applied sciences

MDPI

Article

Joint Semantic Understanding with a Multilevel Branch for Driving Perception

Dong-Gyu Lee [1] and Yoon-Ki Kim [2,*]

[1] Department of Artificial Intelligence, Kyungpook National University, Buk-gu, Daegu 41566, Korea; dglee@knu.ac.kr
[2] Department of Future Convergence, The Cyber University of Korea, Jongno-gu, Seoul 03051, Korea
* Correspondence: ykkim77@cuk.edu; Tel.: +82-2-6361-1937

Abstract: Visual perception is a critical task for autonomous driving. Understanding the driving environment in real time can assist a vehicle in driving safely. In this study, we proposed a multi-task learning framework for simultaneous traffic object detection, drivable area segmentation, and lane line segmentation in an efficient way. Our network encoder extracts features from an input image and three decoders at multilevel branches handle specific tasks. The decoders share the feature maps with more similar tasks for joint semantic understanding. Multiple loss functions are automatically weighted summed to learn multiple objectives simultaneously. We demonstrate the effectiveness of this framework on a BerkeleyDeepDrive100K (BDD100K) dataset. In the experiment, the proposed method outperforms the competing multi-task and single-task methods in terms of accuracy and maintains a real-time inference at more than 37 frames per second.

Keywords: joint semantic understanding; multi-level branch network; drivable area segmentation; lane line segmentation; traffic object detection; real-time inference; multi-task learning

Citation: Lee, D.-G.; Kim, Y.-K. Joint Semantic Understanding with a Multilevel Branch for Driving Perception. *Appl. Sci.* **2022**, *12*, 2877. https://doi.org/10.3390/app12062877

Academic Editors: Antonio Fernández-Caballero, Jae-Mo Kang and Dong-Woo Lim

Received: 9 February 2022
Accepted: 9 March 2022
Published: 11 March 2022

1. Introduction

Driving perception is one of the most challenging tasks for intelligent driving systems because of the high complexity of environments. With the advance of computer vision techniques, visual perception gains a lot of attention in the field of intelligent vehicles. Visual information is extracted from images taken by cameras to assist the decision of the driving assistant system. One of the critical abilities for safe driving in real-world applications is knowing the region where the vehicle can go without any type of danger.

In previous studies, visual perception tasks are handled separately. For example, detection of traffic objects is done by object detection methods, such as Faster R-CNN [1] and YOLO [2]. SCNN [3] and ENet-SAD [4] are used for lane line detection. Semantic segmentation methods such as PSPNet [5] and SegNet [6] provide more detailed information about the road. These studies achieved significant results in their respective fields. Despite the excellence of these methods, these tasks are processed separately. However, in real applications, processing of these tasks is simultaneously required in visual perception for autonomous driving. Limited computational resources of embedded devices should be considered.

Recent advances in deep learning-based multi-task learning approaches unified all problems into a single recognition task, whereas traditional computer vision methods were focused on specific techniques for detecting traffic objects independently. Furthermore, different tasks in the images often have mutual information. For example, lane lines must be located on the ground area of the image along with the drivable area. The ground area where the traffic objects exist is an occupied space where the ego-vehicle should not drive. Therefore, a unified structure that can process this comprehensive information can have a better result while reducing the computational cost. The multi-task learning approach with a shared representation can provide efficient learning and

improved prediction accuracy [7]. Multinet [8] proposes an encoder–decoder structure that is composed of one encoder and three decoders. The network performs well on detection, classification, and segmentation tasks. However, they missed the lane detection task, which is critical for intelligent driving assistance. Lane detection can guide the direction of vehicles, whereas semantic segmentation provides more details. DTL-Net [9] learns a network to solve joint detection of traffic objects, drivable areas, and lane lines in a single architecture. The shared mutual information is well fused; however, it is not performed in real time.

