Supplementary Material for
Example-Guided Style-Consistent Image Synthesis from Semantic Labeling

Abstract

In this supplementary document, we present more details of datasets, network architecture and training, as well as further experimental results.

1. Datasets

As described in the main manuscript, we evaluate our model on face, dance and street view image synthesis tasks, using following datasets and semantic functions:

Sketch→Face. We use the real videos in the FaceForensics dataset [7], which contains 854 videos of reporters broadcasting news. We use the image sampling strategy described in Subsection 3.3 of the main manuscript to acquire training image pairs from video, then apply face alignment algorithm [5] to localize facial landmarks, crop facial regions and resize them to size 256×256. We sample $20,000$ images from videos for training and 500 images from distinct videos for testing. The detected facial landmarks are connected to create face sketches; this is the function $F(\cdot)$, in both training set and test set. For each sketch extracted from a training image, we randomly sample 30 guidance images from other videos for training, and for each testing sketch, we randomly sample 5 guidance images from other videos for testing.

SceneParsing→StreetView. We use the BDD100k dataset [9] to synthesize street view images from pixel-wise semantic labels (*i.e.* scene parsing) maps. For each street view image in the dataset, the corresponding scene parsing map and WEATHER and TIMEOFDAY attributes are provided. Based on these attributes, we divide images into 13 style groups as listed in Table 1, then sample style-consistent image pairs inside each group and style-inconsistent image pairs between groups. The training set contains $2,000$ images and test set contains 400 images, both resized to width 256. We use scene parsing network DANet [2] as the function $F(\cdot)$ for each street view image during testing. For each scene parsing map, we randomly select an image inside each style group as the guidance, both in training and testing phases.

Group	Weather	Timeofday
1	-	Night
2	Foggy	Dawn or Dusk
3	Overcast	Daytime
4	Rainy	Dawn or Dusk
5	Snowy	Dawn or Dusk
6	Clear	Dawn or Dusk
7	Foggy	Daytime
8	Partly cloudy	Dawn or Dusk
9	Rainy	Daytime
10	Snowy	Daytime
11	Clear	Daytime
12	Overcast	Dawn or Dusk
13	Partly cloudy	Daytime

Table 1: Style groups we used to categorize BDD100K street view images.

Pose→Dance. We downloaded 150 solo dance videos from YouTube, cropped out the central body regions and resized them to size 256×256. As the number of videos is small, we evenly split each video into the first part and the second part along the timeline, then sample training data only from the first parts and sample testing data only from the second parts of all the videos. The function $F(\cdot)$ is implemented using concatenated pre-trained DensePose [6] and OpenPose [1] pose detection results to provide pose labels. As a result, we have $35,000$ images for training and 500 images for testing. For each pose extracted from a training image, we randomly sample 30 guidance images from other dancing videos, and for each testing pose, we randomly sample 5 guidance images from other dancing videos.

2. Network Architectures

2.1. Generator

We follow the naming convention used in Johnson et al. [4], CycleGAN [10] and pix2pixHD [8]. Let `c7s1-k` denote a 7×7 Convolution-InstanceNorm-ReLU layer with k filters and stride 1. `dk` denotes a 3×3 Convolution-InstanceNorm-ReLU layer with k filters and stride 2. Reflection padding is used to reduce boundary artifacts. `Rk×t`

denotes residual blocks each contains two 3×3 convolutional layers with k filters, repeated t times. `uk` denotes a 3×3 fractional-strided-Convolution-InstanceNorm-ReLU layer with k filters and stride 0.5.

The architecture of generator is represented as:

```
c7s1-64, d128, d256, d512, d1024,
R1024×9, u512, u256, u128, u64, c7s1-3
```

2.2. Discriminators

We use 35×35 PatchGAN [3] in both of the two discriminators D_R and D_{SC}. Let `Ck` denote a 4×4 Convolution-InstanceNorm-LeakyRU layer with k filters and stride 2. The last layer is send to an extra convolution layer to produce a 1 dimensional output. InstanceNorm is not used for the first `C64` layer. Leaky ReLU slope is set as 0.2.

The architectures of D_R and D_{SC} are identical:

```
C64, C128, C256, C512
```

3. Training Details

All the networks were trained from scratch. Weights were initialized from a Gaussian distribution with mean 0 and standard deviation 0.02. In the first 250K iterations, the learning rate was fixed as 0.0002 with the adversarial style-consistency loss $\mathcal{L}_{\text{SCAdv}}$ turned-off. In the next 250K iterations, we turned on the $\mathcal{L}_{\text{SCAdv}}$ loss. In the final 500K iterations, the learning rate linearly decayed to zero with all the losses turned-on.

