UNIVERSITY of SOUTH FLORIDA
College of Engineering
Department of Electrical Engineering

EEL4914 Spring 2023

Video Analytics at the Edge

FINAL REPORT

- Luke Barber, Mikel Holmquist, Shawn Sheng
- Advisor: Dr. Yasin Yilmaz

Content

2. EXCUTIVE SUMMARY

2.1 Purpose

In this design effort we seek to showcase the effectiveness of video analytics using edge computing techniques. Video analytics refers to the automated analysis of visual data to detect specific objects, events, or activities in a video stream. Edge computing is a distributed computing paradigm that brings computation and data storage closer to the location where it is needed, which is typically at the edge of the network. In this case, "The edge" refers to performing effective video analytic methods, such as machine-learning-based object detection, in the proximity of the optical instrument, rather than relying on a centralized processing system.

The primary objective of this project was to develop a prototype that demonstrates the benefits of video analytics with edge computing techniques. The prototype operates autonomously, providing insights into anomalous objects or motions of interest. The traditional approaches require transporting raw or compressed data of the captured video signal to a central command center, which requires significant bandwidth. **Instead, our prototype aims to reduce the bandwidth requirements between the sensor and the central computer.**

Our product delivers parameters of interest to the command center, where the decision process takes place. In future updates, a user interface will be developed to allow users to customize the system so that the desired anomalous threshold will activate expected visual and audio data outputs.

The methodology used in this design effort involved developing a prototype that optimized the processing power available from off-the-shelf components. Our product is designed to operate in real-world conditions and showcase the potential benefits of video analytics with edge computing techniques. Our prototype has been tested and evaluated using various performance metrics, including accuracy, latency, and resource utilization.

In conclusion, we aim to demonstrate the potential benefits of video analytics with edge computing techniques by developing a prototype that showcases the effectiveness of said techniques. The prototype operates autonomously, providing insights into anomalous objects or motions of interest, while reducing the bandwidth requirements between the sensor and the central computer. The findings and recommendations from this project will contribute to the development of a more robust design and, hopefully, to the future of edge computing techniques.

2.2 List of Figures

2.3 List of Tables

2.4 Abbreviations, Acronyms and Definitions

AI	Artificial Intelligence
C-SWAP	Cost, Size, Weight, and Power
CSS	Cascading Style Sheets
DC	Direct Current
EE	Electrical Engineering
GPIO	General Purpose Input and Output
HDMI	High-Definition Media Input
IP	Internet Protocol
JXNX	Jetson Xavier NX
KNN	k Nearest Neighbors
ME	Mechanical Engineering
MS	Motion Sensor
SOP	Standard Operational Procedures
UDP	User Datagram Protocol
NN	Neural Network
USB	Universal Serial Bus
UPS	Uninterruptible Power Supply
UI	User Interface
VBP	Virtual Buttons Panel
SVM	Support Vector Machine

Table 1. List of Acronyms

3. INTRODUCTION

3.1 Background Information

3.1.1 Video Analytics

Video analytics is a branch in computational technology that centers on analyzing numerous visual frames over a predetermined duration. It is through the utilization of video analytics techniques and algorithms that infrastructure security, machine vision, and the multimedia industry achieve their foundational framework.

3.1.2 Edge Computing

Conventional sensor-processor network structures, particularly those used in video surveillance technologies, usually have a distinct separation between the computational power and sensor networks. A traditional network often consists of a sensor that sends remotely captured raw or processed data to a computer through high-speed data mediums. In this paradigm, trouble often arises when unforeseen burst of inrush data creates a critical congestion to the network, impeding system stability. Edge computing, on the other hand, is a design approach that involves bringing computational resources closer to the source of data. In edge computing, transport related congestive issues are often mitigated, even during periods of intense activities.

3.2 Motivation

The proliferation of connected devices in recent years has significantly increased the demand for processing power. To optimize autonomous systems and reduce the processing load on central servers, edge computing has emerged as a viable solution.

The motivation behind developing a system with reduced bandwidth requirements, leveraging machine learning techniques, for anomalous objects and action detection is to address the limitations of traditional cloud-based systems, which have scaling constraints. These constraints make it difficult for large and distributed surveillance systems to process and analyze data in real time, resulting in delayed responses and potential security risks.

