**DEVELOP A SUITABLE TECHNOLOGY TO TRACK DEEP SEA FISHERMAN OR THEIR LOCATION ENSURING EFFICTIVE MONITORING AND ENHANCING SAFETY**

**ABSTRACT**

In disaster scenarios, rapid detection of human movements is paramount for effective search and rescue operations. This paper introduces an AI-enabled drone system leveraging the You Only Look Once (YOLO) algorithm for automatic detection of human movements. The system integrates a lightweight drone equipped with a camera and real-time object detection capabilities based on YOLO. This algorithm enables the drone to swiftly identify human presence amidst rubble, debris, or hazardous environments, aiding first responders in locating survivors quickly and efficiently. Using deep learning techniques, the system achieves high accuracy in detecting and tracking human movements in real-time. This capability provides valuable information to first responders, allowing for timely intervention and potentially saving lives while minimizing casualties. By combining the agility and versatility of drones with the precision of AI-based object detection, the proposed system offers a cost-effective solution for enhancing disaster response efforts. Its ability to rapidly assess and monitor disaster areas enhances situational awareness, enabling responders to allocate resources effectively and prioritize areas requiring immediate attention. This drone system presents a promising tool for improving the effectiveness and efficiency of disaster response operations.

**OBJECTIVE**

* Develop an AI-enabled drone system for rapid detection of human movements in disaster scenarios.
* Utilize the You Only Look Once (YOLO) algorithm for real-time object detection.
* Integrate a lightweight drone with a camera for agile deployment in disaster zones.
* Achieve high accuracy in identifying human presence amidst rubble, debris, or hazardous environments.
* Provide valuable real-time information to first responders for timely intervention.
* Enhance disaster response efforts with a cost-effective and efficient solution.
* Minimize casualties and potentially save lives by improving search and rescue operations.

**CHAPTER 1**

* 1. **INTRODUCTION**

The purpose of this project is to develop an innovative solution for enhancing search and rescue operations in disaster scenarios. Natural disasters such as earthquakes, hurricanes, and floods often result in widespread destruction, making it challenging for first responders to quickly locate and rescue survivors. Traditional methods of search and rescue are often slow and labor-intensive, leading to delays in providing aid and potentially increasing the risk to survivors. To address this critical issue, we propose the implementation of an AI-enabled drone system that can rapidly detect and track human movements in disaster environments. Disaster environments pose numerous challenges for search and rescue operations. These include collapsed buildings, rubble, debris, and hazardous conditions such as toxic fumes or unstable structures. These conditions make it difficult for first responders to navigate and locate survivors efficiently. Moreover, the time-sensitive nature of rescue operations means that any delay in finding survivors can have dire consequences. The proposed drone system aims to overcome these challenges by leveraging advanced technology. The You Only Look Once (YOLO) algorithm, known for its real-time object detection capabilities, is a key component of this system. By using YOLO, the drone can quickly scan the disaster area and identify human presence amidst the rubble and debris. This enables first responders to pinpoint the location of survivors more accurately and expedite the rescue process. The implementation of this project involves several stages. First, the drone is equipped with a lightweight camera and the necessary hardware for real-time processing of video data. Next, the YOLO algorithm is integrated into the drone's software, allowing it to analyze the live video feed and identify humans in the disaster area. This integration enables the drone to operate autonomously, scanning the environment and sending alerts to first responders when human movements are detected. The project includes testing and validation of the system in simulated disaster scenarios to ensure its effectiveness and reliability. This testing phase involves evaluating the accuracy and speed of the drone's detection and tracking capabilities in various environmental conditions. Feedback from these tests is used to refine the system and optimize its performance for real-world deployment. The implementation of this project involves leveraging advanced AI and drone technology to overcome the challenges of search and rescue operations in disaster environments. By integrating the YOLO algorithm into a lightweight drone system, we aim to provide first responders with a valuable tool for quickly locating and rescuing survivors, ultimately saving lives and minimizing casualties.

**1.2 DEEP LEARNING**

Deep learning is a method in artificial intelligence (AI) that teaches computers to process data in a way that is inspired by the human brain. Deep learning models can recognize complex patterns in pictures, text, sounds, and other data to produce accurate insights and predictions. Deep learning can be defined as the method of machine learning and artificial intelligence that is intended to intimidate humans and their actions based on certain human brain functions to make effective decisions. It is a very important data science element that channels its modeling based on data-driven techniques under predictive modeling and statistics. To drive such a human-like ability to adapt and learn and to function accordingly, there have to be some strong forces which we popularly called algorithms.

**1.2.1 Application of Deep learning**

Deep learning applications are used in industries from automated driving to medical devices. Automated Driving: Automotive researchers are using deep learning to automatically detect objects such as stop signs and traffic lights. In addition, deep learning is used to detect pedestrians, which helps decrease accidents.

**Aerospace and Defense**: Deep learning is used to identify objects from satellites that ovate area so interest, and identify safe or unsafe zones for troops. Medical Research: Cancer researchers are using deep learning to automatically detect cancer cells. Teams at UCLA built an advanced microscope that yields a high-dimensional dataset used to train a deep learning application to accurately identify cancer cells.

**Industrial Automation**: Deep learning is helping to improve worker safety around heavy machinery by automatically detecting when people or objects are within an unsafe distance of machines.

**Electronics**: Deep learning is being used in automated hearing and speech translation. For example, home assistance devices that respond to your voice and know your preferences are powered by deep learning applications.

**1.2.2 Deep Learning Works**

Most deep learning methods use neural network architectures, which is why deep learning models are often referred to as deep neural networks.

The term deep‖ usually refers to the number of hidden layers in the neural network. Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as150.Deep learning models are trained by using large sets of labeled data and neural network architectures that learn features directly from the data without the need for manual feature extraction. One of the most popular types of deep neural networks is known as neural networks (CNN or Convent). A CNN convolves learned features with input data, and uses 2D convolution layers, making this architecture well suited to processing 2D data, such as images.CNNs eliminate the need for manual [feature extraction,](https://www.mathworks.com/discovery/feature-extraction.html) so you do not need to identify features used to classify images. The CNN works by extracting features directly from images. The relevant features are not pre trained; they are learned while the network trains on a collection of images. This automated feature extraction makes deep learning models highly accurate for computer vision tasks such as object classification.

CNNs learn to detect different features of an image using tens or hundreds of hidden layers. Every hidden layer increases the complexity of the learned image features. For example, the first hidden layer could learn how to detect edges, and the last learns how to detect more complex shapes specifically catered to the shape of the object we are trying to recognize. Deep learning is a specialized form of machine learning. A machine learning workflow starts with relevant features being manually extracted from images. The features Are then used to create a model that categorizes the objects in the image. With a deep learning workflow, relevant features are automatically extracted from images. In addition, deep learning performs ―end-to-end learning ―where a network is given raw data and a task toper form, such as classification, and it learns how to do this automatically.

Another key differences deep learning algorithms scale with data, whereas shallow learning converges. Shallow learning refers to machine learning methods that plateau at a certain level of performance when you add more examples and training data to the network.

A key advantage of deep learning networks is that they often continue to improve as the size of your data increases. The three most common ways people use deep learning to perform object classification.

**IMAGE PROCESSOR**

An image processor does the functions of image acquisition, storage, pre-processing, segmentation, representation, recognition and interpretation and finally displays or records the resulting image As detailed in the diagram, the first step in the process is image acquisition by an imaging sensor in conjunction with a digitizer to digitize the image. The next step is the pre-processing step where the image is improved being fed as an input to the other processes. Pre-processing typically deals with enhancing, removing noise, isolating regions, etc. Segmentation partitions an image into its constituent parts or objects. The output of segmentation is usually raw pixel data, which consists of either the boundary of the region or the pixels in the region themselves. Representation is the process of transforming the raw pixel data into a form useful for subsequent processing by the computer. Description deals with extracting features that are basic in differentiating one class of objects from another. Recognition assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects. The knowledge about a problem domain is incorporated into the knowledge base. The knowledge base guides the operation of each processing module and also controls the interaction between the modules. Not all modules need be necessarily present for a specific function. The composition of the image processing system depends on its application. The frame rate of the image processor is normally around 25 frames per second

**FACE RECOGNITION**

Face recognition tends to be the most appealing visitor authentication as it is the most natural process for human identification, it is the least obtrusive method and yet it remains the most challenging modality. The first step in a face recognition or authentication system is face detection and feature extraction, which are necessary to locate the face position and obtain the face features in the image for further processing. The features obtained are then fed into the critical step of face recognition or authentication. The recognition or authentication process remains a challenging endeavor for researchers due to the myriad of faces that can be considered and the variability in the circumstances and ways under which the images of these faces are taken Therefore, feature extraction and subject recognition are considered the two focal points of this research. The system to be built will elaborate on how the learning and recognition phases are integrated into one system as it seeks higher recognition accuracy and faster processing time.

Face is detected automatically by using face detection technique and the entire face recognition is completed without touching with any hardware. Face detection is the first step of the face recognition system. The performance of the entire face recognition system is influenced by the reliability of the face detection. By using face detection, it can identify only the facial part of an image regardless of the background of this image. In this system, Viola-Jones face detection method is used. Viola-Jones rescales the detector instead of the input image and run the detector many times through the image – each time with a different size. Viola-Jones has devised a scale invariant detector that requires the same number of calculations whatever the size. This detector is constructed using a so-called integral image and some simple rectangular features reminiscent of Haar wavelets. Face recognition commonly includes feature extraction, feature reduction and recognition or classification.

