YOLOP: You Only Look Once for Panoptic Driving Perception

Dong Wu\(^1\)  Man-Wen Liao\(^2\)  Wei-Tian Zhang\(^2\)  Xing-Gang Wang\(^2\)
Xiang Bai\(^3\)  Wen-Qing Cheng\(^1\)  Wen-Yu Liu\(^2\)

\(^1\)Hubei Key Laboratory of Smart Internet Technology, School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China
\(^2\)School of Electronic Information and Communication, Huazhong University of Science and Technology, Wuhan 430074, China
\(^3\)School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China

Abstract: A panoptic driving perception system is an essential part of autonomous driving. A high-precision and real-time perception system can assist the vehicle in making reasonable decisions while driving. We present a panoptic driving perception network (you only look once for panoptic (YOLOP)) to perform traffic object detection, drivable area segmentation, and lane detection simultaneously. It is composed of one encoder for feature extraction and three decoders to handle the specific tasks. Our model performs extremely well on the challenging BDD100K dataset, achieving state-of-the-art on all three tasks in terms of accuracy and speed. Besides, we verify the effectiveness of our multi-task learning model for joint training via ablative studies. To our best knowledge, this is the first work that can process these three visual perception tasks simultaneously in real-time on an embedded device Jetson TX2 (23 FPS), and maintain excellent accuracy. To facilitate further research, the source codes and pre-trained models are released at https://github.com/hustvl/YOLOP.

Keywords: Driving perception, multitask learning, traffic object detection, drivable area segmentation, lane detection.

Citation: D. Wu, M. W. Liao, W. T. Zhang, X. G. Wang, X. Bai, W. Q. Cheng, W. Y. Liu. YOLOP: You only look once for panoptic driving perception. Machine Intelligence Research, vol.19, no.6, pp.550–562, 2022. http://doi.org/10.1007/s11633-022-1339-y

1 Introduction

Recently, extensive research on autonomous driving has revealed the importance of a panoptic driver perception system. It plays a significant role in autonomous driving as it can extract visual information from images taken by the camera and assist the decision system in controlling the actions of the vehicle. In order to restrict the maneuver of vehicles, the visual perception system should be able to understand the scene and then provide the decision system with information such as: Locations of the obstacles, judgements of whether the road is drivable, the position of the lanes, etc. We select three tasks: Object detection, drivable area segmentation, and lane detection, to construct our perceptual system. Object detection is essential for the vehicles to avoid obstacles and follow traffic rules. Drivable area segmentation and lane detection are also needed as they are crucial to planning the driving route of the vehicle.

For such a panoptic driving perception system, high-precision and real-time are the two most critical requirements, which are related to whether the autonomous vehicle can make accurate and timely decisions to ensure safety. However, for practical autonomous driving systems, especially the advanced driver assistance systems (ADAS), the computational resources are often marginal and limited. Therefore, it is very challenging to take both requirements into account in real-world scenarios.

Many methods handle these tasks separately. For instance, Faster R-CNN\(^1\) and YOLOv4\(^2\) deal with object detection; ENet\(^3\) and PSPNet\(^4\) are proposed to perform semantic segmentation. SCNN\(^5\) and SAD-ENet\(^6\) are used for detecting lanes. Despite the excellent performance these methods achieve, processing these tasks one after another takes longer than tackling them all at once. When deploying the panoptic driving perception system on embedded devices commonly used in self-driving cars, limited computational resources and latency should be taken into account. In addition, different tasks in traffic scenes understanding often have much related information. As shown in Fig. 1, lanes are often the boundary of the drivable area, and the drivable area usually closely surrounds the traffic objects. A multi-task network is...
more suitable in this situation as it can accelerate the image analysis process by handling multiple tasks simultaneously rather than sequentially. Also, it can share information among multiple tasks as multi-task network often shares the same feature extraction backbone. Therefore, it is essential to explore multi-task approaches in autonomous driving. Actually, DLT-Net is the first multi-task network that is able to handle the three tasks simultaneously, but does not reach the desired speed. Our work is aimed at proposing a novel network that can solve the three tasks in real-time and achieve more promising precision.

In order to solve the multi-task problem for panoptic driving perception, i.e., traffic object detection, drivable area segmentation, and lane detection, while obtaining high precision and fast speed, we designed a simple and efficient network architecture. We use a lightweight convolutional neural network (CNN) as an encoder to extract features from the image. Then these feature maps are fed to three decoders to complete their respective tasks. Our detection decoder is based on the current best-performing single-stage detection network for two main reasons: 1) The single-stage detection network is faster than the two-stage detection network; 2) The grid-based prediction mechanism of the single-stage detector is more related to the other two semantic segmentation tasks, while instance segmentation is usually combined with the region-based detector. Moreover, we verify the two viewpoints in the experiments section. The feature map output from the encoder incorporates semantic features of different levels and scales, and our segmentation branch can use these feature maps to complete pixel-wise semantic prediction excellently.