Most encoder–decoder architecture processes input images at the encoding stage to generate a shared representation, then passes the output to the decoder as a shared representation. Each decoder learns from shared representation for their specific tasks [8–10]. However, depending on the characteristics of each task, the required information may differ. For instance, drivable area segmentation is a highly related task to the lane line segmentation than traffic object detection. Drivable areas share the ground regions with lines, but traffic objects do not. The traffic object partially takes a place on the ground, whereas drivable areas and lane lines are fully placed on the ground. We design a multilevel branch architecture for joint semantic understanding to share the layer weight with more similar tasks. An example of three tasks for driving visual perception is shown in Figure 1. The yellow bounding box indicates traffic objects, the red lines represent the lane lines, and the green areas are the drivable areas.

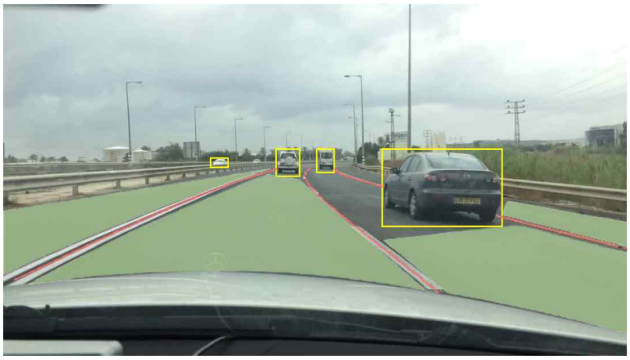

Figure 1. An example of three tasks; traffic object detection, drivable area segmentation, and lane line segmentation.

In this study, we propose an efficient end-to-end multi-task learning framework to accelerate driving perception. The proposed framework consists of one shared encoder and three decoders. Each decoder performs the following three tasks simultaneously: (1) traffic object detection, i.e., detecting other vehicles on the road, (2) drivable area segmentation, i.e., segmentation of road region where vehicles can drive, and (3) lane line segmentation, i.e., segmentation of lane lines on the road. Our encoder generates a shared feature pyramid network. The traffic object detection decoder branch diverges from consecutive shared feature maps. The decoders of drivable area segmentation and lane line segmentation have separate shared feature maps for elaborate representation, rather than sharing feature maps for three tasks at the same layers. The joint semantic understanding of the image is learned based on multilevel shared feature maps. The main contributions of our work can be summarized as follows: (1) We design a unified multi-task learning framework that can jointly handle three critical tasks for driving perception. (2) We propose a multi-level branch structure to share the feature map with similar tasks. (3) The proposed method achieves the superior performances on the BDD100K dataset compared to competing multi-task and single-task models.

This paper is organized as follows: Section 2 reviews the studies related to our work. In Section 3, we describe the details of the proposed framework. The experimental results are presented in Section 4. Finally, we conclude the study in Section 5.

2. Related Work

This section introduces the research on each of the three tasks and multi-task learning. Among many significant studies, we focus on deep learning-based methods.

2.1. Traffic Object Detection

Detecting traffic objects is a fundamental task in the field of vehicular vision. Recently, deep learning-based algorithms showed promising results in the object detection tasks. There are two main approaches: region proposal-based and one-step methods. The object detection in the region proposal-based method is performed in two steps. First, the region proposals are generated. Then, features from the regions are used to regress the location and classify the category. Faster-RCNN [1] is the most representative method. Despite the significant advances of R-CNN [11] and Fast-RCNN [12], there are still slow runtime problems from Selective Search [13]. The region proposal network (RPN) generates a region proposal inside the neural network. This improves the detection speed and accuracy significantly. In one-step methods, category classification and localization problems are handled as a single regression problem. Single Shot MultiBox Detector (SSD) [14] and YOLO [2] are representative one-step object detection approaches. They use a single forward pass for the recognition of objects, which is a simple yet effective approach. SSD showed outstanding performance and speed by utilizing various feature maps of the middle layer and replacing the fully connected layer with convolution operation. The improved versions based on YOLO architecture, such as YOLOv4 [15] and YOLOv5 [16], based on CSPDarknet [17] with spatial pyramid pooling [18], and YOLOX [19], which is an anchor-free version of YOLO, are continuously published. We also model the driving environments by exploiting SCPDarknet architecture for fast and accurate inference.

