Autonomous Water Turret (IoT)

A PROJECT REPORT

Submitted by

BUVICA M [1RVU23CSE123]
GOPIKA R [1RVU23CSE170]
MOHAN CHANDRA SS [1RVU23CSE282]
NEERAJ P RAO [1RVU23CSE303]

as part of the ***Semester End Examination***

for the course

CS2113 AI for IoT

Offered in
Semester IV
of
B.Tech (Hons)

School of Computer Science and Engineering
RV University
RV Vidyaniketan,8th Mile, Mysuru Road, Bengaluru, Karnataka, India - 562112

MAY & 2025

DECLARATION

I, **Buvica M (1RVU23CSE123), Gopika R(1RVU23CSE170), Mohan Chandra SS (1RVU23CSE282), Neeraj P Rao (1RVU23CSE303) ,** student Fourth B.Tech in **Computer Science & Engineering,** at School of Computer Science and Engineering, **RV University,** hereby declare that the project work titled **"Water Turret "** has been carried out by us and submitted in partial fulfilment for the **Semester End Examination** for the course **CS2113 AI for IoT** during the academic year **2024-2025**. Further, the matter presented in the project has not been submitted previously by anybody for the award of any degree or any diploma to any other University, to the best of our knowledge and faith.

Buvica M (1RVU23CSE123)	Signature
Gopika R (1RVU23CSE170)	Signature
Mohan Chandra SS (1RVU23CSE282)	Signature
Neeraj P Rao (1RVU23CSE303)	Signature

Place: RV University

Date: 29/05/2025

School of Computer Science and Engineering

RV University

RV Vidyaniketan,8th Mile, Mysuru Road, Bengaluru, Karnataka, India – 562112

CERTIFICATE

This is to certify that the project work titled **“Water Turret"** is performed by **Buvica M (1RVU23CSE123),Gopika R(1RVU23CSE170), Mohan Chandra SS (1RVU23CSE282), Neeraj P Rao (1RVU23CSE303),** a bonafide students of Bachelor of Technology at the School of Computer Science and Engineering, RV university, Bengaluru as part of the **Semester End Examination** for the course **CS2113 AI for IoT** offered in Semester IV of Bachelor of Technology in Computer Science & Engineering , during the Academic year **2024-2025**.

Course Faculty	Program Head(Minor and Electives)
SOCSE	SOCSE
RV University	RV University
Date:	Date:

Name of the Examiner	Signature of Examiner
1.	
2.	

ACKNOWLEDGEMENT

We extend our heartfelt gratitude to **RV University (RVU)**, the **School of Computer Science and Engineering (SOCSE)**, the **Dean**, and our **Program Head, Ms. Merin Thomas**, for their unwavering support and guidance throughout this journey.

We sincerely thank **Prof. Veena** for his invaluable knowledge, encouragement, and mentorship.

Our deepest appreciation goes to our friends for their constant motivation and camaraderie.

We are forever indebted to our parents and family for their unconditional love, sacrifices, and belief in us. Their encouragement has been our greatest strength.

This accomplishment would not have been possible without the collective support of everyone who stood by us.

Thank you all for being an integral part of our journey.

Date: 29-05-25

BUVICA M [1RVU23CSE123]

GOPIKA R [1RVU23CSE170]

MOHAN CHANDRA SS [1RVU23CSE282]

NEERAJ P RAO [1RVU23CSE303]

Place: RV university, Bengaluru

TABLE OF CONTENTS

LIST OF FIGURES

1.0 Introduction

1.1 Introduction

Fire safety plays a vital role in ensuring the protection of life and property, especially in industrial areas and high-risk environments. Traditional fire suppression systems, such as ceiling-mounted sprinklers, are designed to activate only when the temperature in an area becomes extremely high. This often means the fire has already spread before any action is taken. Additionally, these systems are usually single-use, lack precise targeting, and spray water in all directions regardless of the fire's actual location or size.

Such conventional systems are not ideal for situations that require a fast and focused response. They often lead to delayed suppression, unnecessary water usage, and expensive maintenance after activation. In factories, warehouses, and other fire-prone areas, there is a clear need for a smarter and more efficient solution.

