CS230

Imbalanced Satellite Image Classification using Positive Unlabeled Generative Adversarial Network (Generative Modeling)

Yuxiao Chen
Stanford University
yuxiaoc@stanford.edu

Qingxi Meng
Stanford University
qingxi@stanford.edu

1 Introduction

Many social and geological studies require numerous satellite images to yield accurate result for weather prediction. However, it is expensive to collect images directly from the satellites when the ground is covered by clouds. Hence we will have much more satellites samples without cloud than samples with clouds. The imbalanced dataset causes inaccurate prediction in weather classification task. In this paper, we explore both the generative adversarial neural network and positive unlabelled generative adversarial neural network to generate more cloudy satellite images in order to up-sample the underrepresented class in our dataset. Our method helps balancing the dataset and yields a higher classification accuracy.

2 Related Work

There are several works using GAN to generate new samples for classification task. The work (1) uses GAN to generate new samples to improve the person re-identification problem. The work (2) and (3) use GAN to generate new samples to improve the medical image classification problems. The work (4) use fully-supervised GAN to generate sequence data. However, there has been no work using GAN to generate satellite images to solve data imbalance problem for classification.

For our approach, one of the challenges is to get good data for our task since there is no public benchmark real datasets for images both with and without clouds. Some work (5) (6) about satellite images used artificial clouds because the satellite images with clouds are much rare than normal satellite images. This also justifies our goal of using generative model to generate more satellite images with clouds. Moreover, the sampled images from the generative model may look visually good but does not work well for classification task, which is well known as the adversarial problem (7). Another big challenge is that we plan to use Generative Adversarial Network to generate new samples but GAN is very hard to train in general.

3 Dataset

We use the dataset from this work (8). The dataset contains 97640 single cloudy image and 97640 single clear images. However, in real application, we often have much fewer cloudy satellite images. All cloudy images have both RGB and IR data included, clear images only have RGB data. We pre-process our data by removing all cloudy image's IR band. Then we resize all images to 64x64 to support our generative model dimension.

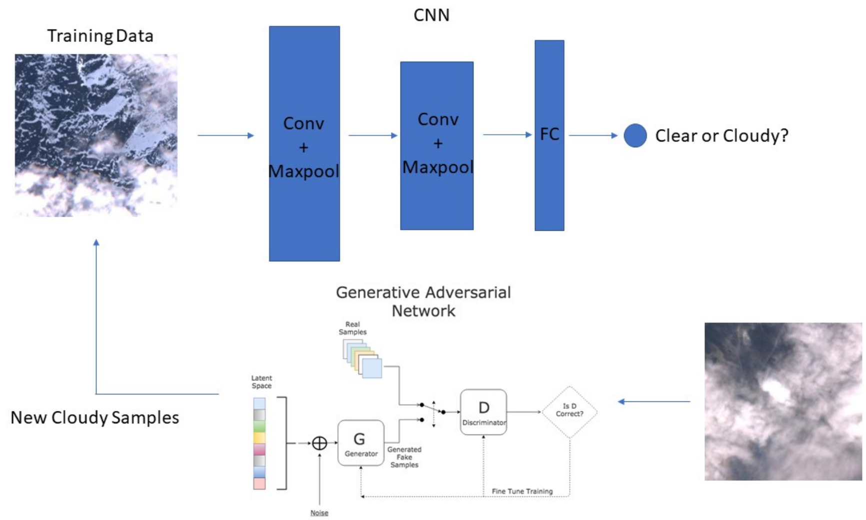

Figure 1: An illustration of the framework for work

4 Methods

For the classification task, we train a CNN model with cross entropy loss. For the generative task, we train both GAN and Positive-Unlabeled GAN to generate 64x64 image with a latent space z. We expect low classification accuracy in the original dataset with imbalanced cloudy images and clear images, and improved classification accuracy on a balanced dataset with generated cloudy images, real cloudy and real clear images.

We use the prediction accuracy to evaluate the classifier. We use the GAN and PU-GAN training loss and human perception to evaluate the generated image quality.

4.1 Classification Task

For the classification task, we train a CNN model with two conv layer and one fully connected layer. The first convolution layer has 6 filters with kernel size 5 and Relu activation. The second convolution layer has 16 filters with kernel size 5 and Relu activation. Each convolution layer is followed by a max pooling layer with kernel size 2 and stride 2. The fully connected layer directly outputs the binary prediction with sigmoid activation.