In addition to the end-to-end training strategy, we attempt some alternating optimization paradigms which train our model step-by-step. On the one hand, we can put unrelated tasks in different training steps to prevent inter-limitation. On the other hand, the task trained first can guide other tasks. So, this kind of paradigm sometimes works well though cumbersome. However, experiments show that it is unnecessary for our model as the one trained end to end can perform well enough. Our panoptic driving perception system reaches 41 FPS on a single NVIDIA TITAN XP and 23 FPS on Jetson TX2; meanwhile, it achieves state-of-the-art on the three tasks of the BDD100K dataset.

In summary, our main contributions are: 1) We put forward an efficient multi-task network that can jointly handle three crucial tasks in autonomous driving: Object detection, drivable area segmentation, and lane detection to save computational costs and reduce inference time. Our work is the first to reach real-time on embedded devices while maintaining state-of-the-art level performance on the BDD100K dataset. 2) We designed the ablative experiments to verify the effectiveness of our multitasking scheme. It is proved that the three tasks can be learned jointly without tedious alternating optimization. 3) We designed the ablative experiments to prove that the grid-based prediction mechanism of the detection task is more related to that of the semantic segmentation task, which is believed to provide a reference for other relevant multi-task learning research works.

2 Related work

In this section, we review solutions to the above three tasks respectively, and then introduce some related multi-task learning work. We only concentrate on solutions based on deep learning.

2.1 Traffic object detection

In recent years, with the rapid development of deep learning, many prominent object detection algorithms have emerged. Current mainstream object detection algorithms can be divided into two-stage methods and one-stage methods.

The two-stage methods complete the detection task in two steps. First, regional proposals are obtained, and then the features of the regional proposals are used to locate and classify the objects. The generation of regional proposals has gone through several stages of development.

The SSD-series and the YOLO-series algorithms are milestones among one-stage methods. This kind of algorithm performs the bounding box regression and object classification simultaneously. YOLO divides the picture into $S \times S$ grids instead of extracting regional proposals with the RPN network, which significantly accelerates de-
 extend Faster R-CNN by structure. Mask R-CNN \cite{he2017mask} can also achieve convolutional sharing of the network tasks. Especially, a CNN-based multi-task learning method represents through shared information among multiple representations through shared information among multiple.

2.2 Drivable area segmentation

Due to the rapid development of deep learning, a number of CNN-based methods have had great success in the semantic segmentation area, and they can be applied in drivable area segmentation tasks to provide pixel-level results. FCN\cite{long2015fully} first introduces a fully convolutional network to semantic segmentation. Despite the skip-connection refinement, its performance is still limited by low-resolution output. PSPNet\cite{zhao2017pyramid} comes up with the pyramid pooling module to extract features on various scales to enhance its performance. Besides accuracy, speed is also a key element in evaluating this task. In order to achieve real-time inference speed, ENet\cite{paszke2015enet} reduces the size of the feature maps. Recently, multi-task learning has been introduced to deal with this task, EdgeNet\cite{choy2016semantic} combines edge detection with the drivable area segmentation task to obtain more accurate segmentation results without compromising the inference speed.

3.1.1 Backbone

Our network shares one encoder, which is composed of a backbone network and a neck network.
lent performance of YOLOv4\cite{2} in object detection, we choose CSPDarknet\cite{25} as the backbone, which solves the problem of gradient duplication during optimization\cite{30}. It supports feature propagation and feature reuse, which reduces the number of parameters and calculations. Therefore, it is conducive to ensuring the real-time performance of the network.

3.2.2 Drivable area segment head & lane line segment head

The drivable area segment head and the lane line segment head adopt the same network structure. We feed the bottom layer of the FPN to the segmentation branch, with the size of (W/8, H/8, 256). After three upsampling processes, we restore the output feature map to the size of (W, H, 2), which represents the probability of each pixel in the input image for the drivable area/lane line and the background. Due to the shared SPP in the neck network, we do not add an extra SPP module to segment branches as others usually do\cite{4}, which brings no improvement to the performance of our network. Additionally, we use the nearest interpolation method in our up-sampling layer to reduce computational cost instead of deconvolution. As a result, not only do our segment decoders gain high precision output but they are also very fast during inference.

3.3 Loss function

Since there are three decoders in our network, our multi-task loss contains three parts. As for the detection loss, Ldet is a weighted sum of classification loss, object loss, and bounding box loss as in (1).

\[ L_{\text{det}} = \alpha_1 L_{\text{class}} + \alpha_2 L_{\text{obj}} + \alpha_3 L_{\text{box}} \]

where \( L_{\text{class}} \) and \( L_{\text{obj}} \) are focal loss\cite{29}, which is utilized to reduce the loss of well-classified examples, thus forcing the network to focus on the hard ones. \( L_{\text{class}} \) is used for penalizing classification and \( L_{\text{obj}} \) for the confidence of one prediction. \( L_{\text{box}} \) is \( L_{\text{1-cls}}\)\cite{29}, which takes into account distance, overlap rate, scale similarity, and aspect ratio between the predicted box and ground truth.