To address this challenge, our project proposes an **Autonomous Fire Detection and Suppression System** using Internet of Things (IoT) and Artificial Intelligence (AI). This system is capable of identifying fire in real time using a camera and machine learning, and then aiming a water jet directly at the fire using servo motors and a submersible pump.

The key components used in this system include:

- Webcam – Captures live video for real-time analysis.
- Arduino Uno – Acts as the control unit, receiving angle and pump signals.
- Servo Motors (X and Y axes) – Provide pan-tilt motion to aim the nozzle at the fire.
- Relay Module – Acts as a switch to turn the pump ON or OFF.
- Submersible Water Pump – Used to spray water when a fire is detected and targeted.
- YOLOv5s Deep Learning Model – Performs fire detection with high speed and accuracy.
- Python Code – Processes the camera feed, detects fire, calculates target position, and communicates with Arduino.

1.2 Problem Statement

In many industrial and high-risk environments, fire accidents can lead to severe damage, loss of resources, and even pose a threat to human life. Traditional fire suppression systems such as ceiling-mounted sprinklers or smoke detectors have several limitations. These systems typically activate only after detecting a high level of heat or smoke, by which time the fire may have already grown significantly.

Moreover, conventional sprinklers lack directional control, they spray water across a large area regardless of the fire’s exact location. This not only delays the suppression process but also causes unnecessary water damage and leads to costly maintenance. Once activated, these systems often cannot be reused immediately and require manual resetting.

In critical areas like factories, warehouses, and storage facilities where fire risks are high, there is a growing need for an intelligent and reusable fire-fighting system that can:

- Detect fire as soon as it appears.
- Accurately locate the fire in the environment.
- Automatically aim and spray water only at the fire source.
- Minimize water wastage and prevent unnecessary damage.
- Operate without requiring human intervention during the emergency.

This project addresses these challenges by designing an autonomous water turret system that uses real-time fire detection powered by machine learning, integrated with servo-controlled targeting and IoT-based automation for fast and focused fire suppression.

2.0 Implementation

2.1 System Overview

The Autonomous Water Turret system is a smart IoT-based fire detection and suppression solution that combines real-time video analysis, machine learning, and servo motor control to locate and extinguish fires quickly and accurately.

At the core of the system is a YOLOv5s -based deep learning model that processes video input from a webcam and detects the presence of fire in each frame. When fire is detected, the system calculates its exact position relative to the center of the video frame. Based on this information, it automatically adjusts the angles of two servo motors (pan and tilt) to aim a water nozzle directly at the fire.

If the detected fire is centered within a defined threshold, a relay module activates a submersible water pump, spraying water precisely on the fire source. This targeted response reduces water usage, prevents damage to surrounding equipment, and ensures faster fire suppression. If no fire is detected, the system keeps the servos centered and the pump turned off.

The components are coordinated through a Python script, which performs real-time fire detection and calculates target offsets. The angle and pump activation commands are sent from the Python code to an Arduino Uno via serial communication. The Arduino then adjusts the servos and controls the relay module accordingly.

This integrated system is capable of:

- Monitoring fire-prone areas continuously using a webcam.
- Detecting fire using a trained YOLOv5s model.
- Automatically tracking and targeting the fire using pan-tilt servos.
- Activating a water pump when the fire is aligned and within range.
- Resetting to a neutral position when no fire is present.

2.2 Components List

The following hardware components were used to build the autonomous water turret system:

S.No	Component	Quantity	Description
1	Arduino Uno	1	Acts as the control unit for receiving angle and pump commands via serial.
2	5V Submersible Water Pump	1	Used to spray water onto the fire source.
3	5V 1-Channel Relay Module	1	Acts as an electronic switch to control the pump.
4	5V Power Adapter	1	External power supply for the pump and relay.
5	USB Cable	1	For programming and powering the Arduino from the laptop.
6	Servo Motors (SG90) with Mounts	2	For pan and tilt motion to aim the nozzle accurately at the fire.
7	Breadboard	1	For easy and temporary circuit connections.
8	Jumper Wires	Multiple	To connect components on the breadboard and Arduino.

