Automatic speed control in self driving vehicles using Deep learning

Mukesh Sekar
Department of Electrical Engineering
Colorado state University
Fort Collins, USA
msekar@colostate.edu

Nipul Balram Avduth
Department of Electrical Engineering
Colorado state University
Fort Collins, USA
nipul369@colostate.edu

***Abstract*— Our project focuses on improving the safety and efficiency of self-driving vehicles through the implementation of deep learning. The system employs neural networks to dynamically control vehicle speed based on real-time analysis of sensor data and adapts to various traffic conditions. Additionally, the proposed system utilizes deep learning algorithms and Lidar sensor to detect and predict potential collisions, allowing the vehicle to take proactive measures to avoid accidents. The models are trained on diverse datasets, ensuring adaptability to different driving scenarios. Experimental results demonstrate the effectiveness of the approach, making it a promising solution for the safe deployment of self-driving vehicles in complex environments.**

Key Words: - Object Detection, RESNET, SSD, Classification, YOLO

I-INTRODUCTION

The evolution of autonomous vehicle technology promises a paradigm shift in transportation, offering heightened safety and efficiency on our roads. At the core of this transformative vision is the imperative to develop sophisticated systems capable of autonomously controlling vehicle speed, navigating dynamic environments, and actively preventing collisions. Deep learning techniques have emerged as a pivotal strategy to navigate the intricacies of automatic speed control and collision avoidance in self-driving vehicles.

The integration of autonomous vehicles into real-world scenarios demands a holistic understanding of the intricate interplay between speed adjustments, object detection, and effective classification. Conventional methods often encounter limitations in adapting to the unpredictable nature of diverse road conditions. This research seeks to overcome these challenges by leveraging cutting-edge technologies, including deep learning algorithms, LiDAR technology, and computer vision, to develop a more adaptive and responsive autonomous driving system.

This Project endeavors to contribute to the ongoing discourse by proposing a comprehensive system that leverages deep neural networks to enhance the safety and performance of autonomous vehicles. By focusing on the crucial aspects of adaptive speed control and proactive collision avoidance, the proposed system aims to overcome challenges associated with navigating diverse and unpredictable road scenarios. The integration of cutting-edge deep learning algorithms allows for real-time analysis of sensor data, enabling the vehicle to make informed decisions and respond to dynamic changes in its surroundings.

While our project strives to enhance autonomous driving capabilities, it is essential to acknowledge certain limitations. Our model may face challenges in operating effectively during nighttime conditions, and adverse weather conditions can potentially impact its performance. Recognizing these limitations, our focus remains on refining and expanding our system to mitigate these challenges in future iterations.

To tackle the complexities of autonomous driving, we employed a robust methodology centered around YOLOv5 and the Darknet framework. YOLOv5, known for its efficiency and accuracy in object detection, served as the cornerstone of our approach. The Darknet framework provided a versatile and powerful platform for implementing YOLOv5 in our project. This combination laid the groundwork for our exploration into automatic speed control, object detection, and classification in real-world driving scenarios.

With our scope outlined and limitations acknowledged, our methodology pivots around the utilization of YOLOv5 within the Darknet framework. As we delve into the specifics of our approach, the subsequent sections will unravel the intricacies of our model, shedding light on its implementation, challenges encountered, and the valuable insights gained throughout the course of this project.

II- RELATED WORK

Autonomous vehicle systems encompass a multifaceted domain, with numerous studies addressing pivotal aspects such as speed control, object detection, and classification [3]. The following review provides an overview of key projects in these areas, highlighting trends and methodologies that have shaped the landscape of autonomous driving research. Previous research has extensively explored methodologies for effective speed control in autonomous vehicles [2]. Works by Yi Zhang and his co-authors delve into adaptive speed control strategies, leveraging sensor data and real-time environmental information to optimize vehicle velocity. Object detection stands as a critical component in ensuring the safety and efficiency of autonomous vehicles.

Traditional object detection algorithms are mainly devoted to the detection of a few types of targets, such as pedestrian detection [5] and human motion recognition [6]. Seminal works by Ashwani Kumar and the co-authors have investigated object detection algorithms, with a particular focus on real-time implementations.

