false alarms is a significant technical challenge. The models should adapt to various environmental conditions and types of forests.

- IoT Integration: Integrating IoT devices into the forest ecosystem without disturbing the natural environment is crucial. The system should be capable of operating autonomously and be resistant to weather conditions and potential tampering.

- Preventive Measures: Alongside detection, the system should incorporate preventive measures, such as sprinkler systems, or automated firebreaks, to mitigate the spread of fires once detected.

1.11 OBJECTIVES

Forest fire detection and mitigation using IoT (Internet of Things) and deep learning involves the integration of technology to mitigate the devastating impact of forest fires. The objectives of such a system are multifaceted, with the primary goals being the early detection of fires and the implementation of preventive measures. Here are the key objectives:

1. **Early Detection of Forest Fires:**

   Early detection of forest fires is paramount for effective wildfire management and prevention. One strategy involves deploying cameras in forested regions to promptly identify fire onset. These cameras capture real-time data, which is then analyzed using deep learning algorithms. This enables timely responses from authorities and firefighting teams, effectively preventing fires from spreading and minimizing potential damage.

2. **Reducing Response Time:**

   Alerts must swiftly reach authorities and stakeholders upon fire detection. By minimizing the time, it takes for firefighters and emergency services to reach the fire's location, damage and potential loss of life can be significantly reduced.

3. **Monitoring and Surveillance:**

   Continuous monitoring helps detect fires at various stages, from small smoldering flames to large-scale blazes. Deep learning technology is employed to distinguish between controlled burns and uncontrolled wildfires, reducing false alarms and ensuring accurate response measures.

4. **Predictive Modeling:**

   The predictive models predict the likelihood of wildfires occurring based on various factors such as weather conditions, terrain characteristics, and other relevant parameters. By analyzing this data, authorities can anticipate where and when wildfires are more likely to occur, enabling proactive measures to mitigate risks and enhance preparedness.

5. **Preventive Measures:**

   To proactively mitigate fire risks, implementing IoT-controlled mechanisms like controlled burners, firebreaks, or automated sprinkler systems is essential. These technologies help contain fires and prevent their spread. Additionally, leveraging deep learning algorithms enables the identification of dry and fire prone areas can aid in establishing preventive measures deployment.

6. **Resettlement of Wildlife:**

   During forest fires, resettling wildlife can be facilitated through the use of high-frequency sounds that mimic natural warning signals or predator calls, triggering innate responses in animals. By employing carefully selected high-frequency sounds known to induce aversive reactions or stimulate movement in wildlife, it may be possible to guide animals away from danger zones, reducing their risk of harm during a forest fire.

7. **Alerting Tribal Community:**

   Alerting tribal communities to forest fires requires tailored communication systems. Establishing autonomous systems, such as strategically placed sirens, ensures prompt alerts. These systems must disseminate information in local languages, considering cultural nuances for effective communication within the community.

8. **Environmental Conservation:**

   Environmental conservation efforts aim to protect and preserve natural habitats by minimizing the ecological impact of wildfires. This can be achieved through early detection and containment measures that prevent fires from spreading and causing extensive damage to ecosystems. By swiftly identifying and controlling wildfires, the long-term health and biodiversity of the environment can be safeguarded.

9. **Data Analysis and Improvement:**

   Through systematically examination the data, insights can be gleaned to refine algorithms and improve the effectiveness of fire detection systems over time. This iterative process ensures that wildfire management strategies remain adaptive and responsive to evolving conditions, ultimately enhancing overall effectiveness in wildfire prevention and mitigation efforts.

CHAPTER 2
LITERATURE SURVEY

2.1 Title: Assessment of forest fire danger using automatic weather stations and MODIS TERRA satellite datasets for the state Madhya Pradesh, India.

Authors: Suresh Babu K.V, Venkata Sai Krishna Vanama, Arijit Roy, P. Ramachandra Prasad.

Abstract: Forest fires are the most frequently occurred phenomenon during summer seasons in the state Madhya Pradesh. Monitoring and assessment of forest fires are the crucial steps in effective forest fire management. Forest fire danger estimation helps the disaster management authorities to take necessary mitigation measures for minimizing the losses and to evacuate the local people. Fire danger rating systems predict the fire danger based on the meteorological station parameters and ground datasets. McArthur Forest Fire Danger Index (FFDI) is the most popularly used fire danger rating systems using in the country Australia. This index requires large amount of ground datasets for the computation of drought parameter. In India, it is very difficult to compute the drought parameter due to the unavailability of instruments and man power. In the present research, McArthur Fire Danger Index was modified by inducing Normalized multiband drought index (NMDI) that was generated from Moderate Resolution Imaging Spectroradiometer (MODIS) TERRA surface reflectance product MOD09GA as a substitute for fuel availability parameter.The results obtained from modified McArthur fire danger index were validated by using MODIS active fire hot spot location data (MOD14) and achieved an overall accuracy of 82%. The research concludes that modified FFDI can be used for assessing the forest fire danger in case of unavailability of fuel availability data for a particular forest.

Advantages: Utilizing MODIS TERRA satellite datasets provides a broader and more comprehensive perspective, allowing for a larger-scale assessment of forest fire danger. The modified FFDI offers a potentially faster and more efficient way to assess forest fire danger by relying on satellite data.

Disadvantages: The accuracy of the drought factor estimation (NMDI) is crucial. Any errors in this may impact the reliability of the forest fire danger assessment. Satellite data may have limitations in temporal resolution, which could affect real time fire danger assessment.

2.2 Title: Forest Fire Alerting System with GPS Co-ordinates Using IoT.

Authors: Jayaram K, Janani K, Jeyaguru R, Kumaresh R, Muralidharan N

Abstract: In the advancing world, it is very crucial to protect our environment. Many incidents of man-made and natural disasters were happening around the world. Forest fires are one such catastrophe for environment. Once the fire inside deep forest starts, it burns and destroys everything and spreads everywhere within the forest. Fire spreads on hot days destroys trees and grasses due to drought conditions peaks in a forest region. Such forest fires disasters should be curbed in order to protect fauna and flora habitats in the forest. The objective of this work is to design and implement an IoT based system which is self-sustaining and would predict and detect the forest fires and sends the exact location to concerned officials which would help firefighting personnel to extinguish the fire in the location where it starts slowly. This would prevent the fire to spread over a huge area and also able to take precautionary measures in order to prevent the fire which may occur in near future.

Advantages: Facilitates efficient allocation of firefighting resources by pinpointing the exact location of the fire Provides continuous monitoring of forest conditions, allowing for proactive measures and preventive actions Enables timely response to forest fires, reducing the potential for widespread damage.

Disadvantages: Remote locations might pose challenges for regular maintenance and troubleshooting. The IoT nature of the system could be vulnerable to cyber threats, necessitating robust security measures. The effectiveness of the system may be limited to the areas where the sensors are deployed.

2.3 Title: IoT Enabled Forest Fire Detection And Early Warning System.

Authors: A.Divya, T.Kavithanjali, P.Dharshini

Abstract: Everyone cognize that, the forest is praise as one of the most significant and compulsory expedient and Forest fire injunction a permanent danger to bionomical systems and environmental aspects. The forest fire detection had become very important issue in the pre suppression process which gives rise to the drastic need to perceive forest fires with greatest speed. The expert usage of wireless sensor network as a potentially explanation to the objective of forest fire has been emphatic in this literary. The proposed system depends on various sensors attached to it and the data from wireless transmission, to fulfil the solution process. A small satellite in the system dispatches these sensor data to the station on ground where they are scrutinized. The discourse plan impend on the data from Wireless sensor reticulation for the former discovery of Forest fire.

Advantages: Prior detection of forest fires and provides response time to emergency situations. One-time installation for sensors. Continuous monitoring of the workers' environment.

Disadvantages: Installation cost is high and complex fault detection in wired communication. Dependence on MQTT protocol without addressing potential drawbacks.

2.4 Title: An Improved Approach for Fire Detection using Deep Learning Models.

Authors: Mohit Dua, Mandeep Kumar, Gopal Singh Charan, Parre Sagar Ravi

Abstract: Fire detection using computer vision techniques and image processing has been a topic of interest among the researchers. Indeed, good accuracy of computer vision techniques can outperform traditional models of fire detection. However, with the current advancement of the technologies, such models of computer vision techniques are being replaced by deep learning models such as Convolutional Neural Networks (CNN). However, many of the existing research has only been assessed on balanced datasets, which can lead to the unsatisfied results and mislead real-world performance as fire is a rare and abnormal real-life event. Also, the result of traditional CNN shows that its performance is very low, when evaluated on imbalanced datasets. Therefore, this paper proposes use of transfer learning that is based on deep CNN approach to detect fire. It uses pre-trained deep CNN architecture namely VGG, and Mobile Net for development of fire detection system. These deep CNN models are tested on imbalanced datasets to imitate real world scenarios. The results of deep CNNs models show that these models increase accuracy significantly and it is observed that deep CNNs models are completely outperforming traditional Convolutional Neural Networks model. The accuracy of MobileNet is roughly the same as VGGNet, however, MobileNet is smaller in size and faster than VGG.

Advantages: Deep learning models can analyze visual data in real-time, enabling early detection of fires before they escalate, allowing for quicker response and mitigation. Deep learning algorithms can be trained on diverse datasets, enhancing their ability to accurately differentiate between actual fires and false alarms, reducing the risk of unnecessary emergency responses.

Disadvantages: Running complex deep learning models on edge devices, such as cameras or sensors, might consume significant energy. This could be a concern for devices with limited power sources. Implementing and maintaining a deep learning-based fire detection system may involve significant costs, including infrastructure, training, and ongoing updates to keep the model effective.

2.5 Title: Early Detection of Forest Fire using Deep Learning.

Authors: Medi Rahul, Karnekanti Shiva Saketh, Attili Sanjeet and Nenavath Srinivas Naik

Abstract: Forest fires have become a serious threat to mankind. Besides providing shelter and protection to a large number of living beings, they have been a major source of food, wood, and a great supply of other products. Since ancient times forests have played an important role in social, economic, and religious activities and have enriched human life in a variety of ways both material and psychological. To protect our nature from these rapidly rising forest fires, we need to be cautious enough of every decision we take which could lead to a disastrous end, once and for all. So for the early detection of forest fires, we propose an image recognition method based on Convolutional Neural Networks (CNN). We have fine-tuned the Resnet50 architecture and added a few convolutional layers with ReLu as the activation functions, and a binary classification output layer which showed a huge impact on the training and test results when compared to the other SOTA methods like VGG16 AND DenseNet121. We achieved a training set accuracy of 92.27% and 89.57% test accuracy with a stochastic gradient descent optimizer and we have avoided the underfit/overfitting on the model with the help of the Stochastic Gradient Descent (SGD) algorithm.