9	Flexible Water Pipe	1	Connects to the pump for directing water to the nozzle.
10	Webcam	1	Captures real-time video frames for fire detection.
11	Laptop/PC	1	Runs the Python script and ML model (YOLOv5) for real-time processing.

2.3 Circuit Diagram

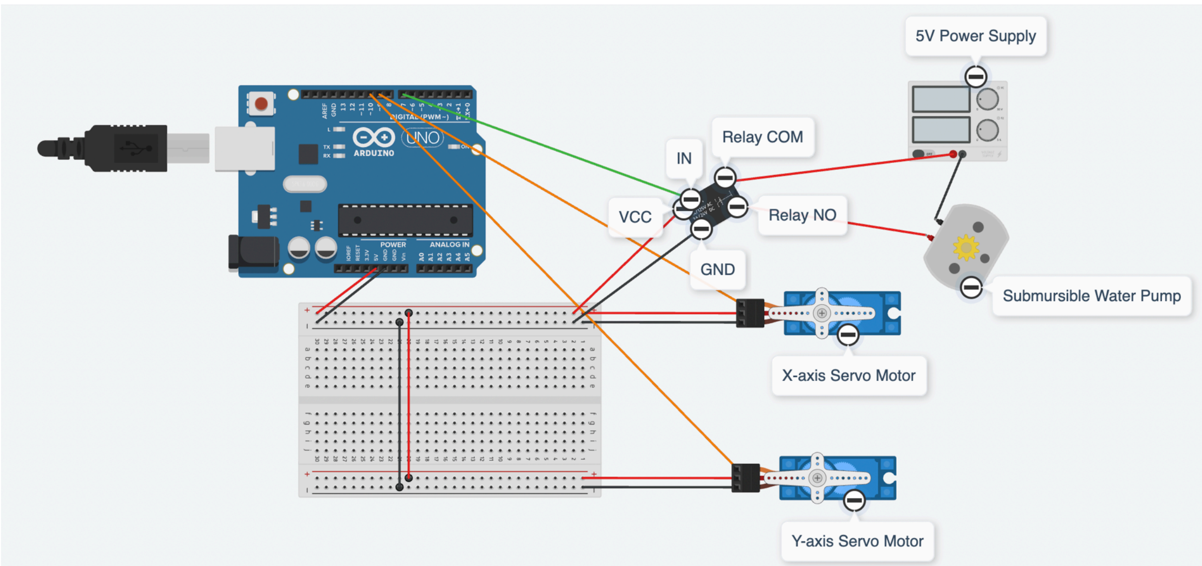

Fig 2.3: Circuit Diagram of the Autonomous Fire-Fighting Turret System

The hardware circuit of the Autonomous Water Turret system is built around an Arduino Uno, which acts as the central controller. Two servo motors are used to control the horizontal (X-axis) and vertical (Y-axis) orientation of the nozzle for precise targeting. These servo motors are connected to digital pins 9 and 10 of the Arduino for PWM control, with their power lines connected to the 5V and GND rails of a breadboard, which in turn are powered by the Arduino itself.

A single-channel relay module is interfaced with the Arduino to control the water pump. The IN pin of the relay is connected to digital pin 7 on the Arduino. The VCC and GND pins of the relay are powered from the same 5V and GND lines on the breadboard. The relay module acts as a switch between the submersible pump and a dedicated 5V external adapter, ensuring that the pump receives sufficient power without drawing it from the Arduino, which could damage the board.

The COM (Common) terminal of the relay is connected to the positive output of the 5V adapter, while the Normally Open (NO) terminal is connected to the positive terminal of the submersible water pump. The negative terminal of the pump is connected directly to the ground of the 5V power supply. This configuration ensures that the pump remains off by default and is activated only when the Arduino sends a LOW signal to the relay IN pin, closing the circuit and powering the pump.

A breadboard is used to organize and distribute power and ground lines effectively among all components. The entire system is powered through the Arduino's USB connection to a computer, which also runs the Python-based machine learning code. This script detects fire using a webcam and YOLOv5s model, and sends angle adjustments and pump activation commands to the Arduino via serial communication.

This circuit ensures safe and efficient control over the movement of the turret and activation of the pump, enabling real-time and accurate fire suppression. The use of a separate power source for the pump protects the microcontroller, and the modular structure makes it easy to debug or expand the system.

