remote sensing

MDPI

Article

Multi-Task Visual Perception for Object Detection and Semantic Segmentation in Intelligent Driving

Jiao Zhan [1], Jingnan Liu [1,*], Yejun Wu [2] and Chi Guo [1,3]

[1] The GNSS Research Center, Wuhan University, Wuhan 430079, China; zhanjiao1994@whu.edu.cn (J.Z.); guochi@whu.edu.cn (C.G.)

[2] School of Computer Science, Wuhan University, Wuhan 430079, China; wyj2016@whu.edu.cn

[3] Hubei Luojia Laboratory, Wuhan University, Wuhan 430079, China

[*] Correspondence: jnliu@whu.edu.cn

Abstract: With the rapid development of intelligent driving vehicles, multi-task visual perception based on deep learning emerges as a key technological pathway toward safe vehicle navigation in real traffic scenarios. However, due to the high-precision and high-efficiency requirements of intelligent driving vehicles in practical driving environments, multi-task visual perception remains a challenging task. Existing methods typically adopt effective multi-task learning networks to concurrently handle multiple tasks. Despite the fact that they obtain remarkable achievements, better performance can be achieved through tackling existing problems like underutilized high-resolution features and underexploited non-local contextual dependencies. In this work, we propose YOLOPv3, an efficient anchor-based multi-task visual perception network capable of handling traffic object detection, drivable area segmentation, and lane detection simultaneously. Compared to prior works, we make essential improvements. On the one hand, we propose architecture enhancements that can utilize multi-scale high-resolution features and non-local contextual dependencies for improving network performance. On the other hand, we propose optimization improvements aiming at enhancing network training, enabling our YOLOPv3 to achieve optimal performance via straightforward end-to-end training. The experimental results on the BDD100K dataset demonstrate that YOLOPv3 sets a new state of the art (SOTA): 96.9% recall and 84.3% mAP50 in traffic object detection, 93.2% mIoU in drivable area segmentation, and 88.3% accuracy and 28.0% IoU in lane detection. In addition, YOLOPv3 maintains competitive inference speed against the lightweight YOLOP. Thus, YOLOPv3 stands as a robust solution for handling multi-task visual perception problems. The code and trained models have been released on GitHub.

Keywords: intelligent driving vehicles; multi-task learning; multi-task visual perception; object detection; semantic segmentation

check for updates

Citation: Zhan, J.; Liu, J.; Wu, Y.; Guo, C. Multi-Task Visual Perception for Object Detection and Semantic Segmentation in Intelligent Driving. *Remote Sens.* **2024**, *16*, 1774. https://doi.org/10.3390/rs16101774

Academic Editor: Jon Atli Benediktsson

Received: 28 March 2024
Revised: 14 May 2024
Accepted: 15 May 2024
Published: 16 May 2024

1. Introduction

More recently, extensive research on intelligent driving vehicles has revealed the significance of environmental perception systems. As the bases for downstream tasks such as vehicle positioning, path planning and 3D reconstruction, environmental perception systems are crucial parts of intelligent driving vehicles. It facilitates a comprehensive understanding of the surrounding environment through various vehicle-mounted sensors such as LiDAR, cameras, millimeter-wave radar, and GPS/IMU. Two commonly used sensors are LiDAR and camera. LiDAR-based perception systems can effectively capture information about the distance and speed of surrounding objects, but they are expensive. In contrast, a camera-based visual perception system can capture rich semantic information from the surrounding environment at a lower cost, enabling vehicles to make optimal driving decisions. Consequently, the visual perception system finds wider application in intelligent driving vehicles.

In practice, to satisfy the safety and stability requirements of intelligent driving vehicles, researchers [1–5] commonly agree that visual perception systems should possess three basic perception capabilities: Firstly, traffic object detection aims at recognizing specific traffic objects within images and locating their positions, thus empowering vehicles to anticipate and respond to the potential risks of collision. Secondly, drivable area segmentation serves as a typical semantic segmentation task that can delineate the road regions suitable for vehicle navigation. Thirdly, lane detection seeks to locate lanes in the current vehicle driving environment, thereby providing accurate lane-keeping operations for safe driving. These capabilities jointly empower intelligent driving systems with a holistic view of the environment, contributing to safe navigation and intelligent decision-making. Over recent years, deep learning has advanced rapidly and made significant achievements. Many excellent methods can be utilized to tackle these tasks separately. For instance, SSD [6], FCOS [7] and YOLO [8] are used for object detection; U-Net [9], SegNet [10], PSPNet [11] and DeeplabV3+ [12] for performing semantic segmentation; and LaneNet [13], SCNN [14], ENet-SAD [15] and ENet [16] to detect lanes. The above approaches achieve remarkable achievements in their respective tasks. Nevertheless, due to the inherent limitation of resources of the intelligent driving system, significant delays caused by continuous image processing via multiple different models pose a significant challenge to safe driving. To tackle this challenge, researchers have introduced multi-task learning methods to speed up image processing as well as enhance network generalization via accomplishing multiple related visual tasks simultaneously. For example, YOLOP [2], built upon the lightweight one-stage detector YOLOv4, enables simultaneous traffic object detection, drivable area segmentation and lane detection. The subsequent HybridNets [3] and YOLOPv2 [4] retain the fundamental design concepts in YOLOP, and employ effective network architectures as well as training strategies for better multi-task prediction performance on the BDD100K dataset [17]. The recent YOLOPX [5] replaces the anchor-based detection head utilized in previous works with an anchor-free decoupled detection head, thus improving the flexibility and extensibility of the multi-task network. However, this anchor-free manner typically requires complex optimization strategies and large computational costs to achieve optimal performance, thus making the multi-task network harder to train. Therefore, this paper focuses on anchor-based multi-task networks and seeks to simplify training and improve the prediction performance of the networks.

For this purpose, a thorough study of existing multi-task visual perception methods is conducted. We recognize that despite the commendable performances exhibited by existing methods, they still suffer from the following drawbacks. Firstly, they fail to fully leverage multi-scale high-resolution features. For example, YOLOP [2] and YOLOPv2 [4] fail to utilize high-resolution features. HybridNets [3] and YOLOPX [5] only employ them for semantic segmentation. This manner is not conducive for the network to detect small objects that are prevalent in intelligent driving scenarios. Secondly, current anchor-based multi-task networks suffer from difficulties in capturing non-local (i.e., long-distance) contextual information. They generally adopt Path Aggregation Networks (PANs) [18], Feature Pyramid Networks (FPNs) [19], or Bidirectional Feature Pyramid Networks (BiFPNs) [20] for obtaining contextual dependencies. Nevertheless, the convolutional layer’s constrained receptive field hinders information propagation over long distances, thus impairing network performance. Thirdly, training optimization remains a significant challenge, such as HybridNets [3], which employs a stage-wise training strategy and numerous pre-defined anchors for optimal results, leading to increased computing expense. Existing anchor-based works [2–4] typically utilize a hand-crafted label assignment strategy that results in ambiguous matching between the prior anchors and the ground truth, thus impairing detection performance. The training strategy of YOLOPX [5] primarily focuses on an anchor-free multi-task network and fails to be directly applied to an anchor-based one.

To tackle these shortcomings, we propose YOLOPv3, an efficient anchor-based multi-task visual perception network, which can jointly handle object detection, drivable area segmentation and lane detection, thus saving computing expense and speeding up inference. YOLOPv3, built upon the one-stage detector YOLOv7, is a classical encoder–decoder architecture. We have made the improvements as follows: (1) We design a novel multi-task learning architecture that can effectively leverage multi-scale high-resolution features to enhance network performance in small object detection and small region segmentation. (2) We propose a lightweight self-attention (SA)-based refined module and integrate it into the lane detection head. This module can capture non-local contextual dependencies and further enhance lane detection with little computing expense. (3) We propose optimization improvements (e.g., hybrid data augmentation, model re-parameterization, dynamic label assignment, and new multi-task loss function) that can optimize network training without increasing inference expense, thus allowing the anchor-based YOLOPv3 to attain best results via straightforward end-to-end training instead of stage-wise training as in HybridNets [3]. In addition, we only utilize 12 anchors (significantly less than HybridNets which has 45 anchors), making the memory consumption acceptable for users with limited computational resources. Following [2–5], we train and evaluate YOLOPv3 on three visual tasks of the BDD100K dataset, which is a large and prevalent driving video dataset and can support the research of multi-task learning in the field of intelligent driving. The experimental results demonstrate that YOLOPv3 sets a new state of the art (SOTA): 96.9% recall and 84.3% mAP50 for traffic object detection, 93.2% mIoU for drivable area segmentation, and 88.3% accuracy and 28.0% IoU for lane detection. Compared to the lightweight networks YOLOP and HybridNets, the proposed YOLOPv3 obviously surpasses them. Furthermore, YOLOPv3 achieves an inference speed of 37 FPS, which is faster than HybridNets (17 FPS) and comparable to YOLOP (39 FPS) on the NVIDIA RTX 3080. These results indicate that YOLOPv3 is suitable for real-time operation. Compared to previous anchor-based SOTA YOLOPv2 and anchor-free SOTA YOLOPX, our YOLOPv3 demonstrates better performance with fewer parameters. Specifically, YOLOPv3 possesses 8.7 million fewer parameters than YOLOPv2 and 2.7 million fewer parameters than YOLOPX.

The main contributions can be summarized as follows:

1. We propose YOLOPv3, an efficient anchor-based multi-task visual perception network capable of simultaneously handling object detection, drivable area segmentation, and lane detection.
2. We propose architecture enhancements to utilize multi-scale high-resolution features and non-local contextual dependencies to improve network performance.
3. We propose optimization improvements aiming at enhancing network training, allowing our YOLOPv3 to achieve optimal performance via straightforward end-to-end training.
4. We empirically validate the effectiveness of our proposed method by achieving superior performance to most current SOTA methods on the BDD100K dataset.

