Satellite-Image-Inpainting-based-on-EdgeLearning

A deep learning method suitable for inpainting satellite images, which is characterized by first extracting the landform texture information that is more important for satellite images, and then performing the inpainting learning.

1.Theory Introduction

For the inpainting of satellite images, more attention should be paid to the restoration of landform textures such as rivers, hills, and roads. Learning the edge information of landforms first, complementing the edges of the lost parts, and then inpainting the entire satellite image will make the restoration of landform information more accurate and real. The theoretical part of this method is referred from EdgeConnect Generative Image Inpainting with Adversarial Edge Learning which suggested that image inpainting is divided into two parts, the first part is the edge learning network, and the second part is the image inpainting network. For details, please refer to this article, and our method has made practical improvements to the algorithm and training process of the edge learning network, mainly including the following:

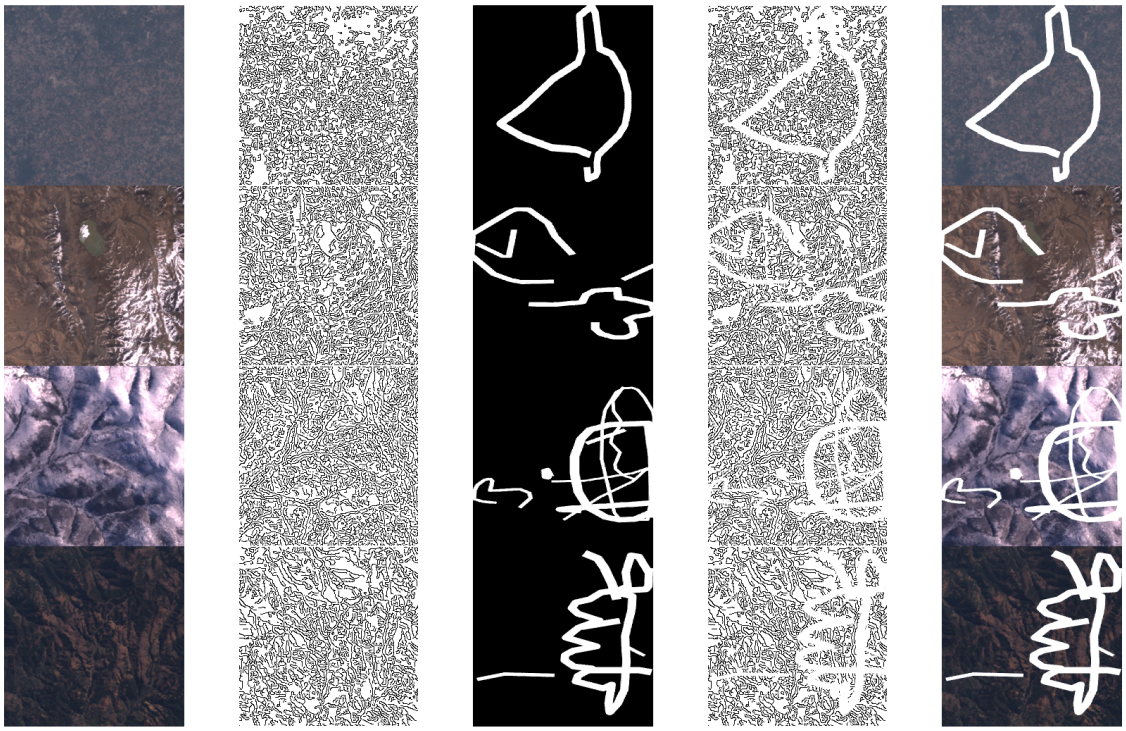
2.Our Improvement in Edge Learning Loss Function

Let $\boldsymbol{edge\_raw}$ be the edge map from the Ground truth Image, this method uses Canny edge detector, and image mask M as a pre-condition (1 for the missing region, 0 for background), then the $\boldsymbol{edge\_miss}$ denotes the corrupted edge map from the Ground truth Image.

