Disentangle then Parse:
Night-time Semantic Segmentation with Illumination Disentanglement

Zhixiang Wei[1*] Lin Chen[1,2*] Tao Tu[1] Pengyang Ling[1] Huaian Chen[1†] Yi Jin[1†]
[1] University of Science and Technology of China [2] Shanghai AI Laboratory
{zhixiangwei,chlin,tutao9036,lpyang27,anchen}@mail.ustc.edu.cn, jinyi08@ustc.edu.cn

1. NightCity-fine

NightCity-fine is a refined dataset for night-time semantic segmentation, which aims to improve the quality of annotations in both the training and validation sets. This dataset builds upon NightCity [5], which is the largest dataset for night-time segmentation, and Nightcity+ [2], a validation set based on NightCity. Our annotation process begins by comparing the annotations in the NightCity dataset with the original images, which identified a significant number of missing labels and incorrectly labeled regions. To address these issues, we utilized the graphical image annotation tool, Labelme [6], to accurately label the previously missing regions and remove incorrect labels from mislabeled regions.

As a result, we successfully eliminate 4747 mislabeled regions and rectify 14288 missing labels with the appropriate label, as depicted in Fig. 1. In total, we refined 84% of the images in the dataset, and added 14228 shapes to each category, including both things and stuff, as illustrated in Fig. 2. Among them, traffic light and traffic sign have the most significant numbers of 2981 and 2963, respectively. We compared the pixel distributions of labeled regions in NightCity and NightCity-fine, as shown in Fig. 3. As some categories have significantly higher numbers of pixels than others, we presented the distributions in log scale. Overall, our refined dataset has more balanced pixel distributions of all classes than the original dataset, with more labeled regions for traffic sign, traffic light, motorcycle, wall, and pole, among others. A qualitative comparison of NightCity and NightCity-fine dataset can be found in Fig. 4 and Fig. 5.

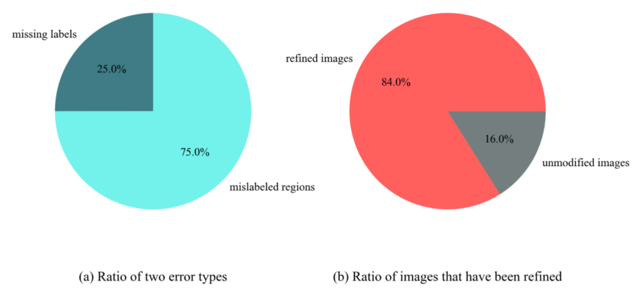

Figure 1. Error correction in datasets: a pie chart analysis of error types and the ratio of refined images.

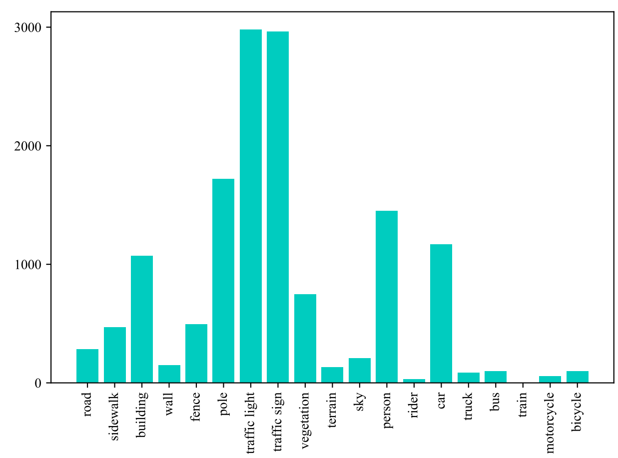

Figure 2. The number of labels added to each category.

