Value of Temporal Dynamics Information in Driving Scene Segmentation

Li Ding[1], Jack Terwilliger[1], Rini Sherony[2], Bryan Reimer[1], and Lex Fridman[*1]

[1]Massachusetts Institute of Technology (MIT)
[2]Collaborative Safety Research Center, Toyota Motor North America

Abstract

Semantic scene segmentation has primarily been addressed by forming representations of single images both with supervised and unsupervised methods. The problem of semantic segmentation in dynamic scenes has begun to recently receive attention with video object segmentation approaches. What is not known is how much extra information the temporal dynamics of the visual scene carries that is complimentary to the information available in the individual frames of the video. There is evidence that the human visual system can effectively perceive the scene from temporal dynamics information of the scene's changing visual characteristics without relying on the visual characteristics of individual snapshots themselves. Our work takes steps to explore whether machine perception can exhibit similar properties by combining appearance-based representations and temporal dynamics representations in a joint-learning problem that reveals the contribution of each toward successful dynamic scene segmentation. Additionally, we provide the MIT Driving Scene Segmentation dataset, which is a large-scale full driving scene segmentation dataset, densely annotated for every pixel and every one of 5,000 video frames. This dataset is intended to help further the exploration of the value of temporal dynamics information for semantic segmentation in video.

1. Introduction

Forming appearance-based representations of still images with convolutional neural networks (ConvNets) has been successfully used in object classification, detection, and segmentation tasks [13, 39, 10]. These representations can be formed via supervised, semi-supervised, or unsupervised methods [33]. The primary source of information in these approaches is the visual characteristics of a single image. In contrast, there is evidence that the human visual system largely relies, in real-world perception tasks, on temporal dynamics information [31], not merely as pre-processing support for object segmentation but as the primary source of information used for dynamic scene understanding.

Put another way, understanding a static scene in a single image may be a fundamentally different task than understanding a dynamic scene in video. From this distinction, a number of efforts focused on video object segmentation have recently emerged [4, 38, 5]. What is not well understood is the degree to which temporal dynamics of the visual scene in video can contribute to the video scene segmentation task, and consequently the dynamic scene understanding task.

Our work helps explore the value of temporal dynamics in driving scene segmentation by formulating the appearance-based and temporal-based as a joint learning problem that reveals the importance of each component for the effective segmentation of various parts of the driving scene. In addition, we provide the MIT Driving Scene Segmentation dataset, which is a large-scale full driving scene segmentation dataset, densely annotated for every pixel and every one of 5,000 video frames. The purpose of this dataset is to allow for exploration of the value of temporal dynamics information for full scene segmentation in dynamic, real-world operating environments.

2. Related Work

Semantic segmentation is the most fine-grained form of two-dimensional image region classification (in contrast to image classification and object detection), and is currently considered to be the frontier of open challenges in computer vision that seek to interpret visual information. We consider the work on semantic segmentation in still images and in video separately.

2.1. Semantic Segmentation in Still Images

The task of semantic segmentation involves assigning each pixel in the image a label. For object segmentation,

---

[*]Corresponding author: `fridman@mit.edu`

a distinct between foreground and background objects is drawn. For full scene segmentation, both foreground and background objects must be classified at the level of a pixel [15, 28, 12, 53, 6]. Over the past five years, several key adjustments to ConvNet-based architectures have been made to improve segmentation accuracy. First, dilated convolution (also known as atrous convolution) have been added to address the reduction of resolution due to pooling or convolution striding while still being able to learn increasingly abstract feature representations [8, 49, 9, 10]. Second, several methods have been proposed to deal with the existence of objects at multiples scales, including (1) image pyramids that deal with the problem at the image level [17, 14, 36, 24, 11, 9], (2) encoder-decoder structure which deals with the problem at the multi-scale feature level [1, 40, 37, 34], (3) cascading extra modules that deals with the problem by capturing long-range context [21, 8, 52, 24, 43, 26, 49], and (4) spatial pyramid pooling that deals with the problem by using filters and pooling operations of various rates and sizes [9, 51].

Overall, progress in semantic segmentation of still images has continued [10], and it is possible that eventually any approach to semantic segmentation in video will eventually completely disregard temporal dynamics of the scene, as it has for the state-of-the-art tracking by detection approaches. However, this possible eventuality is far from guaranteed, and is currently one of the open problems of computer vision: how valuable is temporal dynamics information for scene understanding in video? Our work seeks to take steps toward answering this question.

