[ENGG\*680 INTRODUCTION TO DIGITAL ENGINEERING]

Real-Time Traffic Sign Recognition Using Convolutional Neural Networks to Address Emerging Traffic Violations

# Group 16 – Fall 2024

## Prepared for:

Dr. Hongzhou Yang, Rhea Zambra, Saurav Uprety, Zelin Zhou, and Ananya Vishwanath

|  |  |
| --- | --- |
| Prepared by: |  |
| Emallie Morin-Barich | 30072520 |
| Samira Hosseinzadehkordi | 30229100 |
| Anh Khoa Nguyen | 30262290 |
| Farima Toreihi | 30227977 |
| Henry Normanyo | 30230302 |
| James Inedu | 30265274 |
| Haroon Yaseen | 30246912 |
| Muhammad Talha Alvi | 30265332 |
| Final Report Date: |  |
| December 8th, 2024 |  |

|  |  |
| --- | --- |
| *Title of Assignment* | Final Report |
| *Group Number* | 16 |

We, the undersigned, certify that this is our own work, which has been done expressly for this course, either without the assistance of any other party or where appropriate, we have acknowledged the work of others. Further, we have read and understood the section in the university calendar on plagiarism/cheating/other academic misconduct and we are aware of the implications thereof. We request that the total mark for this assignment be distributed as follows among group members:

|  |  |
| --- | --- |
| *Your Name* | Emallie Morin-Barich |
| *Student ID* | 30072520 |
| *Contribution (%) and Hours* | 80%, 15 |
| *Signature and Date* | Emallie Morin-Barich 2024/12/07 |

|  |  |
| --- | --- |
| *Your Name* | Samira Hosseinzadehkordi |
| *Student ID* | 30229100 |
| *Contribution (%) and Hours* | 2%, 1 |
| *Signature and Date* | Samira Hosseinzadehkordi 2024/12/07 |

|  |  |
| --- | --- |
| *Your Name* | Anh Khoa Nguyen |
| *Student ID* | 30262290 |
| *Contribution (%) and Hours* | 2%, 1 |
| *Signature and Date* | Anh Khoa Nguyen 2024/12/07 |

|  |  |
| --- | --- |
| *Your Name* | Farima Toreihi |
| *Student ID* | 30227977 |
| *Contribution (%) and Hours* | 2%, 1 |
| *Signature and Date* | Farima Toreihi 2024/12/07 |

|  |  |
| --- | --- |
| *Your Name* | Henry Normanyo |
| *Student ID* | 30230302 |
| *Contribution (%) and Hours* | 2%, 1 |
| *Signature and Date* | Henry Normanyo 2024/12/07 |

|  |  |
| --- | --- |
| *Your Name* | James Inedu |
| *Student ID* | 30265274 |
| *Contribution (%) and Hours* | 2%, 1 |

|  |  |
| --- | --- |
| *Signature and Date* | James Inedu 2024/12/07 |

|  |  |
| --- | --- |
| *Your Name* | Haroon Yaseen |
| *Student ID* | 30246912 |
| *Contribution (%) and Hours* | 5%, 2 |
| *Signature and Date* | Haroon Yaseen 2024/12/07 |

|  |  |
| --- | --- |
| *Your Name* | Muhammad Talha Alvi |
| *Student ID* | 30265332 |
| *Contribution (%) and Hours* | 5%, 2 |
| *Signature and Date* | Muhammad Talha Alvi 2024/12/107 |

*\* Contribution total should be 100%.*

# Introduction

Urban traffic management faces significant challenges in monitoring stop sign compliance and ensuring pedestrian safety, particularly at 4-way intersections. Traditional methods of monitoring stop signs, including manual observation and basic automated systems, are labour-intensive, prone to human error, and often fail under adverse conditions. To address these limitations, this project proposes the development of a real-time traffic violation detection system using Convolutional Neural Networks (CNNs). The system aims to enhance the detection of 4-way stop violations by leveraging publicly available datasets and live camera feeds, ultimately contributing to improved pedestrian safety in urban environments.

## Objective:

The primary objective of this project is to develop a real-time traffic sign detection and recognition system that ensures pedestrian safety and enhances compliance at 4-way stop intersections. The system is designed to identify stop signs and other critical traffic signs, to detect violations, and to ensure that signs are properly recognized in varying conditions, such as changes in lighting or crowded scenes. The system aims to achieve high accuracy, speed, and robustness in real-world environments with the following measurable goals:

* **Accuracy target**: Achieve at least 90% classification accuracy.
* **Real-time processing**: Ensure a frame rate of 25+ FPS with minimal latency.
* **Error rate**: Maintain a low Mean Squared Error (MSE) for bounding box predictions.