2. Study on guidance noise

We select two guidance noise distributions. The first noise distribution is based on the Gaussian distribution recommended by [3], which is added to generated illumination to prevent the model to produce an identity result. To be specific, we apply a noise N that follows the standard normal distribution $\mathcal{N}(0, 1)$. Subsequently, to account for the smoothness and value range of illumination, we normalize the noise N to the range of 0 to 1 and pass it through an average pooling layer with a kernel size of 16 and stride size of 16. Moreover, inspired by the work of [1,8], we also use the normalized V channel of the input image in the HSV color space as another guidance noise V. Similarly, this noise is fed to a max pooling layer with a kernel size of 16 and a

[*] indicates equal contributions.
[†] Corresponding authors.

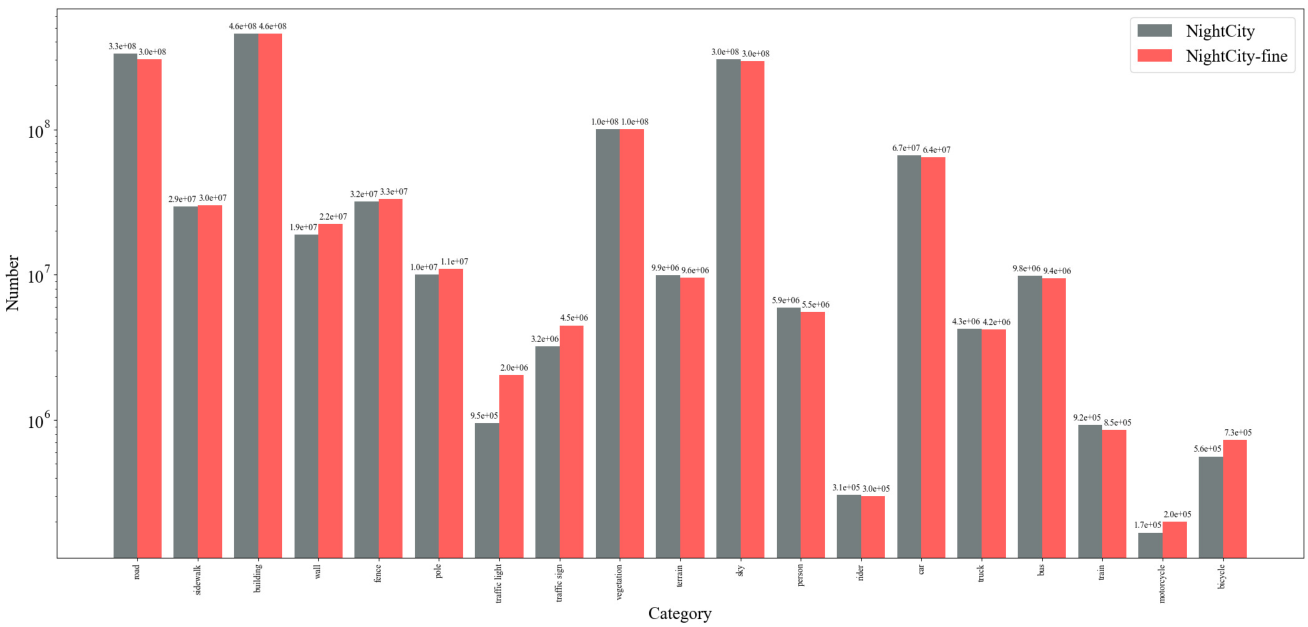

Figure 3. Number of labels in each category before and after refining.

Table 1. Ablation Studies on the effects of different guide illumination. Random is random generated smooth noise, and Max is the maximum channel of a random picture. The best scores are indicated in **bold**.

Guidance Noise	None	N	V	$N+V$
mIoU(%)	61.6	63.9	62.7	**64.2**

Table 2. Detailed architecture of disentanglement model with different depths.

Disentangle model	Small	Base	Large	Huge
Downsample blocks	1	2	3	4
Residual blocks	1	2	3	4
Channels of Residual	64	128	256	512
Upsample blocks	1	2	3	4
Parameters	0.2M	1.5M	5.9M	26.8M

stride size of 16 to constrain its smoothness. By using these two guidance noise distributions, we aim to prevent the disentanglement module from learning a fixed transformation and roughly guide it to generate valid illumination during the initial training stage. Table 1 demonstrates the effectiveness of both noise N and V. Our approach achieves the best performance when both types of noise are applied simultaneously.