2. Related Work

Traffic Object Detection. Object detection aims to locate and classify objects within images. Significant advancements in deep learning have established a robust foundation for its application in diverse fields such as intelligent driving vehicles [21], medical healthcare [22], agricultural robots [23], and remote sensing [24–26]. Currently, object detection methods are mainly categorized into two-stage and one-stage methods. Two-stage methods such as the R-CNN series [27–29] first obtain the regions of interest (ROIs), then predict categories and perform boundary box regression based on these ROIs. These methods perform well but are training complex and computationally intensive. In contrast, one-stage methods like SSD [6] and YOLO [8] can directly and concurrently perform object classification and bounding box regression, leading to a simpler procedure and faster processing. Recently, certain researchers have argued that anchor-based schemes restrict the performance limits of detection methods to some extent. Therefore, they advocate for anchor-free one-stage methods such as CenterNet [30], FCOS [7] and YOLOX [31], which are effective in improving the facilitation of object detection. Furthermore, the emergence of transformer-based structures has opened up new avenues for one-stage object detection.

DETR series [32,33] are typical transformer-based methods that simplify the object detection pipeline and eliminate the need for hand-crafted anchors. However, these methods are computationally intensive and may require substantial training data to achieve optimal results. Thus, it is essential to consider the speed, accuracy, and computational complexity to ensure a method meets our practical needs.

In this work, we employ the one-stage detector YOLOv7 [34], which achieves an excellent balance between accuracy and speed.

Semantic Segmentation. Semantic segmentation is essential for understanding image context at a pixel-wise level. Early methods, such as FCN [35] and SegNet [10], perform end-to-end but coarse prediction. To acquire comprehensive contextual information and promote accurate pixel-wise predictions, subsequent researchers introduce multi-scale high-resolution features into the semantic segmentation pipeline. Numerous methods leverage these features for excellent results. For instance, PSPNet [11] employs the Pyramid Pooling Module (PPM) to consolidate context information across various scales, thereby enhancing the network's capability to capture global information. Deeplab v3+ [12] introduces atrous convolutions and Atrous Spatial Pyramid Pooling (ASPP) to account for both local detail and global context, yielding remarkable achievements in accuracy and scale awareness. In addition, it introduces additional high-resolution feature maps that are 1/4 of the size of the input image into ASPP, thus improving segmentation results for small regions and producing smoother boundaries. However, the convolutional layer's constrained receptive field hinders information propagation over long distances, thus impairing network performance. To tackle this problem, researchers introduce non-local operations [36]. For example, CrossViT [37] explores the fusion of vision transformers and CNN to further improve segmentation capabilities. SETR [38] regards semantic segmentation as the prediction task of Seq2Seq and proposes a novel transformer-based architecture to capture extensive contextual information.

In this study, drivable area segmentation and lane detection are classical pixel prediction tasks, where pixel-wise binary classifications are usually performed to determine whether a pixel belongs to a drivable area or a lane line [13–16,39]. Consequently, it is very beneficial to introduce multi-scale high-resolution features and non-local contextual dependencies for improving network performance.

Multi-task Learning. Multi-task learning seeks to train a single model capable of simultaneously addressing multiple related tasks. In this model, each task branch shares information to improve the generalization of the model and accelerate convergence. In practice, the simultaneous execution of multiple related tasks by multi-task networks can efficiently leverage available resources and substantially diminish computational redundancy. This can confer significant advantages for the edge-side intelligent driving system with limited resources. Current works generally employ the encoder–decoder architectures. A good example is Mask R-CNN [29], which adds a parallel instance segmentation head based on Faster R-CNN to handle classification, object detection, and instance segmentation in a unified manner. MultiNet [40] simultaneously implements scene classification, object detection, and drivable area segmentation through a simple encoder–decoder architecture. DLT-Net [1] designs a context tensor to fuse features from multiple task branches, thus enabling the network to efficiently identify traffic objects, lane lines, and drivable areas according to extensive features. YOLOP [2], built upon the one-stage detector YOLOv4, achieves remarkable performance on three visual tasks of the BDD100K dataset. However, the simplicity of its network architecture leaves room for further improvements. HybridNets [3] enhances YOLOP by employing a more robust network architecture, automatically customized anchor technology, and a new loss function and optimization strategy. YOLOPv2 [4] retains the core design concepts of prior works [2,3], but employs more efficient network architecture and training strategy for better multi-task prediction performance. The recent YOLOPX [5] replaces the anchor-based detection head utilized in previous works with an anchor-free decoupled one, which improves the flexibility and extensibility of the network. Moreover, YOLOPX also employs a lightweight lane detection head and optimization strategy to obtain better network performance.

Although these multi-task networks demonstrate commendable performance, they still suffer from several drawbacks such as suboptimal network architecture and training optimization. Therefore, there is a need to design a new multi-task visual perception network seeking to simplify training and further enhance network performance.

3. Methods

3.1. Network Architecture

As illustrated in Figure 1, YOLOPv3, built upon the one-stage detector YOLOv7 [34], is a typical encoder–decoder architecture, where (1) the shared encoder comprises a backbone network and a neck network for extracting features and fusing features, respectively, and (2) the three task-specific decoders are employed for traffic object detection, drivable area segmentation, and lane detection, respectively.

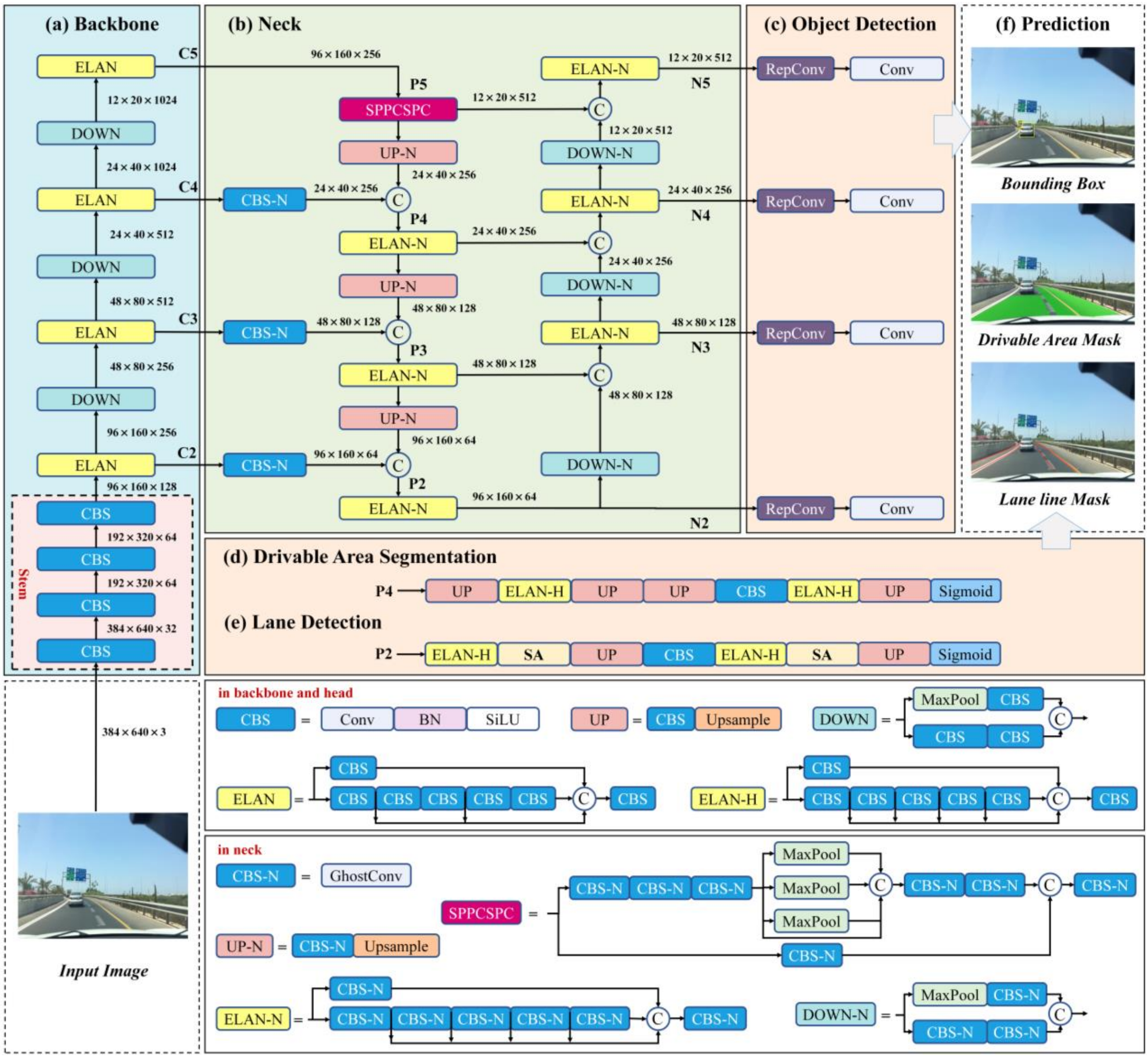

Figure 1. The network architecture of YOLOPv3. YOLOPv3 is a unified encoder–decoder network, consisting of one shared encoder (i.e., a backbone network and a neck network) and three different decoders (i.e., object detection head, drivable area segmentation head, and lane detection head). We introduce the high-resolution features C_2 generated by the backbone network into the neck network.