2.2. Semantic Segmentation in Video

Most of the work in semantic segmentation has been on the problem of *video object segmentation* where a distinction is drawn between foreground and background objects, and the task is focused on temporal propagation of foreground object segmentation. A wide variety of approaches have been proposed for this task. The first set of approaches groups pixels spatiotemporally based on motion features computed along individual point trajectories [3, 30, 20, 18]. These approaches rely on successful feature matching in the temporal domain, and fail when such matching is intermittently erroneous. The second set of approaches formulates segmentation as a foreground-background classification task, detecting regions that correspond to foreground objects and matching the resulting appearance models with other information such as salience maps, shape estimates, and pairwise constraints [32, 22, 46]. The third set of approaches incorporate the classification approach with a memory module for propagating region estimates in time [44, 48]. The latter set of approaches begin to incorporate temporal dynamics information into the learning problem, and due to their success, motivate the method, observation,

and dataset proposed in our work. The primary benchmark dataset used for semantic segmentation in video to date is DAVIS [38, 5]. Our dataset differs in three ways: (1) it provides full scene segmentation not just foreground object segmentation, (2) it includes only driving scenes from the perspective of the ego-vehicle, and (3) it is densely annotated in time for long period at 30 fps.

3. Method

Deep neural networks being pre-trained on large-scale datasets show a better capability of generalization after fine-tuning even without using any temporal information [4, 27]. In addition, [7] also shows that there is a considerable benefit in pre-training on large video dataset other than only still images. However, in the context of video scene parsing, it is too costly to obtain a densely annotated video dataset at large-scale. In order to address the advantage of both pre-trained networks on still images and pre-trained networks on videos, our method focus on combining the appearance network (pre-trained on still images) and the memory network (pre-trained on videos) in a meaningful way, while under the assumption that the appearance network itself already well-performs.

3.1. Generalizing Semantic Visual Memory

Recent work successfully use convolutional recurrent networks, *e.g.* Conv-LSTM [47] and Conv-GRU [44], to address spatiotemporal sequence modeling problem. However, due to the requirement of large-scale data in training deep neural networks, such models usually either have to hold a insufficient small number of hidden states or suffer the lack of ability of generalization. In this case, we build a memory network using Conv-LSTM taking the deep semantic feature map from a pre-trained appearance network as input, and pre-train it again on a large-scale video segmentation dataset. Specifically, we adopt the DeeplabV3 network [10] as the appearance network, and use a Conv-LSTM with 256 hidden units as the memory network. By freezing the weights of the appearance network when training the memory network, we avoid the risk of losing the ability of generalization when feeding highly redundant dense video frames.

3.2. Prediction Refinement with Confidence Gates

From the sense of human perception, it is intuitive that some parts of the scene are easier to perceive statically, others with motion instead. However, it is unclear how this should be managed and distributed throughout the whole image. We design the confidence gate of prediction update, inspired by the way how LSTM updates its cell states.

For each frame, given the prioritary class prediction from appearance network $logits_{appr}$, and the prediction from memory network $logits_{mem}$, which is obtained from a 1x1
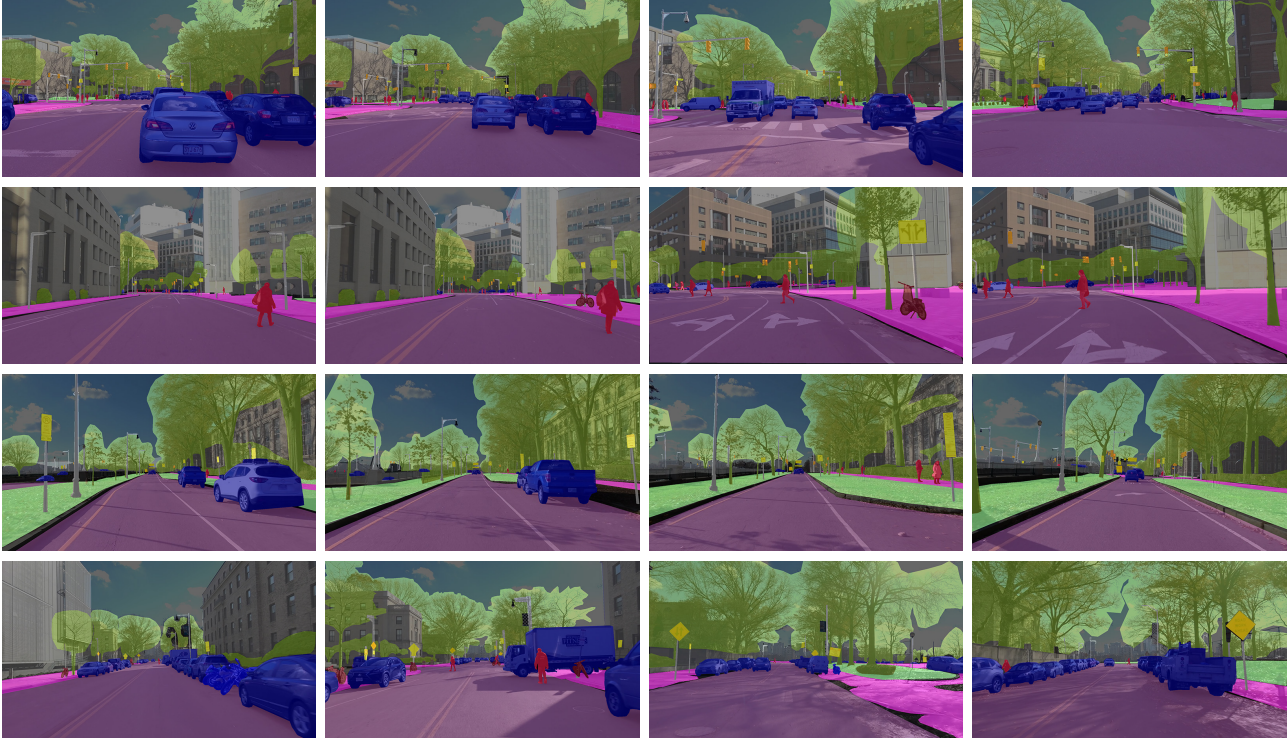
Figure 1: Examples from the proposed MIT Driving Scene Segmentation dataset. Annotations are overlayed on frames.

