REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

CLOUD DETECTION IN SATELLITE IMAGES

20011023 — Mehmet Alperen ÖLÇER
19011038 — Şevval BULBURU

COMPUTER PROJECT

Advisor
Prof. Dr. Mine Elif KARSLIGİL

May, 2023

ACKNOWLEDGEMENTS

Yildiz Technical University, founded in 1911 in Istanbul, Turkey, is a prominent government institution with 10 faculties, 2 institutes, and approximately 25,000 students. It is also regarded as one of the greatest in the country.

We'd like to thank our instructor Mine Elif Karsligil for her significant assistance, continuous follow-up, and direction throughout my thesis on "Cloud Detection in Satellite Images." Her knowledge of computer science, computer vision, algorithms, computer learning, artificial intelligence, image processing, pattern recognition, neural networks, technology and engineering was important in the project's success.

Mehmet Alperen ÖLÇER
Şevval BULBURU

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

RS-Net	Remote Sensing Network
CNN	Convolutional Neural Network
L8	Landsat 8
L7	Landsat 7
IOU	Intersection over union
CCA	Cloud Cover Assessment
NSS	Natural scene statistic
F1	F Score
SVM	Support Vector Machine
SLIC	Simple Linear Iterative Clustering

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

CLOUD DETECTION IN SATELLITE IMAGES

Mehmet Alperen ÖLÇER
Şevval BULBURU

Department of Computer Engineering
Computer Project

Advisor: Prof. Dr. Mine Elif KARSLIGİL

Humans have attempted to fully understand and explain nature throughout history. Weather events may now be forecast owing to technological advancements. This is primarily made possible by tracking the clouds.

Cloud detection uses image processing technology that detects and analyzes clouds in satellite pictures. This method is utilized in meteorology, environmental monitoring, agriculture, forestry, urban planning, military intelligence, and a variety of other fields. Cloud detection technology is given by software that scans satellite photos to assess cloud features such as density, height, size, movement, and kind. This technology enables the collection of data for a variety of reasons, including environmental monitoring and natural catastrophe management.

Some satellite imagery studies make use of data collected from the Earth's surface. Clouds in the atmosphere, on the other hand, can impede the efficiency of such investigations when they are between the surface and the satellite. The primary purpose of this research is to solve such issues and enable the usage of cloud detection in applications. As a solution to this challenge, the U-Net architecture for the model, was adopted. The achieved binary accuracy of %91, obtained after training, can be interpreted as a measure of success in the model's performance.

Keywords: Cloud detection, Image processing, satellite imagery, meteorology, environmental monitoring, data collection, U-Net architecture, semantic segmentation, Landsat 8.

ÖZET

UYDU GÖRÜNTÜLERİNDEN BULUT TESPİTİ

Mehmet Alperen ÖLÇER
Şevval BULBURU

Bilgisayar Mühendisliği Bölümü
Bilgisayar Projesi

Danışman: Prof. Dr. Mine Elif KARSLIGİL

İnsanoğlu geçmişten günümüze kadar doğayı anlama ve analiz etme çabasında olmuştur. Gelişen teknoloji ile hava olayları tahmin edilebilir bir konuma gelmiştir. Bu durumu başlıca bulutların hareketlerini takip edebilmek mümkün hale getirmiştir.

Bulut tespiti, uydu görüntülerinde bulutların tespit edilmesi ve özelliklerinin analiz edilmesi için görüntü işleme teknikleriyle yapılır. Bu teknik, meteoroloji, çevre izleme, tarım, ormancılık, kent planlaması, askeri istihbarat ve diğer birçok uygulama alanında kullanılmaktadır. Bulut tespiti teknolojisi, uydu görüntülerini işleyen yazılımlar aracılığıyla sağlanır ve bulutların yoğunluğu, yüksekliği, boyutu, hareketi ve tipi gibi özelliklerini belirlemek için çeşitli algoritmalar kullanılır. Bu teknoloji, farklı amaçlara hizmet eden veriler elde edilmesine olanak tanır ve çevre izleme, doğal afet yönetimi gibi alanlarda da kullanılır.

Uydu görüntülerinde yapılan bazı çalışmalarda yeryüzünden toplanan verilerle çalışılır. Fakat atmosferdeki bulutlar yeryüzünden toplanacak veriler ile uydu arasında olduğundan, çalışmaların efektif olmasına engel olmaktadır. Yapılan projenin temel hedefi bu gibi sorunlara çözüm olmakla birlikte bulut tespiti ile gerçeklenebilecek uygulamalarda kullanımı sağlamaktır. İlgili soruna çözüm olarak U-Net mimarisi ile oluşturulmuş CNN ağı ile model oluşturulmuştur. Eğitimden sonra elde edilen %91 doğruluk oranı, modelin performansındaki başarının bir ölçüsü olarak yorumlanabilir.

Anahtar Kelimeler: Bulut tespiti, Görüntü işleme, Uydu görüntüleri, Meteoroloji, Çevre izleme, Veri toplama, U-Net mimarisi, semantik segmentasyon, Landsat 8.

1 Introduction

A cloud is an apparent collection of atmospherically condensed water vapor that is normally floating far above the level of the earth. Clouds are crucial because they hand over rain, but they also get in the way of satellite photos' attempts to study the earth's surface. Since clouds on the input image are considered noise, detecting and removing clouds from satellite images is a crucial preprocessing step for the majority of remote sensing applications. The Earth's temperature, the dynamics of the terrestrial atmosphere, the thermodynamic chemistry, and radiative transport are all influenced by clouds. In addition to these, after the Kahramanmaras earthquake on February 6, 2023, the detection of the collapsed areas could not be realized in real-time using satellite images. The reason for this is that clear images cannot be obtained from the ground due to the dust clouds caused by the debris and snowfall. The difficulties encountered have hindered the timely delivery of the necessary aid by making it difficult to identify the areas that have been destroyed. It is anticipated that this situation can be overcome with a design that can detect clouds in the affected areas. Satellite images are one of the most powerful and important tools used by the scientist for the study of earth and space science.

In the past few years, numerous satellite imaging resources have been employed by researchers to examine and monitor the Earth's atmosphere. These resources encompass the MODIS data, the Greenhouse Gas Observing Satellite (GOSAT), Landsat-8, a U.S. Earth observation satellite, France's Sentinel-1/2 satellites, the Spot series of satellites from Europe, the Indian Ocean Infrared Satellite FY2G, China's high-resolution Earth observation satellite GaoFen-1, and the GOSAT. Furthermore, geostationary weather satellites such as Landsat-8, GOES, INSAT-3D/3DR, Kalpana-1, and the Meteosat series from the Indian Space Research Organization, the NOAA Advanced Very High-Resolution Radiometer, the QuickBird satellite, and the Ikonos satellites have also been used. These satellites, positioned high above the Earth, offer invaluable perspectives on cloud movements and other atmospheric events.[1]

Cloud detection from satellite imagery can be difficult due to many kind of causes. The complexity of cloud patterns, which can vary substantially in size, shape, and texture, is one of the key challenges. Clouds can also overlap with other terrain features such as mountains, coastlines, or forests, making detection even more difficult. Another challenge in detecting clouds is the presence of other meteorological phenomena such as haze, dust, or smoke, which can mimic the appearance of clouds and generate false detections. Furthermore, clouds can have different optical properties at different times of day and in different weather conditions, making detection even more difficult.

