

國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

生成式對抗網路用於多時序衛星影像除雲

CTGAN: Cloud Transformer Generative Adversarial
Network

黃繼綸

Gi-Luen Huang

指導教授: 吳沛遠 博士

Advisor: Pei-Yuan Wu Ph.D.

中華民國 112 年 1 月

January, 2023



國立臺灣大學碩士學位論文
口試委員會審定書
MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

(論文中文題目) 生成式對抗網路用於多時序衛星影像除雲

(論文英文題目) CTGAN: Cloud Transformer Generative Adversarial Network

本論文係 黃繼綸 (r09942171) 在國立臺灣大學 電信工程學研究所 完成之碩士學位論文，於民國 111 年 11 月 30 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Institute of Communication Engineering on 30th November 2022 have examined a Master's thesis entitled above presented by Gi-Luen Huang (r09942171) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

吳沛遠

(指導教授 Advisor)

郭榮明

林昌明

系主任/所長 Director:

劉錫學





Acknowledgements

由於大學的專題就與深度學習及電腦視覺相關，因此來到碩班後就馬上接下了太空中心的除霧計畫，而在查閱相關文獻的過程，也發現了現在的碩論題目—衛星影像除雲，很感謝計畫負責人之一的莊學遠學長在我剛進入這領域時給了我很多相關的知識，讓我在很快的時間上手衛星影像的處理。

在碩班的這兩年，很感謝吳沛遠教授給我各方面的訓練，從剛進來實驗室時，一篇論文的報告都還很沒有架構，在經過每週定期開會的訓練後，讓我培養出優秀的研究能力，現在在閱讀一篇論文時可以很快地抓到重點，並以較嚴謹的態度去審慎論文的優缺點。

再來也很感謝實驗室的所有同學，讓我的碩班生活不只有研究，偶爾一起去吃飯、打桌遊，回來再繼續為研究打拼，研究碰到困難時，也會跟我一起討論該如何改善，讓我在碩一就能完成論文並且投稿。

最後，很感謝我女朋友提供給我很多幫助，尤其在英文的撰寫上，有很多不太清楚的文法都會幫我釐清，在我趕論文的那陣子，也給予我很多鼓勵，也因此能在期間內順利的投稿論文，很大的成份都要感謝妳！





摘要

雲的遮蔽干擾了遙測衛星影像的相關應用，包含環境監測、土地覆蓋分類以及貧窮預測等等。在這篇論文中，我們提出了 Cloud Transformer Generative Adversarial Network (CTGAN) 模型來進行衛星影像除雲，此模型的輸入為三張時序的有雲影像，輸出則生成一張對應的無雲影像。相比於過去的生成模型相關的文獻，我們特別著重在設計特徵擷取器 (feature extractor) 使其能保留影像中無雲區域的權重，並且同時減少有雲區域的權重，接著把取出的特徵經過 conformer，利用 attention 的特性，使其能在時序性的影像中找到最關鍵的特徵表示來進行還原。同時，為了解決在這個領域中少量資料的問題，我們自行從哨兵二號衛星蒐集資料進行資料標註，並且提出了 Sen2_MTC 資料集貢獻於這領域。最後，我們也在不同的衛星中進行廣泛的實驗，包括台灣的福爾摩沙衛星二號 (FormoSat-2) 及歐洲的哨兵二號 (Sentinel-2)，實驗表明我們提出的 CTGAN 不僅在這些資料集中都能達到 state-of-the-art 的結果，也在下游任務土地覆蓋分類中，有著顯著的進步，此論文的程式碼公開於此網址中 <https://github.com/come880412/CTGAN>

關鍵字：多時序衛星影像除雲、生成式對抗網路、哨兵衛星二號、福爾摩沙衛星二號、轉換器





Abstract

Cloud occlusions obstruct some applications of remote sensing imagery, such as environment monitoring, land cover classification, and poverty prediction. In this paper, we propose the Cloud Transformer Generative Adversarial Network (CTGAN), taking three temporal cloudy images as input and generating a corresponding cloud-free image. Unlike previous work using generative networks, we design the feature extractor to maintain the cloudless region's weight while reducing the cloudy region's weight. We then pass the extracted features to a conformer module to find the most critical representations. Meanwhile, to address the lack of datasets, we collected a new dataset named Sen2_MTC from the Sentinel-2 satellite and manually labeled each cloudy and cloud-free image. Finally, we conducted extensive experiments on FS-2, the STGAN dataset, and Sen2_MTC. Our proposed CTGAN demonstrates higher qualitative and quantitative performance than the previous work and achieves state-of-the-art performance on these three datasets. We also perform land-cover classification, which can be viewed as a downstream task after cloud

removal. The improved performance on the land-cover classification demonstrates that our model has a high quality of generating cloud-free images compared to the previous works. The code is available at <https://github.com/come880412/CTGAN>



Keywords: Cloud removal for multi-temporal cloudy images, generative adversarial network, conformer, Sentinel-2 satellite, FormoSat-2 satellite



Contents

	Page
Verification Letter from the Oral Examination Committee	i
Acknowledgements	iii
摘要	v
Abstract	vii
Contents	ix
List of Figures	xi
List of Tables	xiii
Chapter 1 Introduction	1
Chapter 2 Related work	5
Chapter 3 Dataset collection	7
Chapter 4 Proposed method	9
4.1 Generator	9
4.2 Discriminator	11
4.3 Loss function	12
Chapter 5 Experiments	15
5.1 Dataset and Implementation Details	16
5.2 Ablation study	17

5.3	Evaluation on the FS-2 dataset	19
5.4	Evaluation on the Sen2_MTC dataset	20
5.5	Visualization on the Sen2_MTC dataset	21
5.6	Evaluation on the downstream task	23
Chapter 6	Conclusion	25
References		27





List of Figures

1.1	The visibility of the same region on the temporal satellite images.	1
1.2	Examples of cloud-free images in the STGAN_dataset (top row) and the Sen2_MTC dataset (bottom row)	2
1.3	The examples of Sen2_MTC images. The top row demonstrates the RGB image, while the down row demonstrates the near-infrared (NIR) image. (a) ~ (c) are cloudy images from different time sequences, and (d) is the corresponding cloud-free image.	2
4.1	Generator of CTGAN (a) Generative network of CTGAN (b) Feature Extractor.	10
4.2	The modules on the feature extractor. (a) Bottleneck module (b) Cloud detection module (c) Atrous convolution module. Where the parameters on the conv are kernel size, stride, and padding, respectively.	10
4.3	The CTGAN discriminator, where the parameters on the conv are kernel size, stride, and padding, respectively.	11
5.1	Ablation study of # of input cloudy images. The x-axis denotes the # of input cloudy images (n) and the y-axis denotes the SSIM metrics.	18
5.2	Visualized results of the generated images on the Sen2_MTC dataset, where the cloud masks are generated by our CTGAN's feature extractor.	22
5.3	The corresponding NIR bands visualization for the cloudy images of the top three rows in the first column in Fig. 5.2	22





List of Tables

5.1	The ablation study of the effectiveness of each module on our CTGAN. . .	17
5.2	The ablation study of the effectiveness of the number of input images. . .	17
5.3	The performance was evaluated by 4-fold cross-validation on the FS-2 dataset	19
5.4	Comparison of PSNR and SSIM results on the STGAN dataset [24]. . . .	20
5.5	The performance compared with the previous works on the Sen2_MTC dataset.	21
5.6	Evaluated the performance of our model through the downstream task land-cover classification, where the accuracy of the cloud-free image can be viewed as an upper bound while the accuracy of the cloudy image can be viewed as a lower bound.	23





Chapter 1 Introduction

Remote sensing imagery has been used in many geoscience observation fields such as land cover classification [8, 15], environment monitoring [12], change detection [18, 19], forest canopy closure estimation [27], and poverty prediction [4, 16]. However, remote sensing imagery is inevitably affected by many factors, such as cloud occlusions, weather, and climate effects. Thick cloud occlusions will lose much of the information. Therefore, cloud removal is an indispensable preprocessing step before using remote sensing imagery in various applications.