**OpenCV**

OpenCV (Open Source Computer Vision) is an open-source library of computer vision and machine learning algorithms, designed to help developers create applications that can interpret and understand visual information from the world around us. It was originally developed by Intel in the late 1990s, and is now maintained by a community of developers. The OpenCV provides a wide range of functionalities, including image and video processing, feature detection, object recognition and tracking, camera calibration, and more. It supports several programming languages, including C++, Python, Java, and MATLAB, and can be used on multiple platforms, including Windows, Linux, macOS, iOS, and Android.

With its extensive library of pre-built functions and algorithms, OpenCV can be used in a variety of fields, including robotics, autonomous vehicles, surveillance systems, medical imaging, and more. Its ease of use and flexibility make it a popular choice for developers and researchers working in the field of computer vision.

**YOLO**

YOLO (You Only Look Once) is an object detection algorithm that was introduced in 2016 by Joseph Redmon, Ali Farhadi, and others. It is a state-of-the-art object detection algorithm that is known for its speed and accuracy. Unlike other object detection algorithms, YOLO looks at the entire image only once to detect objects, hence the name "You Only Look Once".

The YOLO algorithm works by dividing the image into a grid of cells and predicting bounding boxes and class probabilities for each cell. These bounding boxes are then adjusted based on the predicted offsets and the final predictions are made by combining the predictions from all the cells.

One of the major advantages of YOLO is its speed. YOLO can process images in real-time, which is critical for applications such as self-driving cars and surveillance systems. Another advantage is its accuracy, as YOLO can detect objects with high precision and recall rates.

YOLO has been widely adopted and has evolved over time. There are currently three versions of YOLO: YOLOv1, YOLOv2, and YOLOv3. Each version has improvements over the previous version, such as better accuracy, faster processing, and more features.

YOLO has been used in a wide range of applications, including autonomous driving, robotics, surveillance systems, and more. Its speed and accuracy make it a popular choice for real-time applications that require object detection.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1TITLE:** Smart Drone Surveillance System Based on AI and on IoT Communication in Case of Intrusion and Fire Accident

**AUTHOR:** Minh Long Hoang

**YEAR:** 2023

**DESCRIPTION:**

Currently, unmanned aerial vehicles (UAVs) are utilized in a wide range of applications [1,2,3,4,5], especially in surveillance systems [6,7]. Drone surveillance involves visually monitoring an individual, a group, items, or a situation to prevent potential threats. The establishment of an efficient surveillance system with drone fleets necessitates the smooth integration of dependable hardware and sophisticated automation software. In buildings and factories, there is a high demand for smart security systems with drone applications. Drones perform significantly faster than patrol vehicles or security personnel, enabling them to promptly arrive at the location of an incident, thereby facilitating a swift remedial response. Research on developing a smart security system is based on Artificial Intelligence with an unmanned aerial vehicle (UAV) to detect and monitor alert situations, such as fire accidents and theft/intruders in the building or factory, which is based on the Internet of Things (IoT) network. The system includes a Passive Pyroelectric Infrared Detector for human detection and an analog flame sensor to sense the appearance of the concerned objects and then transmit the signal to the workstation via Wi-Fi based on the microcontroller Espressif32 (Esp32). The computer vision models YOLOv8 (You Only Look Once version 8) and Cascade Classifier are trained and implemented into the workstation, which is able to identify people, some potentially dangerous objects, and fire. The drone is also controlled by three algorithms—distance maintenance, automatic yaw rotation, and potentially dangerous object avoidance—with the support of a proportional–integral–derivative (PID) controller. The Smart Drone Surveillance System has good commands for automatic tracking and streaming of the video of these specific circumstances and then transferring the data to the involved parties such as security or staff.

**2.2TITLE:** Human Detection and Action Recognition for Search and Rescue in Disasters Using YOLOv3 Algorithm

**AUTHOR:** Raid Al-Nima

**YEAR:** 2023

**DESCRIPTION:**

Drone examination has been overall quickly embraced by NDMM (natural disaster mitigation and management) division to survey the state of impacted regions. Manual video analysis by human observers takes time and is subject to mistakes. The human identification examination of pictures caught by drones will give a practical method for saving lives who are being trapped under debris during quakes or in floods and so on. Drone investigation for research and security and search and rescue (SAR) should involve the drone to filter the impacted area using a camera and a model of unmanned area vehicles (UAVs) to identify specific locations where assistance is required. The existing methods (Balmukund et al. 2020) used were faster-region based convolutional neural networks (F-RCNNs), single shot detector (SSD), and region-based fully convolutional network (R-FCN) for the detection of human and recognition of action. Some of the existing methods used 700 images with six classes only, whereas the proposed model uses 1996 images with eight classes. The proposed model is used YOLOv3 (you only look once) algorithm for the detection and recognition of actions. In this study, we provide the fundamental ideas underlying an object detection model. To find the most effective model for human recognition and detection, we trained the YOLOv3 algorithm on the image dataset and evaluated its performance. We compared the outcomes with the existing algorithms like F-RCNN, SSD, and R-FCN. The accuracies of F-RCNN, SSD, R-FCN (existing algorithms), and YOLOv3 (proposed algorithm) are 53%, 73%, 93%, and 94.9%, respectively. Among these algorithms, the YOLOv3 algorithm gives the highest accuracy of 94.9%. The proposed work shows that existing models are inadequate for critical applications like search and rescue, which convinces us to propose a model raised by a pyramidal component extracting SSD in human localization and action recognition. The suggested model is 94.9% accurate when applied to the proposed dataset, which is an important contribution. Likewise, the suggested model succeeds in helping time for expectation in examination with the cutting-edge identification models with existing strategies. The average time taken by our proposed technique to distinguish a picture is 0.40 milisec which is a lot better than the existing method. The proposed model can likewise distinguish video and can be utilized for real-time recognition. The SSD model can likewise use to anticipate messages if present in the picture.

**2.3TITLE**: Ai Based Human Detecting Robot for Earthquake Rescue Operation

**AUTHOR:** D.Navya Narayana Kumari, Pappala David, Nallabothula Sai Krishna,

**YEAR:** 2024

**DESCRIPTION:**

Natural disasters, such as earthquakes, can have catastrophic effects on human lives and infrastructure. Earthquake rescue operations are challenging, often requiring rapid response to locate and rescue survivors trapped in the rubble. Detection by rescue workers becomes time consuming and due to the vast area, that gets affected it becomes more difficult. Hence a lot of times humans are buried among the debris, and it become impossible to detect them. A timely rescue can only save the people who are buried and wounded. Detection by rescue workers becomes time consuming and due to the vast area, that gets affected it becomes more difficult. So, the project proposes an autonomous robotic vehicle that moves in the earthquake prone area and helps in identifying the alive people and rescue operations. This project abstract outlines the development of an Artificial Intelligence-Based Human-Detecting Robot designed to assist in earthquake rescue missions. Key components of this project include machine learning, and Embedded design, integrated into a single platform to detect and locate humans trapped under debris during an earthquake. In this based live human detecting robot for earthquake rescue operation project, a new method for detecting surviving humans in destructed environments using simulated autonomous robot is proposed.

**2.4TITLE:** Human Detection Robot for Disaster Management

**AUTHOR:** Hrithik Sivadasan; Pooja Jiresh; Sucheta G S

**YEAR:** 2021

**DESCRIPTION:**

A unique passive Infrared sensor is used in our design that receives infrared rays that are emitted from humans. When a human body emits infrared radiation of micron wavelength it will be received and manipulated by the PIR (Passive infrared sensor). Once a human target is located the system has to give an alert which may help to identify and localize the victim location as soon as possible. The major part of circuit design is the ‘Human detection module’ which will be used for carrying out the search activity. The aim of this project is to provide a prototype of practical design to build a simplified version of a Human detection robot which has to be implemented during calamities to find the casualties. Humans can be used for rescuing people in these areas, but due to high risk of earthquakes and building collapses it is not possible to send human rescue teams in these areas. Thus an affordable high technology equipment which makes this risky job quicker and safer is needed for the hour, which has been described in this paper. It is a simple, yet efficient equipment to indicate casualties and help them with immediate access to first aid. The Arduino monitors PIR sensor and the program is written and uploaded to Arduino. Whenever a human detection occurs within the range of PIR sensor, the LED switches on and buzzer rings which can be seen by the team. If the PIR sensor generates false alerts, this will also be monitored by the Arduino and considered by the operation team. The team will move the robot in the vicinity, stop it and carry on a human search.

**2.5TITLE:** DISASTER: Dedicated Intelligent Security Attacks on Sensor-triggered Emergency Responses

**AUTHOR:** Arsalan Mosenia,Susmita Sur-Kolay,

**YEAR:** 2017

**DESCRIPTION:**

Rapid technological advances in microelectronics, networking, and computer science have resulted in an exponential increase in the number of cyber-physical systems (CPSs) that enable numerous services in various application domains, e.g., smart homes and smart grids. Moreover, the emergence of the Internet-of-Things (IoT) paradigm has led to the pervasive use of IoT-enabled CPSs in our everyday lives. Unfortunately, as a side effect, the number of potential threats and feasible security attacks against CPSs has grown significantly. In this paper, we introduce a new class of attacks against CPSs, called dedicated intelligent security attacks against sensor-triggered emergency responses (DISASTER). DISASTER targets safety mechanisms deployed in automation/monitoring CPSs and exploits design flaws and security weaknesses of such mechanisms to trigger emergency responses even in the absence of a real emergency. Launching DISASTER can lead to serious consequences for three main reasons. First, almost all CPSs offer specific emergency responses and, as a result, are potentially susceptible to such attacks. Second, DISASTER can be easily designed to target a large number of CPSs, e.g., the anti-theft systems of all buildings in a residential community. Third, the widespread deployment of insecure sensors in already-in-use safety mechanisms along with the endless variety of CPS-based applications magnifies the impact of launching DISASTER. In addition to introducing DISASTER, we describe the serious consequences of such attacks. We demonstrate the feasibility of launching DISASTER against the two most widely-used CPSs: residential and industrial automation/monitoring systems. Moreover, we suggest several countermeasures that can potentially prevent DISASTER and discuss their advantages and drawbacks.