Advantages: Addresses the challenge of achieving high detection rates across multiple databases. Utilizes transfer learning and data augmentation to enhance the model performance. Provides a clear model architecture involving data preprocessing, augmentation, and classification. Demonstrates the reliability and performance of ResNet50 with fine-tuning.

Disadvantages: Transfer learning, model may not perform well if it encounters completely different types of forests, landscapes, or fire conditions which are not represented in the training data. Implementing the model in a real-time scenario using Unmanned Aerial Vehicles (UAVs) may pose logistical and regulatory challenges.

2.6 Title: IoT and Image Processing based Forest Monitoring and Counteracting System.

Authors: Shivam Pareek, Shreya Shrivastava, Sonal Jhala

Abstract: Forests are the indispensable resource of our life as they cover one third of the land on earth. They provide us with plenty of amenities required to sustain our life. However, for the past few decades, the forest area has been degrading immensely. Recently forest fire has become the greatest menace to our planet. In 2019 Amazon rainforest wildfire destroyed thousands hectors of forest. To get a control over it, a well-organized forest monitoring system is created. This system is based on the emerging technology of IoT and image processing. In our system, these technologies are utilized with the Wireless Sensor Network (WSN). This system provides a continuous live data of the forest environmental conditions. Utilizing this in our work helps us to detect fire intensity which enables water discharge for extinguishing fire when the conditions become unfavorable. This process will be helpful for controlling the wildfire.

Advantages: Integrates IoT and image processing for effective forest fire monitoring. Uses a variety of sensors for comprehensive data collection. MATLAB based image processing for fire detection. Counteraction system with a submersible pump to extinguish fires. Potential for future enhancements, such as distinguishing animals and protecting endangered species.

Disadvantages: Ultrasonic sensor limitations for precise trespasser detection. Potential false fire detection due to line-of-sight issues.

2.7 Title: Forest Fire Detection and Guiding Animals To A Safe Area By Using Sensor Networks And Sound.

Authors: Vignesh Tarun.M.G, Sankhasubhra Nandi S Sriram.M J Asbok.PS

Abstract: Forest fires are one of the main causes of environmental degradation. More than a million species of animals have lost their lives in the 2019-2020 wildfire that spread in the Amazon Forest. The model that we are proposing, intends to drastically reduce the number of lives lost in such unfortunate events and also alert the first response accelerating their momentum. Our idea is to have Wireless Sensor Networks (WSN) placed in a widely distributed manner across the forest area. Each module consists of a smoke sensor, temperature, humidity sensor, and a speaker which is connected to a Node-MCU. These modules collect data that is necessary for the prediction of wildfires. The data collected is analyzed along with the wind direction by our deep learning algorithm which predicts the wildfire spreading direction. This prediction is used to find a safe route for the animals to move away and get to a safe zone. Then the animals are manipulated to move away from the wildfire with the help of distressing sounds produced from the speaker triggering their flight response for their self-preservation. These distressing sounds are produced in a pattern rather than just producing it wherever wildfire is present. Hence leading wildlife to a safe zone and also nearby villages can be warned by a siren.

Advantages: Saves wildlife and reduces environmental damage. Real-time detection with WSN. Informs first responders and civilians. Guides animals to safety through distress signals. Economic feasibility for implementation.

Disadvantages: The effectiveness of distressing sounds is dependent on the coverage of sound waves across the forest area. While the system focuses on guiding wildlife away from the fire, it does not directly address the prevention of the fire itself.

2.8 Title: Forest-Fire Response System Using Deep-Learning-Based Approaches With CCTV Images and Weather Data.

Authors: Dai Quoc Tran, Minsoo Park, Yuntae Jeon, Jinyeong Bak, and Seunghee Park

Abstract: An effective forest-re response is critical for minimizing the losses caused by forest res. The purpose of this study is to construct a model for early re detection and damage area estimation for response systems based on deep learning. First, we implement neural architecture search-based object detection (DetNAS) for searching optimal backbone. Backbone networks play a crucial role in the application of deep learning-based models, as they have a significant impact on the performance of the model. A large scale re dataset with approximately 400,000 images is used to train and test object-detection models. Then, the searched light-weight backbone is compared with well-known backbones, such as ResNet, VoVNet, and FBNetV3. In addition, we propose damage area estimation method using Bayesian neural network (BNN), data pertaining to six years of historical forest re events are employed to estimate the damaged area. Subsequently, a weather API is used to match the recorded events. A BNN model is used as a regression model to estimate the damaged area. Additionally, the trained model is compared with other widely used regression models, such as decision trees and neural networks. The Faster R-CNN with a searched backbone achieves a mean average precision of 27.9 on 40,000 testing images, outperforming existing backbones. Compared with other regression models, the BNN estimates the damage area with less error and increased generalization. Thus, both proposed models demonstrate their robustness and suitability for implementation in real-world systems.

Advantages: Outperforms existing backbones, including hand-crafted and NAS-based models, with a DetNAS-based searching backbone. Achieves smoke and fire detection with an acceptable mAP of 27.9 Develops a web-based visualization platform for forest-fire response scenarios.

Disadvantages: Limited mAP score due to potential false positive cases in smoke detection. Overfitting issues observed in decision tree models. Unclear whether the BNN's higher accuracy can improve performance. CCTV cameras might not withstand harsh weather conditions of the forest.

2.9 Title: Video Based Forest Fire and Smoke Detection Using YoLo and CNN.

Authors: Sayali Madkar, Anagha P Haral, Dr. Dipti Y Sakhare, Kirti B Nikam, Komal A Phutane, S Tharunyha

Abstract: Forest fire is a matter of concern as it affects our ecological cycle and incurs a huge economical loss. There are various sensor-based systems built to detect fire at an early stage but the maintenance and covering the huge area is not feasible. Deep learning-based systems are also developed for fire detection however early detection, real time processing and monitoring of the region of interest is a major problem. To overcome these issues a remote sensing and deep learning-based forest fire detection system is proposed. The main objective is to monitor the forest and detect fire at an early stage. Remote sensing is used for versatile data collection which is used for training the model. Based model is used to detect fire with accuracy.

Advantages: Early detection of forest fires using a combination of remote sensing and deep learning. Data augmentation for increased dataset and model accuracy. Live monitoring of forest regions for real-time processing.

Disadvantages: Adapt to different environmental conditions such as lighting. This is important for the models to perform consistently in dynamic outdoor settings. Small dataset might limit the generalization capability of the models, especially considering the complexity of fire and smoke patterns.

2.10 Title: A Lightweight Hierarchical AI Model for UAV-Enabled Edge Computing with Forest-Fire Detection Use-Case.

Authors: Mostafa M. Fouda, Sadman Sakib, Zubair Md Fadlullah, Nidal Nasser, and Mohsen Guizani

Abstract: As the unmanned aerial vehicles (UAVs) continue to be deployed for various mission-critical data acquisition, localized computing on the drone-acquired data for efficient analysis, without significantly impacting the limited resources on board the drone, has emerged as a formidable research challenge. In this article, we address this issue with a natural resource management use-case whereby early forest-fire detection using the popular convolutional neural network (CNN)-based inference models are considered in the drone. This can lead to resource exhaustion. To alleviate this, we propose a lightweight hierarchical artificial intelligence (AI) framework, which adaptively switches between a simple machine learning-based model and an advanced deep learning-based CNN model. Then, we formulate a multi-objective optimization problem to model the trade-off between forest-fire detection accuracy and computational performance. We obtain the Pareto-optimal solution of the formulated problem by optimizing a new hyperparameter (i.e., the confidence score threshold) by employing the technique for order of preference by similarity to ideal solution (TOPSIS) for the whole model. Thus, we alleviate the computational burden while retaining a high level of detection accuracy. Finally, based on a real dataset, empirical results are reported to evaluate the performance of our proposal in terms of its lightweight features.

Advantages: Utilizes lightweight edge computing for UAVs, leveraging their mobility and flexibility. Addresses resource depletion in UAVs during early detection of forest fires. Demonstrates the viability of the proposed framework through extensive experiments on real datasets. Offers a multi-objective optimization approach for balancing accuracy and computational cost

Disadvantages: UAVs have inherent resource constraints, including limited computing power, storage capacity, and energy resources. UAVs often have limited payload capacities, restricting the types of sensors and equipment they can carry. Adverse weather conditions can affect UAV stability, navigation, and data acquisition, impacting overall system reliability

CHAPTER 3

PROPOSED SYSTEM

The proposed Eco-Harbor forest fire detection system embodies an innovative approach to safeguarding natural ecosystems and wildlife habitats from the devastating effects of wildfires. Leveraging advanced technology components, the system is designed to detect and respond to fire incidents in real-time, minimizing ecological damage and protecting vulnerable species. The heart of the system are strategically positioned cameras capable of capturing high-resolution video footage of forested areas. This footage serves as input data for the YOLO V8 deep learning algorithm, trained to recognize distinct fire patterns within the natural environment. Upon detection of a forest fire, the algorithm triggers a series of coordinated responses aimed at swift mitigation and wildlife protection.

Upon detection of a fire, the Arduino Uno activates a multifaceted approach to mitigate its spread and safeguard wildlife. Firstly, a high-frequency buzzer emits sound waves designed to irritate and deter animals, effectively prompting them to relocate away from the fire-affected area. This humane approach helps prevent harm to wildlife and encourages their safe evacuation from the danger zone. Simultaneously, the Arduino Uno initiates the activation of an automatic pump system connected to a network of sprinklers strategically positioned throughout the forest. This pump releases water, suppressing the fire and preventing its spread while emergency responders are en route. This dual-response mechanism ensures rapid and effective fire mitigation, safeguarding both natural habitats and nearby communities. Furthermore, the system incorporates a communication module utilizing a Node MCU microcontroller to interface with messaging platforms such as Telegram. This functionality enables real-time notifications and updates to be sent to relevant stakeholders, including forestry officials, emergency services, and conservationists.

In summary, the proposed Eco-Harbor forest fire detection system represents a holistic and compassionate approach to wildfire management. By integrating cutting-edge technology with wildlife-friendly response mechanisms, the system aims to minimize ecological damage and protect biodiversity in forested regions.