Cloud removal methods comprise single-image methods and multi-temporal methods. Single-image methods input one cloudy image to the network and generate a corresponding cloud-free image. Singh *et al.* [25] applied CycleGAN to remove cloud occlusions from synthetically generated cloudy images. Pan *et al.* [17] proposed a spatial-attention-based model for detecting the cloud's location and generating cloud-free images.



Figure 1.1: The visibility of the same region on the temporal satellite images.

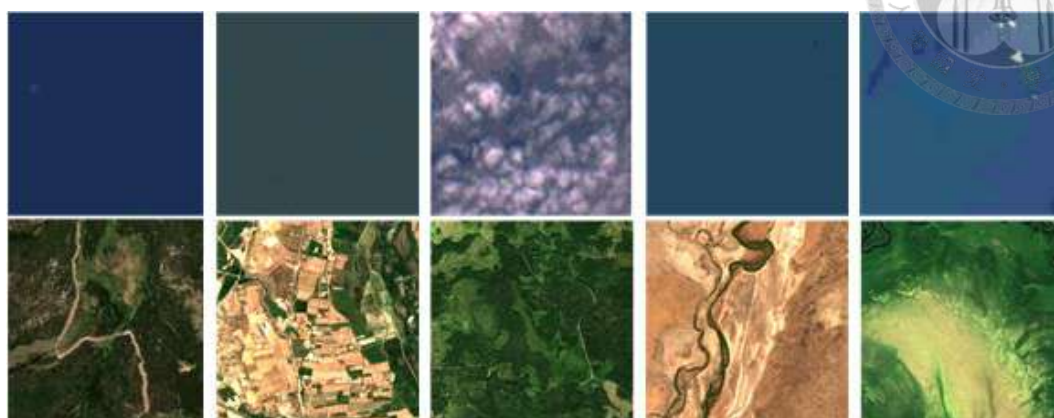


Figure 1.2: Examples of cloud-free images in the STGAN_dataset (top row) and the Sen2_MTC dataset (bottom row)

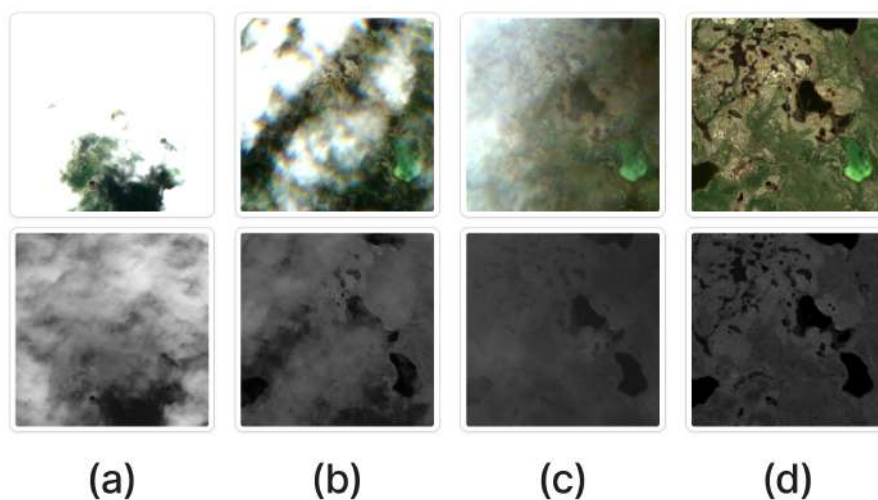


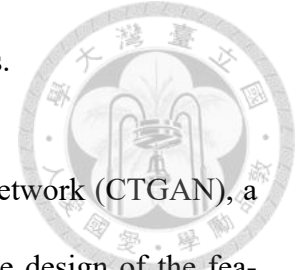
Figure 1.3: The examples of Sen2_MTC images. The top row demonstrates the RGB image, while the down row demonstrates the near-infrared (NIR) image. (a) ~ (c) are cloudy images from different time sequences, and (d) is the corresponding cloud-free image.

Lee *et al.* [11] proposed a CNN-based model to synthesize realistic cloudy images and used the synthesized images to train the network for cloud removal. However, thick cloud occlusions will prevent single-image methods with only a few bands from restoring realistic cloud-free images [14].

To date, there is a great deal of research on single-image methods but comparatively little on multi-temporal methods. Multi-temporal methods can reconstruct thick cloudy images [17, 31]. Lin *et al.* [13] employed an information cloning approach to conduct cloud removal, which clones information from cloud-free regions over temporal images. Sintarasirikulchai *et al.* [26] designed an autoencoder-based model to fuse spectral information across multi-temporal data. Chen *et al.* [3] processed multi-temporal data by integrating feature maps of the spatial and temporal information. Sarukkai *et al.* [24] proposed the spatiotemporal generative network (STGAN) model for multi-temporal end-to-end training. Fig. 1.1 illustrates that the temporal cloudy images may have different visibility in the same region. However, [3, 13, 24, 26] did not make additional processing of the features to differentiate between cloudy and cloud-free regions, which might hinder the model from restoring a realistic cloud-free image.

In addition, to deal with the problem of synthetic data and the lack of real-world temporal data in this field, Sarukkai *et al.* [24] assembled the paired cloudy and cloud-free dataset from the Sentinel-2 satellite. However, Fig. 1.2 illustrates that the images collected by [24] had low resolution and incorrect annotation, causing the model to have high quantitative but low qualitative performance in the early training stage. It also hindered the model from learning to generate a correct cloud-free image. Therefore, we collected another new dataset named Sen2_MTC from the Sentinel-2 satellite for public use, which contains 50 non-overlapping tiles and offers RGB and near-infrared (NIR) channels.

The main contributions of this thesis are summarized as follows.



1. We propose the Cloud Transformer Generative Adversarial Network (CTGAN), a multi-temporal end-to-end training network. We focus on the design of the feature extractor and the processing of the downsampled features. The former uses the cloud mask to force the model to focus on the cloud-free region. The latter uses the attention mechanism in the conformer module to make the model find the most critical representations before restoring the cloud-free image. Meanwhile, our model can simultaneously detect cloud locations and restore the cloud-free image.
2. we collected a new dataset named Sen2_MTC for public use. The images in Sen2_MTC were gathered from the Sentinel-2 satellite, with manually labeled cloudy and cloud-free images. The example images are shown in 1.3.

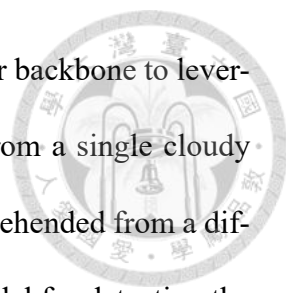


Chapter 2 Related work

Convolutional Neural Network (CNN) has been widely applied to various computer vision tasks. These CNN models are usually pretrained on a large-scale dataset such as ImageNet [22]. With this pretrained weight, CNN models are often chosen as the strong feature extractors, followed by fine-tuning a fully connected network to conduct a large variety of image tasks, including classification, object tracking, and generation.