In conclusion, cloud detection has many challenges despite of its importance. This project is an alternative solution to cloud detection from satellite images. Semantic segmantation based on U-Net is used for classification and a cloud identification model was trained in the scope of the research by including extreme cases in the dataset obtained from Landsat-8 satellite pictures utilized in the project.

2
Preliminary Examination

Cloud detection from satellite photos is critical for a variety of purposes, including weather forecasting, emergency management, and climate research. This report presents a preliminary examination of a project aimed at developing an accurate and efficient cloud detection algorithm from satellite images. To detect clouds in satellite photos, the research was employed machine learning models and image processing techniques. This objective was been accomplished by semantic segmentation and remote sensing using the U-Net architecture. Satellite pictures were been created using the Landsat-8 dataset.

2.1 Literature Review

The research about cloud detection algorithms likely started in the 1990s. Some studies have been examined and five of them information are given below.

- [2] The research was published in 10 October 2017. Model was built on deep learning system to categorize cloud and snow at the pixel level using fully convolutional neural networks. To learn deep patterns, a specialized fully convolutional network was implemented. Then, to integrate low-level spatial and high-level semantic information at the same time, a multiscale prediction technique was used. 50 Gaofen #1 satellite images with the resolution of 16 m, where each image is with a size of 13400×12000 pixels was used for evaluate the model. Precision, recall and mIOU have been used for measuring success. According to research, precision, recall and mIOU were calculated as 92.4, 91.5, 90.6 respectively.
- [3] The research was published in 14 November 2018. It mostly dwells on small and thin cloud detection. The model has been trained and tested on the special dataset gathered from the G1 satellite. The dataset includes 100 testing and 38 training images. After dividing the images into patches, 206000 patches

has gained. The model was designed with NSS and Gabor features. Three steps take place. First, segmentation is done first using the SLIC technique. Then, The uncertain superpixels are then divided into potential thin clouds, thick clouds and nonclouds using the NSS features. Lastly, Gabor features are input to an SVM that can distinguish between cloudy and snowy regions. Training and validation patches are 8400 in total. Testing patches are 9200. Precision and recall values have been used for measuring success. Precision and recall were calculated 91.61, 86.39 respectively.

- [4] The research was published in May 2019. CNN based on segNet was used in the research. The algorithm contains 13 convolution layer and 13 deconvolution layer. The model was trained and evaluated using 32 Landsat 8 images and 38 Landsat 7 images. %60 of the dataset was used for training, %10 was used for backpropogation and %30 was used for testing. Algorithm compared with CFMask and the result of this, the overall accuracies are improved from 89.88 and 84.58 to 95.26 and 95.47 for the Landsat 7 and Landsat 8 images.

- [5] The research was published in January 2019. The purpose of the research is prevent the incorrect cloud detection and make algorithm efficient for the snowy areas. To achieve this RS-Net model was developed. The model was trained and evaluated using the Landsat 8 Biome and SPARCS datasets. Landsat 8 Biome dataset has a size of 195 GB and contains 96 Landsat 8 scenes. Dataset includes 8 different biome and they divided into 4 classes, namely 'cloud', 'thin cloud', 'cloud shadow', and 'clear'. Landsat 8 SPARCS dataset has size of 1.6 GB and consists of 80 1000 x 1000 pixels scenes. It is divided into 7 classes, namely 'cloud shadow', 'cloud shadow over water', 'water', 'snow', 'land', 'cloud', and 'flooded'. Two dataset was merged and SPARCS dataset is converted to cloud, shadow and clear classes. Algorithm was evaluated and tested with combinations of two dataset. Accuracy, precision, and recall metrics have been used for measuring success. The best results are become by using Sparcs dataset for training and testing the model. Accuracy and F1 score were calculated 94.54, 85.59 respectively.

- C: [6] The research published in 2021. The model has generated by using U-Net on Landsat 8 satellite images. In the dataset, 38 images have been divided into 384*384 patches. The patches are split into training, validation, and testing. Training and validation patches are 8400 in total. Testing patches are 9200. Precision and recall values have been used for measuring success. Precision and recall values were calculated %85.28, %83.23 respectively.

The abstracts of the all researches are given below at Figure 2.1.

ID	A	B	C	D	E
ARTICLE	A Cloud Detection Algorithm For Satellite Imagery Based On Deep Learning	Cloud And Cloud Shadow Detection In Landsat Imagery Based On Deep Convolutional Neural Networks	Towards Robust Cloud Detection in Satellite Images Using U-Nets	Cloud Detection in Satellite Images Based on Natural Scene Statistics and Gabor Features	Distinguishing Cloud and Snow In Satellite Images Via Deep Convolutional Network
DATE	8.01.2019	4.03.2019	16.07.2021	14.11.2018	10.10.2017
AUTHOR	Denmark - Aarhus University	China - Zhejiang University US - University of California US - South Dakota State University	Bartosz Grabowski Maciej Ziaja Michal Kawulok Jakub Nalepa	Beijing Key Laboratory of Embedded Real-Time Information Processing Technology, School of Information and Electronics, Beijing Institute of Technology, Beijing,	State Key Laboratory of Space-Ground Integrated Information Technology, Space Star Technology Cooperation
TECHNIQUE	Fmask Algorithm U-Net Architecture RS-Net Architecture (based on U-Net)	CFMask Algorithm CNN	U-Net Architecture	NSS Gabor features	CNN Specialized Multiscale Prediction Module
DATASET	Landsat 8 Biome SPARCS	Landsat 7 (4500 pics) Landsat 8 (4000 pics) Cloud and Cloud Shadow Reference Data	38 Landsat-8 CCA set	Private Dataset	Gaofen 1
DATASET SIZE	The 195 GB Landsat 8 Biome dataset consists of 96 Landsat 8 scenes the smaller 1.6 GB Landsat 8 SPARCS dataset consists of 80 1000 × 1000 pixels scene	206 Scenes 96 Scenes	8400 training and validation patches in total 9201 test patches	100 test images 38 training images	50 Gaofen #1 satellite images with the resolution of 16 m each image is with a size of 13400×12000 pixels
TRAINING DATASET	Biome Ground Truth SPARCS Fmask SPARCS Ground Truth	Landsat 7 Landsat 8	Landsat 8	Private Dataset	Gaofen 1
ACCURACY TYPE / ACCURACY	F1 : 78.35 F1 : 79.34 F1 : 85.59	Overall Accuracy : 94.33(L8)-95.26(L7) Producer's Accuracy : 91.47(L8)-83.60(L7) User's Accuracy : 97.40(L8)-83.18(L7)	Precision : 85.28 Recall : 83.23	Precision : 91.61 Recall : 86.39	Precision : 92.4 Recall : 91.5 mIOU : 90.6

Figure 2.1 Abstracts of All Researches

A: [5], B: [4], C: [6], D: [3], E: [2]

