Zooming Out on Zooming In: Advancing Super-Resolution for Remote Sensing Supplementary Material

Piper Wolters Favyen Bastani Aniruddha Kembhavi
Allen Institute for AI
{piperw,favyenb,anik}@allenai.org

A. Supplementary Material

We provide additional details about the super-resolution human judgement dataset, CLIP for satellite data, the urban land use dataset that is used as a downstream task, as well as training and model details for experiments in the main paper. We also describe a few diffusion and GAN model variations. Following that, we provide visualizations of model outputs from the WorldStrat and S2-NAIP datasets.

A.1. Human Judgement Dataset

To build this dataset, we used Amazon Mechanical Turk (AMT). Prior to the official task, they were tested with a qualification task to ensure they had high level of agreement with other workers, and then afterward during annotation, workers with low agreement with other workers were removed, so that workers who would suddenly begin picking arbitrary images rather than examining the images carefully would be removed.

We ran the study on a large set of outputs from the WorldStrat and S2-NAIP datasets. Model outputs from SRCNN, HighResNet, ESRGAN, and SR3 were used for WorldStrat; the same were used for S2-NAIP in addition to multiple ESRGAN model checkpoints to add more diversity in output quality.

A.2. CLIP for Satellite Data

In the main paper, we mention our attempt to train a CLIP-like model for a new super-resolution metric. We try training a model, using the open_clip codebase and imagery from our S2-NAIP dataset, using 1) Sentinel-2 and NAIP data pairs and 2) sets of NAIP images from different timestamps. The human-correspondence for these two models was 64.54% and 72.58%, respectively, which did not compete with the other metrics in the study.

A.3. Experiment Details

For all experiments in the method study, models are trained from scratch. We use the Adam optimizer, and initialize the learning rate to 10^{-4}.

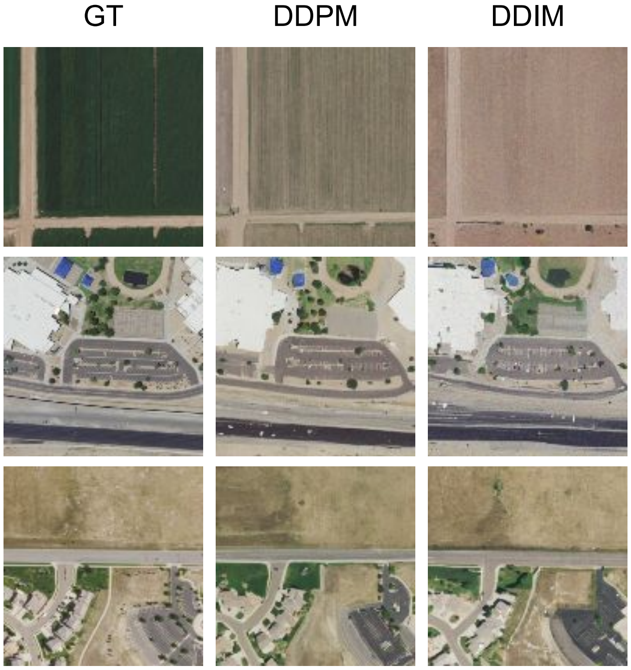

Figure 1. Example outputs of DDPM and DDIM.

Specific to the ESRGAN, we employ the Residual in Residual Dense Block Network (RRDBNet). For the experiments with varying model sizes, we use 64, 128, and 256 features and 32, 64, and 128 grow channels as well as 23, 23, and 30 blocks, all respective to the small, medium, and large sizes.

S2-NAIP. All experiments in the main paper just used the RGB Sentinel-2 bands and eight Sentinel-2 images as input. In Figure 4, we show how performance changes with 1, 2, 4, 8, and 16 Sentinel-2 images as input. We find a big jump in performance between 1-2 images and 4 images, but small performance gains after that. We input a time series of 32x32 pixel low-resolution images into each model and these are upsampled to 128x128 pixels, matching the target images downsampled four times.

