|  |  |
| --- | --- |
| Benefits Of The Raven Microwave Vehicle Detection System, 59% OFF  Vehicles Classification  Deep Learning Group Assignment | **Abstract**  This report explores the application of Neural Networks (NN) for vehicle detection in the context of road safety. It focuses on the significance of YOLO and transfer learning techniques, which enhance real-time vehicle detection. The methodology covers data preparation, CNN architecture, and model configuration. Results indicate strong model performance, especially in reducing false positives and false negatives. Challenges and limitations, including real-time processing and resource constraints, are addressed through techniques like batch normalization and dropout layers. The research demonstrates the practical applicability of the model in enhancing road safety.  Siwaphorn Charoenwitworakit  Yash Patkar |

# Table of Contents

[Introduction 3](#_Toc150710245)

[Literature Review 3](#_Toc150710246)

[Methodology 4](#_Toc150710247)

[Results 5](#_Toc150710248)

[Challenges And Limitation 7](#_Toc150710249)

[Conclusion 8](#_Toc150710250)

[Bibliography 9](#_Toc150710251)

# Table of Figure

[Figure 1: Confusion Matrix Table 5](#_Toc150519752)

[Figure 2: Learning Curve between Train vs Value Loss and Train vs Value Accurate 6](#_Toc150519753)

[Figure 3: Training model with Epoch of 10 6](#_Toc150519754)

[Figure 4: Interface represent the model predicted Vehicle and Non-Vehicle 7](#_Toc150519755)

# Introduction

The development of AI-based road safety monitoring systems has gained significant importance in modern societies. One crucial aspect of these systems is the detection of vehicles from video frames, which plays a fundamental role in preventing road accidents and maintain law and order in the society. This report focuses on the application of Neural Networks (NN) for vehicle detection in the context of road safety. The objective is to implement NNs to accurately identify vehicles in real-time video streams, thereby contributing to the overall goal of enhancing road safety. Vehicle detection plays a pivotal role in ensuring the safety of road users, reducing traffic congestion, and enabling efficient traffic management. The deployment of neural network-based models for this purpose represents a significant advancement in the field [1]. This report delves into the methodologies and techniques employed in this pursuit, highlighting the importance of robust model architecture, training strategies, and real-world applications of vehicle detection using neural networks.

# Literature Review

The integration of YOLO (You Only Look Once) into the realm of road safety and vehicle detection has had a profound impact on the development of advanced and efficient systems for various applications. YOLO is a pioneering object detection model that excels in real-time, single-shot object detection. What sets YOLO apart is its ability to process an entire image in a single forward pass, which leads to both high accuracy and rapid inference. This feature has made it a top choice for vehicle detection applications [2].

In addition to YOLO's core capabilities, the use of transfer learning techniques has played a pivotal role in expediting the development of vehicle detection systems. Transfer learning involves taking pre-trained neural network models such as VGG16, ResNet, or MobileNet and adapting them for specific tasks. This approach has several advantages. It reduces the need for extensive data collection and training time, as the models have already learned valuable features from massive datasets like ImageNet. By fine-tuning these pre-trained models for vehicle detection, developers can save time and resources while achieving accurate results [3]. These strategies have practical applications in various real-time road safety systems, enhancing their accuracy and efficiency. Some of the notable applications include:

1. **Traffic Monitoring Systems**: YOLO-based vehicle detection is instrumental in traffic monitoring systems. It allows for the real-time analysis of traffic flow, congestion, and incidents, enabling authorities to make informed decisions and take action promptly.
2. **Automatic Toll Collection:** YOLO's high-speed object detection capabilities are valuable in automatic toll collection systems. It enables the identification of vehicles passing through toll booths, ensuring efficient and accurate toll collection without manual intervention.
3. **Autonomous Vehicles:** YOLO's real-time object detection is critical for autonomous vehicles. It helps these vehicles identify and track other vehicles, pedestrians, and objects in their environment, ensuring safe and efficient autonomous driving.