3
System Analysis and Feasibility

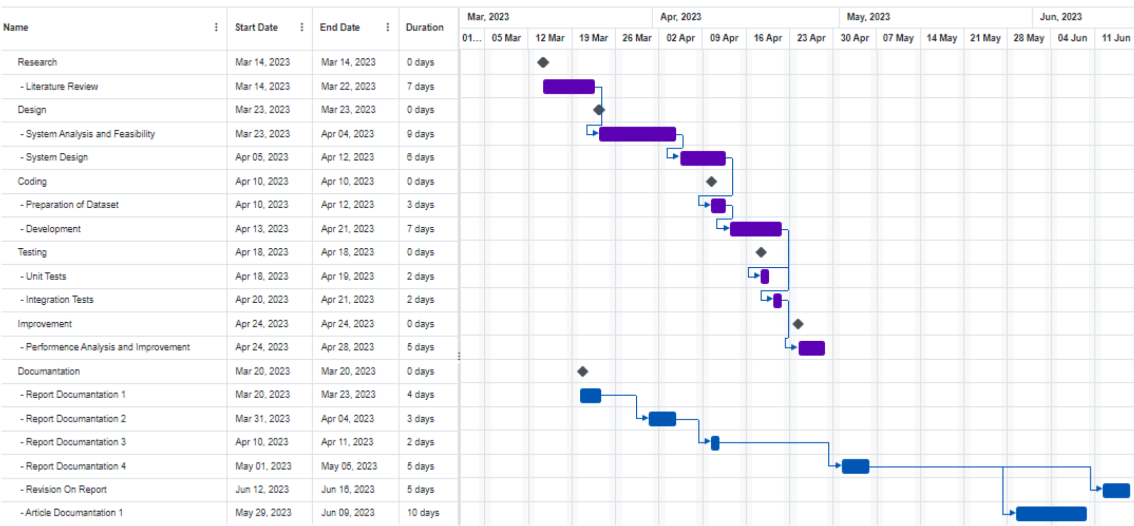
The goal of the project is to develop a model for cloud detection based on U-Net architecture which uses Convolutional Neural Network. In this regard, feasibility studies were conducted.

3.1 Feasibility

3.1.1 Time Feasibility

A three-month period was determined to be proper for project completion. The Waterfall approach was used to create the time feasibility aspect.

Gannt Schema for the project is given below at Figure 3.1.



Name	Start Date	End Date	Duration
Research	Mar 14, 2023	Mar 14, 2023	0 days
- Literature Review	Mar 14, 2023	Mar 22, 2023	7 days
Design	Mar 23, 2023	Mar 23, 2023	0 days
- System Analysis and Feasibility	Mar 23, 2023	Apr 04, 2023	9 days
- System Design	Apr 05, 2023	Apr 12, 2023	6 days
Coding	Apr 10, 2023	Apr 10, 2023	0 days
- Preparation of Dataset	Apr 10, 2023	Apr 12, 2023	3 days
- Development	Apr 13, 2023	Apr 21, 2023	7 days
Testing	Apr 18, 2023	Apr 18, 2023	0 days
- Unit Tests	Apr 18, 2023	Apr 19, 2023	2 days
- Integration Tests	Apr 20, 2023	Apr 21, 2023	2 days
Improvement	Apr 24, 2023	Apr 24, 2023	0 days
- Performence Analysis and Improvement	Apr 24, 2023	Apr 28, 2023	5 days
Documantation	Mar 20, 2023	Mar 20, 2023	0 days
- Report Documantation 1	Mar 20, 2023	Mar 23, 2023	4 days
- Report Documantation 2	Mar 31, 2023	Apr 04, 2023	3 days
- Report Documantation 3	Apr 10, 2023	Apr 11, 2023	2 days
- Report Documantation 4	May 01, 2023	May 05, 2023	5 days
- Revision On Report	Jun 12, 2023	Jun 16, 2023	5 days
- Article Documantation 1	May 29, 2023	Jun 09, 2023	10 days

Figure 3.1 Gantt Schema for Project

3.1.2 Technical Feasibility

3.1.2.1 Hardware Feasibility

The initial work on training the model was done on Google Colab and a draft was created. However, due to GPU and time limitations in the free trial of Colab, the training of the entire dataset was done on our local machine. The dataset contains a total of 16GB of data and the specifications of the computer needs to create the model are as follows:

- Hard Drive: ADATA SX8100NP M2 SSD
- Processor: AMD Ryzen 5600X 6-Core
- Graphics: NVIDIA GeForce RTX 3060 Ti
- RAM: 32GB

The least specifications above are sufficient for training and usage. Two computer which has the specifitacations were needed for the project.

3.1.2.2 Software Feasibility

Python language was preferred for model training. The use of libraries such as TensorFlow, Matplotlib, scikit-image, NumPy, and Hub, as well as their user-friendly interfaces, validates their inclusion in the project. Tensorflow library was used for the design of the model. Python version 3.9.16 and compatible library versions were selected.

3.1.3 Legal Feasibility

The project's goal is to offer an approach for detecting clouds in satellite images. There is no need for an ethics committee's approval because the initiative does not include any possible damage or ethical issues. The dataset used is made up of Landsat 8 satellite images, which are freely available for public use and do not require any rights purchases. In terms of software licensing, open-source resources were used, and because the software was developed internally, no license rights were violated.

3.1.4 Economic Feasibility

The cost of the hardware used is 50.000 TL, since there are two computers with each one costing 25000. An employee works for 40.000 TL monthly. Two employees who work quarter-time for 4 days a week, paid 2 * 40.000 * 0.8 * 0.25 = 16.000. The payment for two individuals for 3 months is 16.000 * 3 = 48.000. So for this project total salary is calculated as 48,000 TL. The total budget is 50.000 + 48.000 = 98.000 TL

Costs of each components in a computer are given below at Table 3.1.

Hardware - Cost Table	
Component	Cost
ADATA SX8100NP M2 SSD	1.500 TL
AMD Ryzen 5600X 6-Core	5.000 TL
NVIDIA GeForce RTX 3060 Ti	11.000 TL
32 GB RAM	2.500 TL
Other components for computer	5.000 TL
Total	25.000 TL

Table 3.1 Hardware - Cost Table for One Computer

3.2 Elements of the System

The goal of this project is to create and implement an efficient cloud detection algorithm that simplifies the use of human, software, hardware, and data resources from satellite images.

The Table 3.2 below provides an overview of the project's various resources, including hardware, software, human, and data resources.

Reasources	
Resource	Element
Hardware Resources	2 Computer
Software Resources	Python - Jupyter Notebook
Human Resources	Şevval Bulburu Mehmet Alperen Ölçer
Data Resources	Landsat 8 Satellite Images

Table 3.2 Resources which used for the project

4
System Design

The goal of this project is to develop a model for cloud detection based on U-Net architecture employing semantic segmentation.

4.1 Dataset Design

The model is trained using Landsat 8 satellite images. L8 dataset contains 38 scenes with each scene divided into 384*384 pixels.There were 8,400 patches created for training and 9,200 patches were created for testing. Each patch consist of four spectral channels which are Red, Green, Blue and Near Infrared. It is important to note that channels are not combined in the dataset.

