Multimodal Human Detection In Disaster Situations Using Deep Learning & Artificial Intelligence

by

Md. Tasbi Rafat

19101493

Fahmida Ahmed Hridy

19101188

Rafid Ibna Zaman

18301249

Hasibul Hassan Santo

18301019

Ibrahim Siddique

19101252

A thesis submitted to the Department of Computer Science and Engineering in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science.

Department of Computer Science and Engineering
Brac University
September 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md. Tasbi Rafat
19101493

Fahmida Ahmed
19101188

Rafid Ibna Zaman
18301249

Hasibul Hassan Santo
18301019

Ibrahim Siddique
19101252

Approval

The thesis titled "Multimodal Human Detection In Disaster Situations Using Deep Learning & Artificial Intelligence" submitted by

1. Md. Tasbi Rafat (19101493)
2. Fahmida Ahmed Hridy (19101188)
3. Rafid Ibna Zaman (18301249)
4. Hasibul Hassan Santo (18301019)
5. Ibrahim Siddique (19101252)

Of Summer, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September, 2022.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Khalilur Rhaman
Professor, PhD
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:

Md. Golam Rabiul Alam
Professor, PhD
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, Ph.D.
Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Mankind has faced natural calamities for survival since the very beginning of human civilization. Even after 65 million years, mankind is still figuring out ways to face natural calamities and survive its post-consequences effectively. Against natural phenomena like- hurricanes, tornadoes, earthquakes, building collapse, forest fires, etc. Humankind is weak and helpless. And no matter how technologically advanced humankind becomes, nature will always remain the strongest opponent that humans have to face for their survival. The revolution of science and technology has helped humankind to invent ways and techniques to survive by fighting against the natural calamities that they face. Technology can reach into places where humans cannot and technology can look deep into details that humans can never go through due to born limitations. Our paper represents the idea of a human detection system that during any calamity, with the help of multiple detection sensors and thermal visualization techniques, can detect trapped human beings. This human detection system combines the knowledge of Machine learning and Artificial intelligence system techniques. We hope to contribute to saving human lives during natural calamities and help them to overcome its aftermath in the quickest possible time.

Keywords: Natural calamity; Survival; Artificial Intelligence; Machine learning; Detection System; Deep Learning.

Dedication

We want to dedicate all our efforts and struggles of educational life to our dear parents, without them we are meaningless. Also we devote the work of this thesis to respectable supervisor Dr. Md. Khalilur Rahman sir who taught and supported us in developing our skills and personality as a competent professional.

Acknowledgement

First of all, all praise to the Almighty Allah for whom our thesis has been completed without any major interruption. Secondly, to our advisor Dr. Md. Khalilur Rahman sir for his kind support and advice in our work throughout the thesis phase. He has helped us whenever we needed any support and assistance. And finally to our parents. Without their thorough support it may have not been possible.

Nomenclature

The next list describes several symbols abbreviation that will be later used within the body of the document -

- AI Artificial Intelligence
- ANN Artificial Neural Network
- AP Average Precision
- CNN Convolutional Neural Network
- CUDA Compute Unified Device Architecture
- cuDNN NVIDIA Deep Neural Network
- CV Computer Vision
- FEN Feature Extraction Network
- FNN Feedforward Neural network
- IoU Intersection Over Union
- LReLU Leaky Rectified linear unit
- MSE Mean Squared Error
- NMS Non Maximum Suppression
- ReLU Rectified linear unit
- RPN Region Proposal Network
- YOLO You Only Look Once
- OID Open Images Dataset

- mAP Mean Average Precision
- FLOPS Floating Point Operation Per Second

Table of Contents

Chapter 1

Introduction

For the last few decades, scientists have been trying to invent an efficient rescue and detection system to face calamities and its post-consequences. According to the research of Jagran Lakecity University (Rathore, 2016), in recent times, in order to deal with tragedies, disasters, and stop unintended deaths, technology such as satellite navigation systems, systems of geographical information, GPS, and remote sensing systems has been deployed. [5].These are a few applications for these technologies in disaster management: establishing emergency warning systems, setting up a quick response team, analysis of post-disaster consequences, etc. The creation of databases, information integration and analysis, emergency response, disaster response planning, and other applications are also included. Due to early warnings of powerful storms and volcanic eruptions, many lives have been saved and significant property damage has been avoided.

Modern technology includes developments that reduce the strain on life from natural disasters in the physical and built environment. Advances in planning and construction techniques have made seismic structures such as skyscrapers, critical lifelines and industrial plants technically feasible and a reality. But scientists and disaster management specialists are still trying to develop a life-saving system that is accessible for all and that can be used during disaster phases by all nations to save human lives. The main challenge of developing a life-detection system and technique is maintaining accuracy for a longer period of time and detecting human lives in a short amount of time. Most of the techniques and technologies developed during the last decade do not maintain accuracy and after a certain period of time, the core system of the machines provides no results. Thus, very few lives are saved and many lives are lost due to proper technological assistance. According to recent research [8], due to delayed technical progress and failure, natural disasters cause 60,000 deaths per year on average, or 0.1 percent of all deaths worldwide. We know that unpredictable high-impact events like earthquakes and tsunamis occur frequently, but high losses of human lives can be prevented by using advanced technology. We are aware that earlier disaster prediction can significantly reduce disaster-related fatalities around the world and quick response to post-disaster consequences can save many lives as well. Human detection is a useful method and can be used by all nations to prevent unwanted deaths during the disaster.

Governments across the world can collect data from different sectors and provide the data to scientists so that they can develop the life detection machine accordingly. Data-driven machines can read a situation better and act accordingly in post-disaster situations. To ensure high chances of survival, a life-detection system is a must along with pre-disaster measurements especially in those countries where natural calamities hit hard every year. An efficient data-driven life detection machine has the capability of detecting human lives in critical conditions where humans can not reach out to help other people. It is high time we realize the necessity of using our resources and technology to develop a life-detection system for the betterment of humankind.

1.1 Research Problem

Building collapses and structural damages due to calamities can cause people to get trapped inside a pile of bricks or walls and rocks. Survival under structural damage and collapsed buildings is challenging as the elements of survival like oxygen, sunlight, are not available. In such cases, proper detection is crucial for rescuing trapped and injured human lives. Researchers suggest that there is a high possibility of survival of an individual trapped inside a collapsed structure if we can accurately detect their whereabouts. The survival rate drops down after 72 hours of being trapped under a collapsed building. Several human factors, including firefighters, police officers, and medical assistants, are involved in the rescue effort. They are all subjected to extremely risky conditions brought on by the destroyed environment they work in, including landslides, craters, and collapsed structures. As a result, the rescuer could turn into a victim who needs to be rescued. As a result, the rescue team members are exposed to a high level of risk during the rescue operation. From this perspective, there has been a significant demand for finding substitutes for human operated rescues missions.[4] In search of independent alternatives to the human factor, we have found sensor technology and thermal image processing. But there are several challenges that arise when using sensor based and thermal based detection systems.

First of all, according to researchers, sensors can detect and collect signals from the smallest of movements and thermal imaging detects every object around them. Which means the system must be capable of processing large amounts of data. To deal with this much data, we needed to come up with a framework that can handle such a large volume of data to showcase results in real-time while practically implementing it. Also thermal cameras detect every living object in the environment and showcase it as data in the frame. Which means the system needs to efficiently distinguish our targeted object from the whole scene. The whole detection system will lag behind and will not be able to fulfill the purpose of rescuing if accurate data can not be collected and detected in time. Moreover, the data needs to be trained and the overall system needs to compat with the trained data. Making the detection process more precise and accurate based on the sensor's and thermal reading requires processing a large scale of data. Also, the overall system needs to be run with labeled data as specification is required. It was difficult to find proper datasets to implement the algorithms' and test the trained data. Also, gas sensors detect the smallest amount of gas found in the atmosphere and thus finding the accuracy for our targeted gas detection was a challenging task. Although multiple research studies have been conducted across the world on detection systems, proper datasets and resources were difficult to find for training our model, its classification and implementing the detection algorithms.

1.2 Research Objectives

In our research, we aim to develop a human detection system to detect trapped human beings using sensors and thermal imaging. The overall process of detecting trapped human figures is a data-driven approach in which we have used supervised learning, artificial intelligence, detection algorithms, machine learning, and deep learning techniques. Our research mainly focuses on -

- Detecting human beings and their chances of survival in complex environments with image processing and sensor data retrieving.
- Collect relevant data and analyze the result accuracy of data by comparing different algorithms and machine learning models.
- Choosing an optimized algorithm and model for our proposed system to work properly in real time environments.

Chapter 2

Literature review

2.1 Related Research

According to (Paul et al., 2013), the three types of technique for classifying objects based on shape, motion and texture. When using shape-based techniques, movable regions like points, boxes, and blobs' shape information is first described. Then, it is typically regarded as a typical template-matching problem. Because of the human body's articulation and different viewing angles, a moving person can appear in a wide range of ways, making it incontestably challenging to tell them apart from other moving things. Applying part-based template matching could help solve this problem. Support vector machines (SVM) are used in texture-based approaches like histograms of oriented gradient (HOG) to recognize human regions by using high-dimensional characteristics based on edges [3]. Also, according to (Paul et al., 2013), By separating an object from the scene captured by a security camera and using background subtraction, it is common practice to identify an object as the foreground. The camera may be stationary, purely translational, or movable. By comparing the differences between the current frame and the reference frame on a pixel-by-pixel or block-by-block basis, background subtraction tries to locate moving objects. Common names for the reference frame include "background image," "background model," and "environment model." A decent backdrop model is required flexible enough to respond to the alterations in dynamic scenarios. This could be accomplished by routinely updating the background data, but it is also possible to accomplish this without doing so [3].

According to (Lin, 2010), The separation of moving objects from backgrounds and the differentiation of people from nonhuman things make up the two halves of a human detection system. Optical flow, stereo-based vision, and temporal difference are only a handful of the techniques used to distinguish moving objects from backgrounds. Although more computationally intensive and sensitive to changes in intensity, the optical flow method might be successful in detecting independently moving objects [1].Zhao and Thorpe, two authors, used the stereo-based segmentation technique to separate items from backgrounds and then used neural networks to identify the separated objects. The stereo-based vision technique has been shown to be more reliable, but it can only be used for close-range detection and needs at least two cameras. Using several cameras to assess the 3D skeletal structure in gait sequences and 3D skeletons to comprehensively extract human body morphologies, another author, Orrite Urunuela, developed the point distribution model (PDM) using Principal Component Analysis (PCA). [1].

According to Lin (2010), there have been developments in shape-based, motion-based, and multicue-based techniques for human recognition . Zhou and Hoang created a codebook using the shape information of human bodies to distinguish humans from other objects. This approach would undoubtedly be effective if the extracted human shape was clear. However, in situations where people are partially obscured or are seen carrying something, this shape-based approach would typically fail. The Fast Fourier Transform-based Algorithms for Histograms of Oriented Gradients (HOG) extracted characteristics from the shape data. In order to determine the initial hypothesis of items, a researcher and IEEE member named Curio first performed the detection method based on the geometrical characteristics of humans. Yoon and Kim, two other IEEE members, classified human or other objects with comparable skin tones by using background removal, background color, and upper body appearance information. Regarding methods utilizing neural networks for human identification [1]. Most often, feature extraction has been the subject of research, but feature selection has received far less attention. In recent years, ICA has been used to extract human traits in order to build a complete set of features describing people. The PCA, which only takes into account second-order statistical reasons, can be viewed as an extension of high-order statistical analysis methods like ICA. Since the ICA features are not sorted like the PCA features are, the feature selection, sorting, and selection of a suitable subset of ICA features are all done using conditional entropy. Sorting variables may be a crucial step in improving the high-dimensional dataset, which is how we came up with the idea of clustering associated or comparable dimensions together in high-dimensional visual space to make it simpler for users to understand the links between those variables [1].

According to (Correa et al., 2011), Thermal sensors, on the other hand, enable accurate detection of human bodies independent of lighting and attitude, and their detection range can reach several meters, which is adequate for home settings. Additionally, by examining faces in the thermal spectrum, persons can be recognized. All of these characteristics make it appear logical to include thermal cameras into both existing and upcoming home service robots. Since the cost of thermal cameras has decreased dramatically over the past few years and is now comparable to the cost of time-of-flight cameras and middle-range laser sensors, both of which are frequently employed in domestic robots, it is no longer an obstacle to their employment in these machines. A FLIR TAU 320 thermal camera powers the robot in the current project. This camera's resolution is 324 256 pixels, and its longwave infrared sensitivity range is 7.5-13.5 m. [2]. The technology primarily uses heat information when there is poor or inconsistent lighting. However, in well-lit environments, thermal and visual information work in tandem. Visual information, for instance, permits a better analysis of textures and a more accurate eye detection in order to align faces before identification. In backgrounds with a lot of detail, thermal information makes it easier to discern between human bodies and faces [2]. The topic of human detection and identification has been covered extensively in contemporary literature. The utilization of stereo vision, monocular vision, sonar and vision, laser and vision, and thermal vision is the foundation of various works in sensor technology. For instance, a stereovision system employs a dense depth image to detect and track individuals; a sonar and skin color detector are combined to detect faces; a laser is used to detect human legs; and a vision system is used to detect faces. It is simpler to distinguish human bodies or human body parts from the background when using thermal vision, which is one of its key advantages [2]. Face information is one of the most often used indications for detecting and recognizing people, according to (Correa et al., 2011). In the current research on face detection using machine learning approaches, the use of thermal imaging has not gotten as much attention as the use of visual photographs. The most effective face detection techniques rely on machine learning algorithms like boosting classifiers, convolutional neural networks, and support vector machines (SVM). The most common face detection methodology uses cascades of boosted classifiers to detect faces in a reliable and effective manner [2]. Also, In the past few years, a variety of face recognition techniques have been created, ranging from traditional Eigenspace-based techniques (like Eigenfaces) to complex systems based on high-resolution photos and 3-D models [2]. There are a number of techniques that have been developed for the identification of faces in thermal images, and the majority of them are based on techniques that are also utilized for visual image recognition. The use of thermal imaging (long wave infrared, 8–12 m) for facial identification is compared. The study focuses on the three methods that gave the best visible spectrum outcomes while accounting for the HRI requirements of online and real-time operation, one image per person, and unrestricted surroundings: Local Binary Pattern (LBP) Histograms, Scale-Invariant Feature Transform (SIFT) Descriptors, and Gabor Jet Descriptors [2]. According to the findings of the article on human-robot interaction, LBP-based algorithms can achieve extremely high recognition rates and have computational and memory needs that are suitable for HRI application [2].

Thus, after studying about these different methods, the key contribution of our work is the suggestion of a multimodal system based on image detection and sensor data for human detection and identification in complex situations. Along with image detection, we have also gathered sensor data to understand a complex environment in which several human recognition techniques are required for accurate results.

