Skeyes Structural Inspection Drone

Noah Levie, Daniel Copley, Mohammed Almershed

NORTHERN ARIZONA UNIVERSITY
College of Engineering, Informatics, and Applied Sciences

Abstract

Building inspection, especially in the case of taller or not easily-accessible buildings, is a mundane task made slow, costly, and at times dangerous by the limitations of the human form. Team Skeyes has created a system which performs an inspection of a structure and detects potentially faulty or noteworthy features with minimal oversight and input from the user.

This structural monitoring drone, equipped with a GoPro camera, uses custom-trained object recognition and image classification networks to identify regions of concern on the building. The video taken by the video camera is wirelessly transmitted to a ground control station – a laptop operated by the user. This ground control station uses the QGroundControl software to display the drone's telemetric data, determine flight paths and issue commands, and display the drone's view with structural features designated by boxes. A two-stage detection system using the You Only Look Once v4 (YOLOv4) object recognition system and custom TensorFlow convolutional neural networks (CNNs) processes the live video feed of the drone and requests input from the user according to whether the feature is deemed faulty or nominal. This system allows the user to perform a comprehensive structural inspection without receiving extensive training or endangering themselves in their field of work.

Requirements

1.1* The drone must be operable using a graphical user interface
2.1* The drone will use image processing in order to isolate and identify features relevant to the operator, which should be at least 80% accurate
2.2.1* Each YOLOv4 feature should be recognizable with at least 90% accuracy
2.3.1* The damage classifiers should be at least 90% accurate in distinguishing between faulty and intact features
2.4* Feature detection within the video feed will trigger appropriate commands to be sent via MAVLink

Design Components

- **Drone Kit:** Using the HolyBro S500 kit with the PixHawk 4 flight controller, we can accurately and reliably control the drone and send mission flight commands via MAVLink
- **Ground Control Software:** We are using QGroundControl to interface between the user and the flight controller, in order to monitor the flight and data, as well as allow the user to provide inputs
- **Visual Damage Classifier:** Using TensorFlow and OpenCV, several open-source Python libraries, we use custom-built and custom-trained classifiers to determine whether a feature is faulty or nominal
- **Object Recognition System:** The open-source, high-level object recognition system, You Only Look Once v4 (YOLOv4), can be optimized and custom-trained in the cloud to identify and crop features of images in real time.

