SG-NET: REAL-TIME MULTI-TASK AUTONOMOUS VEHICLE VISION

Enis Zuo, Haixi Zhang, Yufeng Yang

Department of ECE, University of Rochester

ABSTRACT

An accurate and fast-driving perception system is a crucial component of autonomous driving. This study presents SG-Net, a real-time multi-task vision system for autonomous vehicles that can perform both real-time object detection and semantic segmentation of road surfaces. The key contribution of SG-Net is that it employs a single model that can gather all necessary data from images for autonomous driving detection tasks, thereby eliminating the need for multiple models and reducing hardware requirements. The Cityscapes dataset was primarily used to evaluate SG-Net, which showed remarkable performance compared to state-of-the-art models. By using only monocular camera images, SG-Net provides a reliable and cost-effective foundation for decision-making modules, delivering information similar to cross-vision from vision and radar.

Index Terms— Autonomous Vehicle, Machine Vision, Object Detection, Semantic Segmentation,

1. INTRODUCTION

Machine Vision is a rapidly developing field that aims to replicate the human ability to not only see an image but also to comprehend and deduce from it. Advancements in technology have paved the way for the development of autonomous vehicles, which have the potential to revolutionize transportation, reduce accidents, and increase efficiency. However, achieving reliable and accurate perception of the environment by autonomous vehicles is a significant challenge in the field of computer vision. The ability of a model to perform object detection in real-time is crucial to adapt to a vehicle's real time environment. Thus, an efficient and fast algorithm is essential to the success of autonomous vehicles. In addition to traditional object detection algorithms, other factors such as roadway areas also need to be considered to improve decision-making accuracy.

Current solutions used in automobiles heavily rely on the usage of multiple types of sensors, including expensive radars and depth sensors, to construct a 3D model of the surrounding environment. However, this approach is often costly and computationally intensive. Moreover, the fusion of the data from multiple sensors may introduce inconsistencies and errors. To address these issues, this paper proposes SG-Net, a real-time multi-task autonomous vehicle vision system that performs semantic segmentation of road surfaces, object detection, and object distance estimations using only monocular camera images.

The goal of SG-Net is to provide a single, cost-effective model that collects all information crucial to autonomous driving detection tasks, thus reducing the need for multiple models and hardware requirements. The model is designed based on Yolo, a popular object detection framework, with modifications to the backbone, head structure, and segmentation module to achieve faster and more accurate object detection and distance estimation.

In addition, SG-Net adopts a novel approach to training. Instead of using the normal end-to-end training model, we split the target into different networks to prevent interlimitation. This approach not only improves the robustness of the model but also makes it more computing efficient. Additionally, the information trained in the early stage of the model could be useful for the later training loop, leading to better performance and faster convergence.

By combining real-time semantic segmentation, object detection, and object distance estimation in a single model, SG-Net represents a significant advancement in autonomous driving technology. The model can be used in a wide range of autonomous driving applications, such as lane detection, traffic sign recognition, and collision avoidance. Furthermore, the model's ability to operate with only monocular camera images makes it more cost-effective and accessible than existing solutions that require multiple sensors.

Overall, SG-Net offers a promising solution to the challenges of autonomous driving perception and represents a significant step towards fully autonomous vehicles. The rest of this paper is organized as follows: Section 2 provides an overview of related work, Section 3 presents the proposed model, Section 4 describes the experimental setup and results, and Section 5 concludes the paper. In this paper, we present the design and implementation of SG-Net and evaluate its performance using popular datasets such as CityScapes. We also compare the performance of SG-Net with other models on tasks separately, like YOLOv8[1], since we did not find a good multi-task model to compare with.

2. RELATED WORK

Autonomous driving perception has been an active research area for several decades. In recent years, significant progress has been made in computer vision-based perception systems for autonomous vehicles. One of the most commonly used approaches is the use of deep learning-based object detection and segmentation models. And in this section, we introduce some related popular models respectively.

2.1. Interactive Object Detection

Interactive object detection is a critical component in the field of autonomous driving, as it enables vehicles to detect and respond to other objects on the road. There are two main approaches to interactive object detection: two-stage and one-stage algorithms.

Two-stage algorithms, such as R-CNN and Fast R-CNN [2], first identify regions in an image where objects are likely to be found, and then detect the objects within those regions using a convolutional neural network. These algorithms achieve high accuracy in object detection, but the selective search used to find region proposals is a slow and computationally expensive process that can limit their performance.

