Drivable Area Detection for self-driving Cars Using Deep Learning

Hamza Jidal, Heba Gamal, Mehdi Elouaadi, Mehdi Zakroum
ESIN, International University of Rabat, Morocco
{hamza.jidal, hiba.gamal, mehdi.elouaadi, mehdi.zakroum}@uir.ac.ma

January 20, 2025

Abstract

This review explores the application of deep learning techniques for drivable area detection in autonomous vehicles, focusing on semantic segmentation models. The BDD100K dataset serves as a benchmark for evaluating prominent models, including UNet and DeepLabV3+. The analysis highlights the superior performance of DeepLabV3+, with a Mean Intersection over Union (Mean IoU) of 0.82, establishing it as a reference model for further exploration. The review also examines optimization techniques such as 8-bit quantization, magnitude-based pruning, and class-based thresholding (CBT) early exiting. Magnitude-based pruning demonstrated a substantial model size reduction of 66.53% while maintaining the same performance with a Mean IoU of 0.81, but attempts to implement CBT early exiting were hindered by resource limitations. These insights underscore the potential of optimized deep learning models in real-time drivable area detection, contributing to the advancement of autonomous driving systems.

1 Introduction

Drivable area detection is a crucial component of autonomous driving systems, enabling vehicles to navigate safely by identifying road regions suitable for driving [16]. Accurate detection of drivable areas ensures that self-driving cars can make informed decisions, avoiding obstacles and maintaining optimal paths. The dynamic and unpredictable nature of real-world environments, with varying road structures [10], lighting conditions [15], weather [13], and the presence of obstacles [11][5], presents significant challenges for drivable area detection. Urban settings further complicate the task with crowded intersections, pedestrians, and traffic signs, whereas rural or off-road environments may lack clear boundaries or markings. Numerous approaches have been proposed over time to address this challenge, with current State-of-the-art AI-based methods representing the most significant advancements to date [7]. Traditional techniques often face difficulties in generalizing across diverse scenarios, thereby underscoring the need for the development of robust and adaptable models.

Deep learning, particularly convolutional neural networks (CNNs), has revolutionized the field of semantic segmentation, offering promising solutions for drivable area detection. Semantic segmentation involves classifying each pixel in an image into a predefined category, making it an ideal approach for identifying drivable regions. Models such as UNet[14] and DeepLabV3+[3] have demonstrated exceptional performance in this domain, learning complex features directly from data to provide precise and real-time segmentation (Fig.2). These advancements underscore the potential of deep learning as a critical tool for autonomous driving systems.

Despite the progress, balancing accuracy and efficiency remains a significant challenge, especially for real-time applications. Autonomous vehicles require models that not only achieve high accuracy but also operate within the computational constraints of on-board systems. Therefore, the development of efficient models and the application of optimization techniques are essential for the practical deployment of deep learning-based drivable area detection.

The primary objective of this review is to develop a reliable and feasible approach for drivable area detection using deep learning. By leveraging the BDD100K dataset [17], specifically its segmentation folder, we train and evaluate two models, UNet and DeepLabV3+, to identify the most suitable architecture based on performance. To enhance efficiency, optimization techniques such as 8-bit quan-

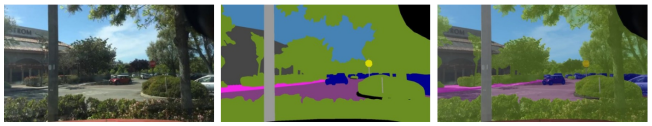

Figure 1: **BDD100K dataset, sample image and corresponding segmentation mask**

tization [8], pruning [9], and class-based thresholding (CBT) [6] for early-exit semantic segmentation are employed. These techniques aim to improve the computational efficiency of the selected model without compromising its accuracy.

2 Related Work

Autonomous driving has seen significant advancements, driven by deep learning techniques. A comprehensive overview categorizes key approaches into perception, localization, planning, and control, highlighting deep learning's essential role in enhancing autonomous systems [7].

Semantic segmentation, crucial for drivable area detection, has benefited from methodological advancements. The introduction of atrous convolutions and a decoder module has improved feature map resolution, while encoder-decoder architectures have proven effective at capturing contextual details [3, 14].

Enhancing model efficiency is critical for real-time deployment. Quantization strategies have been explored to improve inference efficiency, and layer-adaptive sparsity has been introduced for magnitude-based pruning [8, 9]. Additionally, class-based thresholding (CBT) enables early-exit semantic segmentation, reducing computational demands without compromising accuracy [6].