2.4 Working Prototype Logic

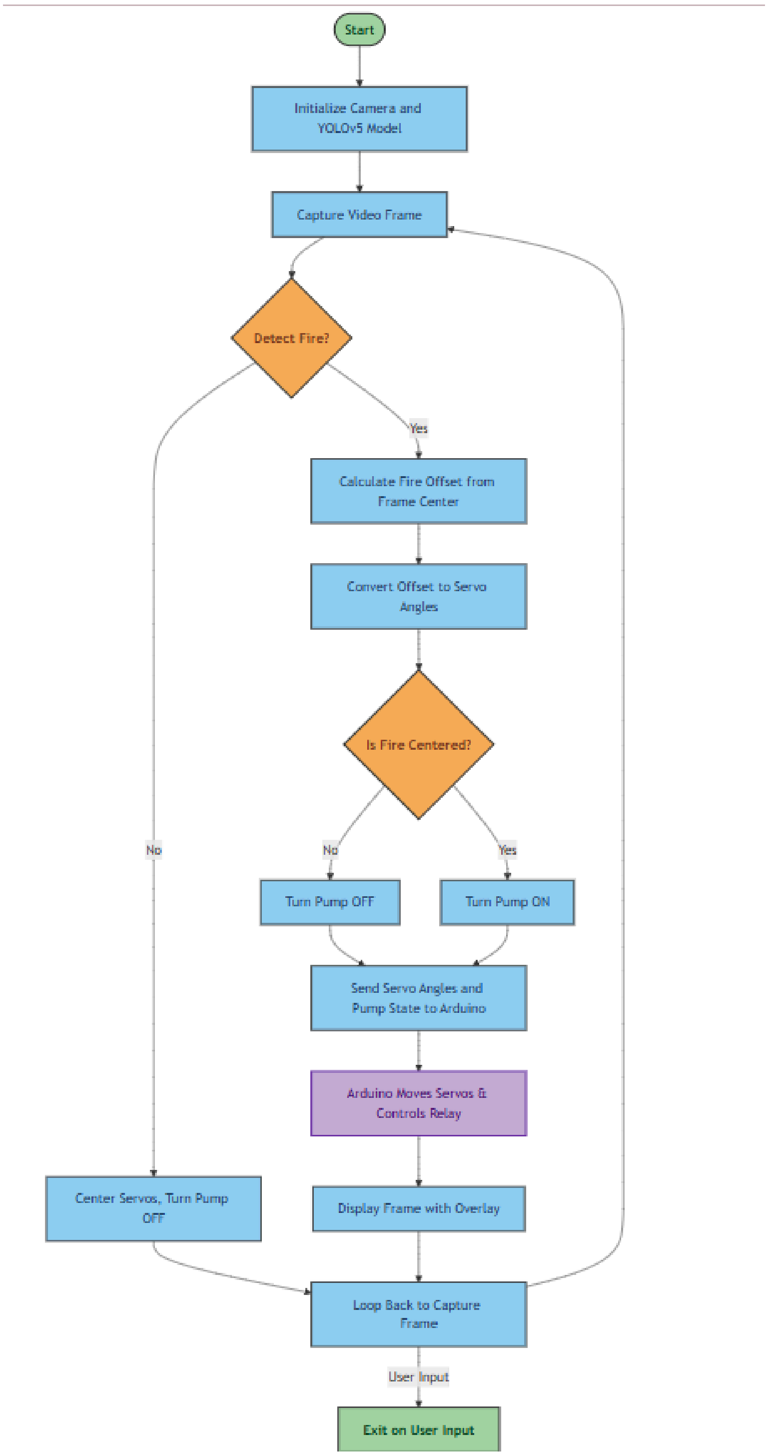

Figure 2.4: Flowchart of the Fire Detection and Tracking Algorithm

The working prototype of the Autonomous Water Turret is a real-time fire detection and suppression system that combines computer vision, hardware control, and serial communication. The core logic of the prototype is based on detecting fire through a live camera feed and responding to it autonomously by aiming a water nozzle at the fire and activating a water pump if the fire is properly aligned.