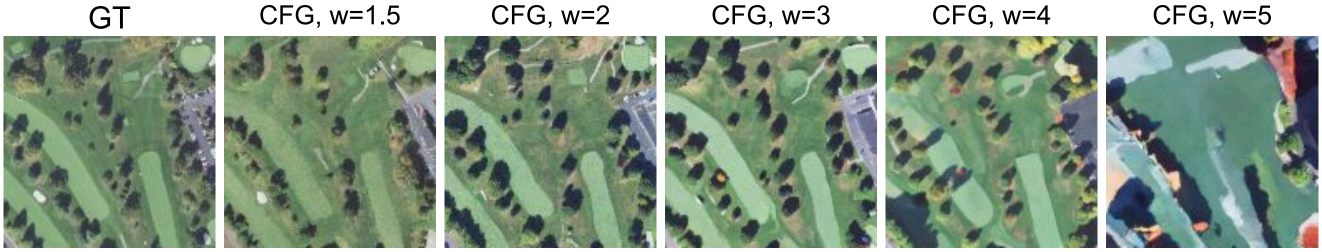

Figure 2. Outputs from training a model with 20% null datapoints (black images) and then using CFG at inference time with varying weight values.

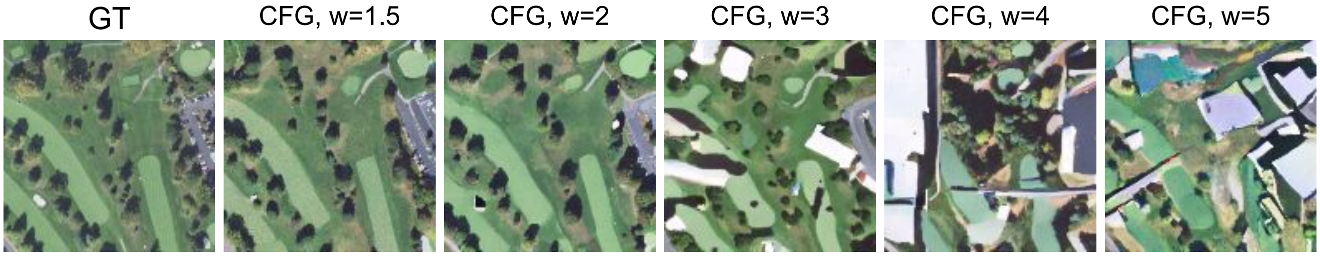

Figure 3. Outputs from using a pretrained model with CFG at inference time with varying weight values.

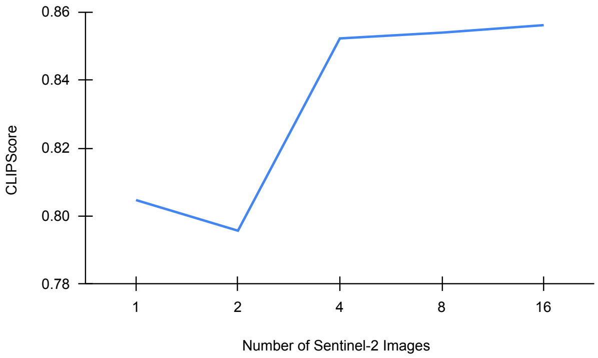

Figure 4. Plot of CLIPScore with varying number of Sentinel-2 images as input.

WorldStrat. We input a time series of 8 160x160 pixel low-resolution images, with just the RGB bands, and models generate outputs that have a spatial resolution of 640x640 pixels.

OLI2MSI. We input one 160x160 pixel low resolution image and models super-resolve this by a factor of 3 to a 480x480 pixel spatial resolution.

PROBA-V. We input a time series of 9 128x128 pixel low-resolution images and models generate outputs of shape 384x384. Because the official test set is not public, we set aside 10% of the available data for testing.