3.1 SYSTEM ARCHITECTURE

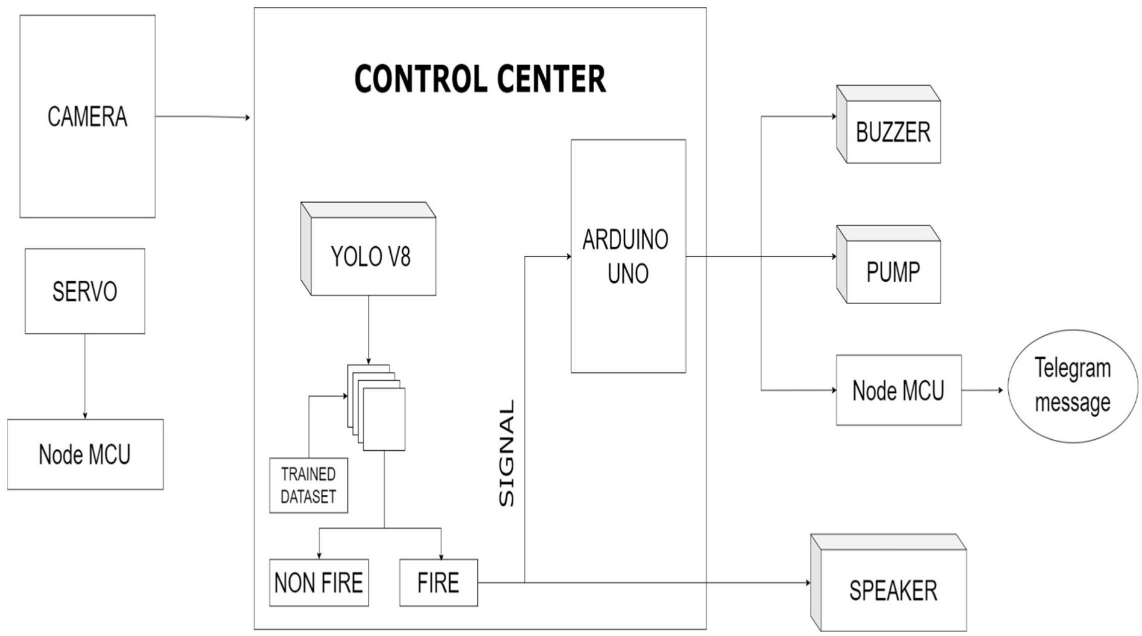

Fig 3.1.1 The system architecture of the proposed system.

This Eco-Harbor fire detection system leverages cameras and software for real-time monitoring. A camera captures live video, feeding it to YOLO V8, a program trained to identify fire in the footage. Upon detecting a fire, an Arduino Uno microcontroller activates a buzzer and speaker for immediate audible alerts and automatic pump is activated for mitigation of fire through sprinklers. The system also sends notifications via Telegram through a Node MCU microcontroller, keeping occupants informed even remotely. This setup utilizes YOLO V8 and microcontrollers for a comprehensive fire detection and notification solution.

3.2 YOLO V8

YOLO (You Only Look Once) v8 is a state-of-the-art object detection algorithm known for its efficiency and accuracy in real-time object detection tasks. It employs a single neural network to simultaneously predict bounding boxes and class probabilities for multiple objects in an image. YOLO v8 incorporates advancements in deep learning architectures and training techniques, resulting in improved detection performance. In forest fire detection, YOLO v8 can be utilized by processing images of forested areas in real-time.

3.2.1 Architecture of YOLO V8

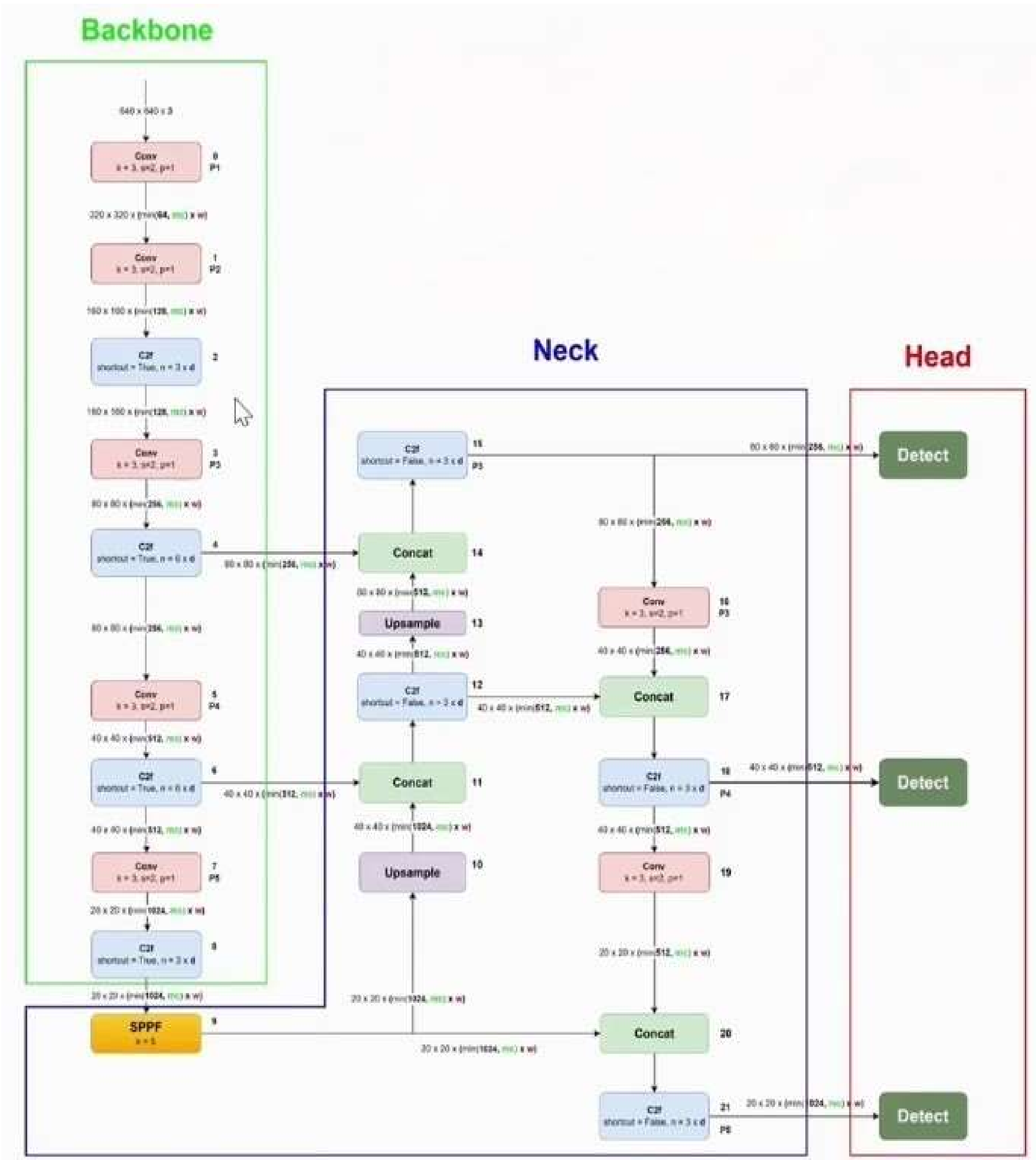

Fig:3.2.1.1 Architecture of YOLO-V8

By training the model on annotated datasets containing images of forests with and without fires, YOLO v8 can effectively identify and localize areas of fire outbreak. This enables timely intervention and response by authorities, potentially mitigating the spread of wildfires and minimizing their destructive impact on the environment and communities. In addition to detecting the presence of flames, YOLO v8 can also identify other crucial factors such as smoke, heat signatures, and changes in vegetation coloration that may indicate the early stages of a forest fire. By integrating YOLO v8 into a system, it offers a cost-effective and scalable solution for proactive wildfire management, enhancing overall preparedness and response capabilities in mitigating the devastating effects of forest fires.

3.2.2 Steps to Train and Implement a YOLO v8 model

The following are steps to train and implement a YOLO v8 model.

1. Data Collection:

To build an effective forest fire detection system using YOLOv8, start by gathering a diverse dataset of forest images. Ensure that the dataset contains instances of both forest fires and non-fire scenarios, covering various lighting conditions, times of day, and environmental settings. This diverse dataset will enable the model to generalize well to different conditions.

2. Data Preprocessing:

Preprocess the collected images to prepare them for training. Resize the images to a uniform size and normalize the pixel values to a common scale. Additionally, annotate the images to indicate the presence and location of fires using bounding boxes. This annotation step is crucial for training the YOLOv8 model to accurately detect forest fires.

3. Train-Validation-Test Split:

Split the annotated dataset into three subsets: training, validation, and testing sets. The training set will be used to train the YOLOv8 model, while the validation set will be used to fine-tune the model's hyperparameters and monitor its performance during training. The testing set will be used to evaluate the final performance of the trained model.

4. Build YOLO v8 Model:

Construct the YOLOv8 architecture by defining the layers of the network. YOLOv8 typically consists of a backbone network, such as Darknet or ResNet, followed by detection layers responsible for predicting bounding boxes and class probabilities. Design the architecture to effectively detect forest fires in the input images.

5. Train the YOLO v8 Model:

Train the YOLOv8 model using the annotated training dataset. Pass the images through the network, compute the loss function, and update the weights using backpropagation. Train the model until it achieves satisfactory performance on the validation set, ensuring that it can accurately detect forest fires while minimizing false positives.

6. Evaluate the YOLO v8 Model:

Evaluate the performance of the trained YOLOv8 model on the testing set. Pass the images through the network and calculate metrics such as precision, recall, and F1-score to assess the model's ability to detect forest fires accurately. Analyze the results to understand the model's strengths and weaknesses.

7. Deployment:

Once the YOLOv8 model has been trained and evaluated, deploy it in a real-world setting for forest fire detection. Integrate the model with surveillance camera systems or other monitoring devices to continuously monitor forested areas for potential fire outbreaks. Implement alert systems to notify relevant authorities and stakeholders in case of detected fires, enabling timely response and mitigation efforts. Regularly monitor and maintain the deployed system to ensure optimal performance and reliability.

3.2.3 Layers of YOLO v8:

1. Backbone Network: The backbone network is the initial component of the YOLOv8 architecture responsible for extracting features from the input images. It typically comprises several convolutional layers that progressively learn hierarchical representations of the input data. These layers play a crucial role in capturing intricate patterns and structures within the images, which are essential for accurate object detection. Common choices for the backbone network in YOLOv8 include architectures like Darknet and ResNet, which have proven effective in feature extraction tasks due to their depth and capacity to capture both low-level and high-level features.

2. Detection Layers: The detection layers in YOLOv8 are tasked with predicting bounding boxes and class probabilities for the objects present in the input images. These layers leverage anchor boxes, which are predefined shapes that serve as reference points for detecting objects of various sizes and aspect ratios. By utilizing anchor boxes and employing convolutional layers, YOLOv8 efficiently predicts the coordinates of bounding boxes along with the associated class probabilities in a single pass through the network. This enables real-time object detection while maintaining high accuracy, making YOLOv8 suitable for applications where speed and precision are paramount.