## Literature Review:

Recent advancements in CNNs have revolutionized traffic sign detection. Models such as ResNet, VGG16, and MobileNet have demonstrated superior classification capabilities, while techniques like HSV colour transformations and shape-based algorithms have enhanced detection accuracy [1]. Hybrid approaches combining colour and shape detection further improve these systems, while attention-based models focus on key image features to refine results [2]. Additionally, one-stage networks like YOLO and SSD enable rapid detection, whereas two-stage networks, such as Faster R-CNN, provide more accurate region proposals for refined classification [3].

Despite these advancements, challenges remain. Small traffic signs, often critical at intersections, are difficult to detect accurately, leading to false positives and decreased system reliability. Weather conditions such as fog, rain, and snow further degrade performance. Dataset imbalances, particularly when traffic signs are obstructed or partially visible, also impact model accuracy. These challenges highlight the need for more robust detection methods in real-world environments [4]. Researchers have made strides in mitigating these issues through advanced architectures like ResNet50-D with attention-guided context feature pyramid networks and real-time image enhancement techniques [5]. AutoAugment technology has also been used to optimize data augmentation processes, further improving model performance. However, practical deployment in real-world environments remains a significant challenge due to the costs associated with implementation and the need for systems that can scale to diverse geographical locations [6].

Additionally, the integration of CNNs with other computer vision techniques, such as notification features for enhanced driver awareness, remains underdeveloped. Addressing these gaps in detection reliability, environmental adaptability, and integration could significantly improve the effectiveness of real-time traffic management systems [7].

## Scope and Problem Definition:

Pedestrian safety at intersections, especially at 4-way stops, is a critical issue in urban environments. Stop sign violations contribute significantly to pedestrian injuries and fatalities. Current monitoring methods, whether manual or automated, often fail to accurately detect violations, particularly under challenging conditions like bad weather or when traffic signs are obstructed. Additionally, small traffic signs are frequently misclassified or overlooked entirely, leading to inaccurate detection results.

This project aims to address these challenges by building a real-time detection system that leverages CNNs to accurately detect violations at intersections. By keeping models such as ResNet, VGG16, and MobileNet, enhanced with techniques like HSV transformations and shape- based algorithms, in mind for future integration, the system will work towards more reliable detection under a variety of conditions. The proposed solution will contribute to the field of engineering by applying advanced machine learning techniques in civil engineering, enhancing traffic management, and supporting pedestrian safety through interdisciplinary collaboration.

## Project Overview:

The project combines elements from multiple engineering fields—computer science, software engineering, civil engineering, and artificial intelligence—to improve urban traffic management and pedestrian safety. The core of the system is a CNN-based architecture designed to detect 4- way stop violations in real-time using publicly available traffic datasets and live camera feeds. By integrating various models and leveraging open-source resources, the system is designed to be scalable and adaptable for diverse urban environments.

Additionally, the project will address key challenges in stop sign detection, such as the misclassification of small traffic signs. The system will focus on providing accurate and timely information to traffic managers, supporting real-time decision-making, and ultimately contributing to safer streets for pedestrians.

This project represents an innovative approach to urban traffic management, demonstrating the potential of CNNs and machine learning to improve safety and efficiency in transportation systems. By focusing on 4-way stop violations and their impact on pedestrian safety, the project will provide valuable insights into traffic patterns and support the development of more effective traffic management solutions.

# Methodology

## Dataset Description

This project utilized the German Traffic Sign Recognition Benchmark (GTSRB) dataset, which is publicly available and commonly used for benchmarking traffic sign recognition models. The dataset provided a structured set of traffic sign images, along with corresponding labels, making it ideal for developing and testing machine learning models in traffic violation detection. The data is formatted in CSV files, which include key metadata such as file paths, class labels, and bounding box information for the traffic signs.

The dataset comprises a total of 51,839 training images and 12,630 test images, covering 43 different traffic sign classes. The primary feature used for training the machine learning model is the pixel values of these images, which are standardized for input into the model. The class label associated with each image is used as the "ground truth" for classification tasks, serving as the

basis for evaluating the model’s performance in correctly identifying and classifying the traffic signs.

