# Instance Level Object Segmentation in Videos
## Project Report for Visual Computing Course, Ecole CentraleSupelec, Paris - Spring 2019

Ayush K. RAI
Ecole CentraleSupelec, Paris
ayush.rai2512student-cs.fr

Kai-Wei TSOU
Ecole CentraleSupele, Paris
kai-wei.tsou@supelec.fr

## Abstract

***In this work we present a technique for Instance Level Object Segmentation in videos using popular deep learning architecture Mask-RCNN[10]. Our work[1] is inspired from Kaggle Challenge on Video Segmentation (part of 2018 CVPR Autonomous Driving Workshop). It is important to specify that due to computational limitations we only worked on the subsample of the competition dataset and focused more on the strategies and methodologies adopted to tackle the problem of Instance Segmentation. In keeping alignment with this idea, throughout this report we perform extensive examination of the results obtained by our approach and also present in-depth analysis of the observation we had.***

## 1. Introduction

The problem of Instance level segmentation is about developing a low level understanding of the image at class as well as instance level. Instance Segmentation has plethora of significant applications in Self Driving Cars, Biomedical Imaging and Remote Sensing. Visual Understanding of complex urban street scenes at an Instance level is also an open research problem in computer vision. In the recent times the Instance Level Object Segmentation task has received huge amount of attention not only from the computer vision research community but also from research and development departments of many giant tech firms working in the field of Autonomus Driving. Infact many large scale publicly available datasets like CityScapes[6] and Apolloscape [12] have been released to help the researchers address this problem. Last year a Kaggle Challenge on Video Segmentation was organized as a part of 2018 CVPR Autonomous Driving Workshop. Furthermore Instance Level Segmentation in videos is an evolving problem and has been less studied as compared to Instance Segmentation in Images.

The rest part of this report is structured as follows : Section 2 explains about the related work, Section 3 describes the Task and Dataset description. In the Section 4, we provide the details about our methodology. Section 5 talks about Evaluation whereas Section 6 discusses about Conclusion and Future Work.

## 2. Related Work

The problem of Instance Segmentation in Images is vastly studied by the Computer Vision Research community. Many researchers have tackled Instance Segmentation problem by first addressing the inherent Semantic Segmentation Problem. A lot of work has been done in the field of Semantic Segmentation in images involving graph based image segmentation methods like normalized cut, applying exact and approximate inference techniques by modeling the images as Markov Random Fields (MRF) and Conditional Random Fields (CRF) and deep learning based methods especially in the recent times. Among various approaches used for Semantic Segmentation, we explored the following:

### 2.1. Graph Based Segmentation Approach

The first approach we considered is to apply normalized cut [16]. The advantage of normalized cut is that it considers not only the cut cost but also the total edge connections to all the nodes within a group. In addition, compared with neural network, the computation of normalized cut is relatively cheap. The objective function of two-class normalized cut is shown in equation 4:

$$\begin{aligned} N_{cut}(A,B) &= \frac{cut(A,B)}{asso(A,V)} + \frac{cut(A,B)}{asso(B,V)} \\ &= 2 - \left(\frac{asso(A,A)}{asso(A,V)} + \frac{asso(B,B)}{asso(B,V)}\right) \end{aligned} \tag{1}$$

[1] Github Link: https://github.com/rayush7/CVPR-2018-WAD-Video-Segmentation-Challenge

where $A, B$ are two partitions set, $V$ is a set of all nodes and:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v) \tag{2}$$

$$asso(A, V) = \sum_{u \in A, t \in V} w(u, t) \tag{3}$$

Normalized cut [16] seek to find the minimum loss, or maximum flow, given a graph. The other advantage of normalized cut is that it considers not only the cut cost but also the total edge connections to all the nodes within a group. In addition, compared with neural network, the computation of normalized cut is relatively cheap. We can apply normalized cut algorithm to the segmentation problem by considering an image as a graph. The pixels of the image can be viewed as vertices and they connect with neighboring pixels. If we want to segment the image into foreground and background, we can introduce additional vertices $s$ for background and $t$ for foreground and find the optimal cut according to this graph. The objective function of two-class normalized cut is shown in equation 4:

$$\begin{aligned} N_{cut}(A, B) &= \frac{cut(A, B)}{asso(A, V)} + \frac{cut(A, B)}{asso(B, V)} \\ &= 2 - (\frac{asso(A, A)}{asso(A, V)} + \frac{asso(B, B)}{asso(B, V)}) \end{aligned} \tag{4}$$

where $A$ is the partition set for background, $B$ is for foreground, and $V$ is a set of all nodes and the cut and association functions are defined as follows:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v) \tag{5}$$

$$asso(A, V) = \sum_{u \in A, t \in V} w(u, t) \tag{6}$$

Then normalized cut algorithm can be used to solve this optimization problem. The same concept can be extended to multiple classes scenario. In [3], the authors aim to minimize the objective function which contains unary and pairwise potentials via $\alpha - \beta$ swap and $\alpha$ expansion algorithm, where they iteratively compare each pair of labels and try to achieve some local optimal point. Fig 1 represents the $\alpha - \beta$ swap algorithm in action on 1D Image.

### 2.2. U-Net

The U-net architecture [15] has achieved very good performance on image segmentation applications. So we decided to adopt U-Net to solve video segmentation problem. The original U-Net architecture is similar to the convolutional auto-encoder architecture. The encoder contains repeated application of $3 \times 3$ convolution layer followed by

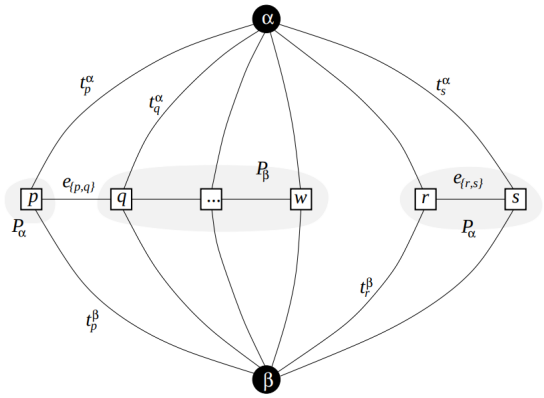


Figure 1. An example of the graph for a 1D image.

ReLU, and $2 \times 2$ max pooling layer. After each convolution layer, U-Net double the channel size. Then The decoder up-sample the obtained encoded map with repeated $3 \times 3$ deconvolution layer and $3 \times 3$ convolution layer. Note that the U-Net also concatenate the corresponding feature maps from the encoder layer with up-sampled feature maps at each deconvolution operation in order to better localize and learn representations. The overall architecture of the U-Net is shown in figure 2. We tried to train the Image using categorical cross entropy Loss and dice Loss but these losses did not converge. Refer to Section 4 for the adopted approach.

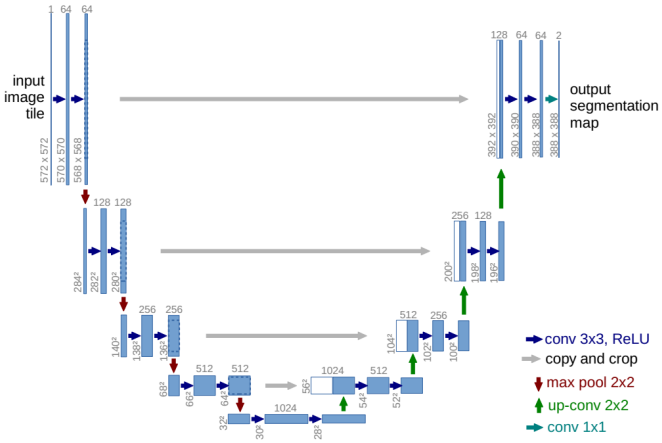


Figure 2. The architecture of U-Net.