2.2 More research

According to IOPScience, microwave signals have been used to try and find human bodies trapped under an obstacle. To determine whether there are any living people underneath the rubble, a microwave signal is sent. This signal can pass past obstacles and return after striking a human body. When a signal beam strikes the body, it is reflected with additional modulation brought on by the heart and lungs working. Receiving modulated signals reveals the presence of people moving toward life beneath the debris. Other signals, such as clutter signals, are reflected from immovable objects. From 2GH (L-band) to 10GHz, the microwave detecting system may operate on various frequency ranges (X-band)[9].

Another proposed system contains three ultrasonic tracking devices that are used on both human legs. The idea is that the sensors are placed horizontally at a certain distance, which corresponds to the typical width of a human being's legs when they are walking normally. The installation of ultrasonic sensors in the area of interception where human legs are being detected is the core idea. When a signal is intercepted between sensors 1 and 7, the robot is informed that the left leg of the human is being sensed. When a signal is intercepted between sensors 2 and 3, the robot is informed that the right leg of the human is being sensed [9].

PIR sensors, which can detect motion, are almost always employed to assess whether a human has entered or left the area of view of the sensor. [27]. PIRs fundamentally comprise a pyroelectric sensor that measures the intensity of infrared radiation. Every thing produces some low-level radiation, and the more radiation an object produces, the hotter it is. A motion detector's sensor actually has two sides. This is so that we can identify motion rather than an average of IR levels. Due to the wiring of the two portions, they cancel one another out. The output will showcase high or low depending on whether one side shows more or less IR radiation compared to the other [27]. There is a lot of supporting circuitry, resistors, and capacitors in addition to the pyroelectric sensor. This chip uses the sensor's output and some light processing to turn the analog sensor's output into a digital output pulse [27]. A pair of pyroelectric sensors are used by passive infrared (PIR) sensors. PIR sensors detect heat energy in the immediate vicinity. When there is a shift in the signal difference between these two sensors, which are placed next to one another, the sensor will turn on. That indicates that it sets off an alarm, alerts the authorities, or perhaps turns on a floodlight [27].

In industrial environments, active IR sensing is quite prevalent. In these scenarios, a pair of emitters and receivers can precisely identify if an object is, for instance, in a specific location on a conveyor. [27].

We have considered all the techniques stated above for our proposed system as well. But given that our main focus was on human detection based on still image data, we decided not to use any motion sensors. Instead we have used gas sensors with image detection for complex situations.

Chapter 3

Background Study

3.1 Artificial Intelligence

Through Artificial intelligence (AI), machines are trained to think like humans using advanced algorithms. Artificial intelligence along with machine learning algorithms can be trained to recognize the image, analyze the data, and check patterns of humans under any circumstance. AI image recognition can help a system to recognize and analyze images with the goal of identifying objects, places, people in any environment. Object or human being recognition work requires big data work and AI can analyze all the collected data to extract relevant information. To make this analysis possible and generate outcomes, we need to train the data with algorithms, validate the data according to trained data and test the data so that we can get results with precision. The main purpose of using AI in our research is to help recognize a trapped human being and showcase results in quick real-time.

The creation of sophisticated machine learning models to support advanced analytics, also known as the use of artificial intelligence, is the purpose of the third phase of the data pipeline, which comes after the gathering and preparation of data.By employing a variety of algorithms, such as logistic or linear regression, these models mimic human expertise in spotting patterns in the data and drawing conclusions.

3.2 Machine Learning

Software programs can accurately predict outcomes if it is employed through machine learning (ML), which is a type of artificial intelligence (AI). In order to forecast new output values, machine learning algorithms use previous data as an input. Traditional machine learning is frequently categorized according to the manner in which a prediction-making algorithm learns to increase its accuracy.There are four major approaches of machine learning: reinforcement learning, semi-supervision learning, supervision learning, and learning with supervision. The type of algorithm is determined by the type of data we want to predict and see the outcome [17].

- **Supervised learning:** Labeled training data and a description of the variables are used for the algorithm to search for connections between supervised learning data.

- **Unsupervised learning:** Unsupervised learning uses the algorithms on unlabeled trained data.The approach seeks out any significant connections that exist between datasets. The data for training and making predictions are predefined. [17].

According to Burns 2021, supervised learning algorithms are effective for the following tasks: [17]:

- **Binary classification:** It classifies the data into two categories. Uses a binary system.
- **Multi-class classification:** It chooses from two different categories of responses and sometimes more than two categories are taken.
- **Regression modeling:** We can predict simultaneous values using regression modeling.
- **Ensembling:** It combines the accurate predictions from various machine learning models.

We have used supervised learning for our proposed work and compared results and estimated outcomes. To assess the correctness of the data we have gathered from our trained model, we have employed a number of machine learning methods.

3.3 Transfer Learning

In transfer learning, a method of machine learning, a new model is built on top of an existing model that has already been trained.To put it another way, an optimization of a model created for one task is applied to another, related task to speed up modeling progress on the first task. Performance can be considerably enhanced over training with only a modest amount of data if we use transfer learning on a new task. It is uncommon to train a model from scratch for tasks linked to image or natural language processing because transfer learning is so ubiquitous. Performance can be considerably enhanced over training with only a modest amount of data by using transfer learning on a new task. It is very rare to train a model from scratch for tasks linked to any image or natural language processing (NPL). Because transfer learning is very ubiquitous. Academics and data scientists, on the other hand, would rather start with a model that has already been trained to classify objects and shapes in images and recognize general features like edges [32].

3.4 Computer Vision

The goal of computer vision (CV), a branch of science, is to enable machines to comprehend and mimic the functions performed by the human visual system. CV primarily focuses on the ways that machines could use to extract information from digital images or movies. and at the moment, the most prominent applications[26].

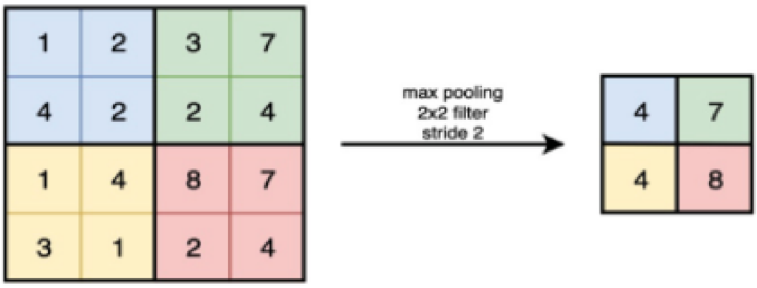

Figure 3.1: Example of max pooling[26]

In this thesis, our primary focus is on the Object Detection task, which entails searching digital images and videos for instances of specific semantic items (such as people, buildings, or automobiles). [28] [16].

We did not use this algorithm for image detection since according to (Why Computer Vision Is Difficult to Implement? (and How to Overcome), 2021), Computer Vision Is Difficult Because Hardware Limits it [19]. It needs regular monitoring otherwise it might break down which can cause terrible loss of data in detection.

3.5 CNN Algorithm

The majority of the time, visual imagery is evaluated using a class of deep neural networks. The class is called convolutional neural networks (CNN). Convolutional neural networks are created by layering artificial neurons together. Artificial neurons simulate biological neurons by creating an activation value from the weighted sum of numerous inputs. The first layer typically extracts fundamental features like horizontal or diagonal edges.This output is received by the layer below, which then looks for more complex features like multiple edges or corners.It is able to recognize increasingly complex elements like faces, objects, etc. [20].

The classification layer creates a series of confidence ratings (numbers between 0 and 1) that represent the likelihood that the image belongs to a "class" based on the activation map of the final convolution layer. The Pooling layer, like the Convolutional layer, is responsible for shrinking the spatial size of the Convolved Features. Despite their enormous power and complicated resource requirements, CNNs produce comprehensive results.It all boils down to simply recognizing patterns and nuances that are so minute and subtle that the human eye misses them.However, it fails to comprehend the contents of an image.These methods and the algorithm were used in our study to train our dataset to recognize humans. [20].

3.6 Retina-Net

With a backbone network and two task-specific subnetworks, Retina-Net is one cohesive network. The backbone, an off-the-self convolution network, computes a conv feature map for the whole input image. The output of the backbone is classified by the first subnet; convolution bounding box regression is performed by the second subnet. [11].

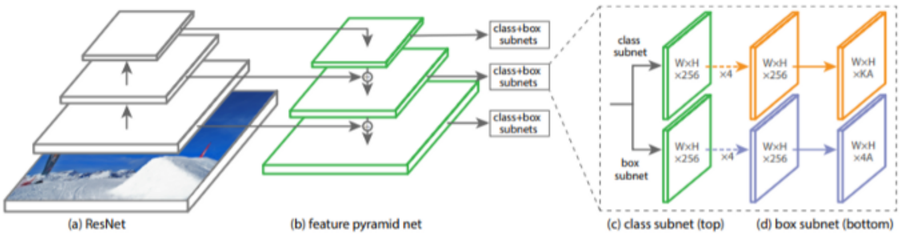

Figure 3.2: Retina Net Architecture[11]

Two subnets receive data from Retina Net's extensive, multi-scale convolutional feature pyramid. The anchor boxes are classified by one subnet, while the ground-level data boxes are regressed from the anchor boxes by the other[11]. We have studied this algorithm for our proposed model for comparison, but we did not use it because YOLO can detect multiple bounding boxes and categories at the same time.

3.7 Single Shot MultiBox Detector (SSD)

The SSD free-forward convolution layer approach produces an intact sized collection of bounding boxes and scores for each instance of an object class identified within those bounding boxes. To obtain the findings, a Non-Max Suppression is also applied. [11].

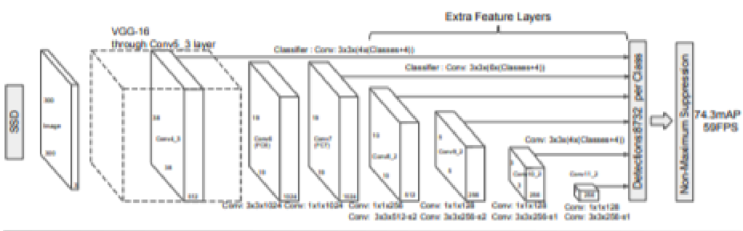

Figure 3.3: SSD Architecture[11]

The SSD architecture is really straightforward. The base network, or conventional ConvNet layers for image classification, serves as the model's first layer. On top of this base network, more layers are added to produce the detections while taking into account feature maps at multiple scales, default boxes, and aspect ratio. [11].

According to (Gao, 2018), shallow layers in a neural network may not create enough high level characteristics to make prediction for small objects, hence we did not utilize this approach for our suggested model for comparison. Consequently, SSD performs less well for little things than for larger objects. The requirement for complicated data augmentation also indicates that it requires a lot of data to train. [6].

3.8 Classification Network

The region suggestion produced by RPN and the feature maps produced by FEN are inputs used by the Classification Network. The Classification Network uses a fully-connected layer and a softmax layer to classify the objects in the region suggestions. Bounding box regression is used in the meantime to modify the region proposal for more accurate bounding boxes. The faster R-CNN output produces the object detection result after classification and bounding box regression. [16].

Chapter 4

Methodology

Our proposed system, with the help of sensors and image processing, can detect human lives considering distinguished organs and human shapes. The proposed sensors can detect the amount of methane gas in the enclosed environment, the amount of LPG gas in the environment, and the amount of CO gas as well. The camera captures the situation and the algorithms process the images for detection and visual confirmation. After the analysis is done, the system can show the exact result of human lives being detected. The infrared of objects is found and measured by an infrared camera. The infrared data is converted into an electrical image by the camera. The target object's apparent surface temperature is depicted in the image. For the image processing part we have used and compared between YOLO v3, v4, v5, v6, and v7 algorithms to measure the accuracy of identification of humans through body parts. The results we have received are promising and comparisons are stated later in the paper. Gas sensors that we have used can identify the amount of methane gas, CO gas, LPG gas, and smoke in the atmosphere. We have compared the sensor data with the normal atmosphere gas data to draw a conclusion about the amount of methane, CO, LPG in the enclosed atmosphere. After the annotation of objects or people in the image is complete, the machine learning models were used to train image data along with all sensor data. After all the data was fetched and analyzed, we got the outcome of the proposed system and we learned about whether there is a human in a specific environment. The additional sensors are used to know further in what state the human being was at that moment.

4.1 Collecting data for image dataset

Our proposed model integrates the image processing data and sensor reading data in a single system. In complex environments we need both the systems for accurate detection. Our image dataset contains data from different sources and different environments. We have tried to keep different sets of images for training for the detection system so that the detection accuracy does not fall in different scenarios. We have collected disaster area images, local shops images, local area images, occluded images of different body parts, roadside images, accident images, etc. As the dataset contained different environment images, the training was diverse and the test and accuracy results were above 75% precision for our selected algorithm and data model.

4.1.1 Collecting data for sensor dataset

We tested our gas sensor data in a construction site given that the environment was similar to after the disaster situation. We tested the gas sensors there and collected sensor readings for detecting methane, LPG, smoke, CO, and other amounts of gasses. The readings of the sensor were taken in a limited scope of environment and the data we collected showed decent accuracy above 50% for all the algorithms we have tested against. For the temperature sensor, we have collected data in the same environment and we also created a manual environment to test the sensors in a limited range of environments. The temperature was collected in both Celcius and Farenheit and the accuracy results for the data was above 50% for the algorithms we have tested against.

4.1.2 AI Models

For image processing, we have studied several algorithms and have chosen the best fit one for our dataset. Our proposed system uses the data trained through machine learning methods to automate the detection process for the image section. After being trained, the image detection system will automatically detect humans in an image with proper accuracy. For the sensor part, the data readings are automated according to environment change and situation change. The data readings will be compared to normal environment readings and the situation will be read by the system. The results will be shown according to the detected and collected data in the monitor from which we will be able to assess the situation. The algorithms used, the models tested upon, the sensor data collected using arduino IDE and the annotated dataset all contribute to the automation of the system and will practice making logical decisions based on available data that we are proposing in this paper. The overall AI model implements real time analytics and predictive analytics for our proposed system.

4.2 Hardware

The image data were collected from different sources. For the sensor data we have used two types of sensors - MQ-2 for LPG, CO, smoke detection, LM-35 for temperature detection. For our sensor dataset we have collected real time sensor data by creating a manual atmosphere around our sensors. We viewed the sensor's output in the serial monitor. Another part of our research focused on temperature sensing and the possibility of determining temperature in an enclosed place for understanding the situation of a human under that circumstance. We have measured the temperature by examining an enclosed atmosphere within a limited range. Under any enclosed atmosphere, the temperature rises high. To measure the temperature in an enclosed environment, we have collected time to time data using the fundamental principles of a diode. The voltage across a diode grows at a known pace as the temperature rises in a semiconductor. We used the sensors to collect data in different ranges and recorded the temperature in Celsius. Then we converted it into Fahrenheit and completed the dataset. We produced heat manually using thermal devices to check the sensor's readability and proper usage. The results were shown in the serial monitor.