Conv layer after Conv-LSTM, the final prediction is calculated as

$$logits = \sigma_{appr} \cdot logits_{appr} + \sigma_{mem} \cdot logits_{mem}$$

where σ_{appr} and σ_{mem} are two spatial sigmoid gates, calculated from two 1x1 Conv layers taking the concatenation of appearance network feature and Conv-LSTM output as input. The gates control how to distribute the weights, *i.e.* confidence on the results provided by both parts of the system. The appearance network except for the last layer is frozen during training due to the risk of over-fitting, although the system is end-to-end trainable.

3.3. Training Process

The goal for our training process is to combine state-of-the-art methods developed on both still images and videos, while at the same time preserve the natural advantage of both. The target testing case is doing semantic segmentation on video frames, while the training can be done either on still images only or with video frames. However, the memory network requires sequences of video frames for training, which should have at least one frame annotated.

In this case, we first adopt the DeeplabV3 pretrained on still images as feature extractor, and train the memory network on a video dataset. Then we fine-tune the memory network on the target dataset with multiple frames but only

calculate the loss from the last frame, where the ground truth segmentation exists. Finally, we fine-tune the whole system with confidence gates on the output from both appearance network and memory network, while not changing the weights in feature extractor layers.

To further explain our idea in the above design, we consider the scenario that the appearance network is good enough, but there will be some edge cases that can not be dealt with using only one still image, such as occlusion, cut on image margin, motion blur, *etc*. So the memory network is to help with those cases, and the confidence gates are used to control the merging of information.

4. Dataset

Since our target problem is video scene parsing in the driving context, we group current datasets into three categories: 1) Pixel-wise annotation of still images, *e.g.* Mapillary Vistas [29]; 2) Pixel-wise annotation of coarse video frames, *e.g.* Cityscapes [12], BDD [50]; 3) Pixel-wise annotation of dense video frames, *e.g.* the MIT Driving Scene Segmentation dataset in this work. Category 1 is the easiest to obtain and get larger variability in different scenes. Category 3 usually lack variability, but is the most suitable to do temporal modeling.
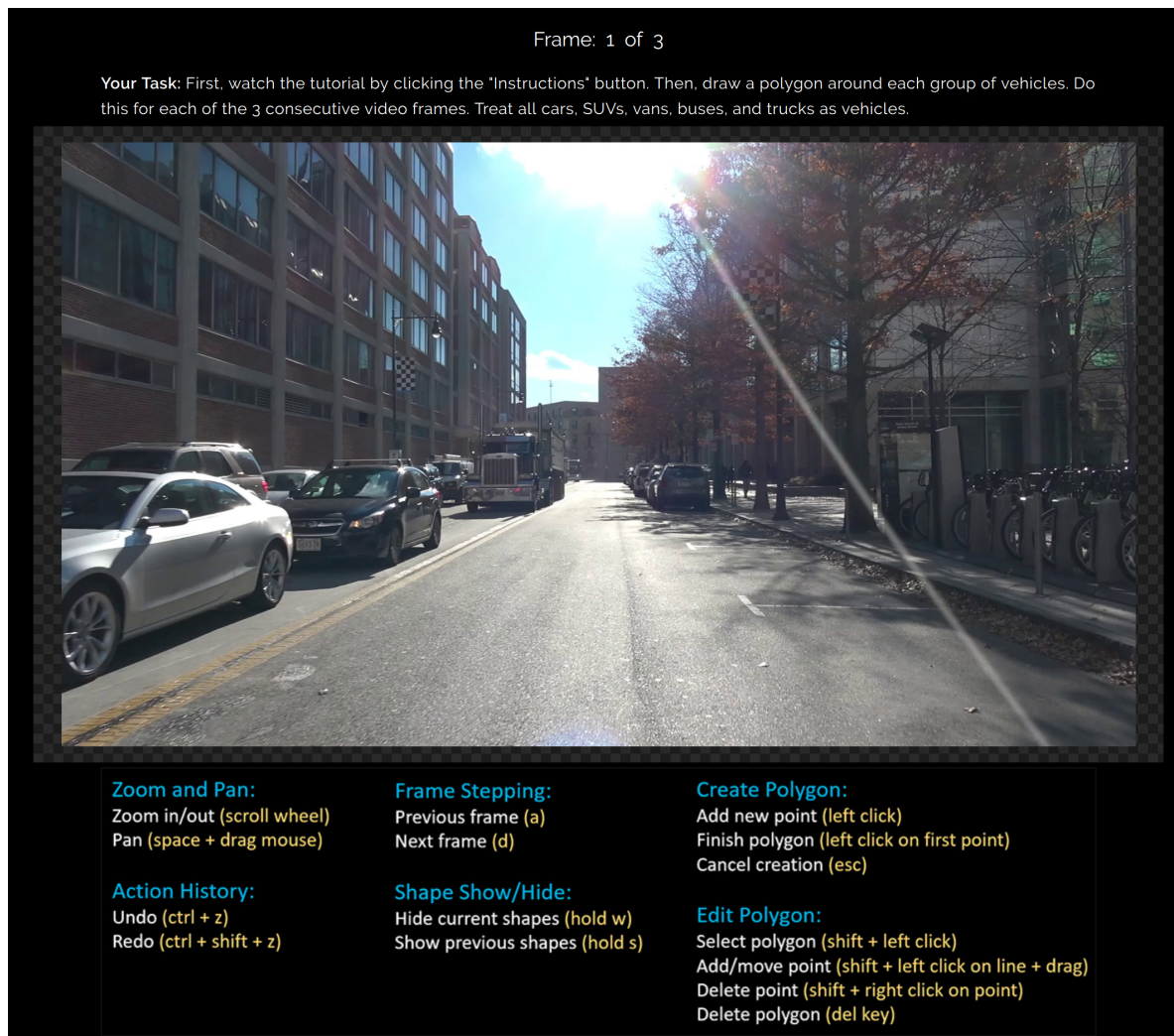
Figure 2: Front-end of our annotation tool.

