Enhancing photorealism enhancement Supplemental material

Stephan R. Richter, Hassan Abu AlHaija, and Vladlen Koltun

We provide additional implementation details (Sec. 1) and results and comparisons to baselines (Sec. 2).

1 Additional Implementation details

Discriminator architecture. All of our discriminator networks share the same base architecture, consisting of a stack of 5 Convolution-GroupNorm- LeakyReLU (CGL) layers and a final Convolution-LeakyReLU- Convolution layer (CLC) (see Tab. 1 for details). Let y denote the per-pixel output of the last CGL layer, z the output of the CLC layer, and e a learned embedding from the semantic segmentation map. Each discriminator then outputs a per-pixel scalar $s = z + \langle y, e \rangle$. The embeddings are learned per discriminator. If the resolution of the semantic segmentation map and y differ, we bilinearly downsample the embedding map accordingly. All LeakyReLUs have a slope of 0.2, all GroupNorms have 8 groups.

Disc.	VGG	CGL_0	CGL_1	CGL_2	CGL_3	CGL_4	CLC
0	1-1	64-2	128-*	256-*	*-*	*-1	*-*
1	1-2	64-2	128-*	256-*	*-*	*-1	*-*
2	2-1	128-2	256-*	*-*	*-1	*-*	*-*
3	2-2	128-2	256-*	*-*	*-1	*-*	*-*
4	3-1	256-2	*-*	*-1	*-*	*-*	*-*
5	3-2	256-2	*-*	*-1	*-*	*-*	*-*
6	3-3	256-2	*-*	*-1	*-*	*-*	*-*
7	4-1	512-2	256-1	*-*	*-*	*-*	*-*
8	4-2	512-2	256-1	*-*	*-*	*-*	*-*
9	4-3	512-2	256-1	*-*	*-*	*-*	*-*

TABLE 1: Architecture of discriminator networks. Each row represents the configuration of a discriminator, consisting of 5 consecutive Convolution-GroupNorm-LeakyReLU layers (CGL_{0-5}) and a final Convolution-LeakyReLU-Convolution layer (CLC). Entries for each layer represent input dimension and stride. * denotes the same value as for the previous layer. The input for each discriminator is a feature map extracted at a specific `relu` layer (VGG column) from a pretrained VGG-16 network.

2 Additional results

2.1 Segmentation

We show exemplary segmentations by MSeg on Cityscapes in Fig. 1, and on GTA in Fig. 2.

2.2 Cityscapes

To give a more comprehensive impression of the consistency of our method, we randomly sample 10 images from GTA V and enhance them via all baselines. The results are shown in Fig. 3 and confirm our findings from the main paper. Color transfer approaches (Color transfer & CDT) only modify low-level features. Textures and objects keep their synthetic appearance. Methods for photo style transfer (PhotoWCT & WCT2) adapt images at a deeper level. They match the style of learned features to those from a reference image. The quality of enhancements thus strongly depends on a favorable reference image. SPADE ignores the input image and synthesizes a new image from a semantic label map. As the scene layouts from GTA are different to the ones from Cityscapes, the global scene priors learned by SPADE are misleading and commonly result in strong artifacts. Image-to-image translation approaches (MUNIT, CUT, CyCADA & TSIT) are trained with an adversarial objective, i.e., a discriminator network. Without explicitly addressing the structural shift between synthetic and real datasets as we detail in the main paper, this commonly leads to typical artifacts such as trees in the sky or stars at the bottom in the case of Cityscapes. In contrast to prior work, our method enhances the photorealism of rendered images while keeping geometric and semantic content consistent with the input image. Results by our method are also temporally stable as can be seen in the supplemental video at https://youtu.be/P1IcaBn3ej0

2.3 Mapillary Vistas

We show more enhancements of GTA images by our method trained on Mapillary Vistas in Fig. 4.

Fig. 1: Exemplary predictions by MSeg on images from Cityscapes.