Both of the loss of drivable area segmentation \( L_{\text{da-seg}} \) and lane line segmentation \( L_{\text{ll-seg}} \) contain cross entropy loss with logits \( L_{\text{ce}} \), which aims to minimize the classification errors between pixels of network outputs and the targets. It is worth mentioning that intersection over union (IoU) loss: \( \mathcal{L}_{\text{IoU}} = 1 - TP/(TP + FP + FN) \) is added to \( L_{\text{ll-seg}} \) as it is especially efficient for the prediction of the sparse category of lane lines. \( L_{\text{da}} \) and \( L_{\text{ll-seg}} \) are
defined as (2) and (3), respectively.
\[ L_{\text{darseg}} = L_{\text{ce}} \]  
\[ L_{\text{lit-seg}} = L_{\text{ce}} + L_{\text{tot}}. \]

In conclusion, our final loss is a weighted sum of the three parts altogether as in (4).
\[ L_{\text{all}} = \alpha_1 L_{\text{det}} + \alpha_2 L_{\text{darseg}} + \gamma_3 L_{\text{lit-seg}} \]  
where \( \alpha_1, \alpha_2, \alpha_3, \gamma_1, \gamma_2, \gamma_3 \) can be tuned to balance all parts of the total loss.

### 3.4 Training paradigm

We attempt different paradigms to train our model. The simplest one is training end to end, and then three tasks can be learned jointly. This training paradigm is useful when all tasks are indeed related. In addition, some alternating optimization algorithms have also been tried, which train our model step by step. In each step, the model can focus on one or multiple related tasks regardless of those unrelated. Even if not all tasks are related, our model can still learn adequately about each task with this paradigm. Moreover, Algorithm 1 illustrates the process of a step-by-step training method.

**Algorithm 1.** One step-by-step training method. First, we only train the encoder and detection head. Then, we freeze the encoder and detection head as well as train two segmentation heads. Finally, the entire network is jointly trained for all three tasks.

**Require:** Target neural network \( F \) with parameter group: \( \Theta = \{\theta_{\text{enc}}, \theta_{\text{det}}, \theta_{\text{seg}}\} \); Training set: \( T \); Threshold for convergence: \( \text{thr} \); Loss function: \( L_{\text{all}} \)

**Ensure:** Well-trained network: \( F(x; \Theta) \)

1) **procedure** TRAIN(\( F, T \))
2) **repeat**
3) Sample a mini-batch \((x_s, y_s)\) from the training set \( T \).
4) \( \ell \leftarrow L_{\text{all}}(F(x_s; \Theta), y_s) \)
5) \( \Theta \leftarrow \arg\min_\Theta \ell \)
6) **until** \( \ell < \text{thr} \)
7) **end procedure**
8) \( \Theta \leftarrow \Theta \setminus \{\theta_{\text{seg}}\} \) //Freeze parameters of two Segmentation heads.*
9) TRAIN(\( F, T \))
10) \( \Theta \leftarrow \Theta \cup \{\theta_{\text{seg}}\} \setminus \{\theta_{\text{det}}, \theta_{\text{enc}}\} \) //Freeze parameters of the encoder and detection head and activate parameters of two segmentation heads.*
11) TRAIN(\( F, T \))
12) \( \Theta \leftarrow \Theta \cup \{\theta_{\text{det}}, \theta_{\text{enc}}\} \) //Activate all parameters of the neural network.*
13) TRAIN(\( F, T \))
14) **return** Trained network \( F(x; \Theta) \)

### 4 Experiments

#### 4.1 Setting

##### 4.1.1 Dataset setting

The BDD100K dataset supports the research of multi-task learning in the field of autonomous driving. With 100K frames of pictures and annotations of 10 tasks, it is the largest driving video dataset. Furthermore, as the dataset has a diversity of geography, environment, and weather, the algorithm trained on the BDD100K dataset is robust enough to migrate to a new environment. Therefore, we chose the BDD100K dataset to train and evaluate our network. The BDD100K dataset has three parts, a training set with 70K images, a validation set with 10K images, and a test set with 20K images. Since the label of the test set is not public, we evaluate our network in the validation set.

##### 4.1.2 Implementation details

In order to enhance the performance of our model, we empirically adopt some practical techniques and methods of data augmentation.

With the purpose of enabling our detector to get more prior knowledge of the objects in the traffic scene, we use the k-means clustering algorithm to obtain prior anchors from all detection frames of the dataset. We use Adam as the optimizer to train our model, and the initial learning rate, \( \beta_1 \), and \( \beta_2 \), are set at 0.001, 0.937, and 0.999, respectively. Warm-up and cosine annealing are used to adjust the learning rate during training, leading the model to converge faster and better.[31]

We use data augmentation to increase the variability of images so as to make our model robust in different environments. In addition, photometric distortions and geometric distortions are taken into consideration in our training scheme. For photometric distortions, we adjust the hue, saturation, and value of the images. We use random rotating, scaling, translating, shearing, and left-right flipping to process images to handle geometric distortions.

