Cloud Removal in Satellite Images Using Spatiotemporal Generative Networks: Supplementary Materials

1. Image Crop Classification

We aim to take publicly-available Sentinel-2 images from 32270 distinct regions, or tiles, where each tile is a 10980×10980-pixel image with a resolution of 10m/pixel, and generate 256×256-pixel image crops that are classified as clear or cloudy images. The cloud mask detection algorithm proposed by Hollstein et al.[2] can classify the entire tile as clear or cloudy based on the percentage cloud cover it detects. However, how can we classify individual 256×256-pixel image crops? The larger 10980×10980-pixel tile is not necessarily uniform in cloud cover. As a result, after applying Hollstein et al.'s algorithm [2] to determine whether an image is clear (cloud cover < 1%) or cloudy (cloud cover > 10%), we apply a thresholding heuristic to each crop to determine an individual crop's level of cloud cover.

Opaque cloud cover of the individual crop can be heuristically detected through measuring the ratio of cloudy pixels in an image crop, r^c. We finally categorize the image crops into two different groups: (1) *cloudy*, $\{Z_\ell^t\}_{t,\ell}$ and (2) *not cloudy*, $\{X_\ell^t\}_{t,\ell}$. This classification is based on the following ratio thresholds:

$$\begin{cases} r^c < 0.01 & Cloud-Free \\ r^c > 0.01 \ and \ r^c < 0.10 & Discard \\ r^c > 0.10 \ and \ r^c < 0.30 & Cloudy \\ r^c > 0.30 & Discard \end{cases} \quad (1)$$

We discard the images with more than 30% cloud cover by our heuristic, as many of the images labeled with higher than 30% of cloud cover are simply indecipherable upon manual inspection. The importance of a two-stage cloud detection process is illustrated in Figure 1: the large tile is classified as "cloudy" by the cloud mask detection algorithm, but one of the two extracted crops appears to be clear and is detected by the thresholding heuristic. We discard any crops that are labeled clear but part of a cloudy tile or vice versa to maximize quality of the dataset.

2. Pix2Pix Model Description

The objective function of the overall Pix2Pix model consists of a conditional GAN loss and an L1 loss, with the weighted sum parameterized by the hyperparameter λ^s:

$$\mathcal{L}_{cGAN}(G^s, D^s) = \mathbb{E}_{Z_\ell^t, X_\ell^t}[logD^s(Z_\ell^t, X_\ell^t)] + \mathbb{E}_{Z_\ell^t}[log(1 - D^s(Z_\ell^t, G^s(Z_\ell^t))] \quad (2)$$

$$\mathcal{L}_{L1}(G^s) = \mathbb{E}_{Z_\ell^t, X_\ell^t}[||X_\ell^t - G^s(Z_\ell^t)||_1] \quad (3)$$

$$G^{s*} = arg \min_{D^s} \max_{G^s} \mathcal{L}_{cGAN}(G^s, D^s) + \lambda^s \mathcal{L}_{L1}(G^s) \quad (4)$$

where G^s, and D^s represent the generator and discriminator networks. The input to the discriminator, D^s, is a clear image, X_ℓ^t, or a fake clean image, $\hat{X}_\ell^t = G^s(Z_\ell^t)$, generated by G^s. Thus, the overall min-max competitiveness of the Pix2Pix model can be interpreted as it attempts to fool the discriminator by generating cloud-free images, $\hat{X}_\ell^t$, similar to the real cloud-free images, X_ℓ^t.

The Pix2Pix discriminator is slightly non-standard in that it has a PatchGAN architecture, meaning that different patches of $\hat{x}$, and x are evaluated in parallel, and each of the values in the 30×30×1 output corresponds to a 70×70×3 patch from the original images. This allows it to have greater granularity and specificity as it returns much more information than the standard binary label.

3. STGAN Model Details

We construct a spatiotemporal generative (STGAN) model called the branched ResNet. The branched ResNet is composed of individual blocks where each ResNet block is a conv block with skip connections, as mentioned in Section 5.2. The ResNet block is composed of a conv layer, a normalization layer, and a non-linearity layer (ReLU). It takes in account the number of channels in the conv layer, the name of padding layer (reflect, replicate or zero), the normalization layer, dropout layer, and bias.