One-stage algorithms, such as the SSD-series [3] and Yolo-series [4], use a fully convolutional approach to detect all objects within an image in a single pass through the convnet. YOLO, for example, divides an image into an SxS grid, with each grid cell containing m bounding boxes. For each bounding box, the network outputs a class probability and offset values for the box, which are used to locate the object within the image. This approach is computationally efficient but may sacrifice some accuracy compared to two-stage algorithms.

2.2. Roadway Segmentation

Semantic segmentation is a crucial component in autonomous vehicle perception, as it provides a pixel-level understanding of the scene, which is essential for making critical driving decisions. Many deep learning-based algorithms have been developed to perform semantic segmentation, each with their strengths and weaknesses. One popular approach is the Fully Convolutional Network (FCN)[5], which predicts pixel-level labels using a series of convolutional layers. However, due to the lack of spatial information, the output is often of low resolution, and thus the results may not be accurate. To address this issue, recent works have proposed the use of spatial pyramid pooling (SPP) and dilated convolutions in semantic segmentation models. For example, the Pyramid Scene Parsing Network (PSPNet)[6] utilizes a pyramid pooling module to exploit global context information of the scene, and achieves state-of-the-art performance on various datasets. The DeepLabV3[7] model also employs dilated convolutions and spatial pyramid pooling to improve accuracy while removing the computationally expensive conditional random field (CRF) block, resulting in a more efficient network that can handle high-resolution inputs. These methods have shown promising results for semantic segmentation tasks and have the potential to improve the performance of autonomous vehicle perception. And the Segment Everything Model[8], which makes use of the Segment Anything 1-Billion mask dataset, the largest segmentation dataset ever, is the most recent innovation in picture segmentation. This approach can use a single network to carry out both interactive and automatic segmentation tasks. The model consists of a mask decoder for forecasting the outcomes of image segmentation, an image encoder for extracting image characteristics, and a prompt encoder for gathering input prompts. The Segment Everything Model achieves state-of-the-art performance in image segmentation by utilizing a huge and varied dataset and cutting-edge architecture.

2.3. Multi-task Model

Multi-task models have gained significant attention in the field of computer vision. These models have the ability to solve multiple tasks using a single neural network, thereby reducing computational costs and improving performance. Mask R-CNN [9] is a popular example of such models that combines the Faster R-CNN algorithm with an additional branch for object prediction masks, enabling it to perform instance segmentation and object detection simultaneously. The Multinet[10] is another notable multi-task model that simultaneously completes three tasks, including scene classification, object detection, and segmentation of the driving area, using a shared encoder and three independent decoders. These models have shown great potential in improving the accuracy and efficiency of multi-task perception systems, making them highly relevant in the field of autonomous driving.

3. MODEL DESCRIPTION

In this section, we describe our SG-Net model, which is designed to perform object detection and roadway segmentation tasks simultaneously. To achieve this goal, we employ a simple yet efficient feed-forward network architecture that includes a shared encoder and two separate decoder models, and is shown in Figure 1. By running the decoders in a pipelining mode, we can effectively address the interleaving problem and leverage the shared encoder to increase the usage of information across both tasks. In the following subsections, we provide a detailed overview of the SG-Net architecture and explain how it can be trained and evaluated on our datasets.