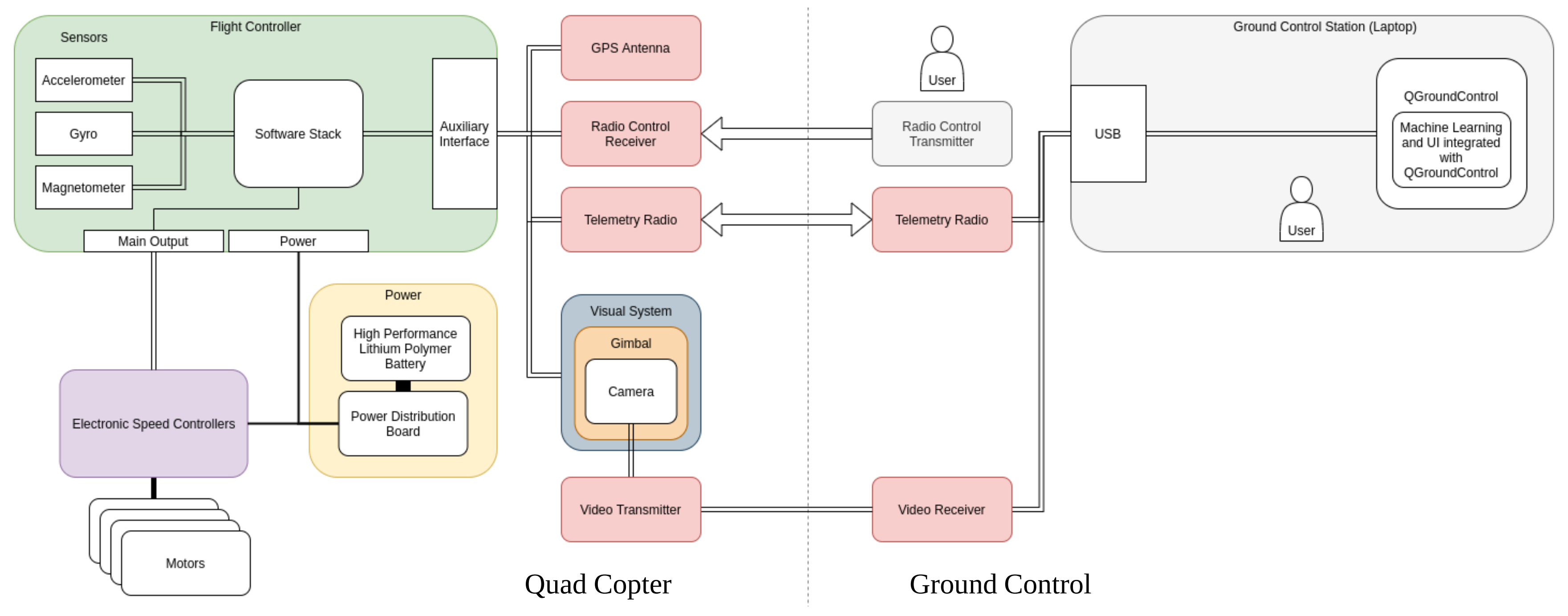

Figure 1: General System Architecture

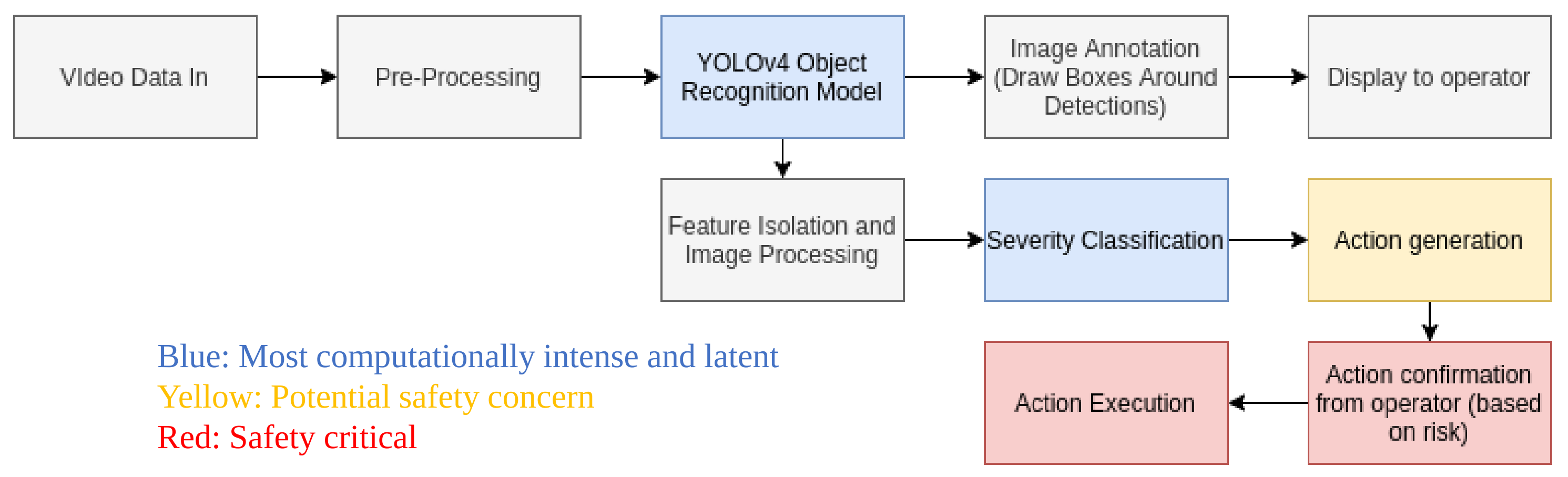
Video Data In
Pre-Processing
YOLOv4 Object Recognition Model
Image Annotation (Draw Boxes Around Detections)
Display to operator
Feature Isolation and Image Processing
Severity Classification
Action generation
Action confirmation from operator (based on risk)
Action Execution
Blue: Most computationally intense and latent
Yellow: Potential safety concern
Red: Safety critical