Techniques such as YOLO (You Only Look Once) [7] and SSD (Single Shot Multibox Detector) [8] have gained prominence for their ability to rapidly detect and localize objects. Beyond detection, projects led by Zhe Zhu [9] have explored object classification methodologies to augment the perception capabilities of autonomous vehicles. These studies emphasize the significance of accurately categorizing detected objects, enabling the vehicle to make informed decisions based on the nature of its surroundings.
We aimed to conduct experiments on object detection and classification using the Darknet framework, exploring various versions of YOLO. Subsequently, we proceeded with the model that demonstrated the highest accuracy. For classification tasks, we employed a customized CNN model and MobileNet. Based on the obtained results, we further proceeded with the MobileNet model. The integration of an embedded platform with cameras to interpret real-time data aligns with the practical implementation of the trained models. This step is essential for translating the learned information into actionable insights that can influence the behavior of the vehicle.
Our emphasis on speed reduction based on different road signs indicates a safety-oriented approach. Adjusting the vehicle's speed according to recognized signs enhances not only the safety of the passengers but also contributes to overall road safety.

III- DATASET

The data collection phase emerged as a pivotal aspect, given its profound impact on the model's accuracy. We encountered a challenge where the acquired data included a mixture of relevant information, such as speed signs, and extraneous data, which adversely affected the model's performance. To address this issue, we strategically categorized our dataset into two subsets: Data-1 and Data-2.

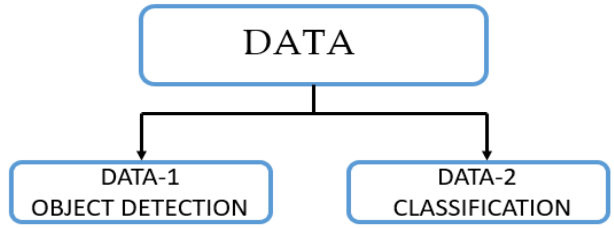

Fig1: - Distribution of Data

Data-1, sourced from a public GitHub repository, played a crucial role in the realm of object detection. This dataset, comprising images of real-time road signs annotated with bounding boxes specifying speed limits, was instrumental in training our model to recognize and delineate specific features within the visual data. The incorporation of bounding boxes facilitated the development of a robust object detection framework, enabling the model to precisely identify and localize speed limit signs in diverse road scenarios.

On the classification front, our initial dataset, Data-2, featured 58 distinct classes. Recognizing the need for a more focused approach, we strategically reduced the dataset to 8 classes, aligning with specific categories relevant to our objectives. However, the challenge arose when the dataset size dwindled to around 1300 images post-reduction.

Fig 2: - Sample Dataset of different Speed Signs

To overcome this limitation, we took a proactive approach by manually collecting additional data on-site. Physically visiting locations allowed us to capture images that complemented the existing dataset, adding a real-world dimension that is often challenging to replicate in synthetic datasets.

Fig 3: - Manual Dataset Sample

Additionally, to further enhance dataset size and diversity, we employed data augmentation techniques. These transformations not only expanded the dataset artificially but also improved the model's ability to generalize across different scenarios.

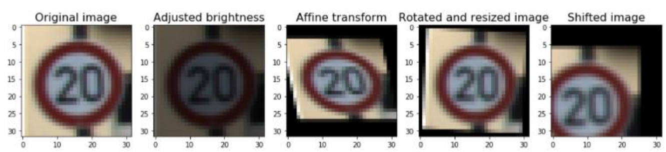

Fig 4: - Data Augmentation example

IV- METHODS

Our project is centered on object detection using YOLO, ResNet SSD, and MobileNet SSD, with a particular focus on identifying and locating traffic signs in images and videos. The approach involves leveraging the strengths of these popular deep learning architectures to achieve high accuracy and versatility in recognizing and localizing objects. Specifically, YOLO's real-time detection capabilities, ResNet SSD's enhanced feature representation through a ResNet backbone, and MobileNet SSD's efficiency with a MobileNet backbone are incorporated. The project recognizes challenges such as the need for sizable datasets, substantial computational power, and meticulous hyperparameter tuning. The emphasis on traffic sign detection is noteworthy, given its importance in applications like autonomous vehicles and driver assistance systems, contributing to advancements in smart transportation and road safety.

