Investigation of Deep Learning Based Semantic Segmentation Models for Autonomous Vehicles

Xiaoyan Wang*, Huizong Li
School of Computer Science and Technology, Nanyang Normal University, Nanyang, Henan, 473061, China

***Abstract*—Semantic segmentation plays a pivotal role in enhancing the perception capabilities of autonomous vehicles and self-driving cars, enabling them to comprehend and navigate complex real-world environments. Numerous techniques have been developed to achieve semantic segmentation. Still, the paper emphasizes the effectiveness of deep learning approaches because they have demonstrated impressive capabilities in capturing intricate patterns and features from images, resulting in highly accurate segmentation results. Although various studies have been conducted in literature, there is needed for a careful investigation and analysis of the existing methods, especially in terms of two critical aspects: accuracy and inference time. To address this need for analysis and investigation, the research focuses on three widely-used deep learning architectures: ResNet, VGG, and MobileNet. By thoroughly evaluating these models based on accuracy and inference time, the study aims to identify the models that strike the best balance between precision and speed. The findings of this study highlight the most accurate and efficient models for semantic segmentation, aiding the development of reliable self-driving technology.**

Keywords—Semantic segmentation; autonomous vehicles; deep learning approaches; performance analysis; accuracy; inference time

I. Introduction

Semantic segmentation plays a pivotal role in the realm of computer vision, enabling machines to comprehend visual scenes by assigning each pixel of an image to a specific object category or class [1, 2]. This technique holds immense significance in a plethora of applications, including the navigation of autonomous vehicles [3]. The autonomous driving landscape, characterized by the emergence of self-driving cars, has transformed transportation paradigms [4]. Precise scene understanding through semantic segmentation is paramount in ensuring these vehicles' safe and efficient control in real-time scenarios [5, 6], enabling them to make informed decisions based on the interpretation of their surroundings from video feeds.

Autonomous vehicles, commonly referred to as self-driving cars, are reshaping the future of transportation [7]. Their ability to navigate complex environments autonomously relies on a myriad of technological advancements, and semantic segmentation stands as a linchpin among these. The process of accurately segmenting objects within a scene in real-time video feeds empowers self-driving cars to make split-second decisions [7-9], ensuring pedestrian safety, identifying lane boundaries, and interpreting traffic signals.

Existing methodologies in semantic segmentation for autonomous vehicles have made substantial strides. Deep learning-based approaches, in particular, have garnered significant attention due to their exceptional performance in complex tasks [10, 11]. This preference is attributed to their ability to automatically learn intricate features and patterns from vast datasets, ultimately leading to heightened accuracy [12]. Among the deep learning architectures, ResNet [13], VGG [14], and MobileNet [15] have emerged as frontrunners due to their efficiency in capturing nuanced spatial relationships and features within images [16]. However, despite these advancements, a need persists to identify the most effective and efficient deep learning-based method that strikes a balance between accuracy and inference time, thus optimizing the performance of semantic segmentation for autonomous vehicles.

The statement of the research problem is: How to achieve semantic segmentation for autonomous vehicles and self-driving cars using deep learning models that have high accuracy and low inference time. Correspondingly, the research questions are: What are the strengths and weaknesses of ResNet, VGG, and MobileNet architectures for semantic segmentation? How do these models compare in terms of accuracy and inference time on different datasets and scenarios? Which model(s) can provide the best balance between precision and speed for semantic segmentation?

In this study, we delve into the realm of DL-based models for semantic segmentation in autonomous vehicles, aiming to identify the most effective and efficient solutions. We examine three popular DL architectures: ResNet, VGG, and MobileNet, renowned for their contributions to computer vision tasks. Through a comprehensive analysis, we evaluate these models in terms of foreground accuracy, dice coefficient, and inference time, three crucial performance metrics in the context of autonomous driving systems.

Our findings reveal that certain DL models exhibit notable accuracy and efficiency in semantic segmentation for autonomous vehicles. By conducting an in-depth comparison between ResNet, VGG, and MobileNet architectures, we shed light on their respective strengths and weaknesses. Moreover, we identify the DL models that excel in terms of accuracy and inference time, providing valuable insights for practitioners and researchers in the field. The results of this study serve as a guide to selecting appropriate DL models for real-time semantic segmentation tasks in autonomous vehicles, ultimately contributing to the advancement and reliability of self-driving technologies. By rigorously evaluating these models' performance on video data, this study aims to contribute insights that advance the state-of-the-art in semantic segmentation for autonomous vehicles. This research endeavors to identify the most effective and efficient deep learning approach through meticulous experimentation and analysis, thereby fostering safer and more reliable autonomous driving systems.

II. Related Works

Ghosh et al. [17] introduced SegFast-V2, an approach to semantic image segmentation tailored for autonomous driving scenarios. Notably, the method prioritizes efficiency by utilizing fewer parameters within deep learning frameworks. With a focus on achieving accurate semantic segmentation, especially in the context of self-driving vehicles, SegFast-V2 presents a solution that balances computational efficiency and performance. The research contributes to advancing the field of autonomous driving by addressing the challenge of efficient and effective semantic segmentation, which is crucial for safe and reliable navigation in complex environments.

Colley et al. [18] investigated the impact of visualizing semantic segmentation in highly automated vehicles on trust, situation awareness, and cognitive load. By examining how providing visual cues of semantic segmentation affects drivers' perceptions and cognitive demands, the research aims to uncover insights into human-vehicle interaction dynamics. By analyzing the implications of semantic segmentation visualization on trust levels, understanding of the driving context, and mental workload, the paper enhances the design and implementation of automated driving systems to optimize driver experience, safety, and overall performance.

Nesti et al. [19] assessed the resilience of semantic segmentation methods employed in autonomous driving scenarios against real-world adversarial patch attacks. Focusing on the critical task of accurately segmenting objects in complex driving environments, the study investigates the vulnerability of these methods to deliberate perturbations introduced by adversarial patches. By subjecting various semantic segmentation models to these real-world attacks, the research endeavors to unravel the potential weaknesses and challenges of such vulnerabilities in ensuring safe and reliable autonomous driving systems. Through meticulous evaluation and analysis, the paper sheds light on the robustness of semantic segmentation techniques under adversarial conditions, offering valuable insights into enhancing the security and performance of self-driving vehicles. The author in Mo et al. [20] conducts a comprehensive review of the latest advancements in semantic segmentation technologies grounded in deep learning methodologies. By critically examining the current state-of-the-art approaches, the study aims to provide an in-depth understanding of the evolution and capabilities of deep learning-based semantic segmentation. Through the analysis of various models, architectures, and techniques, the paper contributes to the field's knowledge by outlining cutting-edge solutions that leverage deep learning for precise object delineation and scene understanding in diverse applications.