##### 4.1.3 Experimental setting

We select some excellent multi-task networks and networks that focus on a single task to compare with our network. Both MultiNet and DLT-Net handle multiple panoptic driving perception tasks, and they have achieved great performance in object detection and drivable area segmentation tasks in the BDD100K dataset. Faster-RCNN is an outstanding representative of the two-stage object detection network. YOLOv5 is the single-stage network that achieves state-of-the-art performance on the COCO dataset. PSPNet achieves splendid performance on semantic segmentation tasks with its superior ability to aggregate global information. We retrain the above networks on the BDD100K dataset and compare them with our network on object detection and drivable area segmentation tasks. Since there is no suitable existing
multi-task network that processes lane detection task on the BDD100K dataset, we compare our network with ENet[3], SCNN, and ENet-SAD, three advanced lane detection networks. Besides, the performance of the joint training paradigm is compared with alternating training paradigms of many kinds. Moreover, we compare the accuracy and speed of our multi-task model trained to handle multiple tasks with the one trained to perform a specific task. Furthermore, we compare the performance of the semantic segmentation task combined with the single-stage detection task and the two-stage detection task. In addition, we ablate various design choices regarding the network architecture and the methods of data augmentation. And we report the performance of our YOLOP model in all different photometric scenes. Finally, we report on how to deploy our YOLOP model to some embedded devices, such as Jetson TX2. Following [6], we resize images in BDD100K dataset from 1280×720×3 to 640×384×3. All control experiments follow the same experimental settings and evaluation metrics. Section 4.3.4 is run on NVIDIA GeForce 3090, and all the other experiments are run on NVIDIA GTX TITAN XP.

4.2 Experimental results

In this section, we train our model end-to-end and then compare it with other representative models on all three tasks.

4.2.1 Traffic object detection results

Visualization of the traffic object detection is shown in Fig.3. Since MultiNet and DLT-Net can only detect vehicles, we only consider the vehicle detection results of five models on the BDD100K dataset. As shown in Table 1, we use Recall and mAP50 as the evaluation metric of detection accuracy. Our model exceeds Faster R-CNN, MultiNet, and DLT-Net in detection accuracy and is comparable to YOLOv5s that uses more tricks than ours. Moreover, our model can infer in real-time. YOLOv5s is faster than ours because it does not have the lane line segment head and drivable area segment head.

Fig.4 shows the qualitative comparison between Faster R-CNN and YOLOP. Due to the information share of multi-task, the prediction results of YOLOP are more reasonable. For example, YOLOP will not misidentify objects far from the road as vehicles. Moreover, the examples of false negatives are much fewer, and the bounding boxes are more accurate.

4.2.2 Drivable area segmentation results

The visualization results of the drivable area segmentation can be seen in Fig.5. In this paper, both “area/drivable” and “area/alternative” classes in the BDD100K dataset are categorized as “Drivable area” without distinction. Our model only needs to distinguish the drivable area and the background in the image. mIoU is used to evaluate the segmentation performance of different models. The results are shown in Table 2. It can be seen that our model outperforms MultiNet, DLT-Net, and PSPNet by 19.9%, 19.4%, and 1.9%, respectively. Furthermore, our inference speed is 4 to 5 times faster than theirs.

The comparison between the results of PSPNet and YOLOP is shown in Fig.6. Both PSPNet and YOLOP have performed well in this task. But YOLOP is significantly better at segmenting edge areas next to vehicles or lane lines. We think it is mainly because the other two tasks provide the edge information for this task. Meanwhile, YOLOP makes fewer unexpected mistakes, such as misjudging the opposite lane area as a drivable area.

4.2.3 Lane detection results

The visualization results of the lane detection can be seen in Fig.7. The lane lines in the BDD100K dataset are labeled with two lines, so it is tricky to directly use the annotation. The experimental settings follow [6] in order to compare expediently. First, we calculate the center lines based on the two-line annotations. Then, we draw the lane line of the training with a width set to 8 pixels.

Fig. 3  Visualization of the traffic object detection results of YOLOP. Top row: Traffic objects detection results in daytime scenes. Bottom row: Traffic objects detection results in night scenes.
while keeping the lane line width of the test set at 2 pixels. We use pixel accuracy and IoU of the lanes as evaluation metrics. As shown in Table 3, the performance of our model dramatically exceeds that of the other three models.

Fig. 8 shows the comparison of lane line detection results of ENet-SAD and YOLOP. The segmentation results of YOLOP are more accurate and continuous than those of ENet-SAD. With the information shared by the other two tasks, YOLOP will not mistake some areas where some vehicles are located or drivable as lane lines, but ENet-SAD always does.

### 4.3 Ablation studies

We designed the following three ablation experiments to further illustrate the effectiveness of our scheme. All evaluation metrics in this section are consistent with the above.

#### 4.3.1 End-to-end versus step-by-step

In Table 4, we compare the performance of the joint training paradigm with alternating training paradigms of many kinds\(^1\). Our model has performed well enough through end-to-end training, so there is no need to perform alternating optimization. However, it is interesting that the paradigm training detection task firstly seems to perform better. We think it is mainly because our model is closer to a complete detection model, and the model is harder to converge when performing detection tasks. What is more, the paradigm consisting of three steps slightly outperforms that with two steps. Similar alternating training can be run for more steps, but we have observed negligible improvements.