'CBS' and 'CBS-N' are the basic building units of YOLOPv3. 'CBS' comprises a convolutional layer, a BatchNorm layer and a SiLU activation function. 'CBS-N' consists of a ghost convolutional layer which contributes to decreasing the model parameters and computation. The detailed structures of other modules (e.g., 'ELAN', 'ELAN-H', 'SPPCSPC', and 'UP') are described in the figure below. 'C' denotes concatenation operation. 'SA' denotes an SA-based refined module that captures non-local contextual dependencies and enhances lane detection with little computation cost.

YOLOPv3 delivers image $I \in \mathbb{R}^{H \times W \times 3}$ into the backbone network and generates multiple features C_2, C_3, C_4 and C_5. The sizes of these features are 1/4, 1/8, 1/16, and 1/32 compared to the input image, respectively. We feed these features into the neck network (i.e., PAN) and generate features P_2, P_3, P_4, and P_5 through a top-down path and then generate features N_2, N_3, N_4 and N_5 through another bottom-up path (Figure 1b). Among them, N_2, N_3, N_4 and N_5 are delivered to the object detection head (Figure 1c), P_4 and P_2 are sent to two semantic segmentation heads (Figure 1d,e). Next, these task-specific decoders receive the corresponding input features and produce three different outputs. Finally, we combine these output results to generate the final multi-task visual perception outputs. For a fair comparison with prior works, we resize images in the BDD100k dataset from 720 × 1280 to 384 × 640 as in [2–5]. Therefore, we set H and W to 384 and 640, respectively, in this work.

3.1.1. Encoder

Backbone Network. The backbone network is the central component of the neural network architecture and is utilized to extract features from the input data. It has a crucial effect on network performance. Current advanced methods typically utilize classical classification networks pretrained on the ImageNet dataset as their backbone networks and achieve excellent results [41–45]. Unlike CSPDarknet [46,47] used by YOLOP, we adopt a robust backbone network employed by YOLOv7 [34], called ELAN-Net, which consists of 'Stem', 'Efficient Layer Aggregation Networks (ELAN)' and 'DOWN' modules. ELAN-Net utilizes 'ELAN' as the basic unit instead of Cross Stage Partial (CSP) networks in YOLOP. 'ELAN' can optimize the gradient length of the whole network by constructing the computing block in a stacked manner, enabling the network to learn more significant features and be more robust. The 'Stem' module consisting of four 'CBS' is utilized for initial feature extraction. 'DOWN' modules are utilized for down-sampling operations. Further details are shown in Figure 1a.

Neck Network. The neck network is responsible for further processing and integrating the features derived from the backbone network for better collaboration with the subsequent prediction heads. Prior work has proposed several different neck networks such as FPN, PAN and BiFPN. YOLOP adopts YOLOv4's PAN as the neck network to integrate different scale features via bidirectional paths (i.e., top-down path and bottom-up path). In addition, YOLOP also embeds the Spatial Pyramid Pooling (SPP) module into PAN to generate and integrate more features. To further boost the performance, we propose an improved multi-scale strategy based on YOLOv7's PAN [34]: (1) High-resolution features play a crucial role in detecting small objects and segmenting small regions. Therefore, we introduce the high-resolution features (i.e., C_2) produced by backbone with the resolution of 1/4 of the input image. (2) Instead of applying the SPP module used in YOLOP, we employ the 'SPPCSPC' module for better generation and integration of higher-level features. (3) To address the increase in computation and parameters induced by (1) and (2), we employ Ghost Convolution [48], which can efficiently generate features with cheap transformations and fewer filters. In conclusion, the above enhancements enable our neck network to effectively capture multi-scale high-resolution features while balancing prediction performance and computational cost. Further details are shown in Figure 1b. It is worth noting that our neck network utilizes 'ELAN-N' as the basic unit instead of 'ELAN' in the backbone network, because 'ELAN-N' can aggregate multi-scale features more efficiently by concatenating five different outputs. 'UP-N' and 'DOWN-N' modules are introduced for up-sampling operations and down-sampling operations, respectively, and three 'CBS-N' are utilized to change the channels of the input features.

3.1.2. Decoder

Object Detection Head. We employ an anchor-based multi-scale prediction approach for detecting different size objects on multi-scale high-resolution features $H_i \times W_i \times C_i$, where $i \in \{2,3,4,5\}$ denotes different scales. Specifically, we first set three prior anchors on each grid cell of these features. These prior anchors are automatically generated by the adaptive anchor generation strategy, where small anchors correspond to the high-resolution features and the opposite for low-resolution features. This setting is advantageous for small object detection. Then, we utilize the 'RepConv' module [49] to change the channel numbers of the input features. Moreover, this module performs re-parameterization to improve the inference efficiency of the network during inference. Finally, we employ 1×1 convolutional layers to produce the final results $H_i \times W_i \times Anchor \times (C + 4 + 1)$, where $Anchor$ denotes the number of anchors (i.e., 3), C denotes the classification probability, 4 denotes four regression offsets (i.e., the offsets of the center coordinates, width and height) relative to each anchor, and 1 denotes the object confidence. Details are shown in Figure 1c.

Drivable Area Segmentation Head. In practice, there are distinct differences in the regions of interest between the two segmentation tasks, where drivable area segmentation focuses on large region segmentation, but lane detection emphasizes small region segmentation. Instead of tackling this difference, YOLOP employs the same head network and input features to output two different segmentation results. This shared manner contributes to faster inference but may lead to mutual disruption between the two task heads, thus impairing the prediction performance of the network. To alleviate this problem, inspired by prior works [4,5,50], we adopt an input decoupling strategy to provide distinct input features for each visual task. For drivable area segmentation, the low-resolution features are adequate for effective prediction. Hence, we employ P_4 as input features and restore them to (H, W) size via a set of convolutional and up-sampling operations. Specifically, we first employ the 'UP', 'ELAN-H' and 'CBS' modules alternately to restore P_4 to (H, W) size, and then normalize the output features via a 'Sigmoid' module to output the final segmentation masks. More details are shown in Figure 1d.

Lane Detection Head. For lane detection, it is likely to lose important information if low-resolution features are utilized, as lane lines typically occupy only a relatively small region within an image. Therefore, as shown in Figure 1e, we introduce multi-scale high-resolution features P_2 as the input to capture more details. In practice, we restore P_2 to the size of $H \times W$ by utilizing 'ELAN-H', 'UP' and 'CBS' modules alternately. However, the above manner fails to capture non-local contextual dependencies and hinders information propagation over long distances. To tackle this limitation, a lightweight SA-based refined module (i.e., 'SA' module) is embedded behind each of the 'ELAN-H' modules in the lane detection head. This module is built upon the Polarized Self-Attention module [51], which can efficiently capture non-local contextual dependencies between pixels with little computing expense. Specifically, the SA-based refined module includes two main steps: (1) We first collapse the input features of size $h \times w \times c$ in channel and spatial direction while maintaining a high resolution in its counterpart direction. We then use Softmax, Sigmoid, LayerNorm and other operations to restore the features to their original sizes. (2) The outputs of the above channel-only and spatial-only self-attention are summed to produce the composition outputs. The composition outputs are then passed through BatchNorm and SiLU activation function [52] to produce the final features of size $h \times w \times c$. More details can be found in Figure 2. In addition, we do not utilize the 'SA' module in other task heads as we find that it brings no gain.

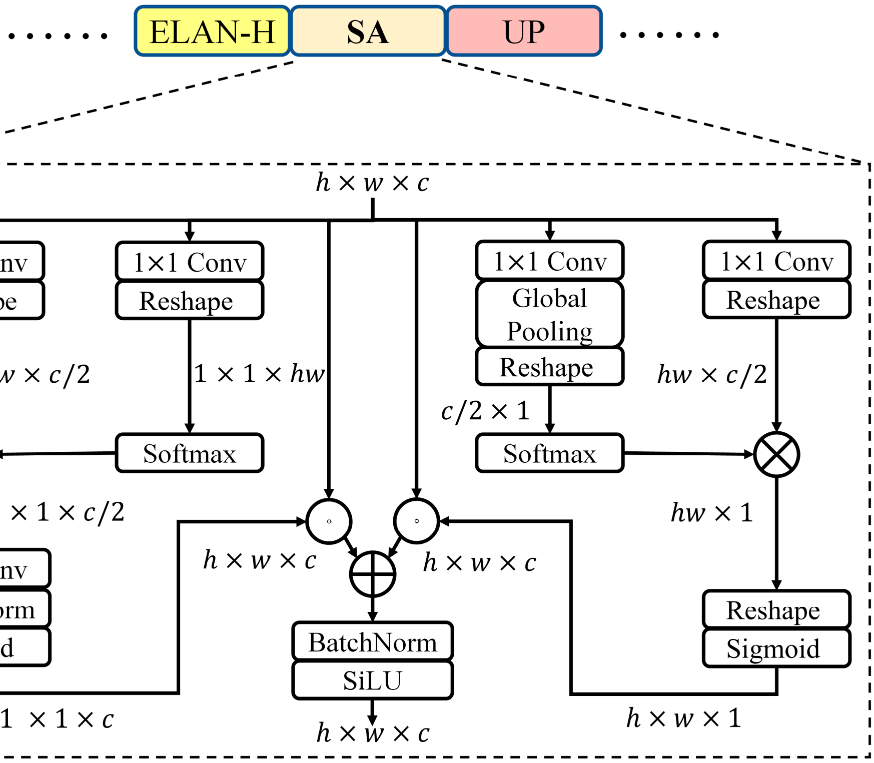

Figure 2. The architecture of SA-based refined module. ⊙ denotes the Hadamard product, ⊗ denotes the matrix product, and ⊕ denotes the matrix addition.