4.2 Generation task with Preliminary GAN

GAN consists of two neural networks: discriminator network D and generator network G. The discriminator D aims to distinguish between real data and the generated data, while the generator G aims to generate fake data that can fool the discriminator D. Formally, the objective function of GAN can be written as

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}}[logD(x)] + \mathbb{E}_{z\sim p_z}[log(1-D(G(z)))] \quad (1)$$

For the discriminator, we have six discriminator blocks where each block is constructed by three spectrally normalized convolution layer. Then we have a spectrally normalized linear layer with Relu activation for the fully connected layer.
For the generator, the randomly generated latent variable get passed to four generator blocks where each block is constructed by three spectrally normalized convolution layer. Then we have a batch norm layer followed by a convolution layer with ReLu activation to generate the final 64x64 image.

4.3 Generation task with Positive-Unlabeled Classification GAN

One issue we observe from the preliminary GAN result is that the discriminator is unable to distinguish between high-quality generated samples and low-quality generated samples. When the discriminator indiscriminately label all generated samples as false, it makes the generator task converge slower. Especially in our case, the low-quality samples will make it extremely challenging for the classification task because the low-quality satellite images with clouds contains too much white pixels to reserve any information of the original satellite images.

Hence, we propose to use the Positive-Unlabeled Classification GAN (9) to allow the discriminator D to treat the high-quality generated samples as real data. Identifying high-quality samples from generated samples under the guidance of real samples is very similar to Positive-Unlabeled classification problems, where only some positive samples were labeled, and the classifier tried to find positive samples from unlabeled samples consisting of positive and negative samples. Let p_{gr} be the data distribution of generated samples with high-quality. Let p_{gf} be the data distribution of generated samples with high-quality. Let π be the proportion of low-quality sample in all generated samples. Therefore the data distribution of generated samples p_g satisfies

$$p_g(x) = \pi p_{gr}(x) + (1-\pi)p_{gf}(x) \tag{2}$$

Then, the new negative log likelihood objective function of the positive-unlabeled classification will be

$$\max_D V(D) = \pi \mathbb{E}_{x\sim p_{gr}}[logD(x)] + (1-\pi)\mathbb{E}_{x\sim p_{gf}}[log(1-D(x)))] \tag{3}$$

Although the p_{gr} and p_{gf} are unknown, we can estimate p_{gf} with the assumption that optimal p_{gr} is same as p_{data}. Then equation (2) can be rewrite as

$$p_g(x) = \pi p_{data}(x) + (1-\pi)p_{gf}(x) \tag{4}$$

Plug in the estimated p_{gf} in equation (4) into the equation (3) and replace p_{gr} with its optimal p_{data}, we get

$$\max_D V(D) = \pi \mathbb{E}_{x\sim p_{data}}[logD(x)] + \mathbb{E}_{x\sim p_g}[log(1-D(x)))] - \pi \mathbb{E}_{x\sim p_{data}}[log(1-D(x)))] \tag{5}$$

By avoiding the loss to be negative, the final objective function of discriminator becomes

$$\max_D V(D) = \pi \mathbb{E}_{x\sim p_{data}}[logD(x)] + max\{0, \mathbb{E}_{z\sim p_z}[log(1-D(G(z)))] - \pi \mathbb{E}_{x\sim p_{data}}[log(1-D(x)))]\} \tag{6}$$

With the generator and discriminator loss, the training algorithm can be summarized as follow

Algorithm 1 Pseudocode for Positive Classificatoin GAN

while θ_g not converged **do**
    for $t = 1, 2, \ldots, N$ **do**
        Randomly sample $\{x^i\}_{i=1}^m \sim p_{data}(x)$
        Randomly sample $\{z^i\}_{i=1}^m \sim p_z(z)$
        Calculate $V_1 \leftarrow \frac{1}{m}\sum_{i=1}^m [log(D(x^i)]$
        Calculate $V_2 \leftarrow \frac{1}{m}\sum_{i=1}^m [log(1-D(x^i)))]$
        Calculate $V_3 \leftarrow \frac{1}{m}\sum_{i=1}^m [log(1-D(G(z^i)))]$
        Update $\theta_d \leftarrow \nabla_{\theta_d} \pi V_1 + max\{0, V_3 - \pi V_2\}$
    end for
    Randomly sample $\{z^i\}_{i=1}^m \sim p_z(z)$
    Update $\theta_g \leftarrow -\nabla_{\theta_g} \frac{1}{m}\sum_{i=1}^m [log(1-D(G(z^i)))]$
end while