4.1. Dataset Selection

There are many large-scale datasets with semantic pixel-wise annotations, *e.g.* Pascal VOC [16], MS COCO [25], ADE20k [53], but more about natural scene/object. In the driving context, there are several well-developed datasets with dense semantic annotations, *e.g.* CamVid [2], KITTI [19], Cityscapes [12], Mapillary Vistas [29], and recently BDD [50]. Among the above datasets, Mapillary Vistas contains 25k images with fine annotation, which is the largest value, but the images are all still images, *i.e.* without temporal connection between each other. Both Cityscapes and BDD choose one frame from each short video clip to cast fine annotation. It is a common approach to make the dataset such way in order to mostly capture the variability of scenes with the least amount of budget. However, this phenomenon also leads current research to focus on single-frame algorithms, ignoring the rich temporal information contained between consecutive frames.

In this case, we collect and annotate a novel dataset, which has over 10k frames with fine annotation, from a single, untrimmed video of the front driving scene at 30 fps. The similar idea can be seen in CamVid, which also features annotated video frames, but only at a low frequency (1 fps) and a small scale (less than 500 frames totally). Out of driving scene domain, DAVIS [35] has densely annotated pixel-wise semantic object annotations for trimmed video clips, each at around 60 frames. Similarly, SegTrack [45] and SegTrack v2 [23] also feature pixel-wise video object annotation.

In order to make our approach comparable with other recent work, we choose Cityscapes as our main source of data in experiments. Cityscapes is the largest dataset focused on urban street views, which contains 5k finely-annotated and

20k coarsely-annotated images. The MIT Driving Scene Segmentation dataset is mainly used for pre-training in this work, which is described in detail in Sec. 4.2

4.2. MIT Driving Scene Segmentation Dataset

The two main purposes of the development of MIT Driving Scene Segmentation dataset are: 1) experiment and develop the full-scene annotation system that is scalable with a large pool of workers, *e.g.* on Amazon Mechanical Turk (MTurk); 2) create an open-source densely-annotated video driving scene dataset that can help with future research in various fields, *e.g.* spatiotemporal scene perception, predictive modeling, semi-automatic annotation process development. In this case, we collect a long, untrimmed video (6:35, 11869 frames) at 1080P (1920x1080), 30 fps, which is a single daytime driving trip, and annotate it with fine, per-frame, pixel-wise semantic labels. Examples from the dataset are shown in Fig. 1. Some other kinds of driving-related metadata such as IMU, GPS are also available.

Existing driving scene datasets rely on hiring a very small group of professional annotators to do full frame annotation, which usually takes around 1.5 hours per image. [12, 29] This is reasonable because the pixel-wise annotation is a task of high-complexity that reach the limits of human perception and motor function without specific training. However, in order to have scalability and flexibility to annotate in potentially much larger scale, we develop the annotation tool that is web-based and understandable. We deploy our tool on MTurk, which contains a large pool of professional and non-professional workers.

4.2.1   Annotation Tool

To support fine and quick annotations, we develop a web-based annotation tool following the common polygon annotator [41] design with the implementation of techniques such as zoom in/out and keyboard shortcuts. A screenshot of our annotation tool is shown in Fig 2. To further promote the accuracy and efficiency of annotation processes, we also address several problems during the process and design specific improvements.