3.2. Training Optimization

Hybrid Data Augmentation. We employ an efficient hybrid data augmentation strategy mainly comprising Albumentations, Mosaic, and MixUp. The networks trained using this strategy exhibit outstanding generalization and robustness to adapt to various scenarios in real life. Albumentations [53] is a powerful image augmentation library with many different image transformation operations. In addition to basic operations such as flipping and rotating, we also introduce many advanced operations from Albumentations such as randomshadow and randomsunflare to further improve the robustness of the network. Mosaic and MixUp [54] are efficient augmentation strategies widely utilized in the YOLO series and other detectors.

Model Re-Parameterization. Model re-parameterization is a technique for improving the efficiency and performance of networks by merging multiple computational modules into a fully equivalent module at the inference stage [55]. Because it can reduce model complexity and improve inference speed without sacrificing accuracy, it has been extensively employed in various computer vision tasks in recent years. In this work, we introduce model re-parameterization into our network architecture. This technique divides ‘CBS’, ‘CBS-N’, and ‘RepConv’ of our network into multiple same or different branching modules during training, and integrates multiple branching modules into a fully equivalent module during inference. An example is illustrated in Figure 3. During training, ‘RepConv’ utilizes a residual structure containing multiple convolutional layers for feature extraction. During inference, ‘RepConv’ re-parameterizes various modules into a fully equivalent 3 × 3 convolutional layer to reduce computation and memory consumption without sacrificing accuracy.

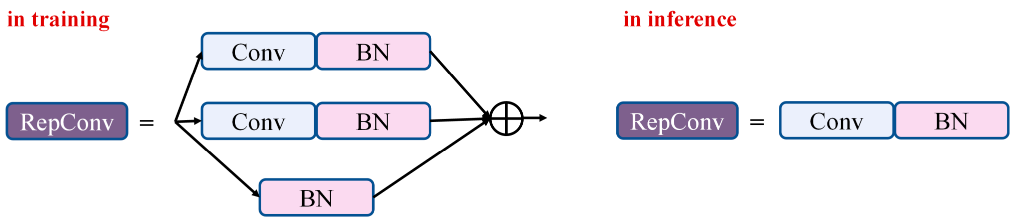

Figure 3. The architecture of 'RepConv' module. $\bigoplus$ denotes the matrix addition.

Dynamic Label Assignment. If we follow the commonly hand-crafted label assignment strategy to assign the optimal anchor to each ground truth, there may be multiple ground truths corresponding to the same anchor, which has a significantly negative impact on network training. To address the issue of fuzzy matching between anchors and ground truths, we use a dynamic label assignment strategy [34]. This strategy is mainly based on SimOTA [31], which tackles the label assignment problem from a global perspective and performs excellently among recent assignment strategies. In this work, we adopt the dynamic label assignment strategy to optimize training and enhance object detection performance.

Multi-Task Loss Function. Our multi-task loss function is the weighted sum of the detection loss $\mathcal{L}_{det}$, drivable area segmentation loss $\mathcal{L}_{da}$, and lane detection loss $\mathcal{L}_{ll}$. Detection loss $\mathcal{L}_{det}$ comprises multiple components: the classification loss $\mathcal{L}_{cls}$, regression loss $\mathcal{L}_{reg}$ and object loss $\mathcal{L}_{obj}$. Both $\mathcal{L}_{cls}$ and $\mathcal{L}_{obj}$ are Focal loss [56], which are utilized to supervise object classification and object confidence, respectively. $\mathcal{L}_{reg}$ is IoU loss, which reflects the ratio of overlap between the predicted box and the ground truth. $\mathcal{L}_{det}$ is defined as follows:

$$\mathcal{L}_{det} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{reg}\mathcal{L}_{reg} + \lambda_{obj}\mathcal{L}_{obj} \tag{1}$$

where λ_{cls}, λ_{obj}, and λ_{reg} are hyper-parameters to balance three parts of $\mathcal{L}_{det}$, which are set to 0.3, 0.7 and 0.05, respectively, as in YOLOv7.

$\mathcal{L}_{da}$ is the Binary Cross-Entropy (BCE) loss frequently utilized in semantic segmentation. For lane detection loss, we adopt a combination of Focal loss $\mathcal{L}_{focal}$ with $\alpha = 0.25$, $\gamma = 2.0$ and Tversky loss $\mathcal{L}_{tv}$ [57] with $\alpha = 0.7$, $\beta = 0.3$ as in YOLOPX [5]. $\mathcal{L}_{focal}$ is utilized to alleviate class imbalance, as positive samples tend to be significantly fewer than negative samples in lane detection. We utilize $\mathcal{L}_{tv}$ to penalize false positives, as our method tends to produce more false positives. $\mathcal{L}_{ll}$ is formulated below:

$$\mathcal{L}_{ll} = \mathcal{L}_{focal} + \mathcal{L}_{tv} \tag{2}$$

$$\mathcal{L}_{focal} = -\frac{1}{N}\sum_{c=0}^{c-1}\sum_{n=1}^{N} t_n(c)(1 - p_n(c))^{\gamma} log(p_n(c)) \tag{3}$$

$$\mathcal{L}_{tv} = C - \sum_{c=0}^{c-1} \frac{TP_p(c)}{TP_p(c) + \alpha FP_p(c) + \beta FN_p(c)} \tag{4}$$

$$TP_p(c) = \sum_{n=1}^{N} p_n(c)t_n(c) \tag{5}$$

$$FN_p(c) = \sum_{n=1}^{N} (1 - p_n(c))t_n(c) \tag{6}$$

$$FP_p(c) = \sum_{n=1}^{N} p_n(c)(1 - t_n(c)) \tag{7}$$

where C represents the total number of classes, while N stands for the total number of pixels in the input image. $p_n(c)$ indicates the probability that pixel n belongs to class c, and

$t_n(c)$ represents the corresponding ground truth. $TP_p(c)$, $FN_p(c)$, and $FP_p(c)$ indicates true positives, false negatives and false positives, respectively.

The overall loss function for our YOLOPv3 is defined below:

$$\mathcal{L}_{all} = \lambda_{det}\mathcal{L}_{det} + \lambda_{da}\mathcal{L}_{da} + \lambda_{ll}\mathcal{L}_{ll} \tag{8}$$

where λ_{det}, λ_{da}, and λ_{ll} are hyper-parameters. We tune both λ_{da}, and λ_{ll} to be 0.2 as in YOLOP and YOLOPX. We set λ_{det} to 0.75 through extensive experiments.

4. Experiments

4.1. Experimental Settings

Dataset. We perform experiments on the BDD100K dataset [16]. It is a large and challenging autonomous driving dataset, which contains images under different weathers (e.g., sunny, rainy, cloudy, foggy), timings (e.g., daytime, night-time) and driving scenarios (e.g., urban roads, highways, rural roads). Therefore, the model trained on the BDD100K dataset has excellent robustness and generalization. The BDD100K dataset contains a training set, a validation set and a test set, with 70,000 images for the training set, 20,000 images for the validation set, and 10,000 images for the test set. For a fair comparison with prior works, we follow the common practice [2–5,15]: (1) We resize the original image to the size of (384, 640). (2) We train the network on the training set and evaluate it on the validation set. (3) We combine four object detection classes (bus, train, truck, car) into one (vehicle), and two drivable area classes (alternative, direct) into one (drivable). (4) We convert the lane lines, which are marked with two edge lines in the BDD100K dataset, to centerlines. Meanwhile, we set the centerline width to 8 pixels in the training set, and 2 pixels in the validation set.

Evaluation Metrics. Following common practice [2–5], we adopt different evaluation metrics for different visual tasks. For object detection, we utilize Recall and mAP50 as the evaluation metrics. For drivable area segmentation, we employ mIoU as the evaluation metric. For lane detection, we adopt Pixel Accuracy and IoU as the evaluation metrics.

Implementation Details. We conduct experiments utilizing the Pytorch framework. YOLOPv3 is built upon the excellent one-stage detector YOLOv7. YOLOPv3 is trained on the NVIDIA Tesla V100 (Sourced by NVIDIA Corporation in Santa Clara, CA, USA) and adopts AdamW as the optimizer. YOLOPv3 is trained for 200 epochs. Meanwhile, to guarantee training stability, we adopt the warm-up strategy with three epochs. The initial learning rate is set to 0.001, and the batch size is set to 32. During training, we also employ cosine annealing to tune the learning rate and set the momentum to 0.937. For anchor-based object detection, we adopt the K-means method to generate 12 different anchors from the BDD100K dataset after 1000 iterations. The code and trained models are available on https://github.com/jiaoZ7688/YOLOPv3 (accessed on 2 March 2024).

4.2. Experimental Results

To illustrate the validity of YOLOPv3, we compare it to several outstanding multi-task networks and those focusing on individual tasks on the BDD100k dataset. YOLOv5s and Faster R-CNN are excellent object detection networks. PSPNet exhibits good performance in segmentation tasks. SCNN, ENet and ENet-SAD excel in lane detection. MultiNet and DLT-Net can simultaneously perform object detection and drivable area segmentation. The previous multi-task visual perception networks YOLOP, HybridNets, YOLOPv2, and YOLOPX demonstrate significant results on three visual tasks of the challenging BDD100K dataset. The primary experimental results are as follows.