For model training and validation, the dataset is split into two sets: 80% of the images are used for training the model, while the remaining 20% are set aside for testing. This split ensures that the model is exposed to a sufficient variety of images during training while retaining enough data for reliable performance evaluation.

## Model Selection, Justification and Validation

For this project, MobileNetV2 was selected as the primary machine learning model. MobileNetV2 is a lightweight CNN optimized for mobile and real-time image classification tasks, making it highly suitable for detecting traffic violations like stop sign compliance. Its efficient architecture maintains performance while reducing computational load, which is crucial when working with video streams or resource-constrained environments, such as traffic cameras or mobile applications.

MobileNetV2 also comes with pre-trained weights from ImageNet, allowing the model to leverage prior knowledge, speeding up training, and improving initial accuracy without requiring a massive dataset. This advantage allowed us to focus more on training the model to recognize traffic signs accurately in real-time video feeds, a critical aspect for detecting 4-way stop violations. To enhance MobileNetV2, two dense layers were added to enable both image classification (identifying traffic signs) and bounding box regression (determining the region of interest, or ROI, around the signs). The input layer accepts 224x224 pixel images with three colour channels (RGB), which are passed through a series of convolutional layers designed to detect critical image features such as edges and shapes. Pooling layers are incorporated to reduce dimensionality while retaining key information. These feature maps are subsequently processed through fully connected layers for classification into one of 43 traffic sign classes. The output layer employs softmax units to predict the most likely class label.

The system primarily used MobileNetV2 as the backbone model, with two separate branches for classification and bounding box regression. This multitask learning approach allows simultaneous detection and classification, making it highly efficient for real-time applications.

### Model Suitability for the Problem

MobileNetV2 was selected as the primary model due to its balance between accuracy and computational efficiency, which is critical for real-time applications, as well as its lightweight architecture, which makes it ideal for deployment on systems with limited computational resources. The model’s architecture is designed to minimize overfitting, and its performance across various deployment environments (e.g., mobile and web applications) has been well-documented in similar tasks. Additionally, its multitask learning capabilities enable end-to-end classification and bounding box regression, reducing the need for separate models.

YOLO was evaluated as an alternative to MobileNetV2 as the primary model. While YOLO offered superior speed for ROI detection, MobileNetV2 provided a balanced trade-off between accuracy and efficiency by incorporating multitask learning. However, MobileNetV2 is slower for ROI detection compared to YOLO. The final decision to use MobileNetV2 was based on its compatibility with classification tasks and ROI prediction within a single framework. However, to further enhance real-time performance, a hybrid approach was explored that combines YOLO for faster ROI detection with MobileNetV2 for more precise classification, which was later implemented during the live video feed integration process.

### Challenges and Limitations

#### Class Imbalance

One of the primary challenges was class imbalance within the GTSRB dataset. Certain traffic signs, such as stop signs, were underrepresented compared to more common ones, like speed limits. This imbalance made it harder for the model to accurately classify less frequent signs, leading to overfitting on the more abundant categories.

#### Overfitting

Overfitting was another issue that surfaced during training. The model achieved high accuracy on training data, but its performance on validation and test sets indicated difficulty generalizing to unseen examples. This issue was particularly noticeable when integrating the model into real- time video analysis.

#### Real-Time Video Accuracy

While MobileNetV2 performed well on static images, the accuracy of detection declined when applied to dynamic video streams. The model sometimes misclassified traffic signs or failed to detect them in video scenarios, which are more complex due to changes in lighting, angles, and motion blur.

### Addressing Challenges

#### Data Augmentation

To mitigate class imbalance and overfitting, data augmentation techniques were employed. By rotating, flipping, scaling, and altering the brightness of the images, the size and diversity of the dataset were artificially increased. This exposed the model to a broader range of examples, reducing the risk of overfitting and improving the generalization ability of the model.

#### YOLO Integration for ROI Detection

Additionally, the incorporation of YOLO was proposed to improve ROI detection. YOLO's robust object detection capabilities paired with MobileNetV2’s image classification can enhance both the localization and identification of traffic signs, especially in dynamic video streams.

## Processing

#### Preprocessing Steps

All images were resized to the 224x224 pixel input size required by MobileNetV2. Labels were one-hot encoded for multi-class classification while bounding box coordinates were normalized to values between 0 and 1 to ensure consistency across different image dimensions. To enhance the model’s generalization capabilities, data augmentation techniques such as flips, rotations, and lighting adjustments were applied, effectively expanding the variety of the dataset. The cross- entropy loss function was chosen to optimize the classification tasks, with Stochastic Gradient Descent (SGD) used as the optimization algorithm, ensuring effective updates of model parameters and faster convergence during training.

