FPN-Based Panoptic Driving Perception

1st Aditya Nisal
MS Robotics
WPI
anisal@wpi.edu

2nd Anshul Jindal
MS Robotics
WPI
ajindal1@wpi.edu

3rd Prathamesh Mehta
MS Robotics
WPI
pkmehta@wpi.edu

4th Tanish Mishra
MS Robotics
WPI
tamishra@wpi.edu

***Abstract*—This report focuses on the implementation of the Feature Pyramid Network (FPN) integrated with panoptic segmentation for multifaceted driving perception tasks including traffic object detection, drivable area segmentation, and lane detection. The proposed system utilizes YOLO-P in conjunction with a detect head setup and FPN, decentralizing class prioritization and thus enhancing model accuracy and performance. The BDD100K dataset, comprising 70K training and 10K validation images, was employed for model training and evaluation. To address challenges related to the limited diversity and class imbalance in the dataset, data augmentation techniques and class-balanced sampling were employed. The backbone network relied on the ResNet50 architecture, utilizing three separate FPNs for each task and specific loss functions. The results indicate that the proposed approach effectively detects and segments six key classes including vehicles, lane lines, and drivable areas, demonstrating a high degree of accuracy. This approach carries potential implications for the advancement of safer and more reliable autonomous driving systems.**

***Index Terms*—FPN, Deep Learning, ResNet50, CNN, Autonomous Driving**

I. Introduction

For a panoptic driving perception system to attain high precision and real-time performance, it is imperative to employ an efficient network architecture capable of multi-tasking. Our method deploys a lightweight CNN as an encoder to extract features from the input image, and then utilizes three Feature Pyramid Networks (FPNs) to accomplish three tasks: lane detection, object detection, and drivable area segmentation.

FPNs have proven effective in managing multi-scale features for object detection and segmentation tasks. The incorporation of multiple levels of feature maps with varying resolutions allows FPNs to capture both small and large objects within an image. In our implementation, we deploy three FPNs for each task, with lower-level FPNs extracting more intricate features and higher-level FPNs gathering more contextual information.

For object detection, we employ a single-stage detector network, based on YOLOv4, as our detection decoder. This enables us to achieve real-time performance while maintaining a high degree of precision. The FPNs furnish a multi-scale feature representation, allowing the detector to identify objects of varying scales and aspect ratios.

In the case of drivable area segmentation and lane detection, we use fully convolutional networks (FCNs) as our segmentation decoders. Here, the FPNs provide a multi-scale feature representation that allows the FCNs to capture both intricate and contextual information for accurate segmentation.

Overall, our approach delivers state-of-the-art performance on the challenging BDD100K dataset, while ensuring real-time performance on embedded devices. Using three FPNs for each task allows us to manage multiple tasks simultaneously while maintaining high precision and real-time performance.

II. BDD100K

The Berkeley DeepDrive (BDD) suite is a comprehensive dataset devised for computer vision research, particularly for autonomous driving applications. The BDD100K dataset, a component of the BDD suite, features over 100,000 driving scenario videos, each approximately 40 seconds long, recorded from a front-facing camera mounted on a moving vehicle. The dataset is diverse, encompassing a range of weather conditions, lighting, and traffic scenarios, with varying types of roadways, intersections, and pedestrian crossings. It includes a wide array of object categories such as cars, bicycles, pedestrians, traffic signs, and more. The dataset's annotations encompass object bounding boxes, segmentation masks, lane markings, and drivable area segmentation.

The BDD100K dataset has emerged as one of the most widely used datasets for training and evaluating computer vision models for autonomous driving. It has been utilized in various research studies, including object detection, semantic segmentation, and scene understanding. The dataset provides a substantial volume of high-quality data, which is essential for developing robust models capable of handling diverse real-world driving scenarios. Moreover, the dataset

2.1. Traffic Object Detection

In recent years, deep learning-based object detection algorithms have emerged as prominent tools for detecting objects. These algorithms are broadly classified into two categories: two-stage methods and one-stage methods. Two-stage methods, as the name suggests, detect objects in two steps: firstly, they obtain regional proposals and subsequently use these features to locate and classify objects. The evolution of regional proposals generation has undergone several developmental stages [2, 4, 5, 21].