The patches were examined to evaluate the presence of a significant number of geographical challenges within the dataset for the model's evaluation. Patches from various channels were combined into a single patch and thoroughly reviewed to achieve this. Following that, it was determined that the dataset is appropriate for training the model.

All of the channelized images were combined and processed to create a dataset that is model-compatible. This dataset is stored in a location called "hub". This storage technique makes it simple for any Python environment to use the dataset.

The dataset includes unchallenging and challenging images for training and testing of the model. Examples of unchallenging images are given below Figure 4.1 while challenging images are provided below Figure 4.2.

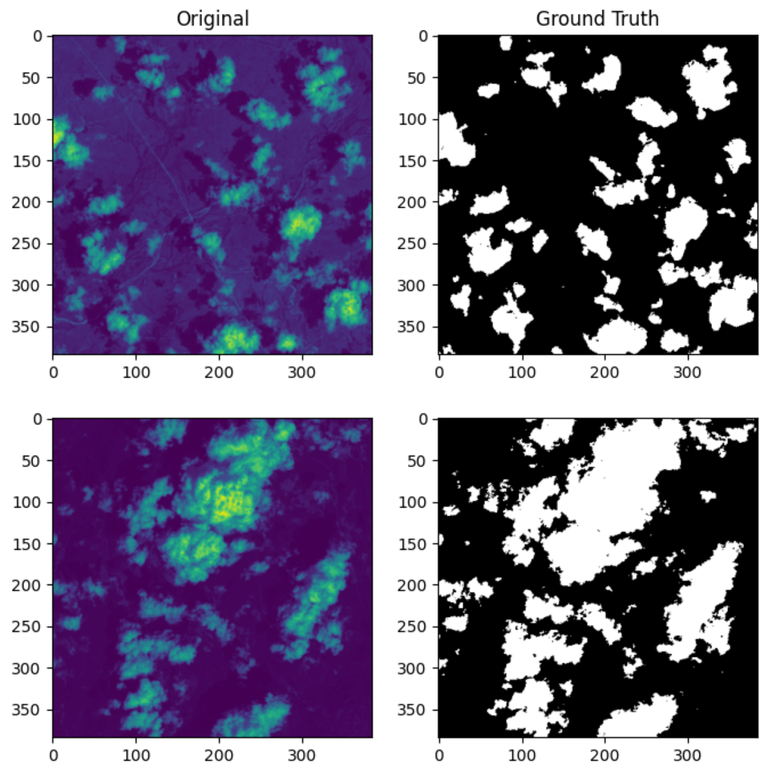


Figure 4.1 Example for unchallenging train images

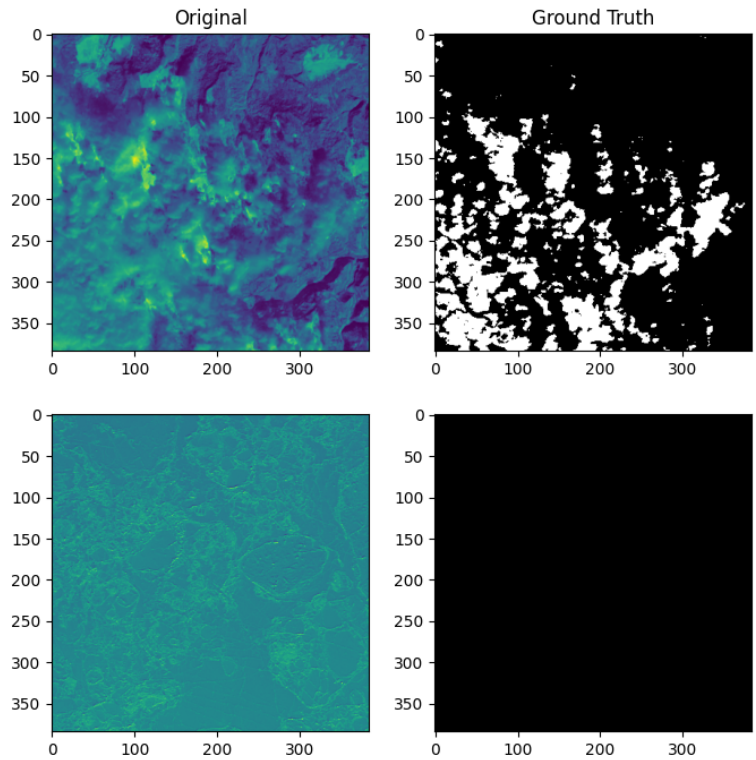


Figure 4.2 Example for challenging train images

4.2 Cloud Detection Algorithm

Deep learning is a machine learning subject which is focused on training artificial neural networks how to recognize information and make predictions or choices. The algorithms develop to identify patterns and extract relevant information from massive volumes of data. They are intended to learn layered structures for input data by gradually learning increasingly complicated and abstract attributes. These hierarchical representations are learned through an approach known as training, in which the neural network's internal parameters are adjusted based on the input data and the desired output.The artificial neural network, which consists of numerous layers of linked nodes, is the most significant component of deep learning. Each node in the network conducts a basic computation and sends the results to the other nodes in the network. Weights are related to node connections, which are modified during the training phase to optimize the network's performance.

Project is created with a U-Net-based software implementation of a Convolutional Neural Network for image segmentation. A deep learning architecture known for its excellent results in image segmentation tasks is U-Net. Effective feature extraction and spatial information reconstruction are made possible by the system's distinctive U-shaped structure, which includes an encoding and decoding network.

The Encoder Network is similar to convolutional neural network. It employs convolutional and pooling layers to extract high-level features and capture context from input images, therefore reducing the spatial dimensions and increasing the number of feature channels. The Decoder Network; employs upsampling, concatenation operations, and convolutional layers to improve spatial resolution and produce a segmentation map. It combines feature maps and improves spatial resolution to provide the decoder precise localization data. With the use of skip connections, the respective levels of the encoder and decoder are connected by U-Net.

The final layer of the U-Net architecture is a 1x1 convolutional layer followed by a softmax activation function. The result is a pixel-by-pixel prediction probability map, where each pixel denotes the likelihood that a certain class is present.

The represantation of the U-NET Architecture is given below Figure 4.3.