2.2. Drivable Area Segmentation

The great success of deep learning appears in the field of semantic segmentation. A fully convolutional network (FCN) [20] attempts to perform semantic segmentation using deep learning, which is trainable end-to-end. They used a 1×1 convolution and upsampling for segmentation. PSPNet [5] extracts various scaled features using a pyramid scene parsing network. Although significant accuracy is achieved in the deep learning-based segmentation task, inference time is still a remaining problem. Asgarian et al. [21] proposed to select special rows in the image to solve the problem of computational cost and speed. A novel decoder network, which is proposed in SegNet [6], upsamples low resolution encoder feature maps to a full input resolution feature map. A small number of trainable parameters provide good performances with competitive inference time. ErfNet [22] computes more efficiently than SegNet while providing better performance by exploiting the one-dimensional (1D) kernel and skip connections.

2.3. Lane Line Segmentation

Lane line segmentation is a basic topic for driving perception. Extensive attempts have been made to divide drivable lanes on the road. Pan et al. [3] proposed a spatial convolutional neural network (SCNN) for traffic scene understanding. Slice-by-slice convolutions effectively preserve the continuity of long thin line structure. Neven et al. [23] proposed a fast lane detection algorithm by applying a learned perspective transformation. ENet-SAD [4] used an attention distillation approach to learn itself. The low-level feature maps are learned from a high-level feature map. Rich contextual information for further representation is encoded from a reasonable level of attention map. Expanded self attention (ESA) [24] designed for segmentation based lane detection in occluded and low-light images. The ESA module predicts the confidence of the lane by extracting global contextual information. The proposed framework can infer elaborate lane lines and drivable area using a separate shared representation simultaneously.

2.4. Multi-Task Learning

Multi-task learning methods aim at a better representation through shared information from multiple objectives. Mask R-CNN [25] combines instance segmentation with object detection by adding a branch for object mask prediction to Faster R-CNN [1]. MultiNet [8] proposed a joint learning framework for simultaneous street classification, car detection, and road segmentation tasks in autonomous driving. The model is designed as an encoder–decoder architecture. Three decoders are learned for each task. DLT-Net [9] explores methods to detect drivable areas, lane lines, and traffic objects. The three most critical tasks for intelligent vehicles are handled independently in a unified network. Each task benefits from others using context tensors. Fabio et al. [26] estimates free space inside each lane by detecting navigable areas. Exploiting road type information facilitates the detection of free space without accuracy decreasing. Lee [27] proposed an efficient multi-task learning method for robust drivable area estimation with lane lines and scene classification. He uses multi-task likelihood loss [7] to light backbone networks for fast and accurate estimation of drivable areas. YOLOP [10] proposed a network to jointly handle three essential tasks for driving perception efficiently. Their work can run in real time on an embedded system while maintaining high performance.

In this paper, we propose a multi-task learning framework based on an encoder–decoder scheme for driving perception. The drivable areas, lane lines, and traffic objects are detected in real time.

3. Method

The proposed framework is composed of one encoder and three decoders. The network uses an encoder for the shared representation generation and is divided into three decoders from different layers according to their tasks; detection of traffic objects, segmentation of drivable areas, and lane lines. The multi-task learning approach optimizes the network for three objectives. The network is easily trained end-to-end. The overall architecture of the proposed network is illustrated in Figure 2.