We have used Gas sensor MQ-2 for measuring the amount of CO, LPG, and smoke. This gas sensor can detect the amount of LPG, smoke, and carbon monoxide. It can detect flammable gas in a range of 300 - 10000ppm. In our case we have collected the data of smoke and carbon monoxide from the sensor. We interfaced our sensor with Arduino uno and took the serial monitor data from the serial monitor. For checking whether our sensor is working perfectly or not we created an atmosphere containing carbon monoxide gas in a limited range. Then we used the sensors and collected data for completing the datasheet. Our dataset contains the amount of LPG, smoke and carbon monoxide detected from the sensors in a limited range.

Figure 4.1: MQ-2

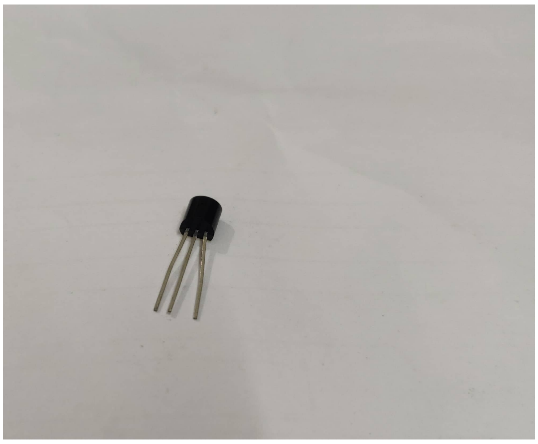

Figure 4.2: LM35

For the sensor data collection we have set up the sensors like the image shown below -

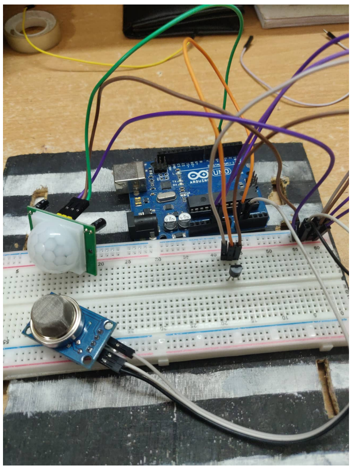

Figure 4.3: Sensor set up for collecting data

The following diagram is a schematic diagram of MQ-2 sensor-

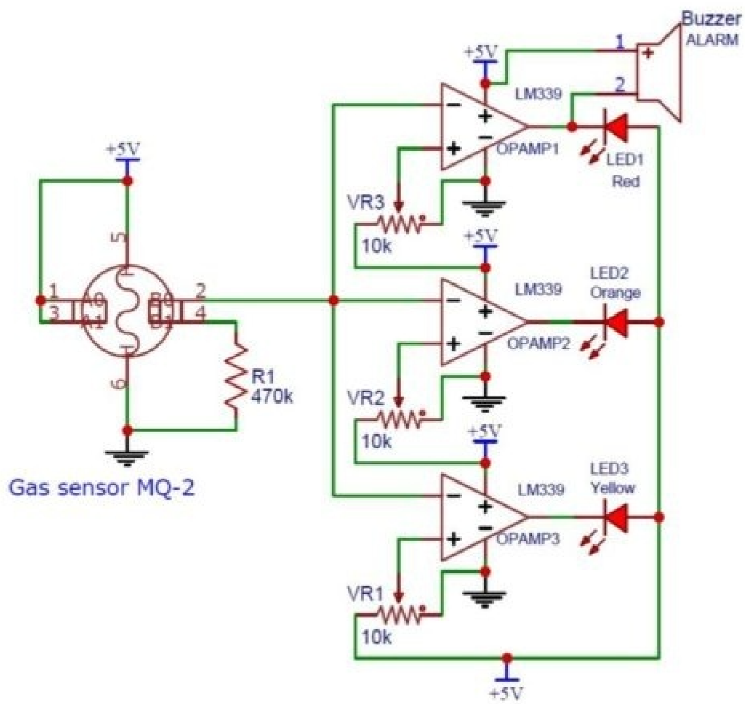

Figure 4.4: MQ-2 sensor diagram [14]

The following diagram shows the Schematic diagram for LM-35 sensor-

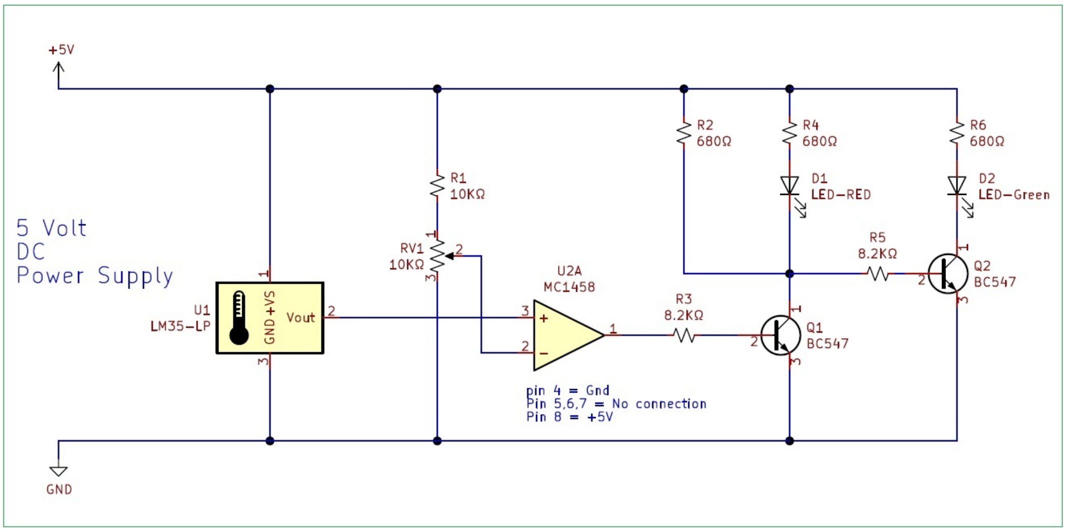

Figure 4.5: LM-35 sensor diagram[25]

Chapter 5

Algorithms Used

We have used YOLO algorithms to check accuracy and compare the results from our dataset. YOLO algorithms use neural networks to provide real time detections.

5.1 Artificial Neural Network (ANN)

A computational network that is based on biological neural networks, which are what give the human brain its structure, is commonly referred to as an artificial neural network. Artificial neural networks also have neurons that are connected to one another in different layers of the network, similar to how neurons are coupled in a human brain. Nodes are the name given to these neurons. [29].

Artificial Neural Network primarily consists of four layers:

1. Input layer
2. Hidden layer 1
3. Hidden layer 2
4. Output layer.

Input Layer: It selects and accepts input from the programmer in many different formats.

Hidden Layer: The hidden layer is displayed between the other two layers. It executes all computations required to check and find out hidden patterns and features.

Output Layer: This layer is used to convey the output after the hidden layer has modified the input. When provided input, the artificial neural network integrates a bias and determines the inputs' weighted sum. A transfer function is used to visualize this computation.

$$\sum_{i=1}^{n} \mathrm{Wi} * \mathrm{Xi} + \mathrm{b}$$

It decides to use the weighted total as an input to an activation function to generate the output.The activation functions of a node decide whether it should fire.Only fired employees have access to the output layer.There are various activation functions that can be utilized based on the kind of job we are doing [29].

5.2 Artificial Neural Network (ANN) presentation

The most effective representation of an artificial neural network is a weighted directed graph, where nodes stand in for artificial neurons. The weighted directed edges represent the link between the inputs and outputs of the neuron. An external source provides the artificial neural network with a pattern and a vector as input signals. Then, each of the n inputs is assigned mathematically using the notations x(n).[29].

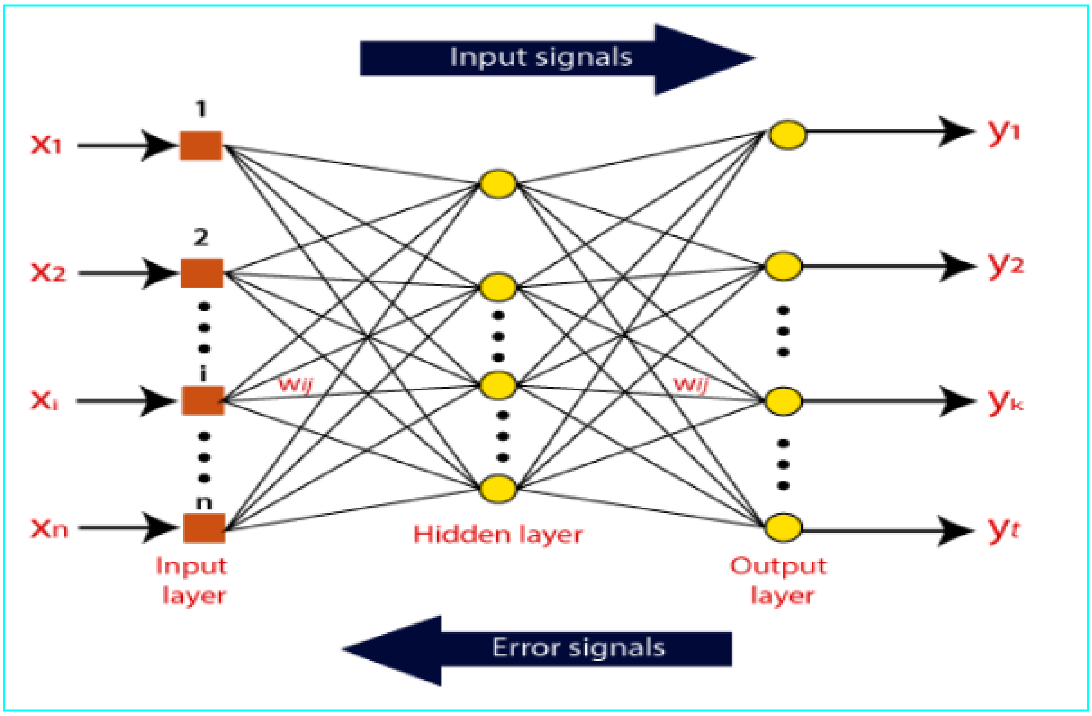

Figure 5.1: How to Artificial Neural Network Work[29]

After that, the weights that correspond to each input are multiplied by them. In the artificial neural network, these weights often indicate how well neurons are connected to one another. Inside the computer unit, a summary of each weighted input is created [29].

If the weighted total is zero, the output is made non-zero by adding bias, or else something else is added to scale the output to the system's reaction.The weight is 1, and the bias input is the same.In this scenario, the sum of the weighted inputs can be anywhere from 0 to positive infinity.The activation function is fed by the sum of the weighted inputs, and a particular maximum value is benchmarked to keep the response within the bounds of the intended value [29]

The activation function is the collection of transfer functions used to generate the desired output. The bulk of activation functions are either linear or non-linear sets of functions, though there are other types as well. There are other kinds of activation functions as well, although linear or non-linear sets of functions make up the majority of them. [29].

Binary: The result of the binary activation function is either a one or a zero. The activation function returns either one or 0 as its final output if the net weighted input of neurons is larger than 1. [29].

Sigmoidal Hyperbolic: The typical illustration of the Sigmoidal Hyperbolic function is a "S" curve. In this scenario, the output of the actual net input is approximated by the tan hyperbolic function..The function has the formula:

F(x) = (1/1 + exp(-Ax))

where the steepness parameter A is [29].

Neurons in ANN are connected with neurons from its previous layer and its subsequent layer. Each connection between the neurons has a weight. A neuron stands for a calculation process: it takes the output of the neuron(s) from the previous layer as input(s), and generates the output by linear calculations of inputs with the weights. Mathematically, a neuron j is simply a function f : f : RN $\rightarrow$ R, defined as [26]:

$$f\left(\boldsymbol{x}; \boldsymbol{w}_j\right) := \sigma\left(w_{0j} + \sum_{i=1}^{n} w_{ij} x_i\right)$$

where x is the input vector, wj $\in$ R^N $+ 1$ is the weight vector corresponding to the inputs and σ : RN $\rightarrow$ R is an activation function. In ANN, neurons are arranged into multiple layers, Feedforward Neural network (FNN) is a classic layered structure of ANN, as shown in the above equation [26]

5.3 Loss Function

The learning objective of a neural network can be stated as follows: Given the input and the ground truth, approximate a function that produces an output that closely resembles the ground truth. There should therefore be a metric for assessing the approximation power of neural networks. The Loss function comes into play here. Loss, or the output of the loss function, rises in tandem with the widening gap between the neural network's output and the ground truth. Two typical loss functions employed in neural networks are the quadratic cost and the cross entropy cost. Triangle-shaped loss function Due to the characteristics of variances, quadratic loss functions are frequently mathematically manageable. The loss is the same whether the inaccuracy is above or below the ground truth [26]:

$$\mathcal{L}_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \bar{Y}_i\right)^2$$

5.4 Cross entropy loss function:

Cross-entropy loss evaluates how well a classification model performs when producing a probability between 0 and 1. Cross-entropy loss penalizes both sorts of errors but is significantly more severe on the more inaccurate ones. It rises exponentially as the prediction output deviates from the truth and declines slowly as it gets closer to the truth. Cross-entropy has a unique quality that results from its mathematical logarithmic nature, as shown in its mathematical formula:

$$\mathcal{L}_{CE} = \frac{1}{N}\sum_{i} L_i = \frac{1}{N}\sum_{i} -\left[y_i \log\left(p_i\right) + \left(1 - y_i\right)\log\left(1 - p_i\right)\right]$$

5.5 Back Propagation

With the loss function defined to measure the approximation ability of a neural network, the learning problem for the neural network can be formulated as searching for such weights of a network that could given the loss function, reduce the network's loss to a minimum. In general, the loss functions in ANNs are non-convex functions. It's usually mathematically impractical to find a fixed expression for the local minima. Instead, the gradient of the loss function could be calculated and the local minimum could be found using gradient descent [26].

To spread the gradient of loss backwards to the former layers, the back-propagation algorithm [24] is used to compute the gradient efficiently via the chain rule in calculus. With back-propagation, the loss at the output could be propagated backwards so that the gradients of the hidden layer could be computed and the weights of each neuron could be subsequently updated.

5.6 Convolutional layer

In classic ANN, neurons in a middle layer are usually fully-connected with the inputs from the neurons in the previous layer. However, as mentioned in Sec.3.3.1, each connection between neurons contains a weight(parameter). In some applications like Image Classification, where the ANN takes images as input, the amount of parameters in an ANN would considerably explode, and consequently produce computing cost at an exceptionally huge level. For example, a single neuron in a fully-connected ANN would generate 12,228 parameters given an image with the size of 64 × 64 × 3(height, width, channel) as input [26].

To handle the parameter amount explosion problem ANN produces in image classification tasks, instead of creating weights for all dimensions of the input in neurons, CNN utilized a specific neuron named convolution kernel to slide over the image and grab feature maps out of the input in the convolutional layer [26].

