ISBN: 978-981-18-1791-5
2021 the 11th International Workshop on Computer Science and Engineering (WCSE 2021)
doi: 10.18178/wcse.2021.06.015

Training Instance Segmentation and Lane Detection Models in One Network Architecture

Dunan Ye [1 +], Jun Guo [2] and Youguang Chen [1]

[1] School of Data Science Engineering, East China Normal University, Shanghai, China
[2] Information Technology Services, East China Normal University, Shanghai, China

Abstract. A new network architecture with a novel training method is proposed in this paper which can achieve two tasks of road defects instance segmentation and lane detection. It is composed of a backbone and two independent output branches for instance segmentation and lane detection. The experiments are conducted on new datasets collected by us. Through our method of alternately training two network branches while continuously reducing the learning rate, it can be found that the accuracy of our two branches can be similar with the accuracy training with two different models. This shows the effectiveness of our training method. Furthermore, our method can reduce model memory.

Keywords: instance segmentation, lane detection, alternately training, road defects

1. Introduction

With the rapid development of the road network, road safety issues have received more and more attention. In the case of vehicle crushing and weather damage, the road itself will suffer a lot of damage. Road defects not only affect pedestrians' walking, but also affect vehicle driving. So how to quickly detect road defect has become a hot issue. Road defects detection can be regarded as an instance segmentation task.

In the existing machine learning algorithms, such as instance segmentation [1-4] and object detection [5-7], tasks are handled separately, that is to train a separate network for each task, and most neural network algorithms are single-task. However, in real life many issues should be considered together. For example, a self-driving car not only needs to distinguish objects in the scene, but also needs to identify lanes for safe navigation. In our research topic, we not only need to segment the road defects, but also we need to detect the lane where the defects is located, so that the defects can be repaired more quickly. The traditional method is to segment the road defects and detect the lane in the two networks separately, and then do a post-processing. However, in this way the detection speed is slow and the network model is large. So we want to achieve two tasks in one network. That is, we want to detect the lane on the road while segment the road defects, so as to determine which lane the road defects is on. In our work, we collected road pictures and marked the road defects with reference to the coco format [8]. At the same time, we also marked the lanes on the collected road pictures. And we propose a new network [9], in our network, we have two different detection branches (instance segmentation branch and lane detection branch), sharing the same backbone. As for the input, we use two data sets in different formats, one is the road defects data set, and the other is the road lanes data set. By alternately training different network heads while continuously reducing the learning rate, our new training method can achieve the two tasks of road defects instance segmentation and lane detection in a network, so as to determine which lane the defects is in.

2. Related Work

2.1. Lane Detection

The lane detection technology [10] can detect the lane on the road, so that the vehicle can be correctly positioned in the lane. Lane detection has been developed for a long time. the traditional lane detection algorithm based on image processing has poor scene adaptability, poor robustness, and poor detection effect.

+ Corresponding author. Tel.: + 15967160232;
E-mail address: 707067920@qq.com.

With the development of deep learning, the lane detection algorithm based on deep learning has shown the superiority in lane detection. These algorithms usually treat lane detection as a semantic segmentation task. For example, LaneNet [11] proposed a two-stage lane detection algorithm, including a lane edge proposal network and a lane positioning network. The lane edge proposal network is a semantic network with an Encoder-Decoder structure. By generating a pixel-by-pixel prediction probability, it is judged whether the pixel is a lane. Based on this method, information can be propagated between neurons in the same layer to solve the detection of occlusion or unobvious sight cues [12]. Ultra Fast Structure-aware Deep Lane Detection (UFDLD) [13] detect lanes based on the rows, and its network structure is simpler. At the same time, a new definition of lane detection is proposed. Lane detection is defined as finding a collection of the positions of lanes in certain rows of the image. This method eliminates the need for pixel-by-pixel classification, which can greatly improve the speed of lane detection.