By incorporating machine learning capabilities, our system can analyze and pre-process the data locally to generate feature data. This low-signature data can be then transmitted to a central system for processing to detect anomalous objects and movements in real-time. We believe this approach can significantly improve the reliability and effectiveness of large-scale surveillance systems, as potential threats can be identified and addressed in real time, reducing the risk of security breaches.

In other words, edge computing can optimize autonomous systems by reducing the processing load on central servers, which will lead to faster processing times and lower infrastructure costs for businesses that rely on surveillance systems.

Traditionally surveillance systems use physical weather proofing to increase survivability in harsh environmental conditions. In this design effort, it is perhaps profitable to take advantage of the increased computational capability and sensor awareness to minimize the need to expose sensitive components, such as advanced imaging sensors, during times with little to no activity.

3.3 Scope

This project aims to develop a surveillance imaging system that preprocesses the data using data analytic and machine learning techniques so that reduced bandwidth is achieved between server and sensor communications. Deep neural network-based machine learning models will be employed to preprocess the images captured by the optical imager as well as originally trained models, such has K-Nearest Neighbors and Support Vector Machines, will be used to incorporate optical flow detection to detect anomalies in real-time. In addition, the system will utilize advanced metrics to incorporate a sensor deployment mechanism to only expose sensitive components during times that features activities of interest. The system will undergo extensive testing and prototyping to ensure it meets all target requirements and specifications, with the goal of delivering a reliable, robust, and efficient imaging system.

4. METHODOLOGY

4.1 System Components

Here we specify all off-the-shelf and significant components utilized in this design effort with their relevant specifications.

4.1.1 Raspberry Pi Camera 3

The Raspberry Pi Camera 3 is the main imaging sensor for this surveillance system. Two Raspberry Pi Camera 3s are used to provide both wide field-of-view and standard field-of-view coverage for surveillance purposes. The Raspberry Pi Camera 3 features both high image quality and as well as auto-focusing capabilities.

4.1.2 Raspberry Pi Zero W

Since the drivers used to operate Raspberry Pi Camera 3 are only available on Raspberry Pi Compute units, the Raspberry Pi Zero W serves as the image pre-processor to interface with the Raspberry Pi Camera 3. The Raspberry Pi Zero W is responsible for standard field-of-view observations. Together with the Raspberry Pi Camera 3 at standard field-of-view, this combination makes up the *standard field-of-view image sensor unit.*

4.1.3 ARRISHOBBY Zhaoyun Plus Gimbal

To provide the surveillance system with pitch-tilt-pan capability, a relatively low cost and open-source gimbal system was selected to be the physical host for the *standard field-of-view image sensor unit.*

4.1.4 Raspberry Pi 4

The Raspberry Pi 4 is used to interface with the Raspberry Pi Camera 3 at wide field-of-view. This device is also used to control and interface the camera deployment mechanism.

4.1.5 Nvidia Jetson AGX Xavier

This project shall utilize a Nvidia Jetson Xavier AGX as the main edge device for the purpose of the detection of anomalies in the context of a surveillance system. The Jetson Xavier AGX is equipped with 512 NVIDIA CUDA Cores and 64 Tensor Cores for advanced machine learning model inferencing. These features allow the AGX to run multiple neural networks with respectable efficiency as while enabling high-resolution data processing from multiple sensors efficiently and quickly.

4.1.6 STEPPERONLINE NEMA 17 Stepper Motor Class 1.5A.

Three NEMA 17 class stepper motors, rated at 1.5A and 63.74oz of torque, are used to drive the camera deployment mechanism.

4.1.7 Trinamic TMC2209 Stepper Motor Driver

Three Trinamic TMC2209 stepper motor drivers are used to control the stepper motor used to drive the camera deployment mechanism. These drivers have been chosen for their StealthChop2 and advanced current limiting capabilities which dramatically reduces the noise generated by the stepper motors.

4.1.8 BIGTREETECH Octopus CNC controller

This project utilizes the BIGTREE Octopus CNC controller to control the stepper motor drivers used for the camera deployment mechanism. The BIGTREE Octopus CNC controller is a part of the STM32F4 family of microcontrollers used to provide basic algorithms to control the stepper motor drivers in serial.

4.1.9 MEANWELL LRS-350-24 Regulated Power Supply

A MEANWELL LRS-350-24 power supply is used to satisfy the power requirements needed for the system.

4.2 System Architecture and Technical Design

This section specificizes the final system design, both hardware and software, of the prototype surveillance system.

4.2.1 Physical Fixture

The physical fixture for the prototype system aims to both physically host the computing platform as well as the deployment mechanism for the external standard field-of-view image sensor unit.

