NightLab: A Dual-level Architecture with Hardness Detection for Segmentation at Night Supplementary Materials

Xueqing Deng[1,2], Peng Wang[2], Xiaochen Lian[2], Shawn Newsam[1]
[1]EECS, University of California at Merced, [2]ByteDance Inc.
{xdeng7, snewsam}@ucmerced.edu, {peng.wang, xiaochen.lian}@bytedance.com

A. HDM implementation details

In this section, we introduce details on HDM. As mentioned in Sec.3.3, we aim to learn better proposals with contexts using HDM instead of RDN. Both RDN and HDM consist of anchor-based region proposal network (RPN) [2]. We adopt the model architecture of RPN designed in maskrcnn [1] as RDN. We then create the annotations (bounding box and class id) of the dataset that was used for hard regions segmentation to train the RPN following the configurations in maskrcnn. With above steps along with the objectives in Sec.3.2 ($\mathcal{L}^R_{cls}$ and $\mathcal{L}^R_{reg}$), the proposed RDN (a baseline of RPN) will be trained to detect hard regions. The hard regions contain the regions with objects that we are interested (selected by hard classes in Sec.3.1)

RDN and HDM are similar, but HDM has a few more steps to learn the proposal with contexts. In other words, HDM is used to filter the proposals from RPN with only good proposals (Sec.3.3) derived from comparisons between image-level and region-level predictions. By such comparison, new labels will be re-assigned for the the proposals. And thus we change our objective to earn the good proposals based on the segmentation results. We assume if Φ^R_{seg} outperforms Φ^I_{seg} in a proposal s, s will be able to provide better context for Φ^R_{seg}. We can infer whether the proposal is providing better context for Φ^R_{seg}. Meanwhile, proposals whose iou scores from Φ^R_{seg} are worse than Φ^I_{seg} will be treated as background, even though they have been selected by RPN (include objects that we are interested but with worse context for Φ^R_{seg}).

In details, we leverage Φ^R_{seg} to re-assign the labels for proposals. We first crop the images by the proposals derived by anchors, which can be later fed through Φ^R_{seg} to obtain P^R. At the same time, we can pull P^I from Φ^I_{seg} and crop images by the same proposals from last step to obtain cropped P^I. The cropped P^I and P^R can be evaluated then compared the scores. Afterwards, we can filter the proposals that can provide better segmentation with Φ^R_{seg}. A vector will be created to derive the new classification labels for proposals following the rule below:

$$v'_j = \begin{cases} 1, iou(P^R_j, Y^R_j) > iou(P^I_j, Y^I_j) \\ 0, otherwise \end{cases} \quad (1)$$

where $V' = \{v'_j\}^M_{j=0}$, M denotes the number of proposals.

We can then derive the new labels for the proposals Y^P by performing element-wise product with V'. In particular, the classification loss will be changed as follow:

$$\mathcal{L}^R_{cls} = \sum_{s \in proposals} CE(p_s, y^P_s) \quad (2)$$

It is noted that even though only the class labels are changed, the bounding box regression will be optimized as well. Since the regression loss is only computed with positive proposals, the proposals with new positive labels will be then used for loss. In summary, the new labels not only recalculate $\mathcal{L}^R_{cls}$ directly, but also affect $\mathcal{L}^R_{reg}$ implicitly.

B. Merge implementation details

As shown in the Fig. 1, given image-level prediction, we use a mask to replace it with region-level prediction. Such a mask (the purple mask in the figure) consists of areas of hard classes (detected in Fig.3 in the paper) from region-level prediction.

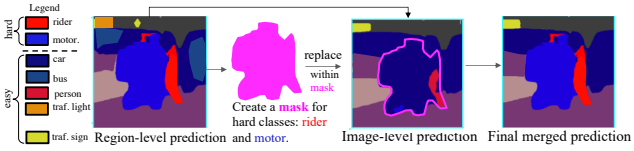

Figure 1. Merging details.

C. Adaptation implementation details

We introduce the implementation details of adaptation in Sec.4.2. Existing approaches are mostly in unsupervised fashion. We realize that directly applying these methods

[1]https://github.com/facebookresearch/maskrcnn-benchmark

would not be a fair comparison. We then take a strategy of adaptation-segmentation, which means we first perform unsupervised adaptation then follow supervised learning pipeline by adding labels for training a segmentation model. We categorize the adaptation into image-based adaptation and model-based adaptation.