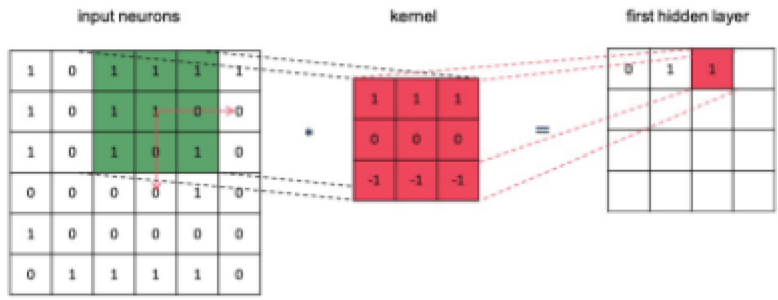

Figure 5.2: Convolutional Layer [26]

The convolution kernel, also named filter, is literally a weight vector shown in the above Figure.. In image classification tasks, the input image is vectorized at the beginning. During the convolving process, the kernel slides along the input image in both directions (columns and rows). For each slide, the kernel maps and multiplies a patch of input, which is a truncated vector, and subsequently generates an output value. Finally the convolving process produces an output feature map from the input. The weight of the filter is shared during the input calculation, which effectively reduces the amount of parameters utilized when handling a high-dimension input. Also, the convolution kernel has the shift invariance advantage over classic neurons when processing image inputs. The convolution operation has three parameters: kernel vector dimension, stride and padding. Stride controls the shifts for each step while padding zero-pads the border of the input. Padding is used to preserve the input dimension [26].

5.7 Pooling layer

The other component of CNN is the pooling layer [13]. The pooling function is what makes up the pooling layer, which is frequently positioned after the convolutional layer. The most popular pooling function is max-pooling, which takes the output of the convolutional layer and each N N block's maximum values are extracted (where N is the pooling filter's size). [26].

5.8 YOLO algorithms

Although it is seen as a routine task for the human brain, identifying things in an image would be more difficult for a machine. In computer vision, the process of locating and identifying objects in images is referred to as "object detection," and numerous methods have been developed in recent years to address the problem.One of the most well-known real-time object identification algorithms is YOLO (You Only Look Once).Redmond et al. first proposed it. [22].

The YOLO algorithm offers a high frame rate for real-time use combined with substantially superior performance on all aspects. Regression-based, the YOLO algorithm predicts classes and bounding boxes for the entire image rather than just the relevant portion in a single run. [11].

To begin, we must comprehend the actual predictions.Our ultimate objective is to predict an item's class and the object's location in a bounding box. Each bounding box is included in four categories..[11]:

1. Center of the box (bx, by)
2. Width (bw)
3. Height (bh)
4. Value c corresponding to the class of an object

We also project an actual number called pc that represents the likelihood that an object is present within the bounding box. [11].Instead of searching potential object-containing regions in the input image, YOLO divides the image into cells, usually in the form of a 19x19 grid. The next step is for each cell to forecast K bounding boxes. [11].

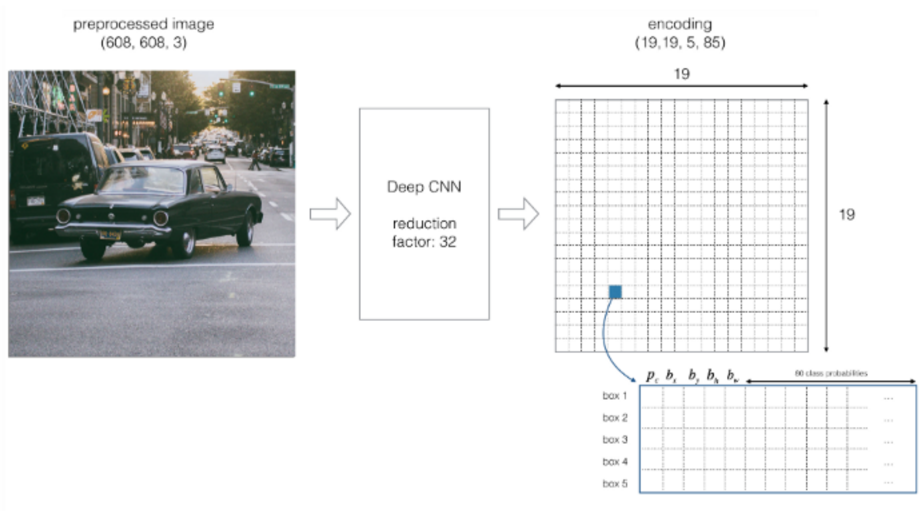

Figure 5.3: Processed image on YOLO [11]

Only when the anchor box's center coordinates fall within a given cell is an object regarded to be in that cell. The height and width of the image are always calculated in relation to the entire size, while the center coordinates are always calculated in relation to the cell because of this attribute. [11]. YOLO calculates the likelihood that a particular class is present in the cell during the first run of forward propagation. The highest probability class is chosen and placed in that particular grid cell.. The technique is the same for every grid cell in the image. [11]. The image may look like this after computing the probabilities for each class above:

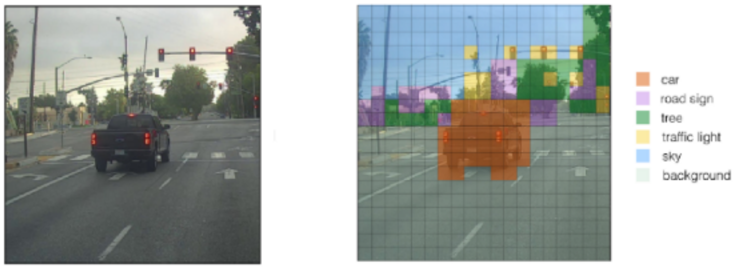

Figure 5.4: Before and after the class probabilities were predicted[11]

This displays the results of forecasting the class probabilities for each grid cell before and after. Non-max suppression comes next when class probabilities have been predicted. It helps the algorithm eliminate the unnecessary anchor boxes when multiple boxes are generated based on the class probabilities. [11].

Figure 5.5: Anchor boxes[11]

By executing the IoU (Intersection over Union) with the one with the highest class probability among them, non-max suppression reduces the extremely near bounding boxes in order to address this issue. [11].

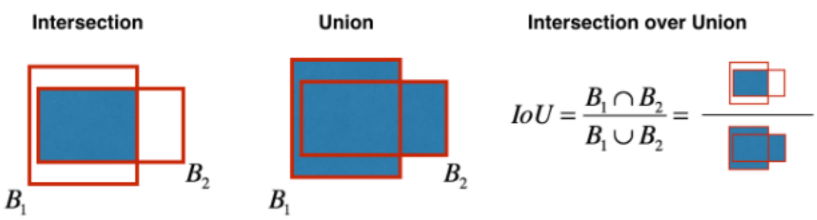

Figure 5.6: IoU Operation[11]

It compares each bounding box's IoU value to that of the one with the highest class probability before excluding those whose IoU value is higher than a certain threshold. It means that although those two bounding boxes are enclosing the same item, there is a low possibility that the other one is as well, so it is discarded. [11].

The algorithm then goes on to find the bounding box with the next highest class probability and does so again and again until all of the bounding boxes have been located. [11].

Figure 5.7: Before and After of Non-max suppression[11]

At this point, almost all of our work has been accomplished, and the algorithm generates the required vector displaying the characteristics of the bounding box for each class. The overall architecture of the algorithm is depicted in the diagram below.

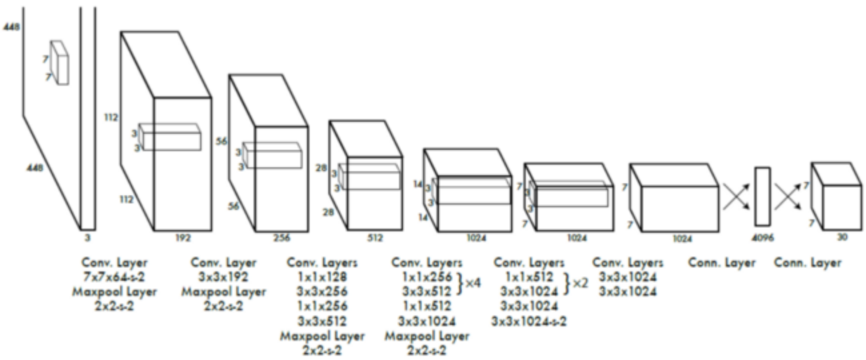

Figure 5.8: YOLO Architecture [11]

Moreover, the Loss function of the algorithm, which is its most crucial parameter. YOLO gains knowledge of all four parameters it predicts concurrently. [11].

5.9 Faster R-CNN

One of the most popular object detection designs, including YOLO (You Look Only Once) and SSD, uses convolution neural networks. Faster RCNN was developed by

Ross Girshick, Shaoqing Ren, Kaiming He, and Jian Sun in 2015 [18].

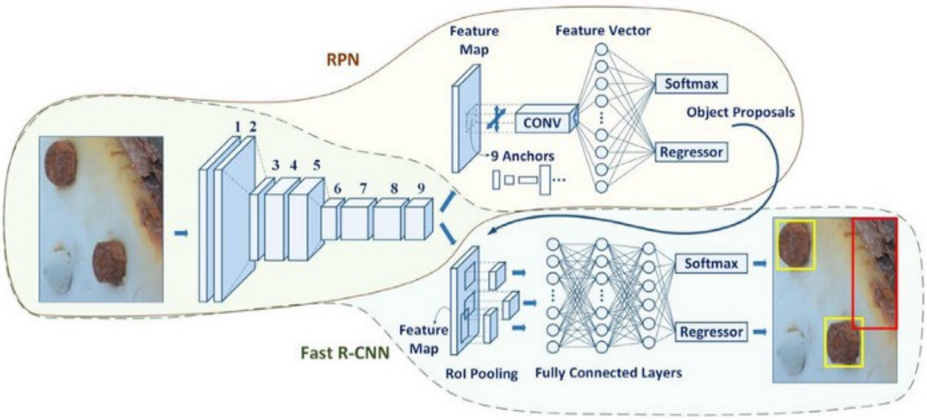

Figure 5.9: Faster R-CNN [18]

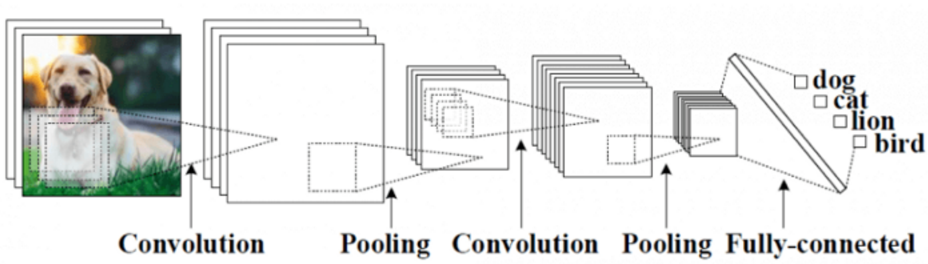

Figure 5.10: Computation of Convolutional Layer[18]

We compute convolution by swiping a filter along the length of our input image. The result is a feature map, a two-dimensional matrix. [18].

5.10 Region Proposal Network (RPN)

A little neural network known as an RPN uses the final feature map created by the convolution layers to forecast the presence or absence of objects and their bounding boxes.[18]. Faster R-CNN uses regional proposal network as stated below in the picture-

5.11 Feature Extraction Network

The input images are first fed into the Feature Extraction Network in Faster R-CNN (FEN). A pre-trained CNN model lacking fully-connected output layers, such as ResNet-50 or Inception V2, often makes up FEN. To create the feature map of the input image, the FEN simply uses its pretrained model. This feature map is then supplied into the Region Proposal Network and Classification Network. [16].

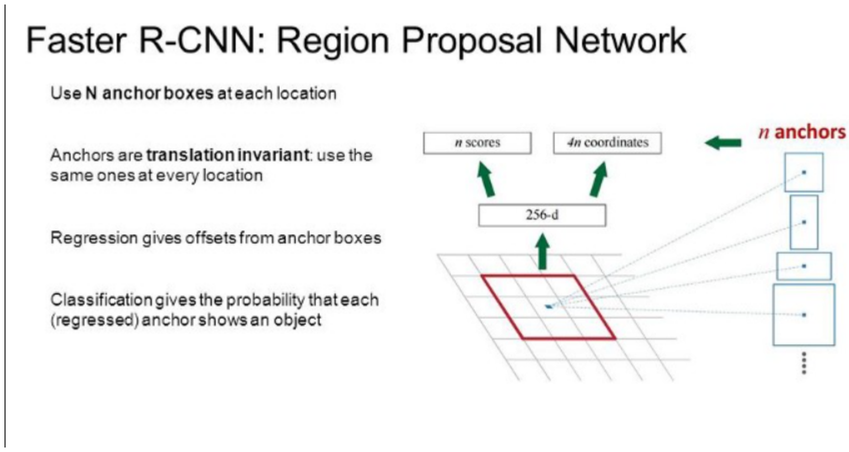

Figure 5.11: RPN[18]

5.12 Yolov3

YOLOv3 utilizes Darknet-53. as contrast to the preceding 19 layers, it uses 53 convolutional layers (ResNet-101 or ResNet-152) [26]. Darknet-53 is more effective than competing backbones and stronger than Darknet-19. In terms of mean average precision (mAP) and intersection over union (IOU) values, YOLOv3 is rapid and accurate. Compared to other detection techniques with comparable performance, it functions substantially faster. [26]. Cross-entropy loss in binary systems and independent logistic classifiers are used by the new YOLOv3 to predict classes during training. The multilabel technique used by YOLO v3 enables classes to be more detailed and to have several members for each bounding box.

5.13 Yolov4

By increasing the number of frames per second and mean average precision (mAP) by up to 10% and 12%, Yolov4 performs better than Yolov3.As can be seen in the image above, the four distinct blocks that make up the Yolov4 architecture are the backbone, the neck, the dense prediction, and the sparse prediction.

Similar to the ResNet architecture, the neck aids in the addition of layers between the dense prediction block (head) and the backbone. The input is aggregated to increase accuracy in the Yolov4 architecture using a modified Path aggregation network, a modified spatial attention module, and a modified spatial pyramid pooling. Pyramid pooling in space is shown in the above image.

The neck helps add layers between the dense prediction block (head) and the backbone, just like in the ResNet architecture.In the Yolov4 architecture, a modified Path aggregation network, a modified spatial attention module, and a modified spatial pyramid pooling are used to aggregate the input to improve accuracy.In the image above, pyramid pooling in space is depicted.