**2.6 TITLE:** From Sensors to Safety: Internet of Emergency Services (IoES) for Emergency Response and Disaster Management

**AUTHOR:** Robertas Damaševičius; Nebojsa Bacanin; andSanjay Misra

**YEAR: 2023**

**DESCRIPTION:**

The advancement in technology has led to the integration of internet-connected devices and systems into emergency management and response, known as the Internet of Emergency Services (IoES). This integration has the potential to revolutionize the way in which emergency services are provided, by allowing for real-time data collection and analysis, and improving coordination among various agencies involved in emergency response. This paper aims to explore the use of IoES in emergency response and disaster management, with an emphasis on the role of sensors and IoT devices in providing real-time information to emergency responders. We will also examine the challenges and opportunities associated with the implementation of IoES, and discuss the potential impact of this technology on public safety and crisis management. The integration of IoES into emergency management holds great promise for improving the speed and efficiency of emergency response, as well as enhancing the overall safety and well-being of citizens in emergency situations. However, it is important to understand the possible limitations and potential risks associated with this technology, in order to ensure its effective and responsible use. This paper aims to provide a comprehensive understanding of the Internet of Emergency Services and its implications for emergency response and disaster management.

**2.7 TITLE**: Potential Use of Artificial Intelligence (AI) in Disaster Risk and Emergency Health Management: A Critical Appraisal on Environmental Health

**AUTHOR:** Lazima Faiah Bari, Iftekhar Ahmed, Rayhan Ahamed,

**YEAR:** 2023

**DESCRIPTION:**

The risk evaluation of natural disasters is an obstacle to ensuring healthcare services during catastrophic events worldwide. Therefore, timely and appropriate environmental health risk evaluation is essential. In this study, we incorporated the information from databases such as PubMed, Google Scholar, and Scopus. We performed this study to explore the feasibility of using artificial intelligence (AI) in disaster risk and emergency health management. Natural disasters have some phenomenon that is bound to happen. So, we can use AI to inform healthcare authorities about environmental health risks and emergency medical management. The recent innovations in technology have created novel opportunities for improving healthcare services across the world. AI technology would be the backbone of the fourth industrial revolution because we are entering an AI-based world this decade. Therefore, we suggest the authorities consider this positive aspect of AI technology to minimize the disastrous outcomes of any catastrophic events.

**2.8 TITLE:** A Novel Approach for Disaster Victim Detection Under Debris Environments Using Decision Tree Algorithms With Deep Learning Features

**AUTHOR:** G. SEEJA , AROCKIA SELVAKUMAR AROCKIA DOSS

**YEAR: 2023**

**DESCRIPTION:**

Search and Rescue operations for victim identification in an unstructured collapsed building are high-risk and time-consuming. The possibility of saving a victim is high only during the first 48 hours, and then the prospect tends to zero. The faster the response and identification, the sooner the victim can be taken to medical assistance. Combining mobile robots with practical Artificial Intelligence (AI) driven Human Victim Detection (HVD) systems managed by professional teams can considerably reduce this problem. In this paper, we have developed a Transfer Learning-based Deep Learning approach to identify human victims under collapsed building environments by integrating machine learning classification algorithms. A custom-made human victim dataset was created with five class labels: head, hand, leg, upper body, and without the body. First, we extracted the class-wise features of the dataset using fine-tuning-based transfer learning on ResNet-50 deep learning model. The learned features of the model were then extracted, and then a feature selection was performed using J48 to study the impact of feature reduction in classification. Several decision tree algorithms, including decision stump, hoeffiding tree, J48, Linear Model Tree (LMT), Random Forest, Random Tree, Representative (REP) Tree, J48 graft, and other famous algorithms like LibSVM, Logistic regression, Multilayer perceptron, BayesNet, Naive Bayes are then used to perform the classification. The classification accuracy of the abovementioned algorithms is compared to recommend the optimal approach for real-time use. The random tree approach outperformed all other tree-based algorithms with a maximum classification accuracy of 99.53% and a computation time of 0.02 seconds.

**2.9 TITLE:** Indoor Human Detection from a Building’s Exterior Using 433 MHz Wireless Transceivers

**AUTHOR:** Sunghoon Jo,Sehee Park and Gu-In Kwon

**YEAR:** 2023

**DESCRIPTION:**

This study introduces a novel system for detecting humans inside a building by utilizing RF signals from the building’s exterior. Existing RF communication devices encounter signal attenuation issues when passing through walls, limiting their effectiveness. In contrast, our system employs a low-power, long-distance communication signal operating at 433 MHz to enhance signal permeability, enabling the accurate detection of individuals within the building. The system analyzes received signal strength indicator (RSSI) data using variance and mean analysis algorithms to determine the presence or absence of people. The evaluation results indicate promising average accuracies of 88% for the variance analysis algorithm and 97.7% for the mean analysis algorithm. The proposed system holds potential for real-world deployment, particularly in challenging scenarios such as fire incidents, where pre-installation is challenging. Continued research and development efforts aim to enhance the system’s performance and address any limitations, making it more effective and robust in various practical applications.

**2.10 TITLE:** Deep Learning for Earthquake Disaster Assessment: Objects, Data, Models, Stages, Challenges, and Opportunities

**AUTHOR:** Jing Jia and Wenjie Ye

**YEAR: 2023**

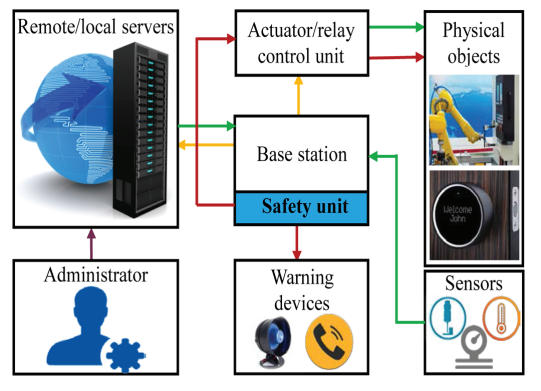
**DESCRIPTION:**

Earthquake Disaster Assessment (EDA) plays a critical role in earthquake disaster prevention, evacuation, and rescue efforts. Deep learning (DL), which boasts advantages in image processing, signal recognition, and object detection, has facilitated scientific research in EDA. This paper analyses 204 articles through a systematic literature review to investigate the status quo, development, and challenges of DL for EDA. The paper first examines the distribution characteristics and trends of the two categories of EDA assessment objects, including earthquakes and secondary disasters as disaster objects, buildings, infrastructure, and areas as physical objects. Next, this study analyses the application distribution, advantages, and disadvantages of the three types of data (remote sensing data, seismic data, and social media data) mainly involved in these studies. Furthermore, the review identifies the characteristics and application of six commonly used DL models in EDA, including convolutional neural network (CNN), multi-layer perceptron (MLP), recurrent neural network (RNN), generative adversarial network (GAN), transfer learning (TL), and hybrid models. The paper also systematically details the application of DL for EDA at different times (i.e., pre-earthquake stage, during-earthquake stage, post-earthquake stage, and multi-stage). We find that the most extensive research in this field involves using CNNs for image classification to detect and assess building damage resulting from earthquakes. Finally, the paper discusses challenges related to training data and DL models, and identifies opportunities in new data sources, multimodal DL, and new concepts. This review provides valuable references for scholars and practitioners in related fields.

**CHAPTER 3**

**EXISTING SYSTEM**

The existing paper discusses the rapid growth of cyber-physical systems (CPSs) due to technological advances, alongside the emergence of security threats, particularly against sensor-triggered emergency responses. It introduces a new class of attacks, called Dedicated Intelligent Security Attacks against Sensor-triggered Emergency Responses (DISASTER), which exploit design flaws and security weaknesses to trigger emergency responses without a real emergency. The consequences of DISASTER attacks are severe due to the widespread deployment of CPSs, their susceptibility to such attacks, and the impact on safety mechanisms. The paper also highlights the vulnerability of residential and industrial automation/monitoring systems to DISASTER attacks and proposes countermeasures to mitigate them.

****

**DISADVANTAGE**

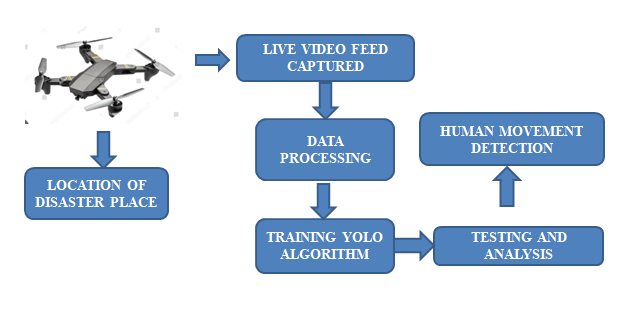
* The paper proposes several countermeasures to prevent DISASTER attacks, it also acknowledges their limitations and potential drawbacks, indicating that existing solutions may not fully mitigate the risks.
* Launching DISASTER attacks can lead to serious consequences, such as triggering unnecessary emergency responses, which can disrupt normal operations and cause panic or confusion among users.