$$\boldsymbol{edge\_miss} = 1 - \boldsymbol{edge\_map} \odot (1 - M) \tag{1}$$

Specifically, Let G_1 and D_1 be the generator and discriminator for the edge generator network which referred in EdgeConnect, the generator predicts the edge map for the masked region can be calculated, and different from EdgeConnect, our method remove the grayscale in the G_1.

$$\boldsymbol{edge\_gen} = G_1(M, \boldsymbol{edge\_miss}) \tag{2}$$



(a) (b) (c) (d) (e)

Figure 1:Satellite Images and edges with masks.From left to right: (a)Ground Truth Image. (b)Canny Edge. (c)Mask. (d)Masked Edge. (e)Masked Image. We choose to put (c) and (d) in the edge generator.

And in our method, the edge adversarial loss denoted by $\mathcal{L}_{adv,1}$ is divided into two types $\mathcal{L}_{adv,11}$ and $\mathcal{L}_{adv,12}$, corresponding to two different training stages. In the stages 1, we use L_1 norm in the $\mathcal{L}_{adv,11}$ to reduce the difference between generated edges and real edges.

$$\mathcal{L}_{adv,11} = \mathbb{E}_{\boldsymbol{edge\_raw}}\, logD_1(\boldsymbol{edge\_raw}) + \mathbb{E}_{\boldsymbol{edge\_gen}}\, log[1 - D_1(\boldsymbol{edge\_gen})] \quad (3)$$

Until the edge training $\mathrm{MAPE} < 0.1$, the training process has come to the stage 2, we adopt $\mathcal{L}_{adv,12}$ only focus on reducing the difference between generated and real edges in the missed region. Also, $\mathcal{L}_{adv,12}$ use L_2 norm to magnify the data difference. (In the practical training process, when the epoch reaches 100, it enters the stage 2).

$$\mathcal{L}_{adv,12} = \mathbb{E}_{\boldsymbol{edge\_raw}}\, logD_1(\boldsymbol{edge\_raw} * \boldsymbol{M}) + \mathbb{E}_{\boldsymbol{edge\_gen}}\, log[1 - D_1(\boldsymbol{edge\_gen} * \boldsymbol{M})] \quad (4)$$

Referring to EdgeConnect, the feature-matching loss $\mathcal{L}_{FM}$ compares the activation maps in the intermediate layers of the discriminator. Similarly, we use L_1 norm and L_2 norm in the two stages respectively.

$$\mathcal{L}_{FM,j} = \mathbb{E}\left[\sum_{i=1}^{L} \frac{1}{N_i} \| D_1^{(i)}(\boldsymbol{edge\_raw}) - D_1^{(i)}(\boldsymbol{edge\_gen}) \|_j\right], \;\; j = 1, 2 \quad (5)$$

Where L is the final convolution layer of the discriminator, N_i is the number of elements in the i 'th activation layer, and $D_1^{(i)}$ is the activation in the i'th layer of the discriminator. Further, to ensure the proper generated edge, our method proposes edge structural loss in order to better generate the edge of the missing region.

$$\mathcal{L}_{str} = SSIM(\boldsymbol{edge\_raw} * \boldsymbol{M},\ \boldsymbol{edge\_gen} * \boldsymbol{M}) \quad (6)$$

where $SSIM$ is the structural similarity index between two image tensors. The training objective of the network is divided into two stages, each of which consists of adversarial loss, feature matching loss and structural loss.

$$\min_{G_1} \max_{D_1} \mathcal{L}_{G_1} = \min_{G_1} \left(\lambda_{adv,1j} \cdot \max_{D_1} (\mathcal{L}_{adv,1j}) + \lambda_{FM,j} \cdot \mathcal{L}_{FM,j} + \lambda_{str} \cdot \mathcal{L}_{str} \right), \quad j = 1, 2 \quad (7)$$

3.Dataset

We use the Landsat 7 satellite images are downloaded using the Google Earth Engine (GEE) and focus solely on the visible RGB bands of the Landsat 7 satellite (B3, B2,B1) with spatial resolutions of 30 meters. 60,000 images of various landforms were downloaded, each with a resolution of 256 × 256, corresponding to a land area of approximately 59 km^2. The masks are downloaded from QD-IMD: Quick Draw Irregular Mask Dataset, this dataset is a manually drawn irregular mask that can simulate various image losses very well, For satellite images and mask datasets, 50,000 of these were for the training set and 10,000 for the test set.