Task complexity.   We found that annotators had difficulty when asked to annotate an entire scene at once, which involved keeping in mind many object classes and keep working for a non-flexible long time. In response, we divided the task of annotating an entire scene into multiple subtasks, in which an annotator is responsible for annotating only 1 object class. We found that this largely reduces classification errors, improves the quality of our annotations, and reduces the time required to fully annotate each scene.

Limits of human perception.   We found many misclassification errors are due to the difficulty of recognizing

objects in still scenes, but that these errors appeared obviously incorrect in the video. In response, we designed the tool such that an annotator could step through consecutive frames quickly with keyboard shortcuts. The motion perceived when stepping through frames, reduce classification errors.

4.2.2   Annotation Process

The intuition behind our annotation process, is that small simple tasks are preferable to large complex tasks. By breaking down the semantic segmentation task into subtasks so that each worker is responsible for annotating only part of a scene, the annotations are: 1) easier to validate 2) easier and more efficient to annotate and 3) higher quality. To accomplish this, we divide the work of annotating the video into tasks of 3 frames in which a worker is asked to draw polygons around only 1 class of object, *e.g.* vehicles. Our annotation process involves 4 stages: 1) task creation 2) task distribution 3) annotation validation and 4) the assembly of sub-scene annotations into full-scene annotations.

For stage 1, the creation of tasks, we label which frames contain the classes we are interested in and group the frames into sets of 3. This stage removes cases where a worker is asked to annotate the presence of a class which is not present in a frame. Since this stage only requires labeling frame numbers in which a member of a particular class enters a visual scene and frame numbers in which the last member of a particular class leaves, it is much faster and cheaper than creating a semantic segmentation task for every frame and let annotators find out the class does not exist. This approach creates significant time-and-cost-savings especially for rare classes, such as motorcycles in our case.

For stage 2, the distribution of tasks, we submit our tasks to MTurk and specify additional information which controls how our tasks are distributed:

- Reward: This is the amount of money a worker receives for completing our task. We specify different rewards for different classes based on the estimated duration and effort in the annotation.

- Qualifications: This allows us to limit the pool of workers who may work on our tasks based on 1) the workers approval rate, or rate of successful annotation, calculated from all a workers work on the MTurk platform. 2) the total number of tasks the worker has completed. 3) the qualification task we designed for every new worker taking our task for the first time, which is a test task that can be evaluated with the known ground truth.

For stage 3, annotation validation, we use both automated and manual processes for assessing the quality of worker annotations. In addition to the first qualification

Method	Road	Sidewalk	Building	Wall	Fence	Pole	Traffic Light	Traffic Sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	Mean IoU
*DeeplabV3 [10]	97.5	81.0	90.3	38.4	53.8	50.8	61.4	71.3	91.0	58.9	93.0	76.3	53.2	93.2	69.2	75.7	63.7	55.2	72.6	70.9
Ours (Cityscapes only)	97.6	81.3	90.5	41.0	55.0	50.6	61.6	71.0	91.0	61.5	92.9	76.1	51.8	93.2	68.9	75.6	62.4	54.4	72.4	71.0
Ours (full)	97.6	81.3	90.6	44.5	55.4	51.0	61.8	71.3	91.0	62.0	92.9	76.2	52.8	93.4	70.2	77.1	64.9	55.6	72.4	71.7

Table 1: Quantitative results on Cityscapes dataset. The proposed model achieve better results on stuff classes, *e.g.* sidewalk, wall, fence, terrain. The improvement is mostly contributed by the memory network, since we fix the weights of all the feature extractor layers in the appearance network. (*We use the officially released DeeplabV3 model checkpoint with MobileNetV2 [42] backbone pre-trained on MS-COCO [25]. The same below.)

task, workers are assigned additional test tasks occasionally, which are indistinguishable from non-ground truth tasks, to check whether they are still following our instructions. If the workers annotation deviates significantly from the ground truth, they are disqualified from working on our tasks in the future. The process of comparing workers annotations with the ground truth is automated, by calculating the Jaccard distance. The threshold score is class dependent since it is easier to score high on less-complex objects like the road than pedestrians. For our manual validation process, we visually validate that a workers annotations are of sufficient quality, using a tool which steps through annotated frames as a video player which allows approving/rejecting work and blocking workers via key presses.

For stage 4, the merging of sub-annotations, we combine the class-level annotations for a given frame into a full-scene annotation. For this task, we automatically compose the final full-scene annotation one class at a time. Our algorithm first draws the background classes, such as road and sidewalk, and then stationary foreground objects, such as poles and buildings, and finally dynamic foreground objects such as pedestrians and vehicles. In order for this to work, we carefully designed the instructions for each class so that they could fit together harmoniously. The order in which we draw the classes dictates the instructions. When annotating the ith class of n total classes, a worker must annotate the boundaries between objects of class i and classes j where $j >= i$. In other words, if we draw the road annotations before vehicle annotations, workers do not need to draw the boundary between road and vehicle when annotating road, since this work will be handled by workers annotating vehicles.

5. Experimental Results

We do experiments on Cityscapes and the proposed MIT Driving Scene Segmentation dataset, following the training process described in Sec. 3.3. The quantitative and qualitative results are both reported on the validation set of Cityscapes.