Dang et al. [21] presented a lightweight pixel-level semantic segmentation technique based on deep learning for the purpose of detecting and analyzing sewer defects. By leveraging deep learning methods, the approach offers an efficient solution for identifying and classifying sewer system issues through pixel-level segmentation. The study's focus on lightweight architecture signifies a commitment to computational efficiency while maintaining accurate defect identification. This research contributes to the field of sewer infrastructure maintenance by offering a streamlined approach that employs deep learning for detailed and effective defect analysis, enhancing the overall assessment and management of sewer systems.

As results, there are many existing methods for semantic segmentation, but they need to be carefully investigated and analyzed, especially in terms of two critical aspects: accuracy and inference time. Accuracy is the measure of how well the model can correctly segment the image and match the ground truth labels. Inference time is the measure of how fast the model can process the image and produce the segmentation output. These two aspects are important because they affect the performance and safety of the autonomous vehicles and self-driving cars. A model that has high accuracy can provide more reliable and detailed information for the vehicle, while a model that has low inference time can respond more quickly and adapt to changing situations. Therefore, the research paper wants to find the best balance between accuracy and inference time for semantic segmentation.

III. Material and Method

A. Dataset Overview

The Cambridge-driving Labeled Video Database, commonly known as CamVid, is a comprehensive and meticulously annotated dataset designed to advance the field of computer vision, particularly in the context of autonomous driving and scene understanding. The CamVid stands as a vital resource in the realm of computer vision with its diverse and meticulously labeled video sequences. CamVid features a diverse collection of high-resolution video sequences captured from a moving vehicle navigating through urban and suburban environments. These videos encompass a wide range of real-world driving scenarios, presenting challenges such as varying lighting conditions, dynamic traffic, and intricate road layouts. One of the distinguishing aspects of CamVid is its extensive labeling. Each frame of the dataset is meticulously annotated with pixel-level semantic segmentation labels. This means that every pixel in the video frames is categorized, providing a detailed understanding of the objects and structures present in the scenes. Such detailed annotations enable the training and evaluation of advanced machine-learning models for tasks like object detection, semantic segmentation, and instance segmentation. CamVid's applications extend beyond autonomous driving research. The dataset's rich annotations make it highly suitable for projects related to urban scene understanding, environmental monitoring, and general semantic segmentation challenges. The dataset consists of a series of videos, each accompanied by semantic labels that categorize object classes. These labels are accompanied by additional metadata. The database includes accurate reference labels that link every individual pixel to one of 32 predefined semantic categories. Fig. 1 [22] demonstrates the semantic classes of the CamVid dataset.

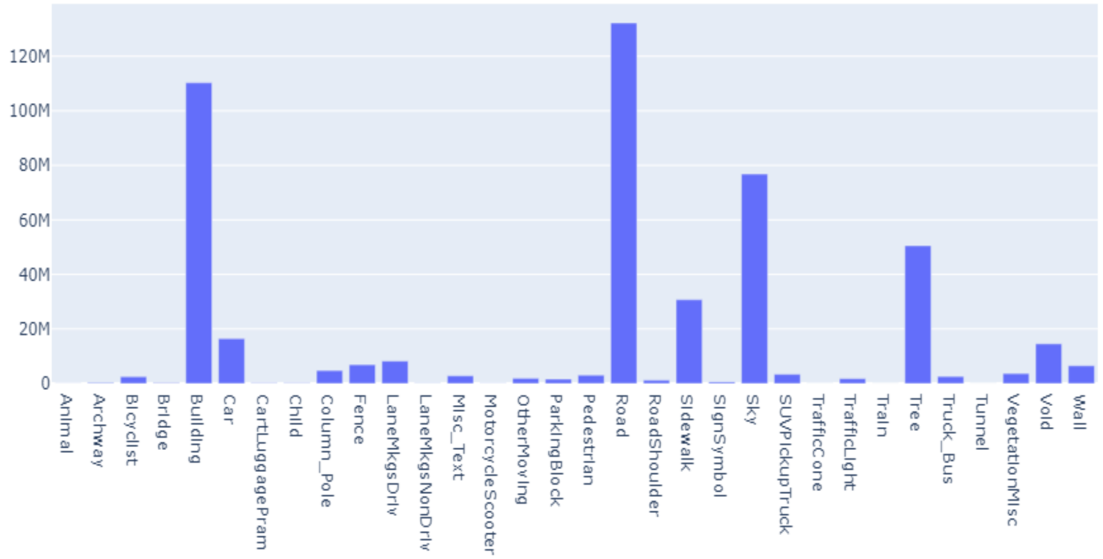

Fig. 1. Semantic classes of the camvid dataset.

B. *Model Learning*

The proposed model in this study is designed to acquire detailed annotation for each individual pixel within a scene recorded from the perspective of an autonomous agent. The primary task of this model is to classify and isolate every pixel in the given scene into one of 32 specific categories. These categories include items such as roads, pedestrians, sidewalks, and cars, as showcased in the animated image of our product. This enables interaction with any particular image.

The main objective is to understand and interpret the scene with exceptional precision. This understanding is achieved by categorizing each pixel into specific classes, which represent different objects or entities within the scene. For instance, a road, a pedestrian, a sidewalk, a car, and more – all of these are examples of classes that the model identifies. Imagine a picture of a street: the road, the people walking on the sidewalk, the parked cars, and other elements are contained. Our model performs something similar but for each and every pixel in the scene. It determines if a pixel belongs to the road, the sidewalk, a person, a car, or one of the other predefined categories – a total of 32 categories.