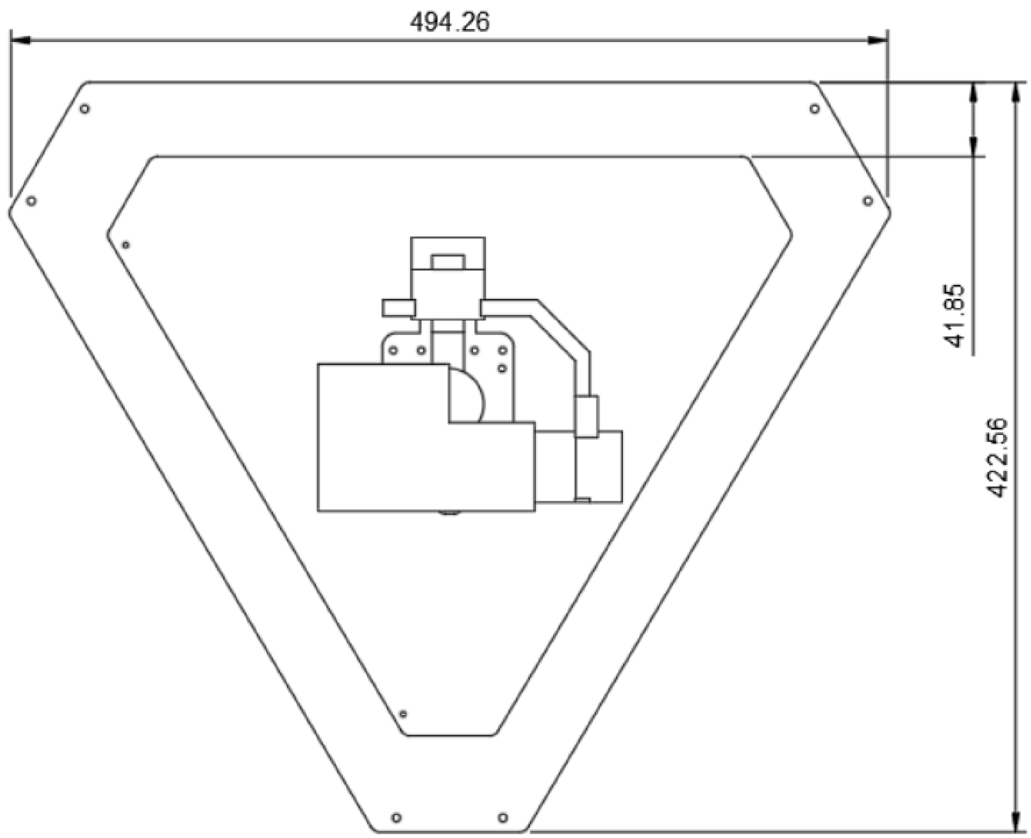

Figure 1. General design and dimensions for the physical platform.

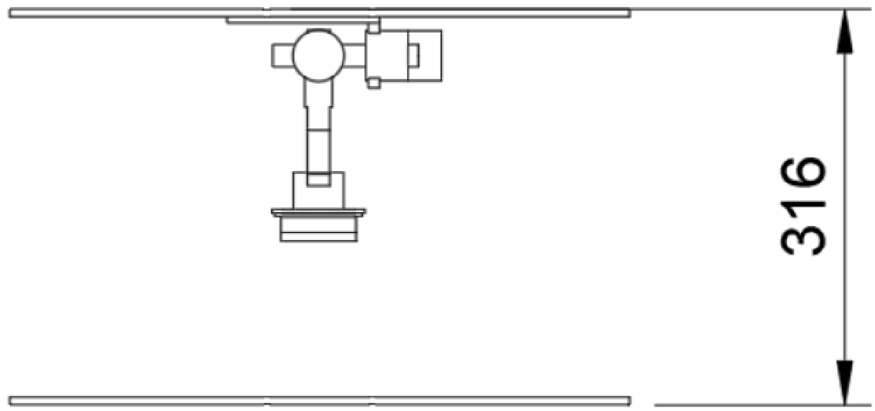

Figure 2. General design and dimensions for the physical platform (Cont.)

The physical frame is constructed by the affixation of two laser-cut ¼” thick acrylic sheets by six high-precision 8mm steel rods.