Generative models have been applied to the cloud removal task using synthetic data. Singh *et al.* [25] leveraged the unpaired image translation model CycleGAN [32] to restore the thin and filmy cloudy image. Enomoto *et al.* [5] added Perlin noise [20] to synthesize the cloudy image from the cloud-free image. Their model Multi-spectral conditional Generative Adversarial Network (MCGAN) is designed to generate the cloud-free image using synthetic and original images for model training. Sandhan *et al.* [23] designed a generative model to train on synthetic data, mainly applied on the extremely filmy high-altitude cloud removal. Bermudez [1] *et al.* applied the conditional generative adversarial networks to generate the cloud-free image using the synthetic aperture radar data for model training. However, the synthetic data usually fails to generate a realistic cloud-free image in the real-world data [24].

Previous works have applied the generative model to the cloud removal task using



real-world data. Meraner *et al.* [14] applied the ResNet [7] as encoder backbone to leverage the multi-spectral information to generate a cloud-free image from a single cloudy input image. However, the multi-spectral information can not be apprehended from a different satellite. Pan *et al.* [17] proposed a spatial-attention-based model for detecting the cloud's location and generating cloud-free images. However, their model is limited to dealing with thin clouds. Sarukkai *et al.* [24] referred to the ResNet [7] and Unet [21] models to propose a spatial-temporal generative model to generate the cloud-free image. However, they ignored the processing of the cloudy and cloud-free patches in the temporal information.

Overall, the primary key points to generating a realistic cloud-free image are as follows:

1. Train the generative model using real-world data instead of synthetic data [24].
2. Single-image with multi-spectral information is feasible to generate a realistic cloud-free image [14]; otherwise, temporal information is required [24].

Considering that most satellites provide information for four bands, namely RGB and NIR, this thesis focuses on processing temporal information with only RGB and NIR channels to generate a realistic cloud-free image.



Chapter 3 Dataset collection

We collect the Sen2_MTC benchmark dataset from the publicly-available Sentinel-2 satellite images for public use. Sentinel-2 satellite has 32,270 distinct tiles, and each tile has a size of 10,980x10,980 with a resolution of 10m/pixel. The captured images from the Sentinel-2 satellite have multi-spectral information with 13 different bands, and the same region was regularly recorded every 6 days on average. In this thesis, we only take the image from RGB and NIR bands. To obtain the data, we randomly pick 50 distinct tiles from the Sentinel-2 satellite, and each tile has 3 cloudy and 1 cloud-free view. Then, we apply the sliding window method with an overlapping size of 128x128 to obtain 7,225 patches in total from each view, where each image patch has a size of 256x256. We manually select 70 cloudy and cloud-free pairs from the 7,225 patches as training data in each distinct tile. The image we selected as cloudy must be obviously occluded by cloud, while the image we selected as cloud-free must not have a single piece of cloud. Fig. 1.3 illustrates an example of cloudy and cloud-free pair. The Sen2_MTC benchmark dataset is collected by the aforementioned criterion, which has 50 distinct tile locations each with 70 image patches, a total of 3,500 images. For the model training, we randomly split the data into training/validation/testing sets with a ratio of 7:1:2 and kept the images from the same tile together. More elaborately, the training data is composed of 35 non-overlapping tiles, for a total of about 2450 images; the validation data is composed of 5 non-overlapping tiles,

for a total of about 350 images; the testing data is composed of 10 non-overlapping tiles,
for a total of about 700 images.





Chapter 4 Proposed method

Like [24], our CTGAN takes three cloudy images to recover a corresponding cloud-free image. Pairwise cloudy and cloud-free images are required to train CTGAN. We denote the input of temporal cloudy images as $x^{in} = \{x_1, x_2, \dots, x_n\}$ and the corresponding cloud-free image as y , where n denotes the number of cloudy images as input. In this thesis, n is set to 3. Given x^{in} , the model learns how to generate the cloud-free image \hat{y} , which should be similar to the corresponding cloud-free image y .

4.1 Generator

The overall CTGAN generative network is illustrated in Fig. 4.1(a). Our generator is based on STGAN [24]. However, unlike STGAN, we focus more on the design of the feature extractor and the processing of the downsampled multi-temporal features. The feature extractor structure is illustrated in Fig. 4.1(b). The bottleneck module, as shown in Fig. 4.2 (a), consisting of three convolutional layers followed by batch normalization and a rectified linear unit after each convolutional layer, extracts the feature representation of the image. This representation proceeds through the cloud detection module, as shown in Fig. 4.2 (b), consisting of three convolutional layers, passing through a sigmoid function before output. The cloud detection module detects the location of the cloud, and