The process begins with a webcam capturing continuous video frames, which are then processed using a Python script powered by a pretrained **YOLOv5s** deep learning model. This model is trained to detect fire with high confidence in real-time. For every frame received, the model checks whether fire is present and, if so, calculates the center of the fire within the image.

Once the location of the fire is identified, the system calculates how far it is from the center of the frame. This offset in the X and Y directions determines how much the nozzle must move horizontally and vertically. These offset values are translated into corresponding servo angles using predefined sensitivity constants. The system then computes the new pan (X-axis) and tilt (Y-axis) angles required to aim the nozzle directly at the fire.

These angle values, along with a command indicating whether the pump should be turned ON or OFF, are sent to the Arduino Uno via a serial connection in a comma-separated format (e.g., 112,80,1). The Arduino receives these values, parses them, and updates the positions of two servo motors accordingly. At the same time, it controls a relay module that switches the submersible water pump ON if the fire is aligned within a defined threshold distance from the center of the frame.

The activation of the pump is conditional. The system only turns on the pump if the fire is both detected and nearly centered in the frame. This ensures that water is not wasted by spraying inaccurately. If no fire is detected or if the fire is significantly off-center, the servo motors reset to their default central position and the pump remains off.

Throughout this process, the user can view a live display of the video feed, overlaid with bounding boxes showing detected fire, current servo angles, distance from center, and the pump's status. This provides real-time feedback and makes the system highly interactive and transparent in its operation.

The integration of real-time fire detection, automated aiming using servo motors, and pump activation through a relay module showcases a fully functional and intelligent fire suppression prototype. It requires no human intervention once activated and can be deployed in fire-prone environments to improve emergency response time and reduce fire damage efficiently.

3.0 MACHINE LEARNING MODEL IMPLEMENTATION

The core of the fire detection capability in the autonomous water turret system relies on a deep learning-based object detection model. This model is responsible for identifying fire in real time from the webcam feed and providing precise bounding box coordinates, which are used to aim the water nozzle accurately. The implementation combines machine learning with embedded systems to enable intelligent and automated fire response.

3.1 Model Selection and Training

For this project, the **YOLOv5 (You Only Look Once version 5)** object detection model was selected. YOLOv5 is known for its balance between speed and accuracy, making it ideal for real-time applications. The **YOLOv5s** (small) version was specifically used due to its lightweight architecture and faster inference time, which are crucial for live video processing on consumer-grade hardware.

The model was either:

- Trained on a custom dataset of fire images, or
- A pre-trained fire detection model (YOLOv5s variant) was used with transfer learning.

The model was loaded in the Python script using **torch.hub.load()** with the **ultralytics/yolov5 repository**. It was set to evaluation mode and integrated into the live video processing pipeline for real-time inference.

