Intelligent Fire Detection and Suppression System: A Machine Learning-Enabled Pan-Tilt Approach

F. Caragay, B. Conlon, D. Charleston, A. Munch and A. Ristevski
EGR 2800 Final Project (Group 5) – Winter 2024
School of Engineering and Computer Science, Oakland University
Instructors: Attila Bulmus, Prof. Marco Gerini-Romagnoli

Introduction

Our team is developing an innovative pan-tilt system equipped with a comprehensive array of imaging sensors, and a fluid system designed to identify and extinguish low-risk Class-A fires. In many cases, a small fire that is extinguished with a modern sprinkler system causes a large amount of collateral damage. This issue can be mitigated by employing a system that can target and directly extinguish a fire, rather than the in-discriminant approach used by sprinkler systems. In contrast to sprinkler systems, our system will use an original machine learning model trained with 3000 images of fire and smoke to detect fires.

Mechanical Design

Our design utilizes a pan-tilt mechanism fabricated from PETG with a 3D printer. It uses NEMA 17 stepper motors to drive the pan and tilt axes, each featuring a gearbox with a 3:1 ratio. The system is outfitted with a thermal imaging camera, a LiDAR Time-of-Flight (ToF) sensor for distance measurement, and a USB webcam. The gearboxes for both axes are depicted in Figure 1 for the pan axis and in Figure 2 for the tilt axis.

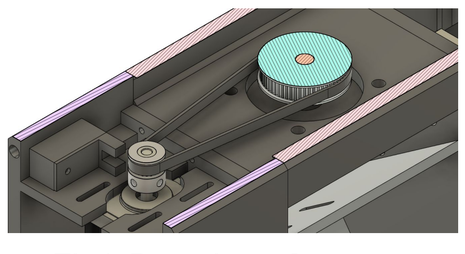

Fig 1: Pan axis gearbox

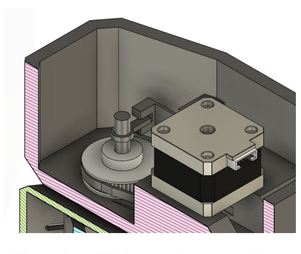

Fig 2: Tilt axis gearbox

The fluid system consists of a 24v fluid pump, and a 12v solenoid valve. The fluid pump draws water from a buildings plumbing system, or from a water source like a bucket. The system is depicted in (Fig 3) below:

Electrical Design / Software System

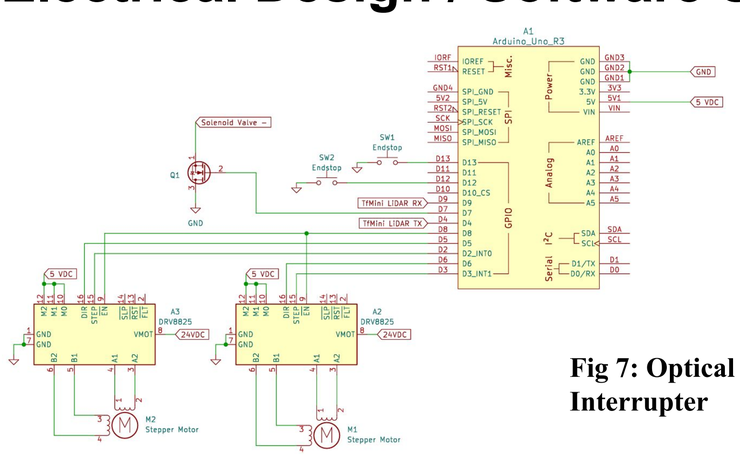

Figure 3: Electrical Schematic & Components



Fig 5: BAE SCC 500



Fig 4: Solenoid Valve



Fig 6: TFMini-S LiDAR



Fig 7: Optical Interrupter

The electrical system of this project is segmented into various subsystems as depicted in Figure 3. Stepper motors are integrated into the design and managed through GPIO pins and DRV8825 drivers. The fluid systems solenoid valve (Fig 4) is controlled by a N-channel mosfet and a GPIO. The TFMini-S LiDAR sensor (Fig 6) is interfaced via software serial communication on pins 9 and 4, allowing distance measurements for spatial analysis. Additionally, two optical interrupters (Fig 7) are used as endstop / limit switches to position the system to an initial position. The system consists of 24v DC power for the motors and driver, and 5v power for logic.

The software system consists of two subsystems, one for the manipulation of hardware components, and one for processing images and generating stepper commands. The arduino, used for the manipulation of hardware components communicates with the image processing and command generation script via a serial interface. The python script runs a YOLOv8 based model that provides positional information of the detected fire. This information is then fed through filtering and command generation functions. The generated commands to control the system are sent via serial communication to the arduino.