3. Model architecture

Disentanglement model. The network architecture includes a stem layer, multiple downsampling convolution layers, several residual blocks, two Swin blocks [4], a pyramid pooling module [7], several upsampling convolution layers, and two output convolution layers. The reflectance is obtained by adding the output and input images. Tab. 2 displays the number of blocks at different depths.

IAParser. The proposed IAParser consists of several components, including a reflectance segmentation model M_{ref} that can be substituted with an existing semantic segmentation network, an illumination segmentation model M_{ill} that adopts a pyramid pooling module architecture [7], a convolution layer W_{mask} that calculates the attention mask, and a convolution layer W_{cls} that transforms the features produced by M_{ref} and M_{ill} into semantic logits.

4. Algorithm

The training procedure of our DTP is summarized in Algorithm 1, which is composed of semantic-oriented disentanglement (SOD) and illumination-aware parser (IAParser). For detailed equations and loss functions, please refer to the main paper.

5. Qualitative results

Fig. 6, 7, and 8 showcase the reflectance and semantic segmentation results produced by our proposed method. Although the limited dataset scale and model parameters resulted in incomplete disentanglement, which led to the presence of redundant lighting-specific components in the reflectance, our approach effectively enhances the model's ability to parse images and generate superior semantic segmentation results. The observed improvement in both visual

Algorithm 1: Training process of DTP.

Input: disentanglement models: M_{dis}; reflectance segmentation model: M_{ref}; illumination segmentation mode: M_{ill}; convolution layers: W_{mask}, W_{cls}; maximum iteration T.

Output: finale network consists of $M_{dis}, M_{ref}, M_{ill}, W_{mask}, W_{cls}$.

```
for t ← 1 to T do
    Get batch data: (X_j, Y_j, X_x, Y_k)
    R_j, I_j = M_dis(X_j)
    R_k, I_k = M_dis(X_k)
    Get I'_j, I'_k by Eq. (3)
    Get R^s_j, I^s_j, R^s_k, I^s_k by Eq. (4)
    Calculate L_disentangle by Eq. (5)
    for R, I, X in (R_j, I_j, X_j), (R_k, I_k, X_k), (R^s_j, I^s_j, R_j ⊙ I'_j), (R^s_k, I^s_k, R_k ⊙ I'_k) do
        Calculate L_retinex(R, I, X) by Eq. (5)
        Calculate L_smooth(R, I) by Eq. (6)
    for R, I, Y in (R_j, I_j, Y_j), (R_k, I_k, Y_k), (R^s_j, I_j, Y_j), (R^s_k, I_k, Y_k) do
        F_ill = M_ill(I)
        F_ref = M_ref(R)
        Calculate A_mask by Eq. (9)
        Calculate Ỹ by Eq. (10)
        Calculate L_segill by Eq. (11)
        Calculate L_seg by Eq. (12)
    Optimize network
```

quality and mIoU metrics (refer to the main paper) supports the effectiveness and competitiveness of our method.