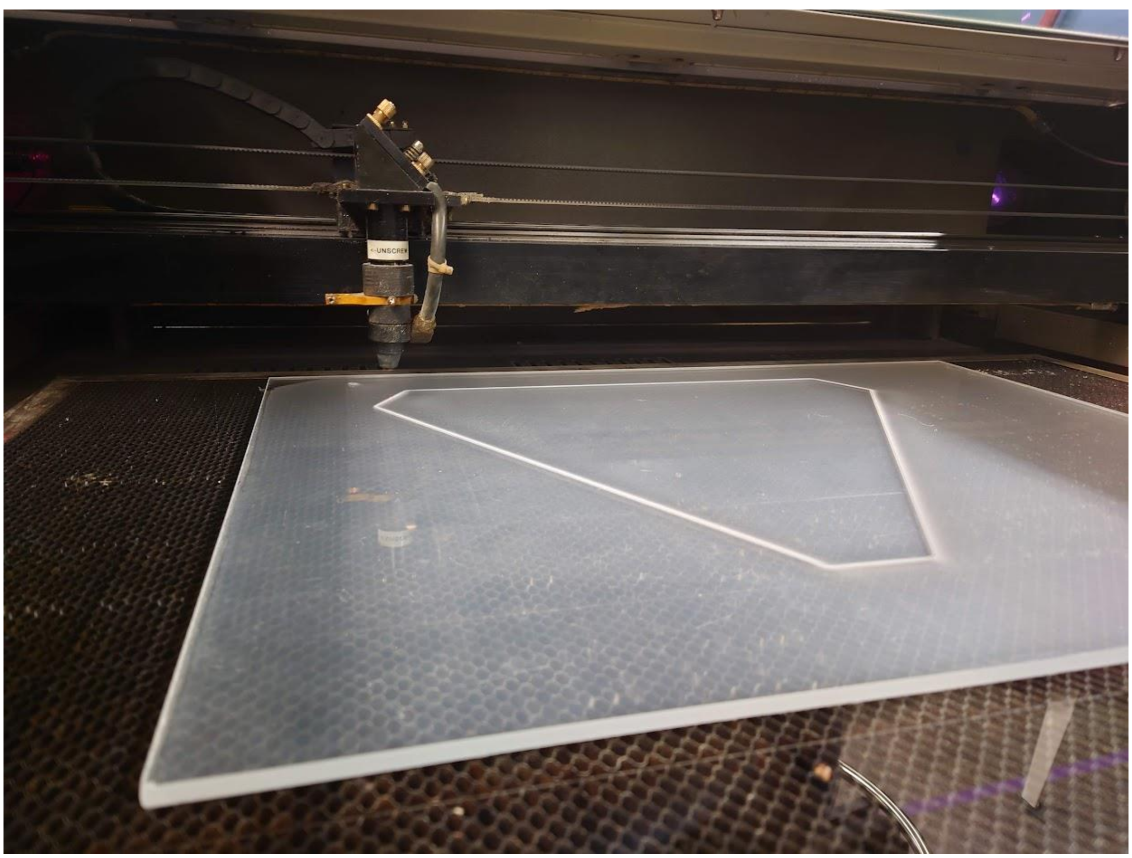

Figure 3. Cutting of the acrylic sheet using an 80 Watt Co2 laser system

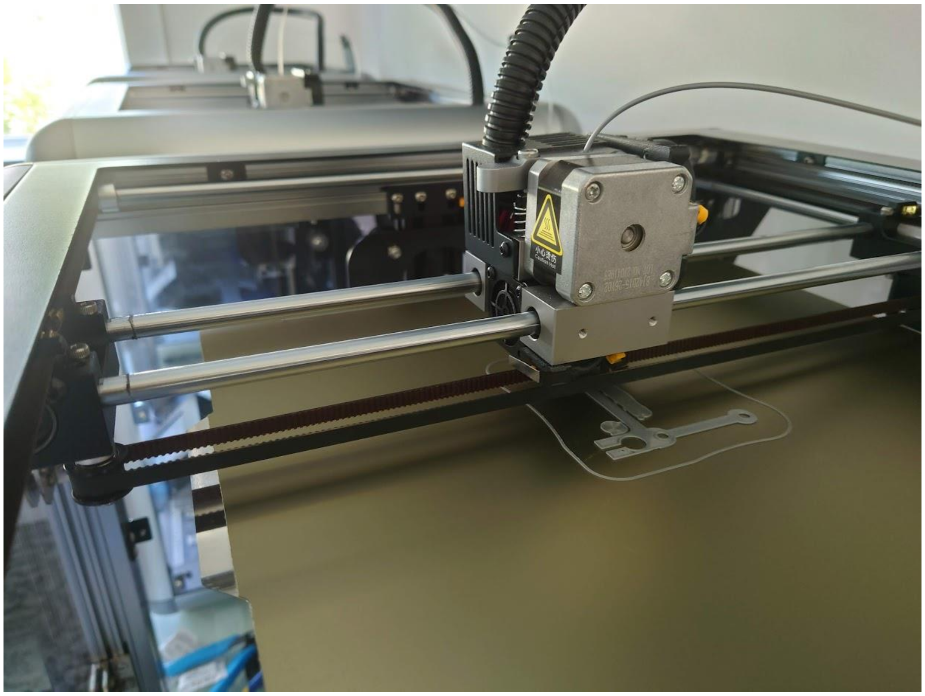

Figure 4. 3D printing of fixtures used to mount modules on the physical frame.

3D printed plastic fixtures are used to fixate various components to the overall physical frame. Although the design is a heavily inspired version of an opensource motion system, the Rostock Mini, multiple components was modified and re-designed to accommodate the large gimbal system.

