2023 International Conference on Multimedia Analysis and Pattern Recognition (MAPR) | 979-8-3503-2741-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/MAPR59823.2023.10288646

TwinLiteNet: An Efficient and Lightweight Model for Driveable Area and Lane Segmentation in Self-Driving Cars

Quang-Huy Che[1, 2],Dinh-Phuc Nguyen[1, 2], Minh-Quan Pham[1, 2],Duc-Khai Lam[1, 2]
[1]University of Information Technology, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
Email:{18520819, 20521766}@gm.uit.edu.vn, {quanpm, khaild}@uit.edu.vn

***Abstract*—Semantic segmentation is a common task in autonomous driving to understand the surrounding environment. Driveable Area Segmentation and Lane Detection are particularly important for safe and efficient navigation on the road. However, original semantic segmentation models are computationally expensive and require high-end hardware, which is not feasible for embedded systems in autonomous vehicles. This paper proposes a lightweight model for the driveable area and lane line segmentation. TwinLiteNet is designed cheaply but achieves accurate and efficient segmentation results. We evaluate TwinLiteNet on the BDD100K dataset and compare it with modern models. Experimental results show that our TwinLiteNet performs similarly to existing approaches, requiring significantly fewer computational resources. Specifically, TwinLiteNet achieves a mIoU score of 91.3% for the Drivable Area task and 31.08% IoU for the Lane Detection task with only 0.4 million parameters and achieves 415 FPS on GPU RTX A5000. Furthermore, TwinLiteNet can run in real-time on embedded devices with limited computing power, especially since it achieves 60FPS on Jetson Xavier NX, making it an ideal solution for self-driving vehicles. Code is available: https://github.com/chequanghuy/TwinLiteNet.**

***Index Terms*—Segmentation, Self Driving Car, Computer vision, Light Weight model, Edge Computing**

I. Introduction

Self-driving cars have emerged as a promising field in recent years, potentially revolutionizing transportation and road development. One crucial component of autonomous driving is perceiving the environment accurately and efficiently. Deep learning has been applied to real-world driving control tasks [1] [2]. Semantic segmentation is a fundamental task in this problem, involving labeling each pixel in an image with a corresponding semantic class, such as road, vehicle, pedestrian, etc. This information can help autonomous cars navigate safely and avoid obstacles. Specifically, accurately detecting drivable areas and lane markings provides critical information for the system to make steering and lane-changing decisions. However, models like UNet [3], SegNet [4], ERNet [5] for semantic segmentation; LaneNet [6] and SCNN [7], ENet-SAD [8] for lane line detection; YOLOP [9], YOLOPv2 [10], HybridNets [11], DLT-Net [12], Multinet [13] for multi-task problems often come with high computational costs and require high-end hardware, which is not suitable for embedded systems used in low-computational power autonomous vehicles.

This paper introduces a lightweight architecture that can be easily deployed on autonomous vehicle systems. The main contributions of our report are (1) A computationally efficient architecture for drivable area segmentation and lane detection. (2) Our proposed architecture, based on ESPNet [14], is an expandable convolutional segment network with depth-wise separable convolutions combined with Dual Attention Network [15], but instead of using a single decoder block, our TwinLiteNet utilizes two decoder blocks similar to YOLOP [9], YOLOPv2 [10] for each task (3) Our experimental results demonstrate that TwinLiteNet achieves comparable performance with fewer parameters for various image segmentation tasks.

The remainder of the paper is represented in the sequence listed below: We evaluate relevant models in Section II to grasp the benefits and drawbacks of modern models, such as Driveable Area Segmentation, Lane Detection, and Multi-task approaches so that we can benefit from their strengths and minimize the weaknesses of our model. The proposed TwinLiteNet, based on the preceding analysis, presents an architecture with methods to boost model performance in Section III. The model we suggest strives for high processing speed and straightforward implementation on embedded computers. In Section IV, we evaluate TwinLiteNet and contrast it with other models performing the same task to determine how well it does. In the final Section, we provide some conclusions and future development directions.

