Improving Semantic Segmentation via Video Propagation and Label Relaxation Supplementary Materials

Yi Zhu[1*] Karan Sapra[2*] Fitsum A. Reda[2] Kevin J. Shih[2] Shawn Newsam[1]
Andrew Tao[2] Bryan Catanzaro[2]
[1]University of California at Merced [2]Nvidia Corporation
{yzhu25,snewsam}@ucmerced.edu {ksapra,freda,kshih,atao,bcatanzaro}@nvidia.com

1. Implementation Details of Our Video Prediction/Reconstruction Models

In this section, we first describe the network architecture of our video prediction model and then we illustrate the training details. The network architecture and training details of our video reconstruction model is similar, except the input is different.

Recalling equation (1) from the main submission, the future frame $\widetilde{\mathbf{I}}_{t+1}$ is given by,

$$\widetilde{\mathbf{I}}_{t+1} = \mathcal{T}\Big(\mathcal{G}\big(\mathbf{I}_{1:t}, \mathbf{F}_{2:t}\big), \mathbf{I}_t\Big),$$

where $\mathcal{G}$ is a general CNN that predicts the motion vectors (u, v) conditioned on the input frames $\mathbf{I}_{1:t}$ and the estimated optical flow $\mathbf{F}_i$ between successive input frames $\mathbf{I}_i$ and $\mathbf{I}_{i-1}$. $\mathcal{T}$ is an operation that bilinearly samples from the most recent input $\mathbf{I}_t$ using the predicted motion vectors (u, v).

In our implementation, we use the vector-based architecture as described in [5]. $\mathcal{G}$ is a fully convolutional U-net architecture, complete with an encoder and decoder and skip connections between encoder/decoder layers of the same output dimensions. Each of the 10 encoder layers is composed of a convolution operation followed by a Leaky ReLU. The 6 decoder layers are composed of a deconvolution operation followed by a Leaky ReLU. The output of the decoder is fed into one last convolutional layer to generate the motion vector predictions. The input to $\mathcal{G}$ is $\mathbf{I}_{t-1}, \mathbf{I}_t$ and $\mathbf{F}_t$ (8 channels), and the output is the predicted 2-channel motion vectors that can best warp $\mathbf{I}_t$ to $\mathbf{I}_{t+1}$. For the video reconstruction model, we simply add $\mathbf{I}_{t+1}$ and $\mathbf{F}_{t+1}$ to the input, and change the number of channels in the first convolutional layer to 13 instead of 8.

We train our video prediction model using frames extracted from short sequences in the Cityscapes dataset. We use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 1×10^{-4}. The frames are randomly cropped to 256×256 with no extra data augmentation. We set the batch size to 128 over 8 V100 GPUs. The initial learning rate is set to 1×10^{-4} and the number of epochs is 400. We refer interested readers to [5] for more details.

Table 1: Accumulated and non-accumulated comparison. The numbers in brackets are the sample standard deviations.

Method	Baseline	Non-accumulated	Accumulated
mIoU (%)	80.85 ($\pm$0.04)	81.35 ($\pm$0.03)	81.12 ($\pm$0.02)

2. Non-Accumulated and Accumulated Comparison

Recalling Sec. 4.1 from the main submission, we have two ways to augment the dataset. The first is the non-accumulated case, where we simply use synthesized data from timesteps $\pm k$, excluding intermediate synthesized data from timesteps $< |k|$. For the accumulated case, we include all the synthesized data from timesteps $\leq |k|$, which makes the augmented dataset $2k + 1$ times larger than the original training set.

We showed that we achieved the best performance at ± 3, so we use $k = 3$ here. We compare three configurations:

1. *Baseline*: using the ground truth dataset only.
2. *Non-accumulated case*: using the union of the ground truth dataset and ± 3;
3. *Accumulated case*: using the union of the ground truth dataset, ± 3, ± 2 and ± 1.

For these experiments, we use boundary label relaxation and joint propagation. We report segmentation accuracy on the Cityscapes validation set.

We have two observations from Table 1. First, using the augmented dataset always improves segmentation quality as quantified by mIoU. Second, the non-accumulated case performs better than the accumulated case. We suspect this is because the cumulative case significantly decreases the probability of sampling a hand-annotated training example

within each epoch, ultimately placing too much weight on the synthesized ones and their imperfections.

3. Cityscapes

3.1. More Training Details

We perform 3-split cross-validation to evaluate our algorithms, in terms of cities. The three validation splits are {cv0: munster, lindau, frankfurt}, {cv1: darmstadt, dusseldorf, erfurt} and {cv2: monchengladbach, strasbourg, stuttgart}. The rest cities will be in the training set, respectively. cv0 is the standard validation split. We found that models trained on cv2 split leads to higher performance on the test set, so we adopt cv2 split for our final test submission. Using our best model, we perform multiscale inference on the 'stuttgart_00' sequence and generate a demo video. The video is composed of both video frames and predicted semantic labels, with a 0.5 alpha blending.

