Memorial University of Newfoundland

Computer Science

Faculty of Science

Project Report: Exploring Image Preprocessing for Improved Vehicle Detection using YOLOv5

COMP 3301: Visual Computing and Applications

By:

	Student 1	**Student 2**
Name:	Anton Guaman	Aabir Basu
Student ID:	202014155	202021630

December 4, 2023

Exploring Image Preprocessing for Improved Vehicle Detection using YOLOv5

Prepared for: Steph McIntyre

Computer Science

Faculty of Science

Memorial University of Newfoundland

COMP 3301: Visual Computing and Applications

Prepared by:

	Student 1	**Student 2**
Name:	Anton Guaman	Aabir Basu
Student ID:	202014155	202021630

Date Submitted: December 4, 2023

Memorial University of Newfoundland

Table of Contents

List of Acronyms

AI: Artificial Intelligence

CNN: Convolutional Neural Network

DL: Deep Learning

EP: Extreme Programming

GM: General Motors

GPU: Graphical Processing Unit

ML: Machine Learning

NN: Neural Network

RGB: Red, Green, and Blue

U.S. DOT: United States Department of Transportation

YOLO: You Only Look Once

1. Introduction

1.1 Motivation

The motivation for this project stems from the Vision Zero initiative. The Vision Zero initiative aims to achieve zero road traffic deaths. The history behind this initiative is that it was adopted first in Sweden in 1997 and has expanded to other countries worldwide [1]. For example, the United States Department of Transportation (U.S. DOT) of the Federal Highway Administration has adopted the initiative and implemented the Safe System approach to achieve zero road traffic deaths [1]. The Safe System approach acknowledges human fragility by recognizing that people are susceptible to injury in car accidents and prone to making errors while driving. As illustrated in Figure 1, there are five layers of protection that the Safe System approach uses: safe road users, safe vehicles, safe speeds, safe roads, and post-crash care [2]. These layers of protection require working together to achieve zero road traffic deaths.

Figure 1: The Safe System Approach

In the car manufacturing industry, multiple companies have demonstrated support for the Vision Zero initiative by incorporating technology into their vehicles that will reduce human error rate. These companies follow the Safe System approach by strengthening the safe vehicle protection layer. Companies such as General Motors (GM) and Tesla have incorporated driver assistance systems into their vehicles, intending to reduce human error and be convenient for drivers [3][4]. The technology that these vehicles have heavily relies on cameras and a strong visual processing system.

This project primarily focuses on enhancing image preprocessing for improved vehicle detection using You Only Look Once (YOLO) version 5, a vital component in the technological advancements that support the Vision Zero initiative. As an introduction, YOLOv5 is a real-time fast object detection algorithm and will be elaborated more throughout this report. By improving the capabilities of vehicle detection systems, this project aligns with the safe vehicles protection layer of the Safe System approach. Utilizing image processing techniques can significantly augment driver assistance systems, a crucial feature in modern vehicles from companies like General Motors and Tesla. Also, this can help address the vision of some car manufacturers developing autonomous vehicles. Furthermore, the application of this technology is demonstrated by deploying the YOLOv5-trained model using an Intel RealSense D435 Camera. This practical implementation showcases how the advancements in vehicle detection can be integrated into real-world scenarios.

Other applications that this project addresses and can be applied to are traffic surveillance and smart city integration.

1.2 Solutions Studied

At the beginning of this project, when the proposal was submitted, the objective was to explore how modifying the YOLOv5 structure could yield improved vehicle detection. The approach to improving vehicle detection shifted as Professor Steph Mcintyre suggested the team members change their approach to exploring how image processing can improve vehicle detection while keeping the YOLOv5 structure as default. Professor Mcintyre suggested using preprocessing techniques that were learned throughout the semester. The solutions explored were to use a test set of normal red, green and blue (RGB) images, grayscale images, and sharpened images. The outcomes of these experimentations will provide insights into the most effective preprocessing techniques for optimizing vehicle detection with a YOLOv5 model trained on RGB images.