II. Related work

A. Drivable Area Segmentation

Semantic segmentation has been extensively researched in computer vision, and many efficient models have been developed for semantic segmentation in self-driving or object segmentation tasks. Specifically, for segmenting drivable areas, recent works have proposed efficient models that can achieve accurate results with low computational costs. ENet [5] is a lightweight CNN model that can run on embedded devices with limited resources in real-time. In the research [16], the authors have found using the Hybrid Dilated Convolution module to extract better feature representations from the input images for the segmentation task. Zhao et al. [17] devised the PSPNet model, which utilizes the Pyramid Pooling Module (PPM)

that applies global average pooling with multiple different bin scales to extract features. Alongside complex computational models, Mehta et al. [14] have proposed ESPNet with low computational costs, using Dilated Convolutions to build an efficient spatial pyramid (ESP) module. In addition to developing and presenting new models, Dual Attention Modules [15] is proposed to enhance feature fusion.

B. Lane Detection

Segmentation based on deep learning is a practical approach for many studies on lane detection. Dual branch features are combined to perform the segmentation of lane versions. Although time-consuming, convolutional slicing of SCNN [7] allows information to propagate across pixels along rows and columns within a layer. On the other hand, Enet-SAD [8] utilizes a self-attention-guided filter method to assist low-level feature maps in learning from high-level feature maps. In addition to segmentation, in recent years road markings [18] [19] have also brought much attention from the community.

C. Multi-task Approaches

Multi-task learning is a popular approach to simultaneously address multiple tasks, allowing the model to learn shared representations and leverage commonalities across different tasks. A standard method for multi-task learning is to use a shared backbone network and separate heads for each task. For example, Mask R-CNN [20] is a model combining object detection and instance segmentation using a shared backbone network and different charges for each task. Similarly, LaneNet [6] is a model combining lane detection and lane segmentation using a shared backbone network and separate heads for each task. MultiNet [13] accomplishes three tasks simultaneously: image classification, object recognition, and region segmentation for autonomous driving. To facilitate the exchange of information between tasks, DLT-Net [12] inherits the encoder-decoder structure and establishes cross-modal attention maps between task-specific decoder modules. [21] proposes inter-connected sub-structures between lane region segmentation and lane boundary detection. Additionally, it offers a unique minimization function to constrain lane boundaries to overlap with the lane region geometrically. Driveable Area Segmentation and Lane Detection are all tasks accomplished by Hybridnets [11] using a shared encoder and three separate decoders. Recently, YOLOPv2 [10] have proposed to use Bag-of-Freebies methods to achieve high accuracy and fast processing speed.

III. Proposed method

In this section, we develop the lightweight model in detail. First, we first propose a model with an architecture of one encoder and two decoders for two separate tasks.; our TwinLiteNet consists of one input and two outputs so that the model learns the representation of two different tasks. We then recommend a Dual Attention Module to improve model performance. In addition, this section also proposes some loss functions used to train the model. We also present the training and inference mechanisms we use. The section below demonstrates our proposed method in detail.

A. Model architectures

This paper presents a cost-effective architecture designed for task segmentation called TwinLiteNet, as illustrated in Figure 1. Our approach utilizes ESPNNet-C as an information encoding block, enabling efficient feature map generation. We incorporate Dual Attention Modules into the network to capture global dependencies in both spatial and channel dimensions. These modules enhance the network's ability to perceive contextual information. The resulting feature map is then fed through two encoder blocks dedicated to performing two specific tasks: Driveable Area Segmentation and Lane Detection. By employing this architecture, we aim to achieve accurate and efficient segmentation results for these tasks at a reduced cost.

Firstly, unlike approaches using backbones and high computational cost methods, we leverage the power of ESPNet with low computational cost but high accuracy. We use ESPNet-C as an encoder to extract features from the input image. In ESPNNet-C, in addition to sharing information through feature maps between ESP modules, it also consolidates input information at different dimensions between blocks in the architecture. After obtaining the feature map $A \in \mathbb{R}^{32\times\frac{H}{8}\times\frac{W}{8}}$ from ESPNet-C, we pass the extracted features through Dual Attention Modules, we pass the extracted features through Dual Attention Modules [15]. The Dual Attention Modules consist of the Position Attention Module (PAM) and the Channel Attention Module (CAM). The PAM module is designed to incorporate a wider range of contextual information into local features, enhancing their representation capabilities. On the other hand, the CAM module leverages the interdependence among channel maps to highlight the interdependencies of feature maps and strengthen the representation of specific semantics. We transform the outputs of two attention modules by a Convolutional layer and employ an element-wise sum operation to achieve feature fusion $B \in \mathbb{R}^{32\times\frac{H}{8}\times\frac{W}{8}}$. Our paper proposes a multi-output design for the Driveable Area and Lane segmentation tasks. Instead of using a single output for all object types requiring segmentation, we employ two decoder blocks to process the feature map and obtain the final results for each task. We recommend this multi-output design for the following reasons:

- Independent Performance Optimization: With two dedicated output blocks, we can optimize the segmentation performance independently for each class. This approach allows us to fine-tune and improve the segmentation results for Driveable Areas and Lanes separately without being influenced by the other class.
- Enhanced Accuracy: Using two output blocks for separate layers also improves segmentation accuracy. By focusing on each layer independently, our model can better learn and adjust the features specific to Driveable Areas and Lanes, leading to more accurate segmentation results for each class.

By adopting a multi-output design with two separate outputs for the Driveable Areas and Lanes segmentation tasks, we achieve independent performance optimization and enhanced segmentation accuracy for each class. Our decoder blocks are designed to be simple but ensure efficiency, relying on ConvTranspose layers followed by batch normalization and the pRelu [22] activation function, as shown in Figure 2. After decoding, TwinLiteNet returns two segmented images for the Driveable Area and Lane Detection tasks. Our TwinLiteNet optimizes the segmentation performance with high accuracy for both driveable area segmentation and lane detection tasks. By utilizing ESPNet-C and the feature analysis blocks Dual Attention Network, we enhance the feature extraction capability of the model. Besides, the simple decoder blocks help reduce computational costs and improve the model's efficiency. The output of the backbone and dual block has a small size of $32 \times \frac{H}{8} \times \frac{W}{8}$, so the calculation of the subsequent blocks is optimized, leading to low inference time, and at the same time, Section IV is also proven with the proposed size. The model still carries enough information for learning and prediction.

B. Loss function

We utilize two loss functions for the proposed segmentation model: Focal Loss [23] and Tversky Loss [24]. Focal loss aims to reduce the classification error among pixels while addressing the impact of easily predictable samples and heavily penalizing hard-to-predict samples, as shown in equation 1. On the other hand, Tversky Loss draws inspiration from Dice Loss [25] and addresses the class imbalance issue in segmentation tasks. However, unlike Dice Loss, Tversky Loss introduces α and β parameters to adjust the importance of false positives and false negatives during the computation, as described in the Tversky equation 2.

$$Loss_{focal} = -\frac{1}{N}\sum_{c=0}^{C-1}\sum_{i=1}^{N} p_i(c)(1-\hat{p}_i(c))^{\gamma} log(\hat{p}_i(c)) \quad (1)$$

where:

- N: A number of pixels in the input image
- C: A number of classes, in this case, one class is Drivable Area or Lane, and the remaining class is the background.
- $\hat{p}_i(c)$: is to determine the prediction for pixel i of class c
- $p_i(c)$: is the ground truth value for pixel i of class c.
- γ: balance correction factor.

$$Loss_{tversky} = \sum_{c=0}^{C}(1 - \frac{TP(c)}{TP(c) - \alpha FN(c) - \beta FP(c)}) \quad (2)$$

where:

- TP: True Positives
- FN: False Negatives
- FP: False Positives
- C: Number of classes, in this case, one class is Drivable Area or Lane, and the remaining class is the background.
- α, β: Control the magnitude of penalties for FPs and FNs.

TABLE I: Computation Performance.

Network	Params	Speed (FPS)
YOLOP [9]	7.9M	93
YOLOPv2 [10]	38.9M	95
HybridNets [11]	13.8M	25
TwinLiteNet (our)	**0.4M**	**415**

The aggregated loss function for each head take the form of the following:

$$Loss_{total} = Loss_{focal} + Loss_{tversky} \quad (3)$$

C. Training mechanism and inference mechanism

We trained our TwinLiteNet with input images of size 640x360. We used the Adam [26] optimizer with a decreasing learning rate over epochs. TwinLiteNet was trained in 100 epochs with a batch size of 32. During the inference process, we applied the Re-parameterization technique to merge the Convolution and Batch Normalization [27] layers into a single layer, which accelerated the inference speed. This merging process only occurs during inference, while during model training, they still operate as separate layers: Convolution and Batch Normalization.