C. *Backbones*

For our initial set of experiments, we opted to employ a straightforward architecture that draws inspiration from the UNet model. This architecture incorporates backbones like ResNet50, VGG19, and MobileNetV2. Despite its simplicity in terms of implementation, this architecture has proven to be remarkably robust in terms of its performance. In other words, it strikes a balance between being relatively easy to create and yielding impressive results in various tasks.

1) ResNet50 Backbone: The UNet architecture is a convolutional neural network (CNN) design that excels in image segmentation tasks. It consists of an encoding path that gradually reduces spatial resolution while capturing features and a decoding path that restores the resolution while refining segmentation maps. In this baseline, we enhance the UNet with a ResNet50 backbone, which is a deep residual network known for its excellent performance in various computer vision tasks. ResNet50 incorporates skip connections to mitigate vanishing gradient issues during training. As shown in Fig. 2 [23], the architecture of ResNet50 is depicted.

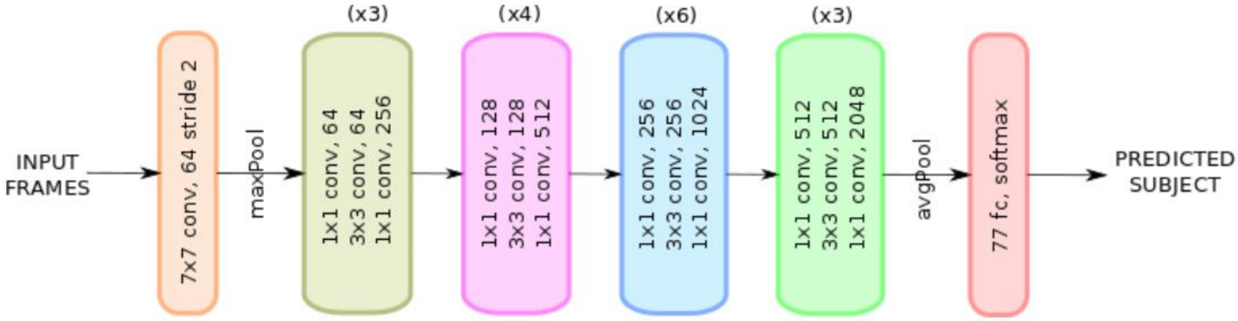

Fig. 2. ReseNet50 architecture [23].

As shown in Fig. 2, the ResNet50 is a specific variant of the ResNet architecture, characterized by its depth of 50 layers. It builds upon the original ResNet's innovation of residual connections, enhancing its capacity to capture complex patterns and features from images. By incorporating a series of residual blocks, ResNet50 enables the efficient training of deeper neural networks while mitigating issues related to vanishing gradients. This architecture has proven highly effective in various computer vision tasks, such as image recognition and segmentation. In the context of enhancing the perception capabilities of autonomous vehicles, ResNet50's depth and feature-extraction prowess contribute to accurate and detailed semantic segmentation, aiding in the vehicles' understanding and navigation of intricate real-world environments.

2) VGG19 backbone: Similar to the previous architecture, this baseline employs a UNet structure but integrates a VGG19 backbone. VGG19 is a deep CNN architecture known for its simplicity and effectiveness. It consists of multiple convolutional layers followed by max-pooling, and it captures progressively complex features through its layers. This backbone enhances the UNet's feature extraction capabilities, contributing to better segmentation performance. Fig. 3 demonstrates the architecture of VGG19.

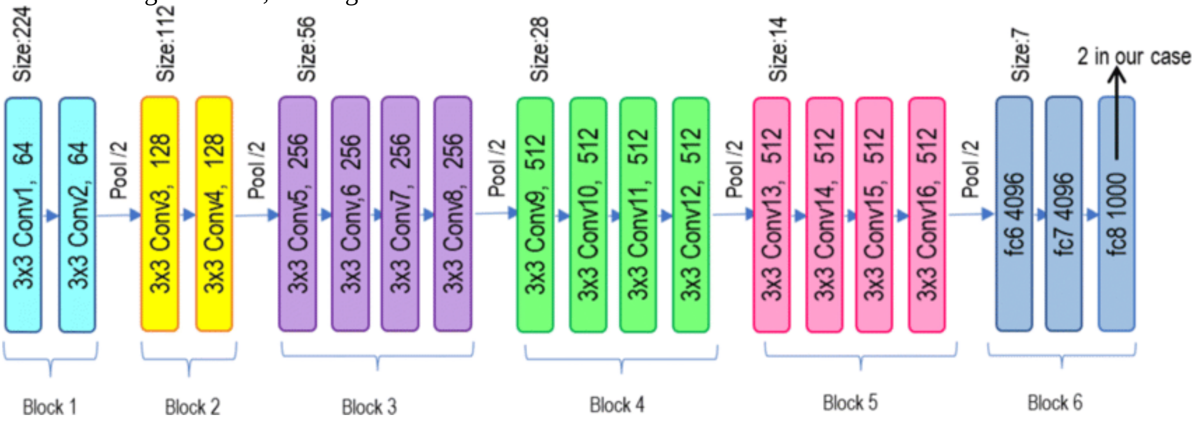

Fig. 3. VGG19 architecture [24].

As shown in Fig. 3, VGG19 is a convolutional neural network architecture renowned for its simplicity and effectiveness in image recognition tasks. With 19 layers, it follows a straightforward design principle of stacking multiple 3x3 convolutional layers, followed by max-pooling layers for down-sampling. This repetitive structure results in a deep network capable of capturing intricate features at different levels of abstraction. VGG19's uniform architecture makes it easy to understand and implement, contributing to its popularity. In the context of enhancing the perception capabilities of autonomous vehicles, VGG19's depth and feature-extraction capabilities play a crucial role in semantic segmentation, enabling the vehicles to accurately perceive and navigate complex real-world scenarios.

3) MobileNetV2 Backbone: The UNet design combined with a MobileNetV2 backbone represents a lightweight yet powerful configuration. MobileNetV2 is optimized for efficiency and speed, making it suitable for real-time applications on resource-constrained devices. It utilizes depthwise separable convolutions to reduce computational complexity while preserving accuracy. Fig. 4 illustrates the architecture of MobileNetV2

The architecture's core component is depth-wise separable convolutions. In these convolutions, the spatial information is decoupled from the channel-wise information, reducing the computational load. Each convolution is divided into a depth-wise convolution, which applies a single convolutional filter to each input channel, followed by a point-wise convolution that merges the outputs into the desired number of output channels. MobileNetV2 also employs skip connections to retain important features, facilitating the flow of gradients during training. These innovative design choices collectively result in a lightweight architecture capable of achieving impressive accuracy on tasks like image classification and semantic segmentation.

