**Cloud Detection, Removal and Reconstruction of Satellite Image:Cloud-GAN**

A dissertation submitted in the partial fulfilment of the academic requirements for the award of degree of

#### Bachelor of Engineering

In

#### Computer Science and Engineering

By

**Baira Sai Vikas (100520733081)**

**Kuthadi Charan (100520733090)**

**Potharaj Bhasker (100520733096)**

**Rallabandi Naveen Kumar (100520733097)**

Under the guidance of **Mrs. A. Gayathri Assistant Professor Department of CSE**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**UNIVERSITY COLLEGE OF ENGINEERING (AUTONOMOUS)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING UNIVERSITY COLLEGE OF ENGINEERING**

**OSMANIA UNIVERSITY**

**CERTIFICATE**

This is to certify that the Major project work entitled Cloud Detection, Removal and Reconstruction Of Satellite Image submitted by Potharaj Bhasker(100520733096), Baira Sai Vikas(100520733081),Rallabandi Naveen Kumar(100520733097), Kuthadi Charan(100520733090) a student of DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, UNIVERSITY COLLEGE OF ENGINEERING, OSMANIA

UNIVERSITY in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering is a record of the bonafide work carried out by them during the academic year 2023-2024.

#### Signature of the Supervisor Signature of Head of the Dept.

Mrs.A.Gayathri Prof. P.V.Sudha

Dept. of CSE, OU Dept. of CSE, OU

### STUDENT DECLARATION

We declare that the work reported in the project report entitled “Cloud Detection, Removal and Reconstruction Of Satellite Image'' submitted by Potharaj Bhasker, Baira Sai Vikas, Rallabandi Naveen Kumar, Kuthadi Charan is a record of the work done by us in the DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, UNIVERSITY COLLEGE OF

ENGINEERING, OSMANIA UNIVERSITY. No part of the report is copied from books/ journals/internet and wherever referred, the same has been duly acknowledged in the text. The reported data is based on the work done entirely by us and not copied from any other source or submitted to any other Institute or University for the award of a degree or diploma.

**SIGNATURES:**

Potharaj Bhasker

Baira Sai Vikas

Rallabandi Naveen Kumar

Kuthadi Charan

### ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude and whole-hearted thanks to my project guide Mrs.A.Gayathri, Assistant Professor, Department of Computer Science and Engineering, University College of Engineering Osmania University, for giving me the privilege of working under her guidance, with tremendous support and cogent discussion, constructive criticism and encouragement throughout this dissertation work carrying out the project work. Department of Computer Science and Engineering

I also thank Prof. P.V.Sudha, Head of Department of Computer Science and Engineering for her support from the Department and allowing all the resources available to us students.

I also extend my thanks to the entire faculty of the University College of Engineering, Osmania University who encourages us throughout the course of our Bachelor degree and allowing us to use the many resources present in the department.

Our sincere thanks to our parents and friends for their valuable suggestions, morals, strength and support for the completion of our project.

**POTHARAJ BHASKER (100520733096) BAIRA SAI VIKAS (100520733081)**

**RALLABANDI NAVEEN KUMAR (100520733097) KUTHADI CHARAN (100520733090)**

## Abstract

The pre-processing of satellite images has a wealth of applications and is now an important area of research, like environmental, radiometric, and geometric correc- tions, as well as noise from sensors. However, the data from satellites may contain excessive amounts of cloud cover, which standard methods like Fmask, Tmask, Sen2cor, and Maja are not able to effectively remove. Fmask has trouble identi- fying cirrus clouds, Tmask overestimates shadows and fails to detect some cirrus, Sen2cor excludes the most clouds while Maja overestimates the presence of clouds. To counter these issues, this study uses a hybrid method, Cloud-GAN, which in- cludes two modules Generator and Discriminator. The generator network learns to generate realistic cloud-free images from cloudy input data. Simultaneously, the discriminator network learns to distinguish between real and generated images, providing feedback to the generator network to improve its performance. Finally, the output of this process is free from the cloud. Our Methodology is efficient such that it can detect and removes the clouds from a vast range of image sets including individual images in contrast to existing algorithms which consider bulk test sets. The Cloud-GAN is efficient with various data sets and effectively outperforms dif- ferent frequently used methods by a significant margin. Cloud-GAN removes the clouds with an accuracy of 81.60%.

**Keywords:** Satellite Image, Pre-processing, GAN, Cloud Detection, Cloud Re- moval.

## Table of Contents

1. INTRODUCTION 1
   1. [Basic Concepts 2](#_TOC_250036)
   2. [Motivation 5](#_TOC_250035)
   3. [Problem Statement 5](#_TOC_250034)
   4. [Scope 5](#_TOC_250033)
   5. [Objectives 5](#_TOC_250032)
   6. [Advantages 5](#_TOC_250031)
2. LITERATURE REVIEW 6
   1. [Cloud removal in remote sensing images using nonnegative matrix factorization and error correction[1] 6](#_TOC_250030)
   2. [Cloud removal in Sentinel-2 imagery using a deep residual neural network and SAR-optical data fusion[2] 7](#_TOC_250029)
   3. Cloud and cloud shadow detection for optical satellite imagery: Fea- tures, algorithms, validation, and prospects[3] 8
   4. A deep-learning reconstruction method for remote sensing images

with large thick cloud cover[4] 9

* 1. Cloud Removal in Remote Sensing Images Using Generative Adver- sarial Networks and SAR-to-Optical Image Translation[5] 10
  2. Thin cloud removal in optical remote sensing images based on gen- erative adversarial networks and physical model of cloud distortion

[6] 11

* 1. Cloud removal for remotely sensed images by similar pixel replace-

ment guided with a spatio-temporal MRF model [7] 12

* 1. Deep learning for multi-modal classification of cloud, shadow and

land cover scenes in PlanetScope and Sentinel-2 imagery[8] 13

1. SOFTWARE REQUIREMENT ANALYSIS 14
   1. [FUNCTIONAL REQUIREMENTS 14](#_TOC_250028)
   2. [NON-FUNCTIONAL REQUIREMENTS 15](#_TOC_250027)
   3. [Software Requirements 15](#_TOC_250026)
   4. [Hardware Requirements 16](#_TOC_250025)
2. SOFTWARE DESIGN 17
   1. [Software Development Life cycle 17](#_TOC_250024)
   2. [UML Diagrams 18](#_TOC_250023)
      1. [Use-Case Diagram 19](#_TOC_250022)
      2. [Activity Diagram 20](#_TOC_250021)
      3. [Sequence Diagram 21](#_TOC_250020)
3. PROPOSED SYSTEM 22
   1. [Process Flow Diagram 22](#_TOC_250019)
   2. [Methodology 23](#_TOC_250018)
      1. [Data Pre-processing 23](#_TOC_250017)
      2. [Training and Testing 24](#_TOC_250016)
      3. [Activation Function 24](#_TOC_250015)
      4. [Output 24](#_TOC_250014)
   3. [Algorithm/Pseudo Code 25](#_TOC_250013)
      1. [Split the Collected data into patches 25](#_TOC_250012)
      2. [Data Preprocessing 25](#_TOC_250011)
      3. [Model Architecture: Cloud-GAN architecture 26](#_TOC_250010)
      4. [Model Training 28](#_TOC_250009)
      5. [Model Evaluation 28](#_TOC_250008)
   4. [Dataset Collection 29](#_TOC_250007)
4. IMPLEMENTATION 31
   1. [Output Screenshots 31](#_TOC_250006)
      1. [Patching 31](#_TOC_250005)
      2. [Pre-processing 32](#_TOC_250004)
      3. [Cloud-free Image Prediction 33](#_TOC_250003)
      4. [Untrained Cloud Image Prediction (Test Image) 34](#_TOC_250002)
   2. [Performance Analysis 35](#_TOC_250001)
5. CONCLUSION AND FUTURE WORK 36

[REFERENCES 37](#_TOC_250000)

**List of Figures**

* 1. UNet Architecture 2
  2. ResNet 3
  3. Patch GAN Architecture 4
  4. Software Development Life Cycle 18
  5. Use Case Diagram 19
  6. Activity Diagram 20
  7. Sequence Diagram 21
  8. Process Flow Diagram for the proposed model 22
  9. Cloudy image 29
  10. SAR image 30
  11. Cloud Free Image 30
  12. Patched Cloud Images 31
  13. Patched SAR Images 32
  14. Patched Ground Truth(cloud-free) Images 32
  15. A Cloudy and Preprocessed Cloudy Image 33
  16. Input to GAN Model 33
  17. Comparison of Generated and Original Image 34
  18. Input to GAN Model 34
  19. Comparision of Generated and original image 35

# Chapter 1 INTRODUCTION

Clouds are a common and significant obstacle in the analysis and in- terpretation of satellite imagery. They can obscure valuable information, hinder accurate land, water, and atmospheric analysis, and adversely affect various ap- plications such as land suitability analysis, outlier detection, routing of ground environmental parameters, power generation budgeting, data fusion, and statisti- cal weather prediction. Therefore, efficient methods for cloud detection, removal, and reconstruction are crucial for obtaining clear and reliable satellite images.

