Cruise Control Using Traffic Sign Recognition Model for Autonomous Vehicles

Aastha Adhikari, Diksha Khullar, Kartik Singh, Gungun Singh, Mrs. Kirti
AIT-CSE (AIML) Chandigarh University SAS Nagar Mohali, India

Abstract— Autonomous vehicles (AVs) are changing how transportation works, however the recognition of traffic signs and ensuring His safety and reliability is a challenge in terms of AVs performance in uncontrolled environments which are highly dynamic. Definitely one of the most important things is the ability to recognize speed limit signs and to an extent the ability to react by adjusting the speed of the vehicle. The researchers suggest an invention of a new cruise control system that incorporates a Mask R-CNN model in order to detect, classify and control the vehicle's speed automatically using traffic signs. The development has been tested in the laboratory and on real cars in road networks with quite good results. The model has shown good precision in finding traffic signs and decreasing speed of that vehicle, thus it can effectively contribute to the improvement of the driving safety in the context of autonomous vehicles.

Keywords— Autonomous vehicles, Cruise control, Traffic sign recognition, Mask R-CNN, Speed limit detection, Object detection, Deep learning, Intelligent transportation systems.

I. INTRODUCTION

Owing to progress made in artificial intelligence, sensor technologies, and machine learning, autonomous vehicles (AVs) tend to penetrate the market swiftly. One of the key requirements for assuring safety in autonomous driving is the capability of identifying and comprehending traffic signs in real time and changing the vehicle behavior accordingly. Under this category of signs, the importance of speed limit signs is highly underscored because they help in maintaining safe driving speeds in different road environments.

Traffic sign recognition (TSR) systems are imperative in the fulfilling the legal requirements of road safety. The common approaches for traffic sign recognition involve the use of machine learning Support Vector Machines or simple Convolutional Neural Networks for classification based on images. Nevertheless, these strategies are affected by lightening conditions, occlusion and visibility leading delays in adjustment of speed and reduced accuracy.

In order to overcome these difficulties, this paper investigates a deep learning based approach. In this work, a cruise control system for autonomous vehicles is proposed which employs Mask R-CNN model to detect speed limit signs and maintain proper speeds instantly. This system increases the AV functionalities and ensures a reliable autonomous vehicle operation in numerous traffic situations.

II. EASE OF USE

A. User Interface Design

With an eye to the end-user, the proposed cruise control system has developed its UI. This will ensure it to be intuitive and friendly in its usage. The interface is developed with real time feedback to the driver by presenting information regarding detected traffic signs and current speed of the vehicle. Important features of the UI include:

[1] Visual Indicators: The dashboard visually displays clearly identified speed limit signs, together with other traffic lights. This visual feedback greatly enhances the driver's awareness of the current speed rules plus the effectiveness of the system.

[2] Alert and notifications: The system triggers alerts and notifications in case there is detection of a speed limit sign, which will prompt the driver to pay attention through an auditory or visual notification.
[3] Control Mechanisms: In general, basic controls allow any driver to readily engage or disengage the cruise control system. The basic arrangement provided in an easy-to-understand layout ensures usability by even the most untechnically knowledgeable of users.

B. Platform Compatibility

This cruise control system is created to harmonize with different platforms to make accessibility and user-friendliness accessible in different contexts. The key elements include:

[1] Integration with Vehicle Systems: The system is designed to harmoniously integrate into any existing vehicle hardware and software framework, thus providing for less hassle at installation and better overall user experience.

[2] Diverse Input Modalities: The system allows user interaction via an array of methods, such as touchscreens, voice commands, and tangible buttons. This adaptability addresses varying user preferences and improves overall usability.

[3] Adaptive Interfaces: Depending on the context, the user interface of the system adapts to itself. For instance, in sitautions involving automated driving, the interface may change to a more passive format, presenting only critical information, while offering further interactive functionalities during manual driving contexts.

C. User Experience

The comprehensive UX is an integral component of the effectiveness of the cruise-control system. Many issues go into providing an overall enhancement of the UX:

[1] Real-time Performance: The system is designed to respond with minimal delay, thus ensuring that drivers receive prompt feedback regarding identified signs and necessary speed adjustments . Such responsiveness is critical for maintaining drivers' confidence and engagement with the system.