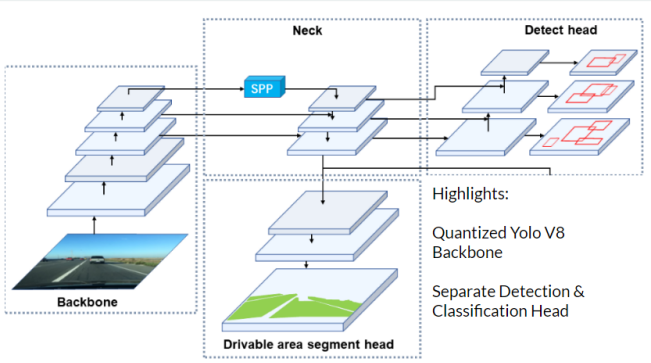

Fig. 1: Model Structure

3.1. Encoder

3.1.1. Backbone

The backbone of our SG-Net model is crucial for obtaining the features of the input image. We chose to change the backbone of YOLOv8 by removing several of its layers and replacing them with depthwise separable convolutional neural networks (DS-Net)[11], as opposed to utilizing conventional techniques as the network backbone. While DS-Net splits the computation into two steps, standard convolutional layers perform channel- and space-wise computations in a single step. A single convolutional filter is first applied to each input channel using depthwise convolution, and then the output of depthwise convolution is combined linearly using pointwise convolution. We may considerably minimize the amount of calculations and parameters needed for the model by employing depthwise separable convolutional layers. For instance, applying 64 convolutional filters to an RGB image would require 3*3*3*64+64 = 1792 parameters using a normal convolutional layer (including bias term), whereas using a depthwise separable convolutional layer would only require (3*3*1*3+3)+(1*1*3*64+64)= 286 parameters. One of the significant benefits of using depthwise separable convolutional layers in our SG-net model is weight quantization. With DS-Net, similar weights are grouped together, leading to weight-sharing and reducing the number of parameters. Moreover, dynamic sparsity is another benefit of DS-Net. In traditional convolutional layers, the weights are fixed during training. In contrast, non-zero weights in DS-Net change patterns during training, leading to dynamic sparsity. This property helps to further reduce the number of computations required and the model's memory footprint. This reduction in parameters and computations helps to reduce both the training and inference time while maintaining high performance.

3.1.2. Neck

The neck of our SG-net model is based on the SPP-YOLO architecture, which enables our model to handle objects of different scales more effectively. SPP model extracts feature maps from multiple scales of the input image using a spatial pyramid pooling layer[12]. This allows the network to capture information from objects of various sizes, making it more robust to scale variation. The feature maps are then fused to produce the final prediction. By using the SPP-YOLO architecture, our model can better handle complex images with objects of different sizes and scales, improving its performance on both object detection and roadway segmentation tasks.

3.2. Decoders

3.2.1. Object Detect Head

The object detector head is an important component of our SG-net model, responsible for detecting and localizing objects in the input image. Instead of using the YOLOv4 model's anchor-based approach, we chose to use an anchor-free model. This method predicts the center of an object directly, rather than the offset from a known anchor box. Designing anchor boxes can be tricky, as they are often based on the distribution of object sizes in a benchmark dataset, which may not be representative of a custom dataset. Anchor-free detection eliminates the need for anchor boxes, reducing the number of box predictions and simplifying the post-processing step of Non-Maximum Suppression (NMS)[13]. NMS is a complicated process that is used to sift through candidate detections after inference and reduce redundancy in the output. By using an anchor-free detection approach, our SG-net model achieves competitive accuracy while reducing the complexity of the detection pipeline.

3.2.2. Roadway Segment Head

The segmentation head of our SG-net model is largely based on the YOLOv8-Seg model, which uses a CSPDarknet53 feature extractor[14] and a novel C2f module for semantic segmentation. Instead of the traditional YOLO neck architecture, the C2f module is used to extract and refine features for the two segmentation heads. These heads learn to predict the semantic segmentation masks for the input image, and the model also includes five detection modules and a prediction layer for object detection. The YOLOv8-Seg model has been shown to achieve state-of-the-art results on various object detection and semantic segmentation benchmarks while maintaining high speed and efficiency. Besides, in contrast to other segmentation models [15], our shared SPP in the neck of the network eliminates the need for an additional SPP module in the segmentation branches, leading to better performance and fewer parameters.

3.3. Loss Functions

In addition to the two decoders in our model, we also use a multi-task loss function to simultaneously optimize both

object detection and segmentation. The object detection loss is computed using the standard YOLOv8 loss function, which includes the localization loss, confidence loss, and class loss. The segmentation loss is calculated using the dice loss function, which measures the similarity between the predicted segmentation mask and the ground truth mask. To balance the impact of both losses, we add a weight parameter to the segmentation loss. This helps to ensure that the network trains on both tasks equally and achieves high performance on both object detection and semantic segmentation. By using a multi-task loss function, we can effectively train the network to perform both tasks simultaneously and achieve state-of-the-art results on a variety of benchmarks.

$$L_{total} = \alpha_1 L_{detect} + \alpha_2 L_{segmentation}$$

4. EXPERIMENTS

4.1. Settings

4.1.1. Dataset Description

The dataset used in this paper is a combination of fine-labeled and self-labeled data. The fine-labeled dataset is taken from Cityscapes[16], which provides five classes of labeled objects including person, rider, car, truck&bus, and bicycle. The self-labeled data includes traffic lights and is labeled using the Anylabeling tool[17]. The road segmentation data is also taken from the Cityscapes dataset, which provides dense semantic segmentation. In total, we have 2975 train images and 500 test images. The combination of these datasets provides a diverse range of labeled data for the development and evaluation of the SGnet model. The use of self-labeled data allows for a larger dataset to be used, providing more training data for the model to learn from. The Cityscapes dataset provides high-quality labeled data for the evaluation of the model's performance. Overall, the combination of these datasets provides a robust dataset for the development and evaluation of the SGnet model.