3. Non-Max Suppression (NMS): Following the detection stage, a non-maximum suppression algorithm is applied to refine the bounding box predictions and filter out redundant detections. NMS ensures that only the most confident and relevant bounding boxes are retained while suppressing overlapping or redundant ones. This process helps streamline the output of the detection algorithm, removing duplicate detections of the same object and producing a more concise and accurate set of predictions. By eliminating redundant bounding boxes, NMS improves the efficiency and interpretability of the object detection results generated by YOLOv8, facilitating downstream tasks such as object tracking and analysis.

CHAPTER 4

SYSTEM DESIGN

4.1 USE CASE DIAGRAM

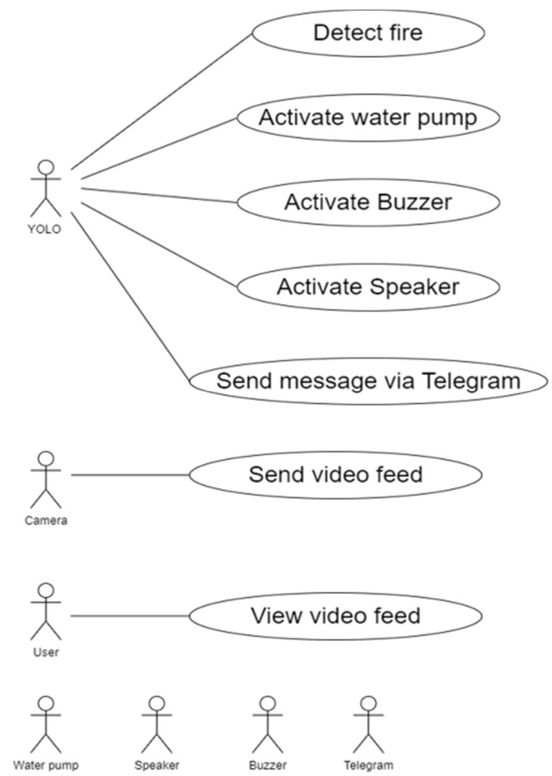

Fig.4.1.1 Use case diagram

This use case diagram showcases the Eco-Harbor fire detection system employing a camera for real-time monitoring. The camera transmits live video footage to YOLO V8, a software program trained on vast datasets to identify objects like fire in real-time. Upon detecting a fire, an Arduino Uno microcontroller receives the signal from YOLO V8 and triggering immediate alerts. A loud buzzer and speaker activation raise awareness of the fire in the vicinity. Pump is activated for mitigation of fire through sprinklers. Additionally, a Node MCU microcontroller leverages the Telegram messaging app to send notifications, potentially reaching designated personnel or emergency services for a swift response. This system offers a comprehensive fire detection and notification solution, utilizing image recognition software and microcontrollers for enhanced safety.

4.2 FLOW CHART

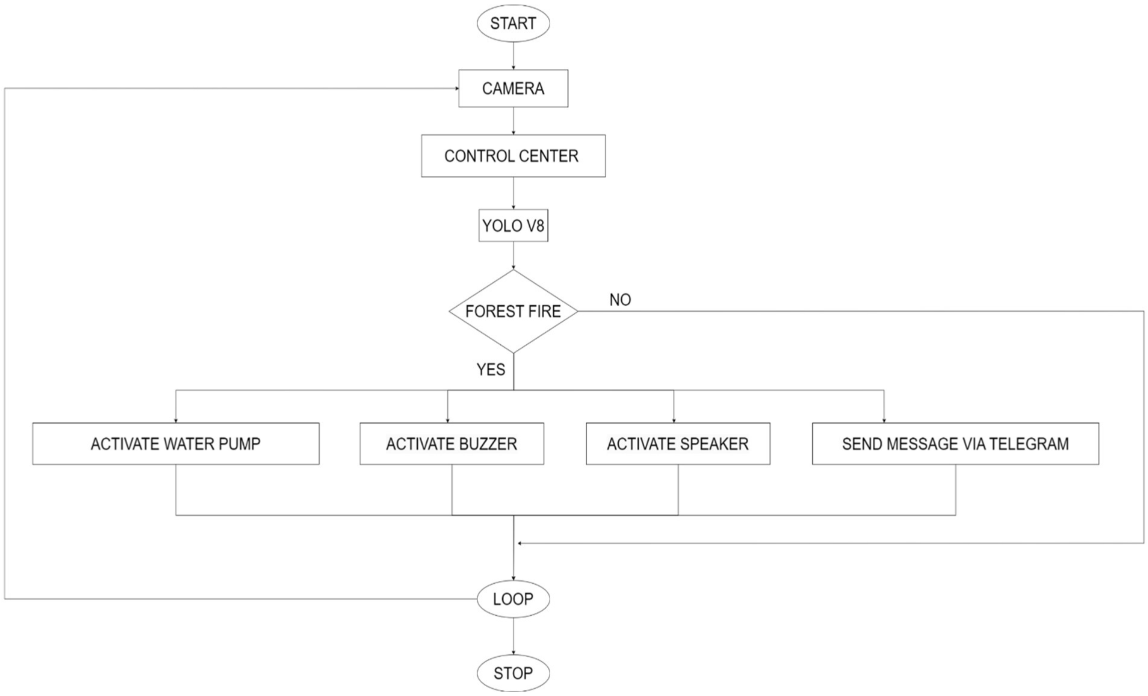

Fig 4.2.1 Flowchart

The diagram illustrates a forest fire notification system designed for early detection and rapid response. A strategically positioned camera continuously captures live video footage of the forest. This video feed is then analysed by YOLO V8, a powerful software program trained on vast datasets to recognize objects in real-time. In this specific application, YOLO V8 is programmed to identify the distinct visual cues associated with forest fires, such as flickering flames. If the analysis by YOLO V8 doesn't detect any signs of fire, the system remains vigilant, continuously monitoring the video feed through a loop back to the camera. However, upon identifying a potential fire, the system springs into action and activates water pump, Buzzer and Speaker. Simultaneously an alert notification is also sent through telegram. The detailed flow description is as follows:

Start:

The forest fire detection system initiates its operation as the camera continuously captures images of the forested area.

1. Image Analysis:

Captured images are swiftly transmitted to a central control center where advanced image recognition software diligently scrutinizes them for any indications of fire. In the absence of any detected fire, the system seamlessly loops back to capturing subsequent images. However, upon detecting the presence of fire, the system proceeds to the next step.

2. Buzzer Alert:

Upon confirmation of a fire outbreak, an immediate alert is triggered through the activation of a high-frequency Buzzer. Utilizing high-frequency infrasound, the Buzzer emits tones beyond the auditory range of most wild animals, ensuring effective evacuation.

3. Fire Mitigation:

To swiftly combat the fire threat, the system activates a water pump to initiate immediate fire suppression through the deployment of sprinklers. Simultaneously, based on the assessed severity of the fire, a firefighting crew equipped with appropriate firefighting gear and equipment is dispatched to the affected area to contain and extinguish the fire.

4. Tribal Alerts:

As part of the system's comprehensive response strategy, efforts are made to identify and assess whether tribal lands or cultural sites are at risk due to the advancing fire. Upon identification of potentially impacted areas, speakers integrated into the system are utilized to issue timely alerts and warnings to the tribal community residing nearby, ensuring their safety and facilitating prompt evacuation if necessary.

5. Notify Authorities:

In tandem with on-the-ground response efforts, urgent notifications are dispatched to relevant authorities, such as forest rangers or firefighters, via the Telegram messaging platform. These alerts provide crucial information regarding the fire's location, severity, and potential impact, enabling swift mobilization of resources and coordinated firefighting efforts.

End:

The forest fire detection system diligently continues to monitor the progression of the fire and implements mitigation strategies until the fire is successfully extinguished. Following the containment of the fire, a comprehensive post-fire assessment may be conducted to evaluate the extent of fire damage and plan for subsequent ecological restoration efforts.

4.3 SEQUENCE DIAGRAM

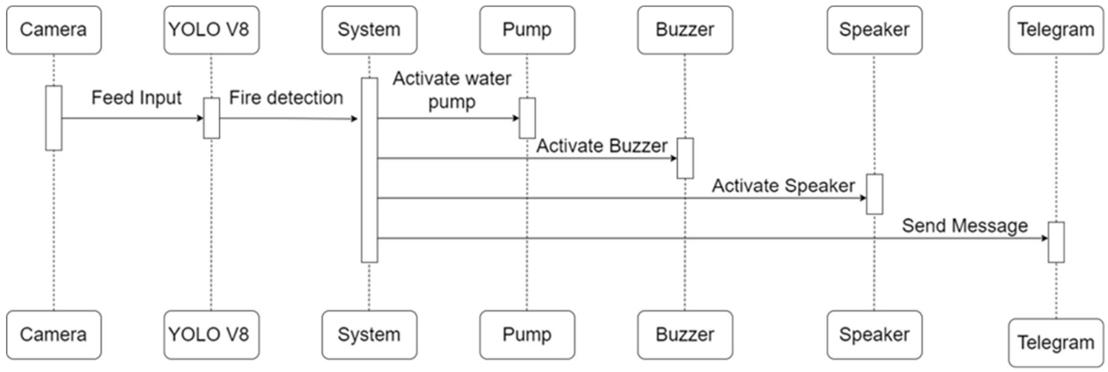

Fig. 4.3.1 Sequence diagram

The fig:4.3.1 depicts the sequence diagram and the sequence of how the messages are transferred between the objects to complete task. In a continuous loop, step 1 begins with the camera capturing live video footage of the monitored area. This captured video is then transmitted to YOLO V8, a powerful object detection program (step 2). YOLO V8 meticulously analyzes the video in step 3, searching for visual indicators of fire based on its extensive training data. If no fire is detected (step 4), the system returns to step 1, capturing a fresh video frame and restarting the analysis loop for uninterrupted monitoring. However, upon identifying a fire in step 5, the loop is broken, and the system initiates immediate action. An audible alarm is triggered by activating the buzzer in step 6, alerting people nearby to the potential danger. Additionally, the speaker may be activated in step 7 to broadcast a pre-recorded fire alarm message, further emphasizing the urgency. Finally, step 8 pump is activated for mitigation of fire through sprinklers, step 9 leverages the Telegram messaging app to send a notification, potentially reaching designated personnel or emergency services, ensuring a swift and coordinated response to the fire threat. The diagram concludes at step 10, although the system would perpetually monitor the environment and repeat this process if another fire is detected.