5.1. Quantitative Results

The quantitative results are calculated using the evaluation scripts provided by [12], shown in Table 1. For Cityscapes only model, we train the memory network from scratch using only sequences of frames and calculate the loss on the last frame with ground truth. Note that our model is designed to be causal, not using future information, which is capable to run in real time.

The two major findings from the quantitative results are: 1) With the memory network pre-trained on the proposed MIT Driving Scene Segmentation dataset, the overall performance gets improved. The improvement is mostly contributed by the memory network, since we fix the weights of all the feature extractor layers in the appearance network. In other words, even with a fixed appearance network that is already well-trained on still images, we can still find clues to improve it from the temporal domain. 2) By adding the memory network, the model performs better on stuff classes, *e.g.* sidewalk, wall, fence, terrain, which indicates that these classes are preferred by the memory network. We have further exploration on this point described in the next section with qualitative examples.

5.2. Qualitative Results

The purpose of getting qualitative results is to further reveal and help understand the advantage of having a memory network to model spatiotemporal information.

5.2.1 Border Denoising for Stuff Classes

We visualize some of the cases where there are visible improvement of our results over the baseline DeeplabV3, as shown in Fig. 3. The baseline model fails to predict some stuff classes on the border area of the image, possibly due to camera effect, motion blur, or lack of context. Our proposed method is able to reduce this kind of mistakes by taking consideration of spatiotemporal context with the memory
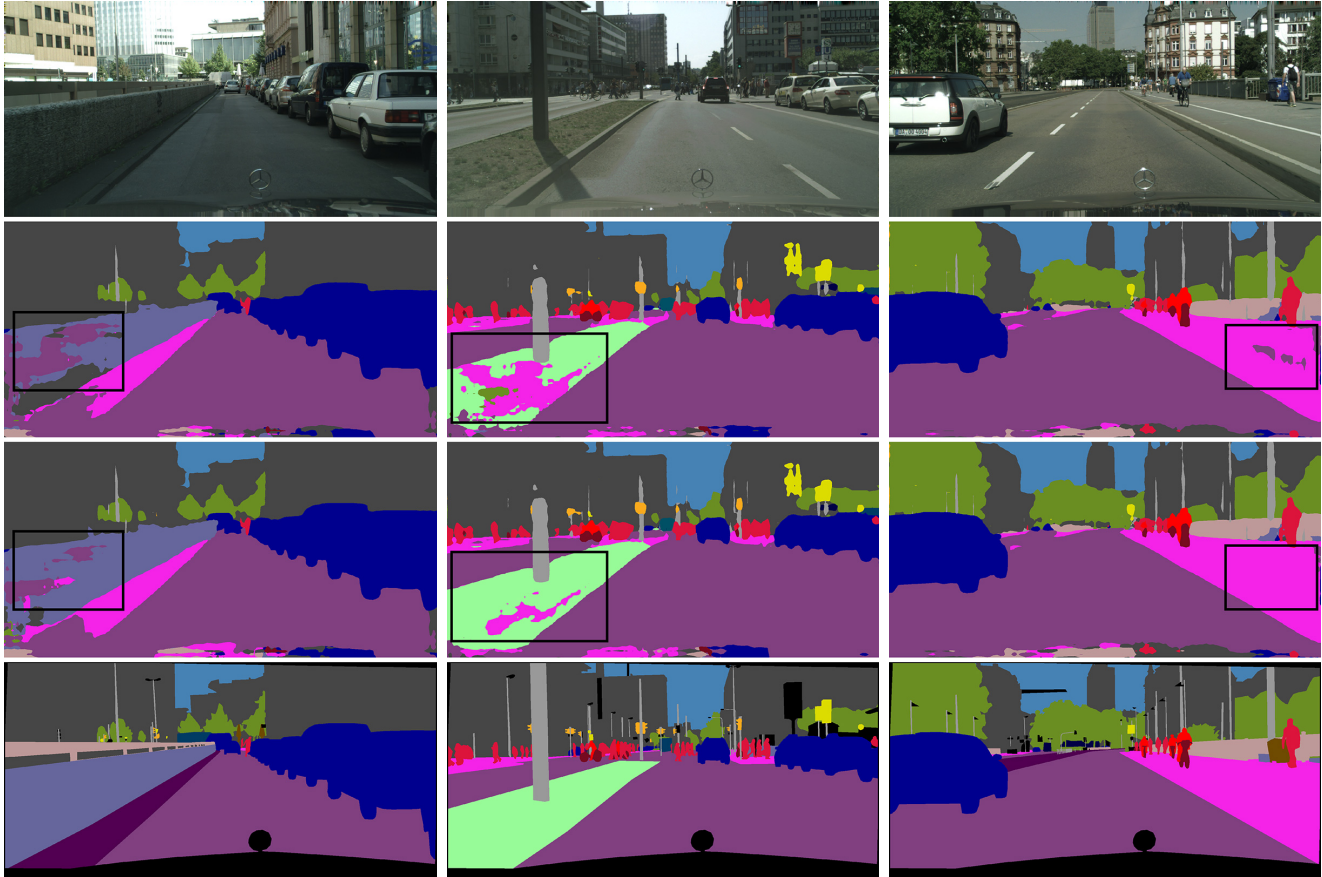
Figure 3: Visualization of segmentation results. From top to bottom: input image, DeeplabV3 results, our results, ground truth. Highlighted areas show the improvement on border area with stuff classes (from left to right: wall, terrain, sidewalk).