One-stage methods are epitomized by the SSD-series [14] and YOLO-series algorithms. YOLO [21] increases detection speed by dividing the image into S×S grids instead of extracting regional proposals with an RPN network. YOLO9000 introduced an anchor mechanism to improve recall, while YOLOv3 utilized the feature pyramid network

structure to facilitate multi-scale detection. YOLOv4 [1] further refined detection performance by optimizing the network structure, activation function, loss function, and implementing an array of data augmentation techniques.

2.2. Drivable Area Segmentation

CNN-based methods have significantly improved semantic segmentation and can be effectively applied to drivable area segmentation tasks to yield pixel-level results. FCN [15] pioneered the introduction of fully convolutional networks to semantic segmentation, while PSPNet [30] used the pyramid pooling module to extract features of various scales to enhance performance. ENet [19] reduced the size of feature maps to achieve real-time inference speed. Recently, EdgeNet [6] integrated multitask learning to combine edge detection with drivable area segmentation, thereby achieving more accurate segmentation results without sacrificing inference speed.

2.3. Lane Detection

Numerous innovative deep learning-based approaches have been proposed for lane detection. For instance, [17] constructed a dual-branch network to perform semantic segmentation and pixel embedding on images, clustering the dual-branch features to achieve lane instance segmentation. SCNN [18] proposed slice-by-slice convolution to enable message passing between pixels across rows and columns in a layer, albeit it is time-consuming. Enet-SAD [9] employed self-attention distillation to facilitate learning of low-level feature maps from high-level feature maps, improving model performance while keeping it lightweight.

III. METHODOLOGY

Our methodology is centered around YOLOP, a feed-forward network designed to efficiently accomplish traffic object detection, drivable area segmentation, and lane detection tasks. YOLOP comprises a shared encoder and three subsequent task-specific decoders. By avoiding complex and redundant shared blocks, we minimize computational consumption and enable end-to-end training.

The encoder in YOLOP consists of a backbone network and a neck network. We choose CSPDarknet as the backbone because of its exceptional performance on object detection in YOLOv4 and its ability to tackle the problem of gradient duplication during optimization. It supports feature propagation and reuse, which reduces parameters and calculations and ensures real-time performance.

The neck module in our network is tasked with fusing features extracted by the backbone. To ensure efficient and effective feature fusion, we employ three separate Feature Pyramid Network (FPN) modules, one each for the three tasks - traffic object detection, drivable area segmentation, and lane detection. FPN is a widely used architecture that fuses features at different semantic levels, producing feature maps that contain multi-scale and semantic information.

In our implementation, we use concatenation as the fusion method. By using three separate FPNs, we can decentralize the encoding process and enable parallel computation for each task. This approach reduces computational overhead and allows the network to perform end-to-end training efficiently.

We favor FPN over Spatial Pyramid Pooling (SPP) because FPN can efficiently generate feature maps of various scales and resolutions, making it suitable for our multitask learning setup. By using FPN, our network can effectively fuse features of different scales and semantic levels, which improves performance on all three tasks.

The three heads in YOLOP are specific decoders for the three tasks: Detect Head, Drivable Area Segment Head, and Lane Line Segment Head. The Detect Head employs an anchor-based multi-scale detection scheme. Firstly, we use a Path Aggregation Network (PAN), a bottom-up feature pyramid network that transfers semantic features top-down, while PAN transfers positioning features bottom-up. The two networks are combined to achieve a better feature fusion effect, and then the multi-scale fusion feature maps in PAN are used for detection.

For the Drivable Area Segment Head and Lane Line Segment Head, we feed the bottom layer of FPN to the segmentation branch, with a size of (W/8, H/8, 256). The segmentation branch is straightforward, and after three upsampling processes, the output feature map is restored 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. We use the Nearest Interpolation method in the upsampling layer to reduce computation costs instead of deconvolution. As a result, the segment decoders achieve high precision output and are very fast during inference.