2.2. Instance Segmentation

An image is a collection of pixels, and instance segmentation is essentially to classify each pixel in the image. With the development of deep learning, today's instance segmentation algorithms use convolutional neural networks (CNN) [14] to extract features in images, which greatly improves the performance of instance segmentation. Instance segmentation can be regarded as solving both semantic segmentation and object detection. There are many popular instance segmentation algorithms. The two-stage top-down Mask-RCNN [2] inherits the basic network of Faster-RCNN [15] , the box branch is used for object detection, and a mask branch is added for semantic segmentation, Mask-RCNN [2] use RPN network to extract the region of interest in the feature map, then use the full convolutional neural network (FCNN) [16] to generate the mask in the region of interest, Mask-RCNN [2] can be seen as the beginning of multi-task learning under the RCNN structure [17] , and still dominates today. The two-stage bottom-up instance segmentation is mainly to perform pixel-level semantic segmentation first, and then distinguish different instances by means like metric learning. Single-stage instance segmentation is inspired by single-stage object detection.

2.3. Multi-task Learning

Since the development of machine learning, most of the machine learning models deal with a certain task individually, such as classifying images and detecting objects. Most of the time we design an algorithm and then through continuous iterative optimization, the task is finally completed. But in actual problems, there are many tasks that need to be considered together. Learning multiple tasks together can reduce model memory. At the same time, multiple tasks can get results at one time and speed up inference. In this article, we need to achieve two different tasks, one is the instance segmentation of road defects, and the other is the lane detection of the road. Through our training method, we can achieve two tasks in a network model.

3. Methodology

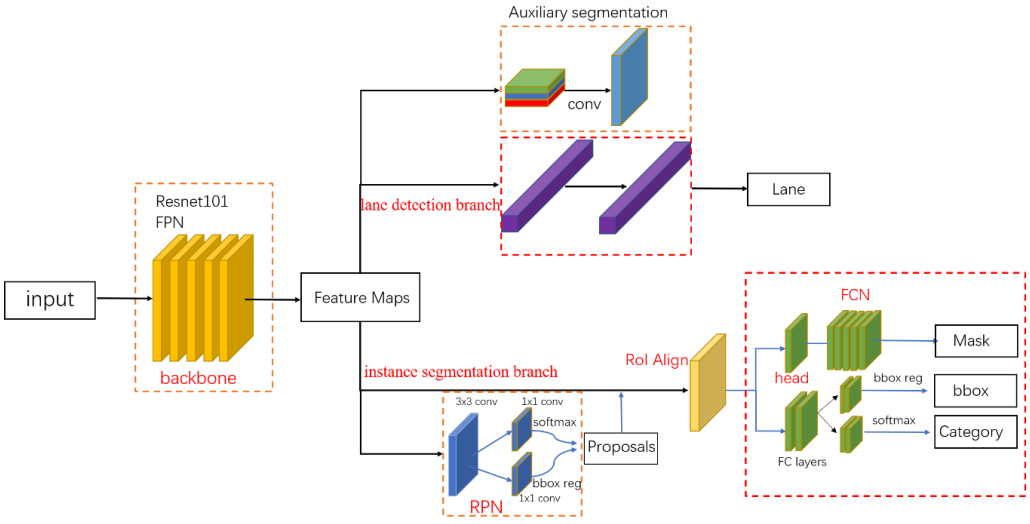


Fig. 1: The architecture of the proposed model.

3.1. Network Architecture

The architecture of our network is shown in Fig. 1. This is a multi-task network that can achieve the two tasks of road defects instance segmentation and lane detection, which includes a backbone and two branch networks. The backbone is ResNet-101 [18], one of the two branch networks is a Mask-RCNN [2] detection network that includes FPN [7], RPN [15] and detector heads, and the other branch is a simple two-layer fully connected layer for lane detection, and an auxiliary network is added for training.