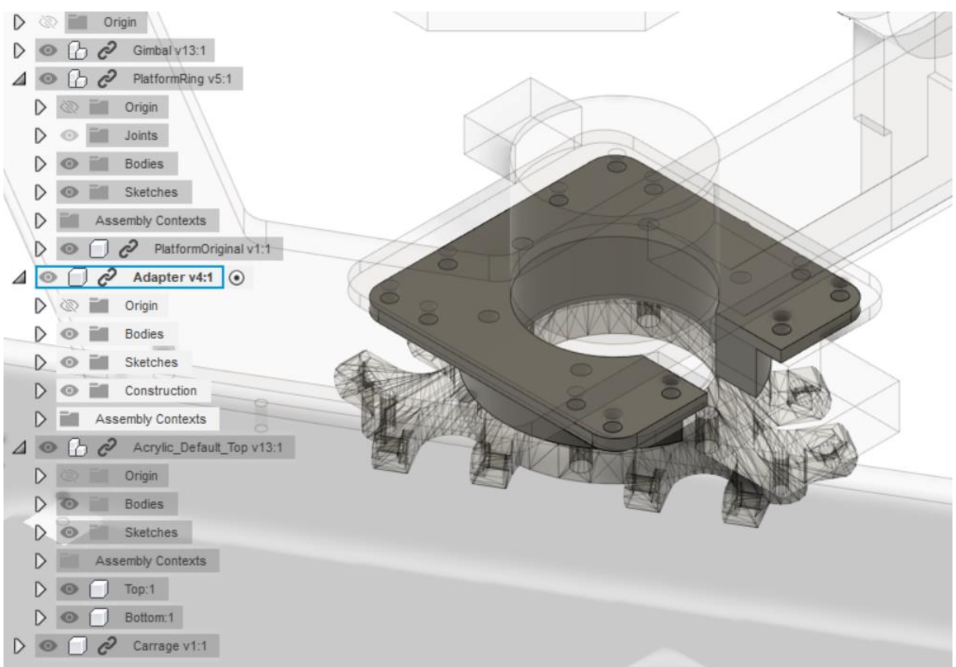

Figure 5. Example of custom designed part for adapting the motion system to the gimbal unit.

4.2.2 The Deployment System

The deployment system used to extend the standard field-of-view image sensor unit leverages a delta bot design to achieve high deployment speed. Carbon-fiber reinforced linkages are used to link the camera gimbal to the actuating units while three stepper motors independently drive the actuating units via the use of a timing belt system.