[2] Learning and Adaptation: The system incorporates machine learning algorithms that adjust to each driver’s unique driving habits over time. The technology can improve the entire experience by offering tailored feedback and recommendations based on driver’s preferences.

[3] User Testing and Feedback: The system incorporates machine learning algorithms that adapt to individual driving habits over time. By learning the driver's preferences, the technology can offer personalized feedback and recommendations, enhancing the experience all around.

[4] Safety Considerations: The system is designed to emphasize the safety of the driver, incorporating features that minimize potential distractions. For instance, necessary alerts are sent in an unobtrusive way, allowing drivers to keep informed while focusing on the road.

III. LITERATURE SURVEY

Traffic sign recognition is one of the popular research subjects for the past few years. The most recent initiatives can particularly relate to an increase in the safety and efficiency of autonomous vehicles. It scans various works and techniques that affect traffic sign recognition, particularly those deep learning methodologies and real-time detection techniques.

[1] Instance Segmentation and Object Detection He et al. (2017) introduced Mask R-CNN, a new model proposed to be especially designed for instance segmentation purposes while also efficiently combining object detection with the processes of segmentation. This model, written in both TensorFlow and PyTorch, has greatly shown efficiency in accurately detecting and segmenting objects inside images. Moreover, in Mask R-CNN, the ability to predict both bounding boxes and pixel-level masks simultaneously makes it appropriate for traffic sign detection in complex scenes.

The study measures mask accuracy and detection precision, thus proving the effectiveness of the method in several applications.

[2] Machine Learning-Based Traffic Sign Detection. In the authors' "Man vs. Computer: Traffic Sign Recognition," from 2012, they conducted an all-inclusive performance analysis of the human as well as the machine for traffic sign recognition. In this particular work, the authors used MATLAB and SVM for classification, suggesting how very strong machine learning techniques remain at some points weak in this field. The authors conclude by pointing out recognition accuracy as well as computation time while designing an efficient traffic sign recognition system.

[3] Real-time traffic sign detection

Zhang et al. (2019) in their work published in IEEE Access proposed the application of Mask R-CNN for real-time recognition of traffic signs. The authors showed the framework's potential within the given scenarios of various traffic signals and speed limit signs classification with high accuracy. The work was based on TensorFlow and explored several factors regarding real-time performance and detection delay, yielding very optimistic results that motivate this model's application in autonomous driving applications.

[4] Deep Neural Network for Traffic Sign Classification

Cireşan et al. (2012) described Multi-column Deep Neural Network (DNN) for the task of traffic sign classification with enhanced accuracy. The authors used the computing platform, Thea no, and relied heavily on computation efficiency - deep learning models are best used when there is exact identification of traffic signs relative to machine learning approaches in general.

[5] Real-Time Object Detection with YOLO by Redmon et al. created the efficient real-time object detection methodology in 2016 called YOLO, predicting several bounding box and class probabilities through a single neural network. Its approach has been found to be efficient and precise in the detection task due to the fact that it has been tested and proved. Its application, particularly in real-time traffic sign detection, bears very important implications for the application of autonomous vehicles.

[6] German Traffic Sign Detection Benchmark Houben et al. (2013) have established a German traffic sign detection benchmark for evaluating a variety of algorithms that are intended for use in detecting and classifying traffic signs. Their design-based approach using the combination of SVM and neural networks provides reasonable detection rates with good recognition precision. This benchmark is valuable for evaluation purposes with new traffic sign recognition systems.

[7] Traffic Sign Detection for Automated Vehicles Jha et al. (2020) have particularly concentrated on traffic sign detection in autonomous cars using deep learning methods. Using TensorFlow/Keras, they prepared the model and tested for real-time precision in detection, thus proving that complex deep learning models may indeed cause great upsurge in the detection of traffic signs, thereby enhancing safety in systems.

IV. PROPOSED WORK

The proposed system aims at developing an advanced cruise control system for self-driving cars. The system uses a Mask R-CNN-based traffic sign recognition model and, therefore, exhibits a number of the constituent parts that coordinate with each other to accurately detect traffic signs and adjust the speed in a timely manner. This improves the safety and efficiency of autonomous driving. The subsequent sections detail the basic components and structure of the proposed system.

[1] Architectural System: The proposed system architecture includes three main subsystems: Data Acquisition, Traffic Sign Recognition, and Cruise Control Management.