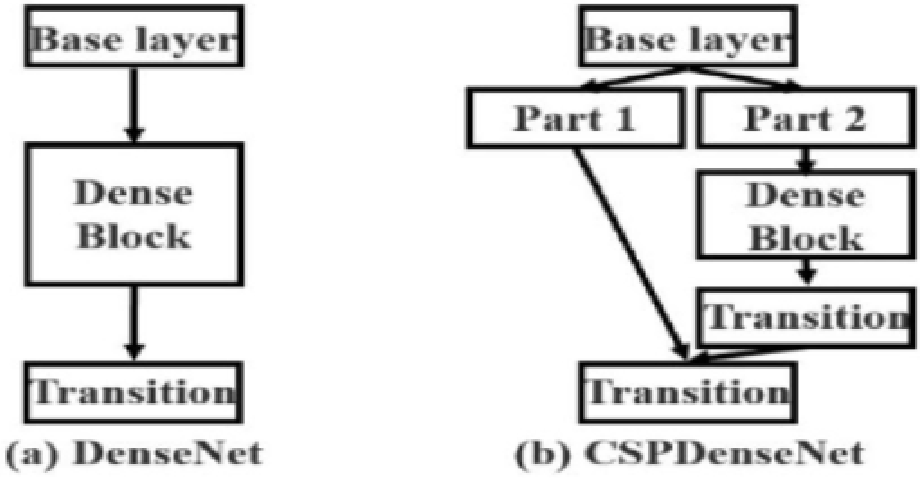

Figure 5.12: YOLOv4 Architecture [10]

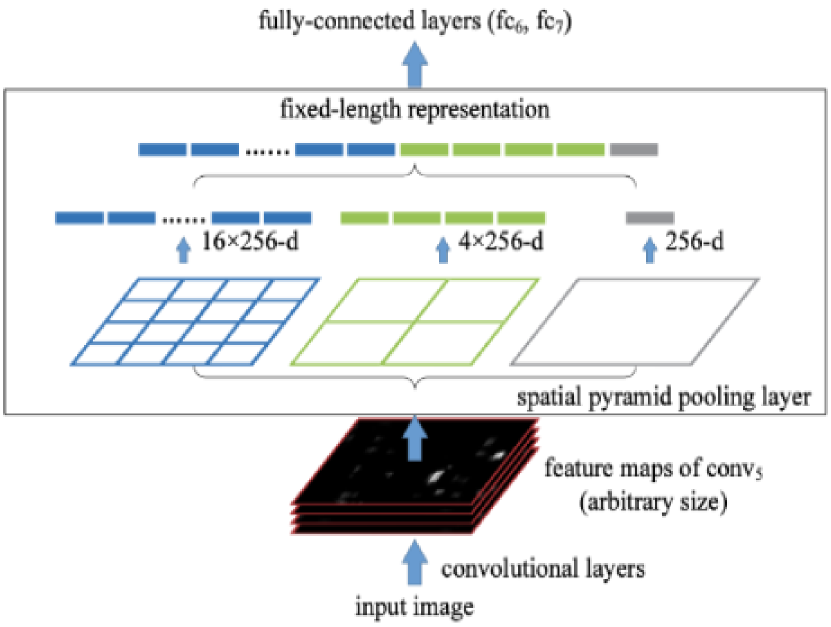

Figure 5.13: Modified path of YOLOv4 Architecture [10]

5.14 Yolov5

YOLO v5 is distinct from all prior releases because it is a PyTorch implementation rather than a Darknet clone.Similar to the YOLO v4, the v5 has a PA-NET neck and a CSP backbone.The two most important enhancements are mosaic data augmentation and auto learning bounding box anchoring. [15].

5.15 Yolov7

The most recent YOLO algorithm outperforms all earlier object detection algorithms and YOLO iterations in terms of speed and precision. It can be taught significantly faster on tiny datasets without any pre-learned weights than other neural networks and requires technology that is several times less expensive. As a result, it is anticipated that YOLO v7 will soon become the industry standard for object detection and surpass YOLO v4 as the current state of the art for real-time applications. [21]. The real-time object detection model for computer vision tasks that is fastest and most accurate is YOLOv7 [21]. The real-time object detection accuracy is significantly increased by YOLOv7 without raising the inference costs. As was previously demonstrated in the benchmarks, YOLOv7 effectively outperforms other well-known object detectors by reducing about 40% of the parameters and 50% of the computation required for state-of-the-art real-time object detections. This allows it to perform inferences more quickly and with higher detection accuracy.

We have chosen the YOLOv5 algorithm for our image dataset because of the accuracy and precision shown later in the paper.

5.16 Other algorithms and classifiers

For the sensor based data that we have collected, we have used Random Forest Algorithm, Decision tree algorithm, MLP and SVM algorithm to train our data and check the accuracy. As a classification algorithm for temperature data, we have used Logistic regression.

5.17 Random Forest

The well-known machine learning algorithm Random Forest is a component of the supervised learning strategy. It can be used to address issues with regression and classification in machine learning. It is built on the concept of ensemble learning, where different classifiers are joined to solve a challenging problem and enhance the performance of the model. [30].

Some decision trees may accurately predict the output, while others may not, because the random forest combines numerous trees to forecast the class of the dataset. When all of the trees are joined, however, the right result is projected. Thus, the

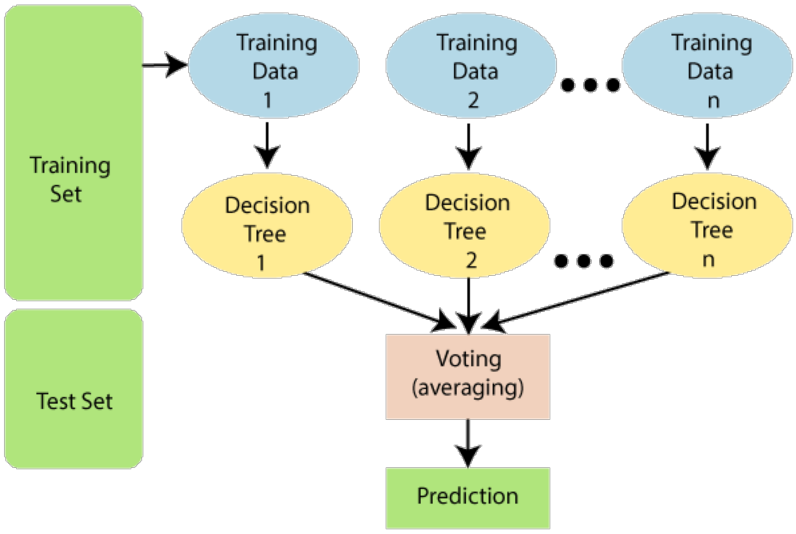

Figure 5.14: Random Forest Model[30]

two hypotheses for a better random forest classifier that are presented below. [30]:

1. The classifier will be able to come up with accurate predictions rather than guesses if there are some actual values in the dataset's feature variable.
2. Each tree's predictions must have very low correlations.

5.18 Support Vector Machine

One of the most well-liked supervised learning techniques for solving Classification and Regression issues is the Support Vector Machine, or SVM. It is used in machine learning, nevertheless, to deal with categorization problems. [31]. In order to swiftly categorize fresh data points in the future, the SVM algorithm seeks to identify the best line or decision boundary that can divide n-dimensional space into classes. An alternative name for this ideal decision boundary is a hyperplane. [31].

The extreme vectors and points that contribute to the formation of the hyperplane are chosen by SVM.The term "support vectors" refers to these extreme conditions, so the algorithm's name is "Support Vector Machine."Take a look at the following diagram, which divides the problem into two distinct groups: decision boundary and hyperplane. [31]:

SVM can be comprehended using the KNN classifier illustration we used.Let's say we see a strange cat that also looks like a dog if we want a model that can tell the difference between a cat and a dog.The SVM algorithm makes it possible for us to create such a model.We will first train our model with several photographs of cats and dogs so that it can become familiar with the various characteristics of cats and dogs before testing it with this strange animal.Consequently, the support vector will see the extreme case of cats and dogs when it selects extreme cases (support vectors) and draws a decision border between these two data (cat and dog).[31].

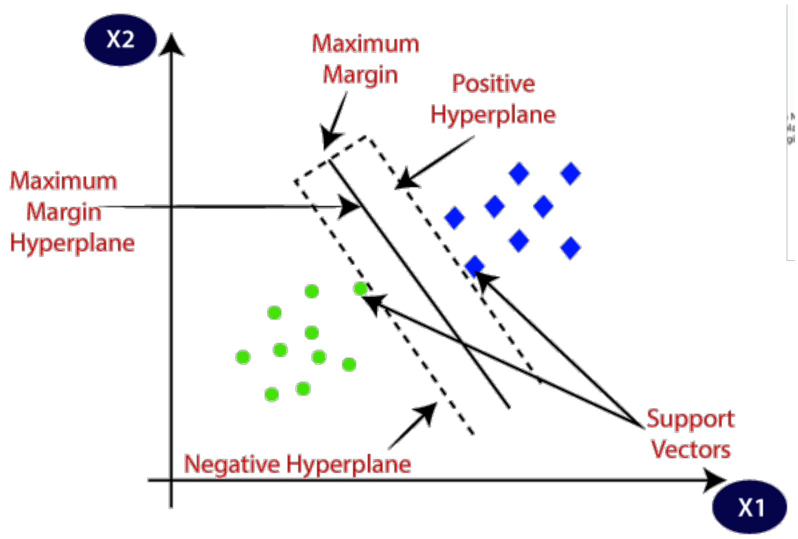

Figure 5.15: SVM graph[30]

5.19 MLP

Multi-layer Perceptron classifier is known as MLP, and the name itself refers to a neural network. In contrast to other classification algorithms like Support Vectors or Naive Bayes Classifier, MLPClassifier performs the classification process using an underlying neural network. [12]. For our dataset, we have classified the sensor data using MLP classifier to detect the change of gas levels in a time series.

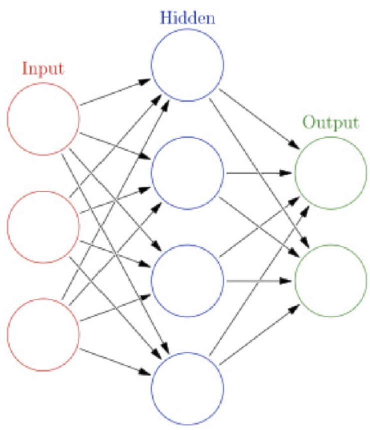

Figure 5.16: MLP architecture[12]

5.20 Decision Tree Algorithm

Continuous data segmentation based on a certain parameter is involved in decision trees and other supervised machine learning techniques. The decision nodes and the leaves of the tree are the two components that can be used to explain it. The options or results are represented by the leaves. The data is partitioned at the decision nodes. A supervised learning algorithm is a decision tree. It is a visual representation of every potential answer. All choices were based on a set of circumstances.[23].

First, we created a model by importing DecisionTreeClassifier from sklearn. Then we took entropy as a criterion for it. Then using our dataset, we trained the model and validated the data. Then we predicted the method to predict the outcome and check accuracy.

Chapter 6

Data Representation

6.1 Data handling

6.1.1 Dataset creation

We have tested our dataset against different algorithms to detect and measure accuracy of the models. To achieve a robust YOLO model, it is recommended to train with over 1500 images per class, and more then 10,000 instances per class. It is also recommended to add up to 10% background images, to reduce false-positives errors [31]. For our customized dataset, we have used **1716 images** in which **1.2k annotated images** were trained and **342 images** were validated. For sensor data, we have collected real time data using sensors and creating a manual atmosphere for collecting data and completing the dataset. The gas sensor dataset contains **118** real time sensor data. All of the numerical data were tested for accuracy. The temperature sensor dataset contains **626** real time data tested against two algorithmic models for accuracy.

6.1.2 YOLO labeling format

Most annotation platforms offer a single annotation text file per image when exporting in the YOLO labeling format. Each image object has a bounding-box (BBox) annotation in its own text file. The annotations, which range from 0 to 1, are displayed normalized to the size of the image. The structure for presenting them is as follows. [24]:

$$\langle object-class-ID\rangle\langle \text{X center }\rangle\langle \text{Y center }\rangle\langle \text{ Box width }\rangle\langle \text{ Box height }\rangle :$$

6.2 Bounding boxes

Bounding boxes are the most common type of annotation in computer vision.The target object's location is specified by means of rectangles known as bounding boxes.It is possible to identify these coordinates by looking at the rectangle's upper-left and lower-right x and y axes.Bounding boxes are typically used in activities related to object detection and localization. [13].

6.3 Model training

For our image dataset we have collected **1716 images**.We have annotated all the images and our main focus was to accurately detect a person in a still picture. We have trained our image dataset in YOLOV3, YOLOV4, YOLOV5 and YOLOV7. After that we have compared the results and accuracy of all training models. Our image dataset included occluded images of human body parts for partial detection.

For our temperature sensor dataset, We have trained our dataset using the machine learning model and got results using the Logistics regression classifier. We had 626 real time data collected through our temperature sensor.

For our gas sensor dataset. We have used SVM, Neural network, Random forest algorithm to train our dataset and compared the accuracy for the dataset against all the algorithms. We had 118 real time data collected through our gas sensor.

Scalar metrics graphs for the trained datasets of image and sensors are shown below-

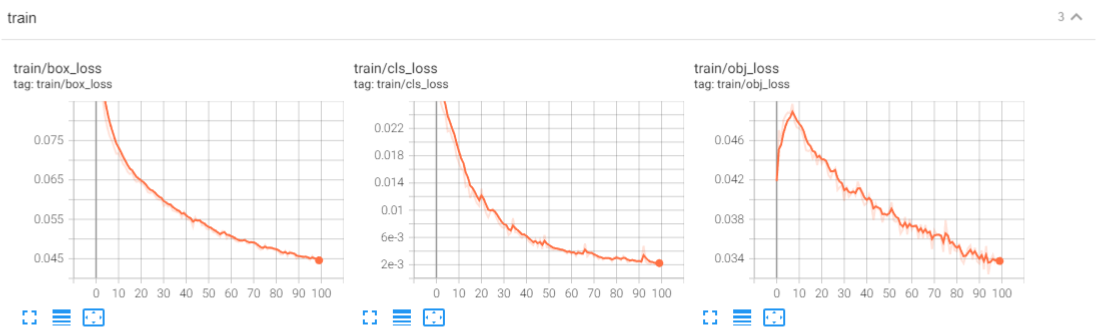

Figure 6.1: Train Loss Graph

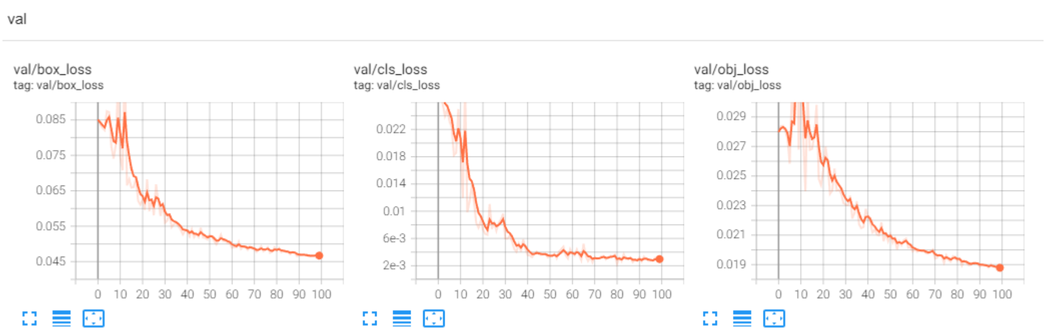

Figure 6.2: Validation Loss Graph

In the graph we can clearly see that, out of 1716 predictions, the YOLO v5 model has given 85% true predictions and 53% wrong predictions about human detection.