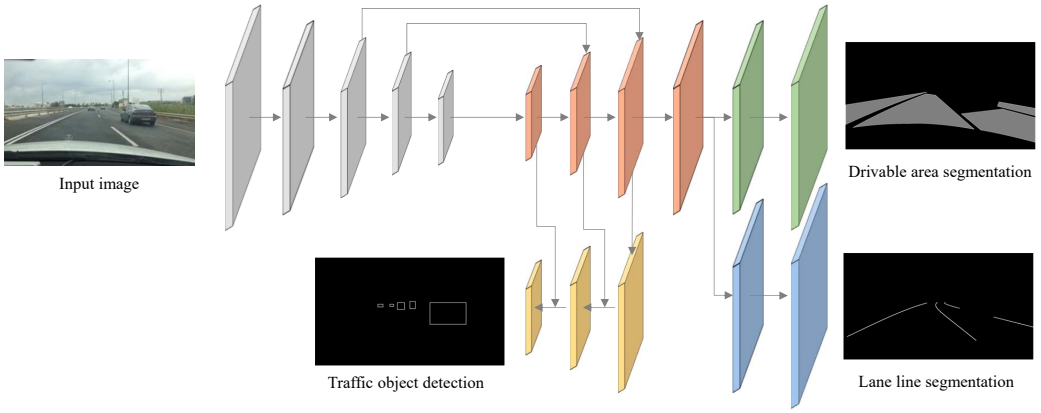

Figure 2. The overall architecture of the proposed network. The input image is fed into the encoder part, then the three decoders share the representations for different tasks at each branch.

3.1. Encoder

The encoder layers of the network learns from three different tasks to extract rich image features. We conduct our network based on the CSPDarknet [17], which successfully preserves the advantage of feature reuse characteristic of DenseNet [28] but prevents excessively duplicate gradient information. Real-time computation of network is conducted by reducing the number of parameters and calculations through feature propagation and reuse. In our encoder, we use a feature pyramid network [29] that fuses features at different semantic levels with multiple pieces of semantic information in multiple scales and a spatial pyramid pooling module [18] that generates and fuses different scale features, in order to fuse features.

3.2. Decoder

Three decoders are used in the proposed framework for the three tasks: traffic object detection, drivable area segmentation, and lane line segmentation. Each decoder exploits the shared layers from the encoder at each level according to their tasks.

3.2.1. Drivable Areas and Lane Line Segmentation

In the image of the driving environment, the drivable areas and lane lines share the bottom region of the image of the road area, whereas the traffic objects are located a little above. At the decoding stage, the first three feature maps fuse the features generated by the backbone in multiple scales. After common layers that share the traffic object detection decoder, the drivable area and lane line segmentation decoders share a separate feature map to learn the elaborate representation of the bottom region of an image. We use the same structure for the rest of the drivable area segmentation and lane line segmentation. The size of the bottom layer in the feature pyramid network to the shared feature map is $(W/8, H/8, 256)$. Each segmentation branch is simply split. The final output feature maps are generated after the upsampling of the drivable area segmentation and lane line segmentation branches have the size of $(W, H, 2)$, where the two-channel represents the probability of each pixel in the input image. We used the nearest interpolation method instead of deconvolution to reduce computational cost of the upsampling feature map.

3.2.2. Traffic Object Detection

As discussed in Section 2, the region proposal-based approach and one-step approach have their own advantages. We use multi-scale detection based on an anchor for traffic object detection to maintain computational efficiency. Similar to YOLOP [10], we use a structure of the path aggregation network [30]. Feature pyramid and path aggregation networks fuse multi-scale feature maps of semantics and positioning. To allow for predictions at multiple scales, each grid cell can be assigned with multiple anchors. The detection head predicts the scaling of the height and width, offset of position, and the corresponding probability of each category.

3.3. Loss Function and Training

Prevalent deep learning-based multi-task learning methods combine multi-objective losses. The losses for each task are calculated with weights and added as a final loss. However, these weight hyperparameters significantly affect model performance and are expensive to tune. Since there have been observations of a decrease in performances when using the same weight for multi-task learning, we learn the weight of each loss function through multi-task likelihood loss [7]. Loss L_{tod} is a weighted sum of following losses; classification L_{cl}, object L_{ol}, and bbox L_{bbl} for the traffic object detection as in Equation (1):

$$L_{tod} = \alpha_1 L_{cl} + \alpha_2 L_{ol} + \alpha_3 L_{bbl}, \tag{1}$$

where α_1, α_2, and α_3 are tuned for the balance of the detection loss. The focal loss [31] is used on L_{cl} and L_{ol} to focus on the hard samples while reducing well-classified samples. For the L_{bbl} loss calculation, we used Distance-IoU [32], which can consider overlap ratio, distance, scale similarity, and aspect ratio between ground truth and detection results.