IV. Experimental results

The BDD100K [28] dataset was used for training and validating TwinLiteNet. With 100,000 frames and annotations for 10 tasks, it is a large dataset for autonomous driving. Due to its diversity in geography, environment, and weather conditions, the algorithm trained on the BDD100K dataset is robust enough to generalize to new settings. The BDD100K dataset is divided into three parts: a training set with 70,000 images, a validation set with 10,000 images, and a test set with 20,000 images. Since no labels are available for the 20,000 images in the test set, we chose to evaluate on a separate validation set of 10,000 images. We resized the images in the BDD100k dataset from 1280×720×3 to 640×360×3. The preprocessing steps described in this study were conducted according to [8]. We used TorchScript to convert our model to a statically typed representation, which resulted in a speedup of inference without sacrificing accuracy.

All experiments used the PyTorch framework on an NVIDIA GeForce RTX A5000 GPU with 32GB of RAM and an Intel(R) Core(TM) i9-10900X processor.

A. Cost Computation Performance

Table I compares the proposed model with other multi-task networks[1]. Our TwinLiteNet has only 0.4M parameters, much lower than previous models. Besides, our TwinLiteNet achieves 415FPS, while other models only achieve inference speed below 100FPS on the same test device.

[1]These models also include task object detection.

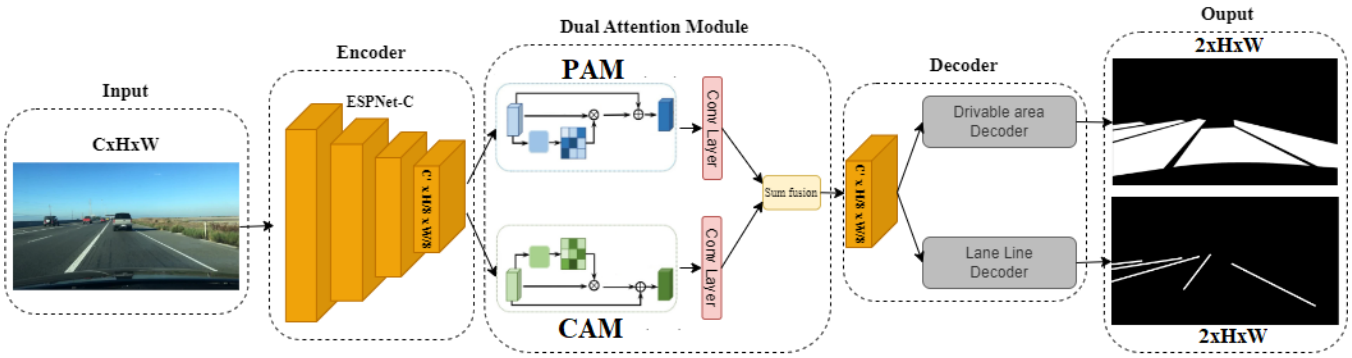


Fig. 1: The detailed TwinLiteNet architecture

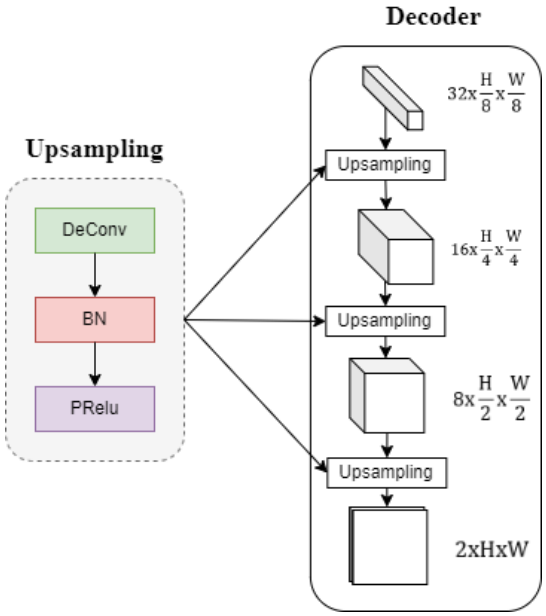


Fig. 2: Design of the proposed encoder block for the TwinLiteNet

B. Drivable Area Segmentation Result

In this paper, both the "drivable area" and "alternative area" in the BDD100K dataset were converted to "drivable area". The mean Intersection over Union (mIoU) metric was used to evaluate the segmentation performance of our TwinLiteNet. The results are shown in Table II. Although the accuracy of TwinLiteNet is higher than some previously proposed models, our TwinLiteNet is slightly lower than the current State-of-the-Art (SOTA) YOLOPv2 (-1.88%) and lower than YOLOP (-0.18%). It is easy to understand because TwinLiteNet is developed to optimize inference time rather than accuracy. Some segmentation results of the model are shown in Figure 3a. The results showcase the impressive performance of TwinLiteNet in accomplishing the task under diverse lighting conditions, including both daytime and nighttime scenarios. Notably, the network demonstrates high accuracy in segmenting the Driveable Area while effectively avoiding confusion with the opposite lane.