4.2.1. Computational Cost

We conduct a comparison of the computational cost of YOLOP, HybridNets, and YOLOPv3 on the NVIDIA RTX 3080 (Sourced by NVIDIA Corporation in Santa Clara, CA, USA). As illustrated in Table 1, YOLOPv3 possesses more parameters compared to the other two networks, since it utilizes a more powerful ELAN-based encoder (including a backbone network and a neck network). Thanks to the efficient compositional modules (e.g., 'ELAN', 'ELAN-N', and 'SPPCSPC') and effective optimization tricks (e.g., model re-parameterization), our ELAN-based encoder can effectively extract and integrate multi-scale high-resolution features while maintaining a balance between accuracy and computational cost. Therefore, YOLOPv3 possesses excellent inference speed. It is worth noting that YOLOPv3 exhibits faster inference speed with significantly larger parameters than HybridNets. This is because depthwise convolution, which is widely utilized in HybridNets, brings more computational processes and intermediate features, thus leading to higher memory access costs (MACs). Compared to parameters, higher MACs impair the inference speed of the network more. Overall, the above results indicate that YOLOPv3 is an efficient solution for accomplishing multi-task visual perception in real time.

Table 1. The comparison results of the computational cost. Speed indicates the inference speed at batch size 1 (excluding data preprocessing and NMS). MRP indicates the model re-parameterization technique. The best scores are in bold.

Method	Anchor	MRP	Backbone	Input Size	Params	Speed (FPS)
YOLOP	√	×	CSPDarknet	384 × 640	**7.9 M**	**39**
HybridNets	√	×	EfficientNet	384 × 640	12.8 M	17
YOLOPv3	√	×	ELAN-Net	384 × 640	30.9 M	26
YOLOPv3	√	√	ELAN-Net	384 × 640	30.2 M	37

4.2.2. Traffic Object Detection

As shown in Table 2, YOLOPv3 achieves the best performance of 96.9% recall and 84.3% mAP50 in traffic object detection, which exceeds YOLOP by 7.7% recall and 7.8% mAP50, exceeds YOLOPv2 by 5.8% recall and 0.9% mAP50, and exceeds YOLOPX by 3.2% recall and 1.0% mAP50. In general, YOLOPv3 sets the new SOTA for traffic object detection.

Table 2. The experimental results on traffic object detection. The confidence threshold and NMS threshold are set to 0.001 and 0.6, respectively. The best scores are in bold.

Method	Recall	mAP50
MultiNet	81.3	60.2
DLT-Net	89.4	68.4
Faster R-CNN	77.2	55.6
YOLOv5s	86.8	77.2
YOLOP	89.2	76.5
HybridNets	92.8	77.3
YOLOPv2	91.1	83.4
YOLOPX	93.7	83.3
YOLOPv3	**96.9**	**84.3**

4.2.3. Drivable Area Segmentation

The experimental results presented in Table 3 indicate that YOLOPv3 realizes 93.2% mIoU SOTA performance in drivable area segmentation.

Table 3. The experimental results on drivable area segmentation. The best scores are in bold.

Method	mIoU
MultiNet	71.6
DLT-Net	71.3
PSPNet	89.6
YOLOP	91.5
HybridNets	90.5
YOLOPv2	**93.2**
YOLOPX	**93.2**
YOLOPv3	**93.2**

4.2.4. Lane Detection

Following common practice [2–5,15], we set the width of the lane lines to 8 pixels in the training set and 2 pixels in the validation set. This contributes to simplifying training, but will result in predicted lane lines being significantly wider than the ground truth. As a result, the evaluation metric IoU commonly remains low. Conversely, pixel accuracy directly determines if the lane lines are detected correctly or continuously [5], and thus better reflects the performance of lane detection. Therefore, we emphasize pixel accuracy in this work.

The experimental results presented in Table 4, YOLOPv3 achieves pixel accuracy of 88.3%, which outperforms YOLOP by 17.8%, HybridNets by 2.9%, YOLOPv2 by 1.0%, and competitive with YOLOPX. For IoU, our YOLOPv3 outperforms YOLOP, YOLOPv2 and YOLOPX, but slightly underperforms HybridNets. This is due to our YOLOPv3 being trained in an end-to-end manner rather than a stage-wise manner like HybridNets for optimal performance and the lane lines predicted by our YOLOPv3 being slightly wider. However, as shown in the comparison chart of lane detection in the following, our segmentation results obviously exhibit better accuracy and continuity because of the higher pixel accuracy. Overall, the general performance of YOLOPv3 is better than the previous SOTA YOLOPX in lane detection.

Table 4. The experimental results on lane detection. The best scores are in bold.

Method	Pixel Accuracy	IoU
ENet	34.1	14.6
SCNN	35.8	15.8
ENet-SAD	36.6	16.0
YOLOP	70.5	26.2
HybridNets	85.4	**31.6**
YOLOPv2	87.3	27.2
YOLOPX	**88.6**	27.2
YOLOPv3	88.3	28.0

4.3. Analysis of Results

Compared with Baseline. We conduct a comparison between YOLOPv3 and the baseline YOLOP on the BDD100K dataset. We can draw several important conclusions from Table 5: (1) Both YOLOPv3 and YOLOP train the network from scratch, rather than utilizing the pre-trained models for fine-tuning. (2) The robust ELAN-based encoder adopted by YOLOPv3 results in more parameters than the lightweight CSP-based encoder adopted by YOLOP. (3) Due to the utilization of data augmentation ('DA' for short later), YOLOPv3 demonstrates better robustness and generalization. (4) Our YOLOPv3 significantly outperforms YOLOP on the three visual tasks of the BDD100k dataset. In traffic object detection, YOLOPv3 surpasses YOLOP by 7.7% recall and 7.8% mAP50. In drivable area segmentation, YOLOPv3 is 1.7% higher than YOLOP in terms of mIoU. In lane detection, YOLOPv3 is 17.8% and 1.8% better than YOLOP in terms of accuracy (i.e., pixel accuracy) and IoU, respectively. This significant improvement in network performance is mainly attributed to our proposed architecture enhancements (e.g., the utilization of the robust multi-task learning architecture and non-local contextual dependencies) and optimization improvements (e.g., the utilization of the data augmentation and dynamic label assignment). Overall, our method makes a significant improvement over the previous baseline YOLOP but requires more network parameters.

Table 5. The comparison results with baseline. DA denotes data augmentation. The best scores are in bold.

Method	Anchor	Fine-Tuning	DA	Recall	mAP50	mIOU	Accuracy	IoU	Params
YOLOP	√	×	×	89.2	76.5	91.5	70.5	26.2	**7.9 M**
YOLOPv3	√	×	√	**96.9 (+7.7)**	**84.3 (+7.8)**	**93.2 (+1.7)**	**88.3 (+17.8)**	**28.0 (+1.8)**	30.2 M (+22.3 M)

Compared with Anchor-Based SOTA. As depicted in Table 6, we compare YOLOPv3 with YOLOPv2 in several aspects: (1) We train YOLOPv3 from scratch instead of employing pre-trained models to fine-tune it as in YOLOPv2. This manner better highlights the validity of YOLOPv3. (2) YOLOPv3 constructs the network employing the simple ELAN instead of the complicated Extended ELAN (i.e., E-ELAN) [33] like YOLOPv2, resulting in fewer parameters. (3) Both YOLOPv3 and YOLOPv2 exhibit good robustness and generalizability due to using 'DA'. (4) In traffic object detection, YOLOPv3 exceeds YOLOPv2 by 5.8% recall and 0.9% mAP50. This superior performance is mainly attributed to the leveraging of high-resolution features as well as new optimization strategies (e.g., the dynamic label assignment strategy), which enable our network to efficiently detect objects with various shapes. (5) In drivable area segmentation, YOLOPv3 and YOLOPv2 perform comparably thanks to adopting an approximate input decoupling strategy. (6) In lane detection, YOLOPv3 has 1.0% accuracy and 0.8% IoU higher than YOLOPv2. By utilizing the multi-scale high-resolution features and SA-based refined modules, YOLOPv3 enables better detection based on extensive context information, thus providing a significant advantage in terms of accuracy and IoU. In general, YOLOPv3 yields better results without utilizing pre-trained models and with fewer parameters compared to YOLOPv2.

Table 6. The comparison results with anchor-based SOTA. DA denotes data augmentation. The best scores are in bold.

Method	Anchor	Fine-Tuning	DA	Recall	mAP50	mIOU	Accuracy	IoU	Params
YOLOPv2	√	√	√	91.1	83.4	93.2	87.3	27.2	38.9 M
YOLOPv3	√	×	√	**96.9 (+5.8)**	**84.3 (+0.9)**	**93.2 (+0.0)**	**88.3 (+1.0)**	**28.0 (+0.8)**	**30.2 M (−8.7 M)**

Compared with Anchor-Free SOTA. As shown in Table 7, we compare YOLOPv3 and YOLOPX in several aspects: (1) YOLOPv3 and YOLOPX train the network from scratch, rather than utilizing the pre-trained models for fine-tuning. Moreover, both YOLOPv3 and YOLOPX utilize 'DA' during training for excellent robustness and generalization. (2) YOLOPX is an anchor-free method that locates an object by directly predicting its center coordinates and bounding box size, without the need for pre-defined anchors. Compared to the anchor-based YOLOPv3, this anchor-free manner typically requires a more complex detection head structure, thus making YOLOPX larger than YOLOPv3 in terms of parameters. Moreover, YOLOPX produces significantly more positive samples that are responsible for predicting objects during training, thus requiring complex optimization strategies and large computational costs to achieve optimal performance. In contrast, the anchor-based YOLOPv3 is easier to train and requires significantly less training time (half of YOLOPX) for better performance. (3) In traffic object detection, YOLOPv3 is 3.2% recall and 1.0% mAP50 higher than YOLOPX. Besides the utilization of anchors, this excellent performance is also attributed to the utilization of high-resolution features, enabling YOLOPv3 to perform more detailed detection, especially for small objects. (4) For drivable area segmentation, both networks perform comparably. (5) In lane detection, YOLOPv3 exceeds YOLOPX by 0.8% in terms of IoU and is competitive with YOLOPX in terms of pixel accuracy. Meanwhile, our lane detection head is more lightweight than YOLOPX, requiring only a single input feature and two up-sampling operations (YOLOPX requires multiple input features and three up-sampling operations) to achieve accurate prediction. This demonstrates the effectiveness of the multi-scale high-resolution features and SA-based refined modules utilized by our YOLOPv3. In conclusion, compared to YOLOPX, YOLOPv3 realizes better performance with fewer network parameters.