#### Python Libraries

* Pandas: Data manipulation and handling.
* NumPy: Matrix operations and image handling.
* OpenCV: Real-time video processing and drawing bounding boxes.
* Keras/TensorFlow: Building, training, and fine-tuning the MobileNetV2 model.
* Scikit-learn: Implementing metrics like accuracy, precision, recall, and confusion matrices.
* Matplotlib: Visualization of model performance through confusion matrices and precision-recall curves.
* Google Colab: For running and evaluating the model using GPU resources.

Model Architecture

The architecture of the traffic sign detection and recognition system centred around MobileNetV2. To customize the model for the use case, the pre-trained base was imported with ImageNet weights, and the top fully connected layers were excluded to allow for task-specific layers to be added. The input layer was defined with a shape of (224, 224, 3), corresponding to the RGB images used in this project:

This setup allowed the model to be retrained on the traffic sign dataset while retaining the powerful feature extraction capabilities learned from ImageNet, speeding up the learning process and improving the overall accuracy with fewer training samples.

#### Custom Layers

To tailor MobileNetV2 for multitask learning, the architecture was extended by adding two custom branches: a classification branch and a bounding box regression branch. These two branches enable the model to simultaneously identify the traffic sign class and predict the bounding box coordinates within the same forward pass, optimizing both accuracy and computational efficiency.

1. Classification Branch: This branch consists of fully connected layers that culminate in a softmax activation function, which outputs the probability distribution across the different traffic sign classes. Softmax ensures that the output values represent the likelihood of the image belonging to each class, making it useful for categorical classification tasks like this one. The number of output neurons corresponds to the number of traffic sign categories in the dataset.
2. Bounding Box Branch: The second branch is designed to handle the bounding box regression task. It also consists of fully connected layers but uses a linear activation function to predict the four coordinates that define the bounding box: [x\_min, y\_min, x\_max, y\_max]. These coordinates represent the top-left and bottom-right corners of the box surrounding the detected traffic sign. Linear activation is ideal for this regression task because it allows the model to output continuous values.

#### Combined Model

Both branches—the classification and bounding box branches—are integrated into a single end- to-end model, allowing the system to output both the predicted class of the traffic sign and its corresponding bounding box coordinates simultaneously. This multitask learning setup reduces the need for separate models for classification and localization, leading to better performance in terms of speed and consistency. By merging these two tasks into one pipeline, it was ensured that the model learned a unified representation of the traffic sign, improving the accuracy of both detection and classification.

This combined architecture was then compiled and trained, with the classification and bounding box predictions evaluated together, allowing the model to learn how to detect and classify traffic signs in real time. This approach streamlined the workflow and optimized the processing speed without sacrificing accuracy.

### Model Training and Validation

The dataset was split into 80% training and 20% validation data to assess performance. Cross- validation was also employed to ensure that the model generalizes well across different data subsets. To further prevent overfitting, early stopping was implemented, halting the training process if validation accuracy failed to improve after three consecutive epochs.

#### Hyperparameter Tuning

Multiple configurations of MobileNetV2 were experimented with, including variations in input size (e.g., 160x160 vs. 224x224) and adjustments to layer structures. The configuration that yielded the best results involved resizing the input images to 224x224 pixels and using augmented data. This configuration provided the highest classification accuracy and reliable bounding box predictions.

## Model Versioning, Testing and Optimization

#### Model Versioning

Tracking different versions of the model throughout the project lifecycle was critical to ensure reproducibility, collaboration, and experiment tracking. Model architectures, hyperparameters, and datasets were tracked, which enabled the team to reproduce experiments and validate results.

#### Model Testing and Optimization

Extensive testing was performed on smaller dataset subsets to evaluate model robustness and identify failure points. By integrating YOLO-based ROI detection, region-specific classifications were optimized, improving both precision and accuracy. The hyperparameters post-training were also fine-tuned, such as the learning rate and batch size, to maximize model performance.