TABLE II: Drivable Area Segmentation Results.

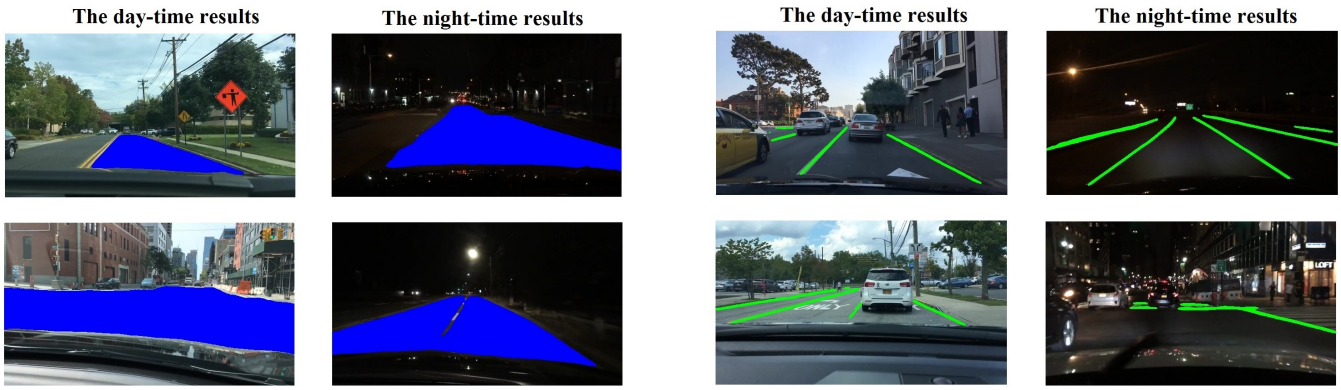
Network	mIoU (%)
MultiNet [13]	71.6
DLT-Net [12]	71.3
PSPNet [17]	89.6
HybridNets [11]	90.5
YOLOP [9]	91.5
YOLOPv2 [10]	**93.2**
TwinLiteNet (our)	91.3 (-1.9)

C. Lane Detection Result

The lane lines in the BDD100K dataset are labeled using lines with low pixel thickness. To compare expediently with previous research, following [8], we adopted the methodology of merging two-lane line annotations into a single central line during the annotation process. In the training set, we applied dilation to expand the annotations by 8 pixels so that the model could learn better while keeping the validation set the same to ensure consistency with previous and future research. We use pixel accuracy and IoU (Intersection over Union) of lane lines as evaluation metrics. As listed in Table III, TwinLiteNet is still lower than the current SOTA HybridNets in terms of IoU (-0.52%), but TwinLiteNet's accuracy reaches 97.3%, much higher than the current models. Some segmentation results of the model are shown in Figure 3b. Our experimental findings reveal that our model exhibits strong predictive capabilities for multi-lane roads, performing admirably in daytime and nighttime scenarios. This observation underscores the model's ability to accurately anticipate and predict lane configurations, regardless of the lighting conditions.

D. Ablation studies

In this section, we investigate the ablation study. The ablation options we proposed and their corresponding results are presented in Table IV. We evaluate the results from a simple baseline that only includes the backbone and one output for the whole Drivable Area as Lane Detection. Then we gradually add Dual Attention Module, Multiple Head, and Re-parameterization methods. The results show that with the baseline model, we achieve the highest inference speed with



(a) Drivable area segmentation (b) Lane detection

Fig. 3: Visualization of the results

TABLE III: Lane Detection Results.

Network	Accuracy (%)	IoU (%)
ENet [5]	34.12	14.64
SCNN [7]	35.79	15.84
R-101-SAD [8]	35.56	15.96
ENet-SAD [8]	36.56	16.02
YOLOP [9]	70.50	26.20
YOLOPv2 [10]	87.31	27.25
HybridNets [11]	85.4	**31.6**
TwinLiteNet (our)	**97.3 (+9.99)**	31.08 (-0.52)

530 FPS but at the expense of much accuracy compared to the fully integrated model, which achieves 415 FPS inference speed.