4.Result

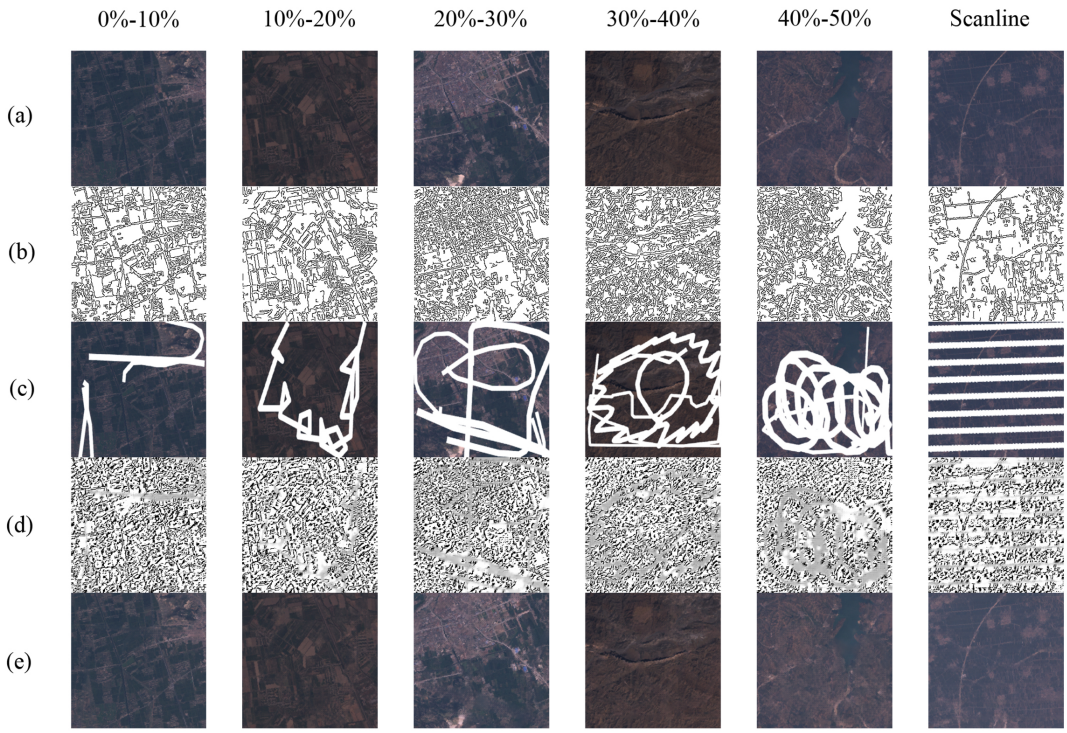


Figure 1: Comparison between different size of mask with our inpaint models. (a) Ground Truth Image. (b) Canny Mask. (c)Ground Truth with Mask. (d) Generate Mask . (e) Our Inpaint Result.

Mask	PSNR¶	SSIM¶	MAPE†	FID†
Scanline	23.91	0.88	0.20	6.53
0%-10%	27.57	0.95	0.14	5.16
10%-20%	28.86	0.94	0.12	5.28
20%-30%	28.60	0.91	0.12	5.70
30%-40%	27.61	0.88	0.13	6.57
40%-50%	25.84	0.85	0.15	7.59

Table 1: The performance of our inpaint model for various mask sizes. (†Low is better; ¶Higher is better)