A.4. Diffusion Model Variations

We refer to experimentation with Diffusion Implicit Models (DDIM) and Classifier Free Guidance (CFG).

DDIM is knowingly much faster as inference time and we found that to be true, with inference being at least 10x faster, but it also caused a 7 point drop in CLIPScore. Example outputs are shown in Figure 1. DDIM struggles more with components such as colors, roads, and buildings.

CFG has shown promise in other domains. We try using CFG at inference time with a previously trained SR3 model; and experiment with scale values of 1.5, 2, 3, 4, and 5 (the scale value controls how much influence the input image will have on the generated output). Figure 2 shows samples for each of these scales in comparison to the target and the original model. We also try training with some percentage of null datapoints (black images) and then using CFG at inference, as described in the original work. We show the outputs on the same datapoint, with the same weights in Figure 3, after training on 20% null datapoints. Although the CFG scale is typically between 7-13, we found that a scale of 1.5 or 2 was the highest we could go before the outputs became too animated.

A.5. ESRGAN Variations

In the main paper, in Section 6, we mention incorporating domain knowledge into the training pipeline. We describe the three most beneficial techniques here.

1. First, we introduce an object-discriminator which, similar to PatchGAN [2], looks at individual patches of

an image and assigns real/fake to each patch. In our case, these patches are extracted from the image based on polygon boundaries of infrastructure like buildings, sports fields, and power facilities, provided by OpenStreetMap [3]. We hope this forces the discriminator to pay extra attention to the sharpness and structure of things like building edges. For this experiment, images with less than at least one OSM object is thrown out. The architecture used for this is a simple 9-layer convolutional neural network. During training, one object is picked at random from each image and input to the object discriminator. The object discriminator loss is added to the main discriminator loss with a weight of 0.1. This results in a 6 point increase in CLIPScore.

2. Second, we find that feeding the discriminator a high-resolution image of the current training datapoint's location, at an older timestamp, is beneficial. This presumably gives the discriminator context of the current location, thus improving it's ability to deduce whether an output is real or fake. Because NAIP imagery is free and dates back many years, we downloaded imagery between 2016-2018, so we would have at least one image for each datapoint in the S2-NAIP dataset. The old NAIP image is just stacked onto the real/fake image before being input into the discriminator. This leads to a 2 point gain.
3. Third, we simply load weights from SatlasPretrain [1] into the generator of ESRGAN. As these weights are trained on a very large-scale remote sensing dataset, it is reasonable that this improves performance on a downstream task such as super-resolution. This provides a 1 point gain.

A.6. OpenStreetMap Dataset

The dataset built for the experiments in Section 7 of the main paper consists of 6,144 128x128 Sentinel-2 images with OpenStreetMap [3] labels. The dataset includes eight binary segmentation categories: roads, buildings, footpaths, rails, park land, water bodies, sports fields, and airports.

A.7. Qualitative Results

We provide example outputs from HighResNet, the best performing L2 loss-based method, and ESRGAN, the best generative method, on the S2-NAIP and WorldStrat datasets. S2-NAIP results are show in Figures 5 and 6. WorldStrat examples are shown in Figures 7 and 8.

References

[1] Favyen Bastani, Piper Wolters, Ritwik Gupta, Joe Ferdinando, and Aniruddha Kembhavi. Satlaspretrain: A large-scale dataset for remote sensing image understanding. *ICCV 2023*, 2023. 3