Similarly, we construct another STGAN called the branched U-Net. In order to do so, we construct a U-Net submodule with skip connections and recursively use that to construct the model as mentioned in Section 5.2. The U-Net submodule takes into account the number of filters in the outer conv layer, the number of filters in the inner conv layer, the number of channels in input images/features, previously defined submodules (recursive approach), if the

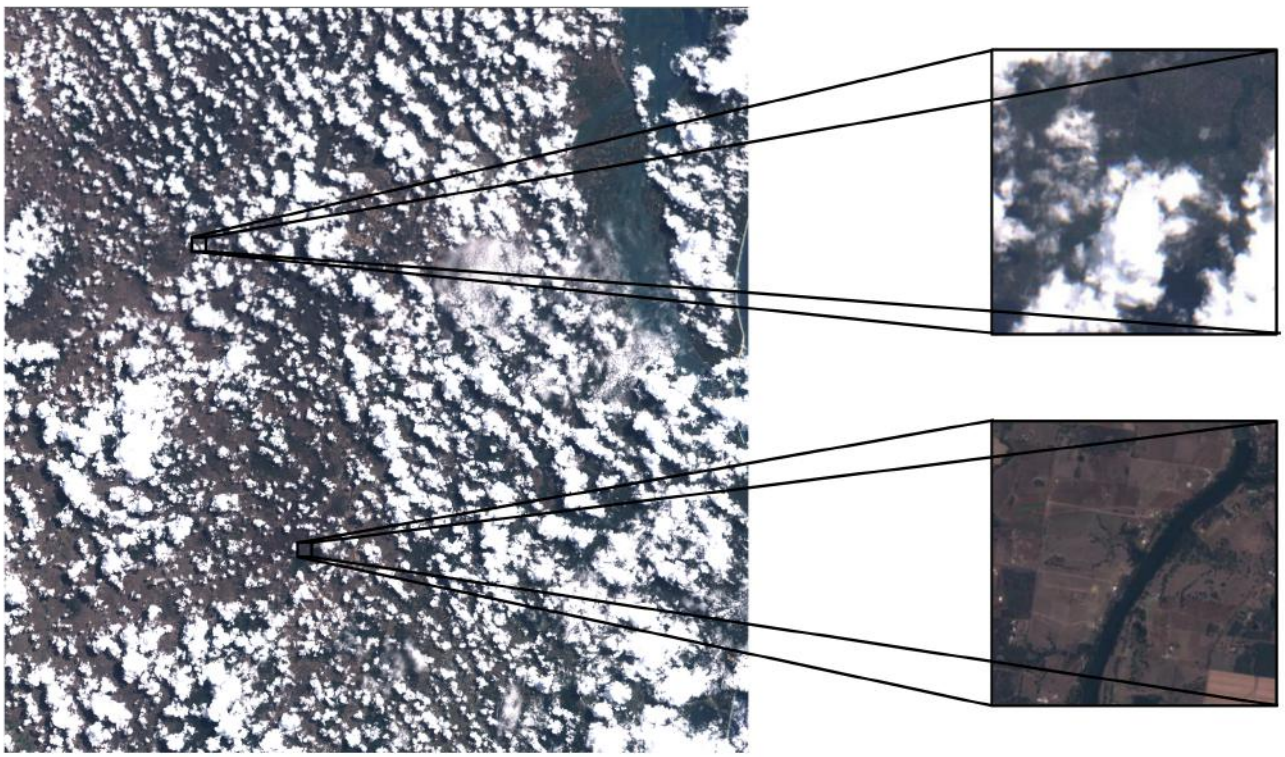

Figure 1: An illustration of the large tile to individual image crops pipeline. The large tile is classified as cloudy by Hollstein et al.'s algorithm [2] as cloudy. Thresholding heuristics confirm one of the crops as cloudy, whereas the other is detected as clear and is discarded.

module being used is the outermost module, if the module being used is the innermost module, normalization layer and dropout layers.

Using the defined ResNet block and U-Net submodule, we are able to build up and customize the STGAN architecture as thoroughly described in the paper.