In recent years, various techniques have been developed to address the challenges posed by clouds in satellite imagery. Traditional approaches often rely on handcrafted features and mathematical models to identify and mitigate cloud ef- fects. However, these methods often struggle with accurately distinguishing clouds from other objects, such as water bodies, and can lead to over or under-correction errors, compromising the quality and reliability of the resulting images. With the advancements in deep learning and generative modeling, machine learning tech- niques, particularly generative adversarial networks (GANs), have emerged as pow- erful tools for cloud detection, removal, and image reconstruction. Cloud-GAN, a specific variant of GANs designed for cloud-related tasks, has shown promising results in enhancing the quality of satellite imagery by effectively detecting, re- moving, and reconstructing clouds.

The key idea behind Cloud-GAN is to leverage the power of deep neural networks to learn the complex patterns and structures associated with clouds and non-cloud regions in satellite images. Using a huge dataset of labeled satellite photos, a GAN architecture is trained to create realistic cloud-free images from cloudy input data. Simultaneously, the discriminator network learns to distinguish between real and generated images, providing feedback to the generator network to improve its performance.

## Basic Concepts

### UNet

It is a “U” shaped convolutional neural network architecture that is widely used in the field of image segmentation. The UNet architecture[9] consists of an encoder and a decoder. The encoder is a series of convolutional layers that gradually reduce the spatial dimensions of the input image while increasing the number of channels. The decoder is a series of up-sampling layers that gradually increase the spatial dimensions of the output while decreasing the number of channels. Skip connections between the encoder and decoder layers that allow for the transfer of spatial information from the encoder to the decoder. This helps to preserve the fine-grained details of the input image during the segmentation process. The Figure 1.1 depicts the UNet Architecture

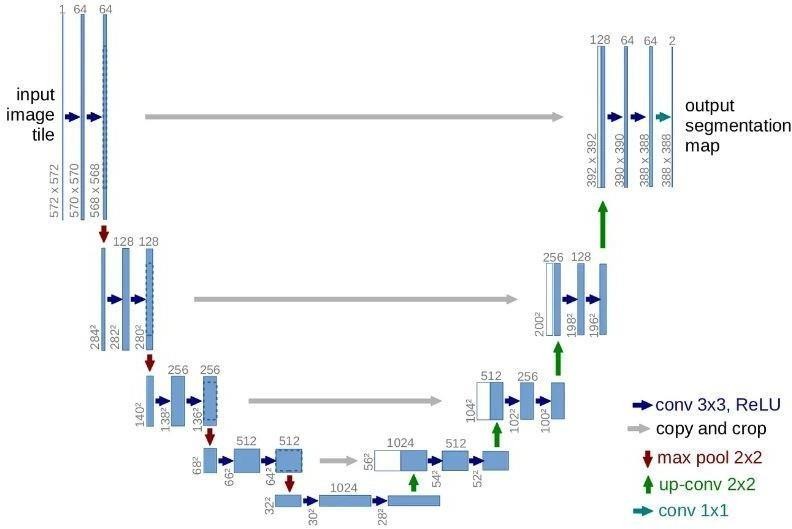


Figure 1.1: UNet Architecture

### ResNet

Figure 1.2 demonstrates the implementation of ResNet[10], a deep neural net- work architecture that addresses the challenge of vanishing gradients in highly deep networks by incorporating shortcut connections. It introduced the concept of residual learning, in which shortcut connections are added to skip some lay- ers in the network and allow the gradient to flow more easily. ResNet achieved state-of- the-art results in many image classification tasks, including the ImageNet

challenge.

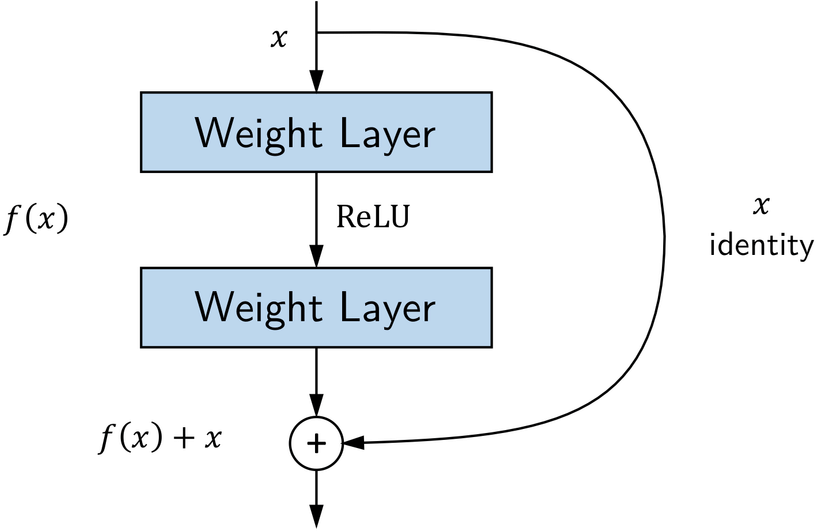


Figure 1.2: ResNet

### Patch GAN

The discriminator utilizes a Patch GAN architecture[11], which incorporates elements from the Style GAN architecture. The PatchGAN architecture utilizes multiple Transposed convolutional blocks, which are illustrated in Figure 1.3. It aims to determine the authenticity of an NxN section of an image, distinguish- ing between real and fake parts. The discriminator applies convolutions across the entire image, averaging the results to generate its output. Each block in the discriminator comprises a convolution layer, batch normalization layer, and LeakyReLU activation. The discriminator is given two inputs:

* The input image and the Target Image, which it must identify as real.
* The discriminator should classify both the generated image and the input image as fake.

The author makes the case that PatchGAN can efficiently preserve high-frequency details in the image while concentrating on low-frequency details using L1-loss, which justifies its adoption.

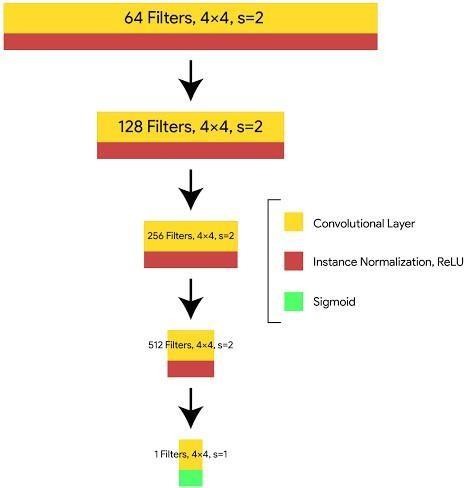


Figure 1.3: Patch GAN Architecture

### GeoTiff

GeoTIFF is a standard format for georeferenced raster images that can be used to store and exchange spatially referenced data. It includes geographic metadata such as coordinate system, projection, and datum, allowing the data to be correctly located and aligned with other spatial data. GeoTIFF is widely used in various fields such as remote sensing, GIS, and cartography. It allows for easy visualization, processing, and analysis of spatial data.

### Multispectral Remote Sensing Images

Multispectral remote sensing images are collected using sensors that detect en- ergy at multiple wavelengths of the electromagnetic spectrum. They provide in- formation about the Earth’s surface composition, structure, and properties. Ap- plications include monitoring crop health, land use and cover mapping, mineral exploration, and water quality monitoring. Techniques such as image processing and machine learning are used to analyze and interpret these images.

## Motivation

The Motivation behind this project is,

* + - Cloud removal projects aim to enhance the quality of satellite or aerial im- agery by removing clouds.
    - This process can improve the visibility, consistency, and accuracy of data analysis.
    - Such projects are useful in both scientific research and commercial contexts.

## Problem Statement

In the present scenario, the usage of satellite images for data interpretation has increased drastically. At present satellite images are being affected by the presence of the Cloud which causes difficulty in interpretation. So the proposed system focuses on the cloud removal and reconstruction of features in multispectral remote sensing image which is affected by the presence of the cloud.

## Scope

The Scope of the project is,

* + - To consider only Multi-Spectral Satellite Images only.

## Objectives

The objective of the project is,

* + - To Detect and Remove the Cloud from Satellite images.
    - To Reconstruct lost features in the Satellite Image.

## Advantages

* + - The model automatically learns and extracts essential features from the input image without the need for manual feature extraction.
    - The ResNet architecture is robust to variations in the input data and can handle input images of different resolutions.
    - It can successfully remove the cloud from satellite photos with minimal com- putational resources.