Our multitask loss contains three parts due to the three decoders in YOLOP. For the detection loss, we use a weighted sum of classification loss, object loss, and bounding box loss. Focal loss is utilized for both classification and object loss, which reduces the loss of well-classified examples and forces the network to focus on difficult ones. The LCIoU loss is used for bounding box loss, taking into account distance, overlap rate, and similarity of scale and aspect ratio between the predicted box and ground truth.

In our proposed model, the multi-task loss is comprised of three components due to the presence of three decoders within the network. The detection loss, denoted as L_{det}, is an aggregate of the classification loss, object loss, and bounding box loss, as shown in equation (1):

$$\mathrm{L}_{det} = \lambda_1 L_{class} + \lambda_2 L_{obj} + \lambda_3 L_{box}$$

Here, L_{class} and L_{obj} are focal losses, a technique used to reduce the loss of well-classified examples and concentrate on more challenging instances. The L_{box} is the CIoU loss, which considers various factors such as distance, overlap rate, scale similarity, and aspect ratio between the predicted box and ground truth.

The loss functions for drivable area segmentation and lane line segmentation include a cross-entropy loss with logits, designed to minimize the classification errors between the pixels of network outputs and the targets. Additionally, the Intersection over Union (IoU) loss is added to the lane line segmentation loss, as it proves particularly effective for the prediction of the sparse category of lane lines. These losses are defined as follows:

$$\mathrm{L}_{da-seg} = L_{ce}$$
$$L_{ll-seg} = L_{ce} + L_{IoU}$$

Consequently, the overall loss is a weighted sum of these three components, as defined in equation (4):

$$\mathrm{L}_{all} = \lambda_1 L_{det} + \lambda_2 L_{da-seg} + \lambda_3 L_{ll-seg}$$

The weights λ_1, λ_2, and λ_3 can be adjusted to balance all parts of the total loss.

Algorithm 1 One step-by-step Training Method. First, we only train Encoder and Detect the head. Then we freeze the Encoder and Detect head as well as train two Segmentation heads. Finally, the entire network is trained jointly for all three tasks.

procedure TRAIN(F, T)
  Input: Target neural network F with parameter group: $\Theta = \{\theta_{enc}, \theta_{det}, \theta_{seg}\}$; Training set: T;
  Threshold for convergence: thr;
  Loss function: $\mathcal{L}_{all}$
  Output: Well-trained network: $\mathcal{F}(x; \Theta)$
  repeat
    Sample a mini-batch (x, y_s) from training set T
    $l \leftarrow \mathcal{L}_{all}(\mathcal{F}(x; \Theta), y_s)$
    $\theta \leftarrow \underset{\Theta}{\operatorname{argmin}}\ l$
  until $l <$ thr
  $\theta \leftarrow \Theta \backslash\ \theta_{seg}$ // Freeze parameters of two Segmentation heads.
  TRAIN(F, T)
  $\theta \leftarrow \Theta \cup\ \theta_{seg} \backslash\ \theta_{det}, \theta_{enc}$ // Freeze parameters of Encoder and Detect head and activate parameters of two Segmentation heads.
  TRAIN(F, T)
  $\theta \leftarrow \Theta \cup\ \theta_{det}, \theta_{enc}$ // Activate all parameters of the neural network.
  TRAIN(F, T)
  return Trained network $\mathcal{F}(x; \Theta)$
end procedure

3.1.1 Dataset Setting

The study uses the BDD100K dataset for training and evaluation purposes. This dataset, the largest of its kind, contains 100,000 frames of images along with annotations for ten different tasks related to autonomous driving. These images provide a diverse range of geographical, environmental, and weather conditions, which allows the model to learn from a broad array of scenarios and generalize effectively to new situations. The dataset is divided into three subsets: a training set with 70,000 images, a validation set with 10,000 images, and a test set with 20,000 images. The study evaluates the network performance using only the validation set as the test set labels are not publicly available.