E. Edge Devices

This section presents the inference speed results of deploying our TwinLiteNet model on embedded devices, specifically the NVIDIA Jetson TX2 and Jetson Xavier NX. We utilized the TensorRT SDK for performing model inference on these Jetson devices. The results demonstrate that our TwinLiteNet model achieves real-time computation on edge devices, with the Jetson Xavier NX achieving a frame rate of 60 FPS and the Jetson TX2 reaching 25 FPS. To evaluate the performance of our model, we recorded the prediction process on the edge devices[2] using test videos from the BDD100K dataset with some visible results shown in Figure 4. These results demonstrate that the TwinLiteNet model achieves accurate predictions in both daytime and nighttime images across various embedded devices.

Furthermore, we monitored the power consumption and operating temperature of the TwinLiteNet model on the embedded devices. Figure 5a illustrates the recorded power consumption, and Figure 5b depicts the operating temperature of the model during inference on the proposed edge devices. These measurements provide insights into the energy efficiency and thermal performance of our TwinLiteNet model in real-world deployment scenarios.

[2]https://youtube.com/shorts/TSklO0L4XJs

V. Conclusion

We introduce a lightweight and energy-efficient segmentation model for autonomous driving tasks, specifically Drivable Area and Lane Detection. Our TwinLiteNet aims to achieve high processing speed with a slight trade-off in accuracy. Evaluation of the BDD100K dataset demonstrates that our model delivers a good balance between accuracy and high computational speed on GPUs and even on edge devices. In the future, we intend to evaluate the performance of the TwinLiteNet model on various publicly available datasets and apply the model to real-world scenarios. This approach enables us to assess its effectiveness in different contexts and address practical challenges.

Acknowledgement

This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number DS2023-26-02.

References

[1] M.-Q. Pham, H.-P. Ly, H. C. Quang, T. Nguyen-Van, D. T. Tung, and M. Le-Hoang, "Transform an electric golf cart into an autonomous vehicle," in *2022 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, 2022, pp. 1–6.

[2] Autopilot — tesla. [Online]. Available: https://www.tesla.com/autopilot

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *CoRR*, vol. abs/1511.00561, 2015. [Online]. Available: http://arxiv.org/abs/1511.00561

[5] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *CoRR*, vol. abs/1606.02147, 2016. [Online]. Available: http://arxiv.org/abs/1606.02147

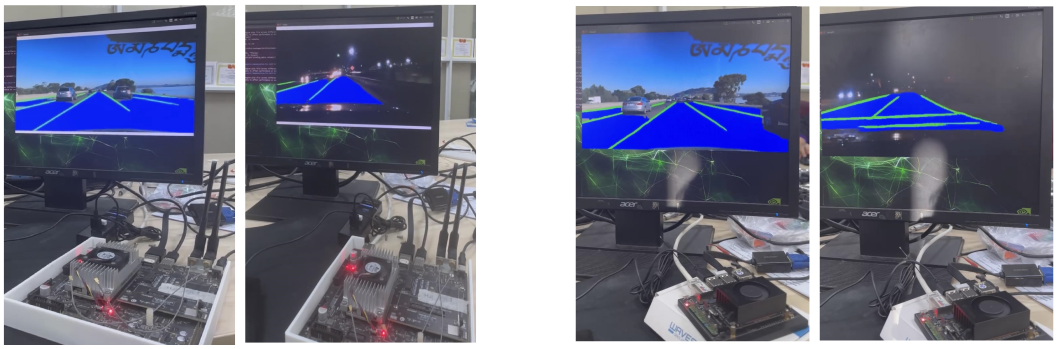
[6] Z. Wang, W. Ren, and Q. Qiu, "Lanenet: Real-time lane detection networks for autonomous driving," *CoRR*, vol. abs/1807.01726, 2018.

[7] A. Parashar, M. Rhu, A. Mukkara, A. Puglielli, R. Venkatesan, B. Khailany, J. S. Emer, S. W. Keckler, and W. J. Dally, "SCNN: an accelerator for compressed-sparse convolutional neural networks," *CoRR*, vol. abs/1708.04485, 2017. [Online]. Available: http://arxiv.org/abs/1708.04485

[8] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning lightweight lane detection cnns by self attention distillation," *CoRR*, vol. abs/1908.00821, 2019.