1.3 Implementation and Experiment Overview

The implementation of this project consisted of finding an open-source custom dataset curated with images for the task of interest, vehicle detection. The next step is to train the YOLOv5 on the custom dataset and obtain the YOLOv5-trained weights. Afterwards, the team members photographed vehicles around the Memorial University of Newfoundland (MUN) Campus to create a test dataset. This test dataset was carefully composed to include various vehicle types captured at varying distances and under different lighting conditions, both during the day and at night. The images from the test dataset were coloured, and this test dataset was used to generate the grayscale and sharpened test datasets. The last step before the experiment was to generate a Python script using the YOLOv5 trained weight to detect vehicles on an input image. The output after running this script is that a bounding box would be generated around the object of interest if detected.

After the implementation, the three test datasets were used as input on the image object detection Python script to generate the resultant images with bounding boxes after object detection had been applied to them using the YOLOv5 trained weight. After the experiment, three sets of resultant images of RGB, grayscale, and sharpened images were obtained. These sets of resultant images were compared to observe which image preprocessing technique improved vehicle detection.

Lastly, to showcase the real-life application, a Python script was created that uses the YOLOv5 trained weight and the Intel RealSense D435 camera to perform object detection on a real-time feed from the camera.

2. Topic Description

2.1 Formal Definition and Problems Studied

This project focuses on exploring different image preprocessing for vehicle detection enhancement using the YOLOv5 object detection algorithm. The specific challenge addressed in this project is to improve vehicle detection accuracy, a critical aspect in advancing driver assistance systems and autonomous vehicle technologies. The primary goal is to determine how different image preprocessing techniques can optimize the performance of the YOLOv5 model specifically for vehicle detection.

2.2 Essential Definitions

The following are definitions

- **Computer Vision:** a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information based on visual inputs [5].
- **Neural Networks (NN):** NNS are part of machine learning (ML) which are vital for deep learning algorithms (DL). A NN comprises an input layer, multiple hidden layers,

and an output layer, where each node's output is determined by a function of its weighted inputs and threshold [5].

- **Convolutional Neural Networks (CNN):** a DL algorithm tailored for classification and computer vision tasks that employ convolutional layers to automatically and efficiently extract and learn features from visual data [5].
- **Object Detection:** identify the location of objects in images. If an object of interest is identified, it will be surrounded by a bounding box. The bounding box generally has the class name of the object detected [6].
- **YOLOv5:** a fast, accurate real-time object detection algorithm developed by Ultralytics [7].
- **Roboflow:** computer vision developer framework. Aids with creating custom datasets, training models and deployment to production [8].
- **Single-Shot Multibox Detection (SSD):** a single-staged capturing object model that skips the proposal stage and achieves the final location and prediction in a single pass [9].
- **Two-Shot Detection:** a two-staged capturing object model in which it begins with a region proposal, continues to the classification of the proposed regions and finalizes with the refinement of the location prediction [9].

3. Solution Description

To be able to explore which image preprocessing technique would improve vehicle detection, the team members began by finding an open-source dataset that is curated for vehicles. This was a challenging task to overcome as there are a limited amount of open-source datasets that target the topic of interest. The dataset was found in Roboflow Universe, named Vehicle Class Specification Computer Vision Project. The reason behind finding a dataset online rather than

the team members creating their own is due to the project time constraints. In Table 1 it can be observed the vehicle types that the dataset select include.

Table 1: Vehicle Types from the Vehicle Class Specification Computer Vision Project Dataset [10]

Vehicle Type
Bus
Heavy 2 axles
Heavy 3 axles
Light 2 axles
Motorcar
Motorcycle

The following figures are annotated sample images from the dataset selected.