ResNet, or Residual Network, is a deep neural network architecture designed to overcome the challenges of training extremely deep networks. Introduced in 2016, its key innovation is the use of residual blocks with shortcut connections, allowing the network to learn residual functions and facilitating the training of very deep models. ResNet comes in various depths (e.g., ResNet-18, ResNet-50), with deeper variants achieving higher accuracy. It has been widely successful in image classification tasks and serves as a feature extractor for other computer vision applications through pre-training and transfer learning. The introduction of residual learning has had a profound impact on the ability to train and deploy deep neural networks effectively.

MobileNet is a type of convolutional neural network designed for mobile and embedded vision applications. Characterized by a streamlined architecture, it utilizes depth wise separable convolutions to build lightweight neural networks. This design choice reduces parameters and computations, making MobileNet suitable for devices with limited resources. The primary goals include achieving low-latency inference for real-time applications on mobile and embedded devices. MobileNet models offer versatility for various computer vision tasks, and their architecture allows for a trade-off between model size and accuracy. The models are compatible with deployment frameworks like TensorFlow Lite, facilitating their integration into mobile and edge devices.

SSD (Single Shot Multibox Detector) is an object detection algorithm designed for efficient and accurate detection of multiple objects in images. Introduced in 2016, SSD follows a single-shot approach, performing both region proposal generation and object detection in a single pass. Its architecture includes a base network (e.g., VGG16 or ResNet) followed by additional convolutional layers. SSD uses default boxes (anchor boxes) at different scales and aspect ratios to predict object locations. Predictions for each default box include a confidence score for object presence and adjustments to the box coordinates. The training process involves minimizing a multi-task loss function. SSD is known for its speed and efficiency, making it suitable for real-time applications such as autonomous vehicles and surveillance.

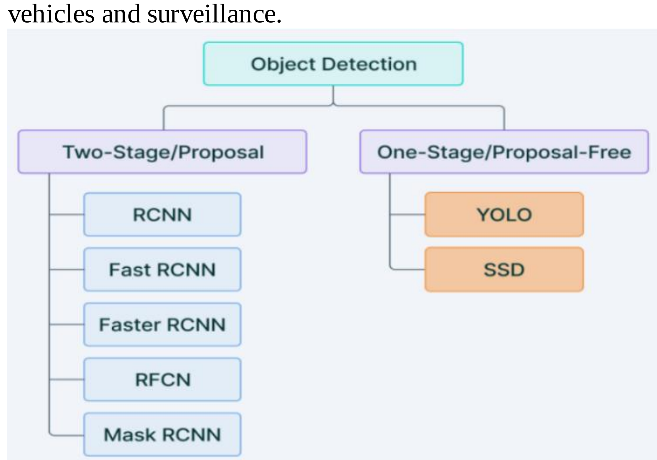

Fig 5: - Stages of Object Detection

(A) - RESNETSSD

Combining ResNet with SSD involves using a ResNet as the backbone for the Single Shot Multibox Detector (SSD) object detection framework. ResNet, known for its ability to capture complex features, serves as the feature extractor in the SSD architecture. This combination enhances the accuracy and robustness of object detection by leveraging the pre-trained ResNet model to improve feature representation in the detection process. The result is a powerful and efficient system capable of accurately identifying and localizing multiple objects in images.

(B) - MobileNetSSD

MobileNet SSD is an object detection model that merges the MobileNet architecture with the Single Shot Multibox Detector (SSD) framework. Designed for real-time object detection on mobile and embedded devices, it leverages MobileNet's computational efficiency through depth wise separable convolutions. The SSD framework enables simultaneous prediction of bounding boxes and class scores at multiple scales, offering versatility in detecting objects of various sizes. Known for its efficiency and speed, MobileNet SSD strikes a balance between accuracy and model size, making it well-suited for applications like augmented reality and image recognition on devices with limited computational resources. Additionally, it supports transfer learning, allowing pre-trained models to be fine-tuned for specific object detection tasks.

(C) – DARKNET & YOLO

Darknet is an open-source neural network framework renowned for its integration of the innovative You Only Look Once (YOLO) algorithm. Written in C and CUDA, Darknet prioritizes efficiency, making it well-suited for high-performance computing with fast training and inference speeds. The framework's modular architecture fosters customization and extension for various computer vision tasks beyond object detection, including image classification, segmentation, and generative adversarial networks (GANs). Darknet's lightweight design enables deployment on resource-constrained devices, offering versatility in applications such as embedded systems and IoT. Its compatibility with different neural network architectures makes it a flexible tool for experimenting with and implementing various deep learning models. Darknet's open-source nature has cultivated a robust community of developers and researchers, ensuring continuous improvement, bug fixes, and the incorporation of state-of-the-art techniques. Additionally, its command-line interface enhances user-friendliness, making it a valuable educational resource for learning about neural networks, deep learning, and computer vision through hands-on experimentation and exploration of diverse datasets.