This research paper focusing on using the completed YOLO-based model for still image analysis, rather than real-time computation. This approach has its own merits, especially when dealing with scenarios where real-time processing isn't a requirement. The model will be integrated into an external application and tested with new images, which showcases its versatility and applicability beyond the initial training and validation datasets. The integration of YOLO and transfer learning techniques has revolutionized vehicle detection in the context of road safety, enabling accurate, efficient, and adaptable solutions for a wide range of applications.

# Methodology

The data for the research paper has a near equal distribution of images consisting of and not consisting of a vehicle, which indicates that the data is not skewed or biased. The methods applied start with normalization of the image which were reshaped to a 64 by 64 size, this was done to reduce the training time as increasing the shape of the image increases the training time considering the limited resources we have at our disposal. We also tried using images with a much higher shape that of 224 by 224 but the limited resources meant we had to drop that configuration as it was computational and time-consuming. Next, the data is split into the test and train factions with an 80 to 20 split ratio. We also use multiple methods of data augmentation to increase the number of data that is to be trained. Some methods are shear range, zoom range, rotation range, width shift, range height, shift range, horizontal flip, and validation split.

The CNN models we employed contained many layers with multiple neurons. We used multiple convolutional layers with varying numbers of filters (32, 64, 128, 512, and 512) and a kernel size of (3,3). The 'Same' padding preserves the spatial dimensions. Convolutional layers are essential for feature extraction. By gradually increasing the number of filters, the network can learn more complex and abstract features from the input images, which is particularly important for detecting vehicles in varying contexts. Pooling layers with a pool size of (2,2) are added after each convolutional layer. This helps down sample the feature maps, reducing the spatial dimensions and computational complexity. This can improve the network's ability to focus on the most relevant features while retaining important spatial information. Batch normalization layers are inserted after every convolutional and fully connected layer. This helps stabilize training by normalizing the input of each layer. It accelerates convergence and reduces the likelihood of overfitting [4]. It accelerates convergence, reduces the risk of vanishing/exploding gradients, and improves the overall performance and generalization of the network. After the last convolutional layer, a flatten layer reshapes the feature maps into a one-dimensional vector. This is necessary to transition from convolutional layers to fully connected layers.

We have several fully connected layers with different numbers of neurons (512, 256, 64, and 2 for the output layer). This performs the final classification based on the learned features. The number of neurons gradually decreases towards the output layer, allowing the network to make decisions about the two classes: "vehicle" and "non-vehicle". Dropout layers with a dropout rate of 0.2 are used after the 512 and 256-neuron dense layers. This is a regularization technique that helps prevent overfitting by randomly deactivating a portion of neurons during training. This enhances the model's generalization capability. The output layer consists of two neurons with a softmax activation function. A softmax activation function is used to compute class probabilities for the binary classification task: "vehicle" and "non-vehicle". We tried to implement a callback function where the model would stop training if the validation loss increased compared to the last attempt, but this did not yield a good result, so we dropped the callback and let the epoch run all 10 iterations in **Figure 3** [5]. We also tried three different models with three different combinations of layer and neuron counts, the most effective model had batch normalization.

# Results

To have a better usability of the model we used streamlit library in python to create an interface to input a completely new image outside the scope of the train test and validation data to test the accuracy of the model on real world data. The model we use has a very low rate of predicting false negatives and false positives as potayed by the confusion matrix in **Figure 1.** We can see the learning curve for the model in **Figure 2** which indicates a large improvement in the validation loss and validation accuracy after the first itteration of the model. **Figure 4** depicts the use of stremlit for the implimendation of the model to classify wheather the image is a vehical or not, testing by using a car picture and a cup of coffee picture. The result of the model testing shows a sucessfully predited a car image as a vehical and a cup of coffee picture as Non-vehical.