#### 4.3.2 Multi-task versus single-task

To verify the effectiveness of our multi-task learning scheme, we compare the performance of the multi-task scheme and the single task scheme. On the one hand, we train our model to perform three tasks simultaneously. On the other hand, we train our model to perform traffic object detection, drivable area segmentation, and lane line segmentation tasks separately. Table 5 shows the comparison of the performance of these two schemes in each specific task. It can be seen that our model adopts the multi-task scheme to achieve performance that is close to that of focusing on a single task. More importantly, the multi-task model can save a lot of time compared to executing each task individually.

### 4.3.3 Region-based versus grid-based

To verify the viewpoint that the grid-based prediction mechanism is more related to the two semantic segmentation tasks than the region-based prediction mechanism. We extended the Faster R-CNN by adding two semantic segment heads to perform three tasks in parallel as our model did, and we call this new model R-CNNP. We train both YOLOP and R-CNNP 1) to perform a detection task and two segmentation tasks separately and 2) to perform three tasks simultaneously. In both of the experiments above, the two segmentation tasks are trained jointly as there is no need to consider the interaction between them. All experimental settings are the same, and the results are shown in Table 6. In the R-CNNP framework, the performance of multi-task training is much worse compared with training the detection task and semantic segmentation tasks separately. Obviously, combining two kinds of tasks conflicts in the R-CNNP framework. But, there is no such problem in our YOLOP framework; the performance of multi-task training is equal to that of focusing only on detection or semantic segmentation task. Thus, we hold the opinion that this is due to the detection head of YOLOP, like two other semantic segmentation heads, directly performing global classification or regression tasks on the whole feature map output by encoder, so they are similar and related in terms of prediction mechanism. Nevertheless, the detection head of R-CNNP needs to select region proposals first and then perform prediction on the feature maps of each proposal, which is quite different from the global prediction mechanism of semantic segmentation. In addition, R-CNNP is far behind YOLOP in terms of inference speed. Therefore, our framework is a better choice for joint training detection and segmentation tasks.

### 4.3.4 The ablations of YOLOP in different experimental settings

We ablate various design choices regarding the network architecture and the methods of data augmentation\(^2\). We report all ablation results in Table 7. Then, we compare the performance of using Resnet50\(^3\) and CSPDarknet as the backbone of the network, respectively. Included FPN, PAN, SPP modules, and CSP-Darknet backbone, our network has achieved slightly better results.

\(^1\) E, D, S and W refer to encoder, detection head, two segment heads and whole network. Therefore, Algorithm 1 can be marked as ED-S-W, and the same for the others.
ter performance as well as faster speed. Besides, we find that the object detection task benefits greatly from data augmentation. And, evidently, the FPN module is critical, which brings a huge improvement to all three tasks. Furthermore, the performance of the object detection task is improved by adding a PAN module to the detection branch. Finally, when adding a shared module to the neck of the network, the performance of all three tasks has improved steadily. However, adding two extra SPP modules to two segment branches does not improve the performance as expected and is time-consuming. Therefore, our model finally adopts the setting of the penultimate line.

4.3.5 The results of YOLOP in different photometric scenes

All photometric environments are integral parts of autonomous driving. Therefore, we report the quantitative results of our YOLOP model in different photometric scenes. As shown in Table 8, the performance of our YOLOP model is robust in different photometric scenes. Moreover, even in extremely challenging scenes, such as at night, our YOLOP model still maintains relatively steady performance.

4.4 Deployment

To ensure that our YOLOP model can actually run on edge devices, we test the performance of the entire framework on Jetson TX2, which is widely used in many real-world scenarios. First, we use a script to transfer the model to an organized binary file. Next, we re-implement
our neural network using TensorRT API in C++ and build the engine. Finally, we load the engine and run inference codes. Our YOLOP model can run at 23 FPS on Jetson TX2 with no loss of accuracy. We have released this part of the code at https://github.com/hustvl/YOLOP, and there are also many other implementations of some common deployment frameworks from third parties, such as OpenCV-DNN, ONNXRuntime, NCNN, MNN, TNN, ect. We believe that all of this will really promote the development of the autonomous-driving industry.

5 Conclusions

In this paper, we present a brand new, simple, and efficient network, which can simultaneously handle three driving perception tasks of object detection, drivable area segmentation, and lane detection and can be trained end-to-end. Our model performs exceptionally well on the challenging BDD100K dataset, achieving or greatly exceeding state-of-the-art levels on all three tasks. And it is the first to realize real-time reasoning on an embedded device Jetson TX2, which ensures that our network can be used in real-world scenarios. Moreover, we have verified that the grid-based prediction mechanism is more related to that of the semantic segmentation task, which may be of certain reference significance to similar multi-task learning research works.