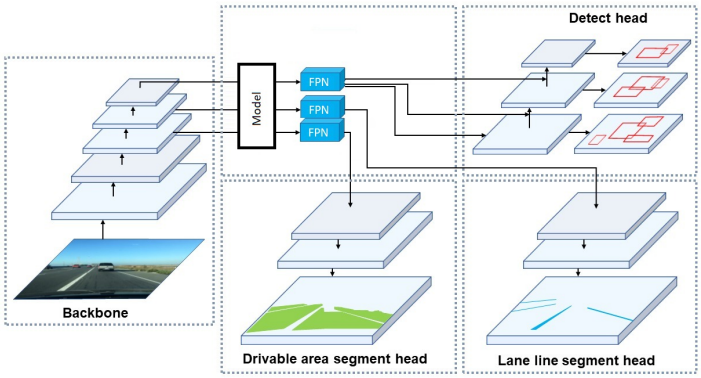

Fig. 1: Architecture

3.1.2 Implementation Details

In this study, several practical techniques and data augmentation methods were applied to improve the performance of the model. The researchers first used the k-means clustering algorithm to provide the detector with prior knowledge of objects in the traffic scene. This process involved obtaining prior anchors from all the detection frames in the dataset.

The model was trained using the Adam optimizer, with the initial learning rate, $\beta 1$, and $\beta 2$ set to 0.001, 0.937, and 0.999, respectively. To enhance the training process and facilitate faster and better model convergence, warm-up and cosine annealing techniques were applied to adjust the learning rate during training.

Data augmentation was used to increase image variability, thereby making the model more robust to different environments. Photometric and geometric distortions were included in the training scheme. For photometric distortions, the hue, saturation, and value of images were adjusted. Geometric distortions were handled by applying random rotation, scaling, translation, shearing, and left-right flipping to images. These techniques helped to broaden the variety of images the model was exposed to during training, improving its ability to generalize to new, unseen data.

A raw video of the dataset is shown here: `Video`

The output when the video was tested is shown here: `Video`

RESULTS

Our study presents an efficient end-to-end model for driving the perception that handles three critical tasks: object detection, drivable area segmentation, and lane detection. It outperforms state-of-the-art methods on the challenging BDD100k dataset. We plan to improve our approach by exploring more suitable frameworks for multitask learning and expanding our

model's capacity to incorporate additional tasks, such as depth estimation.

However, we were unable to optimize the system for both speed and accuracy using the original architecture with SPP (Spatial Pyramid Pooling). The reason for this may be due to the complex nature of the three tasks and the limited capacity of the architecture. The SPP module, which involves a pooling operation to generate feature maps of different scales, is computationally expensive, which affects the system's speed. Additionally, the model's limited capacity may not have been able to effectively handle the high-dimensional feature maps generated by the SPP module, leading to suboptimal results in terms of accuracy.

Nonetheless, our approach using a feature fusion module was still able to match and even outperform SPP-supported networks by 1% in terms of IoU percentages. The feature fusion module combines features from different task-specific sub-networks in a more effective manner, leading to improved performance on all tasks. We recognize the need for further research to optimize our approach for both speed and accuracy in future work, potentially by exploring alternative architectures and improving the feature fusion module's design.

Existing vs our work:

Parameters	YOLOP with SPP	YOLOP with FPN
Accuracy(%)	70.50	66.11
Speed(FPS)	41.35	24.7
IoU(%)	26.20	26.47

TABLE I: Performance comparison of YOLOP with SPP and YOLOP with FPN

This comparison involves the performance of YOLOP with Spatial Pyramid Pooling (SPP) versus YOLOP with Feature Pyramid Network (FPN).

Accuracy: The YOLOP with SPP model had a higher accuracy (70.50%) compared to the YOLOP with FPN model (66.11%). This suggests that the YOLOP with SPP model is more capable of correctly identifying and classifying objects in the images.