Table 7. The comparison results with anchor-free SOTA. DA denotes data augmentation. The best scores are in bold.

Method	Anchor	Fine-Tuning	DA	Recall	mAP50	mIOU	Accuracy	IoU	Params
YOLOPX	×	×	√	93.7	83.3	93.2	**88.6**	27.2	32.9 M
YOLOPv3	√	×	√	**96.9 (+3.2)**	**84.3 (+1.0)**	**93.2 (+0.0)**	88.3 (−0.3)	**28.0 (+0.8)**	**30.2 M (−2.7 M)**

Qualitative Comparison. We thoroughly compare our YOLOPv3 with the current open-source YOLOP and HybridNets on the validation set of the BDD100K dataset. For better visualization of the ground truth, the lane line width in the validation set is expanded from 2 pixels to 8 pixels. For object detection, the confidence threshold and NMS (Non-Maximum Suppression) threshold are set to 0.3 and 0.45, respectively. The visualization results in daytime and night-time are shown in Figures 4–7: (1) Figure 4 presents multi-task prediction results. It can be seen clearly from the comparison results that YOLOPv3 implements better predictions, especially for two segmentation tasks. (2) Figure 5 presents object detection results. YOLOP and HybridNets have significant false negatives (i.e., missed detection) in columns 1, 2, and 4, and false positives (i.e., error detection) in column 3. In contrast, thanks to the multi-scale high-resolution features and training optimizations, YOLOPv3 can accurately predict objects of diverse shapes. (3) Figure 6 presents drivable area segmentation results. Our results demonstrate higher accuracy and better boundary localization, while significant false positives and false negatives are observed in the other two methods. (4) Figure 7 presents lane detection results. Obvious discontinuity and incorrectness are observed in YOLOP and HybridNets. Conversely, YOLOPv3 can utilize comprehensive contextual information for accurate and continuous predictions. In conclusion, these comparison results collectively confirm the excellent performance of YOLOPv3 in different visual tasks.

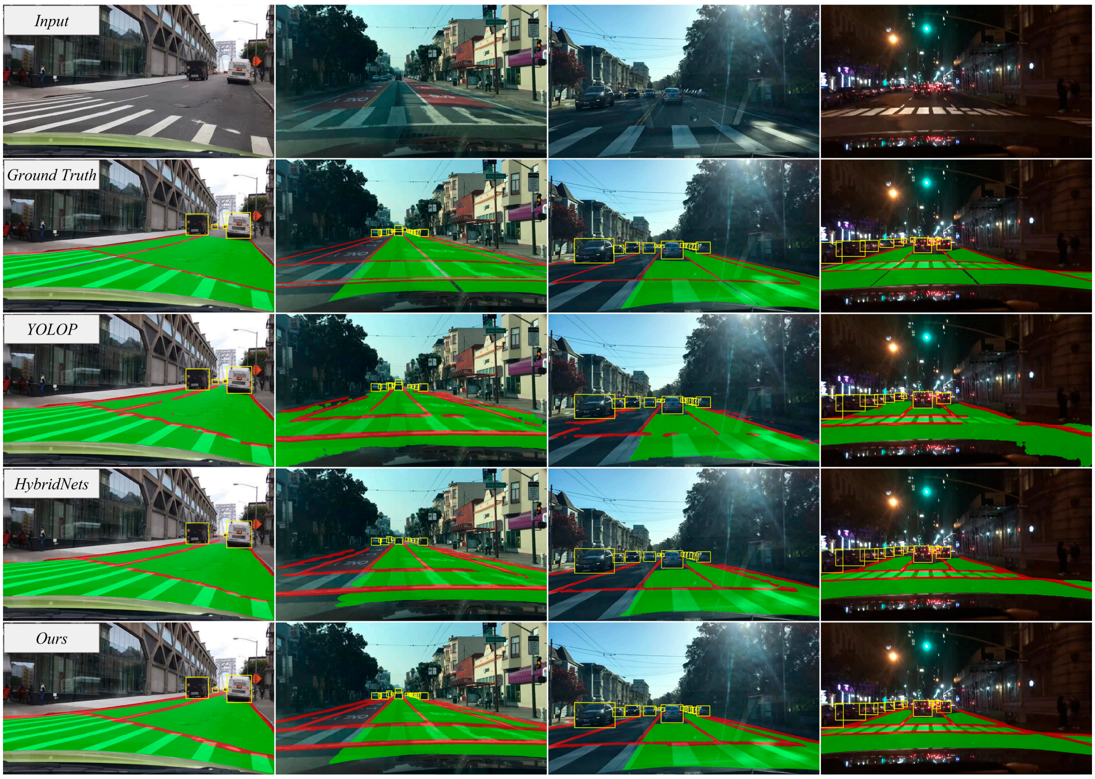

Figure 4. The comparison chart of multi-task predictions. The rows from top to bottom are the input images, the ground truth, and the predictions of YOLOP, HybridNets and ours, respectively. Yellow boxes indicate traffic objects. Green areas are drivable areas. Red lines indicate lane lines.

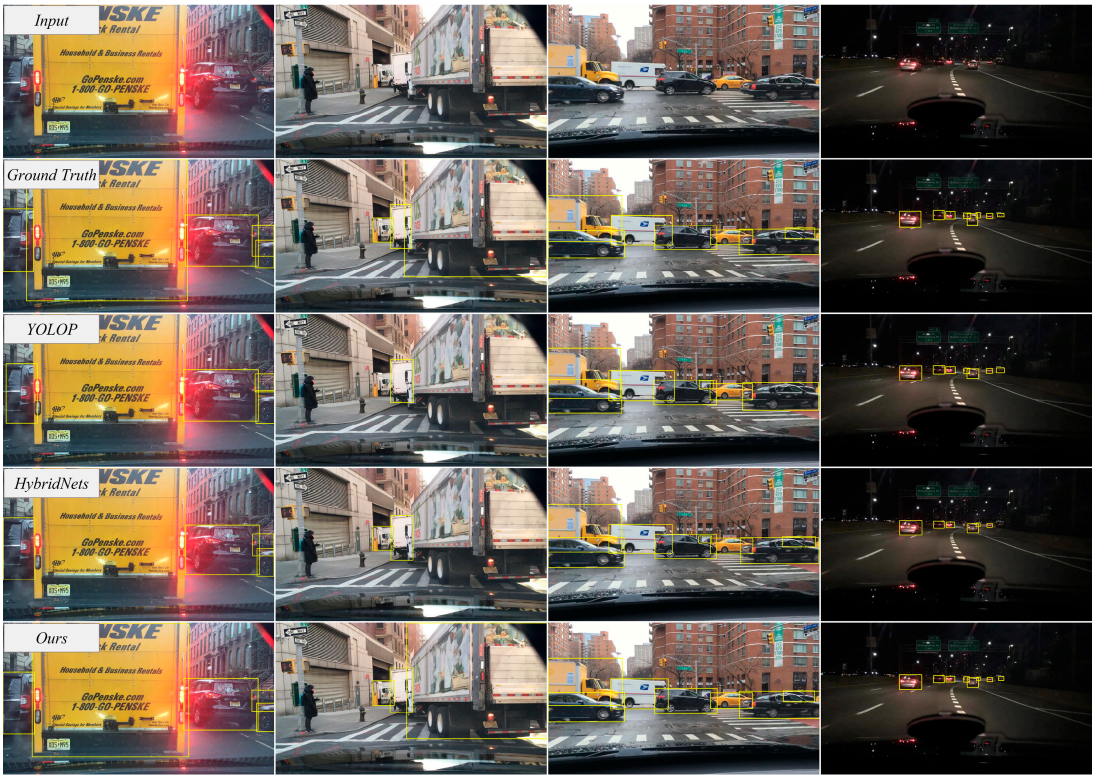

Figure 5. The comparison chart of traffic object detection. The rows from top to bottom are the input images, the ground truth, and the predictions of YOLOP, HybridNets and ours, respectively. The yellow boxes indicate the traffic objects.