A yellow and purple squares with black text

Description automatically generated

Figure 1: Confusion Matrix Table

A graph of a train loss

Description automatically generatedA graph with a line and a line

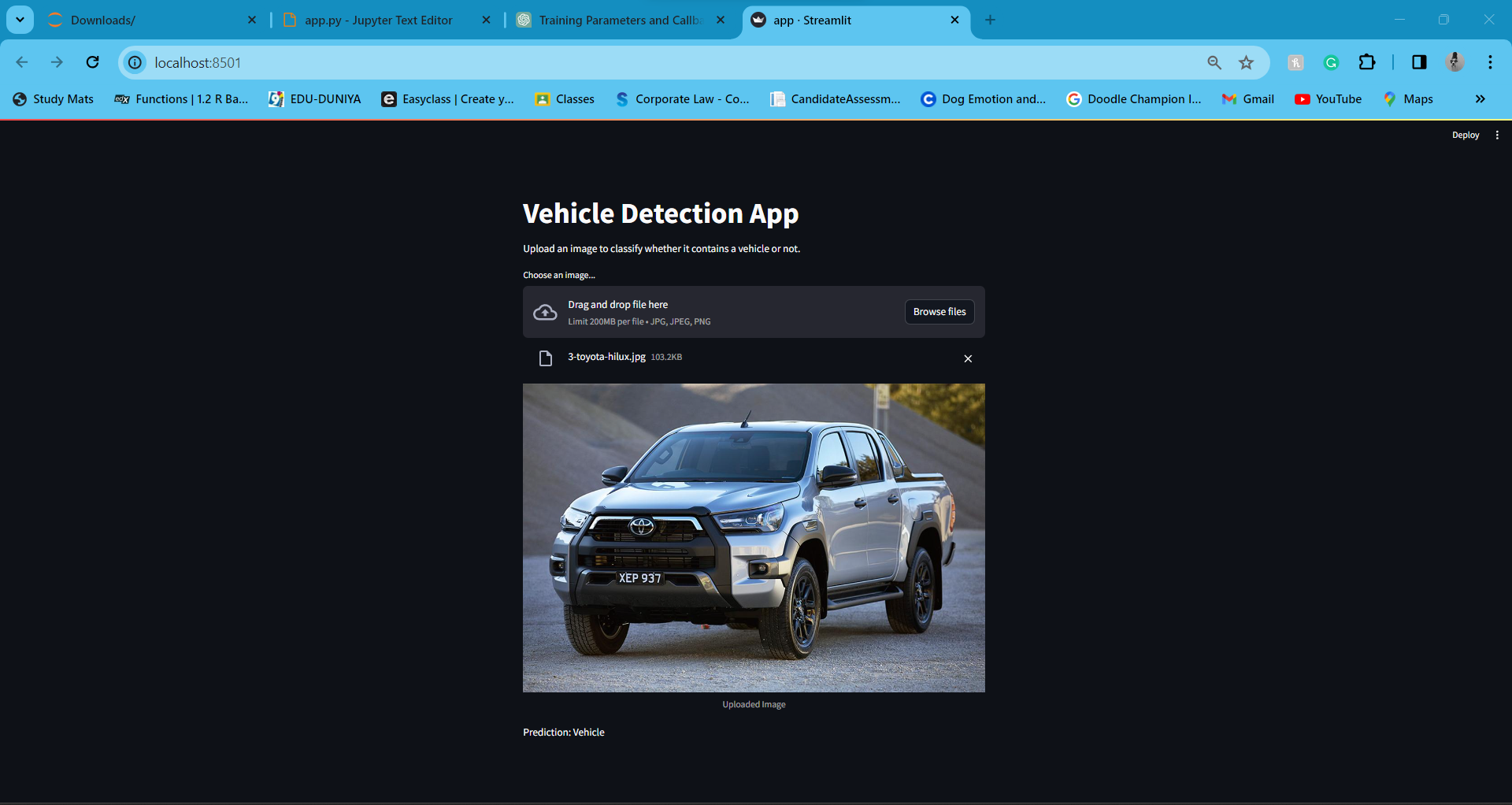
Description automatically generated with medium confidence