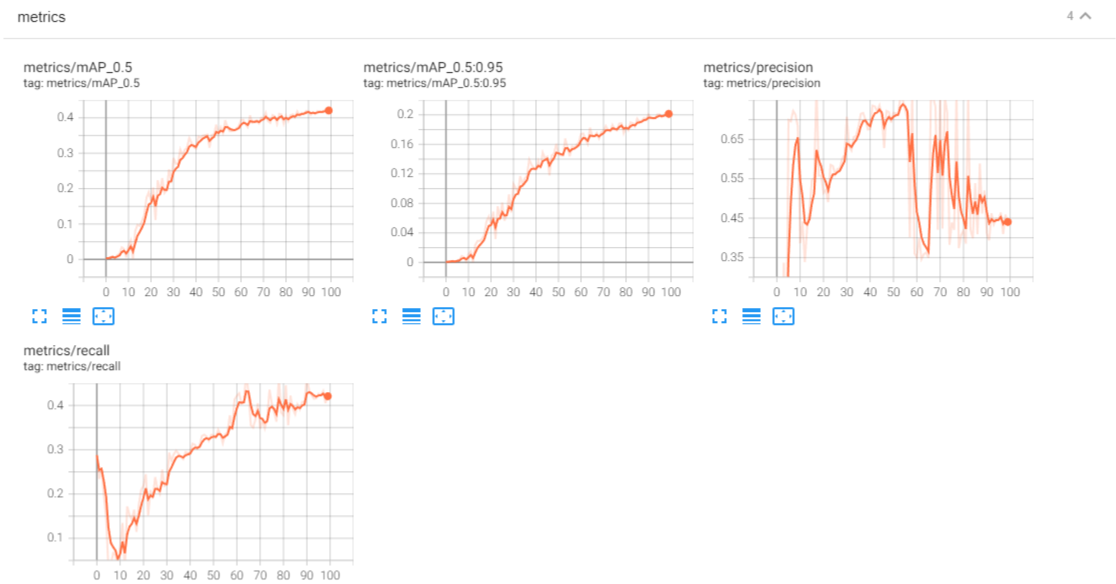

Figure 6.3: Metrics Graph of YOLOv5

6.4 Model Evaluation

6.4.1 The train/test/validation split

To properly evaluate the model, we have split our data as a portion of 70 percent for training, 10% for testing, and 20% for validation. This was essential for ignoring the overfitting of the training set. The validation set was used to evaluate the model while building and tuning the model. We also shuffled the data quite a few times to avoid similarity in the result and present our data.

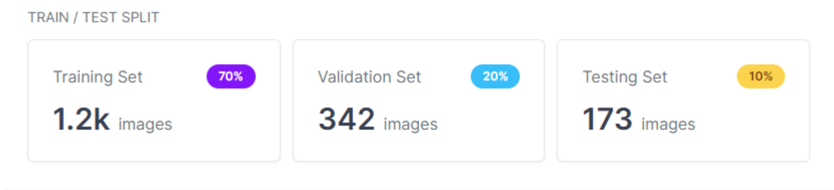

Figure 6.4: Train/Test/Validation split

6.4.2 Classification metrics

We discovered four parameters when attempting to determine categorization metrics. Which are:

- True positive
- True negative
- False positive
- False negative

When an observation genuinely belongs to a class, the true positive result happens. A real negative result happens when an observation doesn't fit into a category. False positive results happen when an observation appears to belong to a class when in fact it does not. Additionally, a false negative result is produced when an observation is made that appears to not belong to a class but actually does.

This classification metrics are known as confusion matrix and for our trained model of YOLOv5, the confusion matrix we have figured looks like below-

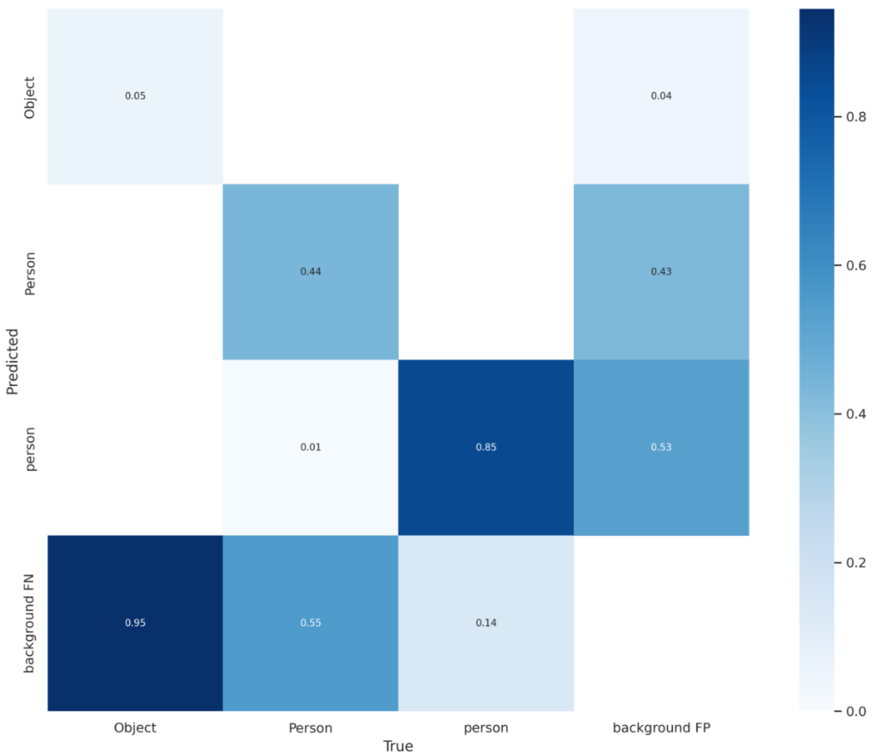

Figure 6.5: Confusion Matrix - YOLO v5

The colors in the above figure state the four metrics discussed earlier. From our YOLOv5 model confusion matrix, we can see that for our dataset, we have obtained 85% of true positives which means that our model is well trained with annotated data.

6.4.3 Accuracy

The percentage of predictions made using test data that were accurate is what is meant by accuracy. The formula is simple: divide the total number of guesses by the total number of correct forecasts. [7].

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{all predictions}}$$

For our implemented model, the accuracy was 0.82%.

6.4.4 Precision

Precision is the percentage of instances out of all the examples that were predicted to belong to a certain class that are actually relevant (also known as true positives). [7].

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

For our implemented model, the precision was 0.78%.

6.4.5 Recall

The ratio of examples correctly predicted to belong to a class to the total number of actual examples in the class is called recall. [7].

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

For our implemented model, the recall was 0.82%.

In circumstances where classes are not evenly distributed, precision and recall are helpful [7]. For our dataset, the classes were not evenly distributed and thus precision and recall helped to calculate the f-score.

$$F_\beta = \left(1 + \beta^2\right) \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

Figure - f-score equation

From the above image, we can also see that, with more samples of data, precision and recall values change and vary.

6.4.6 Regression metrics (Sensor data)

Because we are now predicting in a continuous range rather than a discrete number of classes as we did for classification models, evaluation criteria for regression models are considerably different from those we outlined before for classification models. [7]. After collecting the data from the sensors we have run our data with MLP classifier, Random Forest classifier, Logistic Regression classifier and SVM classifier. After training our data we have detected pre ppc and post ppc comparison on the test data. We have also measured the accuracy for all the algorithms tested through our dataset. After comparing the algorithms we have successfully measured our sensor datasets and we were able to determine the comparison between the algorithmic models. The graph below shows the comparison between algorithms pre ppc and post ppc data of the gas sensors-

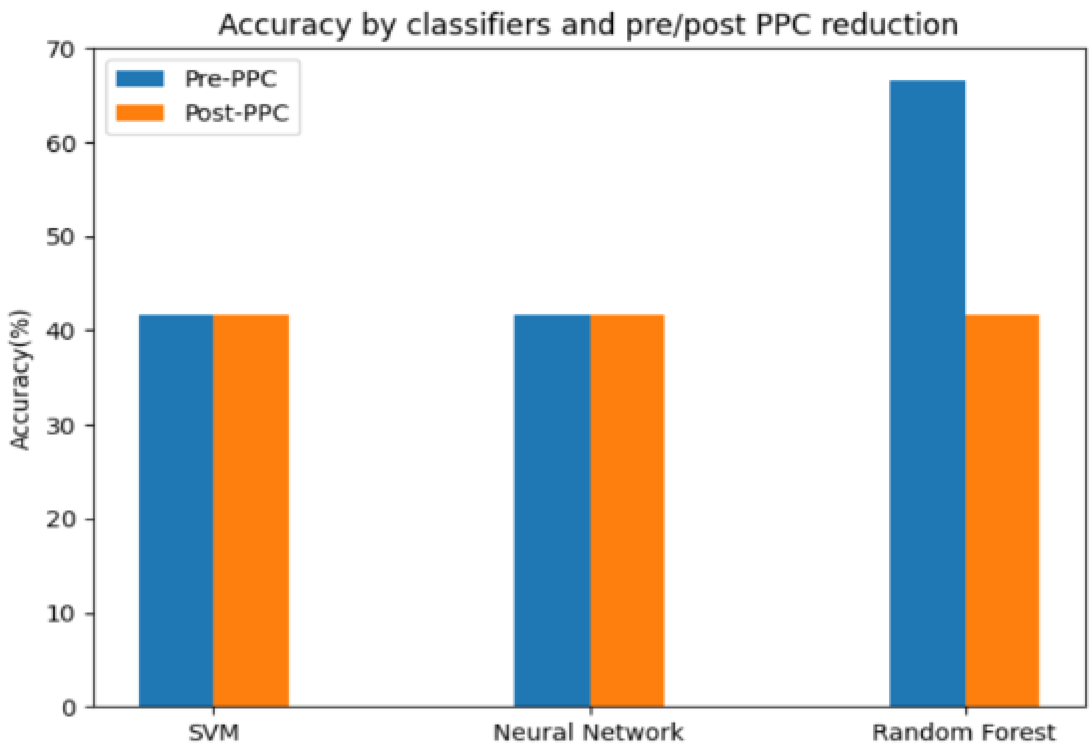

Figure 6.6: Comparison Result of gas sensor

Chapter 7

Results and Analysis

7.1 Detection results

7.1.1 YOLO v5

YOLO v5 algorithm was used to detect the categories of data in the dataset. The following table shows the data precision and recall measured by the algorithm.

```
100 epochs completed in 0.444 hours.
Optimizer stripped from runs/train/yolov5s_results/weights/last.pt, 14.8MB
Optimizer stripped from runs/train/yolov5s_results/weights/best.pt, 14.8MB

Validating runs/train/yolov5s_results/weights/best.pt...
Fusing layers...
custom_YOLOv5s summary: 232 layers, 7251912 parameters, 0 gradients, 16.8 GFLOPs
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 11/11 [00:04<00:00,  2.38it/s]
                 all        342       1173      0.428      0.412      0.422      0.204
              Object        342         73      0.137     0.0411     0.0702     0.0338
              Person        342        243      0.368       0.37      0.334      0.123
              person        342        857      0.777      0.825       0.86      0.456
Results saved to runs/train/yolov5s_results
CPU times: user 23.7 s, sys: 3.03 s, total: 26.8 s
Wall time: 27min 21s
```

Figure 7.1: YOLO v5 detection results

The detection results of YOLO v5 were much improved than other versions of YOLO. The graphs of recall and precision are shown below-