For the drivable area segmentation, L_{das} as in Equation (2), cross-entropy with logits L_{de} is used to train the model. L_{das} is learned to minimize the classification error of drivable area segmentation:

$$L_{das} = e^{-\psi_{das}} L_{de} + \psi_{das}, \tag{2}$$

$$L_{lls} = e^{-\psi_{lls}} L_{le} + \psi_{lls}, \tag{3}$$

$$L_{iou} = e^{-\psi_{iou}} L_{ie} + \psi_{iou}, \tag{4}$$

The cross entropy with logit is also used to calculate the losses of lane line segmentation, denoted as L_{le} and L_{ie}. Loss of lane line segmentation L_{lls} as in Equation (3) is used to find classification errors of lane line segmentation. The loss function can learn a relative weighting automatically from the data. The exponential in loss functions results in smaller intraclass distances and larger inter-class distances from the high penalty on hard samples. We additionally used the IoU loss L_{iou} of the lane line as in Equation (4) for the efficiency of sparse categories of lane lines. Finally, the total loss of the three tasks in our network is defined as in Equation (5):

$$L_{total} = L_{tod} + L_{das} + L_{lls} + L_{iou}. \quad (5)$$

During training, three tasks are learned from one image at the same time. The model learns the weight to minimize a final loss. The loss for joint training is calculated as the weighted sum of all losses for traffic object detection, drivable area segmentation, and lane line segmentation.

4. Experiments

4.1. Dataset and Experimental Setting

We validated the effectiveness of the proposed method by comparing it with the state-of-the-art method on the BDD100K dataset [33]. The BDD100K dataset has been published for autonomous driving research. It includes various annotations for drivable areas, object detection, attributes, road types, and lane markings. Some specific properties for frame such as weather, scene, and time of day. Types of weather conditions are rainy, snowy, clear, overcast, partly cloudy, foggy, and undefined. The time of day includes daytime, night, dawn/dusk, and undefined. The diversity of environment and weather improve the robustness of our network when training on this dataset. The BDD100K dataset consists of 1280×720 images. A total of 100 K images are divided into three splits; 70 K for training split, 10 K for validation split, and 20 K for test split. We evaluated our method following official standards in the literature [9,10,26,27,33].

We trained the network using the training split of the BDD100K dataset. At the training stage, we used the Adam optimizer with the learning rate of 1×10^{-4}. Cosine annealing with warm-up is applied to adjust the learning rate [34]. At the training stage, the initial values of ψ of loss functions start at zero. To increase the variability of images and handle geometric distortions, we used data augmentation and transformation techniques such as translation, shearing, flipping, and random rotation. We resized the input images from $1280 \times 720 \times 3$ to $640 \times 480 \times 3$. All modules were implemented using the PyTorch framework [35] and all experiments were run on NVIDIA TITAN RTX.

4.2. Experimental Result Analysis

We showed the effectiveness of the proposed framework by comparing the performances with the most representative multi-task networks and single-task networks for intelligent vehicles. DTL-Net [9], YOLOP [10], MultiNet [8], FDAE [27], and VisLab [26] are multi-task learning based networks. IBN_PSA/P [36], Faster R-CNN [1], YOLOv5s [16], Asgarian et al. [21], PSPNet [5], ENet [37], ENet-SAD [4], and SCNN [3] are single-task based networks. In particular, DTL-Net and MultiNet are the most representative multi-task learning methods that handle driving perception tasks. We used the following metrics for the evaluation and quantitative comparison with competing methods; Recall(%) and mAP50(%) for the traffic object detection; mIoU(%) for the drivable area segmentation; accuracy(%) and IoU(%) for the lane line segmentation. We also compared the processing speed using frames per second (fps).