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2016. 2

[3] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017. 3

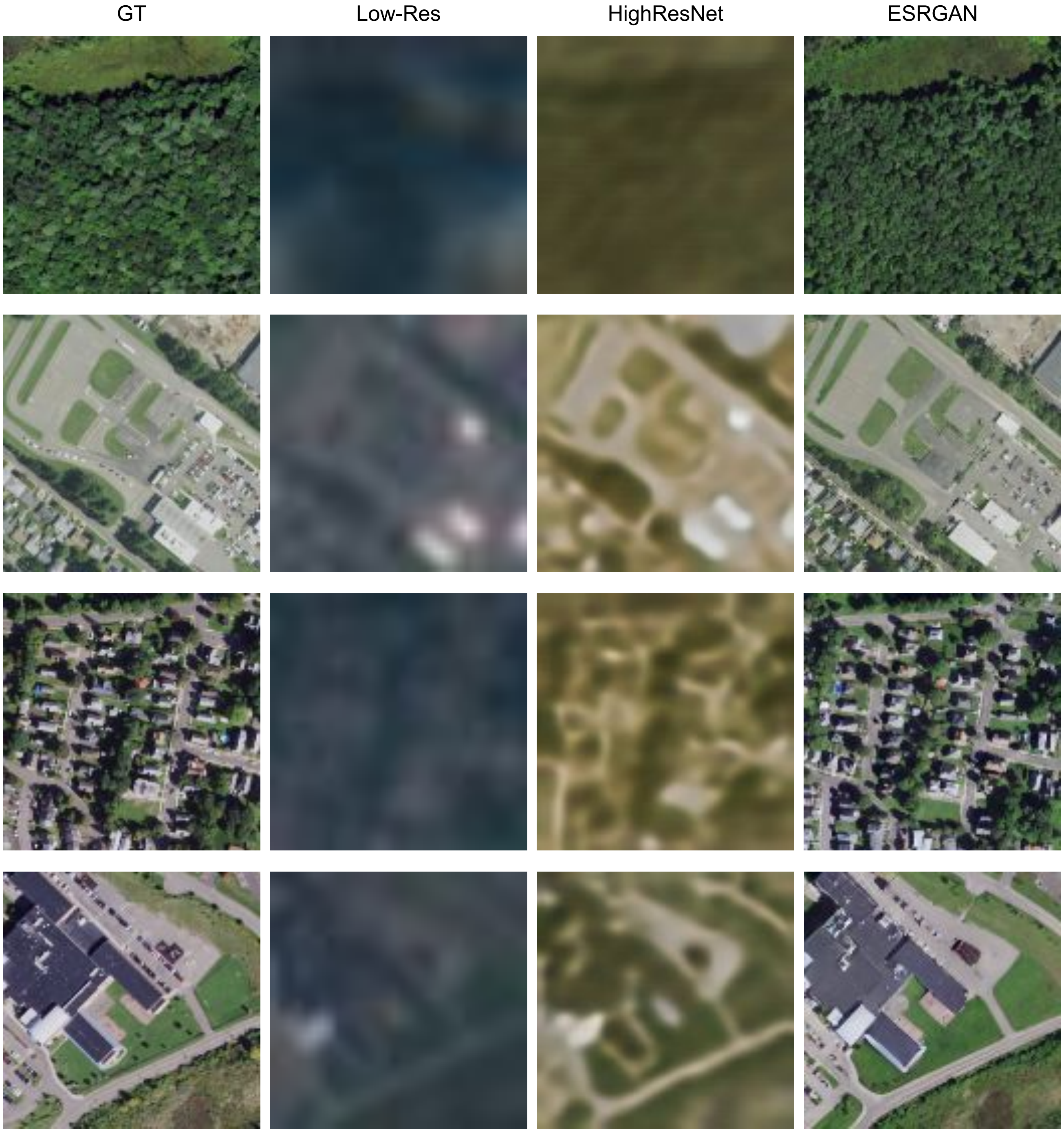

Figure 5. Example outputs from HighResNet and ESRGAN on the S2-NAIP dataset.