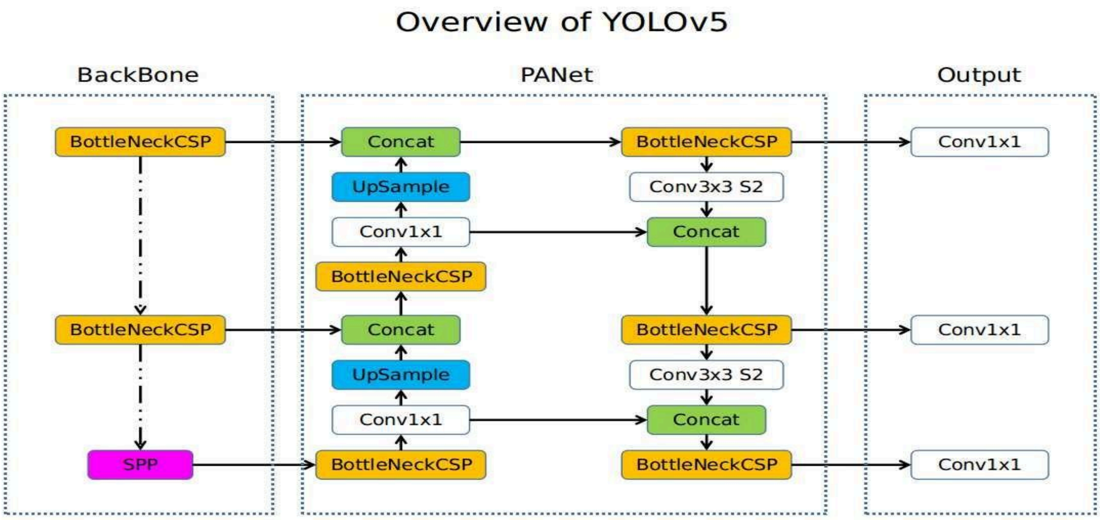

Figure 3.1: Architecture of the YOLOv5 Object Detection Model

YOLOv5s Architecture Table

Index	Layer Type	Input	Filters	Kernel / Params	Output Shape	Description
0	Focus	[3, 640, 640]	32	k=3, s=1	[32, 320, 320]	Input slicing and conv
1	Conv	32	64	k=3, s=2	[64, 160, 160]	Downsampling
2	BottleneckCSP	64	64		[64, 160, 160]	CSP block
3	Conv	64	128	k=3, s=2	[128, 80, 80]	Downsampling
4	BottleneckCSP	128	128		[128, 80, 80]	CSP block
5	Conv	128	256	k=3, s=2	[256, 40, 40]	Downsampling
6	BottleneckCSP	256	256		[256, 40, 40]	CSP block
7	Conv	256	512	k=3, s=2	[512, 20, 20]	Downsampling
8	SPP	512	512	[5, 9, 13] pools	[512, 20, 20]	Spatial pyramid pooling
9	BottleneckCSP	512	512		[512, 20, 20]	CSP block
10	ConvTranspose + Concat	[512, 256]	256	Upsample	[256, 40, 40]	Feature fusion (PANet)
11	BottleneckCSP	512	256		[256, 40, 40]	PANet bottleneck CSP
12	ConvTranspose + Concat	[256, 128]	128	Upsample	[128, 80, 80]	Feature fusion (PANet)
13	BottleneckCSP	256	128		[128, 80, 80]	PANet bottleneck CSP
14	Conv	128	128	Detection head	[3x(num_cls+5), 80, 80]	Small object detection
15	Conv	256	256	Detection head	[3x(num_cls+5), 40, 40]	Medium object detection
16	Conv	512	512	Detection head	[3x(num_cls+5), 20, 20]	Large object detection

3.2 Dataset Used

To enable accurate fire detection, a dataset consisting of annotated images of fire was used. The dataset may include:

- Images of flames in various environments (indoor/outdoor)
- Fire on screens, smoke, and other heat-related visuals
- Data augmentation (flipping, brightness changes) for robustness

Each image in the dataset was annotated using bounding boxes to label the regions containing fire. These labels were saved in YOLO format (.txt) and used during training. Training was performed on a platform such as Google Colab using GPU acceleration for faster model convergence.

3.3 Tracking Algorithm Description

The tracking algorithm works by continuously analyzing the position of detected fire in each video frame and adjusting the servo motors to center the fire. The logic is implemented in the Python script and consists of the following steps:

1. **Capture Frame**: Webcam captures a live frame.
2. **Run YOLOv5 Detection**: The frame is processed by the model to detect fire.
3. **Get Bounding Box Center**: The center (x, y) of the detected fire's bounding box is calculated.
4. **Calculate Offset**: Offset is measured as the difference between fire center and frame center.
5. **Adjust Servo Angles**:
    - The X-axis servo angle is adjusted based on horizontal offset.
    - The Y-axis servo angle is adjusted based on vertical offset.
6. **Check Centering Threshold**: If the Average distance from the center is below a defined threshold, the fire is considered centered.
7. **Activate Pump**: If centered, the system turns the pump ON via relay.
8. **Serial Communication**: Servo angles and pump state are sent to the Arduino.

This tracking loop runs continuously, allowing dynamic alignment as the fire moves.

3.4 Performance Evaluation and Metrics

The performance of the fire detection model and the tracking system was evaluated based on the following metrics:

- **Detection Confidence**: The YOLOv5 model reliably detected fire with confidence scores above 0.7 in most cases.
- **Frame Rate**: The system achieved real-time processing with an approximate frame rate of 15–20 FPS on a standard laptop.
- **Servo Accuracy**: Servo motors responded accurately to offset calculations, typically aligning the nozzle to the fire in under 2 seconds.
- **Pump Trigger Precision**: The pump was only activated when the fire was centered, preventing false positives and unnecessary water usage.
- **Responsiveness**: The full loop from detection to nozzle alignment and pump activation executed smoothly with minimal latency.