Compared to competing methods (Table 1), the proposed network shows the state-of-the-art drivable area segmentation performance. The proposed method achieved 92.68% of mIoU, which is higher than all competing methods. More importantly, the proposed

methods can be executed in real time with 37 fps. This is a significant advantage for driving assistance application that is safety-critical and has limited resources.

Table 1. Drivable area segmentation results on the BDD100K dataset.

Method	mIoU (%)	Speed (fps)
MultiNet [8]	71.60	8.6
DLT-Net [9]	72.10	9.3
IBN_PSA/P [36]	86.18	3.81
Asgarian et al. [21]	83.50	322
PSPNet [5]	89.60	11.1
VisLabs [26]	83.35	23.8
FDAE [27]	84.56	93.8
YOLOP [10]	91.50	41
Ours	92.68	37

Figure 3 shows the results of the proposed drivable area segmentation. We can see that the drivable areas are sufficiently well segmented in various environments. It can be observed that the opposite lane, lane line, and other vehicles are successfully excluded from the drivable area, while the free space on the road is segmented as the drivable area.

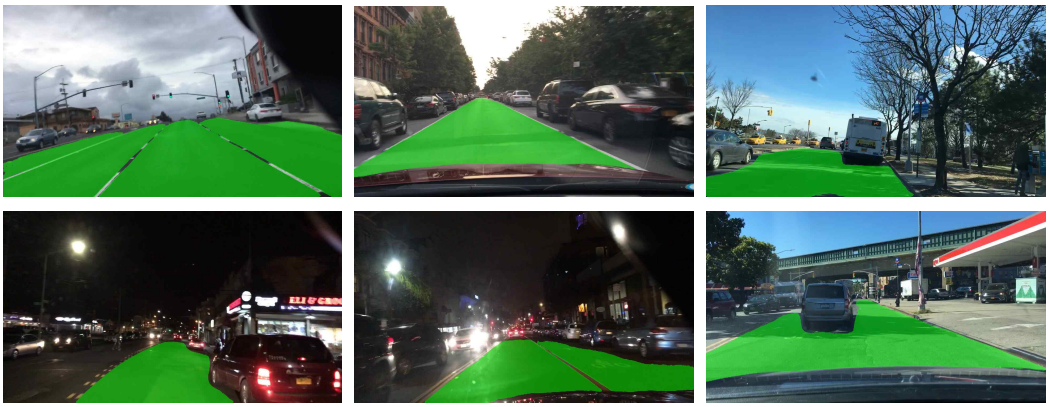

Figure 3. Examples of the drivable area segmentation results.

Table 2 compares our lane line segmentation result with other state-of-the-art approaches. The proposed method also yields the highest performance in the lane line segmentation task with 72.13% and 26.92% of accuracy and IoU score, respectively. The superior performances in drivable area segmentation and lane line segmentation show that the shared layer at the second branch in our proposed network effectively represents the common area where the road is placed. Compared to FDAE, the proposed method shows lower fps but achieves 8.12% higher mIoU in the drivable area, and 8.67% higher accuracy in the lane line segmentation task, while maintaining real-time execution. This is because the FDAE mainly focused on fast inference through global context understanding with simple scene classification using a light backbone network, whereas the proposed network enables the more elaborate prediction of the driving environment incorporating with traffic object detection. In addition, traffic object detection is a very necessary task to avoid a collision. Considering that the intelligent vehicle is a very safety-critical application, a comprehensive understanding of road areas enables the model to achieve better driving perception.

Table 2. Lane line segmentation results on the BDD100K dataset.

Method	Accuracy (%)	IoU (%)
ENet [37]	34.12	14.64
ENet-SAD [4]	36.56	16.02
SCNN [3]	35.79	15.84
FDAE [27]	63.46	21.57
YOLOP [10]	70.50	26.20
Ours	72.13	26.92

Figure 4 shows the qualitative results of lane line segmentation. The green marks represent the lane lines on the road. From the first image (left top) of Figures 3 and 4, the wide space is well separated by distinct regions where the lane lines are placed. Here, note that, following the ground truth of the BDD100K dataset, the left bottom image of Figure 4 also shows that the crosswalk is well segmented as lane lines.