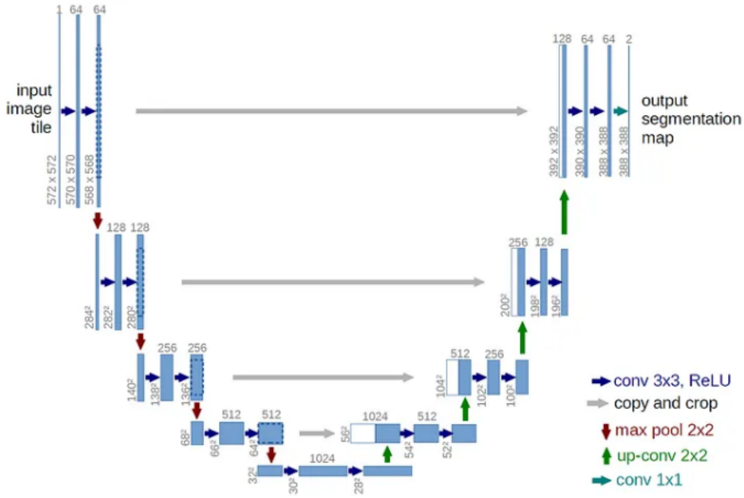


Figure 4.3 U-NET Architecture

The model is created using a collection of cloud pictures and ground truth masks supplied by the hub library. The code offers features for data loading and preprocessing, model definition and training, likewise performance evaluation on a test set. The Adam optimizer and the binary cross-entropy loss function are used to optimize the model. Evaluation metrics such as precision, recall, accuracy and mean Intersection over Union are utilized to assess the model's performance. During the training, checkpoints are implemented to store the optimal weights based on validation performance. Finally, the trained model is applied to forecast segmentation predictions on a test set of images.

The algorithm patterns are represented by the block diagram shown in Figure 4.4.

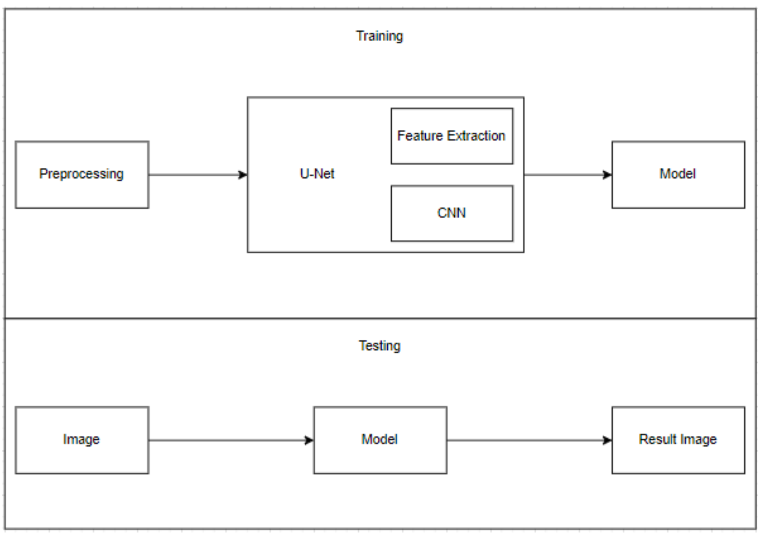


Figure 4.4 Block Diagram

5
Experimental Results

The Landsat 8 dataset contains 5154 images. The model was trained with 4124 random images and it was tested with 515 random images in the dataset. The performance measuring criterias are 'accuracy', 'precision', 'recall', 'IOU' and 'loss function' for the model's evaluation.

Several experiments were carried out to determine the best functions and parameters for training and testing the model with Landsat 8 data. 100 images was used for experiments. An owerview of the experiments and their results are provided below:

- **U-Net Architecture Transformation:**

  The U-Net design was updated by adding an extra convolution function to each of the initial U-Net function's ten convolution layers. In this modification, the first convolution function received the result from the previous layer, while the second convolution function took the convoluted layer as a parameter. The new U-Net architecture was employed for both model training and testing, utilizing a dataset of 100 images. The results demonstrated that the Intersection over Union value decreased from 0.30 to 0.23, while the accuracy increased from 0.54 to 0.76.

- **Modifying Parameters in the U-Net Architecture:**

  In the U-Net design, the convolution function contains numerous hyperparameters, one of which is "kernel size." The kernel size is the size of the iterating window over the image during the convolution operation.

  The size of the kernel influences the information acquired by the convolutional layer. Larger kernel sizes can capture more global information, whereas lower kernel sizes can hold more local details.

Two U-Net designs were trained and evaluated to identify the optimal kernel size for a specific set of data. One architecture used a kernel size of 3, while the other used a kernel size of 5.

Increasing the kernel size from 3 to 5 resulted in a modest improvement in the Intersection over Union value, which improved from 0.30 to 0.31. Yet, there was a minor drop in accuracy, which decreased from 0.54 to 0.53.

These findings indicate that the kernel size chosen might have a minor impact on the performance of the U-Net model, and it is critical to evaluate the trade-off between capturing global information and conserving local features when picking the kernel size for a specific dataset.

- **Modifying Optimization:**

  In the U-Net model, the compile function is used to prepare the model for training. There are three parameters required: optimizer, loss, and metrics.

  The optimizer is an optimization technique that determines how the weights of the model are changed during training. One of the optimizer's key parameters is the "learning rate." The learning rate is a hyperparameter that determines the size of the steps used to update the model's weights during training. It has an effect on how quickly or slowly the model learns from training data. A lower learning rate provides greater stability and the possibility of optimal solutions, whereas a higher learning rate allows for faster convergence.

  Three U-Net topologies were trained and tested to identify the best learning rate for a given dataset. Each architecture employed a distinct learning rate: 1e-3, 1e-4, and 1e-6.

  The results revealed that a learning rate of 1e-6 was insufficient for the model. With this learning rate, the precision and recall scores were approximately zero, indicating poor performance. Learning rates of 1e-3 and 1e-4, on the other hand, were more appropriate for the model. The IOU values obtained were 0.33 and 0.30, with corresponding accuracies of 0.55 and 0.54.

  These findings emphasize the significance of choosing an adequate learning rate while training the U-Net model. Both 1e-3 and 1e-4 performed better than the extremely low learning rate of 1e-6, demonstrating the importance of establishing the correct balance between learning speed and accuracy.

- **Modifying Compile Parameters:**

  The compile function in the U-Net model is used to prepare the model for training. Three parameters are required: optimizer, loss, and metrics.

  Loss function is the function that calculates the discrepancy between the ground truth and the predicted output of the model. The evolution metric is a parameter used to monitor the model's performance throughout training and testing. The accuracy metric is the most commonly used. It calculates the percentage of correctly predicted classes out of all samples for each sample. The binary accuracy metric computes correct predicted labels from all samples.

  Four different compile functions were used for training and testing to determine the best compile parameters for a specific dataset. Binary cross entropy and dice loss functions were among the loss functions considered. Similarly, the metrics chosen included accuracy metrics and binary accuracy metrics. The goal of experimenting with different combinations of these four parameters was to find the compile function that produced the best results for the given dataset.

  The results showed that using the dice loss function and binary accuracy metric produced the highest IOU and accuracy values. In particular, the IOU value for the given dataset was 0.76, while the accuracy was 0.84.

All examination outcomes demonstrated the optimal U-Net architecture and compile function parameters. The features and parameters were determined after thorough examinations. The learning rate parameter in the optimization function was specifically set to "1e -4," while the kernel size was set at 3. Convolution layers were also implemented to the U - Net function. One of the compile function's parameters, the loss function, was set to the dice loss function. Similarly, the compile function's evaluation metric parameter was set to the binary accuracy metric. As a result, significant improvements in the acquired findings were pointed out.

The accuracy of the training and evolution is 0.91. The result outcomes after evaluation of the model are shown below in Figure 5.1, Figure 5.2.