4.4 DATA FLOW DIAGRAM

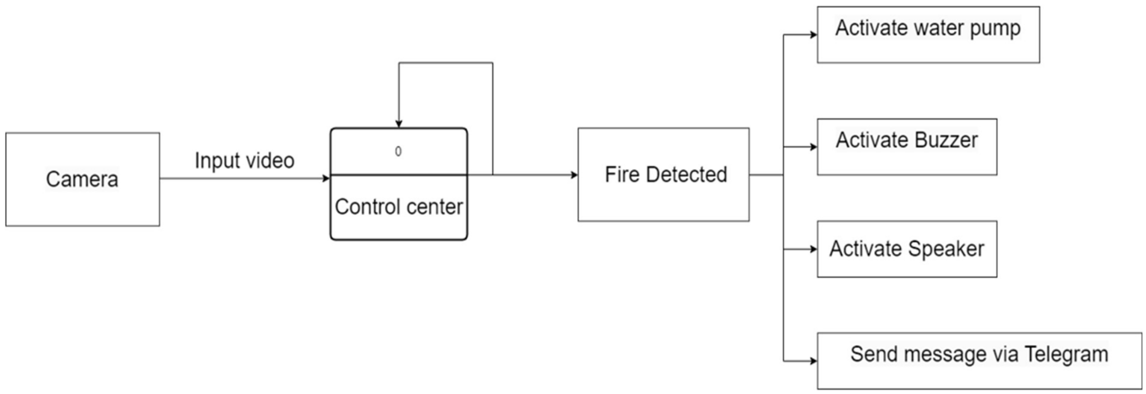

Fig 4.4.1 Data Flow Diagram Level 0

The level 0 data flow diagram depicts Eco-Harbor system in its entirety. Two external entities interact with the system: the forest department, who can activate or deactivate it, and the fire department, who may receive notifications. The core process involves continuous monitoring for signs of fire through video feed captured by cameras. This data is compared against the system's current state stored within by trained YOLO algorithm. If the "Decision: Fire Detected" block identifies a fire based on the data, two data flows are triggered: "Fire Alarm" activates audible alerts within the area, and "Notification" sends information to the forest and fire departments for a swift response. Also, if available it automatically activates a water pump to mitigate the fire.

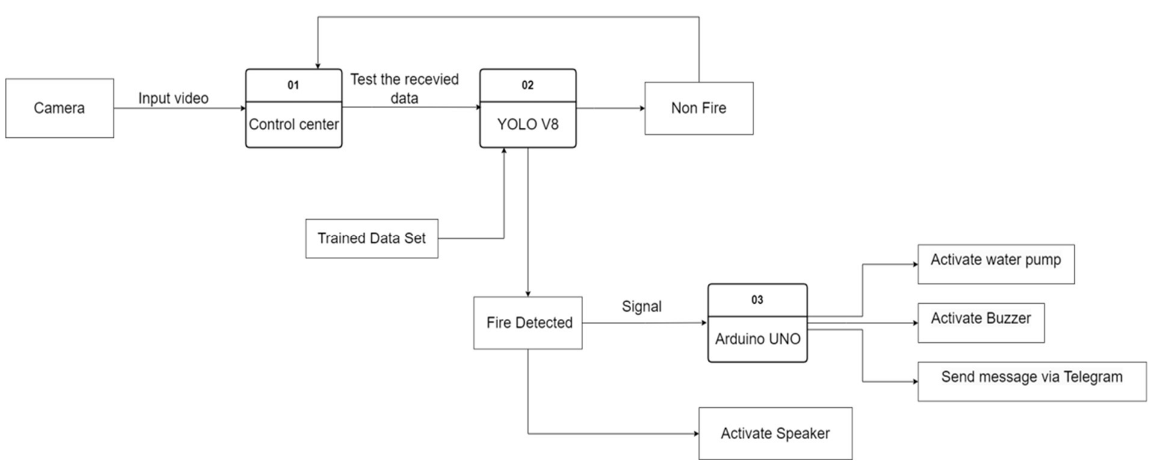

Fig 4.4.2 Data Flow Diagram Level 1

The level 1 data flow diagram delves deeper into the Eco-Harbor system's inner workings. The camera serves as the primary external entity, constantly capturing live video ("Video") of the monitored area. This video feed is then analyzed by the "Analyze Video" process, powered by YOLO V8 software. YOLO V8 relies on a dedicated data store ("Trained Dataset") containing training examples to effectively identify fire within the video footage. Based on the analysis, the "Decision: Fire Detected" block determines if a fire is present.

If a fire is detected, three key actions occur. First, this triggers two immediate alerts: a loud buzzer to the wild life and a speaker that may broadcast a pre-recorded fire alarm message. Secondly Automatic pump is activated for mitigation of fire through sprinklers. Third, a notification ("Notification") is sent via an external entity, possibly the Telegram messaging app, to the forest and fire departments or other designated personnel. This notification ensures a swift response to the fire threat, potentially saving lives and minimizing damage.

4.5 ACTIVITY DIAGRAM

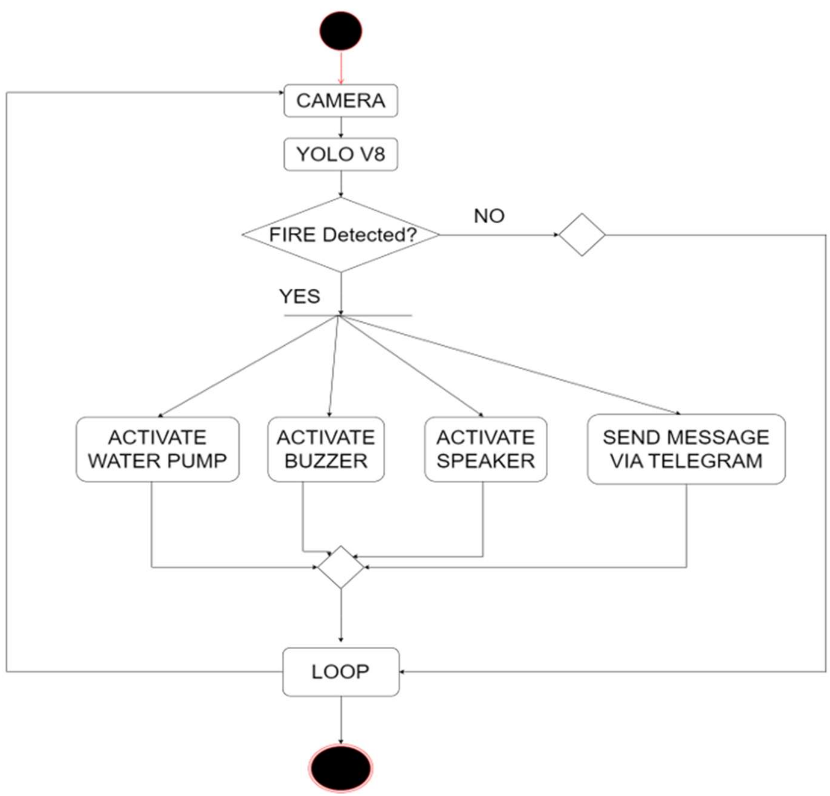

Fig 4.5.1 Activity Diagram

The activity diagram illustrates the continuous operation of the Eco-Harbor fire detection system. In a loop, the camera constantly captures live video footage of the monitored area. This video feed is then passed on to YOLO V8, an object detection program. YOLO V8 meticulously analyzes the video, searching for signs of fire. If no fire is detected, the system returns to the camera, capturing a fresh video frame and restarting the analysis loop. However, upon identifying a fire, the system springs into action. It activates an audible alarm through the buzzer, speaker and pump(sprinklers), alerting those nearby to the immediate danger. Additionally, the system leverages the Telegram messaging app to send a notification, potentially reaching designated personnel or emergency services. This ensures a swift and coordinated response to the fire threat, maximizing the chances of extinguishing the fire and minimizing damage.

CHAPTER 5

SYSTEM REQUIREMENTS

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

5.1 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are the functions offered by the system. It includes time constraints and constraints on the development process and standards. The non-functional requirements are as follows:
Then the customers can use easily, so it does not require much training time.

5.1.1 Speed: The system should process the given input into output within appropriate time.

5.1.2 Ease of use: The software should be user friendly.

5.1.3 Reliability: The rate of failures should be less then only the system is more reliable

5.1.4 Portability: It should be easy to implement in any system.

5.2 HARDWARE REQUIREMENTS:

- **Arduino UNO Microcontroller:** Arduino UNO is a popular microcontroller board based on the ATmega328P microcontroller. It's widely used for prototyping and DIY projects due to its simplicity and versatility.

- **ESP8266 Microcontroller:** ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability. It's commonly used for IoT (Internet of Things) projects due to its built-in Wi-Fi connectivity.

- **ESP32 Camera:** ESP32 Camera is a development board based on the ESP32 microcontroller with a built-in camera module. It allows for easy integration of camera functionalities into IoT projects.

- **Power Supply:** This can refer to any source of electrical power used to energize the components of your project. It could be a wall adapter, batteries, or any other power source.

- **LCD Display:** LCD (Liquid Crystal Display) display is a flat panel display technology commonly used in electronic devices for displaying text and images. It's often used in projects to provide visual feedback.

- **Relay:** A relay is an electromechanical switch used to control high-power devices with low-power signals. It allows you to control devices like lights, motors, or heaters using a microcontroller.

- **Pump:** A pump is a mechanical device used to move fluids (liquids or gases) from one place to another. It could be used in various applications such as water circulation, irrigation, or fluid transfer.

- **Buzzer:** A buzzer is an electronic component that produces sound when an electrical signal is applied to it. It's commonly used for alarms, notifications, or simple sound effects in projects.

- **Speaker:** A speaker is a transducer that converts electrical signals into sound waves. It's used for audio output in projects, such as playing music or speech.

- **Bluetooth Module:** A Bluetooth module enables communication between electronic devices over short distances using Bluetooth technology. It's often used for wireless data transfer or remote control applications.

- **Battery:** A battery is a device that stores chemical energy and converts it into electrical energy. It's commonly used as a portable power source for electronic devices and projects.

- **Servo Motor:** A servo motor is a rotary actuator that allows for precise control of angular position. It's commonly used in robotics and automation projects for controlling the position of mechanical components.

- **Connecting Wires:** These are wires used to establish electrical connections between various components in your project. They come in different sizes, colors, and types (such as jumper wires, breadboard wires, or hookup wires) to suit different needs.