Figure 2: Software System Architecture

Results

- YOLOv4 Mean Average Precision (mAP): 78.85%
  - Window Detection Avg. Precision: 83.68%
  - Gutter Detection Avg. Precision: 74.02%
- CNN Avg. Precision:
  - Window Classifier: 70.73%
  - Gutter Classifier: 73.23%
- CNN Optimized Configuration:
  - Window:
    - 3 Convolutional Layers
    - 2 Dense Layers
    - 32 nodes per layer
  - Gutter:
    - 3 Convolutional Layers
    - 0 Dense Layers
    - 128 nodes per layer
- Display delay: 1.0 sec – 1.4 sec

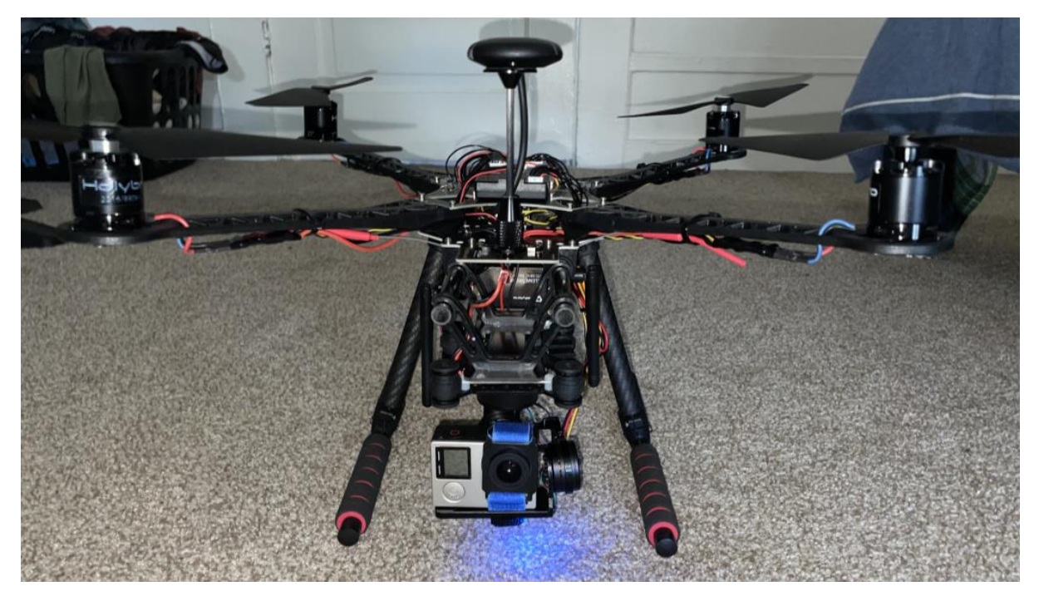
Figure 3: Drone Frontal View

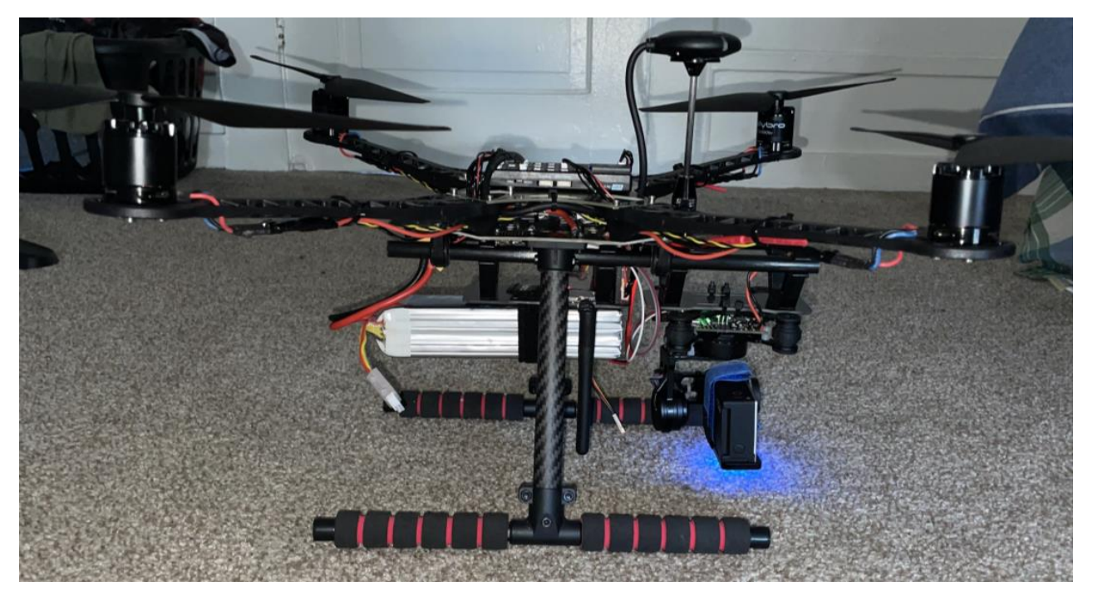
Figure 4: Drone Profile View

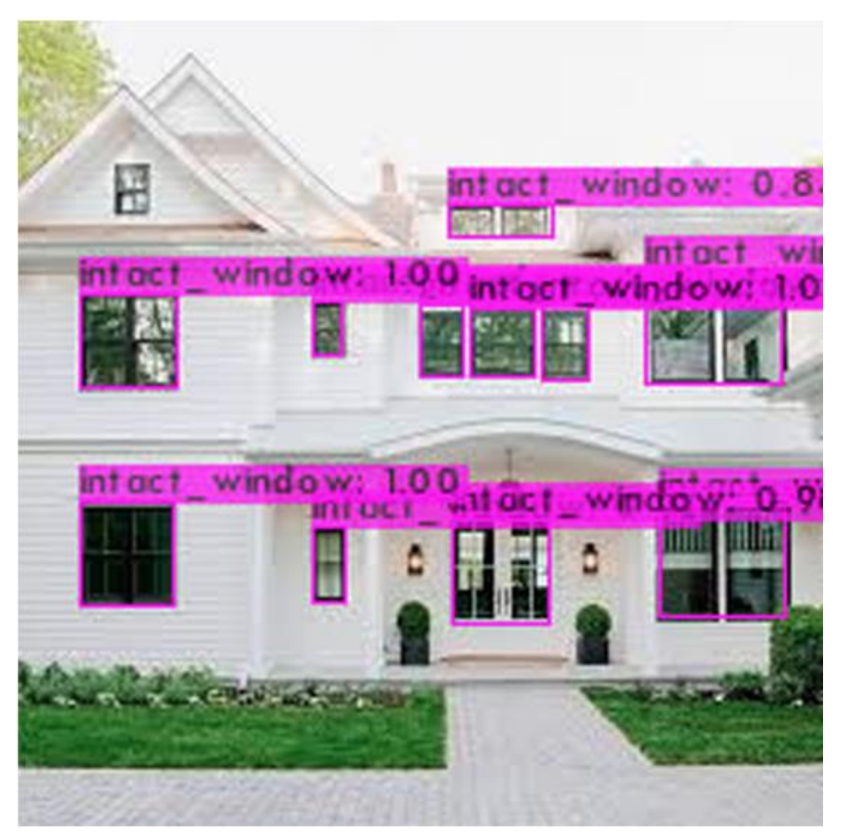

Figure 5: Sample Feature Detection Output

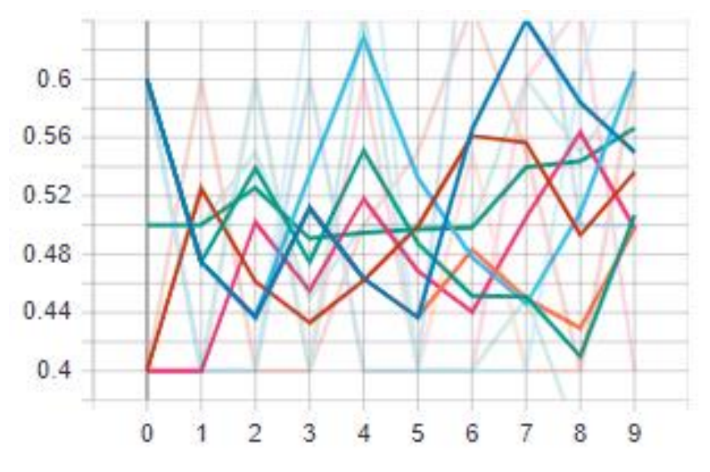
Figure 6: Gutter CNN Validation Epoch Accuracy

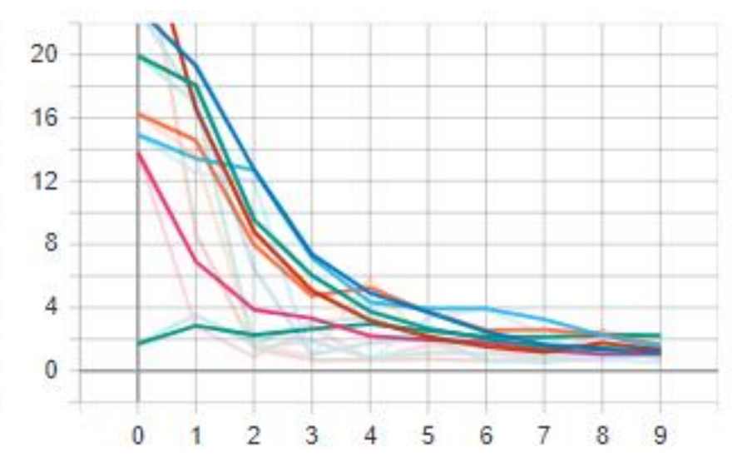
Figure 7: Gutter CNN Validation Epoch Loss

Conclusion