The models were trained on an NVIDIA TITAN 1080 Ti GPU with 11GB memory. The inference time is about 8-10 milliseconds per 256×256 image.

4. Additional Results

In Figure 1 and following pages, we show further experimental results from our method and baselines.

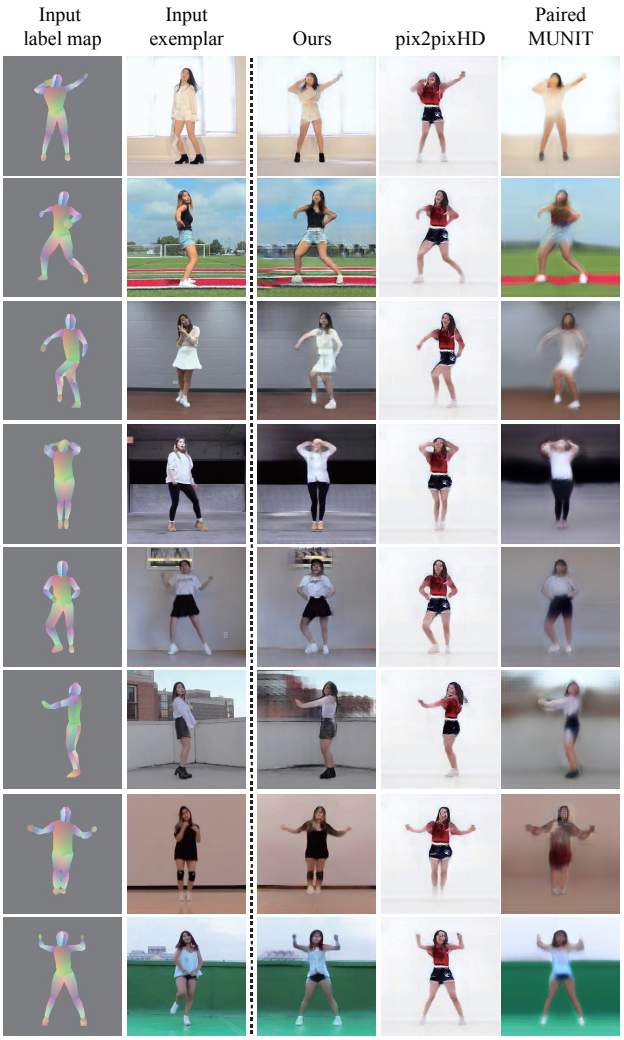

Figure 1: Example-based dance image synthesis YouTube Dance dataset. The first column shows the input pose labels, the second column shows the input style examples, next columns show the results from our method, pix2pixHD and PairedMUNIT.

References

[1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 1

[2] Jun Fu, Jing Liu, Haijie Tian, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. *arXiv preprint arXiv:1809.02983*, 2018. 1

[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arxiv*, 2016. 2

[4] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 1

[5] Davis E King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(Jul):1755–1758, 2009. 1

[6] Iasonas Kokkinos Riza Alp Güler, Natalia Neverova. Densepose: Dense human pose estimation in the wild. *arXiv*, 2018. 1

[7] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv*, 2018. 1

[8] 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. 1

[9] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018. 1

[10] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A

Figure 2: More results of dance synthesis. The first column shows input pose maps. The first row shows input dance exemplars. Other images are the synthetic dance results.

Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 1

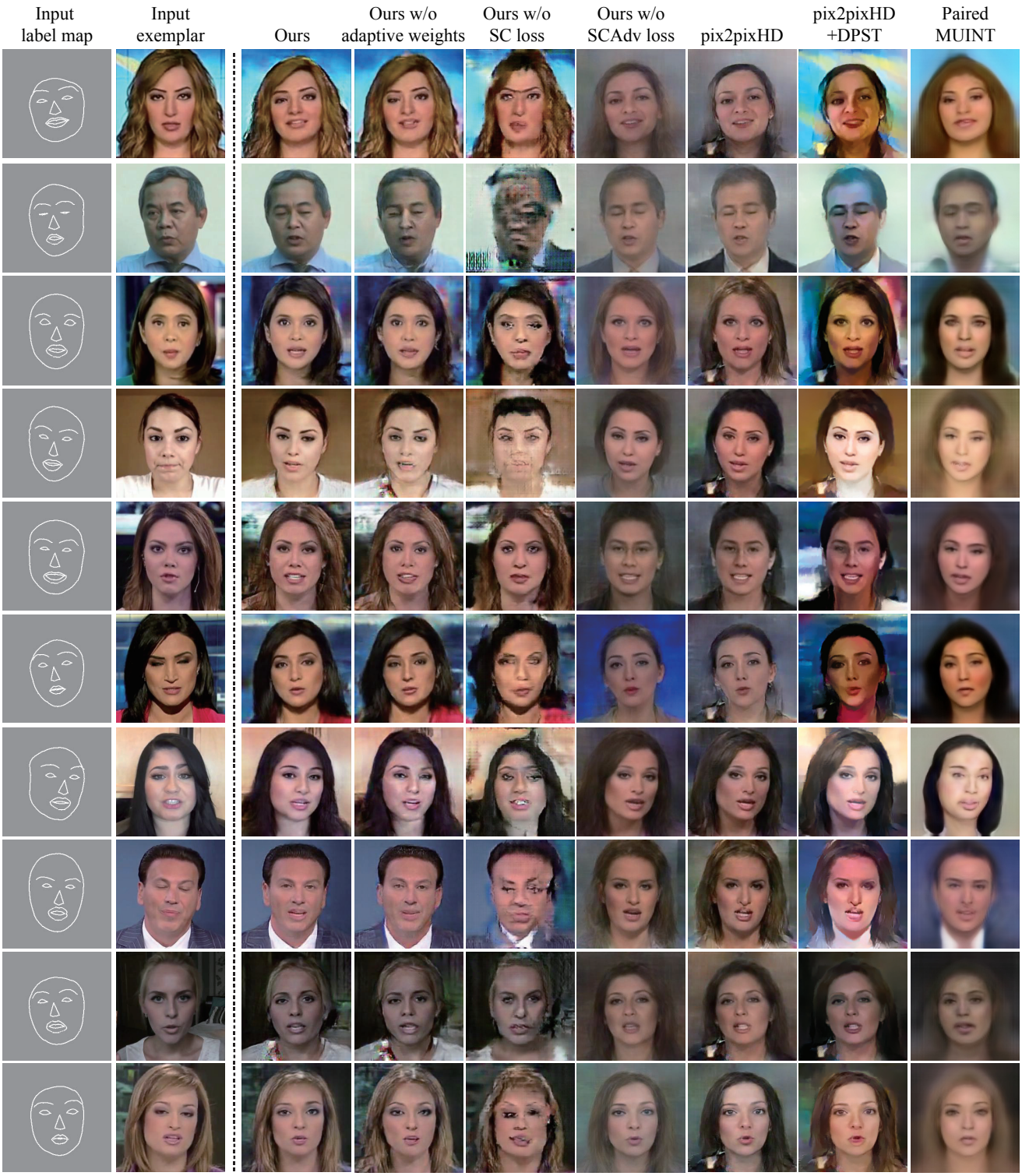

Figure 3: Example-based face image synthesis on the FaceForensics dataset. The first column shows the input labels, the second column shows the input style example, next columns show the results from our method and our ablation studies, pix2pixHD, pix2pixHD+DPST and PairedMUNIT.