4. Experimental Details

Training, validation, and test data splits are determined randomly, with 80%, 10%, and 10% of images allocated to the respective splits. For $\mathcal{Y}_{single} = (X_\ell^t, Z_\ell^{t-1})$, this resulted in 78112, 9764, and 9764 image pairs for train, val and test respectively. For $\mathcal{Y}_{temporal} = (X_\ell^t, Z_\ell^{t-1}, \cdots, Z_\ell^{t-T})$, with 3 cloudy images corresponding to each cloud-free image, this resulted in 2481, 310, and 310 image groups for train, val, and test respectively.

For each of the models, we tune the L1-element of the generator loss and the batch size of the input through a random grid search. For each temporal model, separate weights are trained for each of the different "branches." We train our models for 200 epochs, using the Adam optimizer [3] with a momentum of 0.5 and a beta of 0.99. The learning rate of 1e-3 was kept the same for the first 100 epochs and then linearly decayed to zero over the next 100. We determine the optimal hyper-parameters following a random grid search and used the models with highest SSIM in validation.

5. Composite Baseline

We explore image composite techniques as a baseline algorithm. This technique relies on using 13-band spectral data for cloud mask detection, so it incorporates much more information than the techniques explored in our paper. For each of three input cloudy images, we use Hollstein et al.'s algorithm [2] with 13-band Sentinel-2 spectral data to detect binary cloud masks. Then, we return a composite image that, for each pixel, averages all corresponding cloud-free pixels across the three input images. However, for the vast majority of image pairs in $\mathcal{Y}_{temporal}$, some part of the image is covered by all three cloudy images in the training dataset. This leads to "holes" in the composite image where we have no values, as seen in both Figure 3 and Figure 4. These holes are both qualitatively implausible and make the images unusable for downstream tasks. Figure 4 is a particularly egregious example: filmy and widespread clouds lead the cloud detection algorithm to detect clouds covering

nearly all pixels, leading to an extremely incomplete composite image.

6. Downstream Tasks

We evaluate downstream performance by training a baseline model on the *pre-labeled* Eurosat dataset [1], which consists of 27,000 labeled Sentinel-2 satellite images across 10 classes (examples shown in Figure 5). We then went through our test dataset, examined cloud-free images, *manually-annotated* 149 clear images with an approximately even distribution amongst the 10 classes, and retrieved their corresponding cloudy images. Next, the cloudy images were passed through our models to generate cloud-free images. The Eurosat-trained land classification model then made class predictions for each of the generated cloud-free images. The accuracy of the model was then determined by comparing the prediction against the manually-annotated label and the same set of annotated images was used across all models. Note that the land classification model also made predictions on the raw cloudy and cloud-free images to provide reference points for comparison. The same images were used across all models

7. Dataset and Code

Data can be found at:
https://doi.org/10.7910/DVN/BSETKZ.
Code can be found at:
https://github.com/VSAnimator/stgan.

8. Supplementary Images

Figure 2 has additional examples of the performance of the state-of-the-art STGAN (RGB + IR) across a variety of terrains. Figure 5 illustrates the type of classes and images used in the downstream task of land cover classification.

References

[1] P. Helber, B. Bischke, A. Dengel, and D. Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pages 1–10, 2019.

[2] André Hollstein, Karl Segl, Luis Guanter, Maximilian Brell, and Marta Enesco. Ready-to-use methods for the detection of clouds, cirrus, snow, shadow, water and clear sky pixels in sentinel-2 msi images. *Remote Sensing*, 8(8):666, 2016.