Image adaptation These approaches will create new data for training segmentation model. We select multiple approaches, *i.e*. SingleHDR for converting the images to HDR images, Pix2PixHD and CycleGAN for converting the image style. We adopt the pretrained SingleHDR model from [1] to perform inference on our night image datasets to obtain enhanced night images to feed through segmentation model. While at segmentation inference, test images will be enhanced by the same SingleHDR model first and later fed through segmentation model. Similar steps are followed for Pix2PixHD [4] and CycleGAN [6]. It is noted that, Pix2PixHD is an approach for paired images. Since we don't have paired images, we create synthetic day images to pair night images. Taking NightCity+ and Cityscapes as examples, a pix2pix-day model will be first trained for Cityscapes and its semantic label mask (label→image). Then we can obtain the synthetic day-style images for night image content by their semantic labels. We pair these synthetic day images with NightCity+ night images to train a night-day converter, namely pix2pix-night2day, in order to obtain day images from night directly without accessing semantic label masks anymore. Finally, once the images have been processed, the processed images along with the label masks will be used for training segmentation. In details, We follow the training configurations in pix2pix/cyclegan [2] to transfer the images. We train the adaptation model with 50,000 iterations, with 2 sample per gpu. We then select the model with the best FID score to transfer all the night images for both train and test sets. At last, with the transferred data, we train the segmentation model following the implementation details presented in Sec.4.1.

Model adaptation Model adaptation is different image adaptation. The data is not necessary for pre-processing. The model is adapted directly with the original data to optimize the model weights. Similar to image adaptation approaches, two datasets are used for adaptation, day and night datasets. Existing approaches utilize the unsupervised domain adaptation method to transfer labels from day dataset (with labels) to night dataset (no labels). This is different from the training settings of our method where both labels are available for day and night. Therefore, we first adapt the model weights using unsupervised domain adaptation approach, *i.e*. DANNet [5] and AdaptSeg [3]. Later, we use the adapted model with new weights to finetune with day and night dataset with both labels and images. In summary, we adopt the unsupervised domain adaptation methods as model initialization to train the segmentation model later in an supervised manner. In details, we train the unsupervised domain adaptation model with 50,000 iterations, with 1 image per gpu following [3, 5]. We then take the adapted model which can provide the best performance on night val set to finetune with labels.

[2]https://github.com/phillipi/pix2pix

Thres.	Selected hard classes	mIoU
0.3	terr.	60.36
0.4	pole; terr.; rier; motor.; traf. light	62.31
0.5	pole; terr.; traf. light; bic.; rier; motor.	**62.82**
0.6	pole; terr.; traf. light; bic.; rier; motor.; wall; side walk	62.43

Table 1. Ablation of threshold selection. Performed on NightCity+ jointly train with Cityscapes, mIoU score (%) is reported on NightCity+ val set.

D. Ablation study on hard classes threshold

We show the ablation study of the hard classes threshold in Tab. 1. We can see in the table, increasing threshold results in allowing more classes selected as hard classes which means more hard regions will be created for region-level segmentation model. Thresholding classes with 0.5 achieves the best results as most hard objects are selected, *i.e*. rider, bicycle (bic.), motorcycle (motor.), pole and traffic light (traf. light). While increasing the threshold to 0.6, some classes like side walk and wall that are more likely background will be added to be considered. Such behavior will harm the model performance. When the threshold is decreased to 0.4, some objects like traffic light and bicycle will be removed from hard classes. We can see mIoU is lower. If we decrease the threshold futher to 0.3, only terrain (terr.) is selected as hard class for region-level process resulting in no improvement. In other words, selecting suitable amount of hard objects which decides the class to be improved plays an important role in our method .

E. NightCity+ labeling

We noticed there are multiple images with incorrect label mask in the dataset NightCity. In order to provide accurate evaluation, we relabel the validation set of NightCity which is served as test set to report scores in our work. We show some examples in 2. There are 1299 images in val set, among which about 25% images with 10% area mislabeled and 10% images with 50% area mislabeled and the remaining with 5% area mislabeled. We have 4 human labeller to check and verify the label masks for val set. We divide the people into two groups, and the annotations are crossly verified. For those regions which are too dark and blur, we remain them unchanged.