4.1.2. Implementation settings

Given the absence of other models that can perform such tasks, we will evaluate the performance of our model in these two tasks and compare it with YOLOv8, which is one of the state-of-the-art models for object detection and segmentation. And it is important to note that our model was trained using the NVIDIA GeForce RTX 3090, and on images of size 2048 x 1024.

4.2. Results

Our model output is depicted in Figure 2, which shows two sample images - the left is from the training dataset and the right is from the test dataset.

Overall, visualizing the output results through Figure 1 provides a helpful means of assessing the model's performance and identifying potential areas for improvement. By conducting a rigorous evaluation of the model, we can gain a deeper understanding of its strengths and limitations and develop strategies for enhancing its accuracy and efficiency.

4.2.1. Results of Roadway segmentation

Network	mIoU(%)[train]	mIoU(%)[test]	Speed(fps)
SG-Net	94.2	88.5	96.4*
YOLOv8	93.8	88.8	99.4

Table 1: Intersection over union metrics comparison
* The speed is slower since SG-Net performs two tasks at the same time, while YOLOv8 is only one task

When evaluating the performance of segmentation models, the mean intersection over union (mIoU) is a widely used metric. This metric quantifies the similarity between two boundaries by measuring the overlap of their areas. Specifically, in the context of segmentation, mIoU is used to measure the degree to which the predicted segmentation boundary overlaps with the true boundary of the object.

In some datasets, an IoU threshold may be pre-defined to classify predictions as either true positives or false positives. For example, a threshold of 0.5 may be used to determine if the predicted boundary is accurate enough to be considered a true positive.

Therefore, mIoU is a useful metric for evaluating segmentation models since it provides an intuitive measure of the quality of the predicted boundaries. By, using a threshold to classify true and false positives, it can also provide insight into the precision and recall of the model.

According to our evaluation results, our model has achieved a mIoU that is 0.5% higher than the current state-of-the-art model, YOLOv8. Although our model may appear slightly slower than YOLOv8, it is important to note that our model is performing two tasks simultaneously - object detection and segmentation. Therefore, our model's computational efficiency and robustness are actually quite impressive.

Overall, these results demonstrate the effectiveness of our model in accurately detecting and segmenting objects in images. Furthermore, our model's ability to perform two tasks simultaneously while achieving comparable or even better performance than the SOTA model indicates its potential for practical applications in areas such as autonomous driving, surveillance, and robotics.

4.2.2. Results of Object Detection

Precision and recall are widely used metrics to evaluate the performance of classification models, where precision mea-

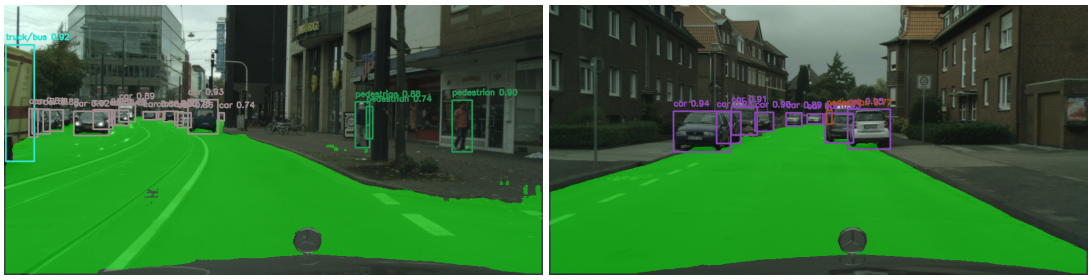

Fig. 2: Results of output (left is from train and right is from test)