## 3. Dataset and Task Description

The **competition dataset (provided by Baidu Inc.)** contains a large number of segmented (groundtruth mask) and original driving images and there are multiple object instances in them. The images are the subsequent frames of videos and there are 44 videos in the training dataset. How-

ever in this competition, only seven different instance-level annotations **car, motorcycle, bicycle, pedestrian, truck, bus, and tricycle** are evaluated. For our work, we randomly sampled 10 videos from the dataset and divided them in train to val ratio of 70:30. Our training dataset contains around 6k images along with their masks with encoded information about the class id and instance id of every instance. Following are few important details about the sampled dataset.

- All the images are the same size (width, height) of the original images
- Pixel values indicate both the label and the instance.
- Each label could contain multiple object instances.
- int(PixelValue / 1000) is the label (class of object)
- PixelValue % 1000 is the instance id

For example, a pixel value of 33000 means it belongs to label 33 (a car) is instance no.0, while the pixel value of 33001 means it also belongs to class 33 (a car) and is instance no.1. These represent two different cars in an image. Fig. 3 represents an example of the Groundtruth Mask.

Figure 3. GroundTruth Mask

### 3.1. Dataset Imbalance Problem

After randomly sampling 10 videos multiple times, we found that the dataset is extremely biased on the Car class as depicted in Fig. 4. This class imbalance problem is addressed by using other datasets like [18], [6] and [7]. However in our work, we did not address this problem because limitations of the amount of compute power we have.

### 3.2. Task Description

The task objective of Instance Level Segmentation (Class and Instance both) is illustrated in the Fig. 5.

## 4. Methodology

Finally we dived into the Region Proposal Based Network Mask-RCNN [10], which inturn helped us to achieve decent results.

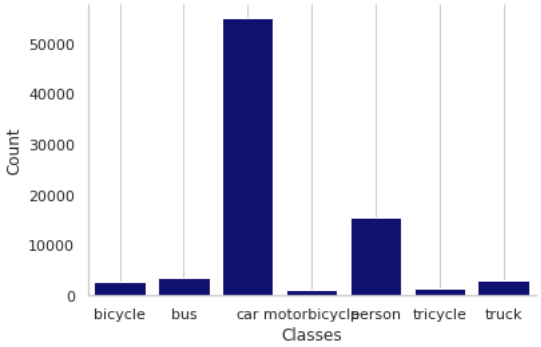


Figure 4. Class Imbalance Statistics

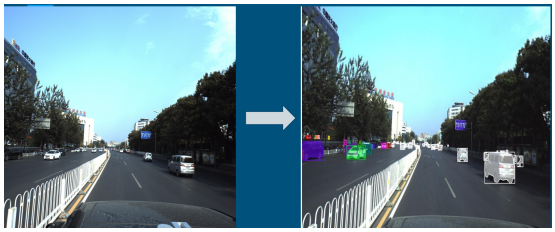

Figure 5. Task Description

### 4.1. Mask R-CNN Based Approach

The main algorithm we use in our approach to solve this problem is [10]. Mask-RCNN inherits alot from many other object detection networks RCNN [9], Fast RCNN [8] and Faster-RCNN [14].

RCNN [9] identifies a manageable number of bounding-box object region candidates by using a technique of Selective Search [17]. And then it extracts CNN features from each region independently for classification. Fig 6 represents RCNN Architecture. Fast RCNN[8] (7) improved this training procedure by unifying three independent models into one jointly trained framework and increasing shared computation results. Instead of extracting CNN feature vectors independently for each region proposal, this model aggregates them into one CNN forward pass over the entire image and the region proposals share this feature matrix. Then the same feature matrix is branched out to be used for learning the object classifier and the bounding-box regressor. Faster-RCNN [14] provides a speedup to Fast R-CNN by constructing a single, unified model composed of RPN (region proposal network) and Fast R-CNN with shared convolutional feature layers. The architecture of Faster R-CNN is highlighted in Fig. 8.