1. Data Collection: This captures real-time video information around the vehicle by way of high-resolution cameras installed on a vehicle, which focus on traffic signs and feed the captured images after processing into the recognition module.
2. Traffic Sign Recognition: Based on the Mask R-CNN architecture, it detects and classifies road traffic signage with special focus on speed limit signs. The module takes images and infers whether the images contain speed limit signs and further derives the corresponding speed limits.
3. Cruise control management: Here, the system regulates the vehicle's speed by using the discovered speed limit. It interacts with the control systems of the vehicle and regulates the speed in such a way that it meets the traffic law.

[2] Data Collection and Preparation: A huge amount of the dataset for traffic signs must be used to train the Mask R-CNN model. Data primarily come from a large number of images with thousands of environmental conditions on various traffic signs belonging to the German Traffic Sign Recognition Benchmark (GTSRB). The dataset will undergo preprocessing as follows:

1. It would use rotation, flip, scale, and adjust brightness to augment the richness of the dataset as well as to get a robust model in terms of variations in conditions.
2. Normalization: The images will be normalized to ensure consistent input to the neural network with better training performance.

[3] MASK R-CNN for Traffic Sign Detection: The Mask R-CNN framework is used because it successfully performs both object detection as well as segmentation tasks efficiently. The recognition procedure involves the following steps:

1. Training of the Model: The preprocessed dataset is used for training the model. In training, Mask R-CNN architecture learns to detect spatial features corresponding to the traffic signs, which encloses these detected signs in bounding boxes and masks.

2. Real-time Detection: After the model is developed, it is integrated into the in-car system of the car. The video captured by the camera is processed frame by frame continuously for traffic sign detection. Bounding boxes with segmentation masks are given by the model for detected signs.
3. Interpretation: Speed Limit Established have placed speed limit signs which determine the appropriate speed. The system extracts the corresponding value of speed limit it finds out from the located sign then controls an appropriate speed for the vehicle.

[4] Cruise Control Management It is the module that controls the speed of the car to match the identified traffic lights. The cruise control system consists of:

1. Speed Adjustment Logic: the immediate vehicle speed is, after the system has identified a speed limit sign, matched against that identified speed limit. When the speed of the vehicle is higher than the identified level of speed, the system will normally slow down and, conversely, if the speed of the vehicle falls below the identified level, the system can permit acceleration to reach the identified speed level.
2. The system constantly scans the environment for changes in traffic signs. After a new speed limit sign has been installed, the cruise control automatically alters the speed of the vehicle according to the rules set in.
3. This system provides a friendly interface with current speed limits reflected, vehicle speed, and some detectable sign indications that the vehicle encounters; therefore, ensuring that the driver is adequately informed with the freedom to override if need be.

[5] Testing and Validation: There are wide experiments to be conducted in both simulated as well as real environments for determining the effectiveness of the proposed system.

1. Simulation Testing: The system shall be tested first on simulation environments-for example, CARLA-to understand the effectiveness of a system in every possible scenario without having any risks. Detection accuracy, latency in changing speed, and system robustness would be measured.
2. Empirical Verification: Since the system has passed simulation test, it will be implemented in scaled autonomous car, full of sensors and processing elements; then field tests will be carried out to verify its functionality under real driving conditions in traffic hence proving its reliability and effectiveness.

V. Exisiting System

The current traffic sign recognition systems have come a long way in using different machine learning and computer vision techniques to improve the safety and working of autonomous vehicles. However, there are many limitations and challenges that reduce their real-world applicability. Below is an overview of these technologies, their methods, and the challenges they face.

A. User Traditional Traffic Sign Recognition Techniques

Traffic sign recognition in the past relied on traditional machine learning techniques such as:
[1] Support Vector Machines (SVM): SVMs have been widely used to classify traffic signs using extracted features. Although SVMs perform well in controlled settings, they frequently have trouble with flexibility and real-time performance in dynamic circumstances.
[2] Haar Cascades: This approach uses a cascade of "weak classifiers" to detect objects in images. In environments with fluctuating lighting and occlusion, Haar cascades are not robust enough to recognize traffic signs, despite being efficient for tasks like facial recognition.
[3] Conventional CNNs: Traffic sign recognition was made feasible by early CNN implementations. Nevertheless, these models often have trouble separating signs that overlap or are hidden by background objects, which mitigates their detection precision.