**CHAPTER 4**

**PROPOSED SYSTEM**

The proposed system comprises an AI-enabled drone equipped with a camera and onboard processing capabilities. This integration allows the drone to capture live footage as it navigates disaster-stricken areas. The key component of the system is the You Only Look Once (YOLO) algorithm, which facilitates real-time object detection. By leveraging YOLO, the drone can analyze the video stream in real-time, enabling it to detect and track human presence amidst rubble, debris, or hazardous environments with high accuracy and efficiency. As the drone moves through the disaster area, the camera captures continuous footage, which is then processed by the YOLO algorithm onboard. This real-time processing capability enables the system to swiftly identify human movements in the environment. The algorithm's ability to process the video stream quickly ensures timely detection of survivors, providing crucial information to first responders for swift intervention and rescue operations. By integrating the YOLO algorithm into the drone's onboard processing system, the proposed solution offers a cost-effective and agile approach to disaster response. This system can rapidly scan large areas and identify human presence without relying on external processing or communication systems. The integration of AI and drone technology in this system enhances the efficiency and effectiveness of search and rescue operations, potentially saving lives and minimizing casualties in disaster situations.

**BLOCK DIAGRAM**

****

**MODULES DESCRIPTION**

**1. Camera:**

The camera module is responsible for capturing live video feed as the drone navigates disaster-stricken areas. It ensures that real-time footage of the environment is available for analysis by the system.

**2. Live Video Feed Captured:**

This module handles the processing of the live video feed captured by the camera. It ensures that the video feed is available for further analysis by other modules in the system.

**3. Data Processing:**

The data processing module is responsible for preprocessing the live video feed before it is input into the YOLO algorithm. It may involve tasks such as resizing, normalization, or other preprocessing techniques to optimize the data for object detection.

**4. Training YOLO Algorithm:**

This module involves training the YOLO algorithm on labeled datasets to enable it to accurately detect and track human movements in the disaster environment. Training involves optimizing the model's parameters and fine-tuning its architecture to achieve high accuracy.

**5. Testing and Analysis:**

The testing and analysis module evaluates the performance of the entire system in simulated disaster scenarios. It involves conducting tests to assess the accuracy, speed, and reliability of the system in detecting and tracking human movements. Analysis of the test results helps identify areas for improvement and optimization.

**6. Human Movement Detection:**

This module utilizes the trained YOLO algorithm to detect and track human movements in real-time. It processes the preprocessed video feed and identifies human presence amidst rubble, debris, or hazardous environments.

**7. Location of Disaster Place:**

The location of the disaster place module utilizes the information gathered from human movement detection to pinpoint the location of survivors. It provides spatial coordinates or a map of the disaster area to guide first responders to the precise location of survivors for swift intervention and rescue operations.

**ADVANTAGES**

* Rapid detection and identification of human movements in disaster zones.
* Real-time monitoring capabilities enable immediate response and rescue efforts.
* Enhanced situational awareness for first responders, leading to more effective decision-making.
* Cost-effective solution compared to traditional search and rescue methods.
* Minimizes risk to rescue personnel by providing remote reconnaissance capabilities

**APPLICATIONS**

* Disaster Assessment
* Wildlife Conservation
* Precision Agriculture

**CHAPTER 6**

**MODULES DESCRIPTION**

**CAMERA**The camera module serves as the primary sensory input for the AI-enabled drone system designed for disaster response. Its responsibility lies in capturing live video feed as the drone traverses through disaster-stricken areas. Equipped with advanced imaging technology, the camera continuously records the surrounding environment, providing a real-time visual stream of the disaster zone. This stream is pivotal for the subsequent analysis carried out by the system to detect and track human movements amidst the rubble, debris, or hazardous conditions. During operation, the camera module must withstand various environmental challenges inherent to disaster scenarios. It needs to function reliably amidst unstable structures, adverse weather conditions, and potentially hazardous materials. Therefore, the camera must be rugged and resilient to ensure uninterrupted data capture even in challenging situations. Additionally, the camera's optics and resolution should be optimized to provide clear and detailed imagery, facilitating accurate analysis by the system's algorithms. Furthermore, the camera module must be designed for flexibility and adaptability to different disaster environments. This may involve adjustable parameters such as focus, exposure, and frame rate to optimize image quality based on the specific conditions encountered by the drone. In dynamic and unpredictable disaster scenarios, the camera module plays a crucial role in providing real-time situational awareness to guide search and rescue efforts effectively.

**LIVE VIDEO FEED CAPTURED**

The Live Video Feed Captured module is a critical component of the AI-enabled drone system, responsible for managing the incoming stream of video data captured by the drone's camera. Its primary function is to process this live feed and ensure that it is available for further analysis by other modules within the system. Upon receiving the video stream from the camera module, this module carries out initial processing tasks such as buffering, frame synchronization, and format conversion to prepare the data for analysis. Once the live video feed is captured and processed, it is made available to other modules in the system for real-time analysis. This includes modules responsible for tasks such as object detection, human movement tracking, and location identification. By efficiently managing the flow of video data, this module ensures that the system can operate seamlessly, providing timely and accurate information to support disaster response efforts. Furthermore, the Live Video Feed Captured module may incorporate features to optimize the performance of the system, such as data compression techniques to reduce bandwidth requirements or quality-of-service mechanisms to prioritize critical video streams. These enhancements help maximize the efficiency of data transmission and analysis, enabling the system to respond rapidly to changing conditions in disaster-stricken areas. The module plays a vital role in facilitating real-time situational awareness and enhancing the effectiveness of the AI-enabled drone system in disaster response operations.

**DATA PROCESSING**

The Data Processing module is a crucial component of the AI-enabled drone system, responsible for preparing the live video feed captured by the camera for analysis by the You Only Look Once (YOLO) algorithm. Its primary role is to preprocess the video data to optimize it for object detection. This preprocessing may involve several tasks such as resizing, normalization, or other techniques to enhance the quality and suitability of the data for input into the YOLO algorithm. One of the key tasks performed by the Data Processing module is resizing the video frames to a standard size suitable for processing by the YOLO algorithm. This ensures consistency in the input data and allows the algorithm to operate efficiently. Additionally, normalization techniques may be applied to adjust the pixel values of the video frames to a standardized scale, further improving the algorithm's performance. Furthermore, the Data Processing module may employ other preprocessing techniques such as noise reduction, contrast enhancement, or edge detection to improve the clarity and accuracy of the video data. These techniques help to reduce the impact of environmental factors such as poor lighting or visual clutter, which can hinder the YOLO algorithm's ability to detect objects accurately in disaster scenarios. By performing these preprocessing tasks, the Data Processing module ensures that the video data is optimized and ready for input into the YOLO algorithm, enabling accurate and efficient object detection. This allows the AI-enabled drone system to effectively identify and track human movements amidst rubble, debris, or hazardous environments, providing valuable information to support disaster response efforts.

**TRAINING YOLO ALGORITHM**

The Training YOLO Algorithm module is a critical stage in the development of the AI-enabled drone system, as it focuses on enhancing the algorithm's ability to accurately detect and track human movements in disaster environments. This module involves training the YOLO algorithm on labeled datasets, which provide examples of human presence amidst rubble, debris, or hazardous conditions. During training, the algorithm learns to recognize patterns and features that distinguish humans from other objects or background elements in the video feed. Training the YOLO algorithm requires careful optimization of its parameters and architecture to achieve high accuracy in human movement detection. This process involves adjusting parameters such as learning rate, batch size, and regularization techniques to fine-tune the algorithm's performance. Additionally, the architecture of the YOLO model may be modified or customized to better suit the specific challenges of disaster scenarios, such as varying lighting conditions, cluttered environments, or occlusions. Moreover, the training process involves iteratively feeding labeled data into the algorithm, allowing it to learn from examples and improve its ability to accurately detect human movements over time. This iterative approach enables the algorithm to adapt to different disaster scenarios and improve its performance through experience. Continuous monitoring and evaluation of the algorithm's performance during training are essential to ensure that it achieves the desired level of accuracy and reliability for real-world deployment. The Training YOLO Algorithm module plays a crucial role in enhancing the AI-enabled drone system's capabilities for disaster response. By training the YOLO algorithm on labeled datasets and fine-tuning its parameters and architecture, this module enables the system to accurately detect and track human movements in real-time, providing valuable information to support search and rescue efforts in disaster environments.

**TESTING AND ANALYSIS**

The Testing and Analysis module is essential for ensuring the effectiveness and reliability of the AI-enabled drone system in disaster response scenarios. This module evaluates the performance of the entire system through rigorous testing conducted in simulated disaster environments. The tests are designed to assess various aspects such as accuracy, speed, and reliability in detecting and tracking human movements amidst rubble, debris, or hazardous conditions. During testing, the system is subjected to a range of simulated disaster scenarios to evaluate its performance under different conditions. These scenarios may include variations in lighting, weather, terrain, and the presence of obstacles or occlusions. By simulating real-world conditions, the testing process provides valuable insights into the system's ability to operate effectively in diverse disaster environments. After conducting tests, the module performs detailed analysis of the results to identify areas for improvement and optimization. This analysis involves evaluating metrics such as detection accuracy, false positives, processing speed, and overall system reliability. By analyzing these metrics, the module can pinpoint any weaknesses or limitations in the system's performance and develop strategies to address them.