Mask R-CNN [10] extends Faster R-CNN to pixel-level image segmentation. The key point is to decouple the classification and the pixel-level mask prediction tasks. Based on the framework of Faster R-CNN, it added a third branch for predicting an object mask in parallel with the existing

branches for classification and localization. The mask branch is a small fully-connected network applied to each RoI, predicting a segmentation mask in a pixel-to-pixel manner. Because pixel-level segmentation requires much more fine-grained alignment than bounding boxes, mask R-CNN improves the RoI pooling layer (named RoIAlign layer) so that RoI can be better and more precisely mapped to the regions of the original image.

The RoIAlign layer is designed to fix the location misalignment caused by quantization in the RoI pooling. RoIAlign removes the hash quantization, for example, by using x/16 instead of [x/16], so that the extracted features can be properly aligned with the input pixels. Bilinear interpolation is used for computing the floating-point location values in the input. The architecture of Mask RCNN is depicted in Fig 9.

The multi-task loss function of Mask R-CNN combines the loss of classification, localization and segmentation mask:

$$\mathcal{L}_{\text{loss}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{bbox}} + \mathcal{L}_{\text{mask}}$$

$$\mathcal{L}_{\text{cls}} = \frac{1}{N_{\text{cls}}} \sum_i -p_i^* \log p_i - (1 - p_i^*) \log(1 - p_i)$$

$$\mathcal{L}_{\text{bbox}} = \frac{\lambda}{N_{\text{box}}} \sum_i p_i^* \cdot L_1^{\text{smooth}}(t_i - t_i^*)$$

$p_i$ represents predicted probability of anchor i being an object, $p_i^*$ represents ground truth label (binary) of whether anchor i is an object, $t_i$ represents predicted four parameterized coordinates, $t_i^*$ represents ground truth coordinates, $N_{\text{cls}}$ is the normalization term, set to be mini-batch size ( 256) in the paper, $N_{\text{bbox}}$ is the normalization term, set to the number of anchor locations ( 2400) in the paper, $\lambda$ is the balancing parameter, set to be 10 in the paper.

$$\mathcal{L}_{\text{mask}} = -\frac{1}{m^2} \sum_{1 \le i,j \le m} \left[ y_{ij} \log \hat{y}_{ij}^k + (1 - y_{ij}) \log(1 - \hat{y}_{ij}^k) \right]$$

where $y_{ij}$ is the label of a cell(i, j) in the true mask for the region of size $m \times m$; $\hat{y_{ij}}$ is the predicted value of the same cell in the mask learned for the ground-truth class k.

The mask branch generates a mask of dimension m x m for each RoI and each class; K classes in total. Thus, the total output is of size $K.m^2$. Because the model is trying to learn a mask for each class, there is no competition among classes for generating masks. $\mathcal{L}_{\text{mask}}$ is defined as the average binary cross-entropy loss, only including k-th mask if the region is associated with the ground truth class k.