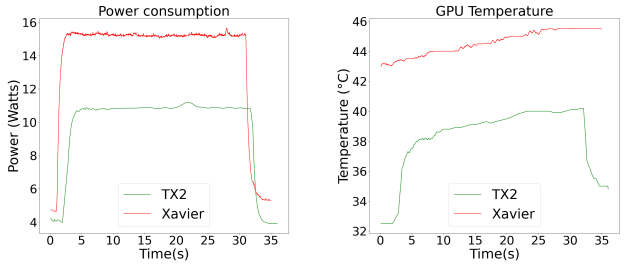
TABLE IV: Ablation studies

Dual Attention Module	Multiple Head	Re-parameterization	Drivable Area mIoU (%)	Lane Detection Accuracy (%)	Lane Detection IoU (%)	FPS	Parameter
			88.7	78.9	24.7	**530**	**417020**
✓			89.3	79.1	25.6	425	436966
✓	✓		**91.3**	**97.3**	**31.08**	400	439633
✓	✓	✓	**91.3**	**97.3**	**31.08**	415	439339



(a) Jetson TX2 (b) Jetson Xavier

Fig. 4: Visualization of some results when implemented on the kit



(a) Power consumption (b) Temperature consumption

Fig. 5: Consumption on TX2 vs. Xavier

[9] D. Wu, M. Liao, W. Zhang, and X. Wang, "YOLOP: you only look once for panoptic driving perception," *CoRR*, vol. abs/2108.11250, 2021. [Online]. Available: https://arxiv.org/abs/2108.11250

[10] C. Han, Q. Zhao, S. Zhang, Y. Chen, Z. Zhang, and J. Yuan, "Yolopv2: Better, faster, stronger for panoptic driving perception," 2022.

[11] D. Vu, B. Ngo, and H. Phan, "Hybridnets: End-to-end perception network," 2022.

[12] Y. Qian, J. M. Dolan, and M. Yang, "Dlt-net: Joint detection of drivable areas, lane lines, and traffic objects," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4670–4679, 2020.

[13] M. Teichmann, M. Weber, J. M. Zöllner, R. Cipolla, and R. Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," *CoRR*, vol. abs/1612.07695, 2016. [Online]. Available: http://arxiv.org/abs/1612.07695

[14] S. Mehta, M. Rastegari, A. Caspi, L. G. Shapiro, and H. Hajishirzi, "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," *CoRR*, vol. abs/1803.06815, 2018. [Online]. Available: http://arxiv.org/abs/1803.06815

[15] J. Fu, J. Liu, H. Tian, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," *CoRR*, vol. abs/1809.02983, 2018. [Online]. Available: http://arxiv.org/abs/1809.02983

[16] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. W. Cottrell, "Understanding convolution for semantic segmentation," *CoRR*, vol. abs/1702.08502, 2017. [Online]. Available: http://arxiv.org/abs/1702.08502

[17] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *CoRR*, vol. abs/1612.01105, 2016. [Online]. Available: http://arxiv.org/abs/1612.01105

[18] Y. Ko, Y. Lee, S. Azam, F. Munir, M. Jeon, and W. Pedrycz, "Key points estimation and point instance segmentation approach for lane detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8949–8958, 2022.

[19] Z. Qin, P. Zhang, and X. Li, "Ultra fast deep lane detection with hybrid anchor driven ordinal classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–14, 2022.

[20] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[21] J. Zhang, Y. Xu, B. Ni, and Z. Duan, "Geometric constrained joint lane segmentation and lane boundary detection," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 502–518.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *CoRR*, vol. abs/1502.01852, 2015. [Online]. Available: http://arxiv.org/abs/1502.01852

[23] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017. [Online]. Available: http://arxiv.org/abs/1708.02002

[24] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, "Tversky loss function for image segmentation using 3d fully convolutional deep networks," *CoRR*, vol. abs/1706.05721, 2017. [Online]. Available: http://arxiv.org/abs/1706.05721

[25] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," *CoRR*, vol. abs/1707.03237, 2017. [Online]. Available: http://arxiv.org/abs/1707.03237

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[28] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," 2020.