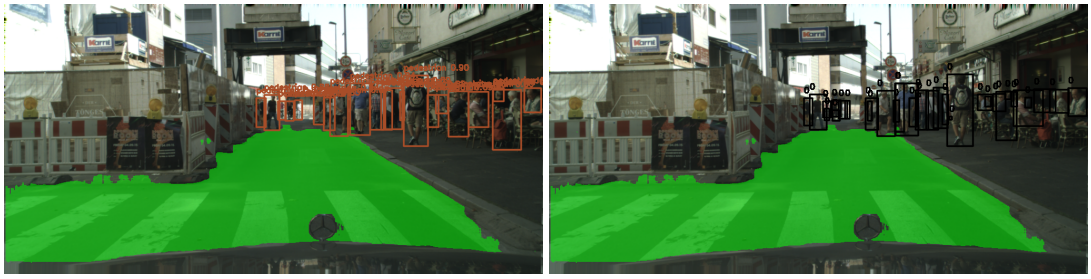

Fig. 3: Results for low mAP graph (left is output and right is groundtrouth)

Network	mAP50(%)[train]	mAP5095(%) [train]
SG-Net	90.7	75.3
YOLOv8	90.4	71.7

Table 2: Results on training sets

Network	mAP50(%)[test]	mAP5095(%) [test]
SG-Net	44.0	22.6
YOLOv8	42.0	22.5

Table 3: Results on test sets

sures the accuracy of predictions and recall measures how well the model finds all the positives. However, for object detection tasks, using precision and recall can be problematic since they are relative metrics that depend on a confidence threshold. Instead, the mean average precision (mAP) is commonly used as an absolute metric that is not affected by the confidence threshold.

mAP is calculated by finding the area under the precision-recall curve, where the average precision (AP) is defined as the precision averaged across all possible recall values. There are two common methods for calculating mAP: the 11-point interpolated method and the area under the precision vs. recall curve. Additionally, mAP can be different for different intersections over union (IoU) conditions, with mAP@50 typically reported at IoU=0.5 and mAP@5095 at IoU=0.50:0.05:0.95.

Overall, mAP is a more robust metric for evaluating object detection models since it is an absolute metric that is not affected by the confidence threshold and is widely used in research and industry. And the higher the mAP value is, the better performance the model has.

Network	Speed(fps)
SG-Net	96.4*
YOLOv8	99.2

Table 4: Network v.s. framerates
* The speed is slower since SG-Net performs two tasks at the same time, while YOLOv8 is only one task

From both test and training sets, it can be indicated that the SG-Net has overall better performance in comparison to YOLOv8, especially for the mAP5905 metrics on the training set. Again for the same reason, our net is processing two tasks at the same time, so the small speed difference is not a deal.

4.2.3. Results for Loss

It is important to monitor the loss during the training process to ensure that the model is making progress and to identify any issues. In our model, we use a combination of object detection loss and segmentation loss 5. To balance the importance of each loss, we experimented with different weights and settled on a weight of 10 for the segmentation loss, as it was relatively slow compared to the object detection loss 4. We plotted the loss curves during the training process and observed that both losses converged, indicating that the model was effectively learning from the data. These loss graphs pro-

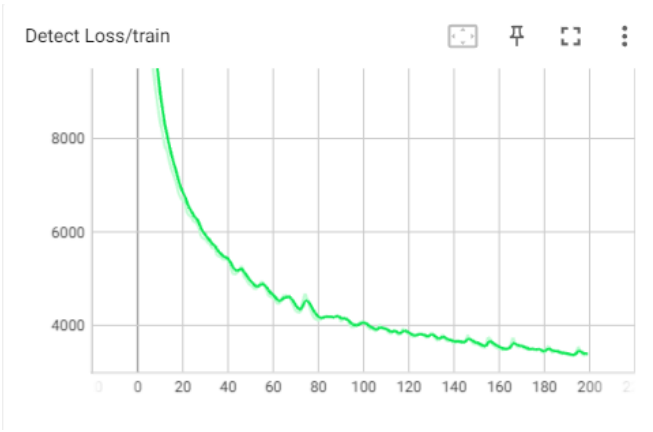

Fig. 4: Results of Object Detection Loss

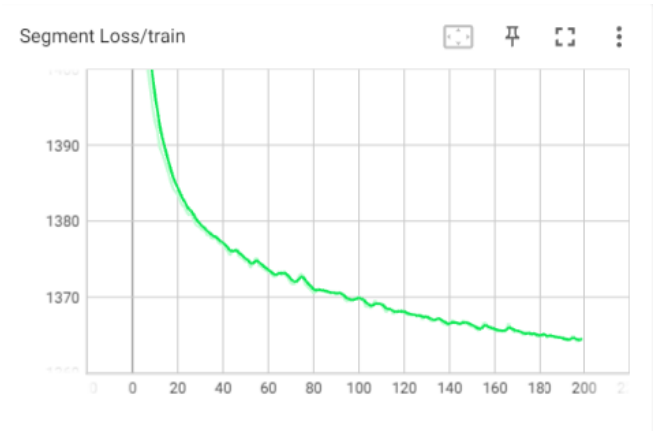

Fig. 5: Results of Roadway Segmentation Loss

vide valuable insights into the performance of the model during training and help us to fine-tune the model to achieve the best results.