**HUMAN MOVEMENT DETECTION**

The Human Movement Detection module is a pivotal component of the AI-enabled drone system, responsible for utilizing the trained YOLO algorithm to detect and track human movements in real-time. Building upon the preprocessing done by earlier modules, this module takes the preprocessed video feed and applies the trained YOLO algorithm to identify human presence amidst rubble, debris, or hazardous environments. In real-time operation, this module processes each frame of the video feed, leveraging the YOLO algorithm's object detection capabilities to identify and track human subjects. It analyzes the visual data for specific patterns and features associated with human movements, allowing the system to accurately detect and monitor the presence of survivors in disaster areas. By continuously analyzing the video feed, the Human Movement Detection module provides valuable information to first responders about the location and movements of survivors, aiding in timely rescue efforts. The module's ability to operate in real-time ensures that critical information is available promptly, enabling responders to prioritize areas for intervention and allocate resources effectively.

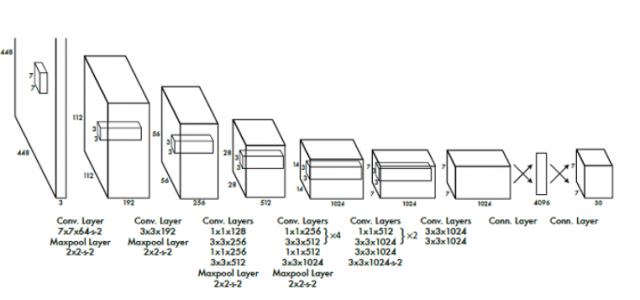
**LOCATION OF DISASTER PLACE**

The Location of Disaster Place module is a critical component of the AI-enabled drone system, responsible for utilizing the information gathered from human movement detection to pinpoint the exact location of survivors within the disaster area. Drawing on data collected by the Human Movement Detection module, this module provides spatial coordinates or a map of the disaster site, guiding first responders to the precise locations of survivors for swift intervention and rescue operations. By integrating data from human movement detection with geographical information systems (GIS) or global positioning system (GPS) technology, this module accurately determines the spatial coordinates of detected human presence. It generates a real-time map of the disaster area, marking the locations where survivors have been detected, and overlays this information onto digital maps accessible to first responders. This real-time mapping capability is invaluable for guiding rescue teams to the precise locations of survivors, especially in complex and chaotic disaster environments. By providing accurate spatial information, the module enables responders to navigate the disaster site efficiently, reducing response times and increasing the chances of successful rescues. The Location of Disaster Place module enhances the effectiveness of the AI-enabled drone system by enabling responders to quickly and accurately locate survivors within disaster areas. By facilitating swift intervention and rescue operations, this module plays a crucial role in saving lives and minimizing casualties during emergency situations.

**CHAPTER 6**

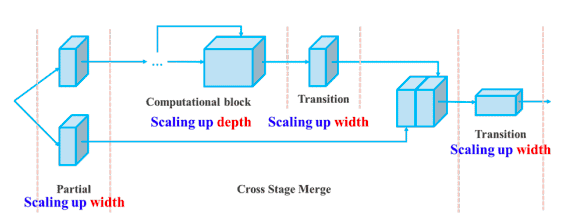
**YOLO ALGORITHM**

YOLO is an algorithm that uses neural networks to provide real time object detection. Object detection is a phenomenon in computer vision that involves the detection of various objects in digital images or videos. YOLO is an algorithm that detects and recognizes various objects in a picture. The YOLO framework (You Only Look Once) on the other hand, deals with object detection in a different way. It takes the entire image in a single instance and predicts the bounding box coordinates and class probabilities for these boxes. The biggest advantage of using YOLO is its superb speed – it’s incredibly fast and can process 45 frames per second. YOLO also understands generalized object representation. This is one of the best algorithms for object detection and has shown a comparatively similar performance to the R-CNN algorithms. In the upcoming sections, we will learn about different techniques used in YOLO algorithm. Object detection is one of the classical problems in computer vision where you work to recognize what and where — specifically what objects are inside a given image and also where they are in the image. The problem of object detection is more complex than classification, which also can recognize objects but doesn’t indicate where the object is located in the image. In addition, classification doesn’t work on images containing more than one object. YOLO uses a totally different approach. YOLO is a clever convolutional neural network (CNN) for doing object detection in real-time. The algorithm applies a single neural network to the full image, and then divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.



YOLO is popular because it achieves high accuracy while also being able to run in real-time. The algorithm “only looks once” at the image in the sense that it requires only one forward propagation pass through the neural network to make predictions. After non-max suppression (which makes sure the object detection algorithm only detects each object once), it then outputs recognized objects together with the bounding boxes. With YOLO, a single CNN simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This model has a number of benefits over other object detection methods: YOLO is extremely fast YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance. YOLO learns generalizable representations of objects so that when trained on natural images and tested on artwork, the algorithm outperforms other top detection methods.You Only Look Once (YOLO) is a network that uses Deep Learning (DL) algorithms for object detection. YOLO performs object detection by classifying certain objects within the image and determining where they are located on it. For example, if you input an image of a herd of sheep into a YOLO network, it will generate an output of a vector of bounding boxes for each individual sheep and classify it as such.

**HOW YOLO IMPROVES OVER PREVIOUS OBJECT DETECTION METHODS-**



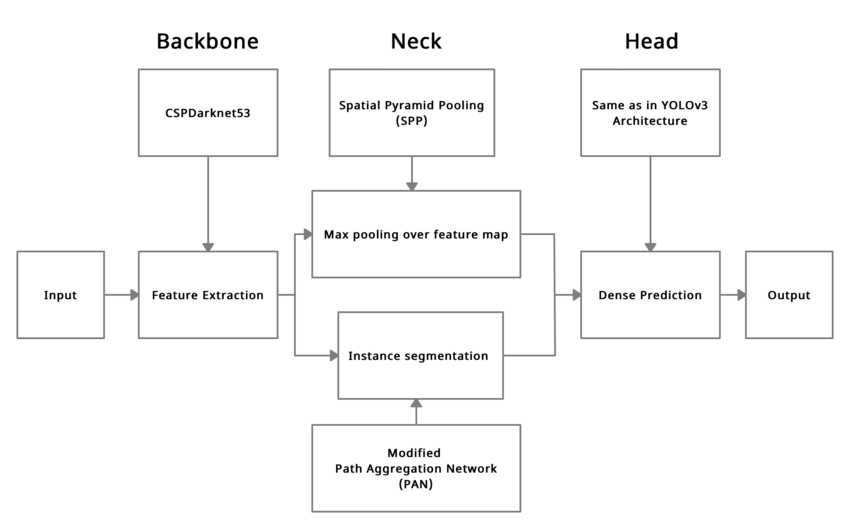
Previous object detection methods like Region-Convolution Neural Networks (R-CNN), including other variations of it like [fast R-CNN](https://missinglink.ai/guides/deep-learning-frameworks/building-faster-r-cnn-on-tensorflow-introduction-and-examples/), performed object detection tasks in a pipeline of multi-step series. R-CNN focuses on a specific region within the image and trains each individual component separately.

This process requires the R-CNN to classify 2000 regions per image, which makes it very time-consuming (47 seconds per individual test image). Thus it, cannot be implemented in real-time. Additionally, R-CNN uses a fixed selective algorithm, which means no learning process occurs during this stage so the network might generate an inferior region proposal.

This makes object detection networks such as R-CNN harder to optimize and slower compared to YOLO. YOLO is much faster (45 frames per second) and easier to optimize than previous algorithms, as it is based on an algorithm that uses only one neural network to run all components of the task.

To gain a better understanding of what YOLO is, we first have to explore its architecture and algorithm.

## YOLO Architecture -Structure Design and Algorithm Operation



A YOLO network consists of three main parts. First, the algorithm, also known as the predictions vector. Second, the network. Third, the loss functions.

**The YOLO Algorithm**

Once you insert input an image into a YOLO algorithm, it splits the images into an SxS grid that it uses to predict whether the specific bounding box contains the object (or parts of it) and then uses this information to predict a class for the object.

  Before we can go into details and explain how the algorithm functions, we need to understand how the algorithm builds and specifies each bounding box. The YOLO algorithm uses four components and additional value to predict an output.

1. The center of a bounding box (**bx by**)
2. Width (**bw**)
3. Height (**bh**)
4. The Class of the object (**c**)

The final predicted value is confidence (pc). It represents the probability of the existence of an object within the bounding box.The (x,y) coordinates represent the center of the bounding box.Typically, most of the bounding boxes will not contain an object, so we need to use the pc prediction. We can use a process called non-max suppression to remove unnecessary boxes with low probability to contain objects and those who share big areas with other boxes.

**THE NETWORK**

A YOLO network is structured like a regular CNN; it contains convolution and max-pooling layers and then two fully connected CNN layers. The Loss Function We only want one of the bounding boxes to be responsible for the object within the image since the YOLO algorithm predicts multiple bounding boxes for each grid cell. To achieve this, we use the loss function to compute the loss for each true positive. To make the loss function more efficient, we need to select the bounding box with the highest Intersection over Union (IoU) with the ground truth. This method improves predictions by making specialized bounding boxes which improves the predictions for some aspect ratios and sizes.