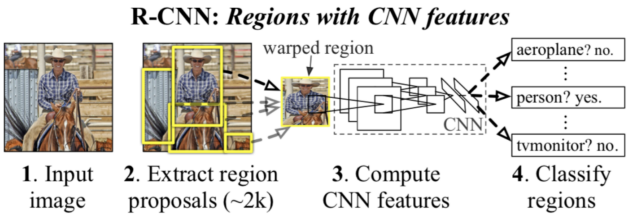


Figure 6. RCNN Architecture

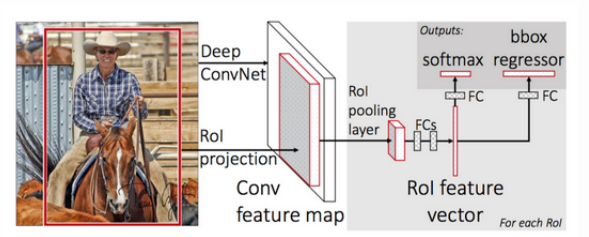


Figure 7. Fast RCNN Architecture

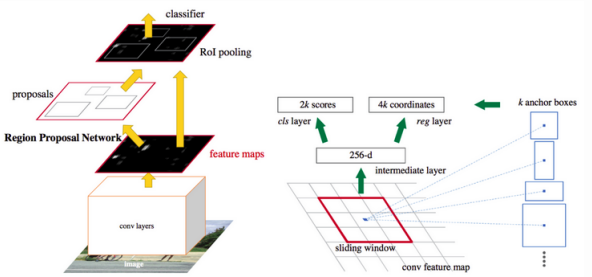


Figure 8. Faster RCNN Architecture

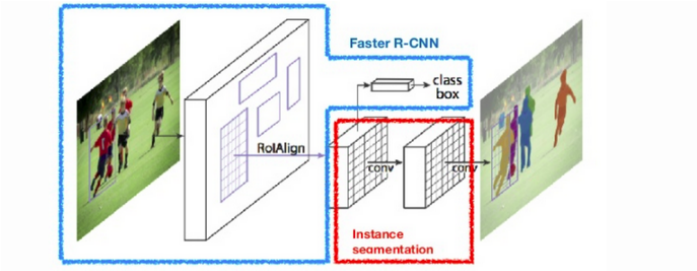


Figure 9. Mask RCNN Architecture

## 4.2. Details of Training

We use an open source implementation of Mask-RCNN [2] in Tensorflow [1] with backbone architecture of ResNet50 [11]. The original width and height of each frame is 3384 and 2710 respectively. As a preprocessing step we cropped 28 pixels from left and right whereas cropped 75 pixels from top and bottom and finally downscaled the image and its mask by a factor of 4. The resulting images had width of 832 and height of 640. For training we use the adam optimizer with the initial learning rate of 0.001, learning mo-

mentum of 0.9 and weight decay of 0.0001. The region proposal network Anchor Scales choosen by us are (8, 16, 32, 64, 128). After several attempts of training the model from scratch on our dataset, we realized that this strategy isn't productive therefore we decided to use a pretrained Mask-RCNN model (trained on COCO [13] dataset) and finetune the final layers on our sampled dataset. This was also very useful because COCO dataset has 91 object categories and many of the categories in our dataset (like pedestrian) already have some similar category in the COCO dataset like person. We trained the model for 100 epochs, with the adam optimizer with . The whole training process took around 16 hours on Nvidia GTX 1060 GPU.

## 5. Evaluation and Results

We use Intersection over Union (IOU) metric per class instead per Instance for evaluation in order to lessen the complexity of the task. The IoU between a predicted instance A and a ground truth instance B is computed by:

$$IoU(A, B) = \frac{A \cap B}{A \cup B}$$

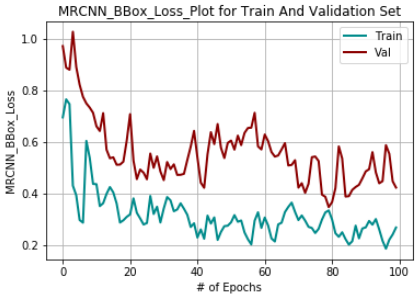


Figure 10. Bounding Box Loss Plot

Table 1 shows the mean Intersection Over Union (IOU) per Class in the Validation dataset. We achieved good results[2] on the Car Class with a mean iou score of 0.2899, low score for Pedestrian, Truck and Bus whereas our model achieved a mean iou score of 0.0 for classes like Motorbicycle, Bicycle and Tricycle.

Fig. 10 shows the plot of bounding box loss versus number of epochs for training and the validation datasets. We observe that the bounding box loss for training and validation decreases with increase in number of epochs. However we notice that the classification loss represented in Fig.11 is diverging. We strongly believe that this is because of the class imbalance problem in the dataset. It might also be the reason that even though we are able to detect masks for motorcycle, bicycle and tricycle but they might be getting classified into a different class and therefore their mean IOU score of 0.0 in Table 1. This is something that we still need to rectify and work on.