After completing the training of datasets using models such as YOLO, ResNet SSD, or MobileNet SSD, the integration into a Raspberry Pi embedded system was successful. Utilizing the Pi's camera, real-time traffic sign detection output was achieved. To present the results in a user-friendly manner and demonstrate practical functionality, a graphical user interface was implemented using Tkinter software. Specifically, Tkinter was employed to display the autonomous speed reduction corresponding to the detected traffic signs.

This implementation highlights the hands-on application of computer vision models on edge devices, showcasing the Raspberry Pi's ability for on-the-fly object detection. The incorporation of a graphical interface through Tkinter not only improves the user experience but also allows for visualizing and potentially interacting with the system's autonomous speed reduction based on identified traffic signs. The project illustrates the seamless integration of machine learning, edge computing, and user interface design for creating a practical and interactive application in the domain of traffic management.

Deceleration = (Current speed – Speed on Sign Board)/Distance of speed sign

Acceleration = (Speed on Sign Board - Current speed)/Distance of speed sign

V – EXPERIEMENTS

We initiated the dataset training with the ResNet model, utilizing a batch size of 32 over 30 epochs. The initial accuracy using the SGD optimizer was 50%, which witnessed a notable boost to 68.9% upon switching to the Adam optimizer. Following this, the MobileNet model was employed with a slightly reduced batch size (30), resulting in a 65% accuracy with SGD and an impressive 80% with the Adam optimizer.

Subsequently, our exploration extended to ResNet SSD and MobileNet SSD models. Initially trained for 10 epochs, ResNet SSD achieved accuracies of 51% and 62%. Further training increased these figures to 58.1% and 70.4%, demonstrating the model's learning capability over additional epochs.

This detailed experimentation with different models and optimizers reflects a strategic approach to address accuracy constraints. The systematic exploration of ResNet, MobileNet, ResNet SSD, and MobileNet SSD, coupled with optimizer variations, provides a comprehensive understanding of their performance dynamics on the dataset. Fine-tuning hyperparameters and considering additional training strategies could further refine the models.

Now we used Darknet framework using Different YOLO packages, Given the critical role of object detection in autonomous vehicle systems, where precise identification and localization of entities in the vehicle's surroundings are paramount, YOLOv5 emerged as an optimal choice for its real-time processing capabilities. The model was implemented with the integration of the Darknet framework, contributing to its robust architecture.

Our YOLOv5 model, comprising a total of 152 layers, underwent training utilizing the Adam optimizer for 30 epochs. The training process, completed in 0.036 hours, yielded noteworthy results. The model's parameter count stood at 7,020,913, reflecting its complexity and capacity to discern intricate features within the dataset.