# Chapter 2 LITERATURE REVIEW

## Cloud removal in remote sensing images us- ing nonnegative matrix factorization and er- ror correction[1]

The paper presents a method for removing clouds from remote sensing im- ages by combining nonnegative matrix factorization (NMF) and error correction. Initially, the original image is decomposed into a low-rank matrix, representing the background, and a sparse matrix, containing cloud and noise details. By ap- plying NMF, the cloud pixels within the sparse matrix are identified. Next, error correction is employed to replace the cloud pixels in the sparse matrix with corre- sponding pixels from the low-rank matrix. The final output is obtained by adding the low-rank matrix and the corrected sparse matrix, resulting in a cloud-free image. This approach effectively separates cloud and background information, enabling accurate cloud removal in remote sensing images. The combination of NMF and error correction enhances the quality and applicability of the proposed method in various remote sensing applications.

#### Advantages:

The method uses NMF, which is a powerful tool for data analysis and has been widely used in remote sensing applications. The method uses NMF, which is a powerful tool for data analysis and has been widely used in remote sensing applications. The error correction step helps to improve the accuracy of cloud removal and reduces the impact of noise on the final result.

#### Disadvantages:

The method assumes that the cloud and noise information in the original image can be represented by a sparse matrix, which may not always be true. The method may not be suitable for images with complex cloud structures or overlapping clouds. The computational cost of the method may be high for large images or datasets.

## Cloud removal in Sentinel-2 imagery using a deep residual neural network and SAR- optical data fusion[2]

This paper presents a method to effectively remove clouds from Sentinel-2 satellite imagery by leveraging a deep residual neural network (DRNN) and the fusion of SAR and optical data. The approach begins by using SAR data to esti- mate cloud cover and generate a cloud mask. This mask is then used to extract cloud patches from the original Sentinel-2 imagery. To train the DRNN, a dataset of cloudy and cloud- free patches is utilized to learn the mapping between input (cloudy) and output (cloud-free) images. The trained DRNN is then employed to remove clouds from the Sentinel-2 imagery. By passing the cloud patches through the network, cloud-free patches are generated, effectively eliminating cloud-related information. Finally, the cloud-free patches are fused with the original image, re- sulting in a high-quality, cloud-free representation. This method improves the usability and interpretability of Sentinel-2 imagery, making it valuable for appli- cations such as land cover mapping, environmental monitoring, and urban devel- opment analysis. The integration of SAR data and the DRNN’s ability to learn complex mappings contribute to the robustness and accuracy of this approach.

#### Advantages:

The proposed method can effectively remove clouds from Sentinel-2 imagery, which is a common problem in remote sensing applications.The method uses a DRNN, which is a powerful tool for image processing and has been widely used in various applications.The method uses SAR data to estimate the cloud cover, which helps to improve the accuracy of cloud removal.

#### Disadvantages:

The method relies on the availability of SAR data, which may not always be available or may have low spatial resolution. The method may not be suitable for images with complex cloud structures or overlapping clouds. The computational cost of the method may be high for large images or datasets.

## Cloud and cloud shadow detection for opti-

**cal satellite imagery: Features, algorithms, validation, and prospects[3]**

This paper presents a thorough and extensive review of features and algo- rithms employed for cloud and cloud shadow detection in optical satellite imagery. It encompasses a comprehensive analysis of both pixel-based and object-based ap- proaches, assessing the performance of various algorithms based on metrics such as accuracy, speed, and complexity. Additionally, the paper addresses the chal- lenges and potential prospects associated with cloud and cloud shadow detection in optical satellite imagery. It also offers valuable insights into future research directions in this field. By providing a comprehensive overview and evaluation of existing techniques, the paper contributes to the advancement of accurate and efficient cloud and cloud shadow detection in optical satellite imagery, benefiting applications such as land cover mapping, environmental monitoring, and disaster assessment.

#### Advantages:

The paper provides a comprehensive review of the existing features and algo- rithms for cloud and cloud shadow detection in optical satellite imagery, which can serve as a valuable resource for researchers and practitioners in the field. The pa- per evaluates the performance of different algorithms using various metrics, which can help to identify the strengths and weaknesses of different approaches. The paper discusses the challenges and prospects of cloud and cloud shadow detection, which can guide future research in the field.

#### Disadvantages:

The paper mainly focuses on the review of existing features and algorithms and does not propose any new methods or techniques. The paper does not provide a detailed comparison of the different algorithms, and the performance evaluation is limited to a few metrics and datasets. The paper may not cover some of the latest developments in cloud and cloud shadow detection in optical satellite imagery.

## A deep-learning reconstruction method for

**remote sensing images with large thick cloud cover[4]**

The paper introduces a deep-learning-based approach for reconstructing remote sensing images characterized by extensive and dense cloud cover. The proposed method leverages a convolutional neural network (CNN) to learn the mapping between input images (cloudy) and desired output images (cloud-free). Training the CNN involves utilizing a dataset consisting of paired images with both cloudy and cloud-free versions. A loss function is employed to penalize dis- crepancies between the predicted and ground-truth images during training. Once trained, the CNN is deployed to reconstruct cloud-free images from the cloudy in- put images. The performance of the proposed method is evaluated using a dataset specifically composed of remote sensing images with a large thick cloud cover. This approach demonstrates the effectiveness of deep learning, specifically CNNs, in addressing the challenge of reconstructing remote sensing images obscured by substantial cloud cover, enhancing the usability and interpretability of such im- agery for various applications.

#### Advantages:

The proposed method can effectively reconstruct remote sensing images with a large thick cloud cover, which is a challenging problem in remote sensing appli- cations. The method does not require any prior information about the cloud cover or the image characteristics, which makes it more versatile and applicable to differ- ent scenarios. The method uses a deep-learning-based approach, which has been shown to be effective in various image-processing tasks.

#### Disadvantages:

The method relies on the availability of paired cloudy and cloud-free images for training, which may not always be available or may require additional data acquisition. The method may not be suitable for images with complex cloud structures or overlapping clouds. The computational cost of the method may be high for large images or datasets.

## Cloud Removal in Remote Sensing Images

**Using Generative Adversarial Networks and SAR-to- Optical Image Translation[5]**

The paper presents a cloud removal method for remote sensing images using Generative Adversarial Networks (GANs) and SAR-to-Optical Image Translation. The proposed approach involves two stages. In the first stage, a GAN is trained to generate cloud-free images based on SAR images. This training process en- ables the network to learn the underlying patterns and structures of cloud-free scenes. In the second stage, the cloud-free SAR images are employed to remove clouds from the original optical images. This is achieved by replacing the cloud regions in the optical images with corresponding regions from the cloud-free SAR images. By utilizing GANs and the fusion of SAR and optical data, the proposed method provides an effective solution for cloud removal in remote sensing imagery. The evaluation of the approach on a dataset with varying cloud cover percentages demonstrates its ability to produce high-quality cloud-free images. This technique has the potential to significantly improve the accuracy and usability of remote sensing data for various applications, including land cover analysis and environ- mental monitoring.

#### Advantages:

The proposed method can effectively remove clouds from remote sensing images, which is a critical task for many remote sensing applications. The method uses GANs and SAR-to-Optical Image Translation, which have been shown to be effective in various image processing tasks. The method does not require paired cloudy and cloud-free images for training, which makes it more applicable to dif- ferent scenarios.

#### Disadvantages:

The method relies on the availability of SAR images, which may not always be available or may require additional data acquisition. The quality of the cloud- free images generated by the GANs may not be consistent or accurate in all cases, which may affect the quality of the cloud removal.

## Thin cloud removal in optical remote sensing

**images based on generative adversarial net- works and physical model of cloud distortion [6]**

The paper introduces a method for effectively eliminating thin clouds in optical remote sensing images by combining a Generative Adversarial Network (GAN) with a physical model of cloud distortion. The method consists of two stages: cloud detection and cloud removal. In the first stage, a pre-trained U-Net is employed to identify the presence of clouds in the images. The second stage involves using a GAN-based image translation technique to remove the detected clouds and generate cloud-free images. The GAN is guided by a physical model that considers the characteristics of cloud distortion, enabling the generation of re- alistic cloud-free images. The proposed method is evaluated on two datasets with varying cloud cover percentages to assess its performance. By combining cloud detection, GAN-based image translation, and a physical model, the method effec- tively addresses the challenge of removing thin clouds in optical remote sensing images, enhancing the accuracy and usability of such imagery for diverse applica- tions.

#### Advantages:

The proposed method can effectively remove thin clouds in optical remote sensing images, which is a challenging task for many cloud removal algorithms. The use of a physical model of cloud distortion helps to generate cloud-free images with realistic texture and details, which improves the quality of the cloud removal results. The proposed method is applicable to different types of optical remote- sensing images, including multispectral and hyperspectral images.

#### Disadvantages:

The proposed method relies on the accuracy of the pre-trained U-Net for cloud detection, which may affect the quality of the cloud removal results. The use of a GAN-based image translation method may require a large amount of training data and computational resources. The effectiveness of the proposed method may be affected by the complexity and variability of the cloud cover in the remote sensing images.