F. Failure cases analysis

`NightLab` is able to provide accurate details of small objects while maintain the correct semantics of large objects as shown in Fig. 3, thanks to the proposed HDM and region-level segmentation model. Despite of the great success of our model, we observe some failure cases as shown in Fig. 4. Both methods fail in ads on the building, and some blur and dark regions. Row 4 shows the difficult examples for fence where the scene is a mixture of road and fence. The model has difficulty in predicting the correct boundary for fence and road. Row 5 shows another difficult scenario where a person does not ride on bicycle. Models are likely to recognize the person as "rider" but the ground truth is "person" due to the fact that the person is not riding on the bicycle. In summary, the blurriness and darkness bring great challenges to night-time segmentation which should draw more attention from the community.

References

[1] Yu-Lun Liu, Wei-Sheng Lai, Yu-Sheng Chen, Yi-Lung Kao, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Single-image hdr reconstruction by learning to reverse the camera pipeline. In *CVPR*. 2

[2] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPs*. 1

[3] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018. 2

[4] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 2

[5] Xinyi Wu, Zhenyao Wu, Hao Guo, Lili Ju, and Song Wang. Dannet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation. In *CVPR*. 2

[6] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *CVPR*. 2

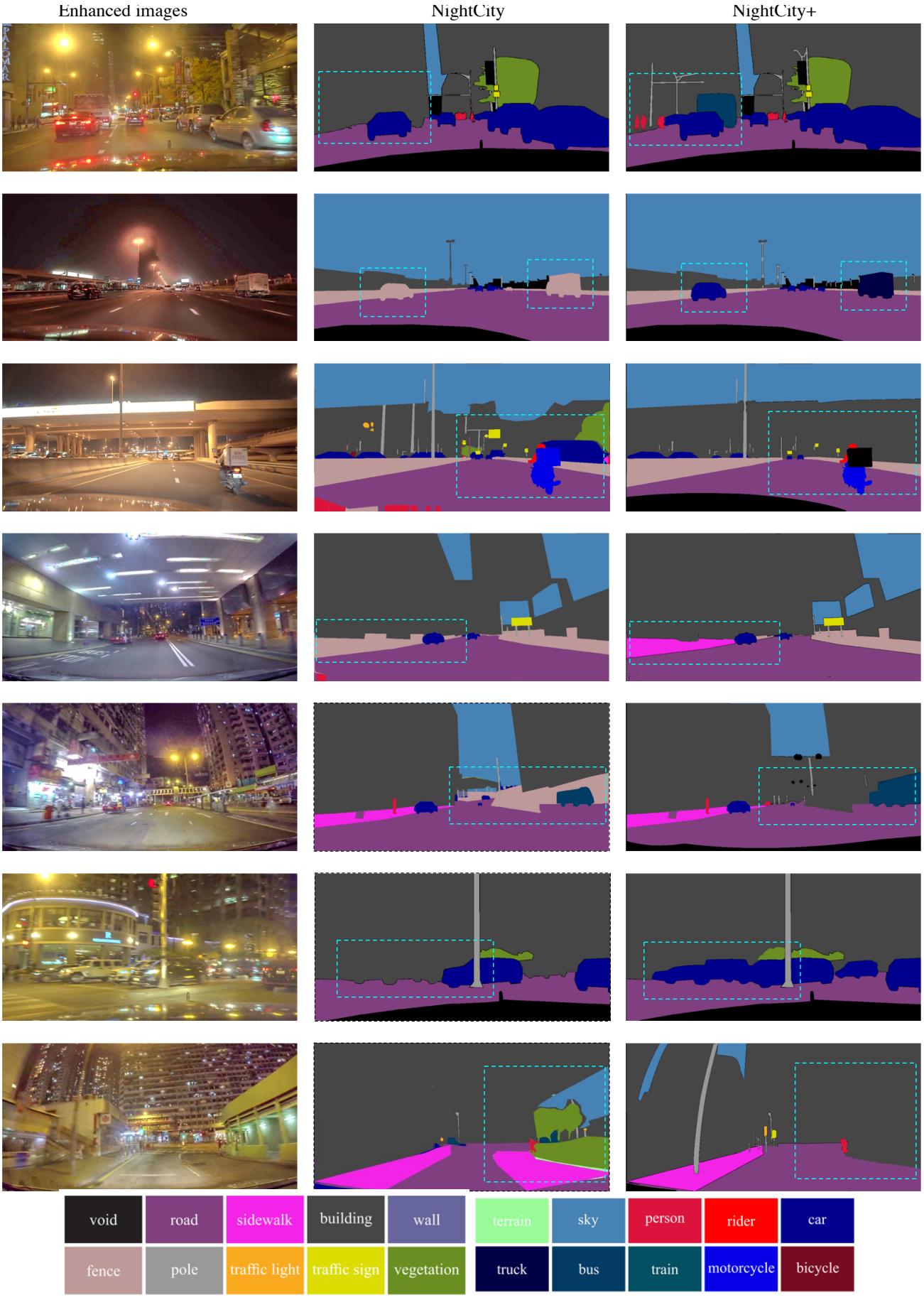

Figure 2. Example corrections. Left: images with enhancement, middle: NightCity, right: NightCity+. Red boxes show the main corrected area.