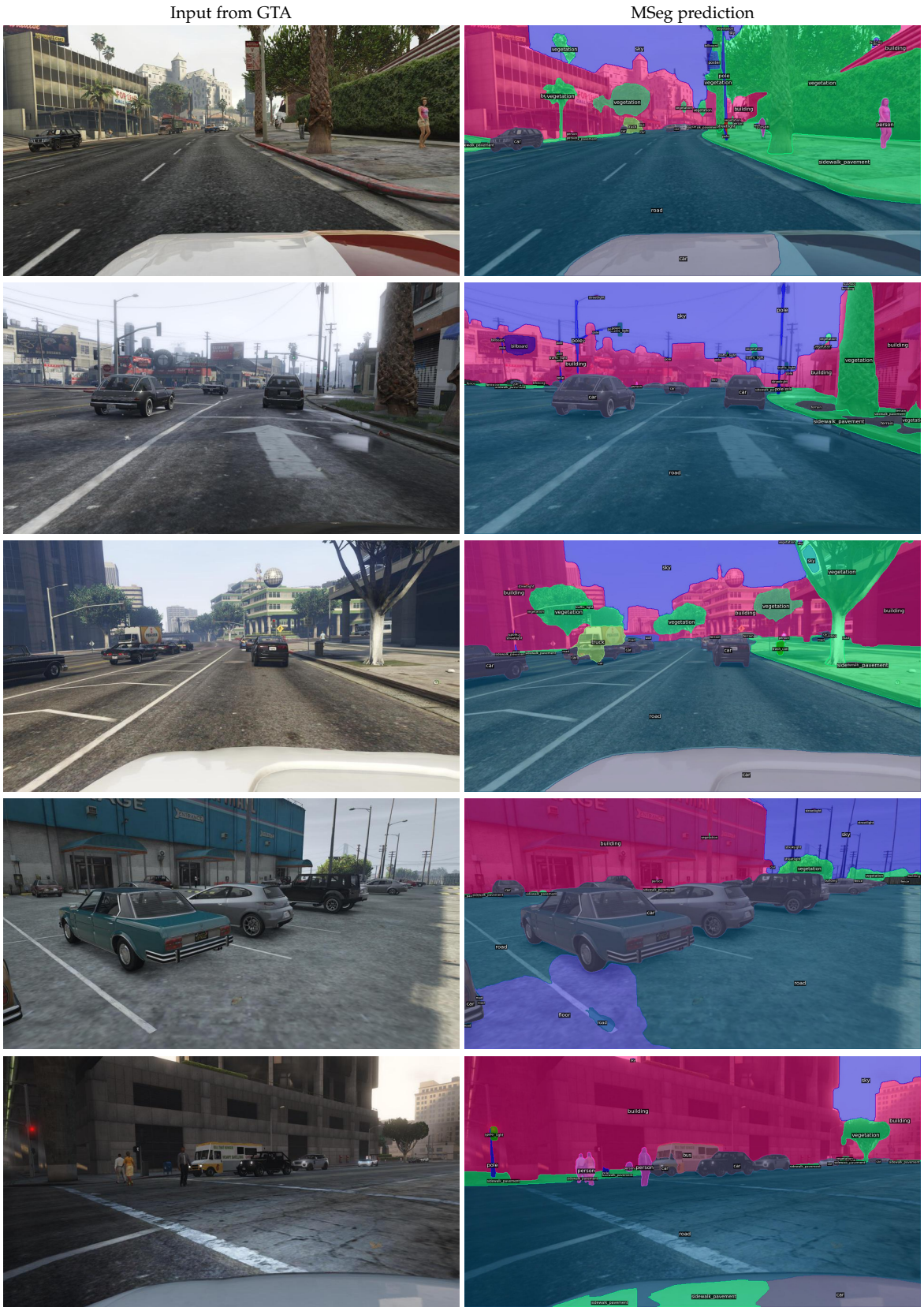

Fig. 2: Exemplary predictions by MSeg on images from GTA.

Fig. 3: We compare our results to original GTA images and a number of baselines. Images from GTA are randomly sampled. Insets show details of the respective image.