Our review leverages the BDD100k dataset, renowned for its extensive annotations, providing a robust foundation for evaluating semantic segmentation techniques across diverse driving conditions. In our implementation, we compared DeepLabv3+ [3] and U-Net [14] models, ultimately selecting DeepLabv3+ for its superior performance in accuracy and real-time inference. To further optimize our model, we employed techniques such as pruning [9], quantization [8], and class-based thresholding (CBT) for early exiting, drawing on insights from [6][12].

3 Methodology

Semantic segmentation, a fundamental task in computer vision, involves the classification of each pixel in an image into a predefined category. This task

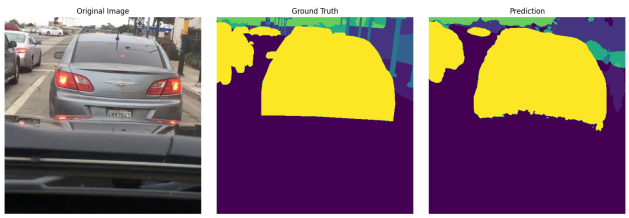

Figure 2: **Semantic segmentation model output i.e., DeepLabV3+. Each element in an input image is affected an identically shaped mask with a different color representing its label (image, Ground Truth, Prediction)**

is crucial in applications such as autonomous driving, medical imaging, and satellite image analysis. Recent advancements in deep learning have significantly enhanced the performance of semantic segmentation models, leading to remarkable improvements in accuracy and robustness. This study evaluates two prominent models, UNet[14] and DeepLabV3+[9], for the detection of drivable areas.

The **BDD100K** dataset [17] (Fig. 1), a comprehensive large-scale driving video dataset, was employed in this research. It features diverse driving scenarios across various weather conditions, times of day, and geographic regions, making it particularly suited for autonomous driving studies. The segmentation subset of the dataset, which contains pixel-level annotations for semantic segmentation tasks, was utilized. These annotations include delineations of drivable areas, road boundaries, and other relevant objects. The dataset consists of 7,000 images for training and 1,000 images for validation.

3.1 Model Selection and Architecture

UNet and DeepLabV3+ were selected for their demonstrated effectiveness in semantic segmentation tasks. Both models are designed to assign a label to each pixel within an image, making them particularly appropriate for identifying drivable areas in complex road environments.

UNet [14] is a convolutional neural network initially developed for biomedical image segmentation. Its architecture features a symmetrical encoder-decoder structure, where the encoder path captures contextual information through repeated convolutional and pooling layers. The decoder path restores spatial resolution through up-convolutions and concatenation with corresponding encoder feature maps. This architecture facilitates the integration of global context with precise localization, making UNet highly effective for segmenting intricate structures, even with limited training data.

DeepLabV3+ [3] enhances segmentation accu-

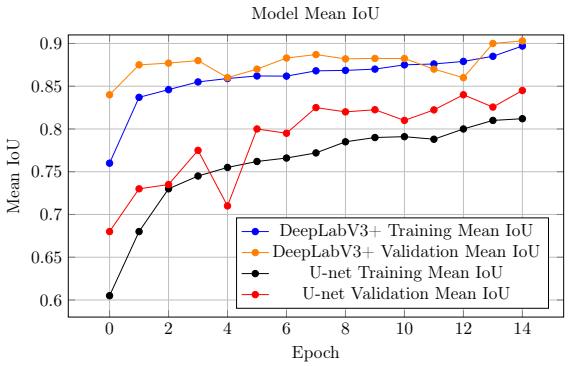

Figure 3: **Comparison of Training and Validation Mean IoU for DeepLabV3+ and U-net models across epochs**

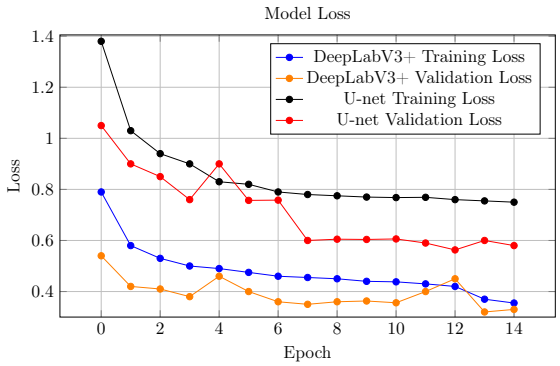

Figure 4: **Comparison of Training and Validation Loss for DeepLabV3+ and U-net models across epochs**