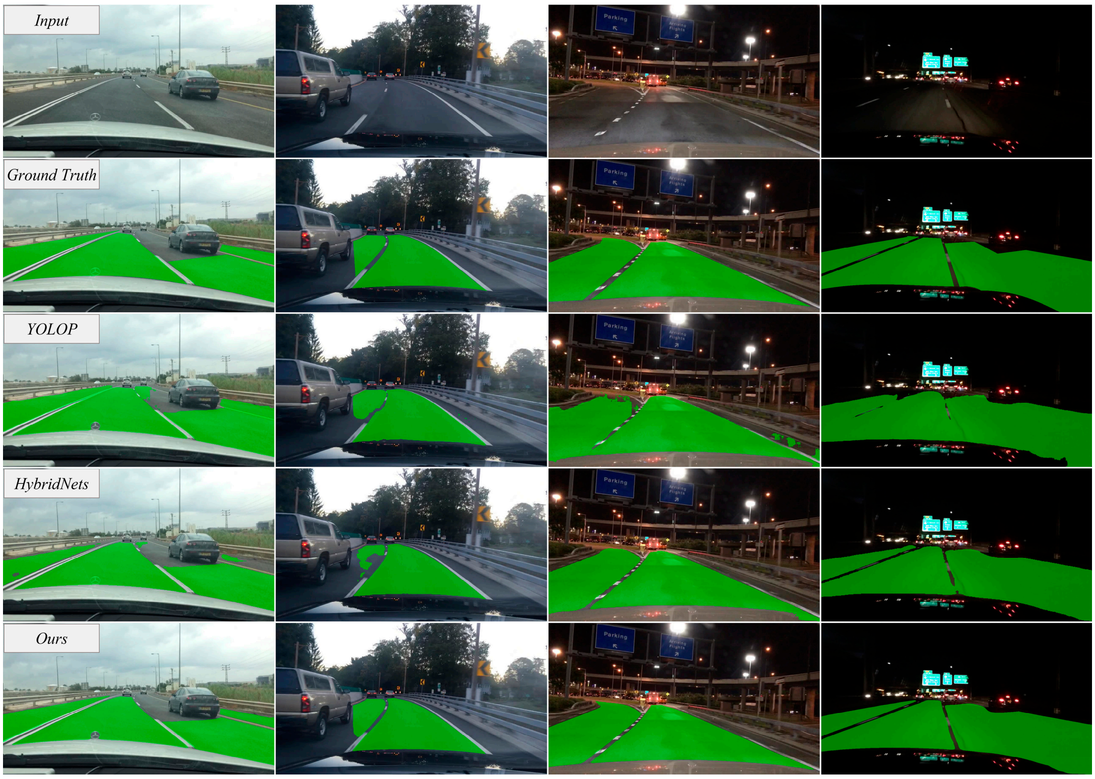

Figure 6. The comparison chart of drivable area segmentation. The rows from top to bottom are the input images, the ground truth, and the predictions of YOLOP, HybridNets and ours, respectively. The green areas are the drivable areas.

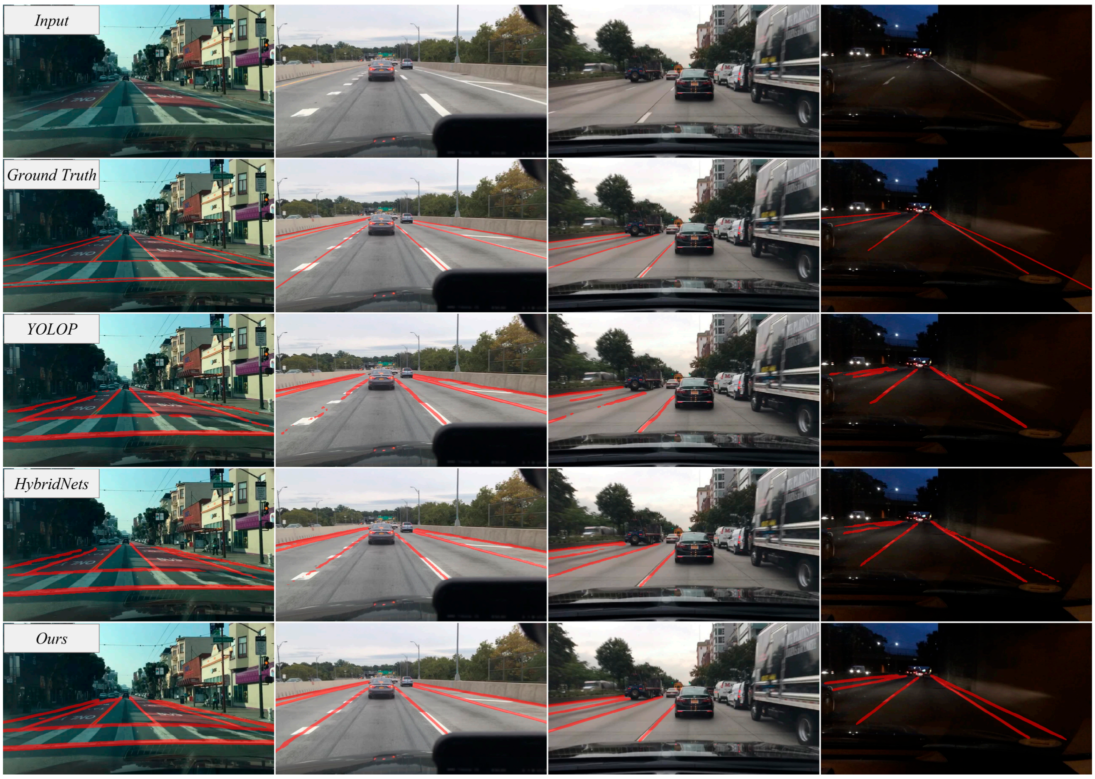

Figure 7. The comparison chart of lane detection. The rows from top to bottom are the input images, the ground truth, and the predictions of YOLOP, HybridNets and ours, respectively. The red lines show the lane lines.

4.4. Ablation Study

4.4.1. Multi-Task vs. Single-Task

We make a comparison between the prediction performance of the multi-task solution and the single-task solution using the BDD100K dataset. First, we train the network for traffic object detection (Det), drivable area segmentation (Da), and lane detection (Ld) independently. Second, we train the network to simultaneously perform multiple tasks (Multi-task). The comparison results are listed in Table 8. The prediction performance of the network with the multi-task solution approaches that of the network focusing only on a single task. In addition, we also compare the inference speed in various manners on the NVIDIA RTX 3080 (the size of the input image is set to 384 × 640). As we can see from the results, this multi-task manner enables significant time savings compared to executing each task separately. Overall, our multi-task solution is effective and efficient.

Table 8. The prediction results of multi-task solutions and single-task solutions.

Method	Recall	mAP50	mIOU	Accuracy	IoU	Speed (ms/Frame)
Det (only)	97.1	83.7	-	-	-	18.87
Da (only)	-	-	93.5	-	-	16.13
Ld (only)	-	-	-	89.1	28.4	16.67
Multi-task	96.9	84.3	93.2	88.3	28.0	27.03

4.4.2. Optimization Tricks

We conduct thorough ablation studies of the different optimization tricks employed in YOLOPv3. The baseline is YOLOP, which consists of a CSPDarknet, a PAN integrated with SPP, and three task-specific heads. Table 9 lists the results of the ablation study, where ‘ELAN’ refers to YOLOv7’s encoder comprising ELAN-Net and PAN integrated with ‘SPPCSPC’, ‘C2’ indicates the high-resolution features C2 obtained from the backbone network, 'GhostConv' indicates the Ghost Convolution, 'Decoupling' indicates the input decoupling strategy utilized for semantic segmentation, 'TverskyLoss' and 'FocalLoss' indicates the Tversky loss and Focal loss utilized for lane detection, respectively, 'DLA' and 'HDA' indicate the dynamic label assignment strategy and the hybrid data augmentation strategy, respectively, 'SA' indicate the SA-based refined module, 'MRP' indicate model re-parameterization technique. The survey results indicate: (1) The introduction of 'ELAN' and 'C2' significantly enhances network performance, but also greatly increases network parameters. (2) The utilization of 'GhostConv' maintains the network performance while drastically reducing the network parameters. (3) The utilization of 'Decoupling' enhances the segmentation performance of the network, and the employment of 'TverskyLoss' and 'FocalLoss' further improves the pixel accuracy. (4) The combination of 'DLA' and 'HDA' enables balancing and optimizing the network without increasing inferencing effort, leading to better performance and robustness. (5) The introduction of 'SA' enables the network to achieve more accurate lane detection based on more comprehensive contextual information. As shown in Figure 8, our prediction results are more continuous and accurate after utilizing the SA-based refined module. (6) The 'MRP' reduces network parameters and speeds up network inference without sacrificing prediction accuracy. In conclusion, the proposed tricks are beneficial for improving network performance.

Table 9. The ablation studies of different optimization tricks in our proposed model. The best scores are in bold.

Method	Recall	mAP50	mIOU	Accuracy	IoU	Params
Baseline	89.2	76.5	91.5	70.5	26.2	**7.9 M**
+ELAN	88.9	76.6	92.7	84.0	23.2	31.6 M
+C2	93.9	81.9	92.5	84.8	23.1	37.9 M
+GhostConv	93.7	82.4	92.3	84.4	23.5	29.3 M
+Decoupling	94.4	82.6	92.8	85.9	24.8	30.9 M
+FocalLoss	94.2	83.0	92.8	86.8	24.4	30.9 M
+TverskyLoss	94.1	82.7	**93.2**	87.9	23.7	30.9 M
+DLA	95.6	82.9	92.7	87.6	23.9	30.9 M
+HDA	**96.9**	83.4	92.8	87.5	26.8	30.9 M
+SA	**96.9**	**84.3**	**93.2**	**88.3**	**28.0**	30.9 M
+MRP	**96.9**	**84.3**	**93.2**	**88.3**	**28.0**	30.2 M

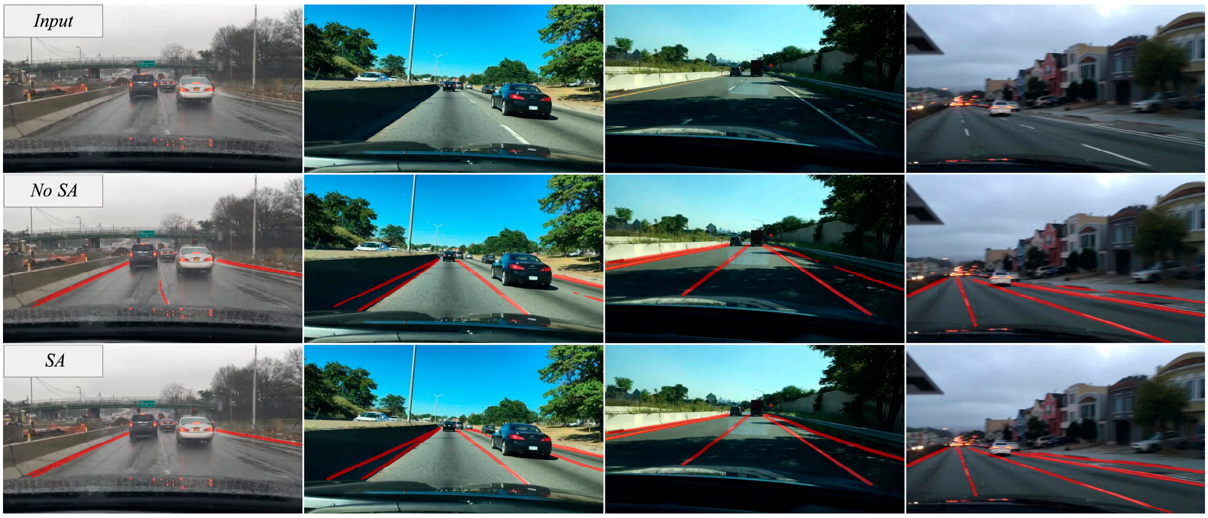

Figure 8. The comparison of lane detection before and after utilizing the SA-based refined module. The red lines show the lane lines. It is obvious that our results demonstrate better accuracy and continuity after utilizing the SA-based refined module.