Black pixels show the absence of clouds, whereas white pixels show the presence of clouds. The resulting image uses purple pixels to represent areas that are inaccurately detected in the prediction. When the pixel is black in the ground truth but white in the prediction then it is purple in the result. The same place is represented by these pixels. In contrast, green pixels show regions that are mistakenly identified as non-overcast in the prediction while being cloudy in the ground truth.

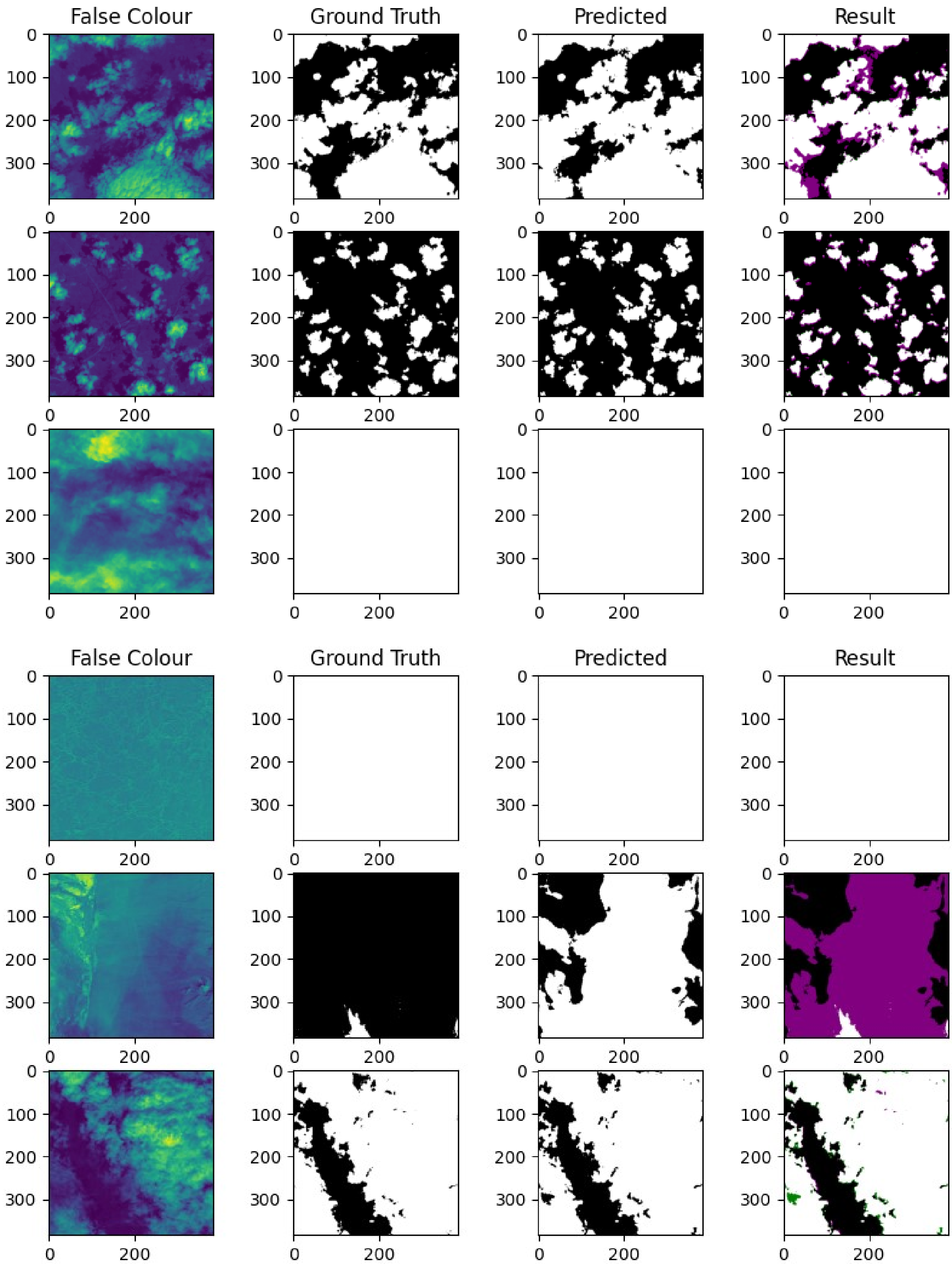


Figure 5.1 Result Table 1

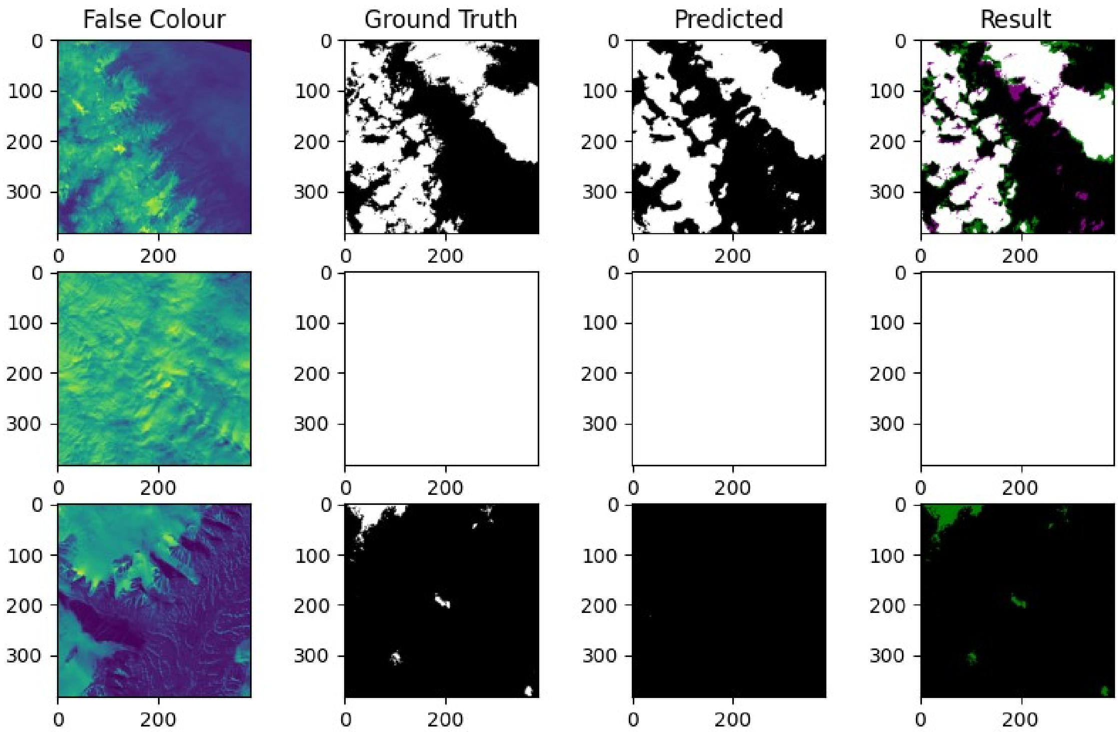


Figure 5.2 Result Table 2

The success measurements obtained from training and evaluating the model are provided in Table 5.1.