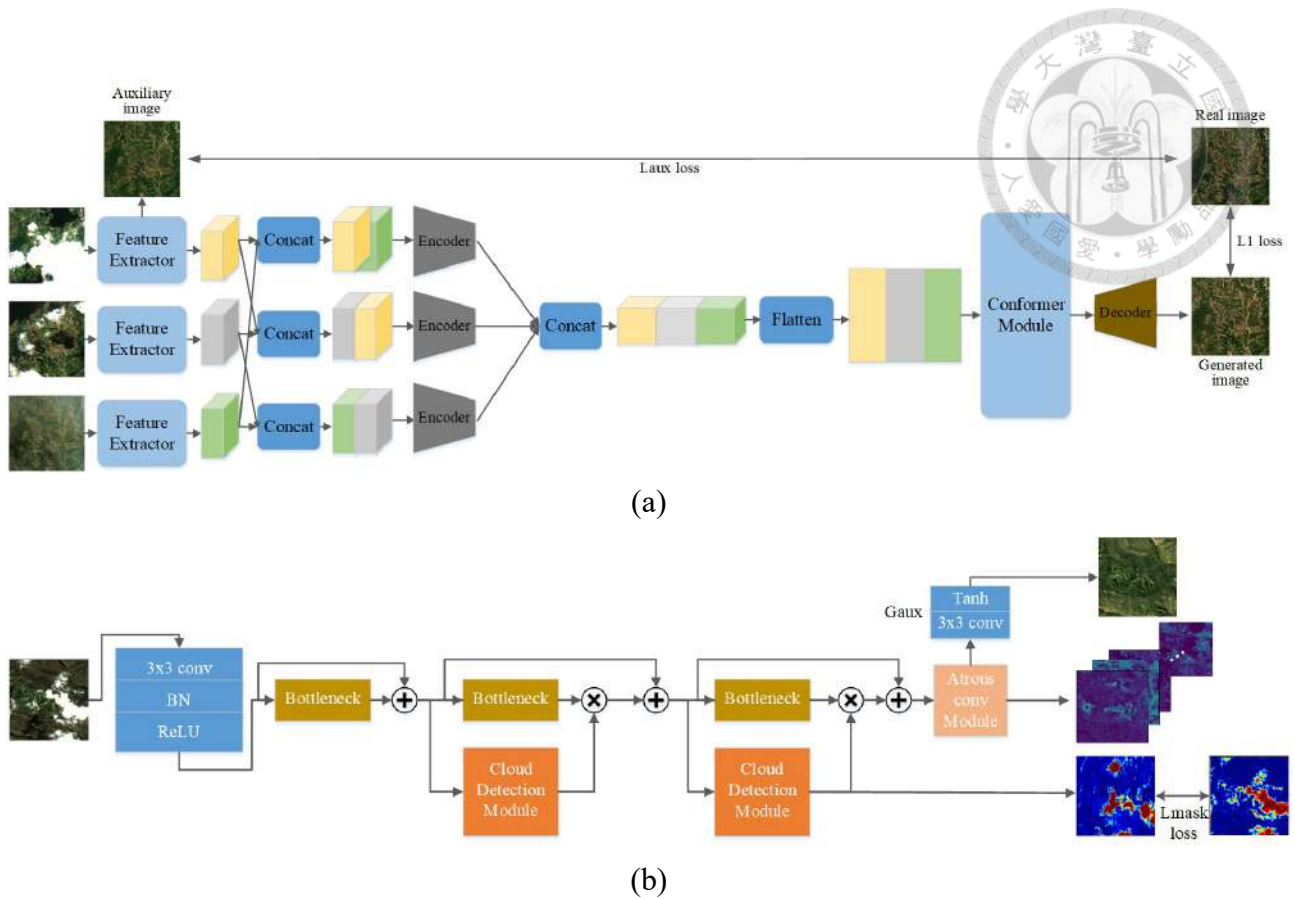


Figure 4.1: Generator of CTGAN (a) Generative network of CTGAN (b) Feature Extractor.

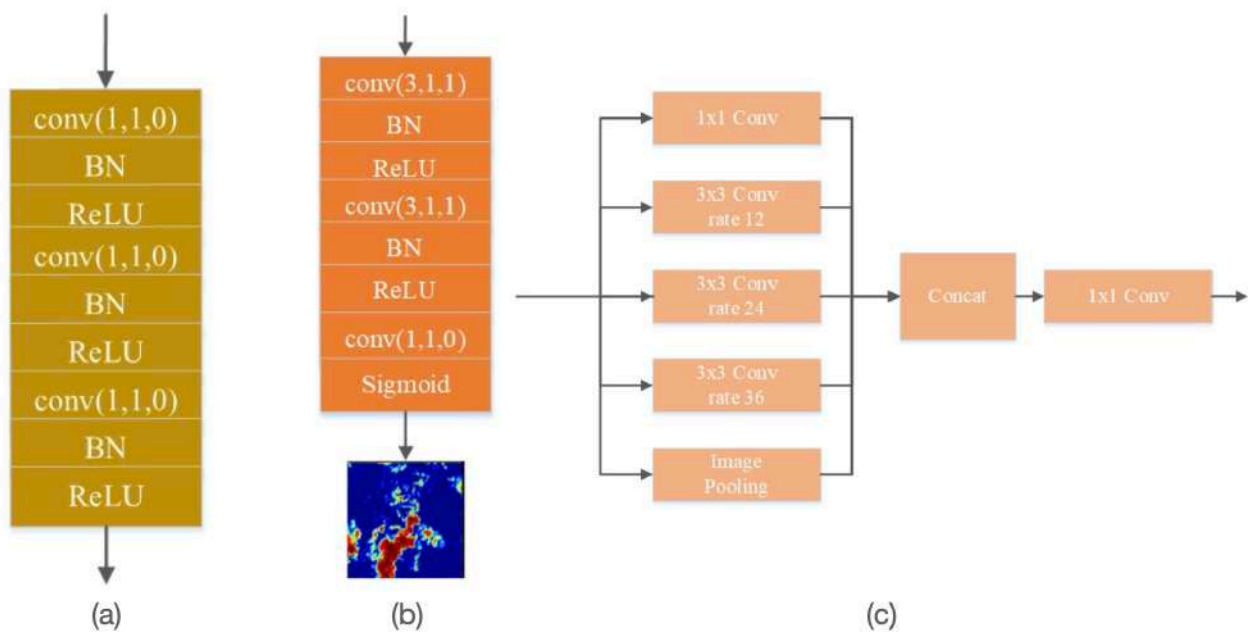


Figure 4.2: The modules on the feature extractor. (a) Bottleneck module (b) Cloud detection module (c) Atrous convolution module. Where the parameters on the conv are kernel size, stride, and padding, respectively.

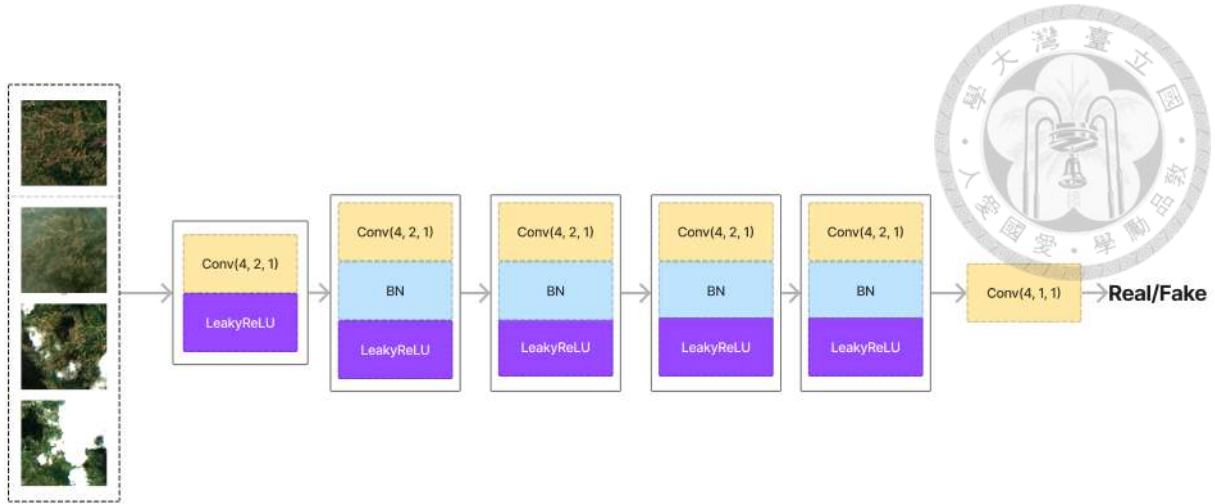


Figure 4.3: The CTGAN discriminator, where the parameters on the conv are kernel size, stride, and padding, respectively.

the generated cloud mask is multiplied by the feature map to keep the weight of the cloud-free region while reducing the weight of the cloudy region. In the previous layer of the feature extractor, we introduce the atrous convolution module [2], as shown in Fig. 4.2 (c), to enlarge the receptive field in the feature extractor. Moreover, inspired by [28], we include an auxiliary generator in the feature extractor to accelerate its convergence. In addition to the design of the feature extractor, we introduce the conformer module [6], which is the modified version of the original Transformer [29], to make the downsampled multi-temporal features find the most critical representations. Finally, the encoder and the decoder are convolutional layers with stride 2 to downsample the feature maps and the transposed convolutional layers with stride 2 to upsample the feature maps, respectively.

4.2 Discriminator

The CTGAN discriminator is a deep convolutional neural network, as demonstrated in Fig. 4.3. We utilize the conditional generative adversarial network (GAN). The network's input is the concatenation of the three cloudy images and the one generated or

cloud-free image. In the prediction phase, the network carries out a binary classification to determine whether the concatenated image matches a generated or cloud-free image.