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

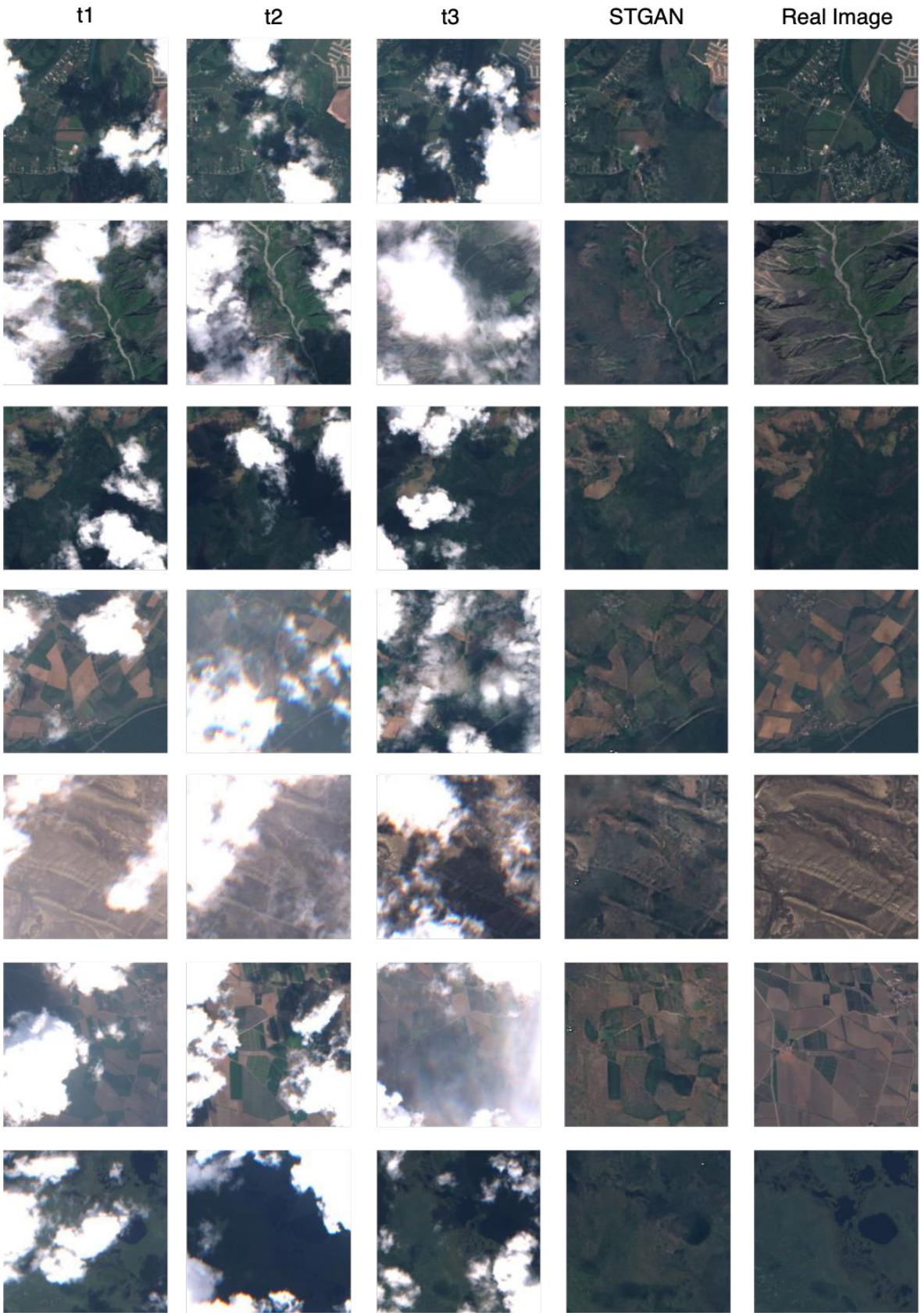

Figure 2: Further examples of cloud-free images generated by the state-of-the-art Resnet-based STGAN utilizing both RGB and IR data. The first three columns represent the three input temporal images, the fourth column is the cloud-free image generated by the STGAN and the fifth is the ground-truth cloud-free image. This model attained top results in both SSIM and PSNR amongst all baselines and models evaluated.