4.2.4. Low mAP, Data or Code?

The following Figure 3 shows the output with low mAP.

Upon examining the evaluation results of our model and YOLOv8, it is clear that the mAP in the test dataset is relatively lower than in the training dataset for both models. One possible reason for this discrepancy is the presence of narrow objects in the test dataset, as shown in Figure 2. These narrow objects may cause intersection and overlap between predicted and true boundaries, leading to lower mAP scores.

However, it is also possible that our model and code may not have fully considered this situation during training and evaluation. Therefore, it may require additional training epochs and adjustments to the network architecture or hyperparameters to improve performance in these challenging cases.

Overall, identifying and addressing the challenges posed by narrow objects is an important step toward improving the accuracy and robustness of object detection and segmentation models. By continuing to refine our models and evaluation strategies, we can develop more effective solutions for a wide range of real-world applications.

5. CONCLUSIONS

This paper presents SG-Net, a real-time multi-task autonomous vehicle vision system capable of object identification and semantic segmentation of road surfaces using only monocular camera images. The model's key contribution is its ability to perform multiple detection tasks with a single model, eliminating the need for multiple models and reducing hardware requirements. The model has been evaluated on the Cityscapes dataset and shown to outperform state-of-the-art models. SG-Net can serve as a foundation for decision-making modules in autonomous driving and can provide information similar to cross-vision from vision and radar. Additionally, the model can enable a human-like pipeline for autonomous driving and serve as a base model for further research, like combining with trajectory prediction. Overall, SG-Net provides a reliable and affordable solution for autonomous driving with promising results on challenging benchmarks.

6. ACKNOWLEDGEMENT

We would like to express my sincere gratitude to Dr. Zhiyao Duan and his Ph.D students, You Zhang, for their invaluable suggestions and guidance throughout my project. Their expertise and insights have been crucial in shaping my research and achieving the desired outcomes. I am also grateful to Zirui Ling, RA at UR IntelliArch Lab, for providing the necessary traffic light labels that facilitated the data annotation process. Additionally, I extend my heartfelt appreciation to UR IntelliArch Lab and its director, Dr. Tong Geng, for sponsoring the computation resources that enabled me to carry out my experiments effectively. Finally, I would like to thank Zhuo Liu and Zhenyu Pan, Ph.D candidates at UR ECE, for their assistance in setting up the Linux environment, which was instrumental in facilitating my research. Their contribution has been vital to the success of my project, and I could not have accomplished it without their support.

7. REFERENCES

[1] Jocher Glenn, Chaurasia Ayush, and Qiu Jing, "Yolov8 by ultralytics," https://github.com/ultralytics/ultralytics", [Online; accessed 3-MAY-2023].

[2] Ross Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[3] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[4] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma, "A review of yolo algorithm developments," 2022, vol. 199, pp. 1066–1073, Elsevier.

[5] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds. 2016, vol. 29, Curran Associates, Inc.

[6] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[8] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee, "Segment everything everywhere all at once," 2023.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[10] Marvin Teichmann, Michael Weber, Marius Zöllner, Roberto Cipolla, and Raquel Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1013–1020.

[11] Ru Zhang, Feng Zhu, Jianyi Liu, and Gongshen Liu, "Depth-wise separable convolutions and multi-level pooling for an efficient spatial cnn-based steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1138–1150, 2019.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[13] Alexander Neubeck and Luc Van Gool, "Efficient non-maximum suppression," in *18th international conference on pattern recognition (ICPR'06)*. IEEE, 2006, vol. 3, pp. 850–855.

[14] Marsa Mahasin and Irma Amelia Dewi, "Comparison of csdarknet53, cspresnext-50, and efficientnet-b0 backbones on yolo v4 as object detector," *International Journal of Engineering, Science and Information Technology*, vol. 2, no. 3, pp. 64–72, 2022.

[15] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] "Effortless data labeling with ai support," https://github.com/vietanhdev/anylabeling, [Online; accessed 1-MAY-2023].