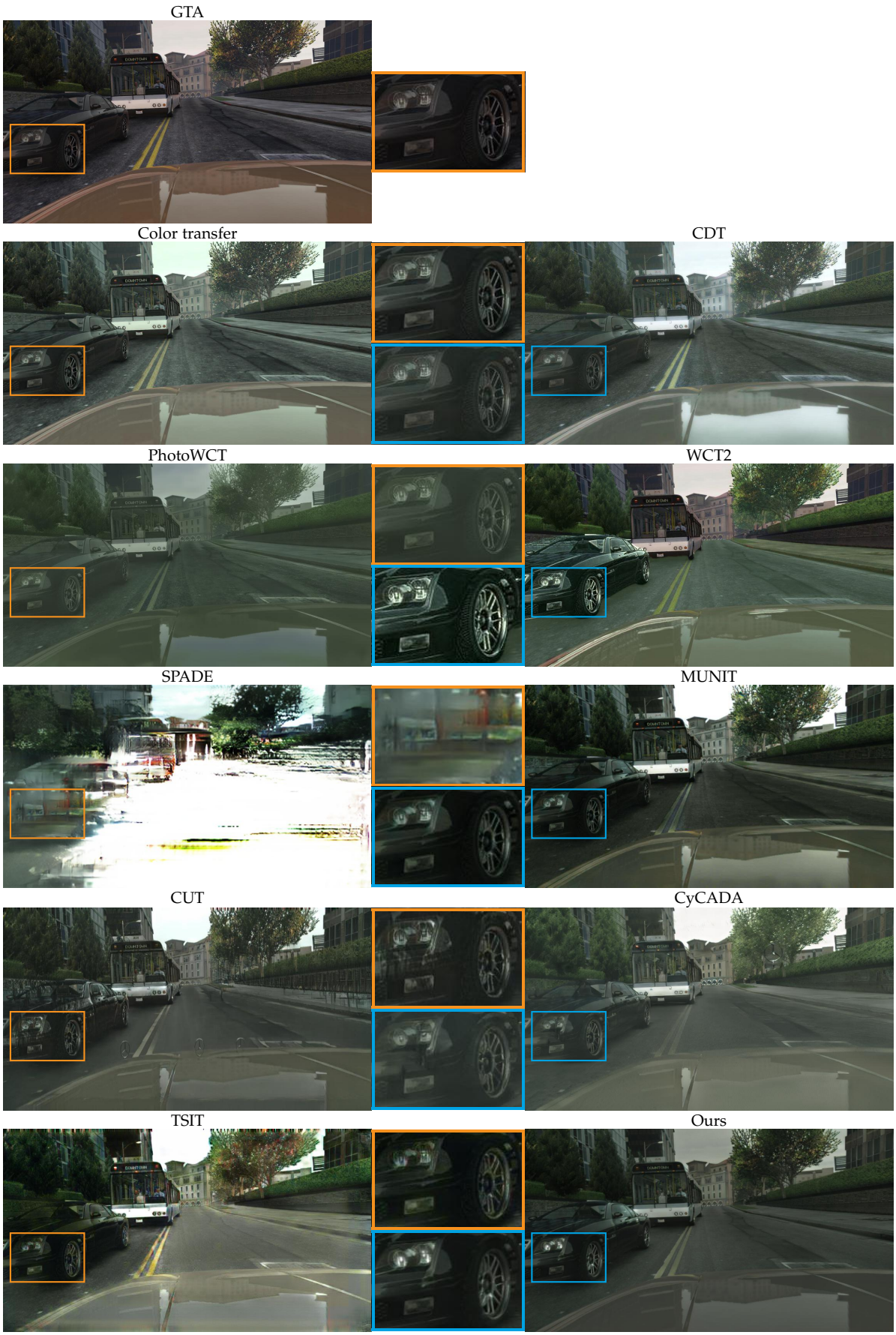

Fig. 3: We compare our results to original GTA images and a number of baselines. Images from GTA are randomly sampled. Insets show details of the respective image. (Continued)