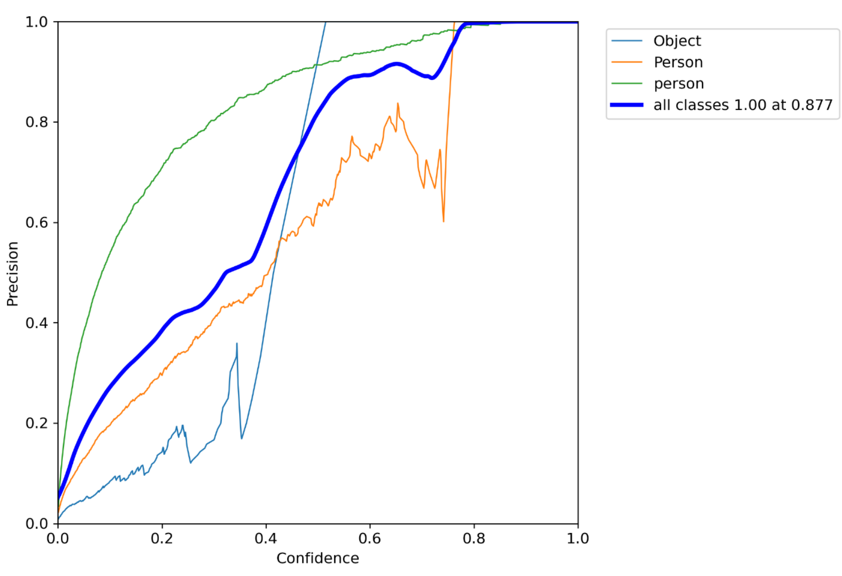

Figure 7.2: Precision curve

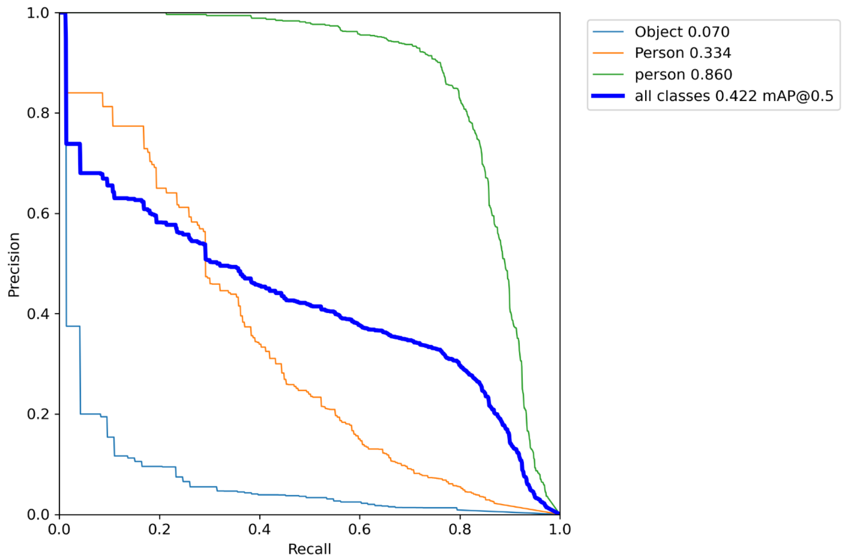

Figure 7.3: Precision curve

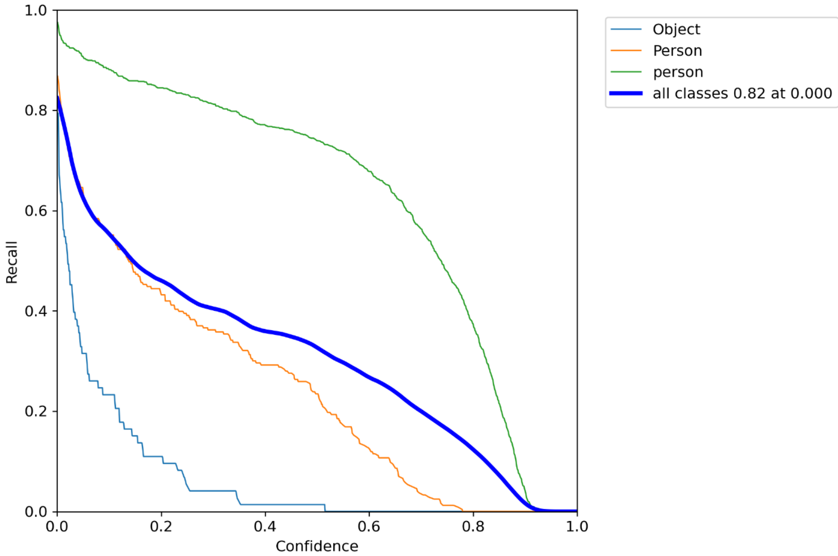

Figure 7.4: Recall curve

The F1 graph for the overall classes is shown below-

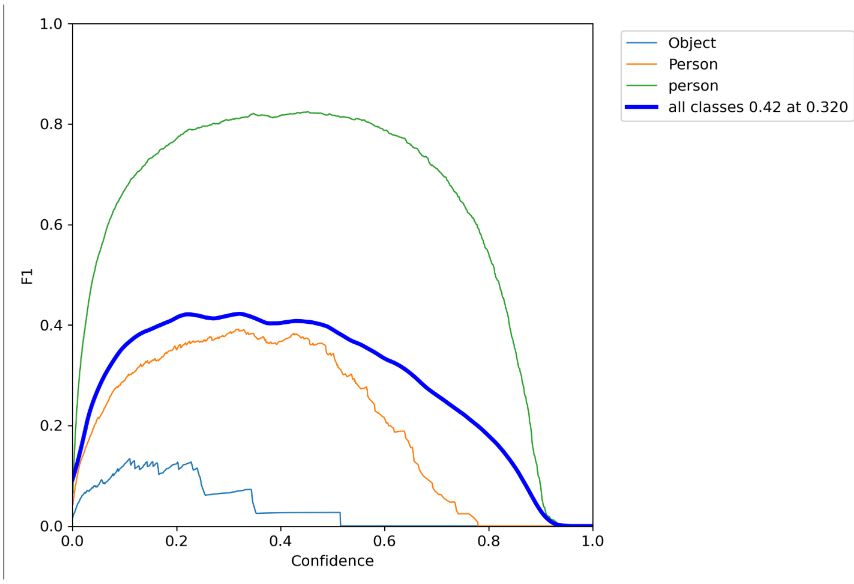

Figure 7.5: F1 curve

The graph for Feature extraction of YOLO v5 is shown below -

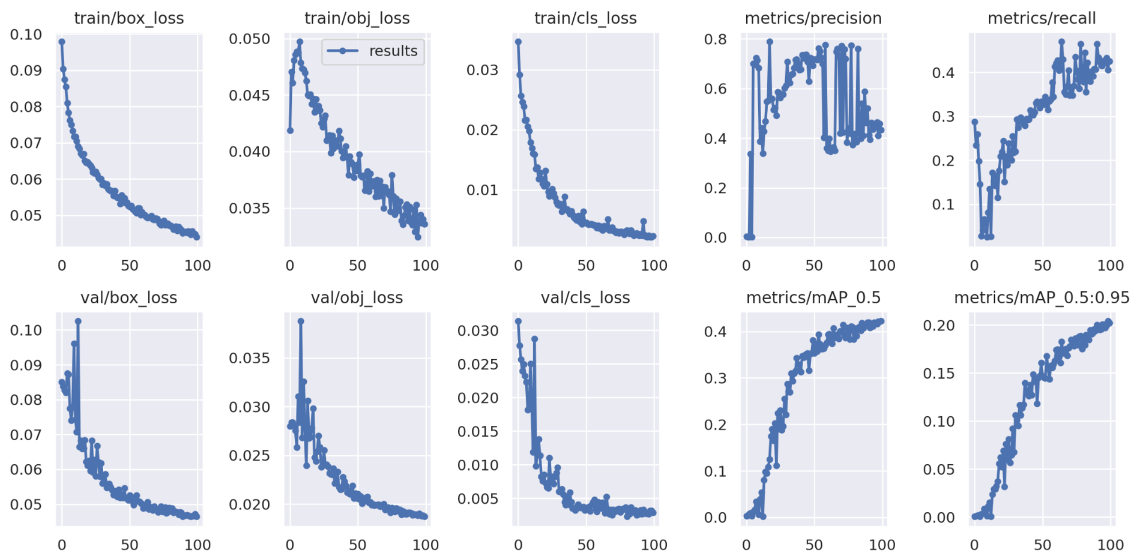

Figure 7.6: Feature extraction graphs

1. **Box loss/Mean squared error:** Mean squared error means how close the regression line is to the data points. From the above graph we can clearly see that, as the sample of data has increased over time, the regression line was closer to the data points than the initial position it started from.
2. **Object loss/Binary cross entropy:** Binary cross entropy shows the confidence of object presence is the object loss.
3. **Class loss/Cross entropy:** The class loss shows the classification loss for the trained model.

Since our image dataset has three classes, the classification error is not zero and there are class misidentifications.

The scalar metrics parameters are shown in the graphs below-

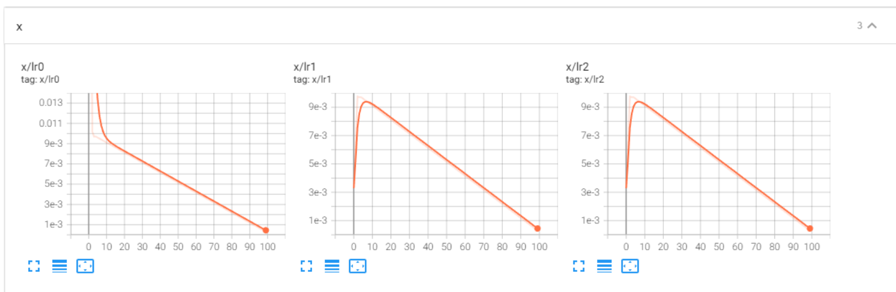

Figure 7.7: Scalar matrix graph

The detected data with bounding boxes are shown below for YOLOv5-

Figure 7.8: YOLOv5 detection

Figure 7.9: Detection of Human Body Part on YOLOv5

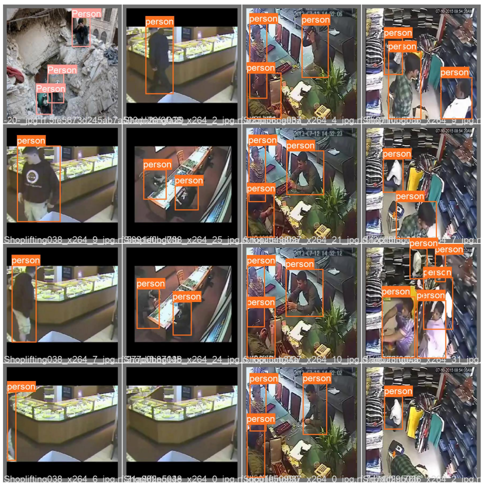

Figure 7.10: Ground Truth Training Data on YOLOv5

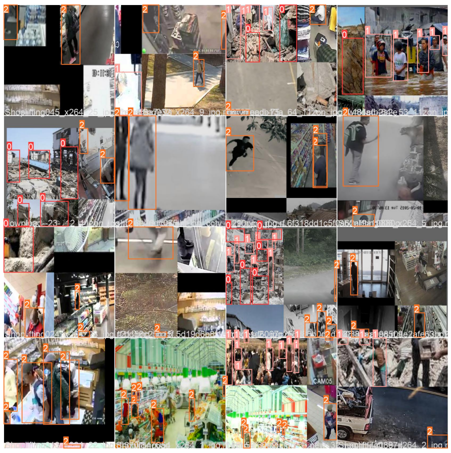

Figure 7.11: Ground Truth Augmented Training Data on YOLOv5

We chose YOLOv5 as our implemented algorithm and the detection results are shown above. For other algorithms that we have tested (YOLOv3, YOLOv4, YOLOv7) the detection results are shown below-

7.1.2 YOLOv4

The detection accuracy for YOLOv4 was 67% maximum out of all the trained data. The image below shows the accurate detection of a person and object through YOLOv4.

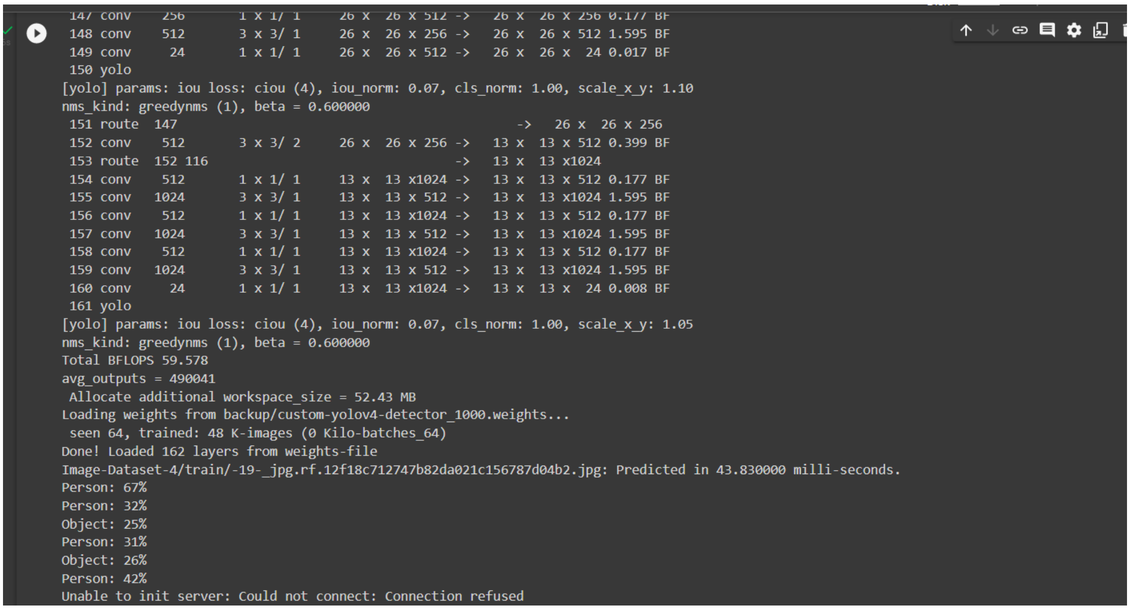

Figure 7.12: YOLOv4 detection result

The precision and recall graph for YOLO v4 is shown below-

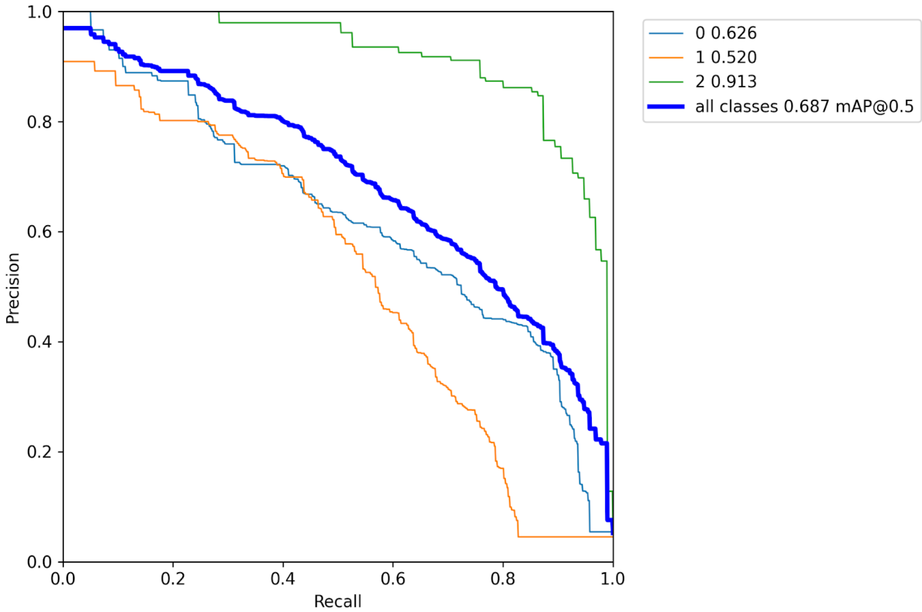

Figure 7.13: Precision and recall graph

The following images show the detected person through YOLOv4-

Figure 7.14: YOLOv4 detection

Figure 7.15: YOLOv4 detection

As the detection accuracy for YOLO v4 was 0.67, we did not proceed with the YOLOv4 model for our image dataset.

7.1.3 YOLOv3

The detection accuracy for YOLOv3 was 53% maximum out of all the trained data. The image below shows the accurate detection of a person and object through YOLOv3. The detection result for YOLO v3 is shown below-

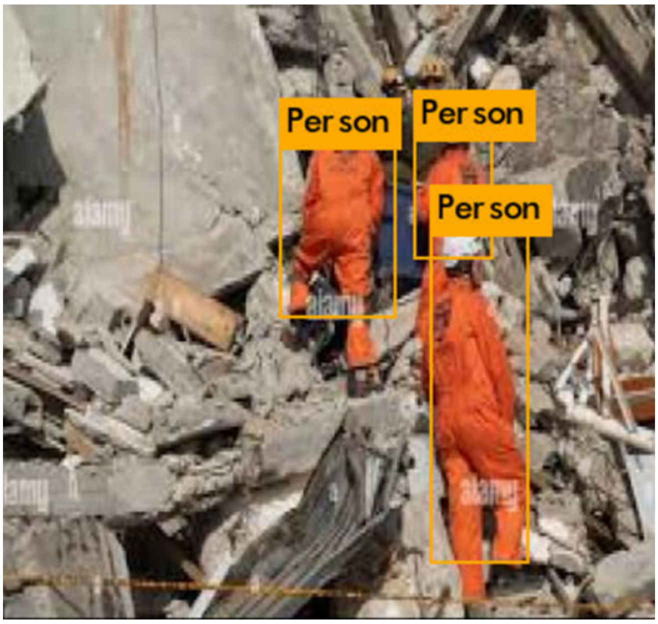

Figure 7.16: YOLOv3 detection

Figure 7.17: YOLOv3 detection

Since the detection accuracy for YOLOv3 was 53% maximum out of all the trained data, we did not proceed with the YOLO v3 algorithm for detection results.

7.1.4 Yolov7

We have tested our data against YOLOv7 and the accuracy for the trained data was 67%. The algorithm detected the images faster than other algorithms but the mAP was less than YOLOv5 for our dataset.