**YOLO V3**

YOLO V3 is an incremental upgrade over YOLO V2, which uses another variant of Dark net. This YOLO V3 architecture consists of 53 layers trained on Image net and another 53 tasked with object detection which amounts to 106 layers. While this has dramatically improved the accuracy of the network, it has also reduced the speed from 45 fps to 30 fps.

**CHAPTER 7**

**SYSTEM REQUIREMENT**

**H/W System Configuration:**

* Processor - Pentium –IV
* RAM - 4 GB (min)
* Hard Disk - 20 GB

**S/W System Configuration:**

* Operating System : Windows 7 or 8
* Front End : html,css
* Back End : python

**SOFTWARE ENVIRONMENT**

**Python Technology:**

**Python** is an interpreter, high-level, general-purpose programming language. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. **Python** is often described as a "batteries included" language due to its comprehensive standard library.

**Python Programing Language:**

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 (including by Meta programming and met objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

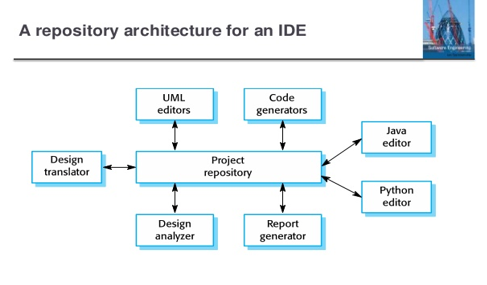
Python packages with a wide range of functionality, including:

* Easy to Learn and Use
* Expressive Language
* Interpreted Language
* Cross-platform Language
* Free and Open Source
* Object-Oriented Language
* Extensible
* Large Standard Library
* GUI Programming Support
* Integrated

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.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal.

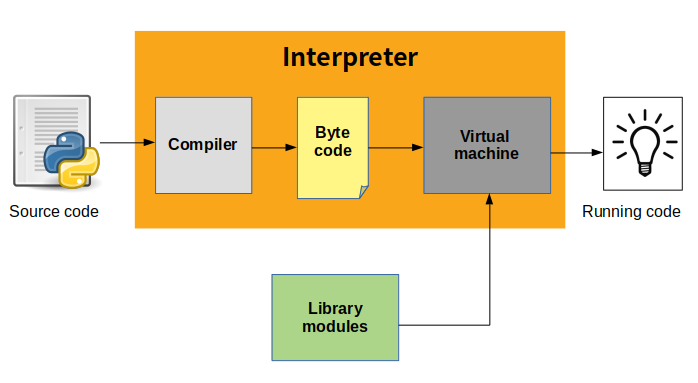


Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one and preferably only one obvious way to do it" design philosophy. Alex Martelli, a Fellow at the Python Software Foundation and Python book author, writes that "To describe something as 'clever' is not considered a compliment in the Python culture."

Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the Python reference implementation that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Python is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name a tribute to the British comedy group Monty Python and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.

Python uses duck typing and has typed objects but untyped variable names. Type constraints are not checked at compile time; rather, operations on an object may fail, signifying that the given object is not of a suitable type. Despite being dynamically typed, Python is strongly typed, forbidding operations that are not well-defined (for example, adding a number to a string) rather than silently attempting to make sense of them.

**The Python Platform:**

The platform module in Python is used to access the underlying platform's data, such as, hardware, operating system, and interpreter version information. The platform module includes tools to see the platform's hardware, operating system, and interpreter version information where the program is running.

There are four functions for getting information about the current Python interpreter. python\_version() and python\_version\_tuple() return different forms of the interpreter version with major, minor, and patch level components. python\_compiler() reports on the compiler used to build the interpreter. And python\_build() gives a version string for the build of the interpreter.

Platform() returns string containing a general purpose platform identifier. The function accepts two optional Boolean arguments. If aliased is true, the names in the return value are converted from a formal name to their more common form. When terse is true, returns a minimal value with some parts dropped.

**What does python technology do?**

Python is quite popular among programmers, but the practice shows that business owners are also Python development believers and for good reason. Software developers love it for its straightforward syntax and reputation as one of the easiest programming languages to learn. Business owners or CTOs appreciate the fact that there’s a framework for pretty much anything – from web apps to machine learning.

Moreover, it is not just a language but more a technology platform that has come together through a gigantic collaboration from thousands of individual professional developers forming a huge and peculiar community of aficionados.

So what are the tangible benefits the language brings to those who decided to use it as a core technology? Below you will find just some of those reasons.

**PRODUCTIVITY AND SPEED**

It is a widespread theory within development circles that developing Python applications is approximately up to 10 times faster than developing the same application in Java or C/C++. The impressive benefit in terms of time saving can be explained by the clean object-oriented design, enhanced process control capabilities, and strong integration and text processing capacities. Moreover, its own unit testing framework contributes substantially to its speed and productivity.

**PYTHON IS POPULAR FOR WEB APPS**

Web development shows no signs of slowing down, so technologies for rapid and productive web development still prevail within the market. Along with JavaScript and Ruby, Python, with its most popular web framework Django, has great support for building web apps and is rather popular within the web development community.

**OPEN-SOURCE AND FRIENDLY COMMUNITY**

As stated on the official website, it is developed under an OSI-approved open source license, making it freely usable and distributable. Additionally, the development is driven by the community, actively participating and organizing conference, meet-ups, hackathons, etc. fostering friendliness and knowledge-sharing.

**PYTHON IS QUICK TO LEARN**

It is said that the language is relatively simple so you can get pretty quick results without actually wasting too much time on constant improvements and digging into the complex engineering insights of the technology. Even though Python programmers are really in high demand these days, its friendliness and attractiveness only help to increase number of those eager to master this programming language.

**BROAD APPLICATION**

It is used for the broadest spectrum of activities and applications for nearly all possible industries. It ranges from simple automation tasks to gaming, web development, and even complex enterprise systems. These are the areas where this technology is still the king with no or little competence:

* Machine learning as it has a plethora of libraries implementing machine learning algorithms.
* Web development as it provides back end for a website or an app.
* Cloud computing as Python is also known to be among one of the most popular cloud-enabled languages even used by Google in numerous enterprise-level software apps.
* Scripting.
* Desktop GUI applications.

**Python compiler**

The Python compiler package is a tool for analyzing Python source code and generating Python bytecode. The compiler contains libraries to generate an abstract syntax tree from Python source code and to generate Python bytecode from the tree.

The compiler package is a Python source to bytecode translator written in Python. It uses the built-in parser and standard parser module to generate a concrete syntax tree. This tree is used to generate an abstract syntax tree (AST) and then Python bytecode.

The full functionality of the package duplicates the built-in compiler provided with the Python interpreter. It is intended to match its behavior almost exactly. Why implement another compiler that does the same thing? The package is useful for a variety of purposes. It can be modified more easily than the built-in compiler. The AST it generates is useful for analyzing Python source code.

**The basic interface**

The top-level of the package defines four functions. If you import compiler, you will get these functions and a collection of modules contained in the package.

**compiler.parse(buf)**

Returns an abstract syntax tree for the Python source code in buf. The function raises Syntax Error if there is an error in the source code. The return value is a compiler.ast. Module instance that contains the tree.

**compiler.parseFile(path)**

Return an abstract syntax tree for the Python source code in the file specified by path. It is equivalent to parse(open(path).read()).

**LIMITATIONS**

There are some problems with the error checking of the compiler package. The interpreter detects syntax errors in two distinct phases. One set of errors is detected by the interpreter’s parser, the other set by the compiler. The compiler package relies on the interpreter’s parser, so it get the first phases of error checking for free. It implements the second phase itself, and that implementation is incomplete. For example, the compiler package does not raise an error if a name appears more than once in an argument list: def f(x, x): ...

A future version of the compiler should fix these problems.

**PYTHON ABSTRACT SYNTAX**

The compiler.ast module defines an abstract syntax for Python. In the abstract syntax tree, each node represents a syntactic construct. The root of the tree is Module object.

The abstract syntax offers a higher level interface to parsed Python source code. The parser module and the compiler written in C for the Python interpreter use a concrete syntax tree. The concrete syntax is tied closely to the grammar description used for the Python parser. Instead of a single node for a construct, there are often several levels of nested nodes that are introduced by Python’s precedence rules.

The abstract syntax tree is created by the compiler.transformer module. The transformer relies on the built-in Python parser to generate a concrete syntax tree. It generates an abstract syntax tree from the concrete tree.

The transformer module was created by Greg Stein and Bill Tutt for an experimental Python-to-C compiler. The current version contains a number of modifications and improvements, but the basic form of the abstract syntax and of the transformer are due to Stein and Tutt.

**AST NODES**

The compiler.ast module is generated from a text file that describes each node type and its elements. Each node type is represented as a class that inherits from the abstract base class compiler.ast.Node and defines a set of named attributes for child nodes.

classcompiler.ast.Node

The Node instances are created automatically by the parser generator. The recommended interface for specific Node instances is to use the public attributes to access child nodes. A public attribute may be bound to a single node or to a sequence of nodes, depending on the Node type. For example, the bases attribute of the Class node, is bound to a list of base class nodes, and the doc attribute is bound to a single node.

Each Node instance has a lineno attribute which may be None. XXX Not sure what the rules are for which nodes will have a useful lineno.