Ultimately, our device performed below standards, as far as the identification accuracy and latency thresholds are concerned. The overall implementation worked perfectly well, as did the general flow of image data since the high-level wrapper code ensures that all the inputs and outputs are normalized and in the proper format. The video input stream, YOLO output cropping function, feature isolation and image filtering, annotated display, and action generation all work completely nominally. We were able to store and implement the YOLO and TensorFlow models and weights files and retrain them as needed without having to change the wrapper code. We had hoped to achieve at least an 80% overall precision, which we were unable to do with such limited datasets; however, we were able to get close enough to create a functioning system, regardless of accuracy or correctness.

One of the biggest challenges for this project was developing the machine learning models. We gathered our own dataset utilizing a mix of images gathered around flagstaff as well as the internet. Finding high quality images of broken or damaged items was very difficult. In the future, the dataset is the area primarily in need of improvement. Doing so would vastly improve the quality of our object detection and image classification, and furthermore the performance of the entire system.

References

[1] C. Wang, A. Bochkovskiy and H. Liao, "Scaled-YOLOv4: Scaling Cross Stage Partial Network", arXiv.org, 2021. [Online]. Available: https://arxiv.org/abs/2011.08036. [Accessed: 10- Feb- 2021].
[2] Wotherspoon, J (2020) YOLOv4-Cloud-Tutorial [YOLOv4-Cloud-Tutorial]. https://github.com/theAIGuysCode/YOLOv4-Cloud-Tutorial

Acknowledgements

*Thank you to **Dr. Abolfazl Razi**, our client, who has mentored and pushed us in creating an innovative, compelling, and enriching project to encapsulate our engineering careers at NAU.*

*Thank you to **Han Peng**, our Graduate Teaching Assistant, who has guided us and helped us maintain a steady course throughout our capstone process.*