4.3 Loss function

In this work, the loss function can be defined as:

$$L = \min_G \max_D L_{cGAN}(G, D) + \lambda_G L_1(G) + L_{mask} + \lambda_{aux} L_{aux}, \quad (4.1)$$

where the parameters G and D represent the CTGAN generator and discriminator, and λ_G and λ_{aux} are the reconstruction quality weights of the loss, which are set to 100 and 50 in our model, respectively. The setting of λ_G refers to STGAN [24]. Since the output of the generator is the cloud-free satellite image we would like to obtain, we set λ_G to be higher than the weight of other terms; since the purpose of L_{aux} is to speed up the convergence of the feature extractor, while the quality of the restored auxiliary image is not essential, we set λ_{aux} to 50 which is lower than λ_G . As the other two terms (L_{cGAN} and L_{mask}) are not the main consideration of this work, we set their weights as 1.

The loss function comprises four parts, where the first part is the loss function of the conditional GAN. We define the loss function of L_{cGAN} as:

$$L_{cGAN}(G, D) = E_{(x^{in}, y)} [\log D(x^{in}, y)] + E_{(x^{in})} [\log (1 - D(x^{in}, \hat{y}))], \quad (4.2)$$

where $x^{in} = \{x_1, x_2, \dots, x_n\}$, y is the corresponding ground-truth cloud-free image, and \hat{y} is the corresponding generated cloud-free image. In this thesis, n is set to 3. The second part is the standard L_1 loss function, defined as:

$$L_1(G) = \frac{1}{CWH} \sum_{c,w,h} \|y^{c,w,h} - \hat{y}^{c,w,h}\|_1, \quad (4.3)$$

where $\hat{y}^{c,w,h}$ denotes the pixels of the generated output image at coordinates (c, w, h) . The third

part is the cloud mask loss, defined as :

$$L_{mask} = \|M - M'\|_2^2, \quad (4.4)$$

where M and M' denote the ground-truth cloud mask and the predicted cloud mask, respectively.

The fourth part is the auxiliary loss, defined as:

$$L_{aux} = \frac{1}{CWH} \sum_{i=1}^n \sum_{c,w,h} \|y^{c,w,h} - G_{aux}(FE(x_i))^{c,w,h}\|_1, \quad (4.5)$$

where G_{aux} denotes the auxiliary generator in the feature extractor, and $FE(x_i)$ denotes the output of the feature extractor when feeding x_i into the network.







Chapter 5 Experiments

In this section, we employed our CTGAN on three different datasets, including FormoSat-2 dataset, STGAN dataset, and Sen2_MTC dataset. The metrics used to evaluate the performance are root-mean-square error (RMSE), peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [30], and spectral angle mapper (SAM) [10]. The definition of each metric is given below:

- RMSE

$$RMSE(x, y) = \sqrt{\frac{1}{CWH} \sum_{c,w,h} (y^{c,w,h} - x^{c,w,h})^2}, \quad (5.1)$$

where x and y are the generated cloudy image and the ground-truth cloud-free image.

- PSNR

$$PSNR(x, y) = 20 \log_{10} \left(\frac{MAX_I}{RMSE(x, y)} \right), \quad (5.2)$$

where MAX_I represents the maximal possible value of x , which is set to 255 in this thesis.

- SSIM

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + (c_1L)^2)(2\sigma_{xy} + (c_2L)^2)}{(\mu_x^2 + \mu_y^2 + (c_1L)^2)(\sigma_x^2 + \sigma_y^2 + (c_2L)^2)}, \quad (5.3)$$

where μ_x, μ_y denote the means of x and y , σ_x^2, σ_y^2 denote the variances of x and y , σ_{xy} denotes the covariance of x and y , and c_1, c_2 are adjustable constants. L represents the value of $MAX_I - MIN_I$, which is set to 255 in this thesis.

- SAM

$$SAM(x, y) = \cos^{-1} \left(\frac{\sum_{c,w,h} (y^{c,w,h} \cdot x^{c,w,h})}{\sqrt{\sum_{c,w,h} (x^{c,w,h})^2 \cdot \sum_{c,w,h} (y^{c,w,h})^2}} \right) \quad (5.4)$$

5.1 Dataset and Implementation Details



FormoSat-2 (FS-2) dataset is a decommissioned earth observation satellite formerly operated by the National Space Organization of Taiwan. This dataset contains 15 non-overlapping tiles, each with three cloudy images and a corresponding cloud-free image, $C = 4$ channels (R, G, B, NIR), and pixel value range $[0, 10000]$. In addition, Due to the lack of data in this dataset, we conducted 4-fold cross-validation to evaluate the performance and ensure the robustness of our model compared to previous works [3, 24, 26].

STGAN dataset [24] contains 945 distinct tiles, a total of 3101 images. This dataset was created using the publicly available Sentinel-2 images. [24] paired each cloud-free image with the three most recent cloudy images, each with size $(w, h) = (256, 256)$, $C = 4$ channels (R, G, B, NIR), and pixel value range $[0, 255]$. In addition, [24] randomly split the data into training/validation/testing sets with the ratio of 8:1:1 and kept the images from the same tile together.

Sen2_MTC dataset was collected by us using publicly available Sentinel-2 images to annotate a new cloud removal dataset for multi-temporal training. This dataset contains 50 non-overlapping tiles, each with 70 images, pixel value range $[0, 10000]$, size $(w, h) = (256, 256)$, and $C = 4$ channels (R, G, B, NIR). We randomly split the data into training/validation/testing sets with a ratio of 7:1:2 and kept the images from the same tile together.

Implementation details. Our proposed CTGAN was implemented via Pytorch and run on a server with two NVIDIA GeForce RTX 3090 GPUs, each with 24 GB of graphics memory. We first divided the pixel value by 10,000 and normalized the pixel value range to $[-1, 1]$. We initially trained our model on the FS-2 dataset in the training phase because we had the ground-truth cloud mask for the FS-2 dataset images. Next, we applied the semi-supervised learning technique to the Sentinel-2 dataset, using the feature extractor trained on the FS-2 dataset to generate the pseudo cloud mask on the Sentinel-2 dataset. In addition, we adopted the Adam optimizer with a learning rate of 5×10^{-4} and exponential decay rates $(\beta_1, \beta_2) = (0.5, 0.999)$. We also used the CosineAnnealing scheduler to decay the learning rate per epoch and stopped training after 200 epochs.

	RMSE	PSNR	SSIM	SAM
CTGAN w/o FE and w/o Conformer	0.1167	18.280	0.614	0.272
CTGAN w/o FE and w/ Conformer	0.1125	18.507	0.624	0.265
CTGAN w/ FE and w/o Conformer	0.1098	19.033	0.650	0.252
CTGAN w/ FE and w/ Conformer	0.1082	19.375	0.666	0.245

Table 5.1: The ablation study of the effectiveness of each module on our CTGAN.

input images	RMSE	PSNR	SSIM	SAM	# of parameters	Inference time (s)
n = 1	0.2652	11.536	0.109	0.580	4.99M	7.848
n = 2	0.1323	17.878	0.584	0.294	42.97M	20.266
n = 3	0.1082	19.375	0.666	0.245	43.12M	29.129
n = 4	0.1072	19.485	0.666	0.248	43.27M	35.304
n = 5	0.1070	19.496	0.672	0.247	43.42M	43.636

Table 5.2: The ablation study of the effectiveness of the number of input images.

5.2 Ablation study

In this section, we performed ablation studies to evaluate the contribution of each component of CTGAN and the effectiveness of the number of input cloudy images. In Table 5.1, we evaluated the contribution of each module in our CTGAN. In this thesis, we focus more on the feature extractor’s design. As seen in Table 5.1, the improvement rate of the CTGAN with and without feature extractor design is the highest. It means that the cloud mask attention mechanism in the temporal information has the most impact on performance improvement. Finally, to make the model find the most critical representation in the downsampled feature map, we add the conformer module [6], which further improves the performance of CTGAN.