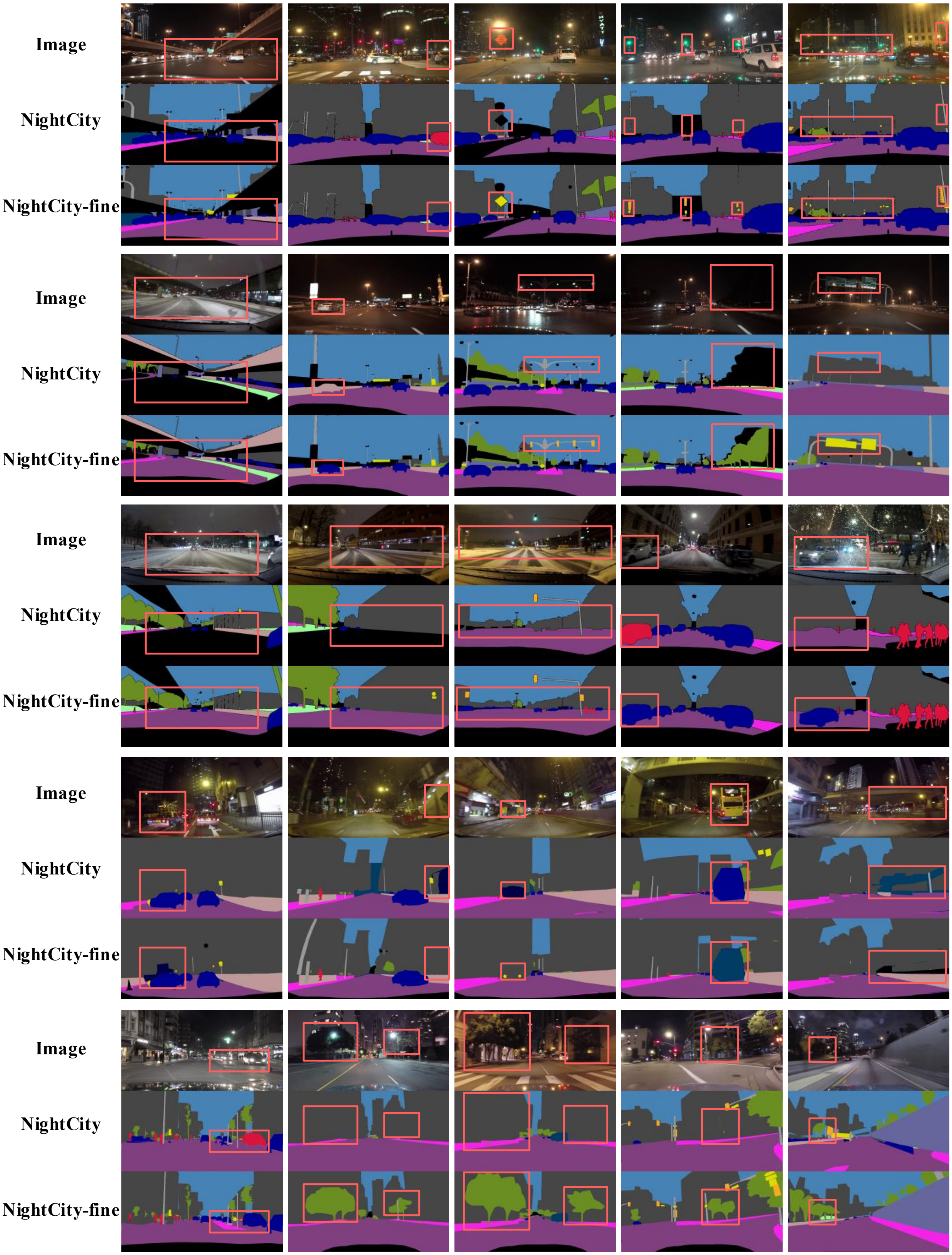

Figure 4. Qualitative Comparison of NightCity dataset with NightCity-fine dataset.

Figure 5. Qualitative Comparison of NightCity dataset with NightCity-fine dataset.

Figure 6. Qualitative Results of the reflectance and semantic segmentation outcomes produced by our proposed method.

Figure 7. Qualitative Results of the reflectance and semantic segmentation outcomes produced by our proposed method.