Testing the model on smaller subsets of the dataset allowed for a detailed evaluation of performance and the optimization of detection workflows. A particular focus was placed on improving precision and accuracy for region-specific classifications of traffic signs through YOLO-based ROI detection. Experiment tracking was utilized to monitor changes, enabling the identification of failure points and necessary adjustments, which helped refine model behaviour across varying conditions. Furthermore, post-training optimization included hyperparameter tuning—adjusting the learning rate, batch size, and the number of epochs—to ensure a balance between convergence speed and overfitting prevention. Fine-tuning YOLO’s confidence threshold and bounding box predictions also improved real-time detection reliability.

## Live Camera Implementation

For real-time video stream processing, OpenCV was selected as the core framework due to its powerful tools for video input handling and frame analysis. Integrating live camera feeds into the system began by capturing frames from the camera and preprocessing them to meet the input requirements of the CNN model. This included resizing the frames to match the model’s expected input dimensions of (224, 224, 3). A streamlined pipeline was developed, allowing these frames to be sequentially fed into the model for real-time predictions while minimizing delays.

During the integration phase, challenges such as latency and misclassification were encountered. Latency, in particular, was a key issue because it affected the system’s ability to provide timely predictions. To address this, optimizations were made in the frame capture and preprocessing steps. Additionally, a checkpoint system was implemented, which allowed the model to load pre-

trained weights quickly, avoiding redundant retraining and reducing initialization times during testing.

## Testing

The model underwent several stages of testing. Initially, it was tested on pre-recorded images to validate its capability to detect and classify traffic signs. During this phase, issues like overfitting to certain classes and difficulty in detecting smaller traffic signs emerged. Once the model achieved satisfactory performance on pre-recorded images, testing transitioned to live camera feeds, where it was evaluated under real-world conditions, including varying lighting, different camera angles, and changes in distance from the traffic signs.

During live testing, the system faced misclassification and detection lag, particularly when signs were small or far away. To address these challenges, the dataset was augmented to balance class representation and improve detection accuracy. Additionally, ROI thresholds were adjusted to ensure the model focused on the most relevant parts of each frame. This iterative process of testing, refining the model, and updating the dataset allowed the system to adapt and improve, especially when distinguishing between similar traffic signs or detecting signs in challenging environments.

# Results

The model demonstrated strong performance, achieving impressive results in both classification and bounding box detection. Key metrics include:

* Accuracy: 99.53%
* Precision: 99.58%
* Recall: 99.53%
* F1 Score: 99.54%

For bounding box detection, the mean squared error (MSE) was 134.0380, and the average Intersection over Union (IoU) was 0.8312, with 99.3% of detected bounding boxes having an IoU of 0.5 or greater. Additionally, the model consistently achieved over 95% accuracy on both the training and testing datasets, reflecting its robustness in classification tasks.

However, when evaluating on live video, the frame rate averaged ~20 FPS, slightly below the desired real-time target of 25+ FPS. This is particularly noticeable in complex scenes, where more computational resources are required to process the additional visual information.

The integration of YOLO significantly improved detection efficiency, reducing the time required to detect traffic signs in each frame. Moreover, combining MobileNetV2 with YOLO enhanced ROI detection accuracy, making the system more effective in real-time video feed classification.

# Analysis/Discussion

The performance of the traffic violation detection system, particularly focused on detecting 4-way stop violations and stop sign compliance, revealed several key insights into the strengths and shortcomings of the model. While the model demonstrated commendable accuracy in recognizing stop signs under standard conditions, several areas of improvement emerged when handling real- world data, reflecting some limitations that may hinder its broader application in a live environment.

One of the primary shortcomings identified was the model's occasional misclassification of traffic signs, especially in situations where environmental factors like poor lighting, inclement weather, or obstruction by vehicles impaired the visibility of the signs. These misclassifications largely stem from data quality issues in the training set. Despite the robustness of the GTSRB dataset, it does not comprehensively cover all the environmental challenges found in live traffic conditions. The field tests highlighted how real-time factors, such as shadows or glare from the sun, resulted in diminished model performance when processing stop signs in some scenarios.

Another limitation was related to the complexity of the features used in training the model. While CNNs were employed for their proven efficacy in image recognition, certain visual nuances in stop signs, particularly those partially obstructed or heavily weathered, presented difficulties. The model’s feature extraction process struggled in these instances, leading to confusion between stop signs and other red-coloured objects, such as advertisement boards or tail lights of vehicles.