We also conducted experiments for the effectiveness of the number of input cloudy images. In this thesis, we take three cloudy images as input and then generate the corresponding cloud-free image. However, The number of input cloudy images is adjustable. Table 5.2 shows the performance, # of model parameters, and inference time when increasing the input cloudy images gradually. As shown in Table 5.2, the model performs better as the number of input cloudy images grows. The trend of the performance gain is demonstrated in Fig. 5.1. The model cannot remove the cloud effectively when n=1, while the SSIM performance is significantly improved when n is increased from 1 to 2. The SSIM performance is slightly improved when n is increased from 2 to 3. However, from n=3 to n=5, the SSIM performance is almost the same. Therefore, in this

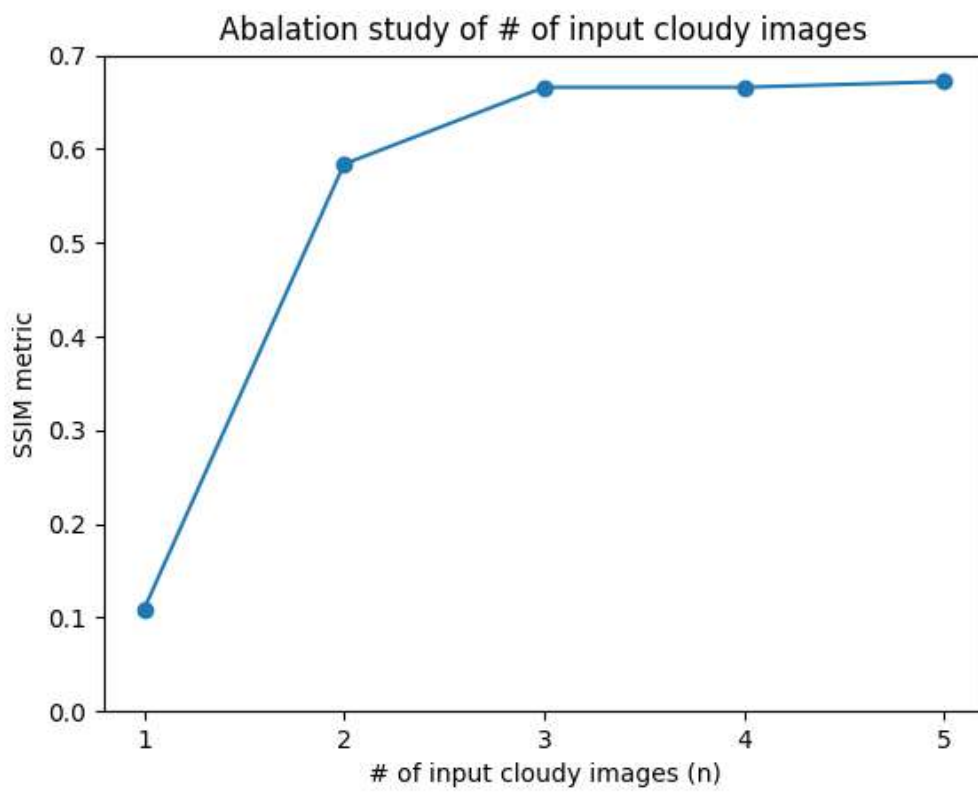


Figure 5.1: Ablation study of # of input cloudy images. The x-axis denotes the # of input cloudy images (n) and the y-axis denotes the SSIM metrics.



RMSE	Fold1	Fold2	Fold3	Fold4	Avg
AE [26]	0.1449	0.1433	0.1422	0.1695	0.1500
ST_net [3]	0.1229	0.1230	0.1299	0.1491	0.1312
STGAN [24]	0.1167	0.1370	0.1242	0.1350	0.1282
CTGAN(ours)	0.1082	0.1023	0.0941	0.1225	0.1068 ± 0.0104
PSNR	Fold1	Fold2	Fold3	Fold4	Avg
AE [26]	16.851	17.038	17.419	15.567	16.719
ST_net [3]	18.267	18.205	18.145	16.572	17.797
STGAN [24]	18.275	17.315	18.277	17.403	17.818
CTGAN(ours)	19.375	19.806	20.585	18.263	19.507 ± 0.965
SSIM	Fold1	Fold2	Fold3	Fold4	Avg
AE [26]	0.577	0.589	0.603	0.541	0.578
ST_net [3]	0.620	0.611	0.564	0.598	0.598
STGAN [24]	0.614	0.604	0.589	0.614	0.605
CTGAN(ours)	0.666	0.662	0.689	0.657	0.669 ± 0.012
SAM	Fold1	Fold2	Fold3	Fold4	Avg
AE [26]	0.324	0.334	0.323	0.372	0.338
ST_net [3]	0.277	0.274	0.310	0.315	0.294
STGAN [24]	0.272	0.279	0.294	0.300	0.286
CTGAN(ours)	0.245	0.240	0.243	0.271	0.250 ± 0.012


Table 5.3: The performance was evaluated by 4-fold cross-validation on the FS-2 dataset

experiment, $n=2$ or $n=3$ is a good trade-off between the SSIM performance and the inference time.

5.3 Evaluation on the FS-2 dataset

We reproduced [3, 24, 26] on the FS-2 dataset to compare the performance between our model and the previous works. When the authors provided the source code [24], we used the provided code to reproduce their model on the FS-2 dataset. Otherwise, we programmed it by ourselves from the description in their paper [3, 26]. We only evaluated their models' performance on our datasets because they did not release their datasets. The results are compared in Table 5.3, where the values in parentheses represent the standard deviation (STD) of our model. On the FS-2 dataset, the results shown in Table 5.3 demonstrate that the design of our model is effective. The improvement of SSIM significantly outperformed the previous state-of-the-art model STGAN, with a breakthrough gain of SSIM 0.064.

We evaluated our CTGAN on the dataset collected by [24]. The performance is shown in



Validation set	PSNR	SSIM
Pix2Pix [9]	23.130	0.442
MCGAN (RGB + NIR) [5]	21.352	0.485
Mean Filter	16.962	0.174
Median Filter	9.081	0.357
Raw Cloudy Images	7.926	0.389
STGAN U-Net (IR) [24]	25.142	0.651
STGAN ResNet(IR) [24]	25.628	0.724
CTGAN(ours)	26.149 ± 0.438	0.805 ± 0.017
Testing set	PSNR	SSIM
Pix2Pix [9]	22.894	0.437
MCGAN (RGB + NIR) [5]	21.146	0.481
Mean Filter	16.893	0.173
Median Filter	9.674	0.395
Raw Cloudy Images	8.289	0.422
STGAN U-Net (IR) [24]	25.388	0.661
STGAN ResNet(IR) [24]	26.186	0.734
CTGAN(ours)	26.264 ± 0.204	0.808 ± 0.011

Table 5.4: Comparison of PSNR and SSIM results on the STGAN dataset [24].

Table 5.4. [24] did not describe their random seed to split the dataset in their paper, so we trained our model 10 times using the same data-splitting method with different random seeds and averaged these results to obtain the final result. The SSIM improvement of our CTGAN is considerable. The SSIM on the validation and testing sets of the previous state-of-the-art STGAN are 0.724 and 0.734, respectively. Our CTGAN significantly outperformed the previous state-of-the-art STGAN; the gain of SSIM on the validation and testing sets are 0.081 and 0.074, respectively. The experimental result on the STGAN dataset also demonstrates that our CTGAN can achieve state-of-the-art performance on the STGAN dataset.