D. Hyperparameter Tuning

To enhance the efficacy of our baseline model, a dual focus on both optimal model selection and hyperparameter tuning becomes imperative. The crux lies in identifying not only the most suitable model architecture but also the optimal configuration of hyperparameters for training. To achieve this, we utilize a Bayesian hyperparameter search methodology, a sophisticated technique aimed at systematically exploring the hyperparameter space to unearth the combination that yields the most favorable results.

The essence of this method revolves around minimizing the model's loss function when evaluated against a dedicated validation dataset. By leveraging a Bayesian approach, we dynamically adapt the search process based on previous iterations, progressively honing in on the most promising areas of the hyperparameter space. This method is especially effective in mitigating the challenges posed by high-dimensional and complex search spaces. Ultimately, the outcome of this meticulous hyperparameter search is a refined model configuration that not only aligns with the chosen architecture but also significantly bolsters the model's performance, setting the stage for more accurate and robust predictions.

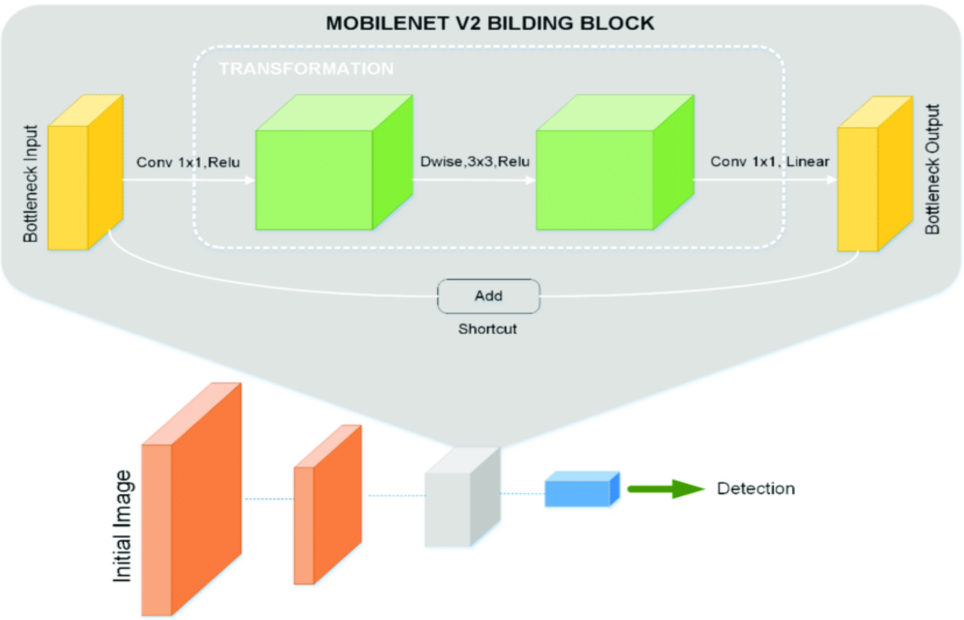

Fig. 4. MobileNetV2 architecture [25].

IV. Experimental Results and Performance Analysis

The forthcoming section delves into the outcomes of our experiments, shedding light on the results obtained through rigorous testing and evaluation. It's important to underscore that for reasons of safety paramountcy, we have prioritized specific classes during the training process. These priority classes encompass entities that play pivotal roles in ensuring safety within various scenarios. The prioritized classes, due to their pronounced safety implications, encompass Pedestrians, Bicyclists, Children, Cars, Heavy Vehicles, and Traffic lights. These categories encapsulate elements that are central to the smooth functioning of urban environments and the safety of both pedestrians and drivers. By focusing on these classes during training, we aim to equip our model with the capability to distinctly recognize and respond to these critical entities.

When evaluating the performance of models in semantic segmentation tasks, two essential metrics that provide valuable insights are Foreground accuracy and the dice coefficient. Foreground accuracy measures the precision with which a model correctly classifies the foreground objects of interest, which is particularly important in scenarios where specific classes carry more significance.

A. Foreground Accuracy

Foreground accuracy, in the context of semantic segmentation, is a metric used to gauge the accuracy of a model's predictions specifically concerning the foreground objects or classes of interest. Unlike overall accuracy, which considers all classes equally, foreground accuracy focuses solely on how well the model correctly identifies and classifies the relevant objects, ignoring the background and other unimportant classes. This metric provides a more insightful evaluation of a model's performance in tasks where certain classes are of greater significance than others, such as object detection or scene segmentation. It is computed by dividing the number of correctly classified foreground pixels by the total number of foreground pixels and can be represented as:

$$Forground\ Accuracy = {True\ Positive\ (TP) + False\ Negative (FN)} \Big/ {True\ Positive\ (TP)}$$

B. Dice Metric

The dice coefficient, also known as the F1 score, offers a comprehensive assessment of segmentation accuracy by considering false positives and false negatives. It quantifies the overlap between the predicted segmentation and the ground truth, producing a value between 0 and 1, where 1 indicates perfect alignment. Fig. 5, 6 and 7 show dice metric for the methods. The dice metric is calculated as follows:

$$Dice = {2 * True\ Positive\ (TP)} \Big/ {2 * True\ Positive\ (TP) + False\ Positive (FP) + False\ Negative (FN)}$$

C. Backbone Experiments

In this section, we delve into the backbone experiments, wherein the focus lies on the fundamental architectural components of our models. These backbone models excel at assimilating contextual information from expansive image regions. This is achieved by adeptly pooling features through a variety of window sizes and seamlessly integrating them using both residual connections and adaptable weights. Our investigation involves subjecting the baseline models to thorough experimentation, including an exploration of different loss functions, to comprehensively understand their performance and capabilities. We present some experiments corresponding to the baselines. Table I presents the experiment name associated with each backbone.