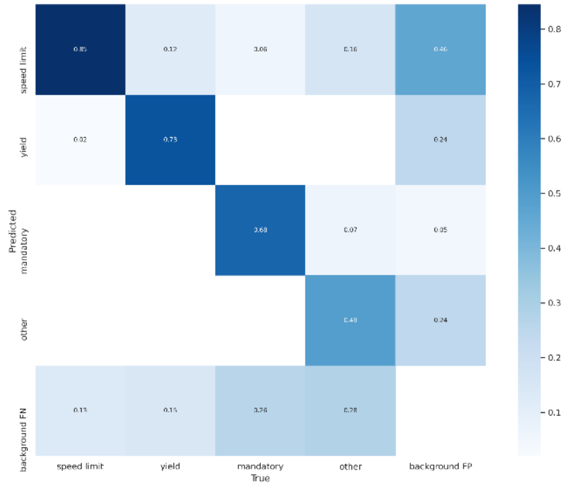

Fig 6: - Confusion matrix

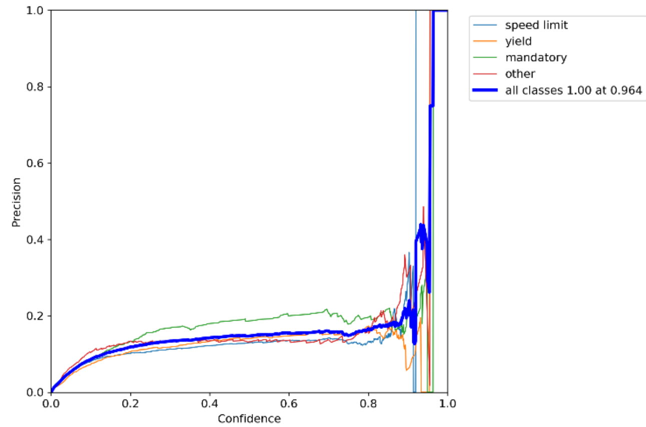

Fig 9: Precision graph

Upon completion of the training epochs, the evaluation yielded a mean Average Precision (mAP) value of 0.79, indicating the model's proficiency in accurately localizing objects. Additionally, the experiment generated a comprehensive set of visual outputs, including a confusion matrix graph and precision-recall graphs, providing a nuanced understanding of the model's performance across different confidence thresholds.

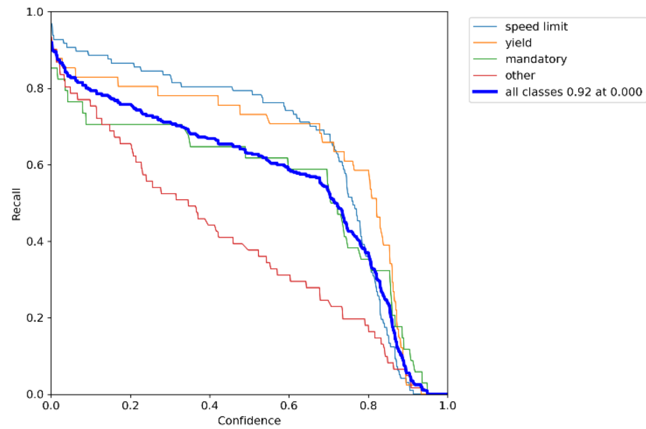

Fig 10: Recall graph

SR.NO	MODELS	mAP
1	RESNETSSD	0.72
2	MobileNetSSD	0.75
3	YOLOv3	0.7
4	YOLOv4	0.74
5	YOLOv5	0.79

Fig 7: - mAP of different models

```
30 epochs completed in 0.036 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.3MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.3MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 5/5 [00:01<00:00,  3.16it/s]
                   all        148        243      0.823      0.728       0.79      0.503
           speed limit        148        108      0.823      0.852      0.907      0.616
                 yield        148         42      0.794      0.857      0.915      0.608
             mandatory        148         32      0.778      0.531      0.577      0.305
                 other        148         61      0.898      0.672       0.76      0.481
Results saved to runs/train/exp
```