Results

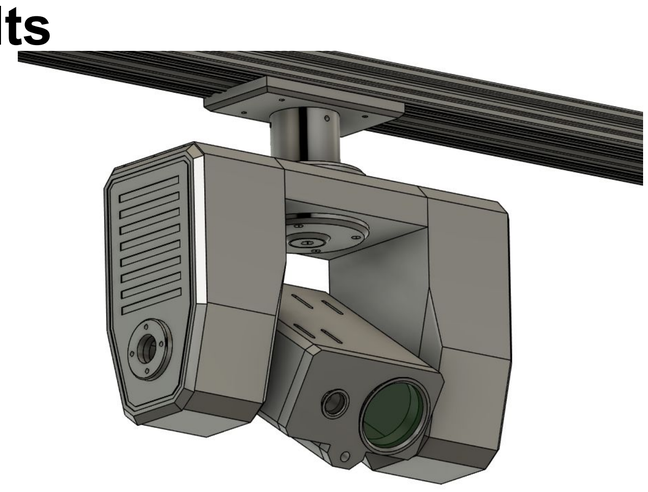

Fig 8: The Assembled Fire Suppression Machine.

The integration of mechanical, electrical, and software components results in the creation of an intelligent fire detection and suppression system (Fig 8). The pan-tilt system equipped with advanced imaging sensors and a fluid delivery system. Through the integration of a custom machine learning model trained on 3000 images of fire and smoke, the system exhibited robust fire detection capabilities (Fig 9). The mechanical design driven by NEMA-17 stepper motors with 3:1 gearboxes, ensured precise targeting of fire incidents through confidence levels calculated in the software. With microstepping enabled the NEMA-17 stepper motors move .2 degrees per step.

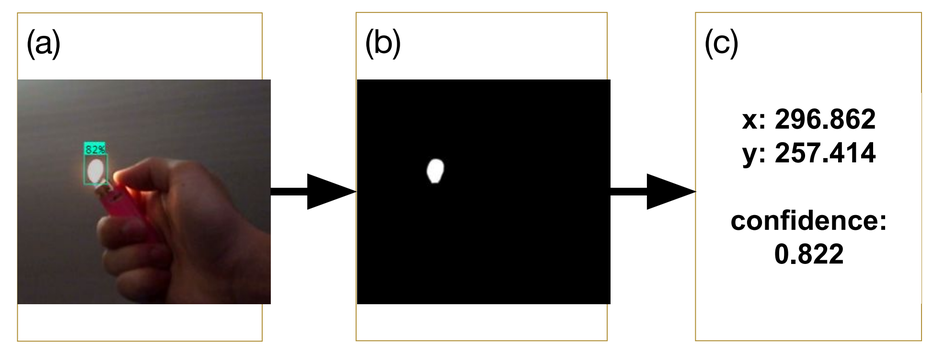

Figure 9: Annotated Image (a), Binarized Image (b), Position Data (c).

The machine learning model is based on the YOLO v8 model. It was trained on a dataset of 3000 publicly available images. The model runs via roboflow inference on python and provides the position of detected flames relative to the upper left corner of the frame (0,0). As illustrated in Fig 8 above, the system captures an image, runs it through the model, and provides two outputs: A binarized image, and a JSON output containing position data, and confidence.

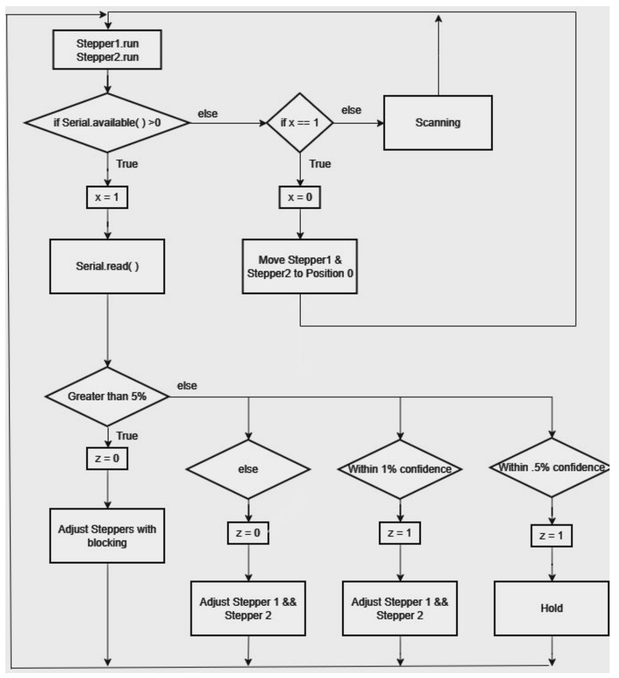

Fig 10: Software Flow Chart

This Arduino code utilizes the AccelStepper library to control the movement of two stepper motors. It establishes serial communication and initializes the motors positions in the setup phase. The main loop continuously monitors serial input for motor control commands from the python script. Based on the received commands, it orchestrates the movement of the motors, ensuring precise steps within .2 degrees. Additionally, the loop handles conditions for resetting the motors to their initial positions when necessary. This code provides a flexible framework for controlling stepper motors, suitable for various applications requiring accurate motor movements in real time. (Fig 10).

Conclusion

Our project attempts to solve the problem of collateral damage caused by modern fire suppression systems such as sprinklers. Sprinkler systems cause excessive water damage in scenarios where the fire spread and damage is minimal. We were able to achieve this through a combination of of electromechanical elements, and powerful software based technology. Throughout the project we faced many challenges such as integrating a custom trained model into a electromechanical system, further challenges included: implementing proportional controls, implementing water trajectory corrections and calculations. However we were able to overcome these challenges and ensure that the system functions as intended. We were able to address our problem by combining the expertise of multiple team members, and designing a system to address the problem directly.