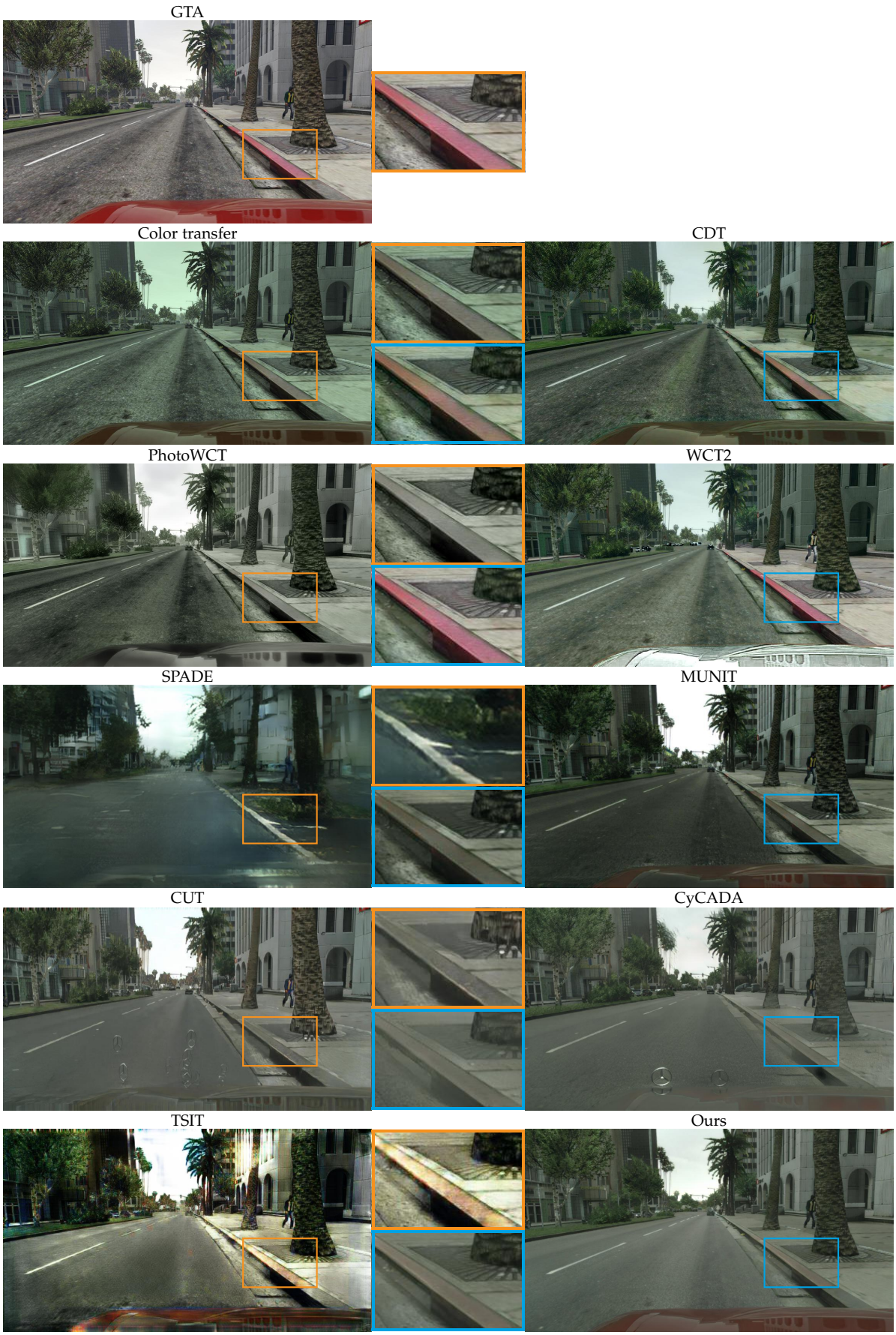

Fig. 3: We compare our results to original GTA images and a number of baselines. Images from GTA are randomly sampled. Insets show details of the respective image. (Continued)

Fig. 3: We compare our results to original GTA images and a number of baselines. Images from GTA are randomly sampled. Insets show details of the respective image. (Continued)