Fig 8: - Output of Code

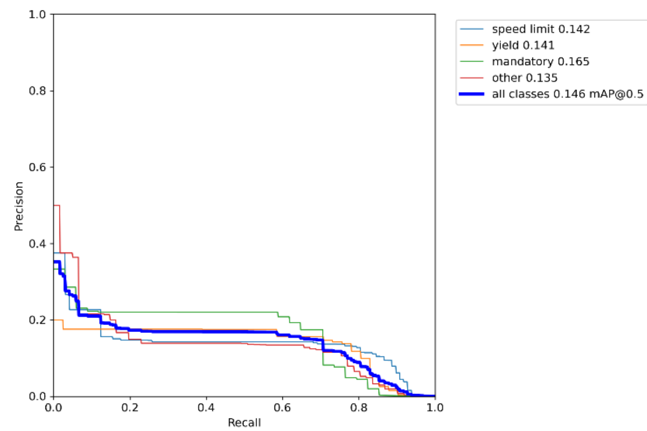

Fig 11: Precision vs Recall

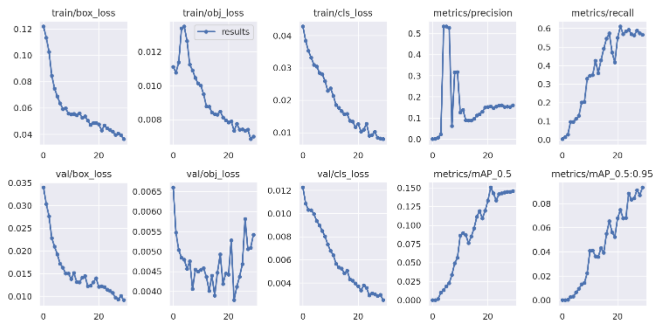

Fig 12: **-** Overall Results

VI – CHALLENGES

We encountered several challenges during the processing of the dataset. Manual data annotation posed difficulties as the code reflected errors. To address this issue, we opted to outsource the datasets. Tuning and handling imbalanced datasets presented additional challenges. Images captured in low-light conditions, covered with snow, or taken on rainy days resulted in vague representations. To mitigate this, we employed data augmentation techniques. When attempting to showcase the output using the Tkinter tool for speed control in autonomous vehicles, errors related to dimensions and accurate speed values emerged. We successfully resolved these issues. During the implementation of the code on the Raspberry Pi, initial attempts resulted in crashes. After making minor adjustments to the code, we were able to obtain the desired

VII – CONCLUSION

In conclusion, our research project, leveraging YOLOv5 within the Darknet framework, has made notable strides in addressing challenges within autonomous driving systems. However, it's crucial to acknowledge the specific limitations intrinsic to our model, notably its vulnerability during nighttime operations and suboptimal performance in adverse weather conditions. Despite these constraints, our methodology demonstrated promising outcomes in real-world scenarios, contributing valuable insights to the autonomous driving field. As we recognize and work to overcome these model-specific limitations, our research sets the stage for future refinements and underscores the potential of YOLOv5 and Darknet integration for more robust and efficient autonomous transportation solutions.

Future developments in the model aim to overcome challenges currently faced, particularly in enhancing object detection during nighttime and bad weather conditions. The model will be refined to exhibit improved performance in low-light scenarios and detecting even during bad weather conditions like raining and during snow, addressing limitations in recognizing objects in the dark. Additionally, the scope extends to transforming the current operational mode, which functions exclusively during cruise control. The objective is to extend the functionality to work seamlessly during manual driving as well. This expansion in capabilities aims to make the model more versatile and applicable across various driving scenarios. Furthermore, the future roadmap includes the implementation of additional safety measures, emphasizing a commitment to enhancing overall system reliability and ensuring the model's effectiveness in diverse driving environments. The ongoing efforts in research and development seek to elevate the model's capabilities, contributing to advancements in both object detection and overall driving safety.

VIII – REFERENCES

[1] Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). Focal Loss for Dense Object Detection. Proceedings of the IEEE International Conference on Computer Vision, 2017, 2980-2988. DOI: 10.1109/ICCV.2017.324

[2] Y. Zhang, P. Sun, Y. Yin, L. Lin, and X. Wang," Human-like Autonomous Vehicle Speed Control by Deep Reinforcement Learning with Double Q-Learning", IEEE, 2018, DOI: 10.1109/IVS.2018.8500630

[3] Álvaro Arcos-García, Juan A. Álvarez-García, Luis M. Soria-Morillo, "Evaluation of deep neural networks for traffic sign detection systems"

[4] Ashwani Kumar, Zuopeng Justin Zhang, Hongbo Lyu, "Object detection in real time based on improved single shot multi-box detector algorithm",2020, Article number: 204 (2020)

[5] Su Songzhi, Li Shaozi, Chen, ShuyuanCai, GuorongWu Yundong," Overview of Pedestrian Detection Technology", DOI: 10.3969/j.issn.0372-2112.2012.04.031

[6] Du Youtian, Chen Feng, Xu Wenli, Li Yongbin," A review of vision-based human motion recognition", DOI: 10.3321/j.issn:0372-2112.2007.01.017

[7] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi," You Only Look Once: Unified, Real-Time Object Detection", IEEE Xplore

[8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg," SSD: Single Shot MultiBox Detector", 2015

[9] Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, Shimin Hu," Traffic-Sign Detection and Classification in the Wild"

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[11] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, F. Mujica, A. Coates, and A. Y. Ng. An empirical evaluation of deep learning on highway driving. CoRR, abs/1504.01716, 2015.

[12] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In Neural Networks (IJCNN), The 2011 International Joint Conference on, pages 1453–1460. IEEE, 2011.

[13] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In Neural Networks (IJCNN), The 2011 International Joint Conference on, pages 2809–2813, July 2011.