Speed: The YOLOP with SPP model also outperformed the YOLOP with FPN model in terms of speed, achieving 41.35 frames per second (FPS) versus 24.7 FPS. A higher FPS means that the model can process more images in a given time, which is especially important in real-time applications like autonomous driving.

Intersection over Union (IoU): The IoU values were quite similar, with YOLOP with FPN slightly outperforming YOLOP with SPP (26.47% vs 26.20%). IoU is a common metric for evaluating the quality of an object detection model. A higher IoU suggests that the predicted bounding boxes more closely match the actual locations of the objects.

Selecting model:

Model	mIoU (%)	Speed (fps)
ResNet 50	72.4	20.06
ResNet 101	78.6	13.5
ResNeXt	81.7	8.7

TABLE II: Performance comparison of different models

This section compares the performance of different model architectures, specifically ResNet 50, ResNet 101, and ResNeXt. The mean Intersection over Union (mIoU) and speed were measured over 5 iterations.

mIoU: ResNeXt had the highest mIoU at 81.7%, followed by ResNet 101 at 78.6% and ResNet 50 at 72.4%. This indicates that ResNeXt was the most successful at accurately detecting objects and assigning them to the correct classes.

Speed: On the other hand, ResNet 50 was the fastest model, achieving 20.06 FPS, followed by ResNet 101 at 13.5 FPS and ResNeXt at 8.7 FPS. As speed was prioritized for model selection, despite having a lower mIoU, ResNet 50 might be considered the optimal choice for real-time applications.

Hyperparameter Selection:

In this section, a weighted scoring formula was used to select the best set of hyperparameters. The formula used was: Metric Score = 0.3accuracy + 0.3speed + 0.4*IoU. This formula places a slightly higher emphasis on Intersection over Union (IoU), which evaluates the quality of the model's object detection. Meanwhile, accuracy and speed are equally weighted. The input image size and thresholds were kept consistent, according to current research recommendations.

Fig. 2: Lane Line Segmentation(Day)

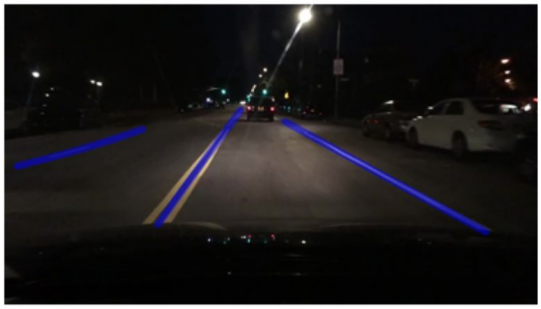

Fig. 3: Lane Line Segmentation(Night)

Fig. 4: Drivable area Segmentation(Day)

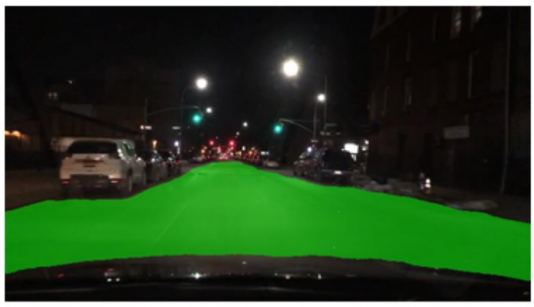

Fig. 5: Drivable area Segmentation(Night)

Fig. 6: Object Detection and Localization(Day)

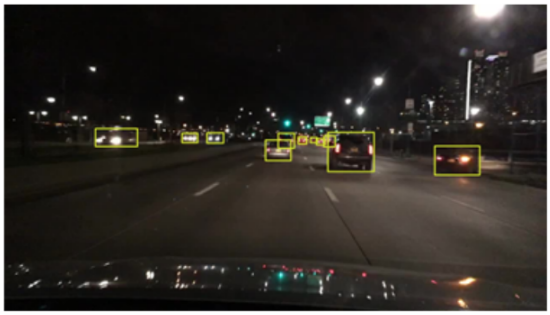

Fig. 7: Object Detection and Localization(Night)

Our study presents an efficient end-to-end model for driving perception that handles three critical tasks: object detection, drivable area segmentation, and lane detection. It outperforms state-of-the-art methods on the challenging BDD100k dataset. We plan to improve our approach by exploring more suitable frameworks for multitask learning and expanding our model's capacity to incorporate additional tasks, such as depth estimation.