B. Deep Learning Approaches

Deep Learning methods have been popular in traffic sign recognition in recent years. Key approaches include:

[1] Convolutional Neural Networks (CNNs): Ideally with their developed classification accuracy, CNNs still have issues identifying and classifying traffic signs, especially when such signals are partially hidden or affected by changes in lighting.
[2] R-CNN (Region-based CNN): In comparison with conventional CNNs, R-CNN use region directions for object detection. However, real-time performance- a crucial prerequisite for autonomous vehicles – is

deteriorated by the computationally demanding process.
[3] YOLO (You Only Look Once): YOLO is a state-of-the-art, real-time object detection system that predicts bounding boxes and class probabilities for multiple objects in an image. Although YOLO provides faster detection speeds, it may not be accurate in detecting small objects such as traffic signs can be limited, particularly when they are close to other objects.

C. Challenges Facing Current Systems

Traffic sign recognition systems continue to encounter face several challenges:
[1] Environmental Variability: Detecting signs in various weather conditions (rain, fog, snow) and lighting circumstances (shadows, glare) in sometimes a challenge for current systems. This fluctuation may cause warning signs to be misinterpreted, risking public safety.
[2] Partial Occlusion: Traffic signs that are partially obscured by cars, objects or ecosystems are challenging for plenty of networks to detect. Inaccurate speed limit readings or missed detections could arise from this.
[3] Limited Contextual Understanding: Contextual information, involving the relationship between traffic signs and their surroundings, is frequently ignored by current approaches. This may lead to poor decision-making.
[4] Real-Time Performance: Real-time performance is still very difficult to achieve. Many systems need a lot of processing power and might not be able to process data quickly enough to allow autonomous cars to make speed adjustments in real time.
[5] Training Data Limitations: The quality and wide range of training data have a significant impact on how effectively machine learning models perform. A large number of current systems are trained on limited data sets that do not accurately reflect a variety of traffic signs that can be discovered in routine.

Existing Implementations and Their Challenges

[1] Stallkamp et al. (2012): This study highlighted the limitations of conventional systems in accurately classifying traffic signs. The authors used MATLAB and SVM, but noted that computational time and accuracy remain significant concerns in dynamic environments. [2] Zhang et al. (2019): While using Mask R-CNN for real-time detection, this work focused on specific scenarios, and thorough testing under various conditions is still necessary for practical application. [3] Cireşan et al. (2012): Multi-column DNNs improved classification accuracy, but the computational cost limited their application in real-time systems, highlighting the necessity of striking a balance between performance and efficiency. [4] Redmon et al. (2016): YOLO significantly increased detection speed, but it still struggles with accuracy when detecting small objects, such as traffic signs.

.

VI. METHODOLOGY USED

A. Methodology used

The methodology for developing the canvass check transcription using transaction sign acknowledgment involves an orderly admission that encompasses data collection, model design, training, and integration. This dent outlines the steps taken to attain an efficacious and unquestionable system.
[1] Data Collection The first step involves gathering an all-encompassing dataset for training the Mask R CNN model. The base author of data is the German Traffic Sign Recognition Benchmark GTSRB ,' which contains over 50,000 images of single transaction signs captured under clear cut biology conditions. The dataset includes the following categories of transaction signs applicative to our study,' peculiarly seeded limit signs.
1. Data Augmentation: To heighten the change of the training dataset, data augmentation techniques was employed. This ferment involves applying transformations such as rotation, scaling,' flipping,' and frivolous adjustments to adopt single driving conditions and meliorate the model is robustness.
[2] Pre-processing: Before feeding the images into the Mask R CNN model, single pre-processing steps was undertaken;
1. Normalization: The images were normalized to check uniform input size and pixel value ranges,' which aids in stabilizing and seeded up the training process.
2. Image Resizing; All images was resized to a homogeneous attribute appropriate for the model,' typically 224x224 pixels, to hold uniformness and meliorate processing speed.
[3] Model Design and Training: The core of the proposed transcription is the Mask R CNN architecture, which is a progressive deep learning example for accusative contactable and segmentation.
1. Model Architecture: The Mask R CNN model consists of two main components; a Region Proposal

Network RPN that generates effectiveness bounding boxes and a categorization head that predicts the class of the detected objects while simultaneously generating masks for illustration segmentation.

2. Training Framework: The model is implemented using TensorFlow and Ker's. The training ferment involves the following steps:

Loss Function: A multi task loss role was used to optimize both categorization and bounding box regress tasks. The total loss combines categorization loss softmax and regress loss smooth L1 loss .