Figure 2: Bus from the Vehicle Class Specification Computer Vision Project Dataset [10]

Figure 3: Heavy 2 axles from the Vehicle Class Specification Computer Vision Project Dataset [10]

Figure 4: Heavy 3 axles from the Vehicle Class Specification Computer Vision Project Dataset [10]

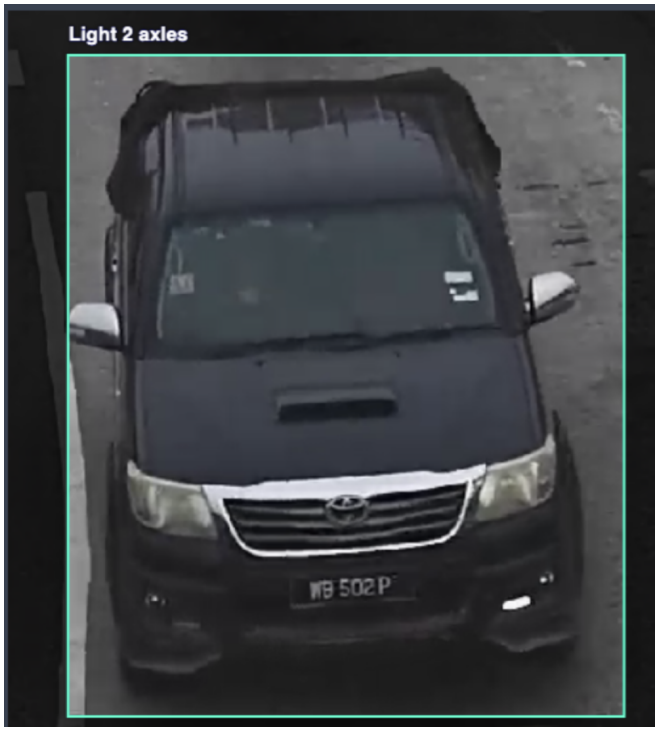

Figure 5: Light 2 axles from from the Vehicle Class Specification Computer Vision Project Dataset [10]

Figure 6: Motorcar from the Vehicle Class Specification Computer Vision Project Dataset [10]

Figure 7: Motorcycle from the Vehicle Class Specification Computer Vision Project Dataset [10]

The YOLOv5 object detection algorithm was chosen for this project as it follows an SSD approach and has a faster performance than other object detection algorithms that instead use two shot detection. Additionally, YOLOv5 was selected due to the amount of resources available online which would guide use if a challenge was faced throughout the implementation of the project.

After the dataset was selected the next step was to train the YOLOv5 model. The method used to train the YOLOv5 was utilizing Roboflow's Google Colab Notebook. The Google Colab allowed the team members to use a remote Graphical Processing Unit (GPU) as the training of an object detection model is computationally demanding. The dataset is imported into the training Colab Notebook and after two hours the training was complete. The YOLOv5 weights were extracted once the training culminated. In Figure 8 it can be observed the several graphs that constitute to the evaluation of the YOLOv5 detector performance.

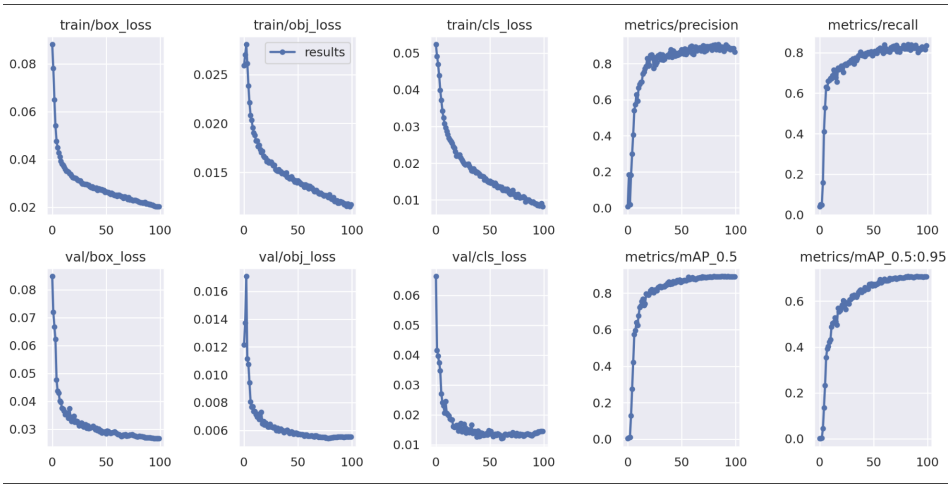

Figure 8: YOLOv5 Detector Performance on Vehicle Class Specification Computer Vision Project Dataset