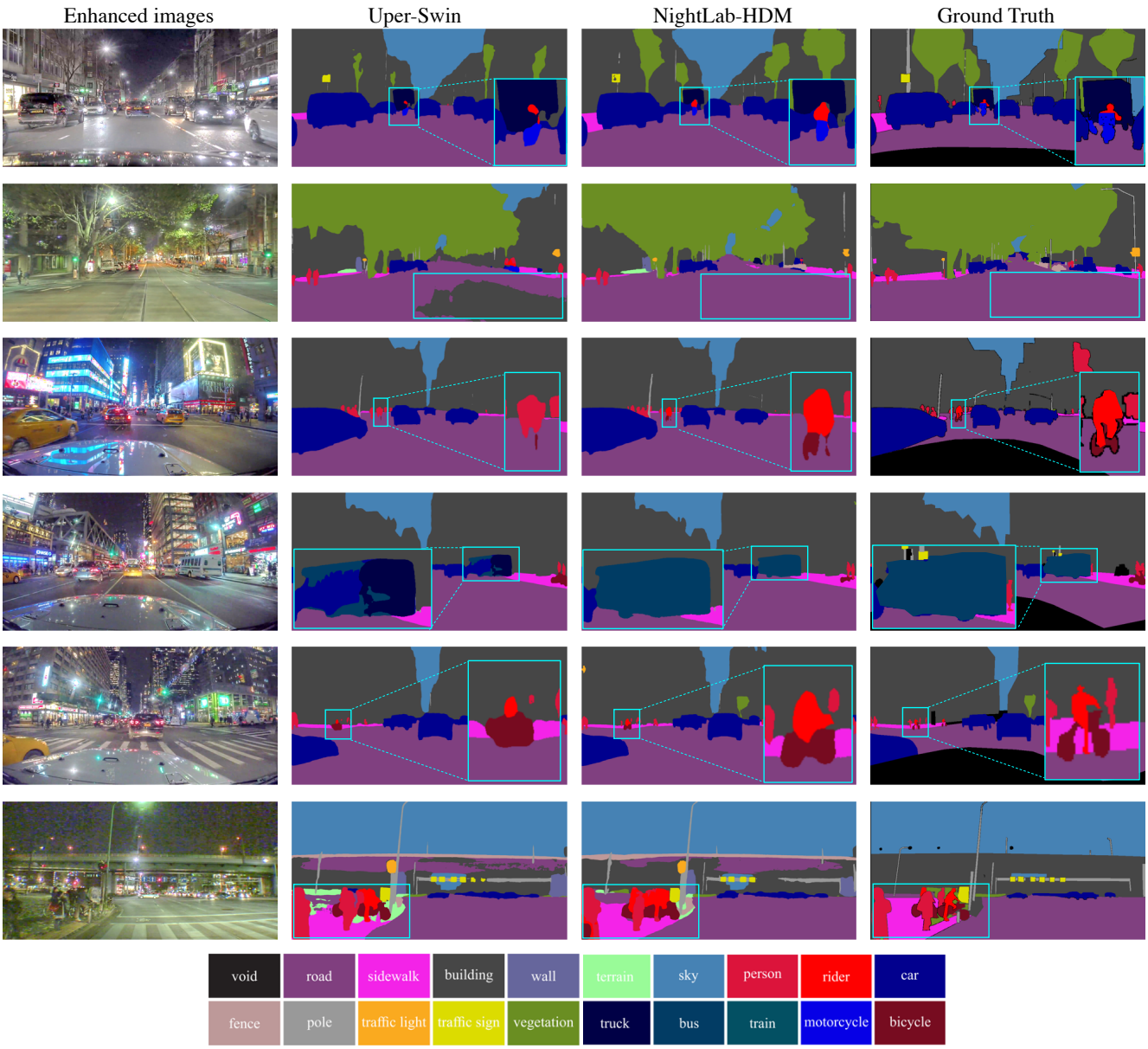

Figure 3. Successful examples.`NightLab` is able to provide accurate details of small objects while maintain the correct semantics of large objects, thanks to HDM and region-level segmentation model.