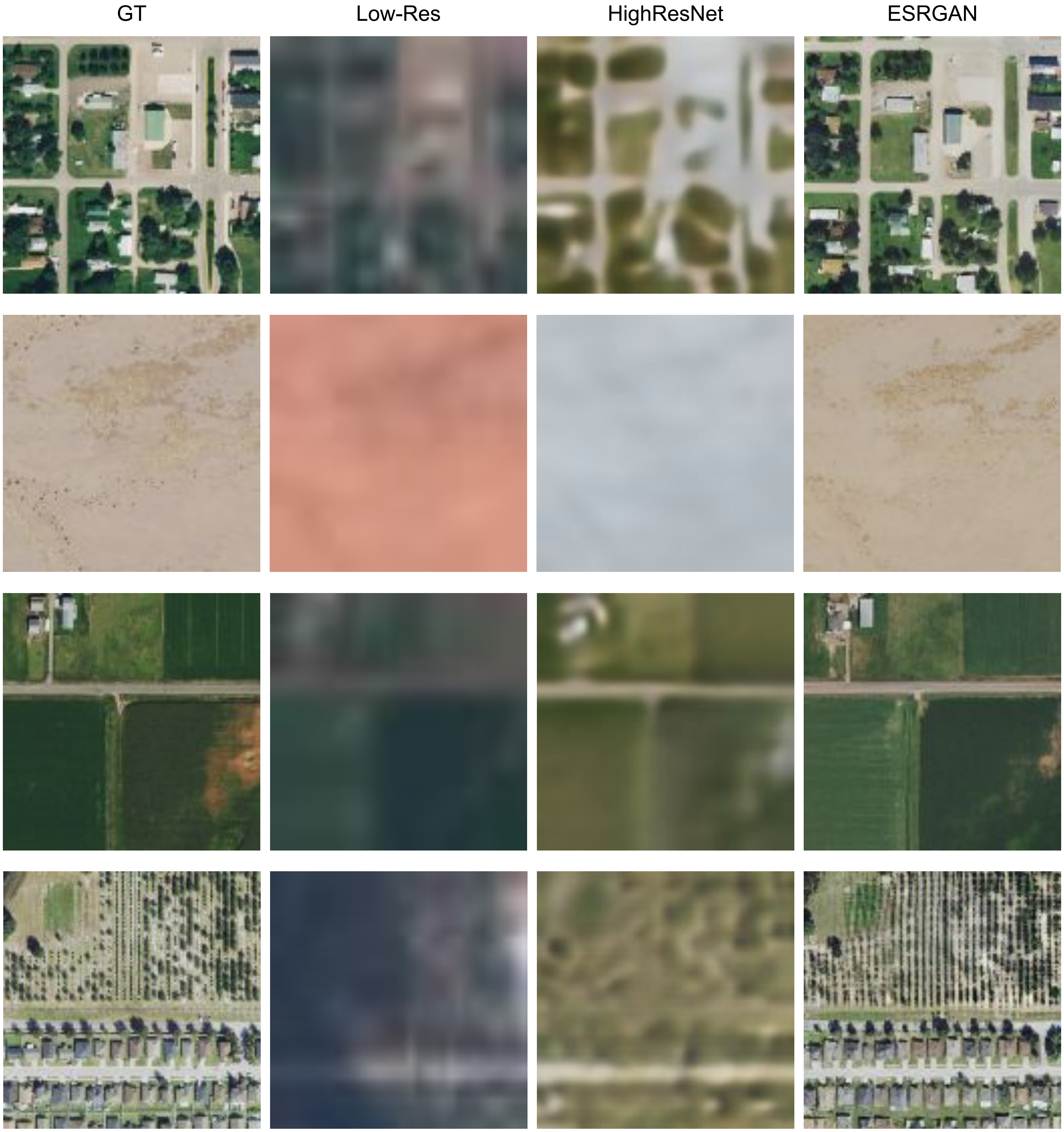

Figure 6. Example outputs from HighResNet and ESRGAN on the S2-NAIP dataset.