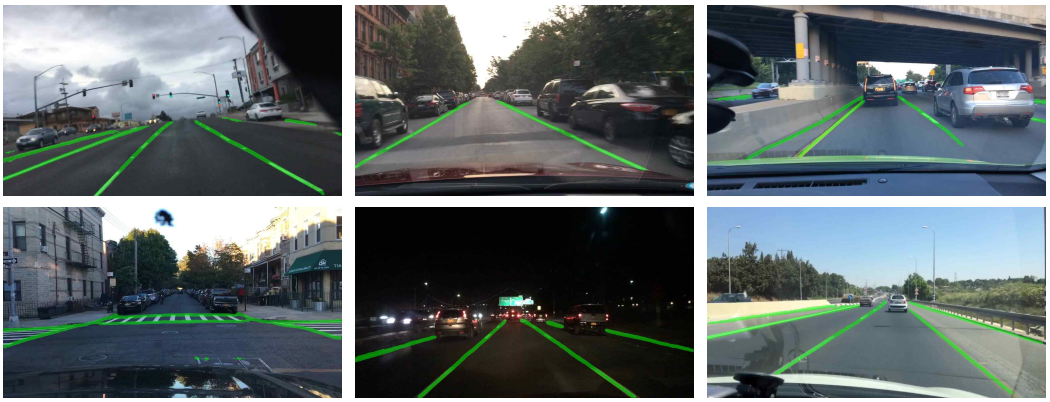

Figure 4. Examples of the lane line segmentation results.

Table 3 lists the traffic object detection performances on the BDD100K dataset. We only consider the vehicle detection, the same as previous research [8–10]. Our model outperformed MultiNet, DLT-Net, and Faster R-CNN in terms of Recall and mAP as the detection performance. However, YOLOP is slightly better than the proposed method in terms of the traffic object detection task because of its concise structure. We assumed that the reason for the decrease is the shared layer at the second branch for drivable area segmentation and lane line segmentation, which affects the optimization of the traffic object detection task. Nevertheless, the practical usage in a real-time application is very high, considering that the proposed method shows superior performance at the drivable area segmentation and lane line segmentation while very close to the state-of-the-art in traffic object detection.

Table 3. Traffic object detection results on the BDD100K dataset.

Method	Recall (%)	mAP50 (%)	Speed (fps)
MultiNet [8]	81.30	60.20	8.6
DLT-Net [9]	89.40	68.40	9.3
Faster R-CNN [1]	77.20	55.60	5.3
YOLOv5s [16]	86.80	77.20	82
YOLOP [10]	89.20	76.50	41
Ours	89.08	76.16	37

5. Conclusions

In this study, we present an efficient multi-task learning framework that is trainable end-to-end for driving perception. Our study suggests a multi-level branch network

to share the layer weight with more similar tasks. The unified architecture consists of an encoder and three decoders. The encoder encodes the input image to proper visual representation. The three decoders then decode multilevel branches to predict the optimal results for each task. The result shows that sharing the layer weight with more similar tasks is effective for the semantic understanding of the image. We design a joint training procedure to efficiently combine multiple losses. The loss function automatically learns optimal relative weight for each task during the training procedure. Compared to not only the multi-task learning methods but also single-task methods, the proposed framework showed promising results. Experimental results show the effectiveness of the proposed model that achieves excellent performance on a BDD100K dataset. The proposed network can perform in real time at 37 fps. However, the lower execution cost is better for the actual vehicle that has limited resources. In addition, considering the safety-critical characteristics of the vehicle, higher performances are required for future practical application.