3.2. Failure Cases

We show several more failure cases in Fig. 1. First, we show four challenging scenarios of class confusion. From rows (a) to (d), our model has difficulty in segmenting: (a) car and truck. (b) person and rider. (c) wall and fence (d) terrain and vegetation.

Furthermore, we show three cases where it could be challenging even for a human to label. In Fig. 1 (e), it is very hard to tell whether it is a bus or train when the object is far away. In Fig. 1 (f), it is also hard to predict whether it is a car or bus under such strong occlusion (more than 95% of the object is occluded). In Fig. 1 (g), there is a bicycle hanging on the back of a car. The model needs to know whether the bicycle is part of the car or a painting on the car, or whether they are two separate objects, in order to make the correct decision.

Finally, we show two training samples where the annotation might be wrong. In Fig. 1 (h), the rider should be on a motorcycle, not a bicycle. In Fig. 1 (i), there should be a fence before the building. However, the whole region was labelled as building by a human annotator. In both cases, our model predicts the correct semantic labels.

3.3. More Synthesized Training Samples

We show 15 synthesized training samples in the demo video to give readers a better understanding. Each is a 11-frame video clip, in which only the 5th frame is the ground truth. The neighboring 10 frames are generated using the video reconstruction model. We also show the comparison to using the video prediction model and FlowNet2 [4]. In general, the video reconstruction model gives us the best propagated frames/labels in terms of visualization. It also works the best in our experiments in terms of segmentation accuracy. Since the Cityscapes dataset is recorded at 17Hz [2], the motion between frames is very large. Hence, propagation artifacts can be clearly observed, especially at the image borders.

Table 2: Per-class mIoU results on CamVid. Comparison with recent top-performing models on the test set. 'SS' indicates single-scale inference, 'MS' indicates multi-sclae inference. Our model achieves the highest mIoU on 8 out of 11 classes (all classes but tree, sky and sidewalk). This is expected because our synthesized training samples help more on classes with small/thin structures.

Method	Build.	Tree	Sky	Car	Sign	Road	Pedes.	Fence	Pole	Swalk	Cyclist	mIoU
RTA [3]	88.4	**89.3**	**94.9**	88.9	48.7	95.4	73.0	45.6	41.4	**94.0**	51.6	62.5
Dilate8 [7]	82.6	76.2	89.0	84.0	46.9	92.2	56.3	35.8	23.4	75.3	55.5	65.3
BiSeNet [6]	83.0	75.8	92.0	83.7	46.5	94.6	58.8	53.6	31.9	81.4	54.0	68.7
VideoGCRF [1]	86.1	78.3	91.2	92.2	63.7	96.4	67.3	63.0	34.4	87.8	66.4	75.2
Ours (SS)	90.9	82.9	92.8	**94.2**	69.9	97.7	76.2	74.7	51.0	91.1	78.0	81.7
Ours (MS)	**91.2**	83.4	93.1	93.9	**71.5**	**97.7**	**79.2**	**76.8**	**54.7**	91.3	**79.7**	**82.9**

4. CamVid

4.1. Class Breakdown

We show the per-class mIoU results in Table 2. Our model has the highest mIoU on 8 out of 11 classes (all classes but tree, sky and sidewalk). This is expected because our synthesized training samples help more on classes with small/thin structures. Overall, our method significantly outperforms previous state-of-the-art by 7.7% mIoU.

4.2. More Synthesized Training Samples

For CamVid, we show two demo videos of synthesized training samples. One is on the validation sequence '006E15', which is manually annotated every other frame. The other is on the training sequence '0001TP', which has manually annotated labels for every 30th frame. For '006E15', we do one step of forward propagation to generate a label for the unlabeled intermediate frame. For '0001TP', we do 15 steps of forward propagation and 14 steps of backward propagation to label the 29 unlabeled frames in between. For both videos, the synthesized samples are generated using the video reconstruction model trained on Cityscapes, without fine-tuning on CamVid. This demonstrates the great generalization ability of our video reconstruction model.

5. Demo Video

We present all the video clips mentioned above at https://nv-adlr.github.io/publication/2018-Segmentation.