3.2. Our Training Method

Our method is to alternately train two network branches while reducing the learning rate. And for the two network branches, we input different data sets. The backbone network extracts feature maps, so sharing a backbone for two different branches does not cause the backbone to offset to a certain branch, and the fluctuation of the backbone gradually decreases during the alternate training process, finally achieve a balanced status. Another important thing is to adjust the learning rate. In our training method, we reduce the learning rate of the two branches after each alternate training. And reducing the learning rate can be seen as a fine-tuning process [19], the accuracy of two branches will gradually increase in this process.

3.3. Training Methods For Different Branches

When training the lane detection branch, we freeze the instance segmentation branch and input the lane data set for training. For training this branch, we have two training stages, the first stage is training lane detector head, and the second stage is training all layers in this branch including backbone [20]. When training the instance segmentation branch, we freeze the lane detection branch, and input the road defects data set for training. For training this branch, we have three training stages, the first stage is training detector head, the second stage is fine-tuning ResNet-101 (backbone) [18] stage 4 and up, finally fine tune all layers in this branch. In general, for training each branch, our method is to freeze backbone first, train the head of this branch, and then train the entire branch including the backbone together. We let the heads adapt to the backbone, so that after each alternate training, the deviation of the backbone is not too large. Through alternately training two network branches while continuously reducing the learning rate, our two branches can have good performance.

4. Experiment

4.1. Datasets

In order to complete our experiment, we will use two datasets, namely the road defects dataset (RDD) and the lane dataset (LD). Our RDD is a rich dataset, it is collected by us to research road defects in cities. Using artificial intelligence to detect road defects and making quick repairs is the purpose of our research on road defects. While collecting road defects, we spent a lot of time referring to the format of the coco dataset [8] to label them with the help of an annotation tool called Labelme, it is worth mentioning that our data set has made a great contribution to the study of urban road defects. Our RDD has 14 different categories, namely hole, rent, net, load-swelling, subsidence, line-damaged, well-PV, manhole-cover, damage-fixed, guardrail, guardrail-stone, human, car, peeling. More details are shown in the table 1 below. Another auxiliary dataset is our lane dataset (LD), Our LD is also collected by us to research lane. In this article, we use this dataset to determine the location of road defects [21]. The scenes we collected are some scenes in downtown Shanghai and highways. Some more details are shown in the table 2 below.

Table 1: Data sets description of RDD

Data set	Training set	Test set	Resolution	Category
RDD	1953	480	1920*1080	14

Table 2: Data sets description of LD

dataset	Train	Test	Resolution	Lane	Environment
LD	1728	444	1920x1080	$\leqslant 4$	Urban and highway

4.2. Evaluation Metrics

We have two network branches, so for different branches, their evaluation metrics are different. For the lane detection branch, we refer to the evaluation metric of the TuSimple dataset, the evaluation metric is accuracy, and the accuracy is calculated by:

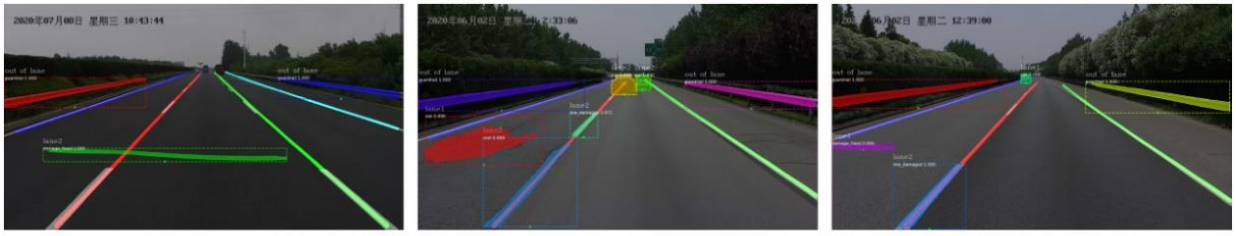
$$accuracy = \frac{\sum_{img} C_{img}}{\sum_{img} S_{img}} \tag{1}$$

C_{img} is the number of lane points predicted correctly and S_{img} is the total number of ground truth in each image.