Author Contributions: Conceptualization, D.-G.L. and Y.-K.K.; methodology, D.-G.L.; software, D.-G.L.; validation, D.-G.L. and Y.-K.K.; formal analysis, D.-G.L.; investigation, D.-G.L.; resources, Y.-K.K.; writing—original draft preparation, D.-G.L.; writing—review and editing, Y.-K.K.; visualization, D.-G.L.; supervision, Y.-K.K.; project administration, D.-G.L.; funding acquisition, D.-G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Kyungpook National University Research Fund, 2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The BDD100K dataset can be found at https://www.bdd100k.com/ (accessed on 7 March 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
2. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July, 2016; pp. 779–788.
3. Pan, X.; Shi, J.; Luo, P.; Wang, X.; Tang, X. Spatial as deep: Spatial cnn for traffic scene understanding. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February, 2018.
4. Hou, Y.; Ma, Z.; Liu, C.; Loy, C.C. Learning lightweight lane detection cnns by self attention distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 1013–1021.
5. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
6. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder–decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
7. Kendall, A.; Gal, Y.; Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7482–7491.
8. Teichmann, M.; Weber, M.; Zoellner, M.; Cipolla, R.; Urtasun, R. Multinet: Real-time joint semantic reasoning for autonomous driving. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1013–1020.
9. Qian, Y.; Dolan, J.M.; Yang, M. DLT-Net: Joint detection of drivable areas, lane lines, and traffic objects. *IEEE Trans. Intell. Transp. Syst. (IVS)* **2019**, *21*, 4670–4679. [CrossRef]
10. Wu, D.; Liao, M.; Zhang, W.; Wang, X. Yolop: You only look once for panoptic driving perception. *arXiv* **2021**, arXiv:2108.11250.
11. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
12. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
13. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]

14. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Amsterdam, The Netherlands, 2016; pp. 21–37.
15. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
16. Jocher, G.; Nishimura, K.; Mineeva, T.; Vilariño, R. yolov5. *Code Repository*. 2020. Available online: https://github.com/ultralytics/yolov5 (accessed on 7 March 2022).
17. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. Scaled-yolov4: Scaling cross stage partial network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13029–13038.
18. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
19. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
20. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
21. Asgarian, H.; Amirkhani, A.; Shokouhi, S.B. Fast Drivable Area Detection for Autonomous Driving with Deep Learning. In Proceedings of the 2021 5th International Conference on Pattern Recognition and Image Analysis (ICPRIA), Kashan, Iran, 28–29 April 2021; pp. 1–6.
22. Romera, E.; Alvarez, J.M.; Bergasa, L.M.; Arroyo, R. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Trans. Intell. Transp. Syst. (ITS)* **2017**, *19*, 263–272. [CrossRef]
23. Neven, D.; De Brabandere, B.; Georgoulis, S.; Proesmans, M.; Van Gool, L. Towards end-to-end lane detection: An instance segmentation approach. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 286–291.
24. Lee, M.; Lee, J.; Lee, D.; Kim, W.; Hwang, S.; Lee, S. Robust lane detection via expanded self attention. *arXiv* **2021**, arXiv:2102.07037.
25. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
26. Pizzati, F.; García, F. Enhanced free space detection in multiple lanes based on single CNN with scene identification. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 2536–2541.
27. Lee, D.G. Fast Drivable Areas Estimation with Multi-Task Learning for Real-Time Autonomous Driving Assistant. *Appl. Sci.* **2021**, *11*, 10713. [CrossRef]
28. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
29. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
30. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
31. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
32. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
33. Yu, F.; Xian, W.; Chen, Y.; Liu, F.; Liao, M.; Madhavan, V.; Darrell, T. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv* **2018**, arXiv:1805.04687.
34. Loshchilov, I.; Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv* **2016**, arXiv:1608.03983.
35. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
36. Zhao, H.; Zhang, Y.; Liu, S.; Shi, J.; Loy, C.C.; Lin, D.; Jia, J. Psanet: Point-wise spatial attention network for scene parsing. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 267–283.
37. Paszke, A.; Chaurasia, A.; Kim, S.; Culurciello, E. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv* **2016**, arXiv:1606.02147.