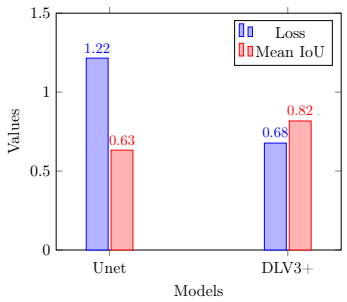

Figure 5: **Comparison of Test Loss and Mean IoU for Unet and DeepLabV3+ models.**

racy through the incorporation of Spatial Pyramid Pooling (SPP) and an encoder-decoder structure. It extends DeepLabV3 [1] by adding a decoder module to refine object boundaries. The model also employs Atrous Spatial Pyramid Pooling (ASPP) [2] and Depthwise Separable Convolutions (DSC) [4] to reduce computational complexity while enhancing segmentation performance, particularly in boundary delineation.

3.2 Training and Evaluation

Both UNet and DeepLabV3+ were trained using identical hyperparameters: 50 epochs and a learning rate of 1e-4 , implemented using TensorFlow. To enhance training stability and performance, regularization techniques were employed. Specifically, the ReduceLROnPlateau callback was used, monitoring validation loss val_loss with a reduction factor of 0.5, a patience of 5 epochs, and a minimum learning rate of . Additionally, the EarlyStopping callback monitored the validation mean Intersection over Union val_mean_iou, with a patience of 15 epochs and the option to restore the best weights. The batch sizes differed due to computational constraints, with DeepLabV3+ using a batch size of 4 and UNet using a batch size of 3. The training was carried out on an NVIDIA GeForce RTX 4070 8GB GPU.

The models were evaluated using the **Mean Intersection over Union** (Mean IoU) metric, which quantifies the accuracy of predicted pixel classifications against ground-truth labels. This metric is computed as the ratio of the intersection of predicted and true pixels to their union, averaged across all classes. A higher mean IoU indicates superior segmentation accuracy.

Both models employed **Sparse Categorical Crossentropy** as the loss function, which is well suited for multiclass classification problems with integer-labeled target values. This setup enabled a comprehensive comparison of the models' performance in detecting drivable areas.

3.3 Optimization Techniques

To enhance model performance, we implement different optimization techniques on the chosen DeepLabV3+ model such as pruning [9], quantization [8], and class-based thresholding (CBT) [6] for early exiting. These techniques aim to reduce computational overhead while maintaining high accuracy, ensuring the model's suitability for real-time applications.

3.3.1 8-bit Quantization

Quantization reduces the precision of the model's parameters from floating point to lower bit-width representations, such as 8-bit integers in our study. This technique reduces the memory footprint and accelerates computation, making the model more suitable for deployment on resource-constrained devices. We used **post-training quantization**[8] to optimize the DeepLabV3 + model using TensorFlow Lite, focusing on 8-bit quantization for both weights and activations. The mean intersection over Union (mean IoU) is integrated into the quantization pipeline to evaluate the model's performance effectively. After quantization, the model

is converted to TensorFlow Lite format, achieving a significant **91.46%** reduction in model size and inference latency. Meanwhile, the quantized model experienced a significant drop in performance, with a **low Mean IoU (0.16)**(Fig.7)(Fig.6) value that indicated a severe loss of accuracy during the quantization process. This outcome can be attributed to the already compact size of the DeepLabV3+ model, where applying 8-bit quantization further compromised its ability to retain essential feature representations. As a result, the trade-off between model size reduction and accuracy proved unfavorable, highlighting the limitations of quantization for models that are already highly optimized in terms of architecture.

3.3.2 Pruning

Magnitude-based pruning[9] involves removing weights with the smallest magnitudes from the network. This method assumes that weights with smaller absolute values have less impact on the model's output, allowing for a reduction in model size and computational requirements while retaining performance. Using TensorFlow Model Optimization (TF-MOT), intermediate layers were pruned to achieve 50% sparsity through a polynomial decay schedule. The pruned model was trained with updated pruning steps, achieving a compression ratio of **66.53%** while maintaining good performance(Mean IoU=0.81)(Fig.7).

3.3.3 Class-Based Thresholding in early-exit semantic segmantation

In an attempt to optimize our DeepLabV3+ model's computational efficiency, we explored implementing the Class-Based Thresholding (CBT) early exit approach proposed in [6]. CBT is an optimization technique that allows a semantic segmentation model to exit early for pixels it can confidently classify, thereby reducing computational overhead while maintaining accuracy.