5.4 Evaluation on the Sen2_MTC dataset

The method of reproducing STGAN [24], ST_net [3], and AE [26] is the same as described in section 5.3. We also evaluated CTGAN on the Sen2_MTC dataset. The results shown in Table 5.5 demonstrate that CTGAN achieves higher quantitative performance than [3, 24, 26]. The SSIM on the validation and testing sets of the previous state-of-the-art STGAN are 0.613 and 0.587, respectively. Our CTGAN outperformed the previous state-of-the-art STGAN; the gain of SSIM

Validation set	RMSE	PSNR	SSIM	SAM
AE [26]	0.1728	16.010	0.431	0.444
ST_net [3]	0.1386	17.741	0.467	0.320
STGAN [24]	0.1040	20.612	0.613	0.276
CTGAN(ours)	0.0953 ± 0.0042	21.259 ± 0.046	0.662 ± 0.003	0.241 ± 0.011
Testing set	RMSE	PSNR	SSIM	SAM
AE [26]	0.2088	15.251	0.412	0.420
ST_net [3]	0.1640	16.206	0.427	0.350
STGAN [24]	0.1374	18.152	0.587	0.289
CTGAN(ours)	0.1353 ± 0.0012	18.308 ± 0.089	0.609 ± 0.007	0.262 ± 0.009

Table 5.5: The performance compared with the previous works on the Sen2_MTC dataset.

on the validation and testing sets are 0.049 and 0.022, respectively. The improvement for SSIM is usually shown that the model is more capable of restoring the visible structure. Therefore, CTGAN can reconstruct more information in the generated cloud-free image than the previous state-of-the-art model STGAN, as demonstrated in Fig. 5.2.

5.5 Visualization on the Sen2_MTC dataset

We visualize the generated results using CTGAN and the previous state-of-the-art STGAN [24]. The results are shown in Fig. 5.2. From top to bottom are the three cloudy input images, three cloud masks generated by CTGAN, the cloud-free image generated by STGAN, the cloud-free image generated by CTGAN, and the corresponding ground-truth cloud-free image. Fig. 5.2 shows that our CTGAN can restore a cloudy image more like the actual image, while the image generated by STGAN has many artifacts. Even if the details of the three images are nearly lost, our CTGAN can still roughly restore the shapes under the clouds (first column of Fig. 5.2). To verify why our CTGAN can do such generation, Fig. 5.3 illustrates the NIR bands for the cloudy images of the first column in Fig. 5.2. From Fig. 5.3, the road under the cloud can be seen in the NIR bands. Note that CTGAN can roughly restore the shapes under the cloud, while STGAN [24] generates vague cloud-free images despite also having the NIR information available.

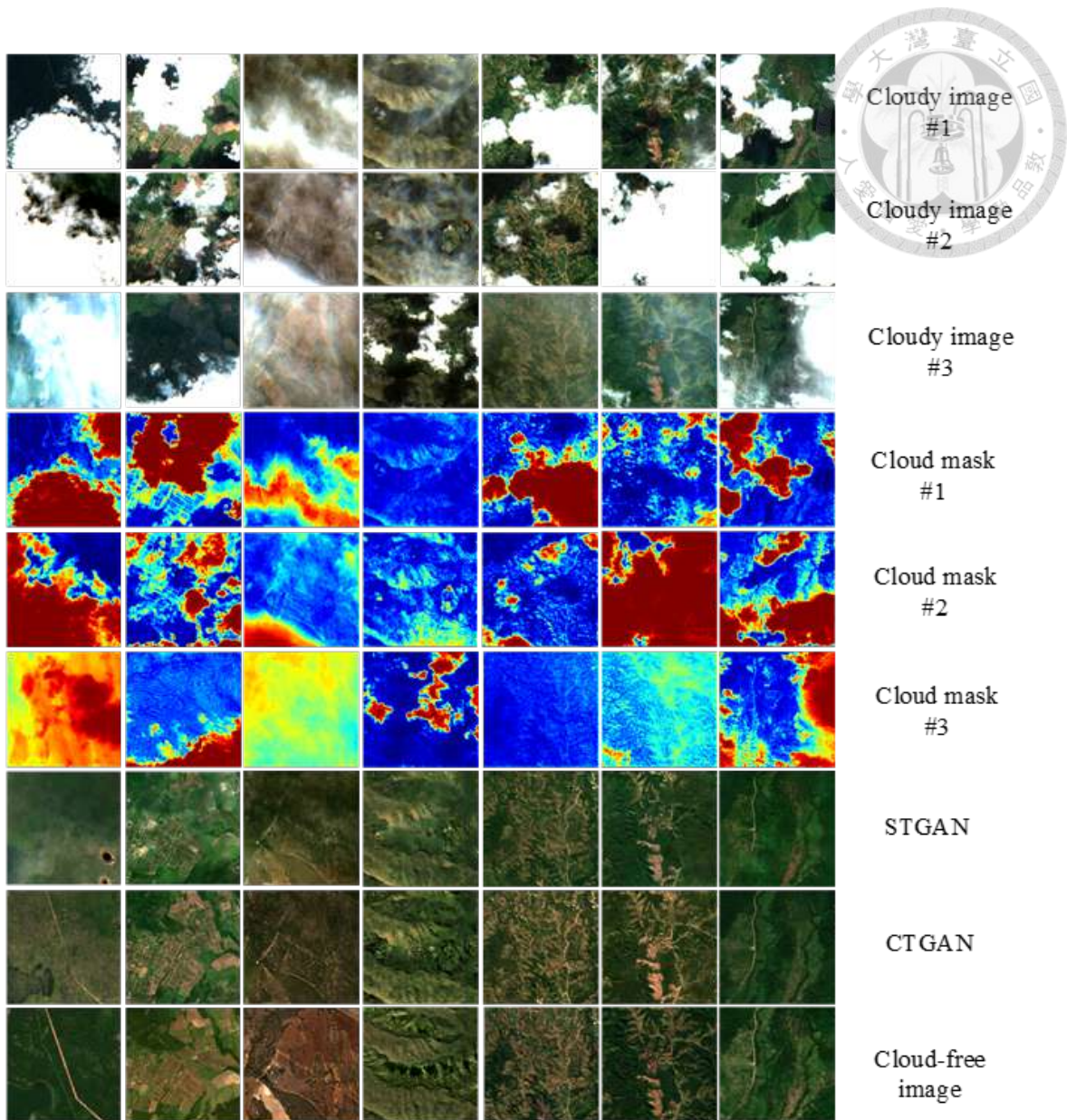


Figure 5.2: Visualized results of the generated images on the Sen2_MTC dataset, where the cloud masks are generated by our CTGAN's feature extractor.



Figure 5.3: The corresponding NIR bands visualization for the cloudy images of the top three rows in the first column in Fig. 5.2



Model	Accuracy(%)
Cloud-free image	97.65
Cloudy image	39.11
AE [26]	41.08
ST_net [3]	51.35
STGAN [24]	68.38
CTGAN(ours)	77.54*

Table 5.6: Evaluated the performance of our model through the downstream task land-cover classification, where the accuracy of the cloud-free image can be viewed as an upper bound while the accuracy of the cloudy image can be viewed as a lower bound.