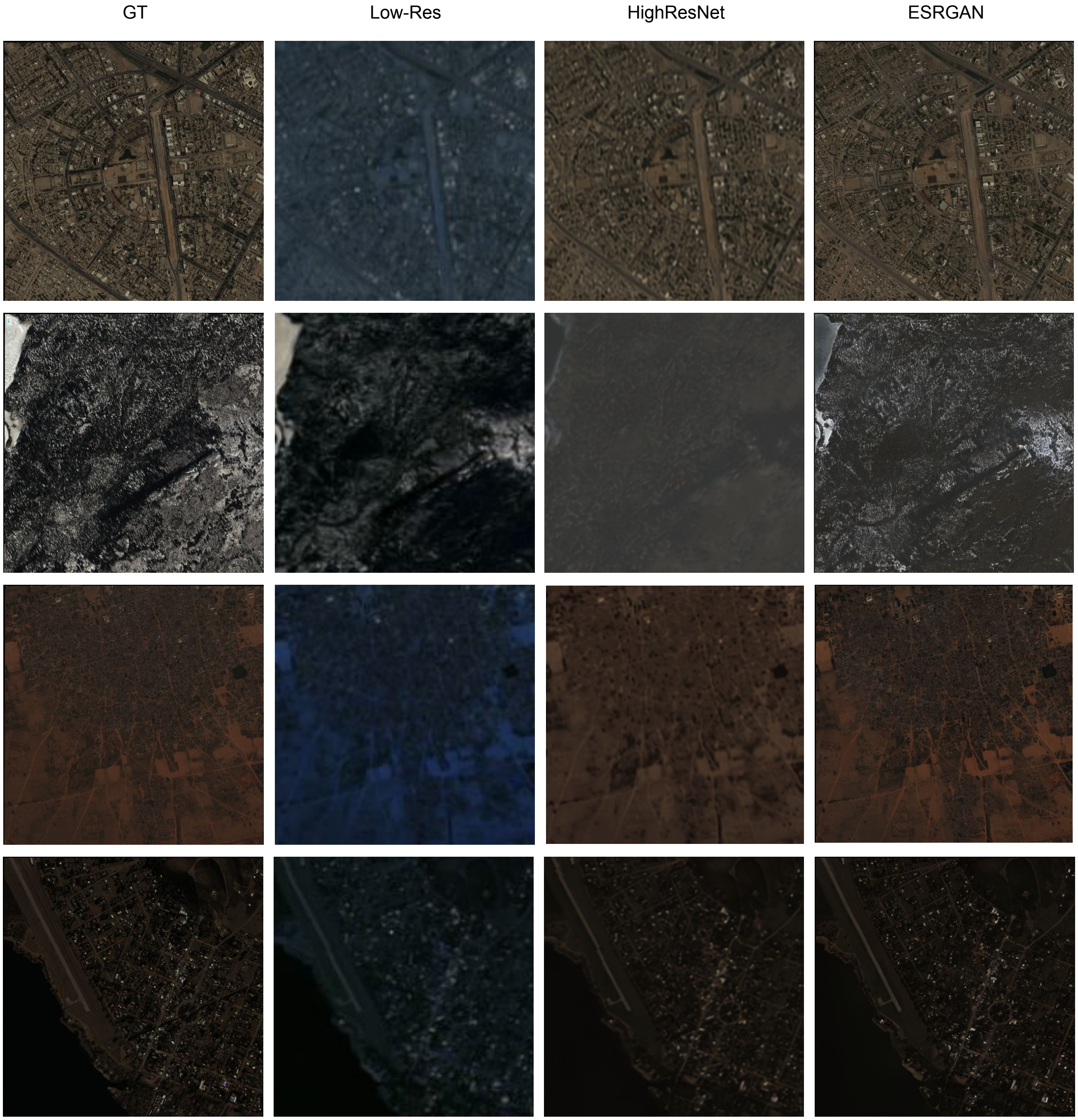

Figure 7. Example outputs from HighResNet and ESRGAN on the WorldStrat dataset.