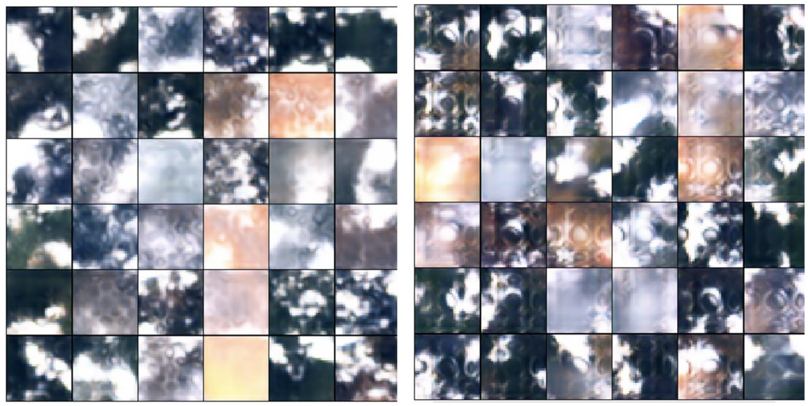

Figure 2: Cloudy samples generated by the preliminary GAN (left) and our PU-GAN (right)

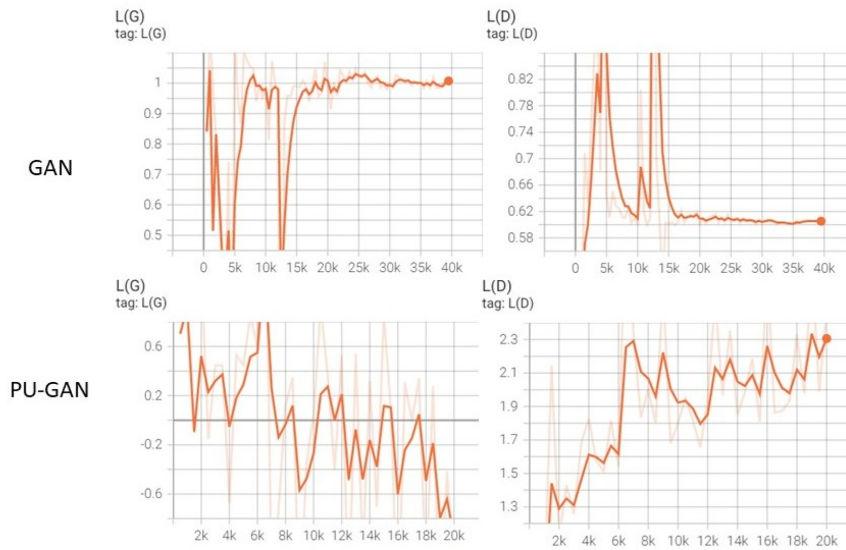

Figure 3: The generator and discriminator training loss for GAN and PU-GAN

5 Experiments/Results/Discussion

We use the 97640 cloudy images as the training set to train both the normal GAN and our PU-GAN. For both the preliminary GAN and PU-GAN model, we use batch size 64 and learning rate $2e^{-4}$. We train the model until it seems to converge (20k steps for PU-GAN and 40k steps for GAN). For the preliminary experiment, we use 0.3 as the estimated proportion of low-quality sample in all generated samples. Figure 2 shows the generated samples from GAN and PU-GAN respectively. We could observe that the samples from PU-GAN have better quality than the samples from normal GAN. Figure 3 shows the generator and discriminator training loss for GAN and PU-GAN respectively. We could observe that the loss of out PU-GAN is more stable than that of the normal GAN. Also, the training for PU-GAN converges faster than GAN.

For the classification task, with all real samples we use 100 cloudy and 1000 clear images for training set, and 1000 cloudy and 1000 clear images for evaluation. With learning rate e^{-3}, batch size 16, and 10 epochs, we reach a classification accuracy around 73% and the loss nearly saturates.