The core principle of CBT is that different semantic classes have varying levels of inherent difficulty for classification. For instance, large uniform areas like roads might be easier to classify than complex objects like traffic signs. CBT leverages this insight by computing class-specific confidence thresholds based on the model's prediction patterns during training. These thresholds are then used during inference to determine whether a pixel's classification at an early exit point is sufficiently confident.

Given a model trained on a semantic segmentation task with K classes, CBT determines a threshold vector $T = [T_1 \cdots T_K] \in [0,1]^K$, where T_k corresponds to class k. For M training inputs with height H and width W, and N exit layers, the prediction probabilities at exit n for each (m,h,w) triplet are represented by the function $\phi_n : \mathbb{R}^{M\times H\times W} \to [0,1]^K$.

The computation of class-specific thresholds follows three main steps:

Step 1: For each exit layer n and class k, compute the mean prediction probabilities $p_{n,k}$ using all pixels belonging to class k (denoted by set S_k):

$$p_{n,k} = \frac{1}{|S_k|} \sum_{(m,h,w)\in S_k} \phi_n(m,h,w) \in [0,1]^K \quad (1)$$

This averaging helps obtain a broad sense of information about the difficulty of pixels belonging to each class.

Step 2: Calculate a global estimate P_k for each class by averaging the predictions across all exit layers. This information sharing across layers leverages insights from both shallow and deep layers:

$$P_k = \frac{1}{N} \sum_{n=1}^{N} p_{n,k} \in [0,1]^K \quad (2)$$

Step 3: Initialize each threshold T_k as the difference between the largest and second-largest elements of P_k. This initialization captures the model's confidence in distinguishing each class. Then, scale the thresholds inversely using parameters α and β:

$$T_k \leftarrow \left(1 - \frac{T_k - \min T}{\max T - \min T}\right)(\beta - \alpha) + \alpha \quad (3)$$

where α and β represent the minimum and maximum user defined threshold values respectively. This inverse scaling ensures that classes with high confidence scores will have low thresholds, allowing for earlier exits, while difficult classes maintain higher thresholds for more careful prediction.

Following the ADP-C paper's architecture [12], our implementation attempted to add two early exit points to the DeepLabV3+ model:

1. Before the ASPP (Atrous Spatial Pyramid Pooling) module to potentially skip the computationally intensive dilated convolutions
2. After the ASPP but before the decoder feature fusion

Each early exit was designed with an encoder-decoder structure, where the number of downsampling operations was determined by the exit's position in the network. This adaptive downsampling

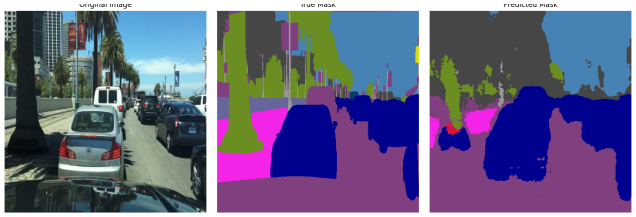

Figure 6: **Test result using the quantized model(image, true label, predicted label)**

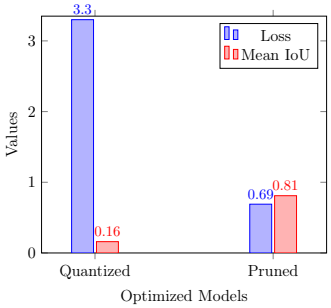

Figure 7: **Comparison Between the Optimized Models During the Test.**

strategy aimed to compensate for the limited receptive field at earlier layers.

However, our implementation faced significant memory constraints during training. Despite attempts to optimize memory usage through various approaches, the model consistently exceeded available GPU memory. This practical limitation highlights an important consideration in deploying optimization techniques: while theoretically promising, the memory overhead of additional exit paths can potentially outweigh their computational benefits, particularly in memory-constrained environments.

This experience underscores the need for careful consideration of hardware constraints when implementing architectural modifications to deep learning models. Future work could explore more memory-efficient implementations of early exit strategies or alternative optimization approaches better suited to resource-constrained environments.

4 Results and Discussion

4.1 Model Performance

The performance of the **DeepLabV3+** model was evaluated on BDD100K (Fig.2). The model achieved a mean Intersection over Union (Mean IoU) of **0.82** and a loss of **0.68**, as illustrated in Figure 5. Additionally, the model was tested on video data, where it demonstrated promising results, processing a total of 1016 frames in **93.00 seconds**, with an average time of **0.0915 seconds per frame**. These results highlight the model's ability to generalize effectively across diverse data formats and environments.