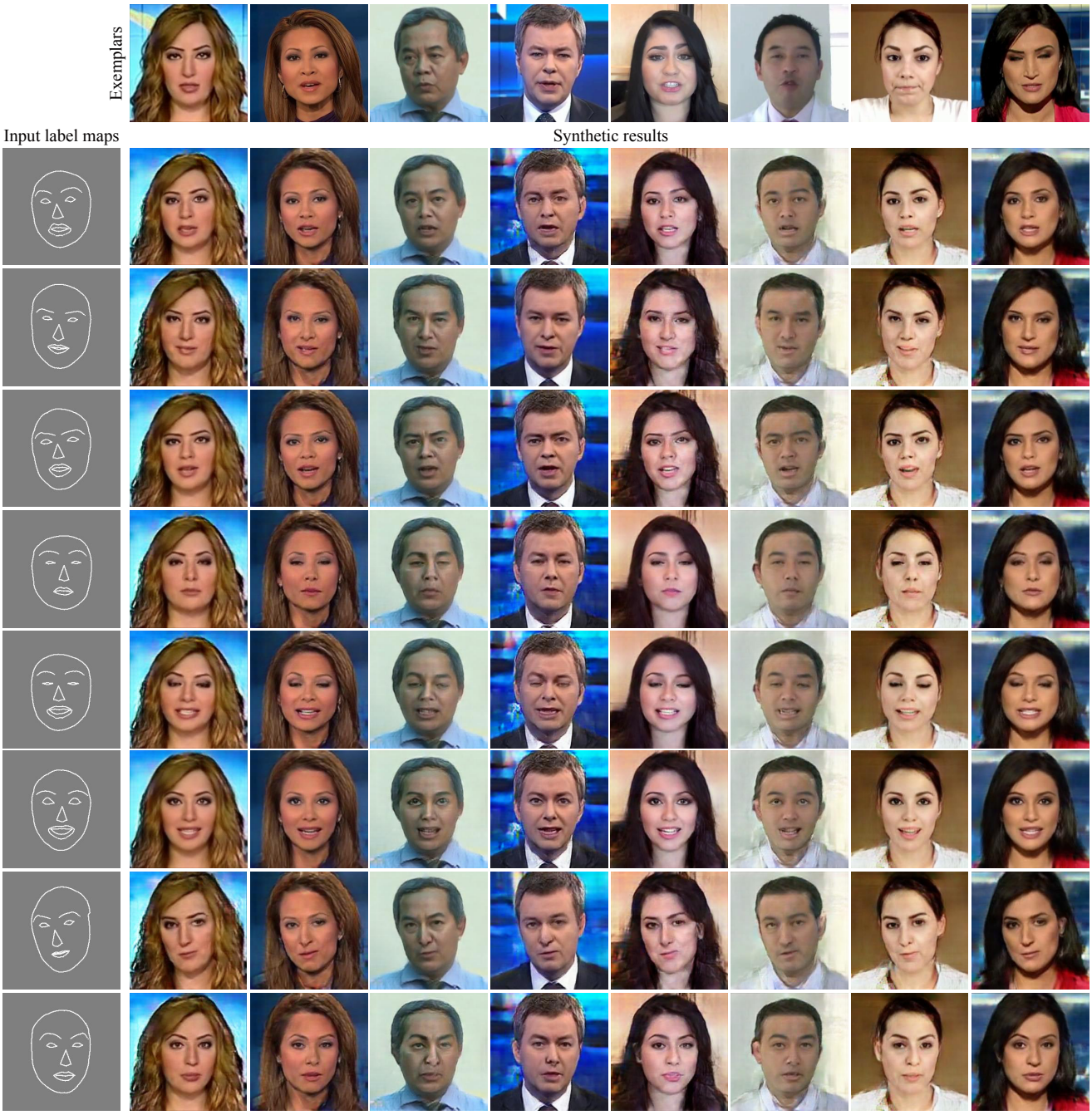

Figure 4: More results of face synthesis. The first column shows input sketch maps. The first row shows input face exemplars. Other images are the synthetic face results.

Figure 5: More in-the-wild Sketch→Face results. The model is trained on our training dataset and tested on Internet images.

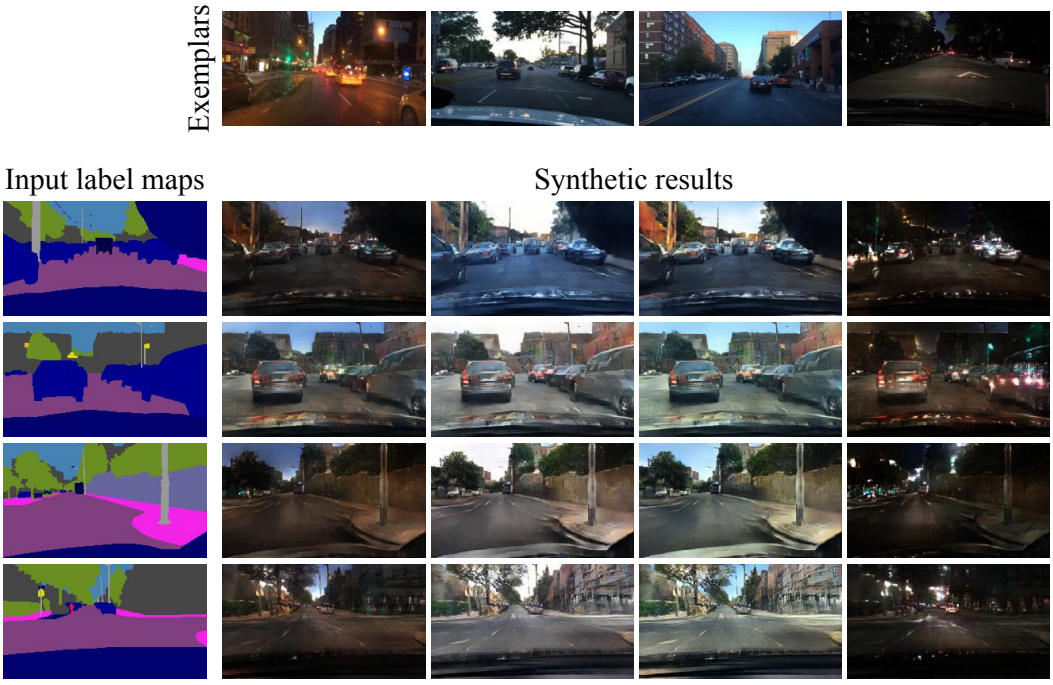

Figure 6: More results of street view synthesis. The first column shows input segmentation maps. The first row shows input exemplars. Other images are the synthetic street view results.