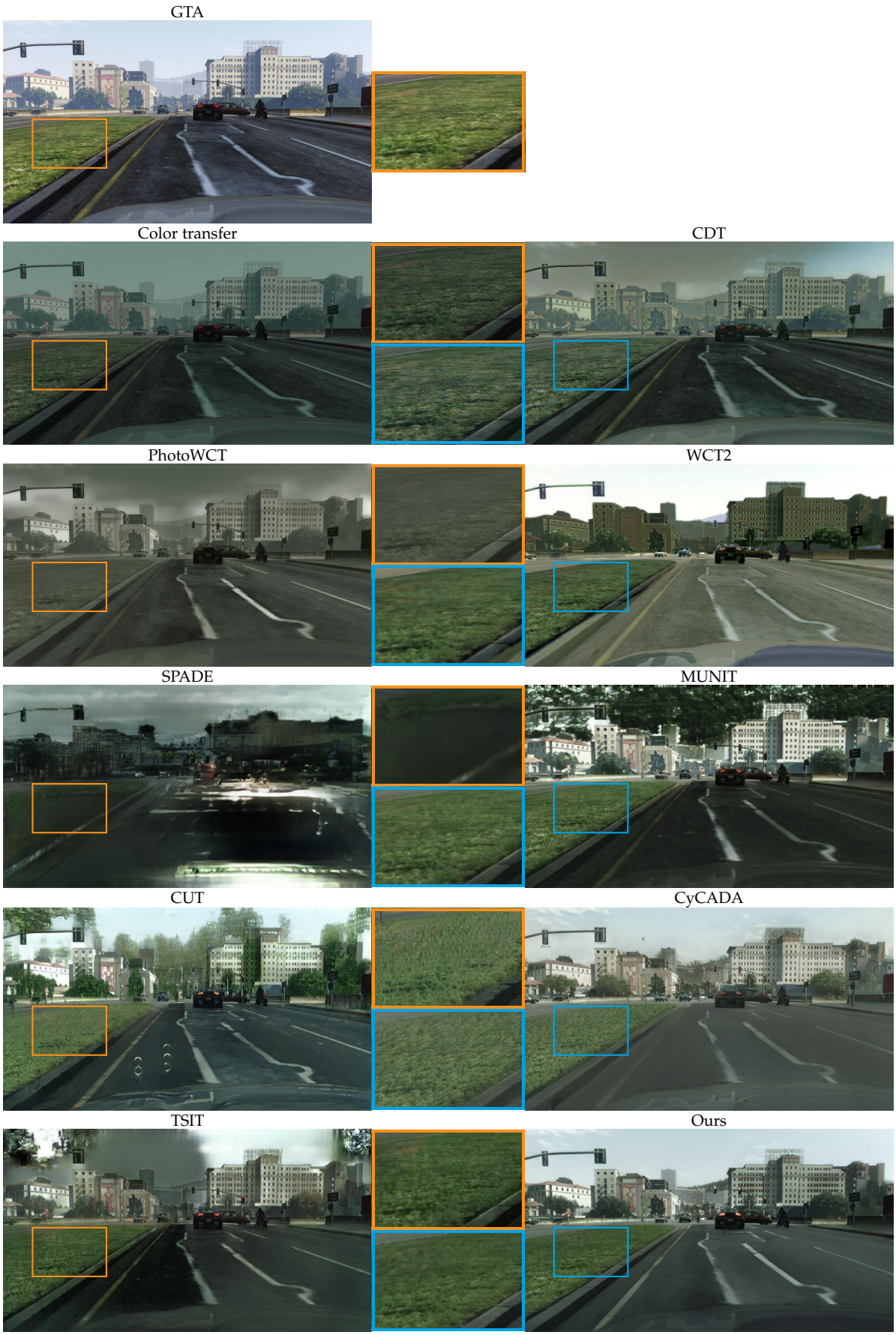

Fig. 3: We compare our results to original GTA images and a number of baselines. Images from GTA are randomly sampled. Insets show details of the respective image. (Continued)