The next step after this was to photograph vehicles through campus at different times of the day to generate a testing dataset. The testing dataset created has 154 images with diverse vehicles from different angles. The testing dataset has most of the vehicles the dataset selected has, but some vehicle types could not be found around campus. Additionally, null images were incorporated into the testing dataset to verify that no bounding boxes should be generated when an object of interest is absent.

The solutions explored regarding two image processing techniques are grayscale and sharpened images. The team members could generate a second testing set of grayscale images using the created RGB test dataset and the equations of converting an RGB image into an intensity image. For this solution, the team members used the equation provided in class and not the OpenCV function; therefore when preprocessing the RGB images to grayscale the time it took to process was significant. To generate the third dataset the grayscale test dataset images were used to create sharpened images inspired by assignment two. The team members followed the same approach of implementing their own functions to achieve a sharpened image instead of

OpenCV functions therefore when preprocessing the grayscale images the time it took to process was significant. The following figures are images from the three datasets: RGB, grayscale and sharpened.

Figure 9: RGB image from the RGB test dataset

Figure 10: Grayscale image from the Grayscale test dataset

Figure 11: Sharpened image from the Sharpened test dataset

3.1 Advantages and Disadvantages of Preprocessing Solutions

When a given image is converted to grayscale, the image's three-channel representation of red, blue, and green are all assigned the same value of the mean of the original values. To the human eye, the most drastic difference this makes is the loss of colour information. Humans are no longer able to pinpoint the exact colour any objects in the image are, though they are still able to tell if one region is darker than the other. This slight loss of information may be beneficial to the model's ability to detect edges - it has also been demonstrated in an edge detection study by researchers at the Chosun University, South Korea [11] that the specific colour-to-grayscale conversion technique also affects a given edge detection algorithm. YOLOv5's edge detection is performed by dividing the image into cells in a grid format. Each cell provides a confidence score of an edge or object being detected, and patterns are developed during training. Even though the human eye may no longer be able to discern colour differences, it can be safely assumed the prediction of the YOLOv5 model will perform better on grayscale test images. It must be noted that for images taken during the nighttime, converting to grayscale may not produce as many differences in the model's output compared

to images clicked in daytime lighting. This can be attributed to the fact that dimly lit images already contain lower levels of colour information, and taking their grayscale may even cause histogram concentration at a gray value. As a result, the model's performance may remain unchanged when the grayscale of a nighttime image was provided. The only other plausible scenario were converting to grayscale can negatively affect a model's performance is when its objective is to recognize specifically coloured objects.

The second image preprocessing technique to be discussed is that of unsharp masking. In this method, a blurred version of the original image is obtained using an averaging filter. This blurred version is then subtracted from the original image, causing edges and regions of largely varied pixel intensities (often called high-frequency components) to become accentuated. The resulting increased contrast at the edges in the image is favourable to the YOLOv5 model's edge detection performance. On the other hand, sharpening an image with any strategy is bound to amplify noise or introduce graininess in the image. This effect is especially noticeable in low-contrast and darker regions. To add to this, sharpening images that are of low quality or have been compressed by file encapsulation may further aggravate the noisy artifacts introduced by sharpening. To mitigate the amount of noise that is amplified and generated by the unsharp masking process, the team members made sure to use the small filter size of 3 pixels by 3 pixels. Though the sharpening effect was subtle due to this, it was decided to be better than potentially affecting the model's performance negatively by introduce unexpected noise.

Compared to the original RGB images, both the grayscale and the sharpened images can be expected to show better results from the YOLOv5 model. This can be attributed to the advantageous reasons proposed above, as well as the considerations made for the sharpening

process mentioned earlier. Both these preprocessing techniques simplify the test image as a whole leading to simpler computations. Grayscale images inherently contain less information, and thus led to favourable outputs from the YOLOv5 model. Sharpened images, in particular, are perfect to aid the performance of the model by enhancing the edges of potential objects and increasing the confidence scores of cells near edges and high-frequency components.