## Cloud removal for remotely sensed images

**by similar pixel replacement guided with a spatio- temporal MRF model [7]**

The proposed cloud removal method for remotely sensed images offers sev- eral advantageous features. Firstly, the method employs a threshold-based ap- proach to accurately identify cloud-covered pixels in the input image. This en- sures precise localization of cloud regions for subsequent processing. Secondly, by utilizing similar images based on spatial and temporal characteristics, the method performs pixel replacement with cloud-free counterparts, enhancing the quality of the resulting cloud-free image. This contextual information contributes to the creation of visually consistent and realistic outputs. Thirdly, the incorporation of a spatio- temporal Markov Random Field (MRF) model in the third stage re- fines the cloud removal results by enforcing smoothness and consistency among neighboring pixels. This helps in maintaining the overall structure and coherence of the image after cloud removal. Lastly, the proposed method’s performance is rigorously evaluated on a dataset of remotely sensed images with varying cloud cover percentages, ensuring its effectiveness and reliability in diverse scenarios. Altogether, this method provides an effective and flexible approach for removing clouds in remotely sensed images, yielding high-quality results suitable for a range of applications.

#### Advantages:

The method uses a spatiotemporal MRF model, which can exploit the spa- tial and temporal correlation of the pixels and improve the quality of the cloud removal results. The method can handle images with complex cloud structures or overlapping clouds.

#### Disadvantages:

The quality of similar cloud-free images may not be consistent or accurate in all cases, which may affect the quality of the cloud removal results. The com- putational cost of the method may be high for large images or datasets.

## Deep learning for multi-modal classification

**of cloud, shadow and land cover scenes in PlanetScope and Sentinel-2 imagery[8]**

In this study, researchers developed deep learning models using convolu- tional neural networks (CNNs) to efficiently and accurately classify cloud, shadow, and land cover scenes in high-resolution satellite imagery. They focused on multi- label classification at the scene level, which allows for faster performance and higher generalizability. The models were trained on multi-modal satellite imagery from PlanetScope and Sentinel-2. The results showed that the CNN models had a high degree of cross-dataset generalization, performing well when applied to different datasets and geographic locations. The models achieved comparable performance to state-of-the-art methods specifically designed for cloud and shadow classification in Sentinel-2 imagery. Additionally, the researchers demonstrated the model’s po- tential for masking cloud and shadow-contaminated areas in the NDVI time series derived from the satellite imagery. Overall, this study contributes to improving satellite image indexing, retrieval, and classification, utilizing deep learning tech- niques on high-resolution satellite imagery. The findings highlight the models’ effectiveness in classifying different land cover types and monitoring changes over time.

#### Advantages:

The use of deep learning models enables efficient and accurate classifica- tion of cloud, shadow, and land cover scenes in high-resolution satellite imagery. Multi- modal classification at the scene level allows for faster performance, higher accuracy, and higher generalizability. The CNN models developed in this study showed high cross-dataset generalization ability, making them suitable for classi- fying satellite imagery with different resolutions.

#### Disadvantages:

The multi-label classification at the scene level may sacrifice some level of detail compared to pixel-level classification. The performance of the CNN models may be influenced by the quality and quantity of training data. The use of deep learning models may require significant computing resources and expertise in model training and evaluation.

# Chapter 3

**SOFTWARE REQUIREMENT ANALYSIS**

Analysis of the requirements, also known as requirements engineering, is the method of evaluating consumer demands for a new or changed product. Anal- ysis of requirements is a team activity involving a mix of experience in engineering hardware, software, and human factors, as well as skills in communicating with people. These characteristics, called criteria, have to be quantifiable, specific, and detailed. Such criteria are also termed functional specifications in software en- gineering. Analysis of specifications is an important part of project management that requires regular contact with authorized users to establish particular function- ality preferences, dispute resolution, or uncertainty in specifications as requested by the different users or community groups.

A specification on software requirements is a detailed overview of the intended function and ecosystem for the under-research program. The Software Require- ments Specification thoroughly explains what the program is going to do, and how it is supposed to function. A Software Requirement specification reduces the pro- grammer’s time and means to improve desired targets, and therefore minimizes production costs. In a wide range of real-world scenarios, a successful Software Requirement Specification determines how an application communicates with the machine hardware, other programs, and human users.

This chapter includes an analysis of the requirements for the proposed project.

This chapter contains

* Functional Requirements.
* Non-Functional Requirements.

## FUNCTIONAL REQUIREMENTS

Functional requirement analysis entails the thorough examination, analysis, and documentation of software and hardware requirements with the goal of addressing specific issues. This analysis aids in gaining a comprehensive understanding of the essential software and hardware elements required to effectively resolve the identified problems.

* + - The input images need to be preprocessed to remove any noise, distortions, or artifacts.
    - The GAN model should be able to analyze the input image, detect the clouds, remove the clouds, and Reconstruct the missing features.
    - The Adam Optimizer can be used to improve the performance of the GAN model.

## NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements describe how a system must behave and establish constraints on its functionality. This type of requirement is also known as the system’s quality attributes.

The Non-functional requirements of this project are:

#### Performance:

The system should demonstrate efficient processing and provide real-time or near-real-time results for cloud detection, removal, and image reconstruction tasks. The processing speed should be optimized to handle large-scale satellite images effectively.

#### Accuracy:

The accuracy of the Cloud-GAN model is crucial for reliable and consistent results. The model should be able to accurately remove Clouds from multispectral remote sensing images.

#### Robustness:

The system should be robust and capable of handling different weather conditions, lighting variations, and diverse types of clouds present in satellite images. It should be able to adapt to various environmental factors and deliver consistent results.

#### User Interface:

The user interface should be intuitive, user-friendly, and visually appealing. It should provide easy navigation and control for users to interact with the system, perform tasks, and visualize the results.

## Software Requirements

* + - Python programming language.
    - A web-based interactive computing platform like Jupyter Notebook or Google Colab.
    - Python Libraries like PIL, TensorFlow, OS, NumPy, matplotlib, sklearn.
    - QGIS(Quantum Geographic Information System) tool.

## Hardware Requirements

* + - Modern Operating System (windows 7 or 10/Mac OS X 10.11 or higher)
    - processor - i3 or above is preferred.
    - x86 64-bit CPU
    - RAM/Main Memory - 8GB DDR4 or above
    - GPU - 4GB or above

# Chapter 4 SOFTWARE DESIGN

Software design is a phase in a software development methodology which outcomes in a brief explanation of how to best solve the problem at hand when executed. The design will go through various iterations before finishing. The design can cover various elements of the program such as Solution Architecture, Application Structure, Database Design, Techniques for Integration etc. The concept feedback is the specification in research and planning. Software design refers not only to the system in general but to any single part of the system as well. Software design will be the coding of the program in phase.

## Software Development Life cycle

* Life Cycle model of the proposed system is Iterative Model which as follows in Figure 4.1.
* Identify the scope of the cloud removal task, including the type and size of cloud that needs to be removed, the time frame for completion, and the resources required for the task.
* Evaluate the current state of the cloud and its impact on the system. Identify any dependencies on the cloud and any potential risks associated with its removal.
* Create a plan for removing the cloud, including the necessary steps, timeline, and resources required. Determine the impact of the removal on the system and any potential disruptions.
* Execute the plan, including removing the cloud and testing the system to ensure it is functioning properly.
* Verify that the cloud has been removed successfully and that the system is functioning as expected.
* Once the DL model is trained, it is validated in which the model is evaluated with a testing data set.
* Implement the changes made to the system after the cloud has been removed.
* Monitor the system to ensure that it is functioning properly and make any necessary adjustments.

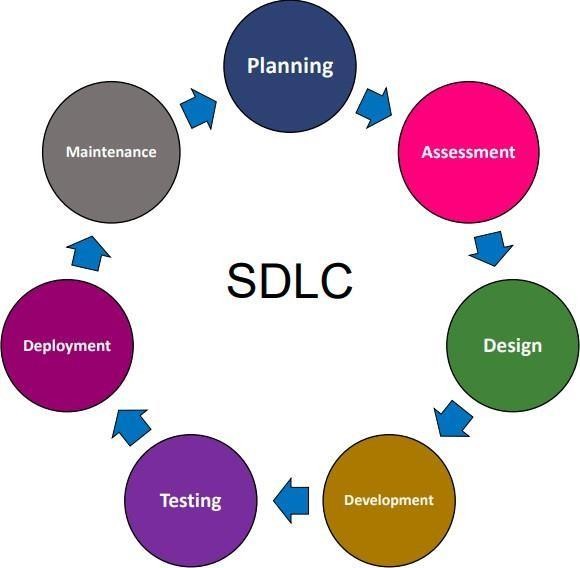


Figure 4.1: Software Development Life Cycle

## UML Diagrams

A UML diagram is a graphical representation based on the Unified Modeling Language (UML), which serves the purpose of visually depicting a system and its key components. These components may include actors, roles, actions, artifacts, or classes. The primary goal of creating UML diagrams is to facilitate a better understanding of the system, support modifications, aid in system maintenance, and document information related to the system. UML diagrams provide a standardized and visual means of representing various aspects and relationships within a system.