D. Experiments with Hyperparameters

As discussed in Section 5.6, in the process of assessing the effectiveness of models such as ResNet50, VGG19, and MobileNetV2, we harness the capabilities of the Sweep tool. This utility streamlines the execution of a Bayesian hyperparameter search strategy, which is employed to minimize the model's loss on the validation dataset. Sweeps significantly simplify our ability to conduct various experiments while utilizing this search method. The results of foreground accuracy and dice are shown in Fig. 8, 9, 10 and 11.

TABLE I. EXPERIMENT NAME FOR THE BACKBONES

Experiment name	Backbone name
baseline-train-1	ResNet50
baseline-train-2	VGG19
baseline-train-3	MobileNetV2

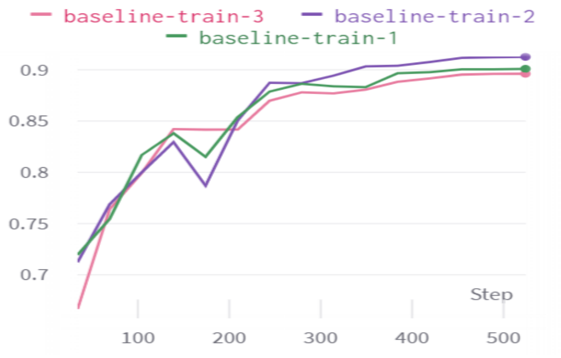

Fig. 5. Foreground accuracy for baseline experiments.

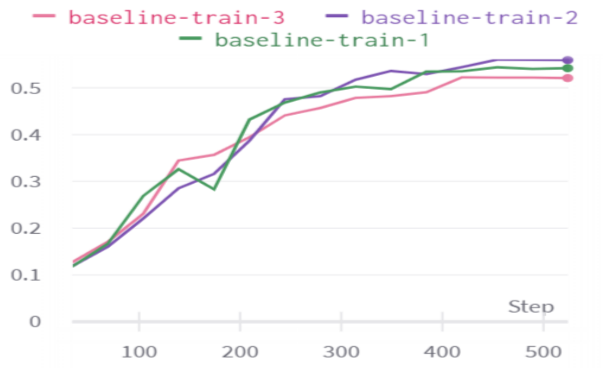

Fig. 6. Dice score for baseline experiments.

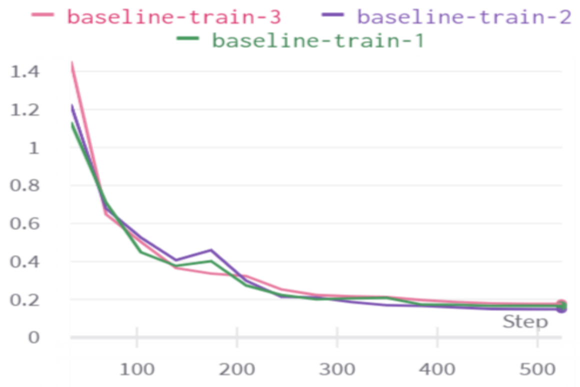

Fig. 7. Validation loss for baseline experiments.

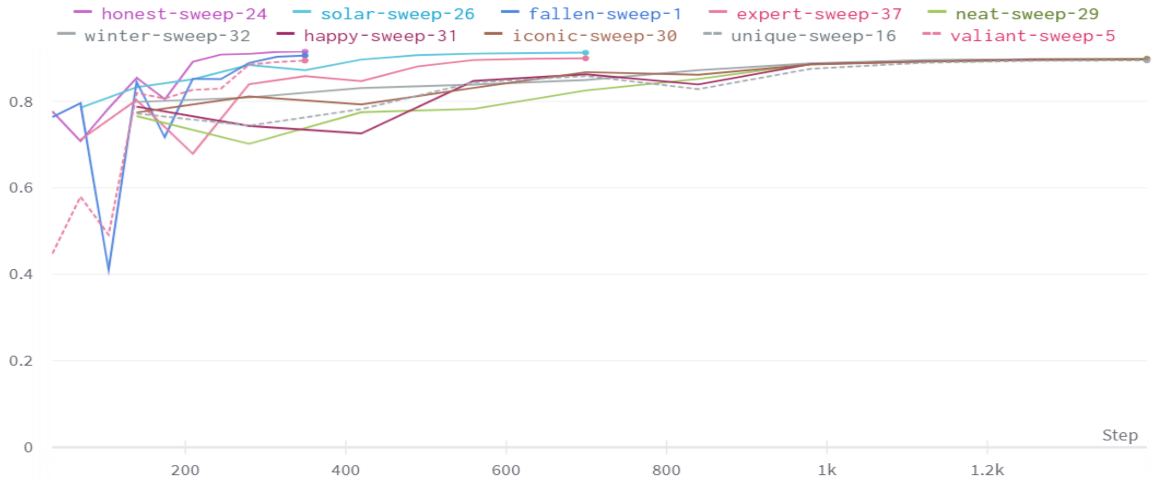

Fig. 8. Foreground accuracy.

Fig. 8 depicts foreground accuracy for various hyper-parameters tuning using Sweep. The provided chart illustrates the Foreground accuracy achieved across ten distinct experiments, offering a comparative analysis of various fine-tuned models. Among these experiments, the recorded accuracy values reveal a hierarchy of performance. Notably, the experiment named "honest-sweep-24" attains the highest accuracy, followed by "solar-sweep-26," "valiant-sweep-5," "solar-sweep-26" once again, and "expert-sweep-37." Conversely, the models "neat-sweep-29," "winter-sweep-32," "happy-sweep-31," "iconic-sweep-30," and "unique-sweep-16" exhibit the lowest accuracy values.

Starting with the highest accuracy achieved in the experiment labeled "honest-sweep-24," it showcases the model's exceptional ability to accurately classify foreground objects. The meticulously tuned parameters of this model contribute to its precision in segmenting relevant classes within the image. This high accuracy score signifies that the model successfully distinguishes and labels the target objects, a crucial feat in tasks like object recognition or scene understanding. The reliability of the "honest-sweep-24" experiment's outcome implies that its fine-tuning process effectively optimized its performance, rendering it a formidable contender in semantic segmentation tasks.