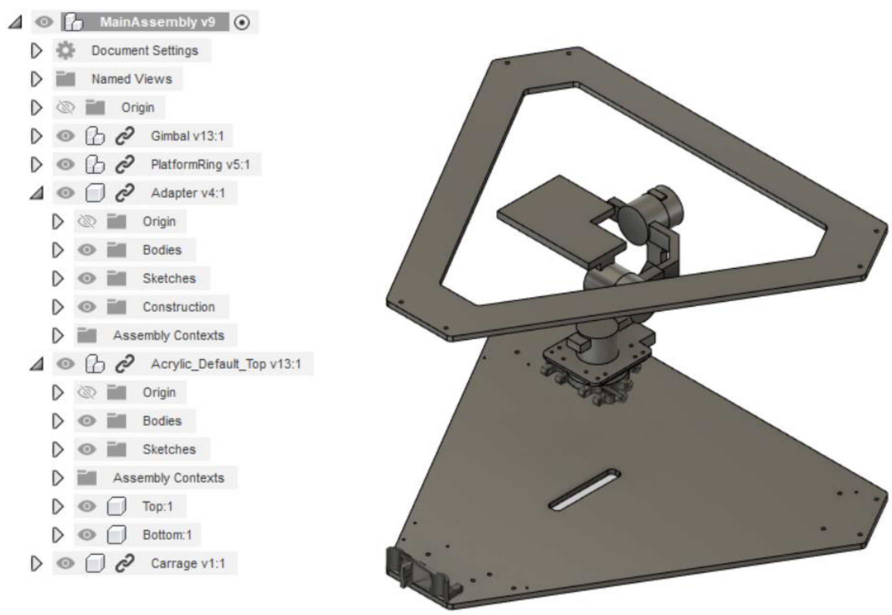

Figure 6. Overall view of the components responsible for gimbal deployment

4.2.3 Imaging System

The imaging system mainly utilizes two imaging sensors with different field-of-view to achieve

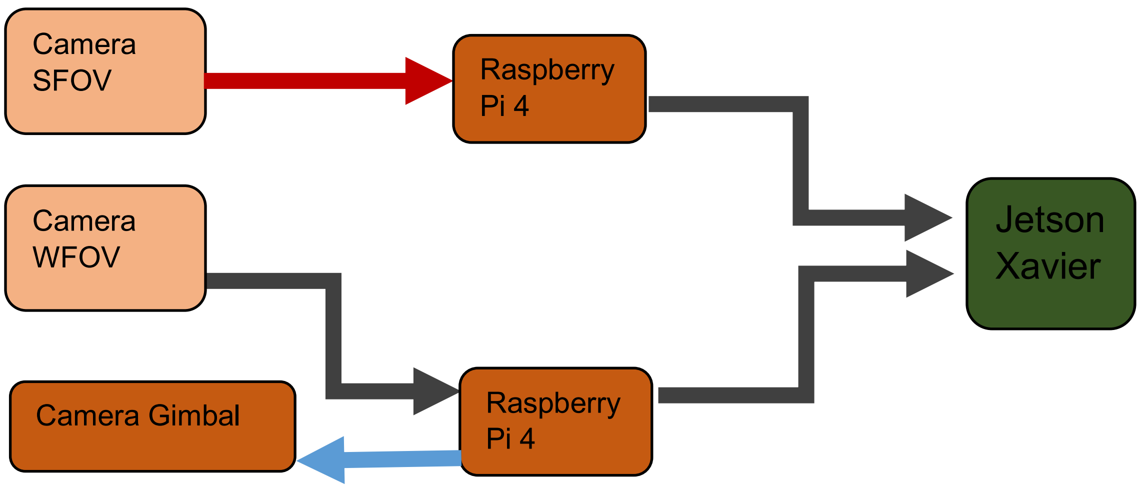

Figure 7. Flowchart of the communication paths of the imaging system.