### Use-Case Diagram

A use case diagram showcases the different use cases or functionalities that a user can perform within the system. Figure 4.2.1 represents the use case diagram.

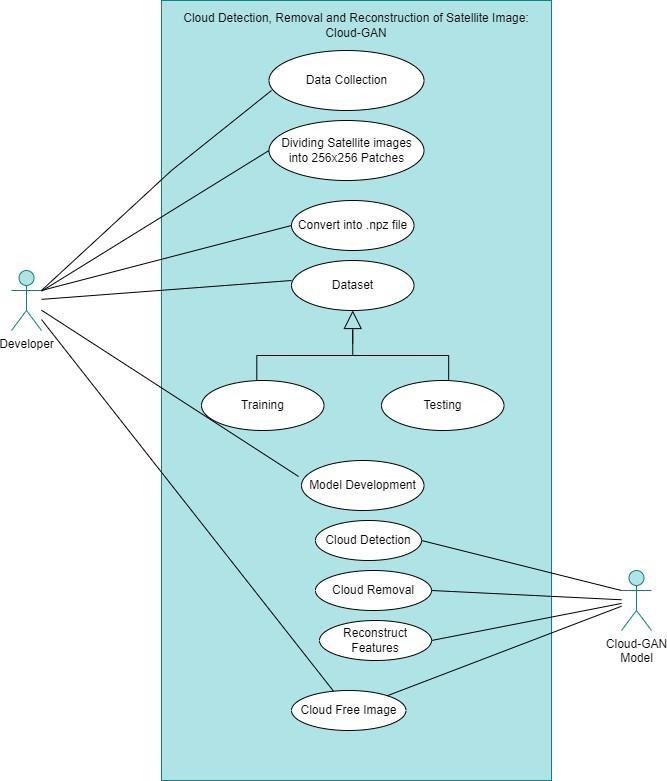


Figure 4.2: Use Case Diagram

### Activity Diagram

Activity diagrams are visual representations that depict workflows consisting of step-by-step activities and actions. They provide support for incorporating choices, iterations, and concurrent processes within the system. Figure 4.3 displays an Activity Diagram for Cloud-GAN Model.

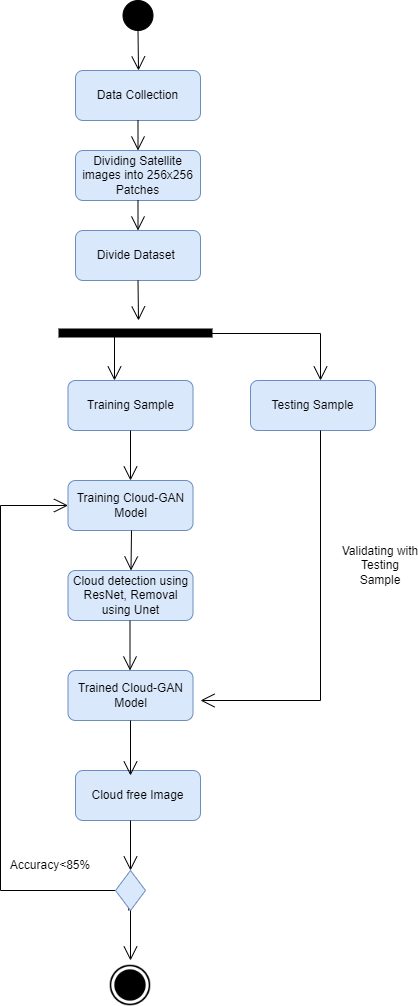


Figure 4.3: Activity Diagram

### Sequence Diagram

sequence diagram or system sequence diagram shows process interactions arranged in time sequence in the field of software engineering. Figure 4.4 presents Sequence Diagram of the Cloud-GAN model.

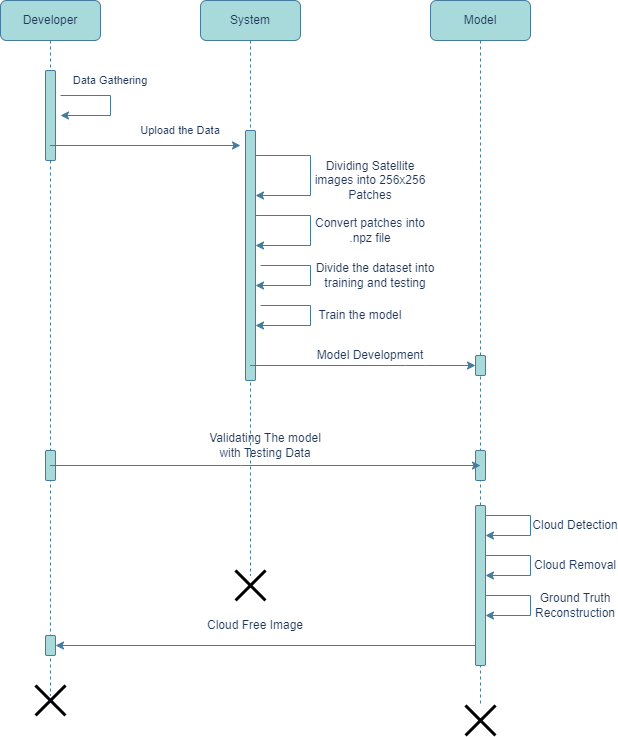


Figure 4.4: Sequence Diagram

# Chapter 5 PROPOSED SYSTEM

## Process Flow Diagram

In the figure 5.1 we can see the flowchart for Cloud Detection, Removal and reconstruction of Satellite image Using Cloud-GAN

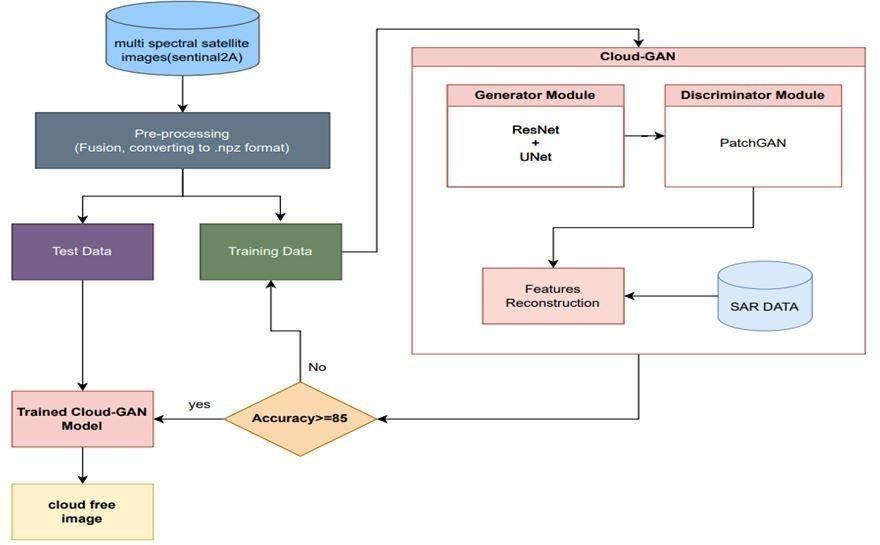


Figure 5.1: Process Flow Diagram for the proposed model

‘

## Methodology

The different modules implemented in the project are discussed in this section. The modules included are Pre-processing, Training and Testing, Activation Function and Output.

### Data Pre-processing

#### Noise Removal:

**Gaussian noise:** Gaussian noise in digital images is primarily generated by natural sources, such as the thermal vibrations of atoms and the discrete nature of radiation from warm objects. It manifests as a disturbance in the grey values within the image. Gaussian noise is characterized by a random distribution and follows a Gaussian or normal distribution pattern. This type of noise can affect the clarity and quality of digital images, introducing variations and imperfections in the grey values across the image.

#### Image Enhancement:

**Contrast adjustment:** This method adjusts the contrast of the denoised image. The function ”cv2.convertScaleAbs” linearly scales the pixel values of the image according to the specified alpha and beta parameters, which control the contrast and brightness of the image, respectively. In this case, the alpha parameter is set to 1.5, which increases the contrast of the image, and the beta parameter is set to 0, which does not change the brightness of the image.

#### NPZ file Conversion:

Converting satellite images into the NPZ format is a crucial prepro- cessing step before analysis or machine learning tasks. NPZ format enables easy transformations, data augmentation, and feature extraction. It is par- ticularly useful for training machine learning models, storing both image data and labels. This conversion streamlines data manipulation, simplifies model training, and ensures compatibility with machine learning frameworks, opti- mizing the utilization of satellite imagery for effective analysis and machine learning tasks.

### Training and Testing

* + - 1. Split the dataset into training and testing sets.
      2. Train the model with the training dataset.Training includes giving the cloudy image, SAR image as well as its corresponding clear image.
      3. Test the model using the dataset and evaluate the performance of the model.
      4. An optimization algorithm, such as stochastic gradient descent (SGD) or Adam is used to update the model’s weights during training and minimize the loss function.