5.3 SOFTWARE REQUIREMENTS:

- **Arduino IDE:** The Arduino IDE is a software for programming Arduino microcontroller boards. It features a code editor with syntax highlighting, integrated compiler, upload tool for transferring code to boards, serial monitor for debugging

- **CVAT:** CVAT, or Computer Vision Annotation Tool, is used for annotating images and videos to generate datasets for training and testing computer vision algorithms.
- It offers features like object detection, image segmentation, keypoint annotation, and tracking, making it invaluable for developing and evaluating machine learning models for tasks like object recognition and scene understanding.

- **Python IDLE:** Python IDLE (Integrated Development and Learning Environment) is a beginner-friendly Python interpreter and code editor.
- It offers features like syntax highlighting, code completion, debugging, and a built-in Python shell.
- IDLE facilitates rapid prototyping, script execution, and interactive Python sessions, making it ideal for learning, experimentation, and small-scale development projects.

- **Google Colab:** Google Colab is a cloud-based platform that provides free access to Jupyter notebooks and computing resources like Graphical Processing Unit (GPU) and Tensor Processing Units (TPU).
- It enables collaborative coding, data analysis, and machine learning experimentation with Python, offering seamless integration with Google Drive and libraries like TensorFlow.

5.4 SOFTWARE ANALYSIS

A software requirements definition is an abstract description of the services, which the system should provide, and the constraints under which the system must operate. It should only specify only the external behavior of the system and is not concerned with system design characteristics. It is a solution, in a natural language plus diagrams, of what services the system is expected to provide and the constraints under which it must operate.

5.4.1 Python 3.10 and Above

Python is a multi-paradigm programming language. Object oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming.

Python uses dynamic typing and a combination of reference counting and a cycle- detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

5.4.2 Important reasons using python

5.4.2.1 Readable and Maintainable Code

5.4.2.2 Multiple Programming Paradigms

5.4.2.3 Compatible with Major Platforms and Systems

5.4.2.4 Robust Standard Library

5.4.2.5 Many Open-Source Frameworks and Tools

5.4.2.6 Simplify Complex Software Development

CHAPTER 6

IMPLEMENTATION

THERE ARE THREE MAIN MODULES IN OUR PROJECT

- Input module –ESP32 Camera, ESP8266, Servo Motor, dataset are provided.
- Processing module – YOLO algorithm.
- Output module – Arduino UNO, Buzzer, Speaker, Water pump, ESP8266.

6.1 INPUT MODULE

- ### ESP32 CAMERA

Fig 6.1.1 ESP32 Camera

The ESP32 Camera is a versatile development board integrating the ESP32 chip with a camera module, making it capable of capturing images and video footage. It features built-in Wi-Fi and Bluetooth connectivity, allowing for wireless communication and remote control. Equipped with a powerful dual-core processor running at up to 240 MHz, the ESP32 Camera offers ample processing power for image processing tasks. The board also provides GPIO pins for interfacing with external components, expanding its functionality for diverse projects. Additionally, it supports microSD card storage, enabling local storage of captured media.

- ### ESP8266 [NODE MCU]

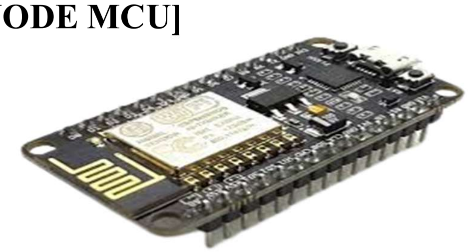

Fig 6.1.2 ESP8266

Node MCU is an open-source firmware and development board that is based on the ESP8266 Wi-Fi chip. It is designed for easy development of IoT projects, providing a low-cost and easy-to-use platform. The board features an ESP8266 chip that provides Wi-Fi connectivity, along with a USB-to-serial interface for easy programming and debugging. Node MCU is programmable with the Lua scripting language, which allows for quick and easy development of IoT applications. It also supports programming with Arduino IDE and MicroPython, making it a versatile platform for a range of projects.

➢ SERVO MOTOR

Fig 6.1.3 Servo Motor

A servo motor is a rotary or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration in a mechanical system. A servo motor typically incorporates a gear train mechanism to increase torque output and reduce speed. By adjusting the gear ratio, servo motors can achieve the desired combination of speed and torque for specific applications.

6.2 PROCESSING MODULE

• The captured is passed through YOLO algorithm through which image processing and classification are done.

• YOLO (You Only Look Once) is one of the most popular modules for real-time object detection and image segmentation.

• YOLO is a Convolutional Neural Network that predicts bounding boxes and class probabilities of an image in a single evaluation.

• There are 3 essential blocks in the algorithm and everything will occur in these blocks, which are: Backbone, Neck and Head.

Upon detecting a fire, an Arduino Uno microcontroller activates a buzzer and speaker for immediate audible alerts and automatic pump is activated for mitigation of fire through sprinklers. The system also sends notifications via Telegram through a Node MCU microcontroller, keeping occupants informed even remotely.

➢ ARDUINO UNO

Fig 6.2.1 Arduino UNO architecture.

The Arduino UNO Microcontroller is a versatile development board favored by makers, hobbyists, and professionals alike. Central to its design is the ATmega328P microcontroller chip, an 8-bit AVR processor clocked at 16MHz. This microcontroller provides ample computational power for a wide array of projects. With 14 digital input/output pins and 6 analog input pins, the Arduino UNO facilitates interfacing with various external devices, including sensors, actuators, and communication modules. Its built-in voltage regulator allows flexible power options, accommodating both DC power supplies and USB connections. Equipped with a USB interface for programming and serial communication, as well as reset and status LEDs, the board offers user-friendly functionality. Moreover, its compatibility with the Arduino IDE and extensive ecosystem of shields and libraries makes it an accessible platform for prototyping and experimentation. With its open-source nature and robust community support, the Arduino UNO empowers creators to bring their ideas to life with ease and efficiency.

6.3 OUTPUT MODULE

➢ SPEAKER

Fig 6.3.1 Speaker

Integrated with fire detection systems, speakers can automatically emit warning signals upon detecting signs of wildfires. Speakers play a vital role in wildlife relocation initiatives by broadcasting alerts to tribal members and volunteers involved in rescue and relocation efforts. They ensure that fire alerts and emergency messages reach all members of the tribal community, including those in remote or rural areas with limited access to electronic devices or internet connectivity.

➢ WATER PUMP

Fig 6.3.2 Water pump

Water pumps are used to power sprinkler systems installed in strategic locations within forests. These systems help in detecting and suppressing fires by spraying water over vulnerable areas, especially during dry seasons or when fire risks are high. Pumps can also be used to draw water from natural sources like rivers, lakes, or ponds to create fire breaks or dampen surrounding vegetation, slowing down the spread of wildfires.

➢ BUZZER

Fig 6.3.3 Buzzer

The high-frequency buzzer utilized in the Eco- Harbor Forest fire detection system serves as a crucial component in the multifaceted approach to wildfire mitigation and wildlife protection. Emitting sound waves designed to irritate and deter animals, the buzzer effectively prompts wildlife to relocate away from fire-affected areas, ensuring their safety and preventing harm. This humane approach complements the system's overall goal of minimizing ecological damage and safeguarding biodiversity in forested regions.

➢ LIQUID CRYSTAL DISPLAY

Fig 6.3.4 Liquid Crystal Display

LCD (Liquid Crystal Display) display is a flat panel display technology commonly used in electronic devices for displaying text and images. It's often used in projects to provide visual feedback.

6.4ALGORITHM

START
Step 1 Initialize:

- Connect to the camera and Arduino board.
- Load a pre-trained YOLOV8 model to detect fire.

Step 2 Loop:

- Capture a frame from the camera.
- Use YOLO to find fire in the frame.
- If fire is detected with confidence > 50:
    - Show a bounding box around the fire on the video.
    - Initiate cooldown timer.
    - Activate voice alert.
    - Activate buzzer and water pump.
    - Initiate alert message through telegram.
- After the specified cooldown period, reset to allow new detections.

END

6.5 CODE STRUCTURE

Hardware code:

```
#include "esp_camera.h"
#include <WiFi.h>
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
#include "camera_pins.h"

// ===========================
// Enter your WiFi credentials
// ===========================
const char* ssid = "ADMIN02_2.4G";
const char* password = "De%p#Tc$E@123";

void startCameraServer();
void setupLedFlash(int pin);

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
```

```
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sccb_sda = SIOD_GPIO_NUM;
  config.jpeg_quality = 12;
  config.fb_count = 1;

  // if PSRAM IC present, init with UXGA resolution and higher JPEG quality
  //                      for larger pre-allocated frame buffer.
  if(config.pixel_format == PIXFORMAT_JPEG){
    if(psramFound()){
      config.jpeg_quality = 10;
      config.fb_count = 2;
      config.grab_mode = CAMERA_GRAB_LATEST;

#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }

  sensor_t * s = esp_camera_sensor_get();
  // initial sensors are flipped vertically and colors are a bit saturated
  if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
  }
  // drop down frame size for higher initial frame rate
  if(config.pixel_format == PIXFORMAT_JPEG){
    s->set_framesize(s, FRAMESIZE_QVGA);
  }


  WiFi.begin(ssid, password);
  WiFi.setSleep(false);
```

```
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  startCameraServer();

  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
}

void loop() {
  // Do nothing. Everything is done in another task by the web server
  delay(10000);
}

#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

// Replace with your network credentials
const char* ssid = "ADMIN02_2.4G";
const char* password = "De%p#Tc$E@123";

// Initialize Telegram BOT
#define BOTtoken "7127617992:AAEZr83aVRUmwX-TWnYgywvOPUCbLHDI8-A"  // your
Bot Token (Get from Botfather)

#define CHAT_ID "884847034"

X509List cert(TELEGRAM_CERTIFICATE_ROOT);
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

void setup() {
  Serial.begin(115200);
  configTime(0, 0, "pool.ntp.org");      // get UTC time via NTP
  client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org

  // Attempt to connect to Wifi network:
  Serial.print("Connecting Wifi: ");
  Serial.println(ssid);
 }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("IP address: ");
```

```
  Serial.println(WiFi.localIP());

  bot.sendMessage(CHAT_ID, "Eco-Harbor Connected successfully", "");
}

void loop() {
  if(digitalRead(D5) == 1) { // PIR Motion Sensor
    bot.sendMessage(CHAT_ID, "Alert! \n   Forest Fire detected! \n   Camera No: 01 \n\n
Immediate action needed! \n\n -------------------------------------------- \n \n अलर्ट!\n   जंगली आग
का पता लगा!\n   कैमरा नंबर: 01 \n\n   तत्काल कार्रवाई की आवश्यकता है! \n\n ---------------------------
------------------- \n \n ಎಚ್ಚರಿಕೆ!\n   ಕಾಡ್ಗಿಚ್ಚು ಗುರುತಿಸಲಾಗಿದೆ!\n   ಕ್ಯಾಮರಾ ಸಂಖ್ಯೆ: 01 \n \n
ತಕ್ಷಣ ಕ್ರಮದ ಅಗತ್ಯವಿದೆ!",  "");
    Serial.println("Fire detected!");
    delay(1000); // delay to avoid continuous messaging
  }
}
```