4. Experiment and Results

As expected, after running all three types of test images (RGB, grayscale, and sharpened) through the YOLOv5 model trained on vehicle recognition, the outputs obtained were significantly better when the image had been preprocessed than when it was not. This not only aligns with the predictions made above but also underscores why in most applications, a live camera feed is often simplified to black and white before object detection is performed. In this section, the discussion of the experiment is formalized by conducting an analysis of the results. This is achieved by examining a selection of example outputs produced by the YOLOv5 model when applied to various test images.

Figure 12: Resultant from YOLOv5 given RGB test image

In this first example, the YOLOv5 model was only able to detect one of the four vehicles in the frame of the RGB test image. Even though the image label is not visible in this example, the colour coding of the bounding box illustrates that the model also wrongly classified the right-most vehicle as a heavy vehicle with either two or three axles. The correct classification would be motorcar, which produces maroon bounding boxes.

Figure 13: Resultant from YOLOv5 model given sharpened test image

The same image, when converted to grayscale and sharpened, allowed the model to perform much better, as seen in Figure 13. All four vehicles have now been detected and classified correctly as motorcars. Clearly, the loss of visual colour information is helpful to the computations performed by the YOLOv5 algorithm for object and edge detection. The model's output was identical to the above image when given the grayscale test image.

Figure 14: Resultant from YOLOv5 model given a null RGB test image

In the second example result, null images are being considered. Null images are those that do not contain the object of interest and form a minority of the test images. These images are vital to examine any model's performance and ensure it does not misidentify the object of interest when it does not exist in an image. Here, the model had the desired output when given a null RGB test image, and did not draw any bounding boxes in the output image.