To address these shortcomings, several potential improvements could be implemented in future iterations of the model. Firstly, augmenting the dataset to include a wider variety of environmental conditions—such as more data representing diverse lighting, weather, and occlusion scenarios— could enhance the model's robustness in live applications. Another avenue for improvement is hyperparameter tuning, which could optimize the model’s performance by adjusting parameters such as learning rates or dropout rates. Additionally, experimenting with different model architectures, such as incorporating attention mechanisms or transformer-based models, may improve the ability to focus on relevant areas in an image, reducing misclassifications.

Finally, the implementation of post-processing techniques, such as bounding box refinement and real-time ROI adjustments, could further bolster accuracy. By refining the ROI, the system would better detect stop signs and reduce interference from irrelevant objects. Incorporating these improvements will be essential as the model moves from theoretical testing to real-world deployment.

# Conclusion

In summary, the traffic violation detection system demonstrated solid initial results, effectively recognizing stop signs and identifying violations at 4-way intersections under controlled conditions. The integration of the GTSRB dataset provided a strong foundation for training, allowing the model to recognize stop signs with high accuracy in ideal conditions. However, the results also underscored key challenges that arose when transitioning from the controlled dataset to live testing scenarios.

The model’s performance revealed some of the inherent limitations in using pre-collected datasets like the GTSRB when attempting to apply them in real-world applications. The system struggled to maintain high accuracy when faced with the variability of real-world environments, such as poor lighting, weather conditions, and partial occlusions. Despite these challenges, the project succeeded in highlighting the potential for using machine learning in traffic violation detection, while also bringing to light critical areas that require further refinement for broader application. The implications of these findings are significant for the field of traffic safety and engineering. As the push for smarter cities and safer roads continues, machine learning-based systems for traffic management have the potential to drastically reduce accidents, especially those involving pedestrians. However, the project underscores the need for a more adaptable, real-world-ready approach to model training and evaluation. The key takeaway is that while the model performed

well in a structured environment, future iterations must focus on making it resilient to the unpredictable nature of live traffic data.

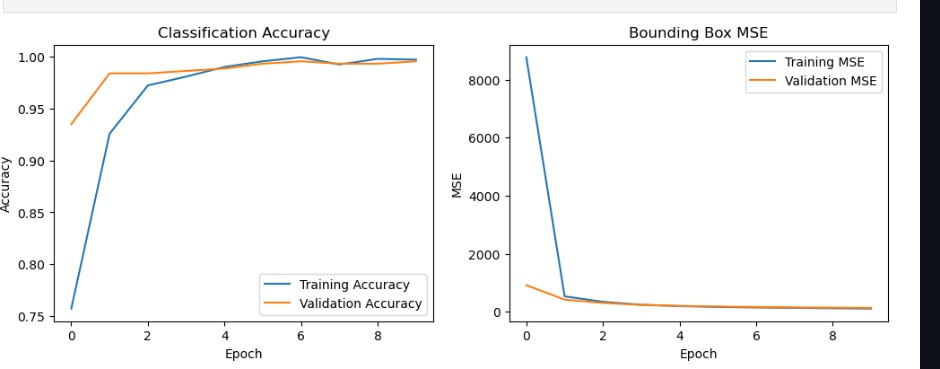
For future research, collecting a more diverse and extensive dataset that better represents the variations found in real-world traffic conditions is recommended. Additionally, refining the model by experimenting with different algorithms and focusing on hyperparameter tuning could further enhance its performance. Moreover, the integration of real-time feedback mechanisms, such as live camera adjustments or autonomous systems that adapt to changing environments, would be a crucial next step in expanding the application of this technology. In conclusion, while the system showed promise, ongoing development and testing are necessary to ensure its success in practical, everyday use.

# References

1. X. Zhang, X. Li, and W. Zhang, "Real-time traffic sign recognition based on RGB images," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 12, pp. 3896-3907, 2018.
2. J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," Neural Networks, vol. 32, pp. 323-332, 2012.
3. F. Yu, W. Zhou, and X. Li, "Hybrid approach for traffic sign recognition using color, shape, and CNN," Pattern Recognition Letters, vol. 92, pp. 71-78, 2017.
4. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real time object detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788, 2016.
5. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 2017.
6. E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation policies from data," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 113-123, 2019.
7. B. Xiao, G. He, Y. Zhu, and F. Yin, "Traffic sign recognition via attention-based convolutional neural networks," International Journal of Machine Learning and Computing, vol. 9, no. 6, pp. 803-809, 2019.

# Appendix A

Accuracy and Bounding Box MSE Graphs



# Appendix B

## Select Static and Dynamic Tests