Actually, we use top1,top2, top3 accuracy. Top1, top2, top3 accuracy are the metrics when the distance of prediction and ground truth is less than 1, 2 and 3, respectively. And top1 accuracy is the standard classification accuracy [14]. For the instance segmentation branch, we use mean average precision (mAP) on IoU = 0.50:0.95 for comparison [2].

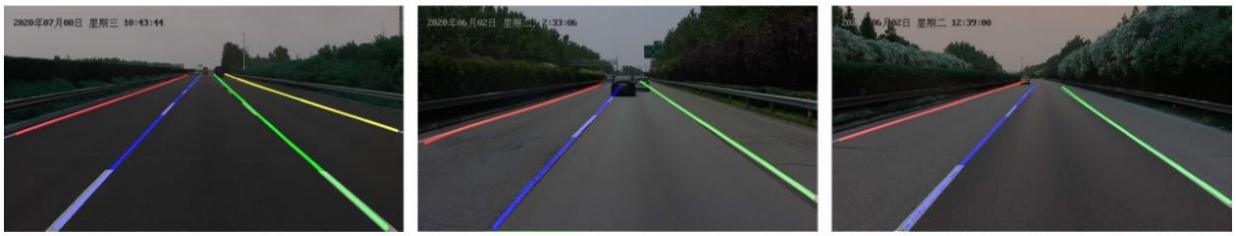
4.3. Training Details

For different branches, the training details are different. For lane detection branch, images are resized to 800*288 following. We use Adam [22] to train this branch with cosine decay learning rate strategy, the batch size is set to 4. For instance segmentation branch, images are resized to 960x540 following, we use SGD to train this branch, the batch size is set to 16. We train and test the model all in an NVIDIA GTX 1080Ti GPU. For the whole training process, firstly we pre-trained the instance segmentation branch with the coco data set [8]. And then during the first alternate training, for training the instance segmentation branch, the learning rate is set to 0.01, and we have three training stages for training this branch introduced in previous section. The training epoch for each stage is 40, 80, 40, for the third stage, the learning rate is one tenth of the one in the first stage. For training lane detection branch, the learning rate is set to 0.0002, cosine decay learning rate strategy is set to 4e-4. We have two training stages for training this branch introduced in previous section. The training epoch for each stage is 200, 100. For the second stage, the learning rate is one tenth of the one in the first stage. After each alternation, we reduce the learning rate and training epoch of the two branches. And we alternately train this model four times in total.



(a) Experimental results on our model

(b) Experimental results on Mask-RCNN



(c) Experimental results on UFDLD

Fig. 2: The experimental results for different models.

4.4. Results And Analysis

a) Compared with other training methods: For our training methods, two methods are used for comparison. The first training method is that we first train the instance segmentation branch, including the head and backbone, and then freeze the instance segmentation branch and backbone, and only train the lane detection head. We call it comparison method I. The second training method is that we first train the lane detection branch, including the head and backbone, and then freeze the lane detection branch and backbone, and only train the instance segmentation head. We call it comparison method II. Both top1, top2, top3 accuracy [14] for lane detection branch and mAP [2] for instance segmentation branch are compared in this experiment. The results are shown in TABLE 3. From the table, we can see that our method achieves great performance in both branches, and our method achieves comparable performance with comparison method I for instance segmentation branch, comparison method II for lane detection branch. And the comparison methods only achieve great performance in one branch.