Although our multi-task network can be trained end-to-end without compromising the performance of each other, we hope to improve the performance of those tasks with a more appropriate paradigm for multi-task learning. Our method is quite simple but can be more compre-
hensive in the future. For example, we can also have different object detection heads for objects of different sizes and have the lane line segment head distinguish between solid and dashed lines and so on. Furthermore, our model is limited to three tasks. More tasks related to autonomous driving perception systems, such as depth estimation, can be added to our future framework to make the whole system more complete and practical.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Nos. 61876212 and 1733007), Zhejiang Laboratory, China (No. 2019NB00AB02), and Hubei Province College Students Innovation and Entrepreneurship Training Program, China (No. S202010487058).

| Training method | Recall (%) | AP (%) | mIoU (%) | Accuracy (%) | IoU (%) |
|-----------------|------------|--------|----------|--------------|--------|
| ES-W            | 87.0       | 75.3   | 90.4     | 66.8         | 26.2   |
| ED-W            | 87.3       | 76.0   | 91.6     | 71.2         | 26.1   |
| ES-D-W          | 87.0       | 75.1   | 91.7     | 68.6         | 27.0   |
| ED-S-W          | 87.5       | 76.1   | 91.6     | 68.0         | 26.8   |
| End-to-end      | 89.2       | 76.5   | 91.5     | 70.5         | 26.2   |

Table 4 Panoptic driving perception results: The end-to-end scheme versus different step-by-step schemes

| Training method | Recall (%) | AP (%) | mIoU (%) | Accuracy (%) | IoU (%) | Speed (ms/frame) |
|-----------------|------------|--------|----------|--------------|--------|------------------|
| Det (only)      | 88.2       | 76.9   | –        | –            | –      | 15.7             |
| Da-Seg (only)   | –          | –      | 92.0     | –            | –      | 14.8             |
| Li-Seg (only)   | –          | –      | –        | 79.6         | 27.9   | 14.8             |
| Multitask       | 89.2       | 76.5   | 91.5     | 70.5         | 26.2   | 24.4             |

Table 5 Panoptic driving perception results: Multi-task learning versus single-task learning

| Training method | Recall (%) | AP (%) | mIoU (%) | Accuracy (%) | IoU (%) | Speed (ms/frame) |
|-----------------|------------|--------|----------|--------------|--------|------------------|
| Det (only)      | 79.0       | 67.3   | –        | –            | –      | –                |
| Seg (only)      | –          | –      | 90.2     | –            | 59.5   | 24.0             |
| Multi-task      | 77.2 (-1.8)| 62.6 (-4.7)| 86.8 (-3.4)| 49.8 (-9.7)| 21.5 (-2.5)| 103.3            |

Table 6 Panoptic driving perception results: Grid-based versus region-based

| Training method | Recall (%) | AP (%) | mIoU (%) | Accuracy (%) | IoU (%) | Speed (ms/frame) |
|-----------------|------------|--------|----------|--------------|--------|------------------|
| R-CNNP Det (only) | 88.2 | 76.9 | – | – | – | – |
| Seg (only)      | –          | –      | 91.6     | 69.9         | 26.5   | –                |
| Multi-task      | 89.2 (+1.0)| 76.5 (-0.4)| 91.5 (-0.1)| 70.5 (+0.6)| 26.2 (-0.3)| 24.4             |
| YOLOP Det (only) | 88.2 | 76.9 | – | – | – | – |
| Seg (only)      | –          | –      | 91.6     | 69.9         | 26.5   | –                |
| Multi-task      | 89.2 (+1.0)| 76.5 (-0.4)| 91.5 (-0.1)| 70.5 (+0.6)| 26.2 (-0.3)| 24.4             |
Table 7  Panoptic driving perception results: Different experimental settings

|    | R   | C   | D-Aug | FPN | PAN | SPP | E-SPP | Rec | AP  | mIoU | Acc  | IoU  | FPS  |
|----|-----|-----|-------|-----|-----|-----|-------|-----|-----|------|------|------|------|
| √  | 47.6| 44.1| 88.1  | 36.9| 20.4| 46.9 |
| √  | 48.0| 43.0| 87.9  | 52.5| 22.6| 52.4 |
| √  | 53.7| 49.6| 87.1  | 44.0| 19.4| 46.9 |
| √  | 52.9| 48.5| 89.8  | 50.6| 21.8| 52.4 |
| √  | 87.9| 72.6| 90.3  | 57.3| 25.4| 48.8 |
| √  | 88.0| 73.0| 89.2  | 53.3| 26.0| 54.9 |
| √  | 88.3| 74.6| 90.1  | 57.1| 25.8| 44.4 |
| √  | 89.0| 74.6| 88.9  | 56.8| 26.4| 48.5 |
| √  | 89.0| 75.7| 91.2  | 66.5| 26.2| 43.7 |
| √  | 89.2| 76.5| 91.5  | 70.5| 26.2| 47.6 |
| √  | 89.5| 76.6| 91.2  | 65.7| 26.5| 42.6 |

Table 8  Panoptic driving perception results: Different photometric scenes

| Photometric scene | Number | Recall (%) | AP (%) | mIoU (%) | Accuracy (%) | IoU (%) |
|-------------------|--------|------------|--------|----------|--------------|--------|
| Daytime           | 5 258  | 88.8       | 77.8   | 91.7     | 71.4         | 26.7   |
| Night             | 3 929  | 89.8       | 73.7   | 91.3     | 69.4         | 25.6   |
| Dawn/Dusk         | 778    | 88.9       | 77.4   | 91.1     | 71.3         | 26.5   |
| Undefined         | 35     | 94.0       | 80.5   | 86.1     | 60.8         | 22.7   |
| Total             | 10 000 | 89.2       | 76.5   | 91.5     | 70.5         | 26.2   |

Open Access

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

[1] S. Q. Ren, K. M. He, R. Girshick, J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, Canada, vol. 1, pp. 91–99, 2015.