network. This finding aligns with the quantitative results where we find most of the improvement of our method lies in the stuff classes. It is important to take this consistency problem seriously in the driving domain, since the driving decisions are always made within a very short time, and the image quality is hard to always maintain due to the variability of dynamic driving scene.

5.2.2 Confidence Gates as Distributed Attention on Spatial v.s. Spatiotemporal Information

One of the general problems in video modeling is to extract useful spatiotemporal information in order to help with recognition. Although it is intuitive that videos always contain more information than still images, they also introduce redundancy and noise. Thus, it is likely that the memory network using spatiotemporal information in some cases performs not as good as the appearance network using spatial information only. To address this problem, our method uses confidence gates to control the ensemble of final output. As shown in Fig 4, the network has more confidence on

the appearance network for foreground objects, *e.g.* person, car, pole. On the other hand, for background stuff objects such as road, sidewalk, terrain, the memory network gets more confidence. Since the values of gates are learned during training, they show the capability of two networks predicting certain objects, which also indicates the underlying difference of spatial and spatiotemporal features.

This finding from another perspective explains the improvement of our method gained on certain stuff classes. The gates serve a role of an attention mechanism that not only lets the network focus on predicting certain classes using more preferred feature, either spatial or spatiotemporal, but also helps interpret how the deep learning model deals with video data. One possible explanation is that some stuff classes are more context-dependent than appearance-dependent, and the spatiotemporal features encodes more context information. We believe there are much more interesting ideas in spatiotemporal modeling, and expect future research to in this area to be fruitful for both understanding the problem of perception and for improving the accuracy and robustness of real-world perception systems.
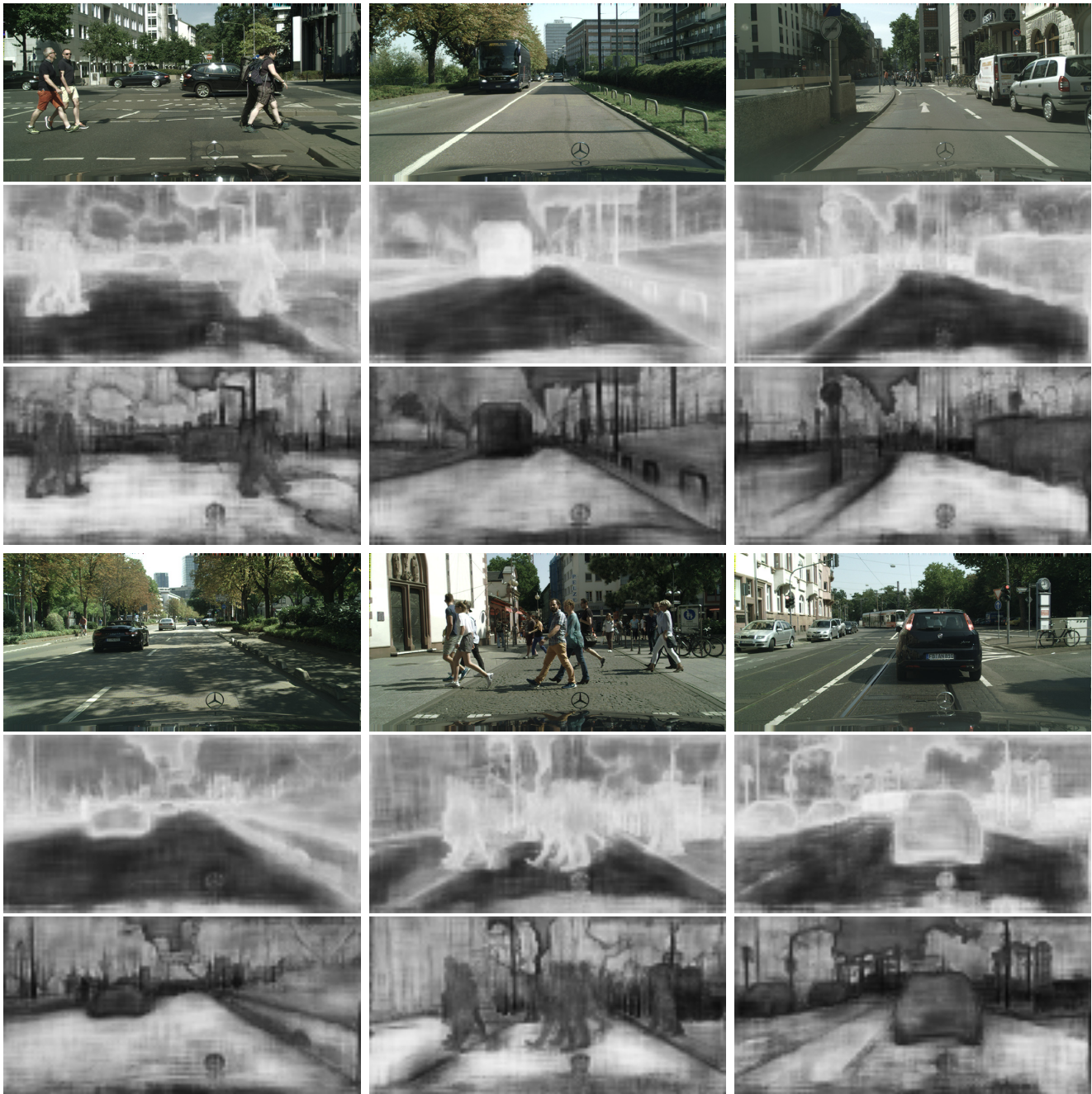
Figure 4: Visualization of confidence gates values. From top to bottom: input image, confidence gate on appearance network σ_{appr}, confidence gate on memory network σ_{mem}. The values are visualized as 0 to black and 1 to white, showing that the network tends to have more confidence on the appearance network for foreground objects, *e.g.* person, car, pole, but more on memory network for background stuff objects, *e.g.* road, sidewalk, terrain.