### Activation Function

#### Relu Activation Function:

The ReLU activation function is used to introduce non-linearity in the output of a neural network. It returns 0 for all negative inputs and returns the input value for all positive inputs. ReLU is a piecewise linear function with a threshold at x = 0. It helps to improve the performance of the network by allowing it to learn more complex representations. It also helps to avoid the issue of ”dying ReLU” where the gradient of the function becomes 0 for all inputs. ReLU is used twice in the ResNet block code provided, after the first BatchNormalization layer and after the Add layer.

#### Sigmoid Activation Function:

The sigmoid activation function is commonly used in neural networks. It is a bounded, S-shaped function that maps input values to values between 0 and 1. It is often used to produce a probability output in binary classification problems. In the ResNet model code provided, the sigmoid function is used as the final activation function. It is applied to the output of the Conv2D layer to produce a pixel-wise probability map.

### Output

The output layer generates the cloud free image, typically using an appropriate activation function, such as sigmoid or relu, to ensure that the output pixel values are within the desired range.

## Algorithm/Pseudo Code

The algorithm for the Cloud detection, removal, and reconstruction of multispec- tral remote sensing images is described as follows:

**Input:** Cloudy Image, Ground Truth(cloud-free Image), and SAR image

**Output:** cloud-free image

### Split the Collected data into patches

**Input:** A TIF Image of size 6013 X 2957 pixels **Output:**

Patches of 256 x 256 pixels in TIF Format **Algorithm:**

**Step 1:** Set the root directory for input images and the patch size.

**Step 2:** Loop through theimagedirectoryand foreach image

**Step 2.1:** Read the image in TIFF format using tiff.imread().

**Step 2.2:** Get the nearest size of the image that is divisible by the patch size. **Step**

**2.3:** Crop the image from the top left corner using Image.crop() from PIL.

**Step 2.4:** Resize the image using Image.resize() from PIL.

**Step 2.5:** Convert the image to a numpy array using np.array().

**Step 2.6:** Extract patches from the image using patchify.patchify() function. **Step**

**2.7:** Loop through the patches and save each patch as a separate TIFF image using tiff.imwrite().

**Step 2.8:** Print the image name and the total number of patches extracted.

**Step 3:** Print the total number of patches extracted.

### Data Preprocessing

**Input:** A patched TIF Image of size 256 X 256 pixels

**Output:** Output file of .npz format

#### Algorithm:

**Step1.** Start by converting the input image to the LAB color space. This is done because the L channel contains the grayscale version of the image, while the A and B channels contain color information.

**Step 2:** Divide the image into small patches of size winSize x winSize. The winSize parameter is a user-defined value that determines the size of each patch. **Step 3:** For each patch, find the most similar patches in the image using a weighted sum of squared differences. The weights are determined by a Gaussian function of the pixel distances between the patches.

**Step 4:** Repeat steps b and c for all patches in the image.

**Step 5:**Finally, convert the image back to the original color space.

**Step 6:**Load the input image.Define a contrast factor (alpha) and a brightness factor (beta).

**Step 7:**Use the cv2.convertScaleAbsfunction to perform the contrast and bright- ness adjustment on the input image. Return the contrast-adjusted image.

**Step 8:** Read and preprocess the satellite images.Convert the preprocessed images to NumPy arrays.

**Step 9:**Create a dictionary to store the image arrays. Store the image arrays in the dictionary.

### Model Architecture: Cloud-GAN architecture

#### Algorithm Generator:

**Step 1:** Create a function that takes the input image shape and SAR (Synthetic Aperture Radar) image shape as parameters and returns the generator model. **Step 2:** Set the initial weights of the convolutional and transpose convolutional layers using a random normal distribution with a standard deviation of 0.02.

**Step 3:** Create input layers for the generator’s RGB image and SAR image inputs.

**Step 4:** Concatenate the RGB image and SAR image along the channel axis.

**Step 5:** Encoder

**Step 5.1:** Apply a series of encoder blocks, defined by the ”define encoder”

function, with increasing numbers of filters.

Σ*k* Σ *k*

*Output*(*x, y*) = *Input*(*x* + *i, y* + *j*) *· Filter*(*i, j*) (5.1)

*i*=*−k j*=*−k*

**Step 5.2:** Each encoder block performs downsampling.

**Step 6:** Apply a convolutional layer to the output of the last encoder block to reduce the spatial dimensions.

**Step 7:** Decoder

**Step 7.1:** Apply a series of decoder blocks, defined by the ”define decoder”

function, with decreasing numbers of filters.

**Step 7.2:** Each decoder block performs upsampling and concatenates skip connections from the corresponding encoder block.

**Step 8:** Output layer

**Step 8.1:** Apply a transpose convolutional layer to upsample the feature map.

**Step 8.2:** Apply the hyperbolic tangent activation function to the output,

resulting in an image with pixel values between -1 and 1.

*Output* = tanh(*Input*) (5.2)

**Step 9:** Create and compile the discriminatormodel

**Step 9.1:** Create a model that takes the source images and target images as inputs and outputs the discriminator’s prediction.

**Step 9.2:** Use the Adam optimizer with a learning rate of 0.0002 and a momentum term of 0.5.

**Step 9.3:** Define the loss as binary cross-entropy.

*Loss* = *−*(*y ×* log(*y*ˆ) + (1 *−* y) *×* log(1 *−y*ˆ)) (5.3)

**Step 9.4:** Assign a weight of 0.5 to the discriminator loss.

**Step 10:** Create and return the generator model:

#### Algorithm Discriminator:

**Step 1.** Create a function that takes the input shape of the discriminator (’des shape’) and returns the discriminator model.

**Step 2.** Set the initial weights of the convolutional layers using a randomnormal distribution with a standard deviation of 0.02.

**Step 3.** Create input layers for the source images and target images.

**Step 4.** Concatenate the source images and target images along the channel axis. **Step 5.** Apply a 4x4 convolutional layer with a stride of 2 and ’same’ padding to the merged images.

**Step 6.** Apply leaky ReLU activation with a negative slope of 0.2.

*Output* = max(0*, Input*) (5.4)

**Step 7.** Repeat the above steps for subsequent convolutional layers with in- creasing numbers of filters (64, 128, 256, 512).

**Step 8.** Batch normalization:

**Step 9.** Apply batch normalization to each convolutional layer output.

**Step 10.** Apply leaky ReLU activation with a negative slope of 0.2 to each batch- normalized output.

**Step 11.** Apply a 4x4 convolutional layer with ’same’ padding to the last activation output.

**Step 12.** Apply sigmoid activation to squash the output into the range [0, 1],

representing the probability of the input images being real or fake.

1

*Output* = 1 + *e−Input* (5.5)

**Step 13.** Create a model that takes the source images and target images as inputs and outputs the discriminator’s prediction.

**Step 14.**Use the Adam optimizer with a learning rate of 0.0002 and a momen- tum term of 0.5.

**Step 15.** Define the loss as binary cross-entropy.Assign a weight of 0.5 to the discriminator loss.

*Loss* = *−*(*y ×* log(*y*ˆ) + (1 *−* y) *×* log(1 *−y*ˆ)) (5.6)

**Step 16.** Return the discriminator model.

### Model Training

**Input:** A preprocessed 256 X 256 images

**Output:** A Trained GAN Model **Algorithm:**

**Step 1.** Use a user defined datagenerator function to divide these cloud free and Cloud images into batches, and map each of those cloud images with cloud free image.

**Step 2.**Divide the dataset into training and testing data.

Training set -*>* 80%images Testing set -*>* 20%images

Build the model using the GAN model function defined earlier. model = gan model(input shape=(256, 256, 3))

Here 3 defines color channels(RGB)

**Step 3.** Train the model with the generated test and train data with batch size of 32. Train this model till the 200 epochs.

**Step 4.** Save the model.

### Model Evaluation

**Step 1:** Identify the training loss of the model.

**Step 2:** Evaluate the model using the testing dataset by calculating the mean Absolute error(MAE) for the testing dataset.

**Step 3:** Analyze the model’s performance on different types of images by draw- ing a comparison graph between the actual and predicted image to identify the actual prediction of an image.

## Dataset Collection

* Description: The dataset used for our project consists of three distinct folders: Sentinel Cloudy, Sentinel, and SAR. These folders contain specific types of images related to our study. The Sentinel Cloudy folder contains images captured by the Sentinel satellite in cloudy conditions. The Sentinel folder contains the corresponding original images captured by the same satellite but without any cloud cover. Lastly, the SAR folder contains Synthetic Aperture Radar images, which provide additional information for our analysis.

To create this dataset, we carefully selected and clipped 20 satellite image scenes of Dima Hasao, using Google Earth Engine platform by Employing the QGIS software This process resulted in a collection of 2529 patches, with each patch having dimensions of 256x256 pixels.

* Data Format: tiff
* Number of Images: 2529
* Train set size: 80
* Test set size: 20
* Image resolution: 256x256 pixels
* The cloudy, SAR, and cloud-free image of the dataset is shown in Fig 5.2, Fig 5.3, and Fig 5.