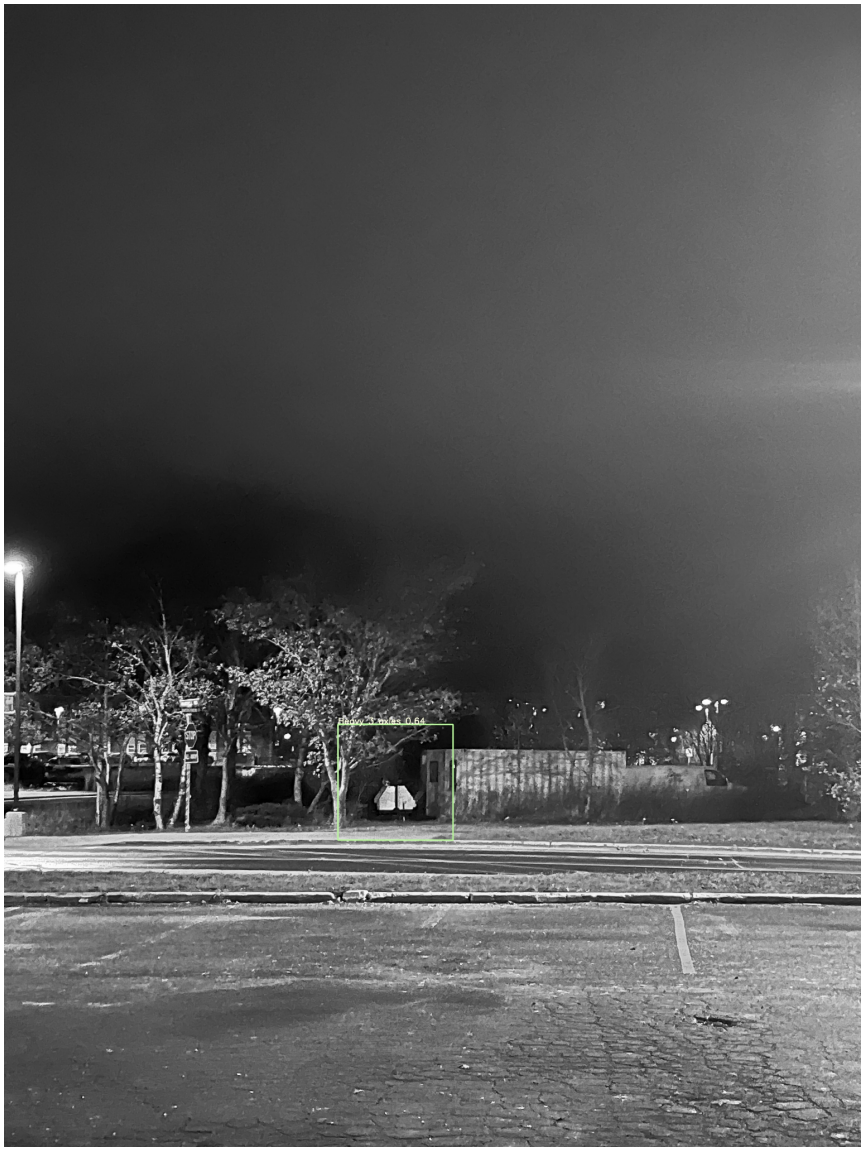

Figure 15: Resultant from YOLOv5 model given null grayscale image

When the null RGB image was converted to grayscale, the model produced an interesting output illustrated in Figure 15. Now, the model has detected a heavy triple-axled vehicle with an overall confidence score of 0.64. This is a false positive since there is no vehicle in this null image. This output illustrates the fact that even though the model's performance is often improved by performing image preprocessing, it may also lead to unexpected results. The false positive was also triggered for the sharpened test image with the same confidence score. It is clear from these observations that there is a threshold to the benefits provided by image

preprocessing, and after a point, it does not provide any net advantage by introducing artifacts as in the last example. To improve the performance after this point, given more time to complete the project, the team would be able to curate a custom dataset to act as the training images for the YOLOv5 model. Clicking images in the specific lighting and weather conditions of Newfoundland and mixing in appropriate stock images from the internet would undoubtedly go a long way toward improving the model's performance. Additionally, curating a training set of images would require manual annotations and labelling. This refers to drawing the bounding box in the image for the model to learn to recognize the pattern, and also classifying each bounding box to provide classification context to the model once the object has been detected.

Figure 16: Resultant YOLOv5 image from RGB test image

In the above Figure 16, the final example result can be observed - one of two vans that have not been detected by the model from the RGB test image. In the following Figure 17, it can be seen that when the image was preprocessed via both techniques under consideration, one of the vans was then detected and misclassified as a bus, and the other van was still not detected by the model. This example once again demonstrates the fact that it is impossible to perfect the performance of the model through image preprocessing alone, though there are discernible benefits that are to be exploited from it.

Figure 17: Resultant from YOLOv5 model given grayscale test image

5. Concluding Remarks

After the experiments conducted with the three test images given a trained YOLOv5 model, the team can state that image preprocessing improves the detection performance of a model, but not necessarily the classification accuracy. Preprocessing images also introduces certain distortions and artifacts that may lead to false positives and misclassifications. If the proper considerations are made to mute the effects of preprocessing that adversely affect the model performance, image preprocessing becomes a very favourable option to increase detection and classification accuracy. Even after a YOLOv5 model has been deployed onto a real camera, say the Intel Realsense D435 used during the project presentation, the script that deploys the model can be modified to first perform image preprocessing on the frame of the live feed before passing it through the YOLOv5 model. To make the Vision Zero Initiative a reality, research is being performed into several different technologies, the most prominent of which is autonomous driving. Providing vision to cars and decreasing the effect that human error has on road accident is of paramount importance, and even simple models such as the one being used in this project would be able to aid traffic safety. Apart from paving the way for the "safe vehicles" goal under the Vision Zero Initiative, the same vision technology using a YOLOv5 model can also be used to further the "safe roads" goal by deploying models at identified high-risk road locations to obtain pertinent accident data and modify road design to curb them. It is possible that for all these real-life applications, the model deployed is not fed the raw camera feed as input but rather a preprocessed frame from the feed in order to improve the performance. Obviously, before being deployed in such significant applications, a more thorough and time-consuming exploration of the effects of image preprocessing on a given YOLov5 model's performance would have to be performed, and this project and report may serve as a template for the results that one may expect.