[2] A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao. YOLOv4: Optimal speed and accuracy of object detection. DOI: 10.48550/arxiv.2004.10934, 2020.

[3] A. Paszke, A. Chaurasia, S. Kim, E. Culurciello. ENet: A deep neural network architecture for real-time semantic segmentation. DOI: 10.48550/arxiv.1606.02147, 2016.

[4] H. S. Zhao, J. P. Shi, X. J. Qi, X. G. Wang, J. Y. Jia. Pyramid scene parsing network. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Honolulu, USA, pp. 6230–6239, 2017. DOI: 10.1109/CVPR.2017.660.

[5] X. G. Pan, J. P. Shi, P. Luo, X. G. Wang, X. O. Tang. Spatial as deep: Spatial CNN for traffic scene understanding. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, USA, pp. 7276–7283, 2018. DOI: 10.1609/aaai.v32i1.12301.

[6] Y. N. Hou, Z. Ma, C. X. Liu, C. C. Loy. Learning lightweight lane detection CNNs by self attention distillation. In Proceedings of IEEE/CVF International Conference on Computer Vision, IEEE, Seoul, Korea, pp. 1013–1021, 2019. DOI: 10.1109/ICCV.2019.00110.

[7] C. Y. Wang, A. Bochkovskiy, H. Y. M. Liao. Scaled-YOLOv4: Scaling cross stage partial network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Nashville, USA, pp. 13024–13033, 2021. DOI: 10.1109/CVPR46437.2021.01283.

[8] K. M. He, G. Gkioxari, P. Dollár, R. Girshick. Mask R-CNN. In Proceedings of IEEE International Conference on Computer Vision, IEEE, Venice, Italy, pp. 2980–2988, 2017. DOI: 10.1109/ICCV.2017.322.

[9] F. Yu, W. Q. Xian, Y. Y. Chen, F. C. Liu, M. K. Liao, V. Madhavan, T. Darrell. BDD100K: A diverse driving video database with scalable annotation tooling. DOI: 10.48550/arxiv.1805.04687, 2018.
[10] R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Columbus, USA, pp. 580–587, 2014. DOI: 10.1109/CVPR.2014.81.

[11] R. Girshick. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, IEEE, Santiago, Chile, pp. 1440–1448, 2015. DOI: 10.1109/ICCV.2015.169.

[12] J. F. Dai, Y. Li, K. M. He, J. Sun. R-FCN: Object detection via region-based fully convolutional networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, pp. 379–387, 2016.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, A. C. Berg. SSD: Single shot MultiBox detector. In Proceedings of the 14th European Conference on Computer Vision, Springer, Amsterdam, The Netherlands, pp. 21–37, 2016. DOI: 10.1007/978-3-319-46448-0_2.

[14] J. Redmon, S. Divvala, R. Girshick, A. Farhadi. You only look once: Unified, real-time object detection. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Las Vegas, USA, pp. 779–788, 2016. DOI: 10.1109/CVPR.2016.91.

[15] J. Redmon, A. Farhadi. YOLO9000: Better, faster, stronger. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Honolulu, USA, pp. 6517–6525, 2017. DOI: 10.1109/CVPR.2017.690.

[16] J. Redmon, A. Farhadi. YOLOv3: An incremental improvement. [Online], Available: https://arxiv.org/abs/1804.02767, 2018.

[17] J. Long, E. Shelhamer, T. Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Boston, USA, pp. 3431–3440, 2015. DOI: 10.1109/CVPR.2015.7298965.

[18] H. Y. Han, Y. C. Chen, P. Y. Hsiao, L. C. Fu. Using channel-wise attention for deep CNN based real-time semantic segmentation with class-aware edge information. IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 2, pp. 1041–1051, 2021. DOI: 10.1109/TITS.2019.2962094.

[19] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, L. Van Gool. Towards end-to-end lane detection: An instance segmentation approach. In Proceedings of IEEE Intelligent Vehicles Symposium, IEEE, Changshu, China, pp. 286–291, 2018. DOI: 10.1109/IVS.2018.8500547.

[20] K. W. Duan, L. X. Xie, H. G. Qi, S. Bai, Q. M. Huang, Q. Tian. Location-sensitive visual recognition with cross-IOU loss. [Online], Available: https://arxiv.org/abs/2104.04999v1, 2021.