The following images show the precision and recall graph for YOLOv7-

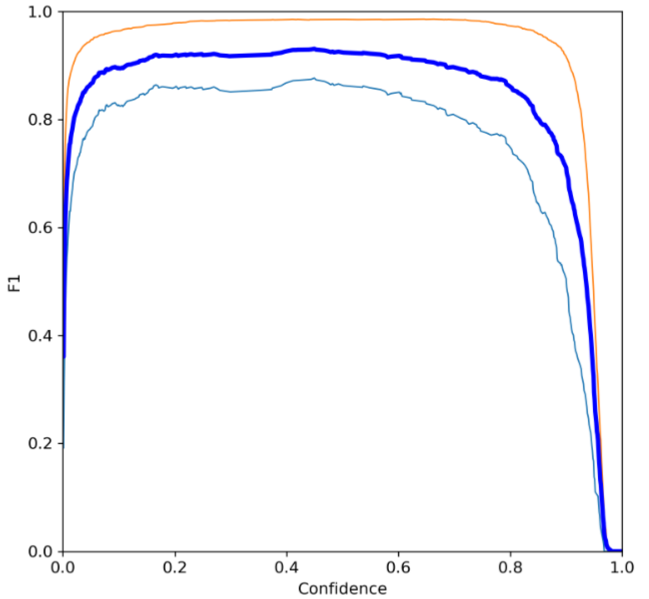

Figure 7.18: F1 graph for YOLOv7

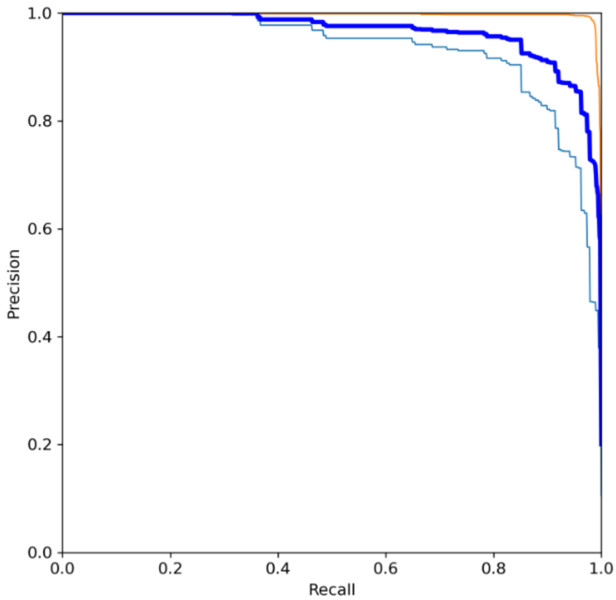

Figure 7.19: Recall and precision graph for YOLOv7

The following images show the detected images from YOLOv7-

Figure 7.20: YOLOv7 detection

The precision and accuracy was tested 67%, so we did not proceed with the YOLO v7 detection.

Figure 7.21: YOLOv7 detection

7.1.5 Sensor data analysis

After comparing the gas sensor data on different algorithms, we have found the accuracy for our gas sensor dataset. The accuracy for training data for logistic regression on training data was 63% and on test data the accuracy was 61%.

On the other hand, the accuracy for the decision tree algorithm for the gas sensor dataset was 58% on test data. Compared to logistic regression, the accuracy showed less value.

For the temperature sensor dataset, the accuracy measured for trained data was 0.067 and the accuracy measured for tested data was 0.063.

7.1.6 Performance

As per the data collected and graphs analyzed, we can clearly say that YOLO v5 had better detection results compared to other versions of YOLO. YOLO v5 was significantly faster and for our dataset, it provided real time results at a quick amount of time. It was faster than other YOLO algorithms for our dataset. Also the number of successfully detected classifiers are more than other YOLO models with strong mAP. The mAP recorded for the YOLOv5 against all classes was 0.5. For YOLO v7 the detection accuracy was 67%. For YOLO v4 the accuracy tested was 67% for our dataset. For YOLOv3 the accuracy tested was 53%. And finally the accuracy for YOLO v5 detected was 78% for our dataset.

Also, our collected sensor dataset had a decent accuracy above 60% in the logistic regression model compared to the decision tree model which gave an accuracy of 58% only. In a real life environment the accuracy could go upto 70% if the proper system is implemented with proper algorithms.

YOLOv3	YOLOv4	YOLOv5	YOLOv7
53%	67%	78%	67%

Table 7.1: Image data comparisons

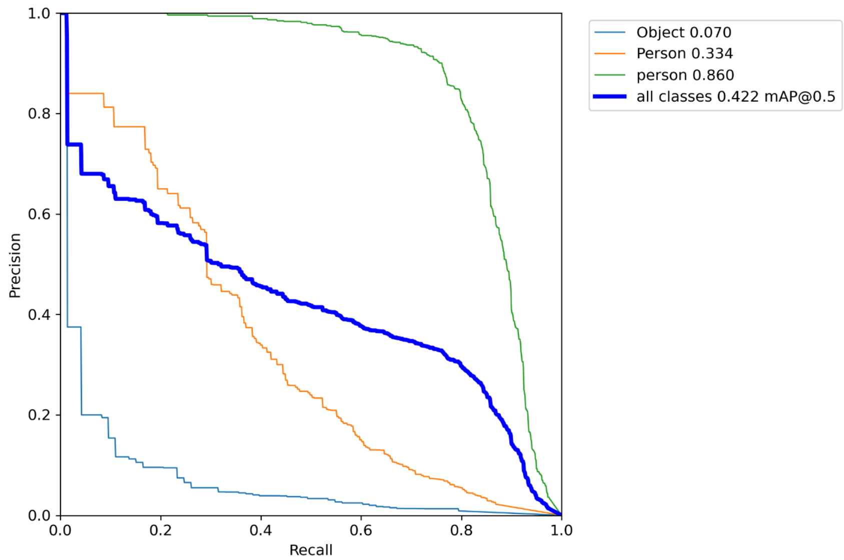

Figure 7.22: PR curve(YOLOv5)

7.1.7 Validity threat analysis

Our machine learning model is built on 1716 images. Although we have collected a significant amount of data for building our model, still disaster management datasets are difficult to find. The scarcity of data specifically, disaster data were hardly available on any given platform. As for the sensor data, the data were easy to collect since we have implemented hardware based systems for collecting real time data. Our image dataset was precisely split between train, test, and validation and the results through different algorithms were very much promising for getting accurate results. We have got accuracy above 80% through our dataset and using YOLO v5 as the primary algorithm. As for the sensor datasets, the data was accurate above 60% using different classifiers and accuracy testing algorithms. The total ratio of loss function lessened with each timeframe.

Chapter 8

Multimodal Prediction

Our proposed system integrates the image detection and sensors for performing in a single state in complex environments. We have thought about the situations where humans are helpless and would want to get help as soon as possible. In those situations, a single detection based system can not provide all the data that we need. If we detect a human being trapped inside or under a collapsed structure through the detection system, next we would want to see a visualization of the situation. Our image detection system would provide the data for the required visualization. Next after getting the detection results, we have to know what is the surrounding atmosphere of the victim and if it is too hazardous for the victim. If the atmosphere is hazardous, we need to act fast and send a rescue team as soon as possible. Our gas sensor system will detect the amount of hazardous gasses in the atmosphere and will let us know what measurements we have to take to save the victim as well as keep the lives of the rescue team safe. If the temperature around a victim is high, there is a lower chance of survival for the victim. Our temperature system detects the rise of temperature and lets us know what safety measurements need to be taken to save that victim as soon as possible. In a complex situation, nothing certain can be known about any victim until and unless the detection parameters are checked properly. Our proposed methods state a multimodal system so that we can be ready in any given situation in complex environments. Although we have not been able to integrate the systems altogether yet, we are working towards the systematic completion and integration for all our proposed and individual tested systems.

Chapter 9

Conclusion and Future Research

9.1 Conclusion

Human beings have the ability to think and to develop the resources that they possess for the betterment of themselves. While in most of the fields higher progress is visualized, in the field of disaster management and efficiently solving its aftermath consequences, only a little progress can be seen. Our proposed technology consisting of Artificial intelligence and Machine learning will be able to detect human beings successfully penetrating the barriers and obstacles where humans cannot reach. Our study includes datasets that are trained on different algorithm models and the datasets are tested with accuracy. And by comparing the outcomes of other models, we can conclude that the YOLOv7 algorithm using temporal information outperforms other algorithms in terms of human detection accuracy. Now with the preliminary developments and internal system design we hope to move ahead with our proposed system and complete its functionality soon by implementing the results and calculations. We want to contribute to the global community and help humanity to stay one step ahead in the field of disaster management using technological advancement.

9.2 Future work

Technology is being used in every sector nowadays starting from making intact products, making high performance softwares, detecting diseases, contributing in astrophysics, managing banks, managing companies, etc. In the field of disaster management, technology can be used to a great extent as well. We believe our proposed thesis theory can bring significant change in disaster rescue missions and it can be developed further with motion detection sensors, radar technologies, and air/land based vehicle systems. Also, methods like color distortion, image scaling, binarization can be applied to get more accurate results on the datasets. We have compared different algorithms on our datasets and on five different human organs. More variants of organs can be tested with the algorithms to know more about how this system responds to different situations. We would like to keep exploring this field and pursue our research interest to gather more accurate data for precise detection of human lives in any disastrous situation.

Bibliography

[1] C.-T. Lin, L. Siana, Y.-W. Shou, and T.-K. Shen, "A conditional entropy-based independent component analysis for applications in human detection and tracking," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 1–14, 2010.

[2] M. Correa, G. Hermosilla, R. Verschae, and J. Ruiz-del-Solar, "Human detection and identification by robots using thermal and visual information in domestic environments," *Journal of Intelligent & Robotic Systems*, vol. 66, no. 1, pp. 223–243, 2012.

[3] M. Paul, S. M. Haque, and S. Chakraborty, "Human detection in surveillance videos and its applications-a review," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, pp. 1–16, 2013.

[4] F. Awad and R. Shamroukh, "Human detection by robotic urban search and rescue using image processing and neural networks," *International Journal of Intelligence Science*, vol. 2014, 2014.

[5] V. Rathore, "Technology in disaster management and disaster risk reduction: A review of applications," *Technology*, vol. 6, no. 4, 2016.

[6] H. Gao *et al.*, "Understand single shot multibox detector (ssd) and implement it in pytorch," *Available online: medium. com/@ smallfishbigsea/understand-ssd-and-implement-your-own-caa3232cd6ad (accessed on 25 February 2019)*, 2018.

[7] J. Jordan, *Evaluating a machine learning model.* Aug. 2018. [Online]. Available: https://www.jeremyjordan.me/evaluating-a-machine-learning-model/.

[8] H. Ritchie and M. Roser, "Natural disasters. our world in data," *The BE Journal of Economic Analysis & Policy*, vol. 8, no. 1, pp. 1935–1682, 2019.

[9] W. Tai, B. Ilias, S. A. Shukor, N. A. Rahim, and M. Markom, "A study of ultrasonic sensor capability in human following robot system," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 705, 2019, p. 012 045.

[10] Ampadu, *Yolov3 and yolov4 in object detection*, Sep. 2020. [Online]. Available: https://ai-pool.com/a/s/yolov3-and-yolov4-in-object-detection?fbclid=IwAR3cC4Dcn3LjWeEXrtEUx-J2vzMT2XrIb-oNBX8JUypPXSVdbfDjTkFao3g.

[11] Manishgupta, *Yolo-you only look once*, May 2020. [Online]. Available: https://towardsdatascience.com/yolo-you-only-look-once-3dbdbb608ec4#:%5C%7E:text=YOLO%5C%20%20algorithm%5C%20is%5C%20an%5C%20%20algorithm,one%5C%20run%5C%2.

[12] A. Nair, *A beginner's guide to scikit-learn's mlpclassifier*, Nov. 2020. [Online]. Available: https://analyticsindiamag.com/a-beginners-guide-to-scikit-learns-mlpclassifier/.

[13] S. Pokhrel, *Image data labelling and annotation-everything you need to know*, Mar. 2020. [Online]. Available: https://towardsdatascience.com/image-data-labelling-and-annotation-everything-you-need-to-know-86ede6c684b1.

[14] Singh, *Mq2 gas sensor circuit diagram and it's working hackatronic*, Nov. 2020. [Online]. Available: https://hackatronic.com/mq2-gas-sensor-circuit-diagram-and-its-working/.

[15] C. Supeshala, *Yolo v4 or yolo v5 or pp-yolo?* Aug. 2020. [Online]. Available: https://towardsdatascience.com/yolo-v4-or-yolo-v5-or-pp-yolo-dad8e40f7109.

[16] W. Zhan and M. Zheng, *From body parts responses to underwater human detection: A deep learning approach*, 2020.

[17] E. Burns, *What is machine learning and why is it important?* Mar. 2021. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML#:%5C%7E:text=Machine%5C%20learning%5C%20(ML)%5C%20is%5C%20a,to%5C%20predict%5C%20new%5C%20output%5C%20valu.

[18] A. KHAZRI, *Faster rcnn object detection*, Oct. 2021. [Online]. Available: https://towardsdatascience.com/faster-rcnn-object-detection-f865e5ed7fc4.

[19] N. Klingler, *Why computer vision is difficult to implement? (and how to overcome)*, Sep. 2021. [Online]. Available: https://viso.ai/computer-vision/why-computer-vision-is-difficult/.

[20] M. Mandal, *Cnn for deep learning: Convolutional neural networks*, Jul. 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/.

[21] G. Boesch, *Yolov7: The most powerful object detection algorithm (2022 guide)*, Aug. 2022. [Online]. Available: https://viso.ai/deep-learning/yolov7-guide/.

[22] S. Goled, *How does yolov6 compare against yolov5?* Jul. 2022. [Online]. Available: https://analyticsindiamag.com/how-does-yolov6-compare-against-yolov5/.

[23] A. K, *Decision tree machine learning algorithm*, Mar. 2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2022/01/decision-tree-machine-learning-algorithm/.

[24] P. Lihi Gur Arie, *The practical guide for object detection with yolov5 algorithm*, Apr. 2022. [Online]. Available: https://towardsdatascience.com/the-practical-guide-for-object-detection-with-yolov5-algorithm-74c04aac4843.

[25] *Lm35 temperature indicator led circuit*, Sep. 2022. [Online]. Available: https://theorycircuit.com/lm35-temperature-indicator-led-circuit/.

[26] V. Meel, *Yolov3: Real-time object detection algorithm (guide)*, Aug. 2022. [Online]. Available: https://viso.ai/deep-learning/yolov3-overview/.

[27] L. Ada, *Pir motion sensor*. [Online]. Available: https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor.

[28] *Artificial neural network*. [Online]. Available: https://www.sciencedirect.com/topics/neuroscience/artificial-neural-network.

[29] *Artificial neural network tutorial - javatpoint*. [Online]. Available: https://www.javatpoint.com/artificial-neural-network.

[30] *Machine learning random forest algorithm - javatpoint*. [Online]. Available: https://www.javatpoint.com/machine-learning-random-forest-algorithm.

[31] *Support vector machine (svm) algorithm - javatpoint*. [Online]. Available: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm.

[32] *What is transfer learning? examples newbie-friendly guide*. [Online]. Available: https://www.v7labs.com/blog/transfer-learning-guide.