4.2.4 Imaging Software Architecture

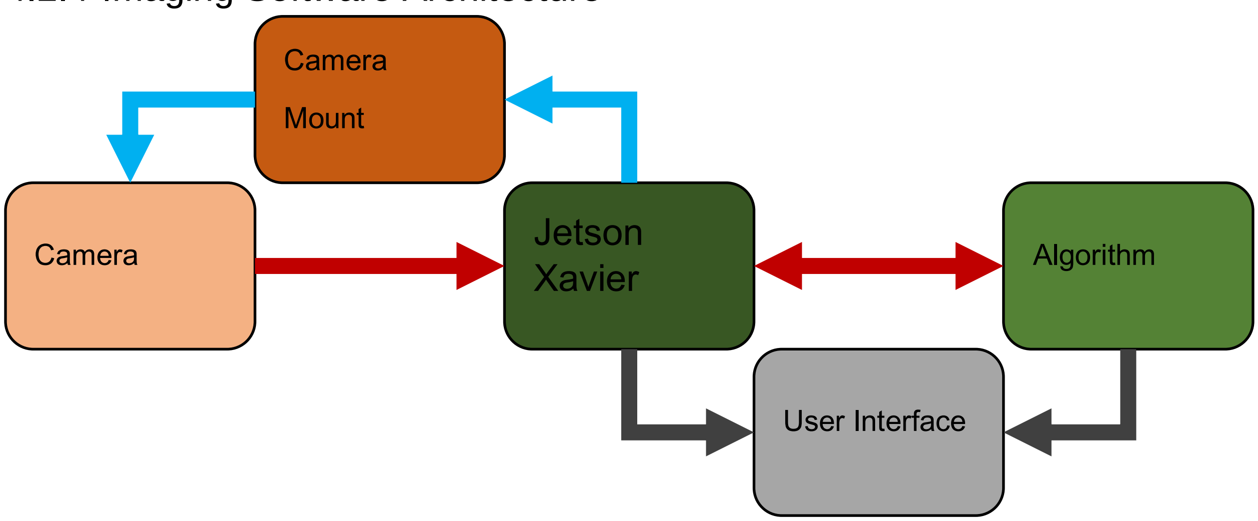

Figure 8. Software Architecture.

4.2.5 Data Transport Architecture

Data Flow Chart

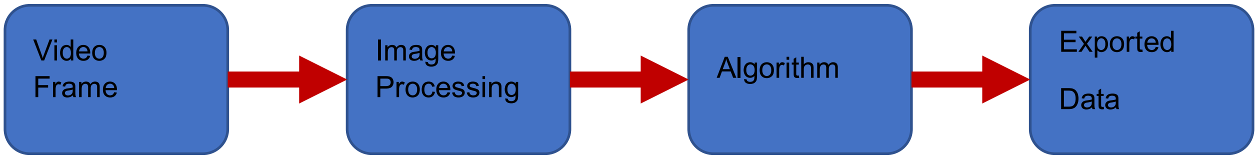

Figure 9. Data Transport Architecture.

4.2.6 Anomaly Detection

It was found that the machine learning model most conducive to the success of the project was KNN. The KNN algorithm is “a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.” (IBM, 2023).

4.2.7 Display

In order to demonstrate the system’s live performance, we will be using a display that will show data KNN, SVM. SVM is used, among other things, to recognize objects and detect anomalies. On the other hand, KNN assists us in calculating the similarity between gathered data and prepared data, which is then classified based on their similarity.

5. RESULTS

5.1 Data Bandwidth

To establish a baseline for data bandwidth usage with traditional techniques, seven commonly used data encoding techniques were selected. The following metrics are measured from the ethernet bandwidth usage transporting a 10-minute frame sequence (Special 4K HDR 120FPS Dolby Vision Demo by @DolbyVisionDemo4K acquired from YouTube). The video's audio channel is removed and, in addition, is pre-processed to be at a frame rate of 30 frames per second at resolution of 1920 by 1080 pixels (3MP).

Technique	Average Bandwidth
HEVC 264	9.47 Mbps
HEVC 265	5.37 Mbps
VP8	10.84 Mbps
VP9	8.16 Mbps
AV1	4.87 Mbps
MPEG-2	23.92 Mbps
MPEG-4	14.01 Mbps
FeatureFlow*	**1.64 Mbps**

**FeatureFlow denotes the technique developed by this effort.*

Table 2. Data Bandwidth Results.

6. DISCUSSION

6.1 Limitations

Storage Capacity: While edge devices are great for improving latency and response times, their storage is usually limited, which may require us to regularly transfer data either manually or by connecting to a central server.

Processing Power: Not having access to central servers limits the amount of data that edge devices can handle on their own, they often require powerful devices to work optimally.

Cost: Edge computing is an expensive solution since it generally requires better hardware to perform efficiently compared to more traditional techniques. Additionally, edge devices require frequent maintenance which may also increase the overall cost.

7. CONCLUSIONS

7.1 Project Outcomes

The rise of edge computing approaches has aided in the growth of automated systems that rely on real-time data processing. Standard techniques of transmitting raw or compressed data to a central command center in the field of video analytics can be extremely CPU-intensive and need substantial bandwidth. Edge computing is a model of distributed computing that moves processing and data storage closer to the point of use, minimizing the volume of data that needs to be transmitted and improving response times.

By building a prototype that functions autonomously and delivers insights into anomalous objects or even atypical motion, this design effort aimed to demonstrate the potential benefits of video analytics using edge computing approaches. This project's methodology consisted of maximizing the processing power available from off-the-shelf components with the goal of creating a prototype that is meant to work in real-world situations.