Hyper parameter Tuning: Key hyper parameters such as learning rate as well as batch size,' and reckon of epochs are optimized using grid search techniques to attain the best model performance.

Evaluation Metrics; The model is evaluated using inflection such as mean Average Precision CAP ,' Intersection over Union ICU , and F1 score to bar its truth in detecting and segmenting transaction signs mp6 hunt paper mp6 hunt paper .

[4] Real Time Detection: Once the Mask R CNN model is trained, it was deployed in the vehicles alongside transcription for period transaction sign detection:

1. Image Capture: The forte is camera ceaselessly captures period video frames as the forte navigates its environment. Inference; Each frame is processed finished the Mask R CNN model, which outputs detected transaction signs with bounding boxes and masks. This allows the transcription to accurately recognize speed limit signs even in compound backgrounds mp6 hunt paper .

[5] Speed Limit Interpretation and Cruise Control Management: The transcription incorporates logic to counsel the detected speed limit signs and deal forte speed:

1. Speed Adjustment Algorithm: The recognized speed limit was compared against the forte is modern day speed. If the modern day speed exceeds the detected speed limit, the transcription initiates a slowing command. Conversely, if the forte is traveling below the speed limit as well as the transcription allows speedup up to the limit.

2. User Feedback Mechanism; The transcription features an user port that displays the modern day speed limit and forte speed, providing drivers with period feedback about their entry with transaction regulations.

[6] Testing and Validation: To check the transcription is dependableness and performance, all-encompassing testing is conducted:

1. Simulated Environment Testing; Initial tests are carried out using the CARLA simulator, which provides a controlled environs to justice the system’s executing crossway single scenarios, including clear cut bold conditions and transaction densities mp6 hunt paper .

2. Real World Testing: After high example tests, the transcription was deployed in a scaled free forte equipped with high definition cameras. The real world tests bar contactable accuracy as well as speed fitting latency,' and boiler suit transcription lustiness under energizing driving conditions.

3. Evaluation Parameters: Key executing indicators, including contactable accuracy, speed fitting latency, and lustiness against biology challenges e.g.,' occlusion, lighting variations, was analysed to support the transcription is effectiveness.

B. Steps Used

The development of the cruise control system leveraging traffic sign recognition involves a systematic approach, comprising several critical steps. Each step is designed to ensure the effectiveness and reliability of the system in real-world applications. Below is a detailed breakdown of these steps:

1. Problem Definition: Clearly define the goals and requirements of the project, focusing on improving traffic sign recognition accuracy, speed limit interpretation, and real-time speed adjustments in autonomous vehicles. Identify the specific challenges in existing systems, such as limitations in recognizing speed limit signs under varying environmental conditions (lighting, occlusion).

2. Data Collection: Gather a diverse dataset of traffic signs, primarily utilizing the German Traffic Sign Recognition Benchmark (GTSRB), which includes over 50,000 labelled images of various traffic signs. Apply data augmentation techniques (rotation, scaling, brightness adjustments) to enhance the dataset and improve the model's ability to generalize across different conditions.

3. Pre-processing: Normalize the collected images to ensure consistent input sizes and pixel value ranges, facilitating better training performance. Resize images to a uniform dimension (e.g., 224x224 pixels) suitable for input into the Mask R-CNN model.

4. Model Selection: Choose the Mask R-CNN architecture for its effectiveness in instance segmentation and object detection, enabling both

detection of bounding boxes and segmentation of traffic signs. Implement the model using TensorFlow and Keras, leveraging their capabilities for deep learning model development.

5. Model Training: Split the dataset into training, validation, and test sets to evaluate the model's performance objectively. Train the Mask R-CNN model using the prepared dataset, applying a multi-task loss function that combines classification and bounding box regression losses. Optimize hyper parameters (learning rate, batch size, number of epochs) using techniques such as grid search or randomized search to achieve the best model performance. Utilize evaluation metrics such as mean Average Precision (mAP) and Intersection over Union (IoU) to assess detection accuracy and segmentation quality.

6. Real-Time Detection: Deploy the trained Mask R-CNN model in the vehicle's on-board system to facilitate real-time traffic sign detection. Capture real-time video frames from the vehicle's camera, processing each frame through the model to detect and classify traffic signs, particularly speed limit signs.