On the other hand, Fig. 9 presents the performances of the models with various backbones in terms of mean foreground accuracy. The backbones involve resnet34, resnet50, Resnet18, vgg19, mobilenetv2_100, mobilenetv3_large_100, mobilenetv3_small_050. The mean foreground accuracy indicates how well the model is performing in correctly classifying instances belonging to the class of interest, often in the context of object recognition or segmentation.

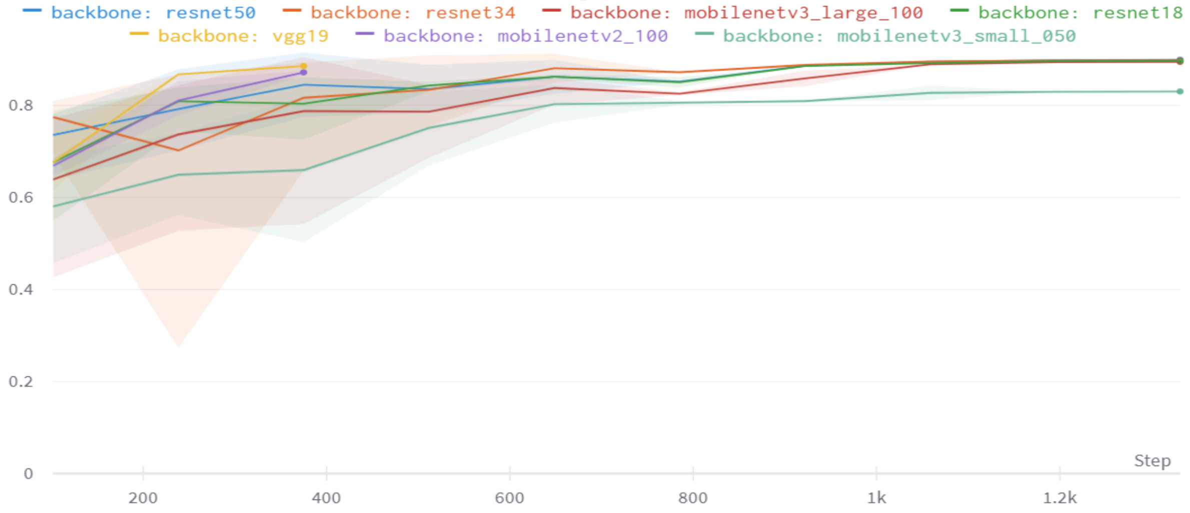

Fig. 9. Mean foreground accuracy for different backbones.

As shown in Fig. 9, among the listed backbone models, "Resnet18" stands out as the best performer, with an accuracy of 89.67%. Resnet18 is a variant of the Residual Network (ResNet) architecture, designed with 18 layers. This architecture utilizes residual blocks, allowing it to efficiently train deep neural networks by mitigating the vanishing gradient problem. Resnet18's success can be attributed to several factors. Firstly, its moderate depth strikes a balance between model complexity and capacity, preventing overfitting while still capturing intricate features in the data. Secondly, Resnet18's residual connections enable efficient gradient flow during training, fostering better convergence and feature representation. Thirdly, Resnet18's design incorporates skip connections, which allow information to bypass certain layers, further enhancing its ability to capture relevant features.

Comparatively, other backbones might struggle due to either excessive complexity leading to overfitting (as with deeper architectures) or limited depth hindering feature extraction (as with shallower architectures). Resnet18 strikes a favorable balance, resulting in its superior mean foreground accuracy. Its intermediate depth, residual connections, and skip connections collectively contribute to achieving a strong balance between capacity and generalization, making Resnet18 a top performer in the comparison.

Finally, the mean dice score presents different backbones, as shown in Fig. 8. It quantifies the similarity between predicted and ground truth segmented regions by measuring the overlap of pixels. This metric's significance lies in its ability to assess the model's capability to accurately delineate object boundaries and capture fine-grained details in complex scenes, providing a comprehensive measure of segmentation quality and performance.

As demonstrated in Fig. 10 presents the mean dice score values collected for different backbone architectures, including "resnet34," "resnet50," "vgg19," "mobilenetv2_100," and "mobilenetv3_large_100." These scores reflect the performance of each backbone on specific tasks or datasets. Notably, "resnet34" emerges as the top-performing architecture with the highest dice scores across multiple columns, followed closely by "resnet50." Both these architectures consistently exhibit superior performance compared to others like "vgg19," "mobilenetv2_100," and "mobilenetv3_large_100." The provided scores reflect the efficacy of these backbones in tackling the given tasks, with "resnet34" standing out as a particularly strong model.