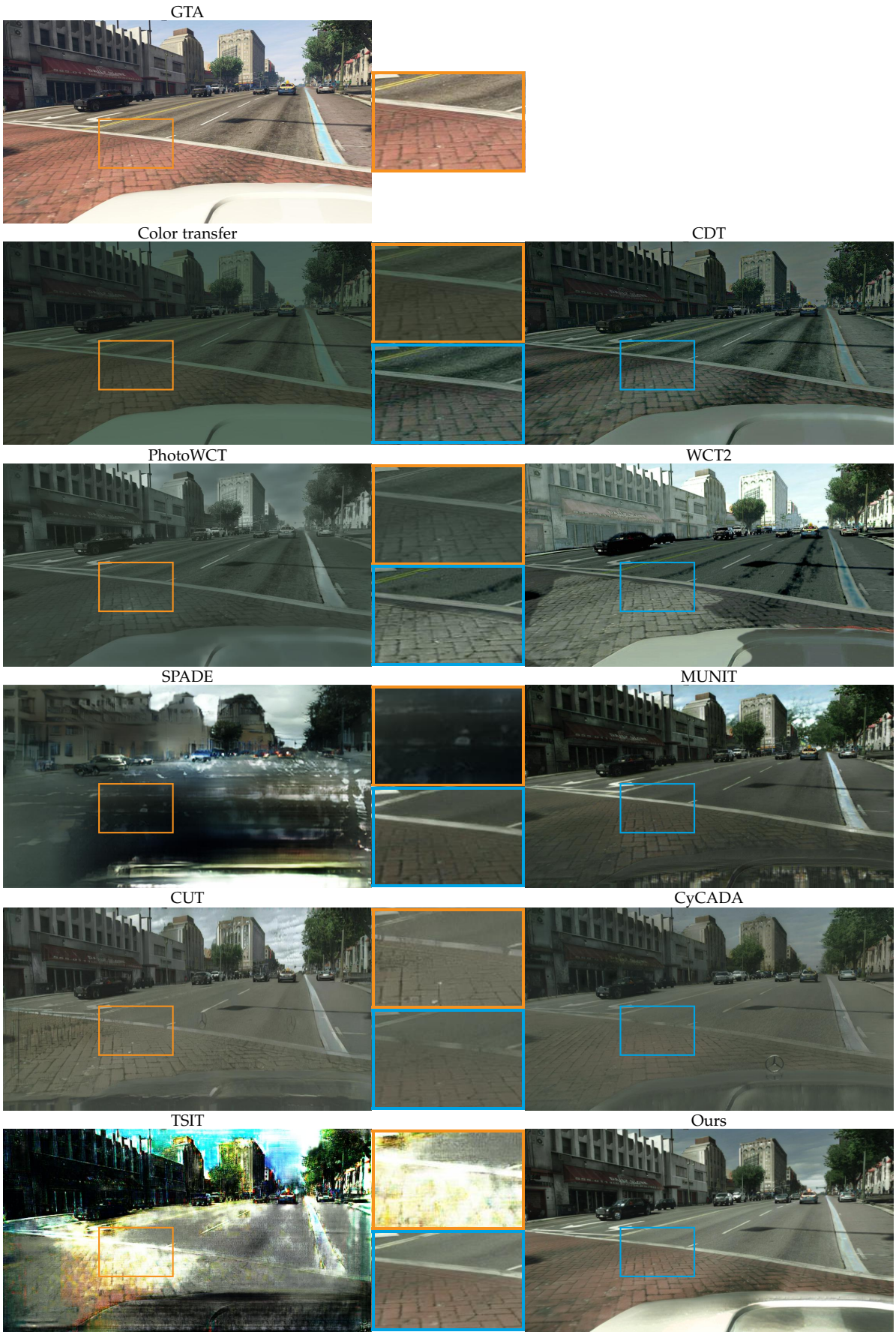

Fig. 3: We compare our results to original GTA images and a number of baselines. Images from GTA are randomly sampled. Insets show details of the respective image. (Continued)