**All Node objects offer the following methods:**

**getChildren()**

Returns a flattened list of the child nodes and objects in the order they occur. Specifically, the order of the nodes is the order in which they appear in the Python grammar. Not all of the children are Node instances. The names of functions and classes, for example, are plain strings.

**getChildNodes()**

Returns a flattened list of the child nodes in the order they occur. This method is like getChildren(), except that it only returns those children that are Node instances.

The While node has three attributes: test, body, and else\_. (If the natural name for an attribute is also a Python reserved word, it can’t be used as an attribute name. An underscore is appended to the word to make it a legal identifier, hence else\_ instead of else.)

The if statement is more complicated because it can include several tests.

The If node only defines two attributes: tests and else\_. The tests attribute is a sequence of test expression, consequent body pairs. There is one pair for each if/elif clause. The first element of the pair is the test expression. The second elements is a Stmt node that contains the code to execute if the test is true.

The getChildren() method of If returns a flat list of child nodes. If there are three if/elif clauses and no else clause, then getChildren() will return a list of six elements: the first test expression, the first Stmt, the second text expression, etc.

The following table lists each of the Node subclasses defined in compiler.ast and each of the public attributes available on their instances. The values of most of the attributes are themselves Node instances or sequences of instances. When the value is something other than an instance, the type is noted in the comment. The attributes are listed in the order in which they are returned by getChildren() and getChildNodes().

**DEVELOPMENT ENVIRONMENTS:**

Most Python implementations (including CPython) include a read–eval–print loop (REPL), permitting them to function as a command line interpreter for which the user enters statements sequentially and receives results immediately.

Other shells, including IDLE and IPython, add further abilities such as auto-completion, session state retention and syntax highlighting.

**IMPLEMENTATIONS**

**Reference implementation**

CPython is the reference implementation of Python. It is written in C, meeting the C89 standard with several select C99 features. It compiles Python programs into an intermediate bytecode which is then executed by its virtual machine. CPython is distributed with a large standard library written in a mixture of C and native Python. It is available for many platforms, including Windows and most modern Unix-like systems. Platform portability was one of its earliest priorities.

**Other implementations**

PyPy is a fast, compliant interpreter of Python 2.7 and 3.5. Its just-in-time compiler brings a significant speed improvement over CPython but several libraries written in C cannot be used with it.

Stackless Python is a significant fork of CPython that implements microthreads; it does not use the C memory stack, thus allowing massively concurrent programs. PyPy also has a stackless version.

MicroPython and CircuitPython are Python 3 variants optimized for microcontrollers. This includes Lego Mindstorms EV3.

RustPython is a Python 3 interpreter written in Rust.

**Unsupported implementations**

Other just-in-time Python compilers have been developed, but are now unsupported:

Google began a project named Unladen Swallow in 2009, with the aim of speeding up the Python interpreter five-fold by using the LLVM, and of improving its multithreading ability to scale to thousands of cores, while ordinary implementations suffer from the global interpreter lock.

Psyco is a just-in-time specialising compiler that integrates with CPython and transforms bytecode to machine code at runtime. The emitted code is specialized for certain data types and is faster than standard Python code.

In 2005, Nokia released a Python interpreter for the Series 60 mobile phones named PyS60. It includes many of the modules from the CPython implementations and some additional modules to integrate with the Symbian operating system. The project has been kept up-to-date to run on all variants of the S60 platform, and several third-party modules are available. The Nokia N900 also supports Python with GTK widget libraries, enabling programs to be written and run on the target device.

**Cross-compilers to other languages**

There are several compilers to high-level object languages, with either unrestricted Python, a restricted subset of Python, or a language similar to Python as the source language:

* Jython enables the use of the Java class library from a Python program.
* IronPython follows a similar approach in order to run Python programs on the .NET Common Language Runtime.
* The RPython language can be compiled to C, and is used to build the PyPy interpreter of Python.
* Pyjs compiles Python to JavaScript.
* Cython compiles Python to C and C++.
* Numba uses LLVM to compile Python to machine code.
* Pythran compiles Python to C++.
* Somewhat dated Pyrex (latest release in 2010) and Shed Skin (latest release in 2013) compile to C and C++ respectively.
* Google's Grumpy compiles Python to Go.
* MyHDL compiles Python to VHDL.
* Nuitka compiles Python into C++.

**PERFORMANCE**

A performance comparison of various Python implementations on a non-numerical (combinatorial) workload was presented at EuroSciPy '13.

**API DOCUMENTATION GENERATORS**

Python API documentation generators include:

* Sphinx
* Epydoc
* HeaderDoc
* Pydoc

**USES**

Python has been successfully embedded in many software products as a scripting language, including in finite element method software such as Abaqus, 3D parametric modeler like FreeCAD, 3D animation packages such as 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo, MotionBuilder, Softimage, the visual effects compositor Nuke, 2D imaging programs like GIMP, Inkscape, Scribus and Paint Shop Pro, and musical notation programs like scorewriter and capella. GNU Debugger uses Python as a pretty printer to show complex structures such as C++ containers. Esri promotes Python as the best choice for writing scripts in ArcGIS. It has also been used in several video games, and has been adopted as first of the three available programming languages in Google App Engine, the other two being Java and Go.

Python is commonly used in artificial intelligence projects with the help of libraries like TensorFlow, Keras and Scikit-learn. As a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing.

Many operating systems include Python as a standard component. It ships with most Linux distributions, AmigaOS 4, FreeBSD (as a package), NetBSD, OpenBSD (as a package) and macOS and can be used from the command line (terminal). Many Linux distributions use installers written in Python: Ubuntu uses the Ubiquity installer, while Red Hat Linux and Fedora use the Anaconda installer. Gentoo Linux uses Python in its package management system, Portage.

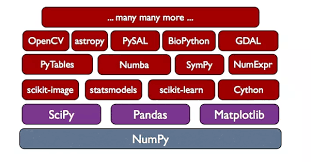
Python is used extensively in the information security industry, including in exploit development.

Most of the Sugar software for the One Laptop per Child XO, now developed at Sugar Labs, is written in Python. The Raspberry Pi single-board computer project has adopted Python as its main user-programming language.

LibreOffice includes Python, and intends to replace Java with Python. Its Python Scripting Provider is a core feature since Version 4.0 from 7 February 2013.

**PANDAS**

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.



**Library features**

* Data Frame object for data manipulation with integrated indexing.
* Tools for reading and writing data between in-memory data structures and different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of data sets.
* Label-based slicing, fancy indexing, and sub setting of large data sets.
* Data structure column insertion and deletion.
* Group by engine allowing split-apply-combine operations on data sets.
* Data set merging and joining.
* Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
* Time series-functionality: Date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.
* Provides data filtration.

**CHAPTER 7**

**SYSTEM DESIGN**

**UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

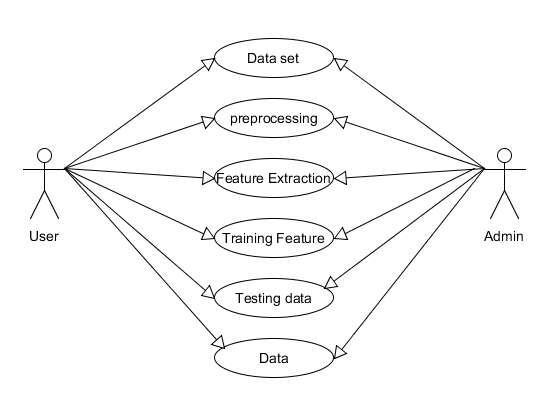
**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

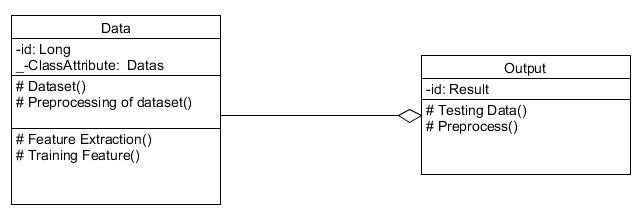
**USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

****

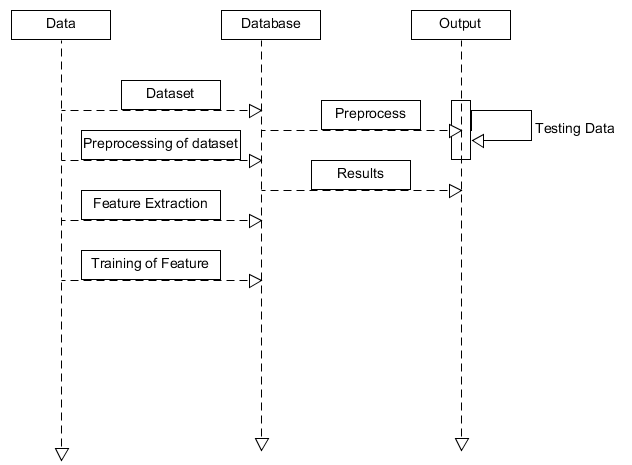
**CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**SEQUENCE DIAGRAM:**

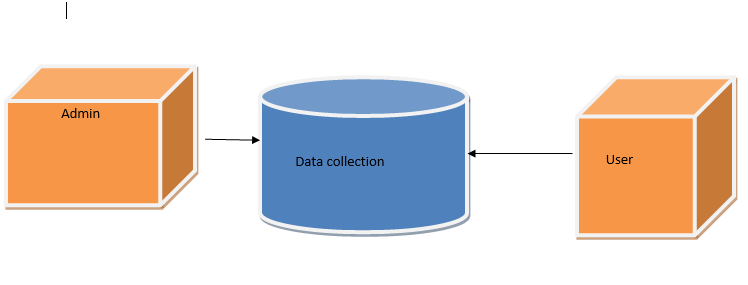
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**DEPLOYMENT:**

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components.