Overall, the integration of the YOLOv5 model with the servo control algorithm produced consistent and accurate results, making the system effective for real-time fire detection and suppression.

4.0 Result and Discussion

4.1 Prototype Output and Behavior

The autonomous fire detection and suppression system was successfully implemented and tested. The prototype was able to detect fire from a live video feed using the YOLOv5 model with a high level of confidence. When a fire was detected, the bounding box and confidence score were displayed on the screen, and the center of the detected fire was calculated.

The system responded by adjusting the servo motors to aim the water nozzle toward the fire source. If the fire was centered within a pre-defined threshold distance, the pump was automatically activated via a relay, and water was sprayed at the target.

In the prototype demonstration:

- When a fire image was shown on a mobile screen, the camera detected the fire with a confidence score (e.g., 0.75 as shown in Figure 2).
- The servo angles adjusted **according to the fire's position (Servo X: 91, Servo Y: 93).**
- The system displayed the offset from the center, and if the offset was within the threshold, the pump state changed to **ON**.

- If the fire was far from the center, the system showed "PUMP: OFF (Centering)" and the servos tried to align to the fire.

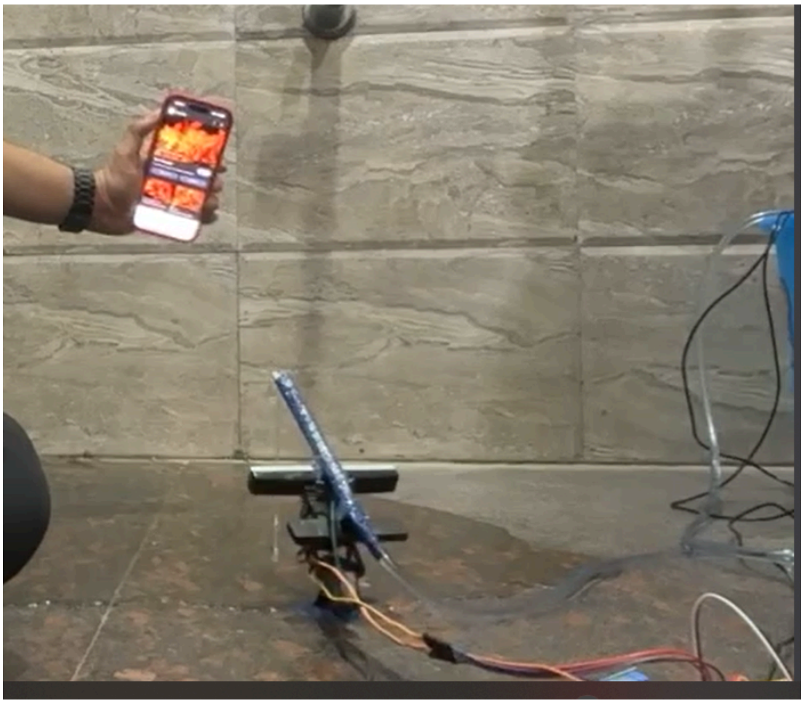

Figure 4.1: Fire Detection and Servo Targeting in Action(1)

Figure 4.1: Real-Time Fire Detection Output with Tracking Metrics(2)

4.2 Analysis of Tracking Accuracy

The fire tracking accuracy was observed to be highly effective in controlled indoor conditions. The YOLOv5 model was able to detect fire images on a smartphone screen and successfully identify the bounding box and central coordinates. The following behaviors were analyzed:

- **Detection Accuracy**: The model consistently detected fire with confidence scores above the threshold of 0.2.
- **Servo Alignment**: The pan-tilt mechanism accurately followed the fire's movement in both X and Y directions, recentering the nozzle within 1–2 seconds.
- **Pump Trigger Precision**: The water pump only activated when the fire was sufficiently centered, which prevented unnecessary water usage.
- **Real-Time Performance**: The system ran smoothly at approximately 15–20 frames per second, providing real-time feedback and adjustments.