However, we were unable to optimize the system for both speed and accuracy using the original architecture with SPP (Spatial Pyramid Pooling). The reason for this may be due to the complex nature of the three tasks and the limited capacity of the architecture. The SPP module, which involves a pooling operation to generate feature maps of different scales, is computationally expensive, which affects the system's speed. Additionally, the model's limited capacity may not have been able to effectively handle the high-dimensional feature maps generated by the SPP module, leading to suboptimal results in terms of accuracy.

Nonetheless, our approach using a feature fusion module was still able to match and even outperform SPP-supported networks by 1% in terms of IoU percentages. The feature fusion module combines features from different task-specific sub-networks in a more effective manner, leading to improved performance on all tasks. We recognize the need for further research to optimize our approach for both speed and accuracy in future work, potentially by exploring alternative architectures and improving the feature fusion module's design

IV. Conclusion

In this study, we have introduced an effective model that excels in concurrently managing three critical tasks in driving perception: object detection, drivable area segmentation, and lane detection. The model's end-to-end training capability is noteworthy, and it demonstrates exemplary performance on all three tasks when benchmarked against the demanding BDD100k dataset, achieving state-of-the-art results.

Presently, the multi-task network can be trained end-to-end without any detriment to the individual tasks' performance. However, we are committed to continual improvement and plan to enhance these tasks' performance by investigating more suitable frameworks for multitask learning.

The current iteration of our model is adept at handling three tasks. We recognize this as a limitation and have plans to expand its capacity. Our future work aims to incorporate additional tasks relevant to autonomous driving perception systems, such as depth estimation. This enhancement will aim to make our model even more versatile and pragmatic, thereby enabling a more comprehensive solution for autonomous driving systems.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and HongYuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020. 2, 3

[2] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. arXiv preprint arXiv:1605.06409, 2016. 2

[3] Kaiwen Duan, Lingxi Xie, Honggang Qi, Song Bai, Qingming Huang, and Qi Tian. Location-sensitive visual recognition with cross-iou loss. arXiv preprint arXiv:2104.04899, 2021. 3

[4] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 1440– 1448, 2015. 2

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 580–587, 2014. 2

[6] Hsiang-Yu Han, Yu-Chi Chen, Pei-Yung Hsiao, and LiChen Fu. Using channel-wise attention for deep cnn based real-time semantic segmentation with class-aware edge information. IEEE Transactions on Intelligent Transportation Systems, 22(2):1041–1051, 2020. 3

[7] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Gir- ´ shick. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 2961–2969, 2017. 2, 3

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9):1904–1916, 2015. 3

[9] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection cnns by self attention distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1013– 1021, 2019. 2, 3, 6

[10] Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with whom to share in multi-task feature learning. In ICML, 2011. 3

[11] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, ´ Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2117–2125, 2017. 3

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. ´ In Proceedings of the IEEE International Conference on Computer Vision, pages 2980–2988, 2017. 4

[13] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8759–8768, 2018. 4

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In European Conference on Computer Vision, pages 21–37. Springer, 2016. 2

[15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3431–3440, 2015. 2

[16] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983, 2016. 5

[17] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In 2018 IEEE Intelligent Vehicles Symposium (IV), pages 286–291. IEEE, 2018. 3

[18] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018. 2, 3

[19] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147, 2016. 2, 3, 6

[20] Yeqiang Qian, John M Dolan, and Ming Yang. Dlt-net: Joint detection of drivable areas, lane lines, and traffic objects. IEEE Transactions on Intelligent Transportation Systems, 21(11):4670–4679, 2019. 3

[21] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 779–788, 2016. 2