Performance Criteria	Value
IOU	0.8305
Precision	0.9309
Recall	0.9056
Binary Accuracy	0.9162
Loss	0.2952

Table 5.1 Performance Criteria - Value

6
Performance Analysis

4124 random images was employed in training, 515 random images was employed in validation and 515 random images was employed in testing process. The performance measuring criterias are 'accuracy', 'precision', 'recall', 'IOU' and 'loss function' for the model's evaluation.

The checkpoint callback is utilized during training, and the model is trained on a train dataset for 20 epochs while utilizing a different validation dataset for validation. When the model gets its best IoU score on the validation data, the weights will be saved.

While training the model, the CPU performance is shown in below Figure 6.1.

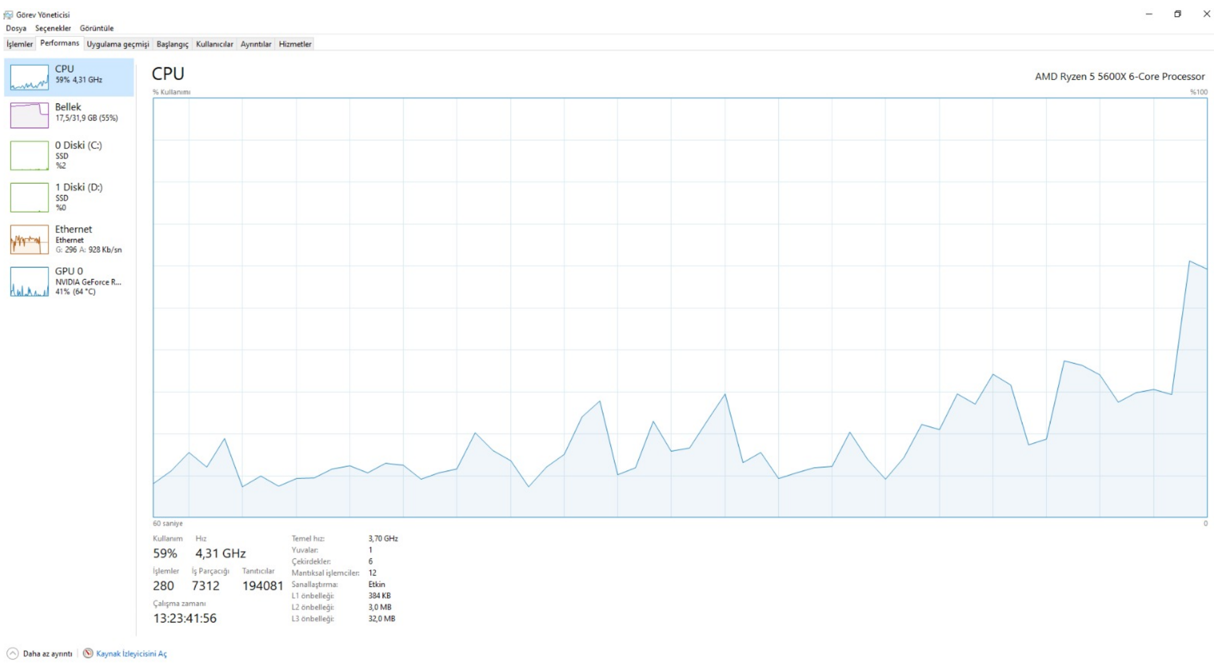


Figure 6.1 CPU Usage

While training the model, the memory performance is shown in below Figure 6.2.

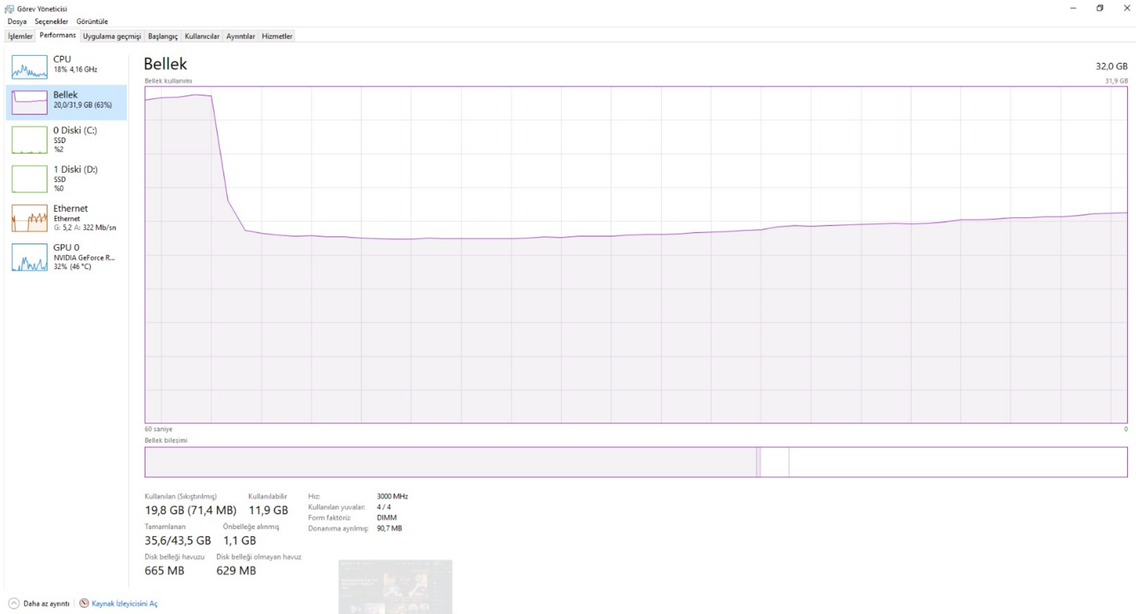


Figure 6.2 Memory Usage

While training the model with, the GPU performance is shown in below Figure 6.3.

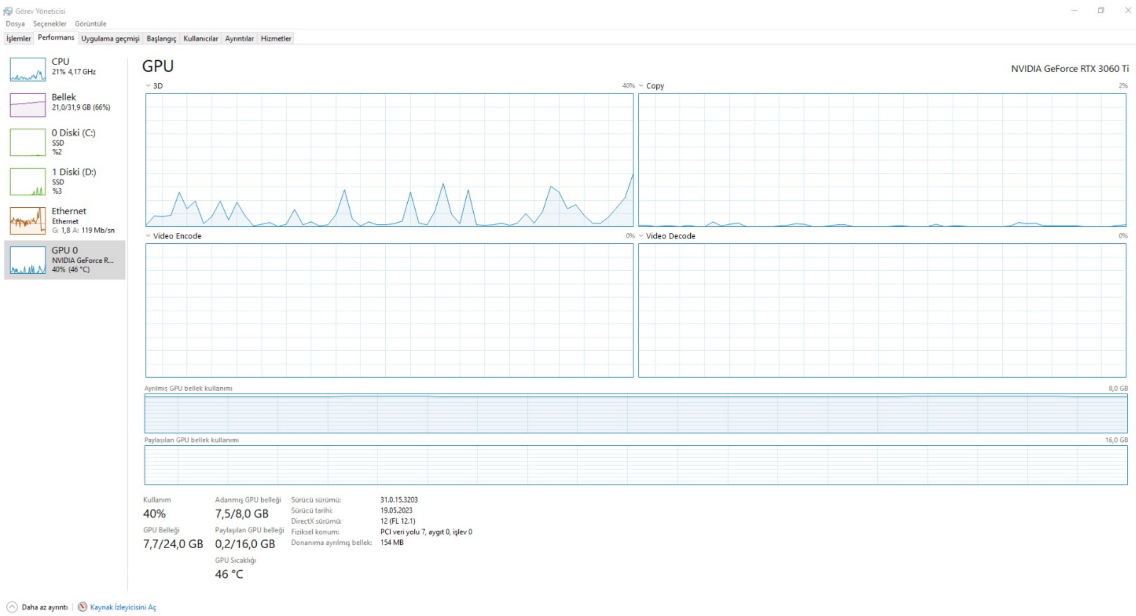


Figure 6.3 GPU Usage

The Figure 6.4 below presents the model performance on epochs during training with 4124 train images and 515 validation images. The training had taken 6.91 hours in total.