The testing findings reveal that our prototype is successful at lowering the bandwidth requirements between the sensor and the central computer while still delivering accurate insights into unusual items or motions of interest.

The capacity to handle and analyze data locally, minimizing the need for centralized data processing and storage, is one of the most significant advantages of edge computing systems. In automated systems, this can contribute to faster decision-making and lower latency. Edge computing can further reduce the total amount of data that has to be transferred by putting computation and data storage closer to the site where it is needed, decreasing network congestion, and boosting overall system efficiency.

7.2 Future Improvements

1. Development of a User Interface: A UI could be developed to enable users to customize the system such that the relevant anomalous threshold activates the expected visual and auditory feedback outputs. Thus, users could have more control over the system which would allow them to modify it to their specific needs.

2. Incorporation of multiple sensors: Including additional sensors such as thermal imaging, infrared, or sound sensors may provide further information and improve the system's accuracy.

3. Integration of Systems: To construct a more comprehensive automated system, the prototype could be combined with additional systems and technologies, such as drones or autonomous cars. This has a wide range of uses, including surveillance, security, and transportation.

8. REFERENCES

[1] Edinburgh Sensors, “Edinburgh Sensors,” 2022. [Online]. Available : https://edinburghsensors.com/news-and-events/environmental-technology-the-impact-of-technology-on-the-environment-and-environmental-technology/.

[2] R. Rautela, S. Arya and S. Vishkwarma, “E-waste management and its effects on the environment and human health,” Elsiever, 2021.

[3] J. R. Saura, D. Ribeiro-Soriano and D. Palacios-Marques, “Assessing behavioral data science privacy issues in government artificial intelligence deployment,” Elsiever, 2022.

[4] Z. Yan, W. Susilo, E. Bernito, J. Zhang and L. T. Yang, “AI-driven data security and privacy,” 2020.

[5] Micorsoft, “Introduction to PyTorch,” Microsoft, [Online]. Available : https://learn.microsoft.com/en-us/training/modules/intro-machine-learning-pytorch/.

[6] J. Brownlee, “PyTorch Tutorial: How to Develop Deep Learning Models with Python,” Machine Learning Mastering, 23 March 2020. [Online]. Available: https://machinelearningmastery.com/pytorch-tutorial-develop-deep-learning-models/.

[7] S. Chintala, “Deep Learning with PyTorch: A 60 Minute Blitz,” PyTorch, 2022. [Online]. Available: https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html.

[8] S. Raschka, Y. Liu and V. Mirjalini, “Machine Learning with PyTorch and Scikit-Learn,” packt, 19ebruary 2022. [Online]. Available: https://www.packtpub.com/product/machine-learning-with-pytorch-and-scikit-learn/9781801819312.

[9] IEEE, “IEEE Reference Guide,” Piscataway, 2018.

[10] P. Silva, C. Goncalves, N. Autunes, M. Curado and B. Walek, “Privacy risk assessment and privacy-preserving data monitoring,” Elsiever, 2022.

9. APPENDICES

9.1 Appendix A

Bill of Materials and Cost

#	Component Name	Short description	QTY.	Cost
1	Image Sensor	Raspberry Pi Camera 3 Standard FOV	1	$25.00
2	Image Sensor	Raspberry Pi Camera 3 Wide FOV	1	$35.00
3	Sensor Interface Computer	Raspberry Pi Zero W	1	$15
4	Sensor Interface Computer, Gimbal Controller	Raspberry Pi 4	1	$50.00
5	Jetson Xavier NX	NVIDIA Jetson Xavier NX Developer Kit	1	$400.00
6	128gb MicroSD	SanDisk 128GB Ultra microSD	2	$35.30
7	Gimbal Assembly		1	$499.00
8	PI cam	Raspberry Pi Camera Module V2	1	$29.95
9	HDMI Monitor	Full HD (1920 x 1080) HDMI monitor	1	$99.99
10	HDMI Cable	HDMI Cable (6 ft)	1	$6.50
11	USB Cable	USB 3.0 Male to Male Type A (1.5 ft)	1	$6.99
12	USB Keyboard	Basic USB Keyboard	1	$10.77
13	USB Mouse	Basic USB Mouse	1	$5.99
14	5V/4A Power Supply	Power Supply for Jetson Nano	1	$12.99
Subtotal				$1232.48
Est. Tax				$86.27
Total				$1318.75

Table 3. Bill of Materials.