Table 3: THE Tops (%),mAPs (%) for different methods

Train method	Top1	Top2	Top3	mAP
Ours method	79.1	96.4	98.0	59.7
comparison I	73.2	91.6	94.6	59.9
comparison II	80.0	96.7	99.2	25.2

b) Compared with separately trained model: we compare the results of our training method with the separately trained model. We train the Mask-RCNN [2] on our road defects dataset and train the Ultra Fast Structure-aware Deep Lane Detection (UFDLD) [14] on our lane dataset. The result for instance segmentation branch is compared with Mask-RCNN [2] and the result for lane branch is compared with UFDLD [14]. It can be seen in TABLE 4. From the table, our method achieves comparable performance with Mask-RCNN [2] for instance segmentation branch, UFDLD [14] for lane detection branch.

Table 4: THE Tops (%),mAPs (%),memory (KB) on different models

Model	Top1	Top2	Top3	mAP	Model memory
Ours	79.1	96.4	98.0	59.7	346655
Mask-Rcnn	--	-	-	59.9	250715
UFDLD	79.7	96.6	98.2	-	239221

We can see that through our training method, our model can achieve two tasks of instance segmentation and lane detection. And both of two branches have great performance. It shows the effectiveness of our proposed training method. The visualizations of our model and other compare models are shown in Fig. 2. We can see our model performs well.

5. Conclusion

In this article, our main contribution is that we propose a new training method. Through our method of alternately training two network branches while continuously reducing the learning rate, we can achieve two different tasks of instance segmentation and lane detection in a network, it can be found that the accuracy of our two branches can be similar with the accuracy of training with two different models from experiments results. This shows the effectiveness of our method. Not only can our method reduce the model memory, but also speed up the detection speed. In addition, we have collected a new road defects data sets for research on road defects problems, it is conducive to the research of road defects, while ensuring the safety of the road.

6. References

[1] J. Hui, C. Changyan, B. Zhaoxu, S. Zhaofeng, and X. Min, “Common interference gpr image and its characteristic analysis in city road defects detection,” Geotechnical Investigation & Surveying, vol. 11, 2012.

[2] K. He, G. Gkioxari, P. Doll´ar, and R. Girshick, “Mask r-cnn,” in Proceedings of the IEEE international

conference on computer vision, 2017, pp. 2961–2969.

[3] Z. Cai and N. Vasconcelos, “Cascade rcnn: Delving into high quality object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6154–6162.

[4] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “Yolact: Real-time instance segmentation,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9157–9166.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C.Berg, “Ssd: Single shot multibox detector,” in European conference on computer vision. Springer, 2016, pp. 21–37.

[7] T.-Y. Lin, P. Doll ár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.

[8] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Doll ár, and C. L. Zitnick, “Microsoft coco captions: Data collection and evaluation server,” arXiv preprint arXiv:1504.00325, 2015.

[9] T. Evgeniou and M. Pontil, “Regularized multi–task learning,” in Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, pp. 109–117.

[10] A. A. Assidiq, O. O. Khalifa, M. R. Islam, and S. Khan, “Real time lane detection for autonomous vehicles,” in 2008 International Conference on Computer and Communication Engineering. IEEE, 2008, pp. 82–88.

[11] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, “Towards end-to-end lane detection: an instance segmentation approach,” in 2018 IEEE intelligent vehicles symposium (IV).IEEE, 2018, pp. 286–291.

[12] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, “Spatial as deep: Spatial cnn for traffic scene understanding,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018.

[13] Z. Qin, H. Wang, and X. Li, “Ultra fast structure-aware deep lane detection,” arXiv preprint arXiv:2004.11757, 2020.

[14] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” arXiv preprint arXiv:1404.2188,2014.

[15] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” arXiv preprint arXiv:1506.01497, 2015.

[16] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[19] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B.Gotway, and J. Liang, “Convolutional neural networks for medical image analysis: Full training or fine tuning?” IEEE transactions on medical imaging, vol. 35, no. 5, pp. 1299–1312, 2016.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Doll´ar, “Focal loss for dense object detection,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[21] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, “Understanding convolution for semantic segmentation,” in 2018 IEEE winter conference on applications of computer vision (WACV). IEEE, 2018, pp. 1451–1460.

[22] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.