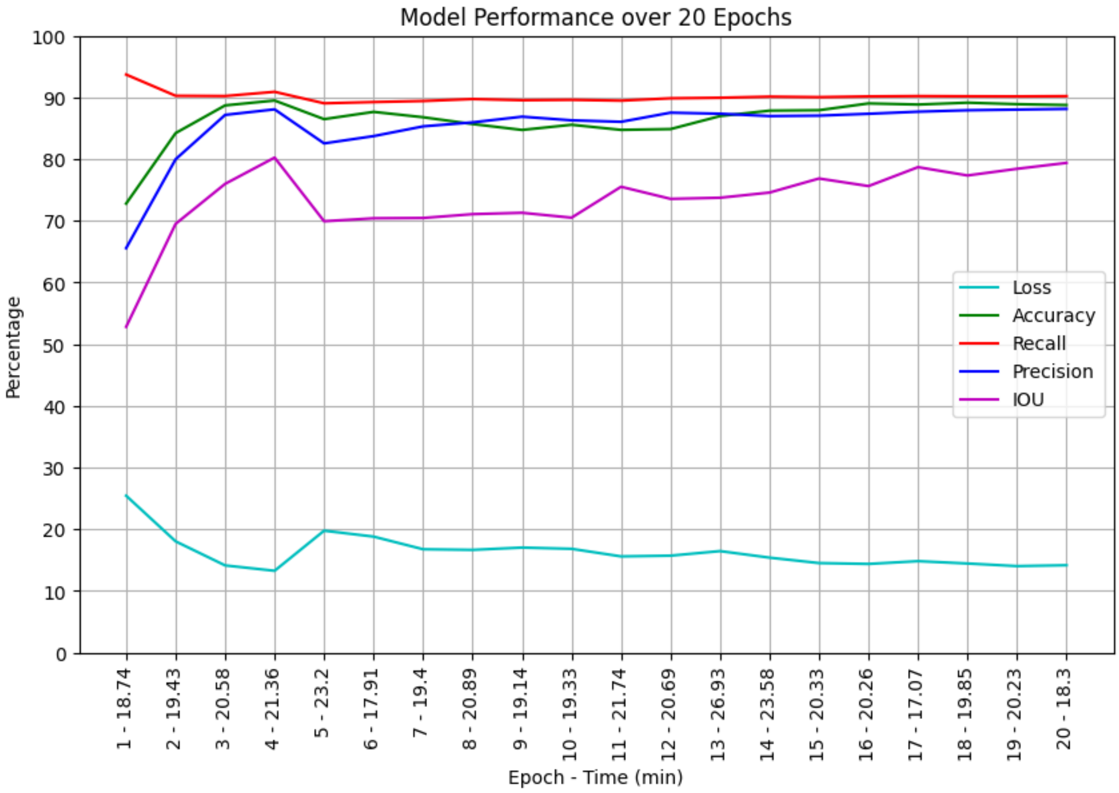


Figure 6.4 The best weights saved at fourth epoch

7 Conclusion

The project focuses on development of a deep learning model that can recognize clouds in satellite images. The U-Net architecture, a convolutional neural network based method, is used to define the model. The Landsat 8 satellite imagery dataset was used to training and testing the model. The task of binary classification for cloud presence and absence was successfully handled by the model. Additionally, the model performed semantic segmentation to precisely identify cloud regions within the images.

The model produced good results in cloud detection from satellite images after intensive training over 20 epochs using IOU as the saving criterion. The accuracy value was reached at %91. By leveraging the information provided by the model, the non-informative regions dominated by clouds can be efficiently filtered before subjecting the imagery to further algorithmic processing. This contributes to enhancing the overall effectiveness and accuracy of information retrieval procedures utilizing satellite imagery.

References

[1] S. Mahajan and B. Fataniya, "Cloud detection methodologies: Variants and development—a review," *Complex & Intelligent Systems*, vol. 6, pp. 251–261, 2020.

[2] Y. Zhan, J. Wang, J. Shi, G. Cheng, L. Yao, and W. Sun, "Distinguishing cloud and snow in satellite images via deep convolutional network," *IEEE geoscience and remote sensing letters*, vol. 14, no. 10, pp. 1785–1789, 2017.

[3] C. Deng, Z. Li, W. Wang, S. Wang, L. Tang, and A. C. Bovik, "Cloud detection in satellite images based on natural scene statistics and gabor features," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 4, pp. 608–612, 2018.

[4] D. Chai, S. Newsam, H. K. Zhang, Y. Qiu, and J. Huang, "Cloud and cloud shadow detection in landsat imagery based on deep convolutional neural networks," *Remote sensing of environment*, vol. 225, pp. 307–316, 2019.

[5] J. H. Jeppesen, R. H. Jacobsen, F. Inceoglu, and T. S. Toftegaard, "A cloud detection algorithm for satellite imagery based on deep learning," *Remote sensing of environment*, vol. 229, pp. 247–259, 2019.

[6] B. Grabowski, M. Ziaja, M. Kawulok, and J. Nalepa, "Towards robust cloud detection in satellite images using u-nets," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, IEEE, 2021, pp. 4099–4102.

Curriculum Vitae

FIRST MEMBER

Name-Surname: Mehmet Alperen ÖLÇER
Birthdate and Place of Birth: 19.07.2001, Balikesir
E-mail: alperen.olcer@std.yildiz.edu.tr
Phone: 0553 841 79 50
Practical Training: During my internship at Turkcell, I received hands-on experience in software development and DevOps by working on a Jira plugin in Java and SQL. This experience helped me improve my abilities and knowledge of the sector.

SECOND MEMBER

Name-Surname: Şevval BULBURU
Birthdate and Place of Birth: 02.03.2000, Istanbul
E-mail: sevval.bulburu@std.yildiz.edu.tr
Phone: 0541 840 39 60
Practical Training: I worked on developing game engine and GUI elements with C++ at Nitra Game Software.

Project System Informations

System and Software: Jupyter Notebook, Python3, Tensorflow, Keras, Hub
Required RAM: 32GB
Required Disk: 32GB