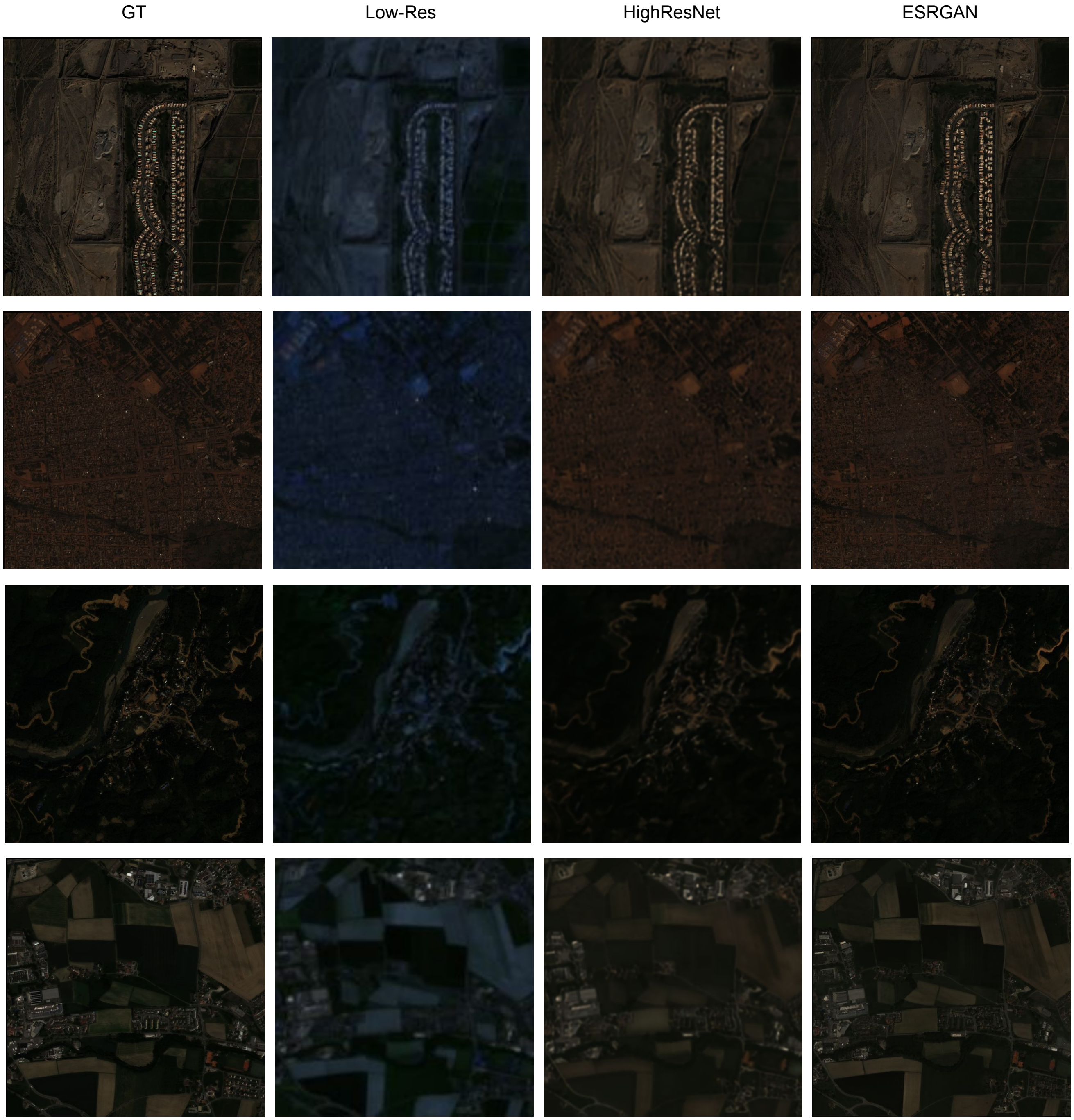

Figure 8. Example outputs from HighResNet and ESRGAN on the WorldStrat dataset.