**DATA FLOW DIAGRAM:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

**CHAPTER 8**

### **SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**TYPES OF TESTS**

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

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.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**8.1 Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# **8.2 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**8.3 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**CHAPTER 9**

**CONCLUSION**

In conclusion, the integration of AI-enabled drones utilizing the You Only Look Once (YOLO) algorithm represents a significant advancement in disaster response technology. This innovative solution offers unparalleled efficiency in identifying human movements in challenging disaster environments. By leveraging real-time object detection, the system enables swift and accurate detection of survivors amidst rubble, debris, or hazardous conditions, thus improving search and rescue operations. The agility, cost-effectiveness, and remote monitoring capabilities of the proposed drone system make it a valuable asset for disaster response teams. Its ability to rapidly scan large areas and provide real-time data to first responders enhances situational awareness and facilitates timely intervention. Moreover, the system's remote monitoring capabilities allow responders to assess the extent of the disaster and allocate resources effectively, ultimately saving lives and minimizing casualties. Looking ahead, continued advancements in AI and drone technology hold great promise for further improving the effectiveness and reliability of such systems. With ongoing research and development, these systems can become even more adept at navigating complex environments and detecting human presence with greater accuracy. They will play an increasingly vital role in mitigating the impact of disasters on communities worldwide, ultimately saving more lives and safeguarding against future catastrophes.

**FUTURE ENHANCEMENT**

* Continued advancements in AI algorithms, particularly in object detection techniques, can enhance the system's ability to detect and track human movements with greater accuracy and efficiency.
* Developing autonomous navigation capabilities for drones can enable them to navigate complex disaster environments more effectively, including indoor areas or areas with limited GPS signal.
* Improved communication systems, including better connectivity and bandwidth, can enable drones to relay real-time data more efficiently to disaster response teams, facilitating faster decision-making and coordination.
* Implementing collaborative swarm systems where multiple drones work together can increase coverage area, redundancy, and overall system robustness in disaster scenarios.
* Integrating the drone system with existing emergency response infrastructure, such as command centers and mapping systems, can streamline coordination and enhance the overall effectiveness of disaster response efforts.

**REFERENCES**

1. Khan, S. Gupta, and S. K. Gupta, “Emerging UAV technology for disaster detection, mitigation, response, and preparedness,” Journal of Field Robotics, vol. 39, no. 6, pp. 905–955, 2022.
2. Liu, W.; Quijano, K.; Crawford, M.M. YOLOv5-Tassel: Detecting Tassels in RGB UAV Imagery with Improved YOLOv5 Based on Transfer Learning. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2022, 15, 8085–8094.
3. Gai, W.; Liu, Y.; Zhang, J.; Jing, G. An Improved Tiny YOLOv3 for Real-Time Object Detection. Syst. Sci. Control Eng. 2021, 9, 314–321
4. Karthi, M.; Muthulakshmi, V.; Priscilla, R.; Praveen, P.; Vanisri, K. Evolution of YOLO-V5 Algorithm for Object Detection: Automated Detection of Library Books and Performace Validation of Dataset. In Proceedings of the 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 24–25 September 2021; IEEE: Piscataway, NJ, USA
5. Matinrad, N.; Reuter-Oppermann, M. A review on initiatives for the management of daily medical emergencies prior to the arrival of emergency medical services. Cent. Eur. J. Oper. Res. 2022, 30, 251–302.
6. Wang, X.; Gao, H.; Jia, Z.; Li, Z. BL-YOLOv8: An Improved Road Defect Detection Model Based on YOLOv8. Sensors 2023, 23, 8361.
7. N. Zhang, F. Nex, G. Vosselman, and N. Kerle, “Training a disaster victim detection network for UAV search and rescue using harmonious composite images,” Remote Sensing, vol. 14, no. 13, p. 2977, 2022.
8. K. Sharma, R. Doriya, S. K. Pandey, A. Kumar, G. R. Sinha, and P. Dadheech, “Real-time survivor detection system in SaR missions using robots,” Drones, vol. 6, no. 8, p. 219, 2022.
9. J. Rosselló, S. Becken, and M. Santana-Gallego, “The effects of natural disasters on international tourism: a global analysis,” Tourism Management, vol. 79, no. 2020, Article ID 104080, 2020.
10. B. W. Robertson, M. Johnson, D. Murthy, W. R. Smith, and K. K. Stephens, “Using a combination of human insights and “deep learning” for real-time disaster communication,” Progress in Disaster Science, vol. 2, Article ID 100030, 100114 pages, 2019.
11. B. Mishra, D. Garg, P. Narang, and V. Mishra, “Drone-surveillance for search and rescue in natural disaster,” Computer Communications, vol. 156, pp. 1–10, 2020.
12. E. Lygouras, N. Santavas, A. Taitzoglou, K. Tarchanidis, A. Mitropoulos, and A. Gasteratos, “Unsupervised human detection with an embedded vision system on a fully autonomous UAV for search and rescue operations,” Sensors, vol. 19, no. 16, p. 3542, 2019.
13. K. Muhammad, J. Ahmad, and S. W. Baik, “Early fire detection using convolutional neural networks during surveillance for effective disaster management,” Neurocomputing, vol. 288, pp. 30–42, 2018.
14. M. Aqib, R. Mehmood, A. Albeshri, and A. Alzahrani, “Disaster Management in Smart Cities by Forecasting Traffic Plan Using Deep Learning and GPUs,” Smart Societies, Infrastructure, Technologies and Applications, Springer, Manhattan, NY, USA, 2018.
15. M. S. Hakimy Salem, F. H. Kamaru Zaman, and N. Md Tahir, “Human detection implementation on UAVs in search and rescue operation,” Journal of Electrical & Electronic Systems Research, vol. 18, pp. 70–76, 2021.

**APPENDIX**

**SOURCE CODE**

import cv2

import numpy as np

import matplotlib.pyplot as plt

import time

import gtts

import playsound

import os

from playsound import playsound

import random

import pygame

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

from email.mime.image import MIMEImage

import os

import smtplib

net = cv2.dnn.readNetFromDarknet("yolov8.cfg","yolov8.weights")

class\_ = None

classes = open("coco.names").read().strip().split("\n")

def report\_send\_mail(label, image\_path):

with open(image\_path, 'rb') as f:

img\_data = f.read()

fromaddr = "vimalmuthu623@gmail.com"

toaddr = "vimalmuthu623@gmail.com"

msg = MIMEMultipart()

msg['From'] = fromaddr

msg['To'] = toaddr

msg['Subject'] = "Alert Suspect Found on Cam 1"

body = label

msg.attach(MIMEText(body, 'plain')) # attach plain text

image = MIMEImage(img\_data, name=os.path.basename(image\_path))

msg.attach(image) # attach image

s = smtplib.SMTP('smtp.gmail.com', 587)

s.starttls()

s.login(fromaddr, "txkhhjvvvnnphgsy")

text = msg.as\_string()

s.sendmail(fromaddr, toaddr, text)

s.quit()

def audio():

pygame.mixer.init()

pygame.mixer.music.load("buzzer.mpeg")

pygame.mixer.music.play()

cap = cv2.VideoCapture(0)

while 1:

\_, img = cap.read()

img = cv2.resize(img,(1280,720))

hight,width,\_ = img.shape

blob = cv2.dnn.blobFromImage(img, 1/255,(416,416),(0,0,0),swapRB = True,crop= False)

net.setInput(blob)

output\_layers\_name = net.getUnconnectedOutLayersNames()

layerOutputs = net.forward(output\_layers\_name)

boxes =[]

confidences = []

class\_ids = []

for output in layerOutputs:

for detection in output:

score = detection[5:]

class\_id = np.argmax(score)

confidence = score[class\_id]

if confidence > 0.7:

center\_x = int(detection[0] \* width)

center\_y = int(detection[1] \* hight)

w = int(detection[2] \* width)

h = int(detection[3]\* hight)

x = int(center\_x - w/2)

y = int(center\_y - h/2)

boxes.append([x,y,w,h])

confidences.append((float(confidence)))

class\_ids.append(class\_id)

indexes = cv2.dnn.NMSBoxes(boxes,confidences,.5,.4)

indexes = cv2.dnn.NMSBoxes(boxes,confidences,.8,.4)

font = cv2.FONT\_HERSHEY\_PLAIN

colors = np.random.uniform(0,255,size =(len(boxes),3))

if len(indexes)>0:

for i in indexes.flatten():

x,y,w,h = boxes[i]

label = str(classes[class\_ids[i]])

if label=='person':

print(label)

cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),2)

cv2.putText(img,'Person Found',(x,y+200),font,2,(0,0,255),2)

cv2.imwrite('person.jpg',img)

report\_send\_mail('Person Found', 'person.jpg')

audio()

confidence = str(round(confidences[i],2))

color = colors[i]

cv2.rectangle(img,(x,y),(x+w,y+h),color,2)

cv2.putText(img,label + " " + confidence, (x,y+400),font,2,color,2)

cv2.imshow('img',img)

if cv2.waitKey(1) == ord('q'):

break

cap.release()

cv2.destroyAllWindows()