5.6 Evaluation on the downstream task

In this section, we evaluated the downstream task land-cover classification performance of images after cloud removal by CTGAN model trained on the Sen2_MTC dataset. We first trained a baseline model ResNet50 on the EuroSat dataset [8]. This dataset contains 27,000 labeled images from the Sentinel-2 satellite and consists of 10 categories of land cover (sea, lake, river, residential, permanent crop, pasture, industrial, highway, herbaceous vegetation, forest, and annual crop). To compare the performance, we hand-labeled the images on the testing set of the Sen2_MTC dataset with an approximately equal distribution of all ten categories of land cover.

As shown in Table 5.6, the cloud-free image generated from CTGAN performs the highest accuracy, an improvement of 9% compared with the previous state-of-the-art model. In addition, we also list the accuracy of the ground-truth cloud-free image and the cloudy image, where the former can be viewed as the upper bound, and the latter can be considered as the lower bound. Our model improves the classification accuracy of land-cover from 39% to 77%.





Chapter 6 Conclusion

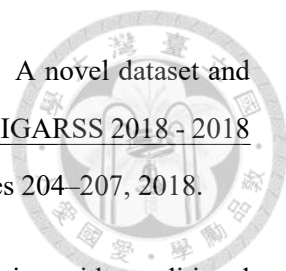
We propose CTGAN for multi-temporal cloud removal. Unlike previous work, We focus more on the feature extractor's design and processing of the downsampled multi-temporal features. In addition, to solve the lack of datasets for multi-temporal cloud removal, we collect a new dataset from Sentinel-2, which we name Sen2_MTC, and manually labeled each cloudy and cloud-free image. Finally, we experimentally demonstrate that CTGAN can achieve high qualitative and quantitative performance and significantly outperform the previous state-of-the-art models. In addition, experiments on the downstream task land-cover classification also verify that the cloud-free images generated by our CTGAN are of high quality enough to perform the classification task. Finally, We will also release the Sen2 MTC dataset for public use.

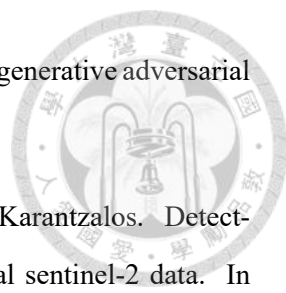




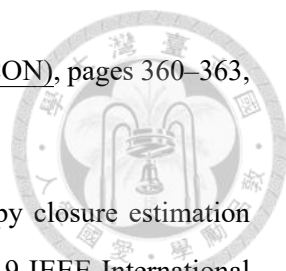
References

- [1] J. D. Bermudez, P. N. Happ, R. Q. Feitosa, and D. A. Oliveira. Synthesis of multispectral optical images from sar/optical multitemporal data using conditional generative adversarial networks. IEEE Geoscience and Remote Sensing Letters, 16(8):1220–1224, 2019.
- [2] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. arXiv, 2017.
- [3] Y. Chen, Q. Weng, L. Tang, X. Zhang, M. Bilal, and Q. Li. Thick clouds removing from multitemporal landsat images using spatiotemporal neural networks. IEEE Transactions on Geoscience and Remote Sensing, 60:1–14, 2020.
- [4] P. S. Das, H. Chhabra, and S. K. Dubey. Socio economic analysis of india with high resolution satellite imagery to predict poverty. In 2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence), pages 310–314, 2020.
- [5] K. Enomoto, K. Sakurada, W. Wang, H. Fukui, M. Matsuoka, R. Nakamura, and N. Kawaguchi. Filmy cloud removal on satellite imagery with multispectral conditional generative adversarial nets. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 48–56, 2017.
- [6] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. arXiv, 2020.
- [7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

- 
- [8] P. Helber, B. Bischke, A. Dengel, and D. Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, pages 204–207, 2018.
- [9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1125–1134, 2017.
- [10] F. A. Kruse, A. Lefkoff, J. Boardman, K. Heidebrecht, A. Shapiro, P. Barloon, and A. Goetz. The spectral image processing system (sips)—interactive visualization and analysis of imaging spectrometer data. Remote sensing of environment, 44(2-3):145–163, 1993.
- [11] K.-Y. Lee and J.-Y. Sim. Cloud removal of satellite images using convolutional neural network with reliable cloudy image synthesis model. In 2019 IEEE International Conference on Image Processing (ICIP), pages 3581–3585, 2019.
- [12] A. Li, J. Bian, G. Lei, X. Nan, and Z. Zhang. Remote sensing monitoring and integrated assessment for the eco-environment along china-pakistan economic corridor. In IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, pages 6421–6424. IEEE, 2019.
- [13] C.-H. Lin, P.-H. Tsai, K.-H. Lai, and J.-Y. Chen. Cloud removal from multitemporal satellite images using information cloning. IEEE transactions on geoscience and remote sensing, 51(1):232–241, 2012.
- [14] A. Meraner, P. Ebel, X. X. Zhu, and M. Schmitt. Cloud removal in sentinel-2 imagery using a deep residual neural network and sar-optical data fusion. ISPRS Journal of Photogrammetry and Remote Sensing, 166:333–346, 2020.
- [15] R. Minetto, M. P. Segundo, and S. Sarkar. Hydra: An ensemble of convolutional neural networks for geospatial land classification. IEEE Transactions on Geoscience and Remote Sensing, 57(9):6530–6541, 2019.
- [16] A. Okaidat, S. Melhem, H. Alenezi, and R. Duwairi. Using convolutional neural networks on satellite images to predict poverty. In 2021 12th International Conference on Information and Communication Systems (ICICS), pages 164–170. IEEE, 2021.

- 
- [17] H. Pan. Cloud removal for remote sensing imagery via spatial attention generative adversarial network. arXiv, 2020.
- [18] M. Papadomanolaki, S. Verma, M. Vakalopoulou, S. Gupta, and K. Karantzas. Detecting urban changes with recurrent neural networks from multitemporal sentinel-2 data. In IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, pages 214–217. IEEE, 2019.
- [19] D. Peng, Y. Zhang, and H. Guan. End-to-end change detection for high resolution satellite images using improved unet++. Remote Sensing, 11(11):1382, 2019.
- [20] K. Perlin. Improving noise. In Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pages 681–682, 2002.
- [21] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.
- [22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3):211–252, 2015.
- [23] T. Sandhan and J. Young Choi. Simultaneous detection and removal of high altitude clouds from an image. In Proceedings of the IEEE International Conference on Computer Vision, pages 4779–4788, 2017.
- [24] 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), pages 1785–1794, 2020.
- [25] P. Singh and N. Komodakis. Cloud-gan: Cloud removal for sentinel-2 imagery using a cyclic consistent generative adversarial networks. In IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, pages 1772–1775, 2018.
- [26] W. Sintarasirikulchai, T. Kasetkasem, T. Isshiki, T. Chanwimaluang, and P. Rakwatin. A multi-temporal convolutional autoencoder neural network for cloud removal in remote sensing images. In 2018 15th International Conference on Electrical Engineering/Electronics,

Computer, Telecommunications and Information Technology (ECTI-CON), pages 360–363, 2018.

- 
- [27] S. Sun, Z. Li, X. Tian, Z. Gao, C. Wang, and C. Gu. Forest canopy closure estimation in greater khingan forest based on gf-2 data. In IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, pages 6640–6643, 2019.
- [28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015.
- [29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.
- [30] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, 2004.
- [31] M. Xu, X. Jia, M. Pickering, and A. J. Plaza. Cloud removal based on sparse representation via multitemporal dictionary learning. IEEE Transactions on Geoscience and Remote Sensing, 54(5):2998–3006, 2016.
- [32] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision, pages 2223–2232, 2017.