Software Code:

```
from ultralytics import YOLO
import cvzone
import cv2
import math
from pyfirmata import Arduino, util
import time
import pygame

url = 'http://192.168.0.113:81/stream'

# Connect to Arduino
board = Arduino('COM11')
pygame.init()
alarm_sound = pygame.mixer.Sound('firealert.mp3')
# Initialize iterator
it = util.Iterator(board)
it.start()

# Define the pin connections to aurdino
buzz_pin = 9
relay_pin = 10
nodemcu_pin= 8

# Initialize pin as output
board.digital[buzz_pin].mode = 1
board.digital[relay_pin].mode = 1
board.digital[nodemcu_pin].mode = 1
# Running real-time from webcam
cap = cv2.VideoCapture(url)
model = YOLO('best1.pt')
```

```
# Reading the classes
classnames = ['fire']
# Flag to track if the alarm is already triggered
alarm_triggered = False
# Time duration (in seconds) for the alarm cooldown period
cooldown_duration = 60  # Change this to set the cooldown duration
# Initialize the timestamp for cooldown tracking
cooldown_start_time = 0
while True:
    ret, frame = cap.read()
    cv2.imshow('IP Camera', frame)
    result = model(frame, stream=True)
    # Getting bbox, confidence, and class names information to work with
    for info in result:
        boxes = info.boxes
        for box in boxes:
            confidence = box.conf[0]
            confidence = math.ceil(confidence * 100)
            Class = int(box.cls[0])
            if confidence > 50:
                x1, y1, x2, y2 = box.xyxy[0]
                x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
                cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 5)
                cvzone.putTextRect(frame, f'{classnames[Class]} {confidence}%', [x1 + 8, y1 + 100],
                                   scale=1.5, thickness=2)
                # Activate buzzer and pump when fire is detected
                if not alarm_triggered:
                    alarm_sound.play()
                    board.digital[buzz_pin].write(1)  # Turn buzzer on
                    board.digital[relay_pin].write(1)  # Turn pump on
                    board.digital[nodemcu_pin].write(1)
                    alarm_triggered = True
                    cooldown_start_time = time.time()
                    time.sleep(6)
                    board.digital[buzz_pin].write(0)   # Turn buzzer off
                    board.digital[relay_pin].write(0)  # Turn pump off
                    board.digital[nodemcu_pin].write(0)

    if alarm_triggered and time.time() - cooldown_start_time > cooldown_duration:
        alarm_triggered = False
    cv2.imshow('frame', frame)
    cv2.waitKey(1)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

CHAPTER 7

SOFTWARE TESTING

7.1 SOFTWARE TESTING INTRODUCTION

Software testing is a process used to help identify the correctness, completeness and quality of developed computer software. Software testing is the process used to measure the quality of developed software. Testing is the process of executing a program with the intent of finding errors. Software testing is often referred to as verification &validation.

7.2 EXPLANATION FOR SDLC

The software development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application.

7.3 PHASES OF SOFTWARE TESTING

* Requirement Analysis
* Design
* Testing
* Maintenance System

7.3.1 Requirement Analysis

The requirements of a desired software product are extracted. Based the business scenario the SRS (Software Requirement Specification) document is prepared in this phase.

7.3.2 Design

Plans are laid out concerning the physical construction, hardware, operating systems, programming, communications, and security issues for the software. Design phase is concerned with making sure the software system will meet the requirements of the product.

7.3.3 Testing

Testing is evaluating the software to check for the user requirements. Here the software is evaluated with intent of finding defects.

7.3.4 Maintenance

Once the new system is up and running for a while, it should be exhaustively evaluated. Maintenance must be kept up rigorously at all times. Users of the system should be kept up- to-date concerning the latest modifications and procedures.

7.4 TYPES OF TESTING

- White-Box Testing
- Black-Box Testing
- Grey-Box Testing

7.4.1 White Box Testing

White box testing as the name suggests gives the internal view of the software. This type of testing is also known as structural testing or glass box testing as well, as the interest lies in what lies inside the box.

7.4.2 Black Box Testing

It is also called as behavioral testing. It focuses on the functional requirements of the software. Testing either functional or non-functional without reference to the internal structure of the component or system is called black box testing.

7.4.3 Grey Box Testing

Grey box testing is the combination n of black box and white box testing. Intention of this testing is to find out defects related to bad design or bad implementation of the system.

7.5 LEVELS OF TESTING USED IN THE PROJECT

7.5.1 Unit Testing

Initialization testing is the first level of dynamic testing and is first the responsibility of developers and then that of the test engineers. Unit testing is performed after the expected test results are met or differences are explainable/acceptable**.** All module which make application are tested. Integration testing is to make sure that the interaction of two or more components produces results that satisfy functional requirement**.**

7.5.2 System Testing

To test the complete system in terms of functionality and non- functionality. It is black box testing, performed by the Test Team, and at the start of the system testing the complete system is configured in a controlled environment.

7.5.3 Functional Testing

The outgoing links from all the pages from specific domain under test. Test all internal links. Test links jumping on the same pages. Check for the default values of fields.

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

7.6 TEST CASES

Sl. No	Description	Input	Expected Result	Actual Result	Remarks
01	Verify Camera Feed Connectivity	Camera not connected	Displays live footage without distortion	Camera not connected; unable to verify feed	Fail
02	Verify Camera Feed Connectivity	Camera connected and operational	Camera feed is properly connected and displays live footage without distortion	Camera feed is stable and displays live footage clearly without distortion	Pass
03	Verify Camera Feed Connectivity	No Internet connectivity	Displays live footage without distortion	Camera not connected; unable to verify feed	Fail

04	Test YOLO Algorithm Fire Detection	Live camera feed with fire present	YOLO algorithm accurately detects fire in the camera feed.	YOLO algorithm detects fire with high accuracy, minimal false positives/negatives	Pass
05	Test Fire Response Mechanism	Fire detected in camera feed but devices not connected.	Upon fire detection, water pump, buzzer, and speaker are activated promptly	Water pump, buzzer, and speaker do not activate promptly upon fire detection	Fail
06	Test Buzzer and Speaker Activation	Fire detected in camera feed	Buzzer and speaker produce audible alerts promptly upon fire detection	Buzzer and speaker produce loud, audible alerts upon fire detection	Pass
07	Verify Water Pump Functionality	Fire detected in camera feed, low voltage power supply.	Water pump activates but fails to spray water effectively	Water pump activates but sprays insufficient water; ineffective firefighting	Fail
08	Test YOLO Algorithm with Obstructed View	Fire partially obscured in camera feed	YOLO algorithm accurately detects partially obscured fire in the camera feed	YOLO algorithm detects partially obscured fire accurately	Pass
09	Test Fire Detection with Low Light Conditions	Low light conditions in camera feed	YOLO algorithm accurately detects fire in low light conditions	YOLO algorithm detects fire accurately despite low light	Pass
10	Verify Water Pump Activation Timing	Delayed activation of water pump upon fire detection	Water pump activates with a slight delay upon fire detection	Water pump activates with a slight delay; still effectively sprays water	Pass

11	Buzzer Sound Signal	Fire detected in camera feed	Buzzer emits a loud sound signal to deter wildlife from approaching the fire zone	Buzzer emits a loud sound signal to deter wildlife from approaching the fire zone	Pass
12	Telegram Alert	Fire detected in camera feed	Telegram message containing fire alert sent to designated recipients	YOLO algorithm detects partially obscured fire accurately	Pass