References

[1] S. Chandra, C. Couprie, and I. Kokkinos. Deep Spatio-Temporal Random Fields for Efficient Video Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

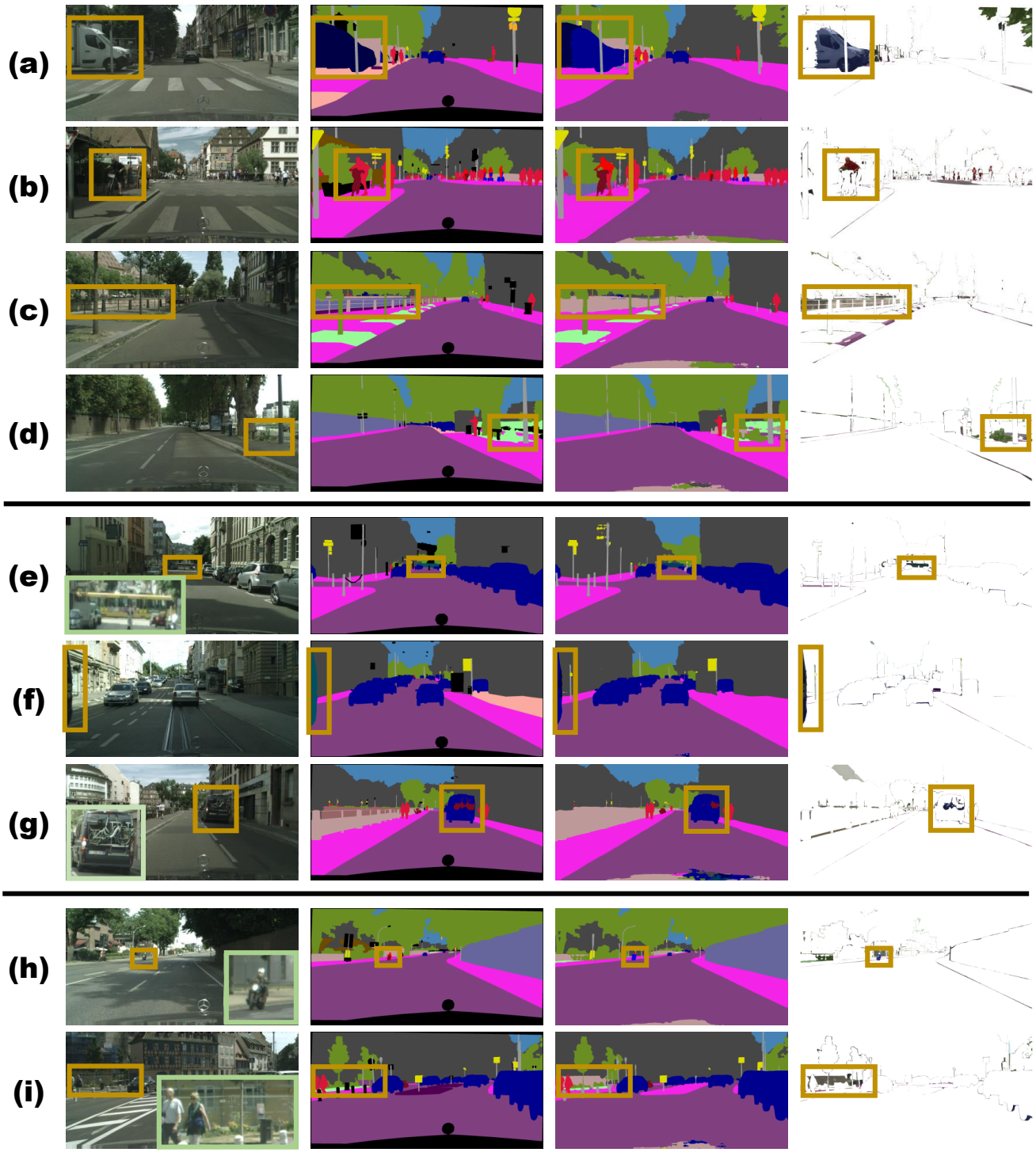

Figure 1: Failure cases (in yellow boxes). From left to right: image, ground truth, prediction and their difference. Green boxes are zoomed in regions for better visualization. Row (a) to (d) show class confusion problems. Our model has difficulty in segmenting: (a) car and truck. (b) person and rider. (c) wall and fence (d) terrain and vegetation. Row (e) to (f) show challenging cases when the object is far away, strongly occluded, or overlaps other objects. The last two rows show two training samples with wrong annotations: (h) mislabeled motorcycle to bicycle and (i) mislabeled fence to building.

[2] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[3] P.-Y. Huang, W.-T. Hsu, C.-Y. Chiu, T.-F. Wu, and M. Sun. Efficient Uncertainty Estimation for Semantic Segmentation in Videos. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[4] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[5] F. A. Reda, G. Liu, K. J. Shih, R. Kirby, J. Barker, D. Tarjan, A. Tao, and B. Catanzaro. SDC-Net: Video Prediction using Spatially-Displaced Convolution. In *European Conference on Computer Vision (ECCV)*, 2018. 1

[6] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[7] F. Yu and V. Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. In *International Conference on Learning Representations (ICLR)*, 2016. 2