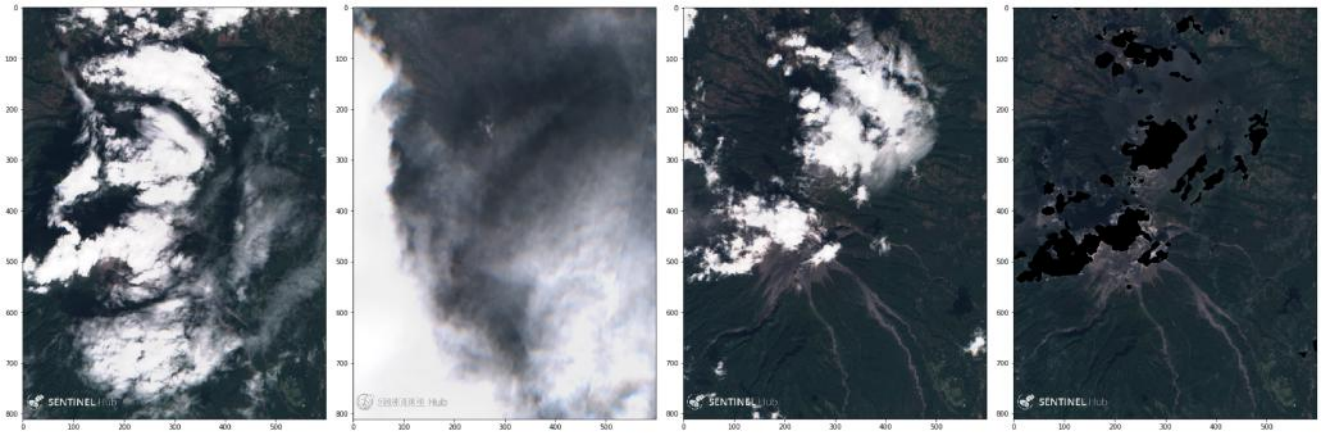

Figure 3: From left to right, three cloudy images and one generated composite "clear" image. The image on the right is generated by first calculating binary cloud masks for the three cloudy images, then calculating pixel values by averaging across all images where a given pixel is not covered by a cloud mask (i.e. marked as "not cloudy"). The assorted black regions (clear pixels could not be found in any of the input images) in the composite image show the consequences of persistent cloud coverage on this technique.

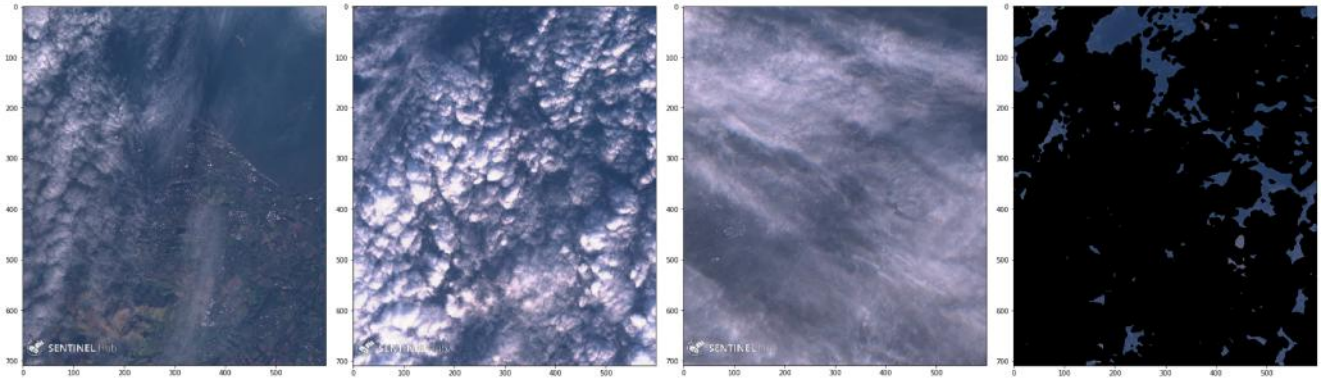

Figure 4: Another example of the composite baseline. This is a failure case where the cloud detection algorithm detects most pixels as covered by clouds in all three input images, even though the clouds are fairly filmy. As a result, the resulting generated "clear" image is composed almost entirely of black pixels. This starkly contrasts with Figure 3 where most pixels are visible in at least one input image. However, in both images substantial areas are not successfully reconstructed.

Figure 5: Examples of each of the ten classes based on the EuroSat dataset [1] which was used for the downstream task of land cover classification. As seen, each classification requires nuanced and explicit details in the images, making this an appropriate downstream task for evaluating the usability of the generated cloud-free images.