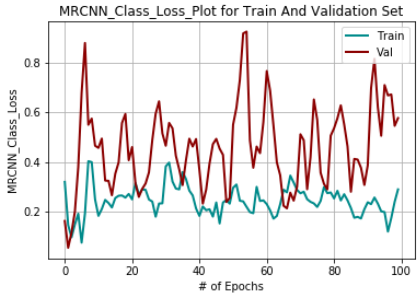


Figure 11. Classification Loss

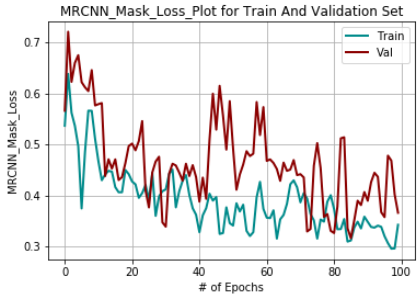


Figure 12. Mask Loss

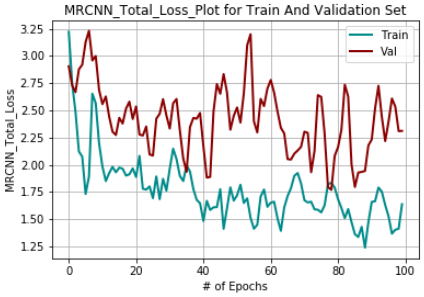


Figure 13. Total Loss

From Fig. 12 we observe that the mask loss is also decreasing with the increase in the number of epochs. A similar behaviour is observed in Region Proposal Network (RPN) classification and bounding Box loss highlighted in Fig. 15 and 14 respectively.

We would like to comment that the weighted aggregation of the classification, bounding box and mask loss with more

[2]A sample video of our results can be found here : https://www.youtube.com/watch?v=nRgVNnI4-AM


Figure 14. RPN BBox Loss Plot

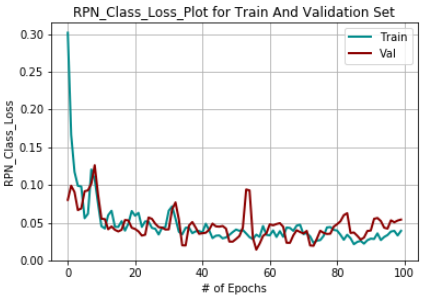


Figure 15. RPN Class Loss Plot

| Class | Mean IOU Score |
|---|---|
| Car | 0.2899 |
| Motorbicycle | 0.0 |
| Bicycle | 0.0 |
| Pedestrian | 0.0096 |
| Truck | 0.0035 |
| Bus | 0.0132 |
| Tricycle | 0.0 |

Table 1. Mean IOU Score of Evaluated 7 Classes (Validation Set)

weight on bounding box loss and mask loss as compared to classification loss could have helped us to improve Mean IOU score on the classes like motorbicycle, bicycle, tricycle.

## 6. Conclusion and Future Work

In our project, we addressed the problem of Instance Segmentation in Videos using popular deep learning architecture of Mask-RCNN [10]. Instance segmentation in videos is still an evolving problem and has been less studied as compared to Instance Segmentation in Images. In our project we did not work on utilizing the temporal information present in the susbsequent frames of the video. Therefore our future work involves diving into strategies that use the temporal information in the video to achieve better results like [5] and [4].

## References


[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016. 4

[2] W. Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017. 4

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 377–384. IEEE, 1999. 2

[4] S. Chandra, C. Couprie, and I. Kokkinos. Deep spatio-temporal random fields for efficient video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8915–8924, 2018. 6

[5] K. Chen, J. Wang, S. Yang, X. Zhang, Y. Xiong, C. C. Loy, and D. Lin. Optimizing video object detection via a scale-time lattice. *arXiv preprint arXiv:1804.05472*, 2018. 6

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 3

[7] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017. 3

[8] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 3

[9] 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*, pages 580–587, 2014. 3

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 1, 3, 6

[11] 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*, pages 770–778, 2016. 4

[12] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang. The apolloscape dataset for autonomous driving. *arXiv preprint arXiv:1803.06184*, 2018. 1

[13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[14] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3

[15] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In

*International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. 1, 2

[17] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 3

[18] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018. 3