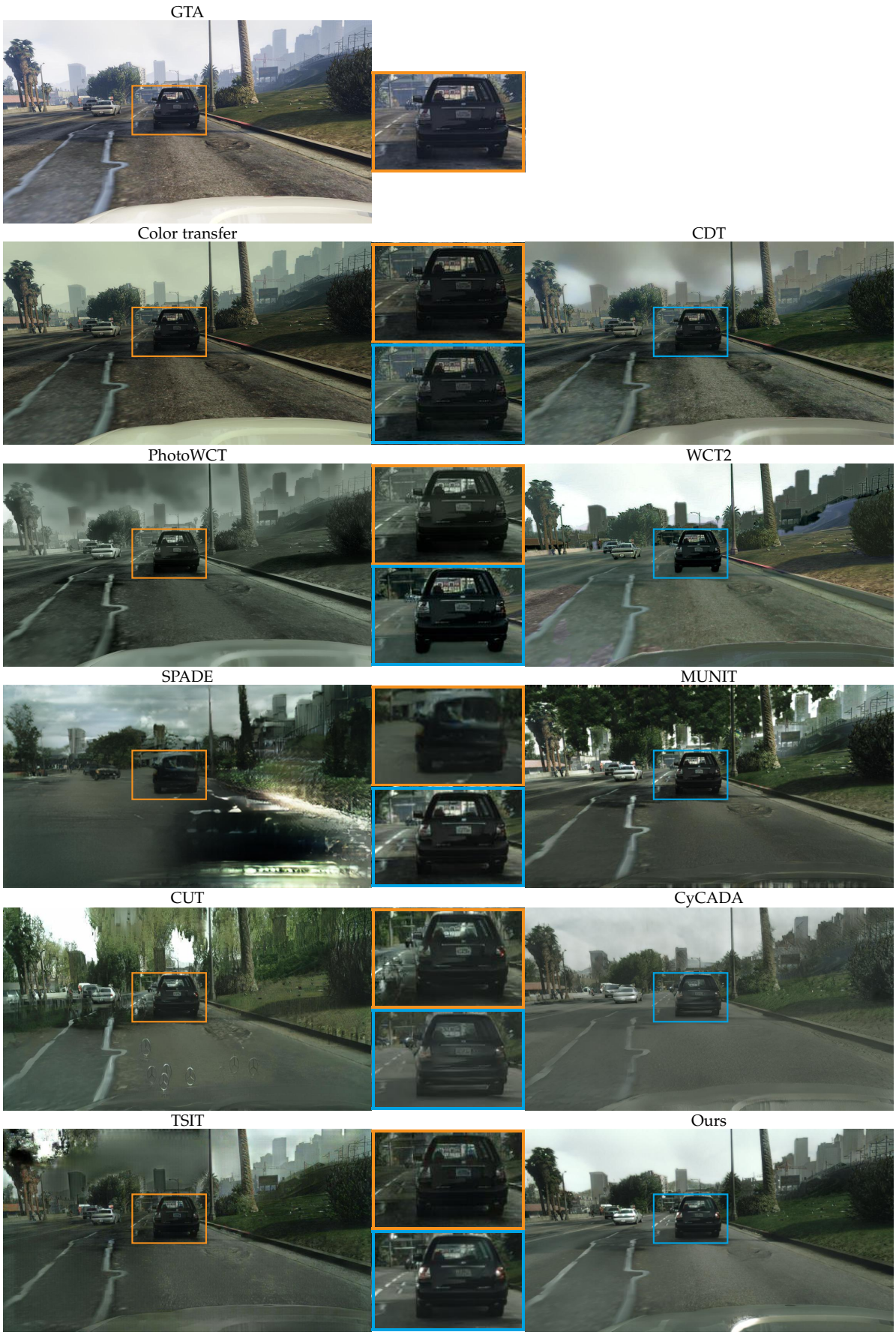

Fig. 3: We compare our results to original GTA images and a number of baselines. Images from GTA are randomly sampled. Insets show details of the respective image. (Continued)

Fig. 3: We compare our results to original GTA images and a number of baselines. Images from GTA are randomly sampled. Insets show details of the respective image. (Continued)

Fig. 3: We compare our results to original GTA images and a number of baselines. Images from GTA are randomly sampled. Insets show details of the respective image. (Continued)

Fig. 3: We compare our results to original GTA images and a number of baselines. Images from GTA are randomly sampled. Insets show details of the respective image. (Continued)

Fig. 4: More results of enhancing GTA images with Mapillary Vistas as the target dataset.

Fig. 4: More results of enhancing GTA images with Mapillary Vistas as the target dataset. (Continued)