5. Conclusions

We propose YOLOPv3, a novel anchor-based multi-task visual perception network capable of handling traffic object detection, drivable area segmentation, and lane detection simultaneously. We build YOLOPv3 upon the one-stage detector YOLOv7 and achieve significant improvement in network performance and training efficiency. Firstly, we propose a novel multi-task learning architecture designed to effectively exploit multi-scale high-resolution features, thereby enhancing the network's predictive capabilities in small object detection and small region segmentation. Secondly, we propose a lightweight self-attention (SA)-based refined module and integrate it into the lane detection head. This module aims at capturing non-local contextual dependencies to boost lane detection performance. Finally, we propose optimization improvements to optimize the training process without increasing inference expense, thus enabling the anchor-based multi-task network to attain the best results via straightforward end-to-end training. Experimental results demonstrate that YOLOPv3 sets a new SOTA on three visual tasks of the challenging BDD100K dataset and runs in real time. As a result, YOLOPv3 is able to accurately recognize various traffic objects in real time despite intricate road conditions as well as constrained computing sources. We intend to broaden the functionality of YOLOPv3 in the future to meet the practical requirements of intelligent driving vehicles. For instance, by adding an instance segmentation head to obtain the geometry information of traffic objects. In addition, we also plan to further compress the model to enable its real-time operation on embedded devices. We hope that our efforts will facilitate researchers and developers in achieving improved performance across diverse intelligent driving scenarios and provide inspiration for future advancements in this field.

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

Funding: This research was funded by National Key Research and Development Program of China, grant number 2022YFB3903801, Major Science and Technology Project of Hubei Province of China, grant number 2022AAA009.

Data Availability Statement: Data used in this study is available at https://doc.bdd100k.com/index.html upon request. The code that support the findings of this study is available at https://github.com/jiaoZ7688/YOLOPv3, accessed on 2 March 2024.

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

References

1. Qian, Y.; Dolan, J.M.; Yang, M. DLT-Net: Joint detection of drivable areas, lane lines, and traffic objects. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 4670–4679. [CrossRef]
2. Wu, D.; Liao, M.-W.; Zhang, W.-T.; Wang, X.-G.; Bai, X.; Cheng, W.-Q.; Liu, W.-Y. Yolop: You only look once for panoptic driving perception. *Mach. Intell. Res.* **2022**, *19*, 550–562. [CrossRef]
3. Vu, D.; Ngo, B.; Phan, H. Hybridnets: End-to-end perception network. *arXiv* **2022**, arXiv:2203.09035.
4. Han, C.; Zhao, Q.; Zhang, S.; Chen, Y.; Zhang, Z.; Yuan, J. Yolopv2: Better, faster, stronger for panoptic driving perception. *arXiv* **2022**, arXiv:2208.11434.
5. Zhan, J.; Luo, Y.; Guo, C.; Wu, Y.; Meng, J.; Liu, J. YOLOPX: Anchor-free multi-task learning network for panoptic driving perception. *Pattern Recognit.* **2024**, *148*, 110152. [CrossRef]
6. 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 Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37. [CrossRef]
7. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-stage Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636. [CrossRef]
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, Real-Time Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.

9. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241. [CrossRef]
10. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
11. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890. [CrossRef]
12. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the Computer Vision–ECCV 2018: 16th European Conference, Munich, Germany, 8–14 September 2018; pp. 801–818.
13. Wang, Z.; Ren, W.; Qiu, Q. Lanenet: Real-time lane detection networks for autonomous driving. *arXiv* **2018**, arXiv:1807.01726.
14. Pan, X.; Shi, J.; Luo, P.; Wang, X.; Tang, X. Spatial as deep: Spatial cnn for traffic scene understanding. In Proceedings of the 2018 AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32. [CrossRef]
15. 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, Republic of Korea, 27 October–2 November 2019; pp. 1013–1021. [CrossRef]
16. Paszke, A.; Chaurasia, A.; Kim, S.; Culurciello, E. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv* **2016**, arXiv:1606.02147.
17. Yu, F.; Chen, H.F.; Wang, X.; Xian, W.; Chen, Y.Y.; Liu, F.C.; Madhavan, V.; Trevor, D. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 2633–2642. [CrossRef]
18. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network For Instance Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
19. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
20. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790. [CrossRef]
21. Tian, D.; Han, Y.; Liu, Y.; Li, J.; Zhang, P.; Liu, M. Hybrid Cross-Feature Interaction Attention Module for Object Detection in Intelligent Mobile Scenes. *Remote Sens.* **2023**, *15*, 4991. [CrossRef]
22. Yang, R.; Yu, Y. Artificial Convolutional Neural Network in Object Detection and Semantic Segmentation for Medical Imaging Analysis. *Front. Oncol.* **2021**, *11*, 638182. [CrossRef]
23. Zhang, H.; Li, X.; Wang, L.; Liu, D.; Wang, S. Construction and Optimization of a Collaborative Harvesting System for Multiple Robotic Arms and an End-Picker in a Trellised Pear Orchard Environment. *Agronomy* **2024**, *14*, 80. [CrossRef]
24. Pan, M.; Xia, W.; Yu, H.; Hu, X.; Cai, W.; Shi, J. Vehicle Detection in UAV Images via Background Suppression Pyramid Network and Multi-Scale Task Adaptive Decoupled Head. *Remote Sens.* **2023**, *15*, 5698. [CrossRef]
25. Lang, K.; Cui, J.; Yang, M.; Wang, H.; Wang, Z.; Shen, H. A Convolution with Transformer Attention Module Integrating Local and Global Features for Object Detection in Remote Sensing Based on YOLOv8n. *Remote Sens.* **2024**, *16*, 906. [CrossRef]
26. Ma, Y.; Zhou, D.; He, Y.; Zhao, L.; Cheng, P.; Li, H.; Chen, K. Aircraft-LBDet: Multi-Task Aircraft Detection with Landmark and Bounding Box Detection. *Remote Sens.* **2023**, *15*, 2485. [CrossRef]
27. Girshick, R. Fast r-cnn. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
28. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
29. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2961–2969.
30. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
31. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
32. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end Object Detection with Transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229. [CrossRef]
33. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
34. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
35. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
36. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7794–7803.

37. Chen, C.-F.R.; Fan, Q.; Panda, R. Crossvit: Cross-attention multi-scale vision transformer for image classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), ELECTR Network, Montreal, BC, Canada, 11–17 October 2021; pp. 357–366. [CrossRef]
38. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H.S.; et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 6877–6886. [CrossRef]
39. Wang, Q.; Wang, L.; Chi, Y.; Shen, T.; Song, J.; Gao, J.; Shen, S. Dynamic Data Augmentation Based on Imitating Real Scene for Lane Line Detection. *Remote Sens.* **2023**, *15*, 1212. [CrossRef]
40. 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. [CrossRef]
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
42. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
43. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114. [CrossRef]
44. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
45. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 18–24 July 2021; pp. 10012–10022.
46. Alexey, B.; Wang, C.-Y.; Liao, H.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
47. 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 (CVPR), Virtual, 19–25 June 2021; pp. 13029–13038. [CrossRef]
48. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589. [CrossRef]
49. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. Repvgg: Making vgg-style convnets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742. [CrossRef]
50. Li, Z.; Wang, W.; Xie, E.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P.; Lu, T. Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 1280–1289. [CrossRef]
51. Liu, H.; Liu, F.; Fan, X.; Huang, D. Polarized self-attention: Towards high-quality pixel-wise regression. *arXiv* **2021**, arXiv:2107.00782.
52. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.
53. Buslaev, A.; Iglovikov, V.I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A.A. Albumentations: Fast and flexible image augmentations. *Information* **2020**, *11*, 125. [CrossRef]
54. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
55. Hu, M.; Feng, J.; Hua, J.; Lai, B.; Huang, J.; Gong, X.; Hua, X.-S. Online convolutional re-parameterization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 568–577. [CrossRef]
56. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988. [CrossRef]
57. Salehi, S.S.M.; Erdogmus, D.; Gholipour, A. Tversky loss function for image segmentation using 3D fully convolutional deep networks. Machine Learning in Medical Imaging. In Proceedings of the 8th International Workshop, MLMI 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, 10 September 2017; Proceedings 8; Springer International Publishing: Cham, Switzerland, 2017; pp. 379–387. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.