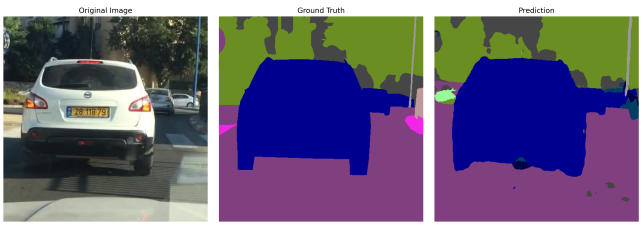

Figure 8: **Test results using the pruned model(image, true label, predicted label)**

4.2 Optimization Results

As shown in Figure 7, the optimization process provided valuable insights into the performance of different optimized models. Among the techniques applied, **Magnitude-Based Pruning** demonstrated the best balance between accuracy and efficiency, achieving a Mean IoU of **0.81** and a loss of **0.69**. In contrast, the **Quantized model** experienced a significant drop in accuracy, with a Mean IoU of **0.16** and a higher loss of **3.3**, indicating a severe trade-off in performance due to the quantization process.

4.3 Final Model Selection

Based on the performance evaluation and optimization results, the **DeepLabV3+** model was selected as the baseline due to its superior Mean IoU and loss metrics on the BDD100K dataset(Fig.2)(Fig.5). Following the optimization phase, the pruned DeepLabV3+ model was chosen as the final optimized model(Fig.8)(Fig.7). This decision was driven by its ability to maintain performance with a Mean IoU of **0.81**, while significantly reducing model size (**66.53% reduction**) and improving computational efficiency. The pruned model demonstrated an ideal balance between accuracy and resource optimization, making it the best candidate for practical deployment.

5 Conclusion and Future Work

5.1 Conclusion

This review successfully demonstrated the application of deep learning for drivable area detection in autonomous vehicles. By leveraging the BDD100K dataset[17], DeepLabV3+[3] emerged as the best-performing model, achieving superior accuracy compared to UNet[14]. Optimization techniques were applied to enhance the model's efficiency. Among these, magnitude-based pruning[9] achieved significant reductions in model size (66.53%) while

maintaining high accuracy (Mean IoU: 0.81). These findings underscore the effectiveness of optimization techniques in balancing accuracy and computational efficiency, paving the way for real-time deployment in resource-constrained environments.

5.2 Future Work

Future research can focus on further enhancing model efficiency through advanced optimization methods, such as mixed-precision quantization and adaptive sparsity techniques. Expanding the dataset to include more challenging scenarios, such as adverse weather conditions or off-road environments, would improve generalization. Additionally, integrating multi-sensor data, such as LiDAR and radar, could complement image-based segmentation for robust drivable area detection. Finally, exploring real-time inference on edge devices will ensure practical deployment in autonomous vehicles.

References

[1] Liang-Chieh Chen. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[3] 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)*, pages 801–818, 2018.

[4] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[5] Noa Garnett, Shai Silberstein, Shaul Oron, Ethan Fetaya, Uri Verner, Ariel Ayash, Vlad Goldner, Rafi Cohen, Kobi Horn, and Dan Levi. Real-time category-based and general obstacle detection for autonomous driving. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 198–205, 2017.

[6] Alperen Görmez and Erdem Koyuncu. Class based thresholding in early exit semantic segmentation networks. *IEEE Signal Processing Letters*, 2024.

[7] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of field robotics*, 37(3):362–386, 2020.

[8] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

[9] Jaeho Lee, Sejun Park, Sangwoo Mo, Sungsoo Ahn, and Jinwoo Shin. Layer-adaptive sparsity for the magnitude-based pruning. *arXiv preprint arXiv:2010.07611*, 2020.

[10] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4628–4634. IEEE, 2022.

[11] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7345–7353, 2019.

[12] Zhuang Liu, Zhiqiu Xu, Hung-Ju Wang, Trevor Darrell, and Evan Shelhamer. Anytime dense prediction with confidence adaptivity. *arXiv preprint arXiv:2104.00749*, 2021.

[13] Chenghao Qian, Mahdi Rezaei, Saeed Anwar, Wenjing Li, Tanveer Hussain, Mohsen Azarmi, and Wei Wang. Allweather-net: Unified image enhancement for autonomous driving under adverse weather and low-light conditions. In *International Conference on Pattern Recognition*, pages 151–166. Springer, 2025.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[15] Aashish Sharma and Robby T Tan. Nighttime visibility enhancement by increasing the dynamic range and suppression of light effects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11977–11986, 2021.

[16] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE intelligent vehicles symposium (IV)*, pages 1013–1020. IEEE, 2018.

[17] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.