Figure 5.2: Cloudy image

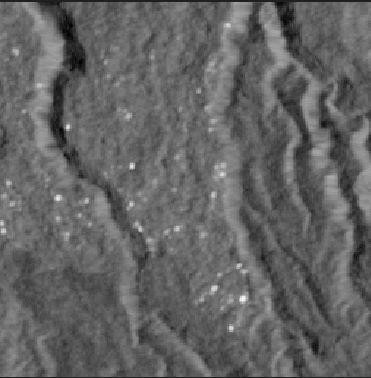


Figure 5.3: SAR image



Figure 5.4: Cloud Free Image

# Chapter 6 IMPLEMENTATION

This chapter presents the output screenshots, various test cases, result analysis of the proposed system.

## Output Screenshots

The results obtained through the proposed system are discussed in this section. The results are obtained from the successful execution of the proposed system. The images of the dataset are collected from Google Earth Engine.

Outputs from each module are shown in this chapter.

### Patching

When training any deep learning algorithm, small images produce better results allowing more accuracy without loss of information. Here the images are patched from 6013 X 2957 to 256 X 256 pixels using splitting through the patchify method as shown in figure 6.1 and 6.2.



Figure 6.1: Patched Cloud Images

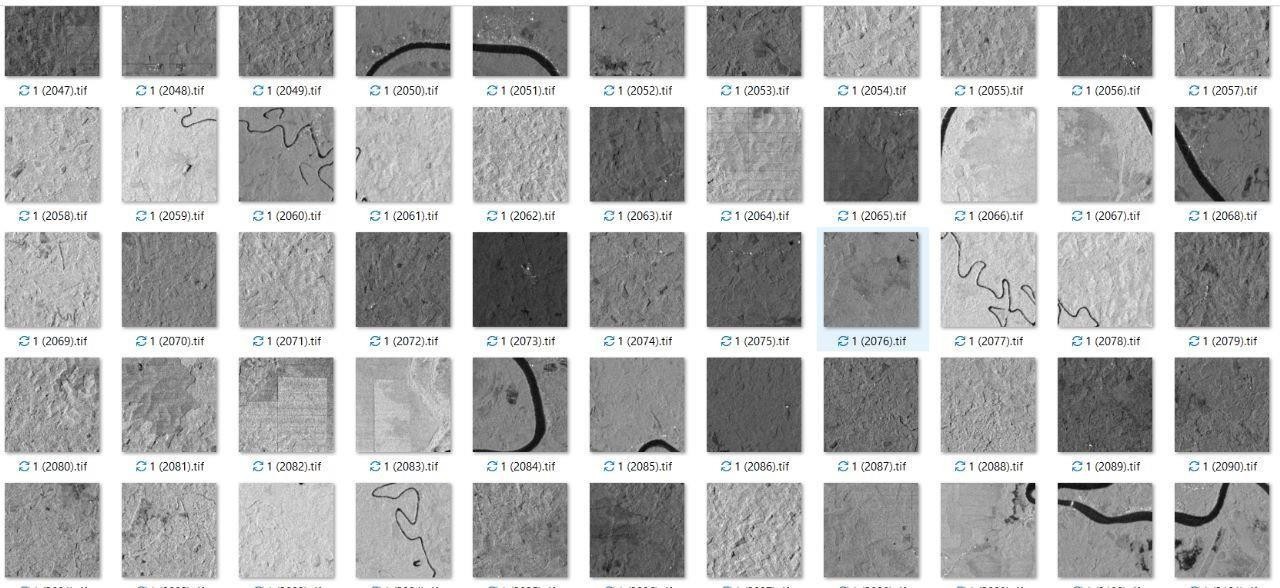


Figure 6.2: Patched SAR Images

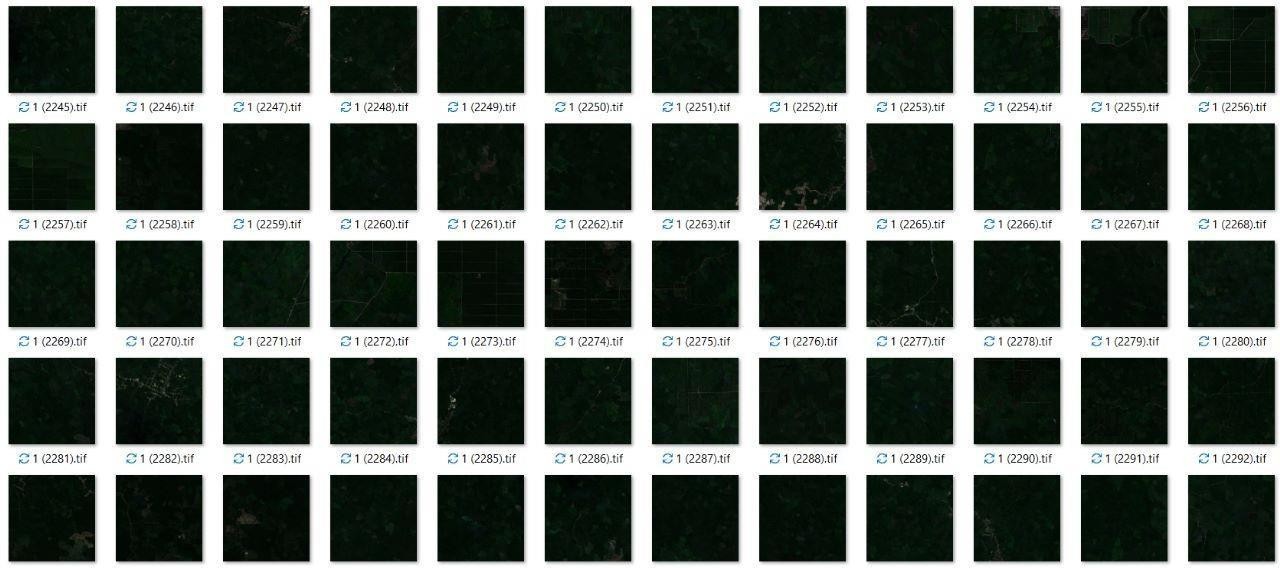


Figure 6.3: Patched Ground Truth(cloud-free) Images

### Pre-processing

* + - 1. **Denoising:** We use f1meansdenoisingcoloured() to remove the gaussian noise.
      2. **Contrast Enhancement:** We use convscaleabs() method to color contrast uniformly across the image after the denoising.

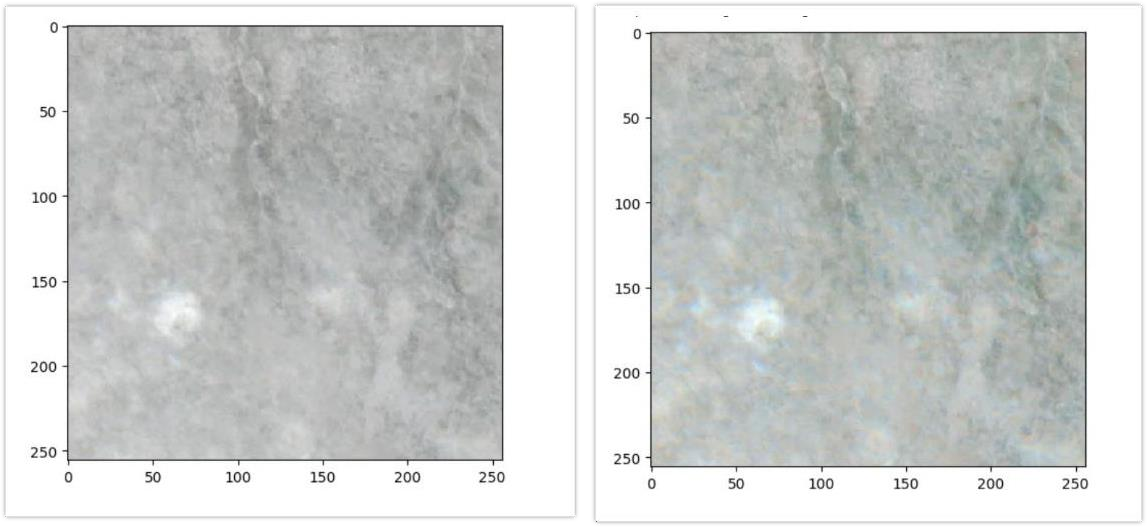
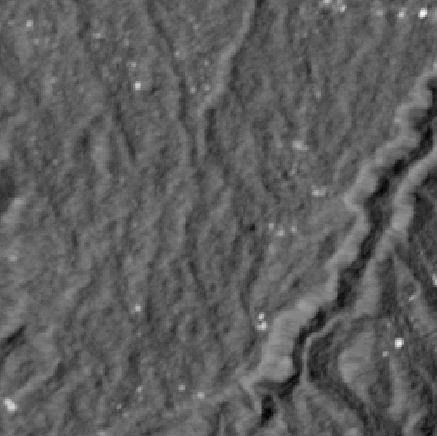


Figure 6.4: A Cloudy and Preprocessed Cloudy Image

### Cloud-free Image Prediction

The inputs into the training GAN model include Cloud, SAR, and cloud-free image data. After the successful execution of the inputs through the model, it produces outputs that remove and reconstruct features for the testing data.