6. Conclusion

We show that temporal dynamics information in video of driving scenes contains valuable information for the task of semantic segmentation. In particular, we find that background classes are more commonly context-dependent and thus benefit from memory models more than from appearance models. And conversely, foreground objects are more accurately segmented from appearance information, and do not benefit as much from modeling the object's trajectory in time. The MIT Driving Scene Segmentation dataset re-

leased with this work is used to show the value of temporal dynamics information in this paper and allows the computer vision community to explore modeling both short-term and long-term context as part of the driving scene segmentation task.

Acknowledgments

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015. 2

[2] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009. 4

[3] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *European conference on computer vision*, pages 282–295. Springer, 2010. 2

[4] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *CVPR 2017*. IEEE, 2017. 1, 2

[5] S. Caelles, A. Montes, K.-K. Maninis, Y. Chen, L. Van Gool, F. Perazzi, and J. Pont-Tuset. The 2018 davis challenge on video object segmentation. *arXiv preprint arXiv:1803.00557*, 1(2), 2018. 1, 2

[6] H. Caesar, J. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. *CoRR, abs/1612.03716*, 5:8, 2016. 2

[7] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4724–4733. IEEE, 2017. 2

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 2

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 2

[10] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1, 2, 6

[11] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3640–3649, 2016. 2

[12] 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. 2, 3, 4, 5, 6

[13] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014. 1

[14] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015. 2

[15] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 2

[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 4

[17] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013. 2

[18] K. Fragkiadaki, G. Zhang, and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1846–1853. IEEE, 2012. 2

[19] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 4

[20] M. Keuper, B. Andres, and T. Brox. Motion trajectory segmentation via minimum cost multicuts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3271–3279, 2015. 2

[21] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011. 2

[22] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1995–2002. IEEE, 2011. 2

[23] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 4

[24] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203, 2016. 2

[25] 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. 4, 6

[26] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1377–1385, 2015. 2

[27] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. Video object segmen-

tation without temporal information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 2

[28] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2014. 2

[29] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, pages 5000–5009, 2017. 3, 4, 5

[30] P. Ochs and T. Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. 2011. 2

[31] J. S. Pan and G. P. Bingham. With an eye to low vision: Optic flow enables perception despite image blur. *Optometry and Vision Science*, 90(10):1119–1127, 2013. 1

[32] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1777–1784, 2013. 2

[33] D. Pathak, R. B. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *CVPR*, volume 1, page 7, 2017. 1

[34] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun. Large kernel mattersimprove semantic segmentation by global convolutional network. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1743–1751. IEEE, 2017. 2

[35] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016. 4

[36] P. H. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *31st International Conference on Machine Learning (ICML)*, number EPFL-CONF-199822, 2014. 2

[37] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. Fullresolution residual networks for semantic segmentation in street scenes. *arXiv preprint*, 2017. 2

[38] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 1, 2

[39] 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. 1

[40] 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

[41] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1):157–173, 2008. 5

[42] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *arXiv preprint arXiv:1801.04381*, 2018. 6

[43] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015. 2

[44] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. *arXiv preprint arXiv:1704.05737*, 3, 2017. 2

[45] D. Tsai, M. Flagg, and J. M.Rehg. Motion coherent tracking with multi-label mrf optimization. *BMVC*, 2010. 4

[46] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3395–3402, 2015. 2

[47] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015. 2

[48] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang. Youtube-vos: Sequence-to-sequence video object segmentation. *arXiv preprint arXiv:1809.00461*, 2018. 2

[49] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2

[50] 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, 4

[51] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*, 2016. 2

[52] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015. 2

[53] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 4