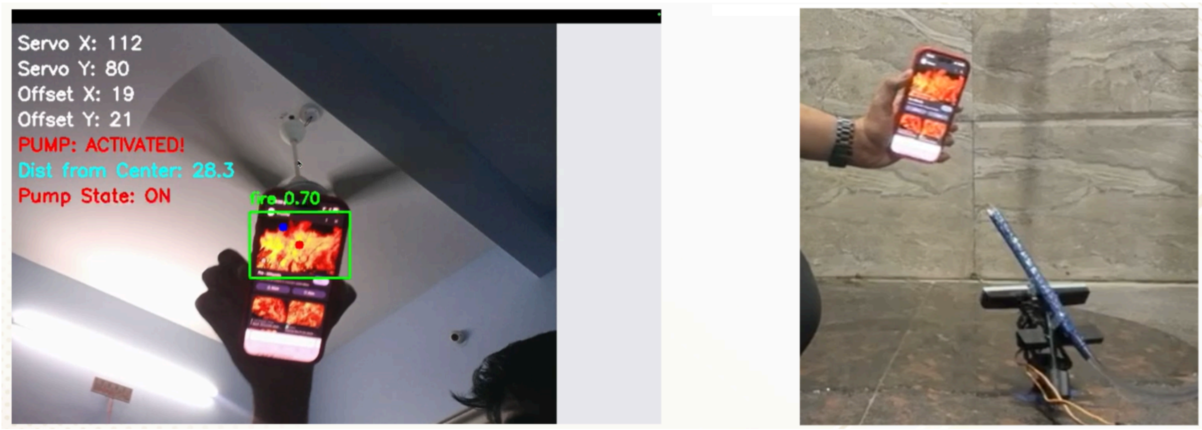

Figure 4.2: Real-Time Fire Detection and Pump Activation Demonstration

However, slight delays could occur when the fire was at the edges of the frame or when the bounding box was too small.

4.3 Challenges Faced During Development

Several technical challenges were encountered during the design and implementation of the system:

- **Serial Communication Delay**: Timely synchronization between the Python script and Arduino was critical. Minor delays in serial communication sometimes caused lag in servo motion.

- **Servo Motor Instability**: Rapid adjustments or loose mounting occasionally caused servo jitter or misalignment.

- **YOLOv5 Model Optimization**: Initially, the fire detection model detected false positives (e.g., bright lights or reflections). This was resolved by training or selecting a better fine-tuned model.

- **Power Supply Issues**: The water pump could not be powered directly by the Arduino, requiring the addition of an external 5V adapter and a relay circuit to avoid overloading.

- **Calibration**: Tuning the servo angle sensitivity and pump activation threshold required trial and error for precise targeting.

Despite these challenges, the final prototype achieved the intended functionality with acceptable reliability and accuracy.

5.0 Conclusion

In this project, an IoT-based fire detection and suppression system was successfully designed and implemented. The prototype utilized a YOLOv5s deep learning model for real-time fire detection, servo motors for automatic targeting, and a submersible water pump controlled via a relay to extinguish detected fires.

The system was able to respond autonomously by adjusting its aim and activating the pump only when the fire was centered within a threshold. This intelligent control mechanism made the system efficient, responsive, and capable of minimizing water usage.

The project demonstrated how machine learning and embedded systems can be integrated to develop smart safety solutions for fire-prone environments. The prototype is reusable, accurate, and reduces the need for manual intervention during emergencies.

6.0 Future Scope

The current version of the system is a working prototype tested under controlled indoor conditions. In the future, several improvements and extensions can be made:

- **Flame Size Estimation**: Use bounding box dimensions to estimate fire size and adjust water pressure accordingly.

- **Remote Monitoring**: Integrate with IoT dashboards or mobile apps for remote fire alerts and control.

- **Multi-Axis Coverage**: Add vertical sliders or rotational bases to expand the coverage area in larger environments.

- **Smoke and Gas Detection**: Incorporate additional sensors for early fire indicators.

- **Battery Backup**: Add a rechargeable battery to ensure operation during power failures.

These enhancements can help transform the prototype into a fully deployable smart fire safety system for real-world applications.