Figure 6.5 and 6.6 show the comparison of the Cloud Image, Corresponding Clear Image, and predicted cloud-free image.



(a) Cloud image (b) SAR Image

Figure 6.5: Input to GAN Model

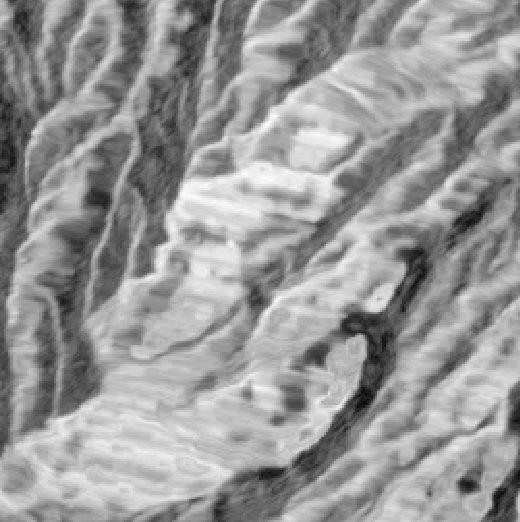
 

(a) Model predicted image (b) Ground Truth Image

Figure 6.6: Comparison of Generated and Original Image

### Untrained Cloud Image Prediction (Test Image)

Figure 6.6 and 6.7 depict the prediction of cloud-free images for the untrained cloud image using the model.



(a) Cloud image (b) SAR Image

Figure 6.7: Input to GAN Model

(a) Model predicted image (b) Ground Truth Image

Figure 6.8: Comparision of Generated and original image

## Performance Analysis

The performance of results obtained through the proposed system is discussed in this section.

**Accuracy:** A measurement of how close a predicted value is to the actual value, indicating the correctness of a model’s predictions.

Accuracy = 0.8181060791015625

**Precision:** A metric that quantifies the proportion of correctly identified positive instances out of all instances predicted as positive, providing insight into the model’s ability to avoid false positives.

Precision = 0.6230441436661839

**Recall:** A metric that measures the proportion of correctly identified positive instances out of all actual positive instances, indicating the model’s ability to capture true positives and avoid false negatives.

Recall = 0.7919434104699737

**MAE (Mean Absolute Error):** A metric used to evaluate the average magnitude of errors between predicted and actual values, providing a measure of the model’s overall prediction accuracy.

MAE = 0.1507513

# Chapter 7 CONCLUSION AND FUTURE WORK

Our project has a primary objective of detecting clouds, removing them from the images, and reconstructing the ground truth that is lost during the cloud removal process. To start, we employ preprocessing techniques on the sentinel2 cloud images in order to enhance their quality and eliminate any noise present. We acquire 20 satellite images from various scenes of Dima Hasao, specifically selecting areas where clouds obstruct the view. Using QGIS, we clip these images and convert them into smaller 256x256 patches. All 2529 resulting patches are then stored in a convenient .npz file format. The dataset is subsequently divided into training and testing sets to facilitate the evaluation of our model’s performance.

Our cloud-GAN model is designed to take three inputs: the sentinel cloudy image, SAR image, and sentinel original image (which is non-cloudy). By in- corporating these inputs, our model becomes capable of effectively detecting and removing clouds from the images. Notably, our approach also aims to reconstruct the ground truth that is lost during the cloud removal process. To achieve this, we utilize SAR images, which provide additional information about the underlying ground features. However, at present, the accuracy of our ground truth recon- struction stands at 81.81%.

In our future work, we will concentrate on enhancing the accuracy of our model to achieve more precise reconstruction of the ground truth. This involves explor- ing various strategies such as refining the architecture of our cloud-GAN model, optimizing its training process, and exploring additional data augmentation tech- niques. Furthermore, we will also focus on making the model more robust by fine- tuning its hyperparameters. By addressing these aspects, we aim to improve the overall performance of our cloud detection, removal, and ground truth recon- struction system.

# REFERENCES

1. Li, X., Wang, L., Cheng, Q., Wu, P., Gan, W., Fang, L. (2019). Cloud removal in remote sensing images using nonnegative matrix factorization and error correction. Isprs Journal of Photogrammetry and Remote Sensing, 148, 103–113. https://doi.org/10.1016/j.isprsjprs.2018.12.013
2. Meraner, A., Ebel, P., Zhu, X. X., Schmitt, M. (2020b). 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. https://doi.org/10.1016/j.isprsjprs.2020.05.013
3. Li, Z., Shen, H., Weng, Q., Zhang, Y., Dou, P., Zhang, L. (2022). Cloud and cloud shadow detection for optical satellite imagery: Features, algorithms, validation, and prospects. ISPRS Journal of Photogrammetry and Remote Sensing, 188, 89–108.

https://doi.org/10.1016/j.isprsjprs.2022.03.020

1. Tu, S., Li, X., Chong, H., Wu, Y., Li, Y., Jia, J., Wang, S., Wang, J.,

Chen, X. (2022b). A deep-learning reconstruction method for remote sensing images with large thick cloud cover. International Journal of Applied Earth Observation and Geoinformation, 115, 103079. https://doi.org/10.1016/j.jag.2022.103079

1. Darbaghshahi, F. N., Mohammadi, M. R., Soryani, M. (2021b). Cloud Removal in Remote Sensing Images Using Generative Adversarial Networks and SAR-to- Optical Image Translation. IEEE Transactions on Geoscience and Remote Sensing, 60, 1–9.

https://doi.org/10.1109/tgrs.2021.3131035

1. Li, J., Wu, G., Hu, Z., Zhang, J., Li, M., Mo, L. F., Molinier, M. (2020).

Thin cloud removal in optical remote sensing images based on generative adversarial networks and physical model of cloud distortion. Isprs Journal of Photogrammetry and Remote Sensing, 166, 373–389. https://doi.org/10.1016/j.isprsjprs.2020.06.021

1. Cheng, Q., Shen, H., Zhang, L., Yuan, Q., Zeng, C. (2014). Cloud re- moval for remotely sensed images by similar pixel replacement guided with a spatio- temporal MRF model. Isprs Journal of Photogrammetry and Remote

Sensing, 92, 54–68. https://doi.org/10.1016/j.isprsjprs.

2014.02.015

1. Shendryk, I., Rist, Y., Ticehurst, C., Thorburn, P. J. (2019). Deep learn- ing for multi-modal classification of cloud, shadow and land cover scenes in PlanetScope and Sentinel-2 imagery. Isprs Journal of Photogrammetry and Remote Sensing, 157, 124–136. https://doi.org/10.1016/j.isprsjprs.2019.08.018
2. J. Zhang, “UNet Line by Line Explanation,” Medium, Oct. 18, 2019. https://towardsdatascience.com/unet-line-by-line-explanation-9b191c76baf5
3. “ResNet (34, 50, 101): Residual CNNs for Image Classification Tasks,” Jan.

23, 2019. https://neurohive.io/en/popular-networks/resnet/

1. “Image-to-Image Translation using Pix2Pix,” GeeksforGeeks, Oct. 13, 2020. https://[www.geeksforgeeks.org/image-to-image-translation-using-pix2pix/](http://www.geeksforgeeks.org/image-to-image-translation-using-pix2pix/)
2. ”Illustration of a typical convolution layer” https://d1m75rqqgidzqn. cloud- front.net /wp-data/2020/10/19185325
3. ”Transposed Convolution” https://d1m75rqqgidzqn.cloudfront.net/wp-data

/2020/10/ 19185724

1. ”Beds of Unpooling” https: //d1m75rqqgidzqn.cloudfront.net /wp-data/2020

/10/19185548

1. ”Objective function in GAN formulation-1” https: // i0.wp.com/neptune.ai/ wp- content/uploads/fig4.-Objective-function-in- GAN-formulation.png ? re- size = 636
2. ”Objective function in GAN formulation-2” https: // i0.wp.com/neptune.ai/ wp- content/uploads/fig4.-Objective-function-in-GAN-formulation.png ? re- size

=636

1. D. S. Candra, S. Phinn and P. Scarth, ”Cloud and cloud shadow removal of landsat 8 images using Multitemporal Cloud Removal method,” 2017 6th In- ternational Conference on Agro-Geoinformatics, 2017, pp. 1-5, doi: 10.1109

/Agro-Geoinformatics. 2017.8047007.

1. X. Zhang, L. Liu, X. Chen, S. Xie and L. Lei, ”A Novel Multitemporal Cloud and Cloud Shadow Detection Method Using the Integrated Cloud Z-Scores Model,” in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 12, no. 1, pp. 123-134, Jan. 2019, doi: 10.1109/ JSTARS.2018.2889150.