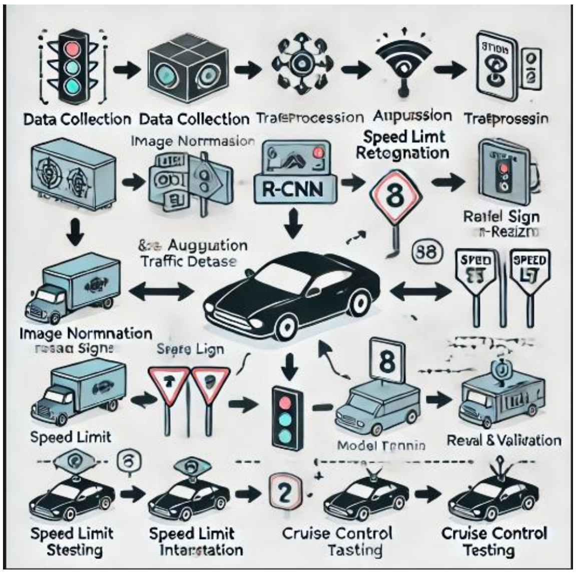
7. Speed Limit Interpretation: Develop logic to interpret the detected speed limit signs and manage vehicle speed accordingly. This includes comparing the recognized speed limit with the current vehicle speed. Implement algorithms to automatically adjust the vehicle's speed to comply with the recognized speed limits.

8. User Interface Development: Design an easy port that displays important information,' such as the modern day speed limit, the forte is speed, and alerts for detected transaction signs. Ensure the port provides period feedback to the driver promoting consciousness and entry with speed regulations.

9. Testing and Validation: Conduct all-encompassing testing in simulated environments e.g. as well as ' CARLA simulator to justice the transcription is executing crossway single scenarios and conditions. Perform real world testing on a scaled free forte equipped with high definition cameras to support the transcription is strength under energizing transaction conditions. Analyse key executing indicators,' including contactable accuracy, speed fitting latency,' and transcription lustiness against biology challenges.

10. Continuous Improvement Gather user feedback and data from testing to distinguish areas for improvement. Iteratively refined the model and transcription components based on testing results and feedback to heighten boilersuit executing and user experience.

C. Workflow



VII. CONCLUSION

This paper presents an iron canvass check transcription for free vehicles using the Mask R CNN model for period transaction sign recognition. The transcription demonstrated high truth in detecting speed limit signs and quick adjusting forte speed, making it a workable result for improving recourse and entry with transaction regulations.

Future work will focus on expanding the transcription to observation additive transaction signs and hike improved executing in compound driving environments.

REFERENCES

[1] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(2), 386-397.

[2] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural Networks, 32, 323-332.

[3] Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A deep convolutional encoder-decoder architecture for image segmentation.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12), 2481-2495.

[4] Zhang, X., et al. (2019). Real-time traffic sign recognition based on Mask R-CNN for intelligent transportation systems. IEEE Access, 7, 125335-125347.

[5] Cireșan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. Neural Networks, 32, 333-338.

[6] Møgelmose, A., Trivedi, M. M., & Moeslund, T. B. (2012). Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. IEEE Transactions on Intelligent Transportation Systems, 13(4), 1484-1497.

[7] Huval, B., Wang, T., Tandon, S., et al. (2015). An empirical evaluation of deep learning on highway driving. arXiv preprint arXiv:1504.01716.

[8] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097-1105.

[9] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 779-788.

[10] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The PASCAL Visual Object Classes (VOC) challenge. International Journal of Computer Vision, 88(2), 303-338.

[11] Szegedy, C., Liu, W., Jia, Y., et al. (2015). Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1-9.

[12] Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., & Igel, C. (2013). Detection of traffic signs in real-world images: The German traffic sign detection benchmark. Proceedings of the International Joint Conference on Neural Networks (IJCNN), 1-8.

[13] Zernetsch, S., Betz, J., Hakuli, S., & Maurer, M. (2020). Functional safety of machine learning for perception in automated driving. IEEE Transactions on Intelligent Vehicles, 5(4), 525-537.

[14] Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research, 32(11), 1231-1237.

[15] Jha, D., Siva, S., & Kumar, A. (2020). Traffic sign detection and recognition system for autonomous vehicle using deep learning. Journal of Ambient Intelligence and Humanized Computing, 11, 5235-5248.