Figure 2: Learning Curve between Train vs Value Loss and Train vs Value Accurate

A screenshot of a computer screen

Description automatically generated

Figure 3: Training model with Epoch of 10

A screen shot of a computer

Description automatically generated

Figure 4: Interface represent the model predicted Vehicle and Non-Vehicle

# Challenges and Limitation

The primary challenge addressed in this research is the accurate detection of vehicles in real-time video streams using neural networks, particularly Convolutional Neural Networks (CNNs). Vehicle detection is a critical component of road safety monitoring systems, and the deployment of AI-based solutions to achieve this has gained significance. To overcome this challenge, the research aims to design and implement advanced neural network architectures and training strategies capable of making accurate detections without causing delays or latency in the monitoring system. Furthermore, vehicles in video frames can appear in various contexts and orientations, making it necessary to extract complex and abstract features for robust detection. The research tackles the challenge of developing CNN models that can effectively learn and extract these relevant features from input images [6].

Additionally, resource constraints, particularly limited computational resources, pose another challenge. As a result, the research optimizes the training process by employing techniques like image resizing and data augmentation to maximize the utilization of available resources. Ensuring that the neural network models generalize well is also a significant challenge, and techniques such as batch normalization and dropout layers are employed to prevent overfitting and enhance the model's ability to perform accurately on unseen data [7]. Finally, the research emphasizes not only model development but also its practical applicability in real-world scenarios. The challenge here is to demonstrate the accuracy of the model on new, unseen images outside the scope of the training and testing data, thereby proving its usefulness in enhancing road safety. Overall, this research addresses multiple challenges associated with vehicle detection using neural networks, aiming to provide a robust and efficient solution for road safety monitoring systems.

# Conclusion

In conclusion, this research has highlighted the critical role of AI-based road safety monitoring systems, particularly in the realm of vehicle detection from video frames. The use of Neural Networks (NN) for accurate vehicle detection is pivotal in reducing road accidents, ensuring traffic management, and enhancing road safety. We discussed the integration of YOLO (You Only Look Once) and the application of transfer learning techniques, which have significantly improved vehicle detection accuracy and efficiency. These advancements are crucial for various real-time road safety applications.

While our focus was on still image analysis rather than real-time computation, we successfully integrated the model into an external application, demonstrating its potential for real-world use.Despite our successes, we faced challenges, including resource limitations and the need for efficient real-time processing. Overcoming these challenges has led to a robust and efficient vehicle detection solution, contributing to the goal of road safety enhancement in modern societies.

# Bibliography

|  |  |
| --- | --- |
| [1] | Z. Zhang, C. Xu and W. Feng, "Road vehicle detection and classification based on Deep Neural Network," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), China, 2016. |
| [2] | J. Redmon, S. Divvala and R. Girshick, "You Only Look Once: Unified, Real-Time Object Detection," CVF, 2016. |
| [3] | A. Farid , F. Hussain , K. Khan, M. Shahzad , U. Khan and Z. Mahmood, "A Fast and Accurate Real-Time Vehicle Detection Method Using Deep Learning for Unconstrained Environments," MDPI, 2023. |
| [4] | L. Chen, . F. Ye, Y. Ruan, H. Fan and Q. Chen, "An algorithm for highway vehicle detection based on convolutional neural network," SpringerOpen, 2018. |
| [5] | Y. Rouizi, "Real-Time Vehicle Detection, Tracking and Counting in Python," The Python Code, Aug 2023. [Online]. Available: https://thepythoncode.com/article/real-time-vehicle-tracking-and-counting-with-yolov8-opencv#google\_vignette. |
| [6] | F. ELRIZANDA, "Vehicle Classification (CNN >98% accuracy)," Kaggle, Aug 2023. [Online]. Available: https://www.kaggle.com/code/fadhilelrizandamicr/vehicle-classification-cnn-98-accuracy. |
| [7] | Z. Wang, J. Zhan, C. Duan, X. Guan, P. Lu and K. Yang, "A Review of Vehicle Detection Techniques for Intelligent Vehicles," IEEE Transactions on Neural Networks and Learning Systems, 2023. |
| [8] | K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2015. |
| [9] | B. DINCER, "Vehicle Detection Image Set," Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/brsdincer/vehicle-detection-image-set. |