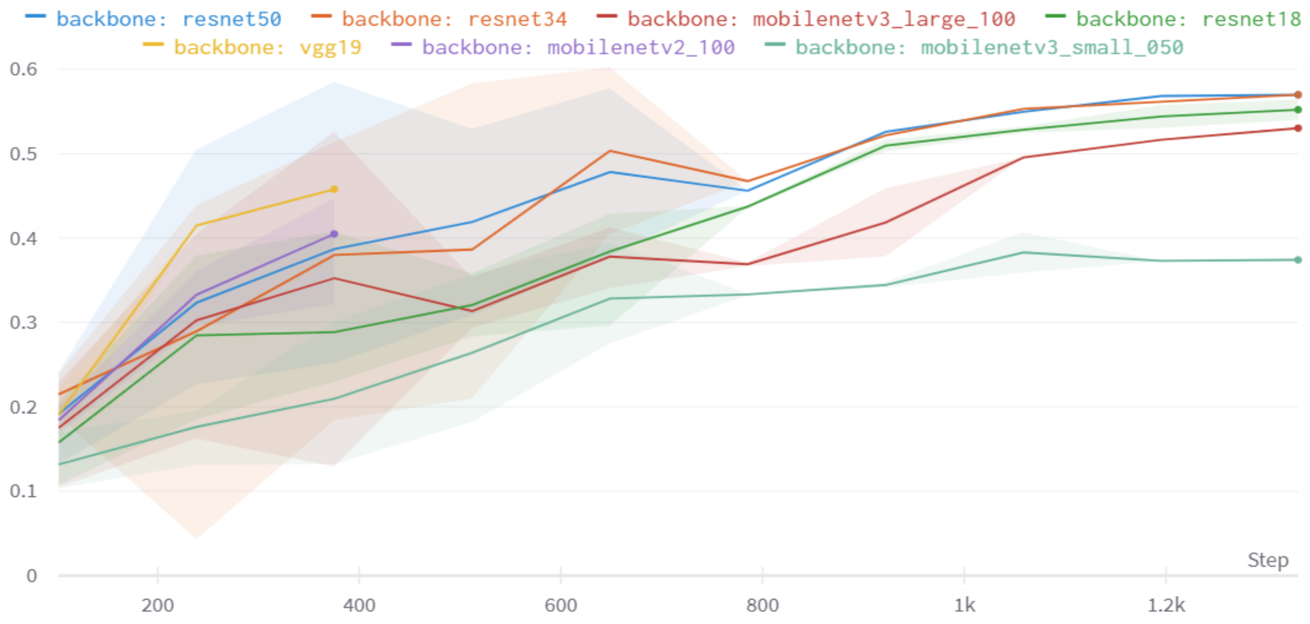

Fig. 10. Mean dice score for the backbones.

E. Inference Time

Inference time refers to the amount of time a model takes to process an input and produce an output prediction. In the context of semantic segmentation models, it measures how quickly a model can analyze an image and generate pixel-wise segmentation results.

The inference times were collected for various backbone models, including "resnet34," "resnet50," "Resnet18," "vgg19," "mobilenetv2_100," "mobilenetv3_large_100," and "mobilenetv3_small_050." Notably, the "Resnet18" model achieved the lowest inference time, while "Vgg19" had the highest.

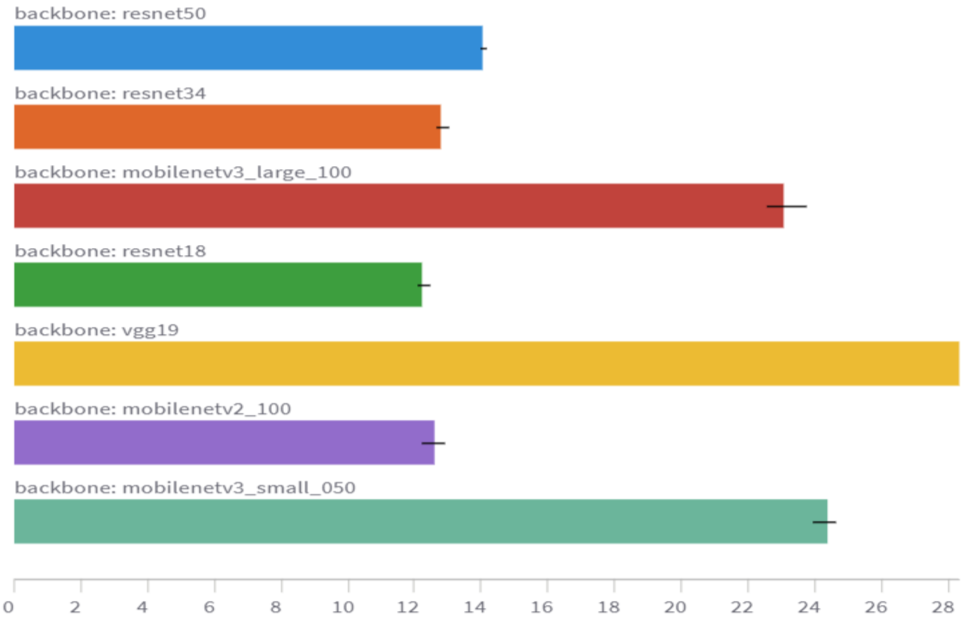

Fig. 11. Inference time of different backbones.

As illustrated in Fig. 11, the superior model in terms of inference time appears to be "Resnet18," which exhibits the lowest processing time among the listed models. This faster inference time can be attributed to Resnet architectural design, which balances depth and complexity, enabling efficient feature extraction while minimizing computational overhead. In contrast, "Vgg19," while offering strong performance, likely incurs higher inference times due to its greater depth and more complex architecture. Therefore, "Resnet18" emerges as the superior choice for applications that prioritize faster semantic segmentation inference times.

V. Discussion

The analysis conducted yields critical insights that significantly inform the optimization of the training process and the selection of backbone architectures for a semantic segmentation model. Firstly, it's evident that employing lower learning rates and weight decay parameters leads to improved foreground accuracy and dice scores. This underscores the necessity of precise parameter tuning to achieve superior segmentation results.

Secondly, the study identifies the batch size and image resize factor as key factors with strong positive correlations to the evaluation metrics. This highlights the pivotal role these factors play in shaping model performance, emphasizing their potential for enhancing accuracy and dice scores.

Additionally, caution is advised against the utilization of VGG-based backbones for the final model. The findings suggest that these architectures are susceptible to vanishing gradients, which can impede gradient propagation during training and hinder optimal model performance.

Ultimately, the analysis underscores the superior performance of ResNet backbones across various metrics. ResNet34 and ResNet50 emerge as optimal choices for the final model due to their impressive performance and quicker inference times compared to other architectures. These insights provide actionable recommendations for refining model training, selecting appropriate backbones, and ultimately improving the efficiency and accuracy of semantic segmentation models.

VI. Conclusion

This study extensively explores various deep learning models for semantic segmentation, particularly ResNet, VGG, and MobileNet architectures, aiming to determine the most effective and efficient approach in terms of accuracy and inference time. Through thorough analysis of real-world video data, the research strives to advance semantic segmentation for autonomous vehicles, enhancing their safety and reliability. The investigation's outcomes offer crucial insights into optimizing training processes and selecting backbone architectures, with lower learning rates and weight decay parameters enhancing accuracy, while the batch size and image resize factor positively influence model performance. Caution against using VGG-based backbones due to vanishing gradients is noted, favoring ResNet34 and ResNet50 for their strong metrics and quicker inference times. These findings provide actionable guidelines for refining model training and selecting suitable backbones to enhance the efficiency and precision of semantic segmentation models. For future studies, firstly, researchers could explore hybrid architectures that combine the strengths of ResNet, VGG, and MobileNet for semantic segmentation. This approach seeks to harness the unique features of each architecture to create novel solutions that offer a balance between accuracy, efficiency, and robustness. Secondly, a promising avenue involves enhancing the adversarial robustness of semantic segmentation models. This entails investigating techniques to counter real-world adversarial attacks, particularly in the context of self-driving vehicles. By developing defense mechanisms against such attacks, researchers could contribute to improving the reliability and safety of autonomous driving systems.