References

[1] U.S. DEPARTMENT OF TRANSPORTATION Federal Highway Administration. "Zero Deaths and Safe System." [Online]. Available: https://highways.dot.gov/safety/zero-deaths. [Accessed: 25 October 2023].

[2] M. Doctor and C. Ngo. "Making our Roads Safer through a Safe System Approach." [Online]. Available: https://highways.dot.gov/public-roads/winter-2022/01. [Accessed: 1 December 2023].

[3] Tesla. "Autopilot and Full Self-Driving Capability." [Online]. Available: https://www.tesla.com/en_ca/support/autopilot. [Accessed: 1 December 2023].

[4] General Motors. "Path to Autonomous." [Online]. Available: https://www.gm.com/commitments/path-to-autonomous. [Accessed: 25 October 2023]

[5] International Business Machines. "What are convolutional neural networks" [Online]. Available: https://www.ibm.com/topics/convolutional-neural-networks#:~:text=Convolutional%20neural%20networks%20power%20image,inputs%2C%20it%20can%20take%20action. [Accessed: 1 December 2023].

[6] L. Hulstaert. "A Beginner's Guide to Object Detection." [Online]. Available: https://www.datacamp.com/tutorial/object-detection-guide. [Accessed: 25 October 2023].

[7] G. Jocher. "Yolov5." [Online]. Available: https://github.com/ultralytics/yolov5. [Accessed: 25 October 2023].

[8] Roboflow. “Roboflow.” [Online]. Available: https://roboflow.com/. [Accessed: 25 October 2023].

[9] “The Battle of Speed vs. Accuracy: Single-Shot vs Two-Shot Detection Meta-Architecture.” [Online]. Available: https://clear.ml/blog/the-battle-of-speed-accuracy-single-shot-vs-two-shot-detection. [Accessed: 1 December 2023].

[10] K. Izham. “Vehicle Class Specification Computer Vision Project.” [Online]. Available: https://universe.roboflow.com/khairul-izham-aje9q/vehicle-class-specification/browse?queryText=&pageSize=50&startingIndex=0&browseQuery=true. [Accessed: 25 October 2023].

[11] I. Ahmad, I. Moon, and S. J. Shin, “Color-to-grayscale algorithms effect on edge detection — a comparative study,” *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, Honolulu, 2018. doi:10.23919/elinfocom.2018.8330719 [Accessed: 2nd December 2023].

Contributions of Team Members

To develop all production code for this project, the pair programming technique was used. Pair programming is a teamwork approach in Agile software development, rooted in Extreme Programming (XP). It involves two developers collaborating on a single computer to collectively design, code, and test user stories. Ideally, both individuals are equally skilled and share equal keyboard time. In this setup, the programmer actively typing is known as the driver, while the other is the navigator, focusing on the overall programming direction. The navigator may even be looking ahead at the next problem facing the team to look for appropriate resources. This collaboration can occur in person or remotely with the emphasis being on effective communication between the two. The goal is to address challenges and discuss approaches that may be challenging for a solo developer. Despite its collaborative nature, pair programming requires specific skills in both soft aspects of teamwork and hard coding and testing abilities. The hunt for potential open-source datasets to serve as the training set of images was also performed as a group, with each individual first researching on their own and then comparing and filtering each other's results to get to the best possible dataset. To generate the set of test images, both members of the team spent time in various parking lots on campus, photographing both vehicles and null images. Once the original RGB set of test images had been obtained, scripts that had been developed by the team were used to generate the grayscale and sharpened sets of test images. Finally, the three different types of resultant images from the model were compared by both team members independently with the same conclusions discussed in the previous section. The presentation deliverable was also prepared by both team members equally dividing the slides between each other, and this final project report was also written by first dividing all the sections in the report outline equally between the two team members.