13	Telegram Alert Content	Fire detected in camera feed	Telegram message contains accurate information about the fire in multiple languages	Telegram message contains accurate information about the fire in Kannada, English and Hindi	Pass
14	Verify Integration of All Components	Improper wire connections	All components work together seamlessly in real-time fire detection and response scenarios	No response mechanisms activated	Fail
15	Verify Integration of All Components	All components operational, fire detected in camera feed	All components work together seamlessly in real-time fire detection and response scenarios	Camera feed, YOLO algorithm, response mechanisms, and Telegram message sending integrated seamlessly	Pass

Table 7.6.1: Test Cases

CHAPTER 8

RESULTS

Fig 8.1 Eco-Harbor Implementation Model

This Eco-Harbor fire detection system leverages cameras and software for real-time monitoring. A camera captures live video, feeding it to YOLO V8, a program trained to identify fire in the footage. Upon detecting a fire, an Arduino Uno microcontroller activates a buzzer and speaker for immediate audible alerts and automatic pump is activated for mitigation of fire through sprinklers. The system also sends notifications via Telegram through a Node MCU microcontroller, keeping occupants informed even remotely. This setup utilizes YOLO V8 and microcontrollers for a comprehensive fire detection and notification solution.

The hardware setup of the **"ECO-HARBOR"**

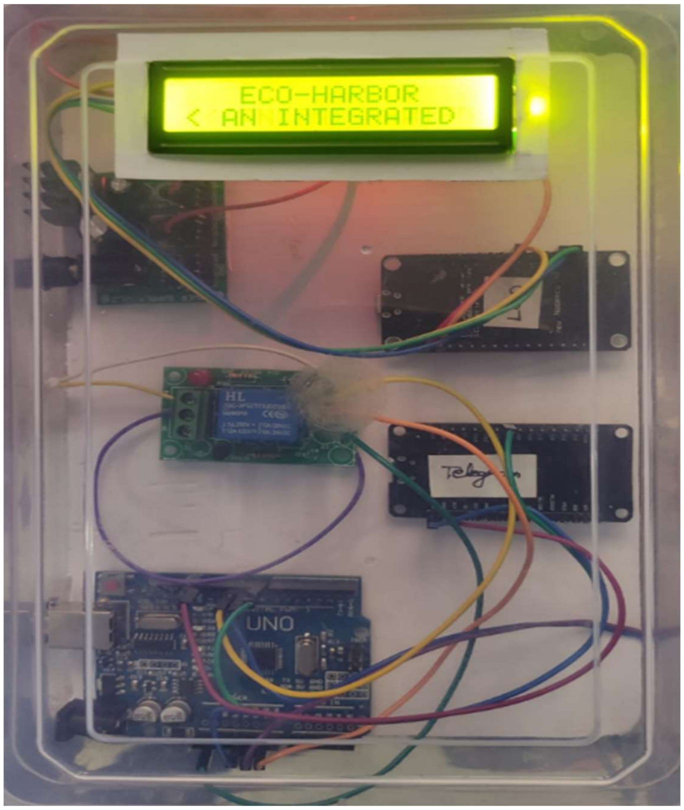

Fig 8.2: Hardware setup

The system uses:

- Arduino UNO microcontroller
- ESP 8266 microcontroller
- ESP32 Camera
- power supply
- LCD display
- Relay
- Pump
- Buzzer
- Speaker
- Bluetooth Module
- Battery
- Servo Motor
- Connecting Wires

Fig 8.3: Fire Detection

Captured images through ESP32 Camera are swiftly transmitted to a central control center where advanced image recognition software diligently scrutinizes them for any indications of fire. In the absence of any detected fire, the system seamlessly loops back to capturing subsequent images. However, upon detecting the presence of fire, the system proceeds to the next step or next actions.

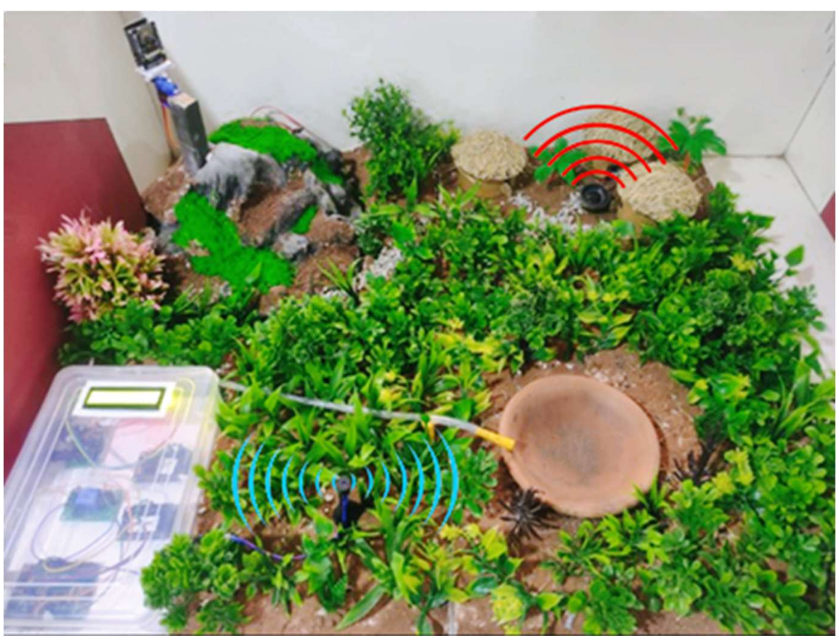

Fig 8.4: Wildlife Relocation, And Tribal Alert

Upon confirmation of a fire outbreak, an immediate alert is triggered through the activation of a high-frequency Buzzer. Utilizing high-frequency infrasound, the Buzzer emits tones beyond the auditory range of most wild animals, ensuring effective evacuation and Upon identification of potentially impacted areas, speakers integrated into the system are utilized to issue timely alerts and warnings to the tribal community residing nearby, ensuring their safety and facilitating prompt evacuation if necessary.

Fig 8.5: Fire Mitigation.

To swiftly combat the fire threat, the system activates a water pump to initiate immediate fire suppression through the deployment of sprinklers. Simultaneously, based on the assessed severity of the fire, a firefighting crew equipped with appropriate firefighting gear and equipment is dispatched to the affected area to contain and extinguish the fire.

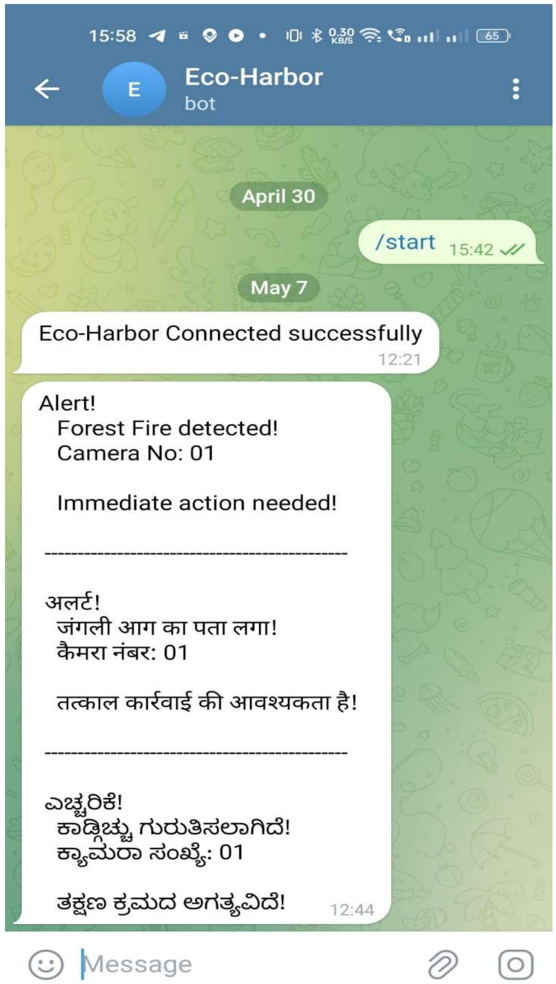

Fig 8.6: Telegram alert for authorities.

In tandem with on-the-ground response efforts, urgent notifications are dispatched to relevant authorities, such as forest rangers or firefighters, via the Telegram messaging platform. These alerts provide crucial information regarding the fire's location, severity, and potential impact, enabling swift mobilization of resources and coordinated firefighting efforts.

CONCLUSION

The system utilizing YOLO V8 and custom-trained datasets, presents a sophisticated solution for forest fire management, wildlife protection, and tribal community safety. Through the integration of Deep learning technology and IoT devices such as buzzers, speakers, water pumps, and sprinkler systems, we are developing an innovative approach to detect, mitigate, and respond to wildfires with precision and efficiency. YOLO V8's object detection capabilities enable real-time identification of fire outbreaks, while the deployment of alert systems ensures immediate notifications to both wildlife and tribal community, facilitating prompt evacuation and relocation efforts. Moreover, the utilization of water pumps and sprinkler systems enhances our project's ability to suppress fires effectively, minimizing their spread and mitigating potential damage to forests and surrounding areas. This comprehensive and scalable solution represents a significant advancement in forest fire management practices, with the potential to make a substantial impact on preserving natural ecosystems and safeguarding communities worldwide.

FUTURE ENHANCEMENT

Future enhancements for the proposed Eco- Harbor Forest fire detection system could focus on improving its capabilities, scalability, and integration with emerging technologies. Here are some potential avenues for enhancement:

1. Implement techniques for ongoing algorithm refinement, ensuring adaptability to changing fire patterns and environmental conditions without extensive retraining.
2. Integrate with cloud services for scalable data processing and analysis enhanced decision-making capabilities

REFERENCES

[1] Suresh Babu K.V, Venkata Sai Krishna Vanama, Arijit Roy, P. Ramachandra Prasad., "Assessment of forest fire danger using automatic weather stations and MODIS TERRA satellite datasets for the state Madhya Pradesh, India", International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017.

[2] Jayaram K, Janani K, Jeyaguru R, Kumaresh R, Muralidharan N, "Forest Fire Alerting System with GPS Co-ordinates Using IoT", 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019.

[3] A.Divya, T.Kavithanjali, P.Dharshini, "IoT Enabled Forest Fire Detection And Early Warning System", International Conference on Systems, Computation and networking, 2019.

[4] Mohit Dua, Mandeep Kumar, Gopal Singh Charan, Parre Sagar Ravi, "An Improved Approach for Fire Detection using Deep Learning Models", International Conference on Industry 4.0 Technology (I4Tech), 2020.

[5] Medi Rahul, Karnekanti Shiva Saketh, Attili Sanjeet and Nenavath Srinivas Naik, "Early Detection of Forest Fire using Deep Learning", IEEE Region 10 Conference (Tencon), Japan, 2020.

[6] Shivam Pareek, Shreya Shrivastava, Sonal Jhala, Juned A. Siddiqui, Savitanandan Patidar, "IoT and Image Processing based Forest Monitoring and Counteracting System", Proceedings of the Fourth International Conference on Trends in Electronics and Informatics (ICOEI), 2020.

[7] Vignesh.S, Tarun.M.G, Sankhasubhra Nandi, Sriram.M, Ashok P, "Forest Fire Detection And Guiding Animals To A Safe Area By Using Sensor Networks And Sound", 4th International Conference on Computing and Communications Technologies (ICCCT), 2021.

[8] Dai Quoc Tran, Minsoo Park, Yuntae Jeon, Jinyeong Bak, and Seunghee Park, "Forest-Fire Response System Using Deep-Learning-Based Approaches with CCTV Images and Weather Data", IEEE Access, June 21, 2022.

[9] Sayali. Madkar, Dr. Dipti.Y. Sakhare, Komal.A. Phutane, Anagha.P. Haral, Kirti.B. Nikam, S. Tharunyha, "Video Based Forest Fire and Smoke Detection Using YoLo and CNN", International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), 2022.

[10] Mostafa M. Fouda, Sadman Sakib, Zubair Md Fadlullah, Nidal Nasser, and Mohsen Guizani, "A Lightweight Hierarchical AI Model for UAV-Enabled Edge Computing with Forest-Fire Detection Use-Case, IEEE Network-November/December 2022.

[11] Vishal K. Singh, Chhaya Singh, and Haider Raza, "Event Classification and Intensity Discrimination for Forest Fire Inference With IoT", IEEE sensors journal, vol. 22, no. 9, may 1, 2022.

[12] Junwei Xie, Fan Yu, Hanxiang Wang, and Haotian Zheng, "Class Activation Map-Based Data Augmentation for Satellite Smoke Scene Detection", IEEE Geoscience And Remote Sensing Letters, Vol. 19, 2022.

[13] Lidong Wang, Huixi Zhang, Yin Zhang, Keyong Hu, and Kang An, "A Deep Learning-Based Experiment on Forest Wildfire Detection in Machine Vision Course" IEEE Access -March 2023.

[14] Liangji Zhang, Chao Lu, Haiwen Xu, Aibin Chen, Liujun Li, And Guoxiong Zhou, "MMFNet: Forest Fire Smoke Detection Using Multiscale Convergence Coordinated Pyramid Network With Mixed Attention And Fast-Robust NMS", IEEE Internet Of Things Journal, Vol. 10, No. 20, 15 October 2023.

[15] Mingming Luo, Jianhua Huang, Xiyan Sun, Zhengyao Yu, Yixuan Wan, "Small Target Forest Fire Recognition Method based on Deep Learning", 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), 2023

PUBLISHED SURVEY PAPER AND CERTIFICATES