Acknowledgment

This work was supported in part by the Natural Science Foundation of Anhui Province of China under Grant 1808085MG221.

References

[1] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "A review of semantic segmentation using deep neural networks," International journal of multimedia information retrieval, vol. 7, pp. 87-93, 2018.

[2] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," Neurocomputing, vol. 406, pp. 302-321, 2020.

[3] B. Chen, C. Gong, and J. Yang, "Importance-aware semantic segmentation for autonomous vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 1, pp. 137-148, 2018.

[4] Q. Sellat, S. Bisoy, R. Priyadarshini, A. Vidyarthi, S. Kautish, and R. K. Barik, "Intelligent semantic segmentation for self-driving vehicles using deep learning," Computational Intelligence and Neuroscience, vol. 2022, 2022.

[5] M. Ivanovs, K. Ozols, A. Dobrajs, and R. Kadikis, "Improving semantic segmentation of urban scenes for self-driving cars with synthetic images," Sensors, vol. 22, no. 6, p. 2252, 2022.

[6] M. C. ANG, A. AGHAMOHAMMADI, K. W. NG, E. SUNDARARAJAN, M. MOGHARREBI, and T. L. LIM, "MULTI-CORE FRAMEWORKS INVESTIGATION ON A REAL-TIME OBJECT TRACKING APPLICATION," Journal of Theoretical & Applied Information Technology, vol. 70, no. 1, 2014.

[7] A. Moorthy, B. Sivashanmugam, R. Sriram, and M. Swathi, "Real Time Image and Video Semantic Segmentation For Self-Driving Cars," Journal of Survey in Fisheries Sciences, vol. 10, no. 2S, pp. 3208-3216, 2023.

[8] Q. H. Che, D. P. Nguyen, M. Q. Pham, and D. K. Lam, "TwinLiteNet: An Efficient and Lightweight Model for Driveable Area and Lane Segmentation in Self-Driving Cars," arXiv preprint arXiv:2307.10705, 2023.

[9] M. Ang, E. Sundararajan, K. Ng, A. Aghamohammadi, and T. Lim, "Investigation of Threading Building Blocks Framework on Real Time Visual Object Tracking Algorithm," Applied Mechanics and Materials, vol. 666, pp. 240-244, 2014.

[10] Q. Sellat, S. K. Bisoy, and R. Priyadarshini, "Semantic segmentation for self-driving cars using deep learning: a survey," in Cognitive Big Data Intelligence with a Metaheuristic Approach: Elsevier, 2022, pp. 211-238.

[11] V. Bhavadharshini, S. Mridula, B. Sakthipriya, and J. J. Gracewell, "Semantic Segmentation using Convolutional Neural Networks," in 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), 2023: IEEE, pp. 481-488.

[12] C. Chen, C. Wang, B. Liu, C. He, L. Cong, and S. Wan, "Edge intelligence empowered vehicle detection and image segmentation for autonomous vehicles," IEEE Transactions on Intelligent Transportation Systems, 2023.

[13] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: Generalizing residual architectures," arXiv preprint arXiv:1603.08029, 2016.

[14] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: VGG and residual architectures," Frontiers in neuroscience, vol. 13, p. 95, 2019.

[15] D. Sinha and M. El-Sharkawy, "Thin mobilenet: An enhanced mobilenet architecture," in 2019 IEEE 10th annual ubiquitous computing, electronics & mobile communication conference (UEMCON), 2019: IEEE, pp. 0280-0285.

[16] M. Abu, A. Amir, Y. Lean, N. Zahri, and S. Azemi, "The performance analysis of transfer learning for steel defect detection by using deep learning," in Journal of Physics: Conference Series, 2021, vol. 1755, no. 1: IOP Publishing, p. 012041.

[17] S. Ghosh, A. Pal, S. Jaiswal, K. Santosh, N. Das, and M. Nasipuri, "SegFast-V2: Semantic image segmentation with less parameters in deep learning for autonomous driving," International Journal of Machine Learning and Cybernetics, vol. 10, pp. 3145-3154, 2019.

[18] M. Colley, B. Eder, J. O. Rixen, and E. Rukzio, "Effects of semantic segmentation visualization on trust, situation awareness, and cognitive load in highly automated vehicles," in Proceedings of the 2021 CHI conference on human factors in computing systems, 2021, pp. 1-11.

[19] F. Nesti, G. Rossolini, S. Nair, A. Biondi, and G. Buttazzo, "Evaluating the robustness of semantic segmentation for autonomous driving against real-world adversarial patch attacks," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 2280-2289.

[20] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, "Review the state-of-the-art technologies of semantic segmentation based on deep learning," Neurocomputing, vol. 493, pp. 626-646, 2022.

[21] L. M. Dang et al., "Lightweight pixel-level semantic segmentation and analysis for sewer defects using deep learning," Construction and Building Materials, vol. 371, p. 130792, 2023.

[22] S. Rakshit, "Training Semantic Segmentation Models for Autonomous Vehicles (A Step-by-Step Guide)," 2022. [Online]. Available: https://wandb.ai/av-demo/CamVid/reports/Training-Semantic-Segmentation-Models-for-Autonomous-Vehicles-A-Step-by-Step-Guide---VmlldzoyNTMyMjc4

[23] M. N. S. Jahromi et al., "Privacy-constrained biometric system for non-cooperative users," Entropy, vol. 21, no. 11, p. 1033, 2019.

[24] A. Khattar and S. Quadri, "Generalization of convolutional network to domain adaptation network for classification of disaster images on twitter," Multimedia Tools and Applications, vol. 81, no. 21, pp. 30437-30464, 2022.

[25] J. S. Talahua, J. Buele, P. Calvopiña, and J. Varela-Aldás, "Facial recognition system for people with and without face mask in times of the covid-19 pandemic," Sustainability, vol. 13, no. 12, p. 6900, 2021.