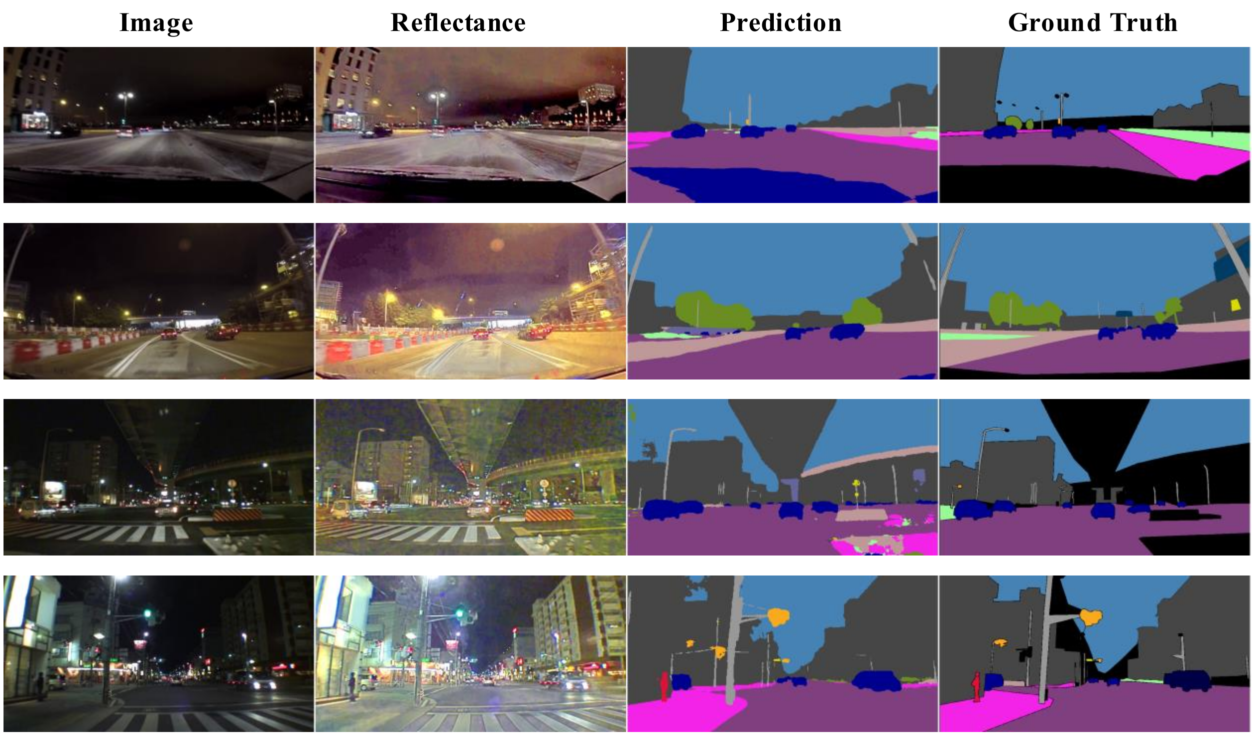

Figure 8. Qualitative Results of the reflectance and semantic segmentation outcomes produced by our proposed method.

References

[1] Jianrui Cai, Shuhang Gu, and Lei Zhang. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE Transactions on Image Processing*, 27(4):2049–2062, 2018. 1

[2] Xueqing Deng, Peng Wang, Xiaochen Lian, and Shawn Newsam. Nightlab: A dual-level architecture with hardness detection for segmentation at night. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16938–16948, 2022. 1

[3] Abel Gonzalez-Garcia, Joost Van De Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. *Advances in neural information processing systems*, 31, 2018. 1

[4] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[5] Xin Tan, Ke Xu, Ying Cao, Yiheng Zhang, Lizhuang Ma, and Rynson WH Lau. Night-time scene parsing with a large real dataset. *IEEE Transactions on Image Processing*, 30:9085–9098, 2021. 1

[6] Kentaro Wada. Labelme: Image Polygonal Annotation with Python. 1

[7] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 2

[8] Zunjin Zhao, Bangshu Xiong, Lei Wang, Qiaofeng Ou, Lei Yu, and Fa Kuang. Retinexdip: A unified deep framework for low-light image enhancement. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(3):1076–1088, 2021. 1