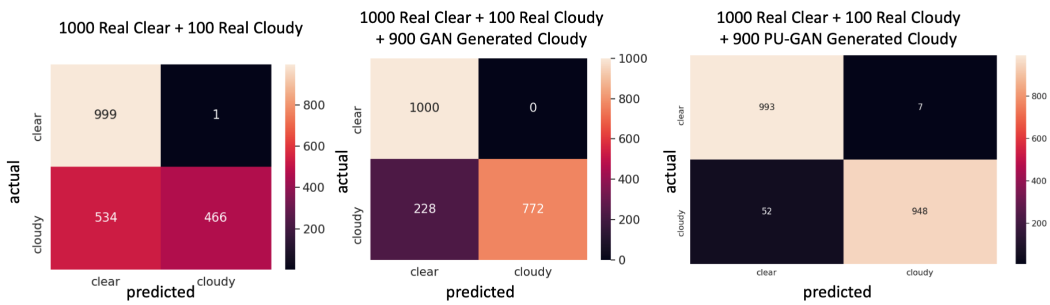

Figure 4: Confusion Matrix for classification task with imbalanced dataset. balanced dataset with GAN, and balanced dataset with PU-GAN

With GAN, we use 100 real cloudy images, 900 generated cloudy images, and 1000 clear images for the training set, and 1000 real cloudy and 1000 real clear images for evaluation. Using the same learning rate, batch size, and epochs, we reach a classification accuracy around 88%, which is around 20% better than the performance without generated samples.

Similarly with PU-GAN, we also use 100 real cloudy images, 900 generated cloudy images, and 1000 clear images for the training set, and 1000 real cloudy and 1000 real clear images for evaluation. Using the same learning rate, batch size, and epochs, we reach a classification accuracy around 97%, which is around 33% better than the performance without generated samples and 10% better than the performance with normal GAN. Therefore, the accuracy of the model with PU-GAN is better than both the model without any generated samples and the model with a normal GAN. The reason may be that the generated new cloudy images with higher quality help the classifier to train, predict and generalize better. It is also very interesting to notice that the accuracy of the model with PU-GAN (around 97%) is very close to the accuracy of the model trained with same amount of real cloudy images (around 98%). This indicates that the generated images by PU-GAN work as nearly the same role as the real data for the classifier.

The confusion matrix for each training set is shown below in the figure 4. As shown in the confusion matrices, the true positive rate remains very high (close to 100%) for all three methods. However, the true negative rate of PU-GAN outperforms both that without any generated samples and that with normal GAN, which indicates that the generated cloudy images by PU-GAN help the model to generalize much better. It is also interesting to notice that the false negative rate of PU-GAN is higher than the one without generated samples and the one with a normal GAN. This indicates that the generated cloudy images from PU-GAN also potentially hurt the performance of the model although the effect is very minor in our dataset.

6 Future Work

For future work, we plan to explore other GAN models, like Wasserstein GAN, BiGAN, and CycleGANs to generate higher quality samples.

We can also use some standard sample quality metrics in addition to human perception, like structural similarity (SIM) and peak signal-to-noise ratio (PSNR) that evaluates local patterns of pixel intensities and smoothness.

7 Contributions

Yuxiao preprocessed the data, and implemented the classification task. Qinxgi implemented GAN and PU-GAN, and generated new sample with these generative models. Both authors worked on the reports, and the final video presentation.

References

[1] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," (Los Alamitos, CA, USA), pp. 3774–3782, IEEE Computer Society, oct 2017.

[2] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification," vol. 321, pp. 321–331, Neurocomputing, 2018.

[3] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using gan for improved liver lesion classification," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 289–293, 2018.

[4] A. Ali-Gombe and E. Elyan, "Mfc-gan: Class-imbalanced dataset classification using multiple fake class generative adversarial network," vol. 361, pp. 212–221, Neurocomputing, 2019.

[5] J. Zheng, X.-Y. Liu, and X. Wang, "Single image cloud removal using u-net and generative adversarial networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 8, pp. 6371–6385, 2021.

[6] Y. Zhao, S. Shen, J. Hu, Y. Li, and J. Pan, "Cloud removal using multimodal gan with adversarial consistency loss," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2021.

[7] F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro, "Intriguing properties of adversarial ml attacks in the problem space," in *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 1332–1349, IEEE, 2020.

[8] V. Sarukkai, A. Jain, B. Uzkent, and S. Ermon, "Cloud removal in satellite images using spatiotemporal generative networks," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1785–1794, 2020.

[9] T. Guo, C. Xu, J. Huang, Y. Wang, B. Shi, and D. Tao, "On positive-unlabeled classification in gan," in *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8385–8393, 02 2020.