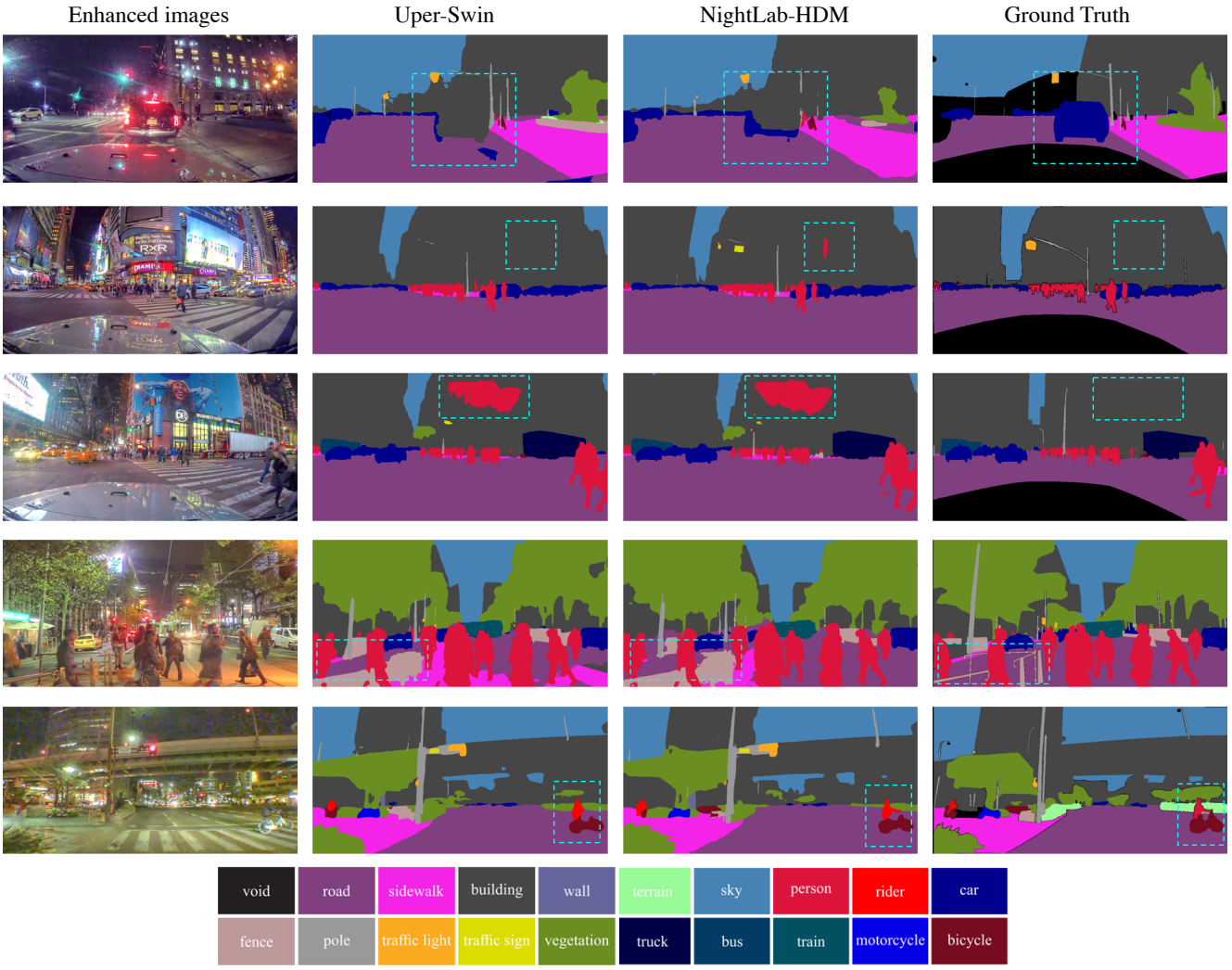

Figure 4. Failure case examples. Both methods fail in ads on the building, and some blur and dark regions. Row 4 shows the difficult examples for fence where the scene contains both road and fence. Row 5 shows another difficult scenario where person does not ride on bicycle. Models are likely to recognize the person as "rider" but the ground truth is "person".