[21] M. Teichmann, M. Weber, M. Zällner, R. Cipolla, R. Urtsasun. MultiNet: Real-time joint semantic reasoning for autonomous driving. In Proceedings of IEEE Intelligent Vehicles Symposium, IEEE, Changshu, China, pp. 1015–1020, 2018. DOI: 10.1109/IVS.2018.8500504.

[22] Y. Q. Qian, J. M. Dolan, 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. DOI: 10.1109/TITS.2019.2943777.

[23] J. Zhang, Y. Xu, B. B. Ni, Z. Y. Duan. Geometric constrained joint lane segmentation and lane boundary detection. In Proceedings of the 15th European Conference on Computer Vision, Springer, Munich, Germany, pp. 502–518, 2018. DOI: 10.1007/978-3-030-01246-5_30.

[24] Z. L. Kang, K. Grauman, F. Sha. Learning with whom to share in multi-task feature learning. In Proceedings of the 28th International Conference on International Conference on Machine Learning, Bellevue, USA, pp. 521–528, 2011.

[25] C. Y. Wang, H. Y. M. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, I. H. Yeh. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, IEEE, Seattle, USA, pp. 1571–1580, 2020. DOI: 10.1109/CVPRW50498.2020.00203.

[26] K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904–1916, 2015. DOI: 10.1109/TPAMI.2015.2398924.

[27] T. Y. Lin, P. Dollar, R. Girshick, K. M. He, B. Hariharan, S. Belongie. Feature pyramid networks for object detection. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Honolulu, USA, pp. 936–944, 2017. DOI: 10.1109/CVPR.2017.106.

[28] S. Liu, L. Qi, H. F. Qin, J. P. Shi, J. Y. Jia. Path aggregation network for instance segmentation. In Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, USA, pp. 8759–8768, 2018. DOI: 10.1109/CVPR.2018.00913.

[29] T. Y. Lin, P. Goyal, R. Girshick, K. M. He, P. Dollar. Focal loss for dense object detection. In Proceedings of IEEE International Conference on Computer Vision, IEEE, Venice, Italy, pp. 2999–3007, 2017. DOI: 10.1109/ICCV.2017.324.

[30] Z. H. Zheng, P. Wang, W. Liu, J. Z. Li, R. G. Ye, D. W. Ren. Distance-IOU loss: Faster and better learning for bounding box regression. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, USA, pp. 12993–13000, 2020. DOI: 10.1609/aaai.v34i07.6999.

[31] I. Loshchilov, F. Hutter. SGDR: Stochastic gradient descent with warm restarts. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 2017.

[32] K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun. Deep residual learning for image recognition. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Las Vegas, USA, pp. 770–778, 2016. DOI: 10.1109/CVPR.2016.90.

Dong Wu is a senior undergraduate student in electronic information engineering at School of Electronics Information and Communications, Huazhong University of Science and Technology (HUST), China. His research interests include computer vision, machine learning and autonomous driving.

E-mail: riserwu@hust.edu.cn

ORCID ID: 0000-0002-2306-5709
Man-Wen Liao is a senior undergraduate student in electronic information engineering at School of Electronics Information and Communications, Huazhong University of Science and Technology, China.

His research interests include computer vision, machine learning, robotics and autonomous driving.

E-mail: mwliao@hust.edu.cn

Wei-Tian Zhang is a senior undergraduate student in electronic information engineering at Huazhong University of Science and Technology, China.

Her research interests include computer vision and machine learning.

E-mail: wtzhang@hust.edu.cn

Xing-Gang Wang received the B.Sc. and Ph.D. degrees in electronics and information engineering from Huazhong University of Science and Technology, China in 2009 and 2014, respectively. He is currently an associate professor with School of Electronic Information and Communications, HUST, China. He services as Associate Editors for Pattern Recognition and Image and Vision Computing and an editorial board member of Electronics.

His research interests include computer vision and machine learning.

E-mail: xgwang@hust.edu.cn (Corresponding author)
ORCID iD: 0000-0001-6732-7823

Xiang Bai received the B.Sc., M.Sc., and Ph.D. degrees in electronics and information engineering from Huazhong University of Science and Technology, China in 2003, 2005, and 2009, respectively. He is currently a professor with School of Artificial Intelligence and Automation, HUST, China.

His research interests include object recognition, shape analysis, and scene text recognition.

E-mail: xbai@hust.edu.cn

Wen-Qing Cheng received the B.Sc. degree in telecommunication engineering and the Ph.D. degree in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China in 1985 and 2005, respectively. She is currently a professor and associate dean with School of Electronic Information and Communications, HUST, China. Her research interests include information systems and e-learning applications.

E-mail: chengwq@hust.edu.cn

Wen-Yu Liu received the B.Sc. degree in computer science from Tsinghua University, China in 1986, and the M.Sc. and Ph.D. degrees, in electronics and information engineering from Huazhong University of Science and Technology, China in 1991 and 2001, respectively. He is now a professor of School of Electronic Information and Communications, HUST, China.

His research interests include computer vision, multimedia, and machine learning.

E-mail: liuwy@hust.edu.cn