

# CLOUD DETECTION IN SATELLITE IMAGES

Şevval Bulburu, Mehmet Alperen Ölçer  
Bilgisayar Mühendisliği Bölümü  
Yıldız Teknik Üniversitesi, 34220 İstanbul, Türkiye  
{sevval.bulburu, alperen.olcer}@yildiz.edu.tr

**Özetçe** —İnsanoğlu, doğayı anlama ve açıklama çabası içerisinde olmuştur ve teknolojiadaki ilerlemeler sayesinde hava olayları tahmin edilebilir hale gelmiştir. Bu başarı büyük ölçüde bulutların hareketlerinin takip edilmesiyle mümkün olmaktadır. Bulut tespiti, uydu görüntülerindeki bulutların tespit edilmesi ve analiz edilmesi için görüntü işleme teknolojilerinden yararlanmaktadır. Bu yöntem meteoroloji, çevre izleme, tarım, ormancılık, kent planlaması, askeri istihbarat gibi birçok alanda kullanılmaktadır. Bulut tespit teknolojisi, uydu fotoğraflarını analiz eden yazılımlar aracılığıyla bulut yoğunluğu, yükseklik, boyut, hareket ve tür gibi özellikleri değerlendirerek veri toplamayı mümkün kılmaktadır. Bazı durumlarda uydu görüntüleme çalışmaları yeryüzünden toplanan verilerle gerçekleştirilmektedir. Fakat atmosferdeki bulutlar, yeryüzü ile uydu arasında yer aldığı için bu çalışmalara verimlilik açısından zorluklar yaratmaktadır. Bu araştırmanın temel amacı, bu gibi sorunları çözmek ve bulut tespitin uygulamalarda kullanımını mümkün kılmaktır. Bu zorluğun üstesinden gelmek için U-Net mimarisi kullanılarak bir model oluşturulmuştur. Eğitim sonucunda elde edilen %91 başarı oranı, modelin performansının bir ölçüsü olarak değerlendirilmektedir.

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

**Abstract**—Throughout history, humans have strived to comprehend and explain nature and advancements in technology now allow for weather forecasting by tracking clouds. Cloud detection, employing image processing technology, is widely utilized across various fields such as meteorology, environmental monitoring, agriculture, forestry, urban planning, and military intelligence. Software-based cloud detection scans satellite images to analyze cloud properties including density, height, size, movement, and type, facilitating data collection for purposes like environmental monitoring and disaster management. However, when clouds obstruct the view between the surface and satellite, satellite imagery studies face efficiency challenges. This research aims to overcome such obstacles by adopting the U-Net architecture. With an impressive accuracy of %91 achieved through training, the model's performance can be deemed successful.

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

## I. INTRODUCTION

Clouds, being condensed atmospheric water vapor, serve vital roles like rain provision, but also obstruct the examination of the Earth's surface via satellite imagery, necessitating their detection and removal for most remote sensing applications. This importance was underscored by the inability to capture real-time images after the Kahramanmaraş earthquake in 2023 due to dust

and snowfall clouds. Researchers have utilized various satellite data sources, including MODIS, GOSAT, Landsat-8, France's Sentinel-1/2 satellites, Europe's Spot series, FY2G, GaoFen-1, and geostationary weather satellites to study atmospheric phenomena. However, cloud detection is challenging due to varying cloud patterns and their overlap with other features, different optical properties, and resemblance to other meteorological phenomena. This project offers a solution, employing semantic segmentation based on U-Net for cloud detection, and a model trained on Landsat-8 data was developed to handle such extreme cases.

## II. PRELIMINARY EXAMINATION

The report explores the development of a cloud detection algorithm from satellite images, crucial for weather forecasting, emergency management, and climate research. The project employs machine learning models and image processing techniques for cloud detection, specifically using semantic segmentation and remote sensing through the U-Net architecture, with satellite images sourced from the Landsat-8 dataset.

### A. Literature Review

Cloud detection algorithm research, which likely began in the 1990s, has seen considerable advancements. [1] In a 2017 study, a deep learning model was built to distinguish cloud and snow using convolutional neural networks and a multiscale prediction technique, achieving precision, recall, and mIOU of 92.4, 91.5, and 90.6 respectively. [2] A 2018 study focused on small and thin cloud detection, employing an NSS and Gabor featured model with a precision and recall of 91.61 and 86.39. [3] In 2019, a study used a CNN (Convolutional Neural Network) based on segNet, improving overall accuracies to 95.26 and 95.47 for Landsat 7 and 8 images. [4] Another 2019 study sought to prevent incorrect cloud detection in snowy areas using the RS-Net model, achieving an accuracy and F1 score of 94.54 and 85.59. [5] In 2021, a study employed U-Net on Landsat 8 images, achieving precision and recall values of 85.28 and 83.23.

The abstracts of the all researches are given below at Figure 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	8.02.2019	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 Gredowski Maciej Zieliński Michał Kawulok Jakub Narepa	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 Based on U-Net	CFMask Algorithm CNN	U-Net Architecture	NIS Gabor Features	CNN Specialized Multiscale Prediction Module
DATASET	Landsat 8 Biome SPANCS	Landsat 7 (4500 pics) Landsat 8 (4000 pics) Cloud and Cloud Shadow Reference Data	38 Landsat-8 CCA set	Private Dataset	Geofern 1
DATASET SIZE	The 195 GB Landsat 8 Biome dataset consists of 96 Landsat 8 scenes the smaller 1.6 GB Landsat 8 SPANCS 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 Geofern #1 satellite images with the resolution of 30 m each image is with a size of 13400×12000 pixels
TRAINING DATASET	Biome Ground Truth SPANCS Fmask SPANCS Ground Truth	Landsat 7 Landsat 8	Landsat 8	Private Dataset	Geofern 1
ACCURACY TIME / 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: 89.28 Recall: 83.23	Precision: 91.61 Recall: 86.59	Precision: 92.4 Recall: 91.5 mIOU: 96.6

Figure 1 Abstracts of All Researches

A: [1], B: [2], C: [3], D: [4], E: [5]

### III. SYSTEM DESIGN

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

#### A. Dataset Design

The model's training utilized Landsat 8 satellite images, with the L8 dataset consisting of 38 scenes divided into 384x384 pixel patches. For training purposes, 8,400 patches were created, while 9,200 patches were designated for testing. Each patch comprised four spectral channels: Red, Green, Blue, and Near Infrared. Notably, the dataset did not include combined channels.

To assess the dataset's suitability for model evaluation, the patches were examined to identify any significant geographical challenges. Patches from different channels were merged into single patches and thoroughly evaluated, confirming the dataset's appropriateness for model training.

All channelized images were combined and processed to generate a model-compatible dataset, stored in a location referred to as the "hub." This storage approach facilitates easy utilization of the dataset in any Python environment.

The dataset includes unchallenging and challenging images for training and testing of the model. Examples of unchallenging images are given below Figure 2 while challenging images are provided below Figure 3.

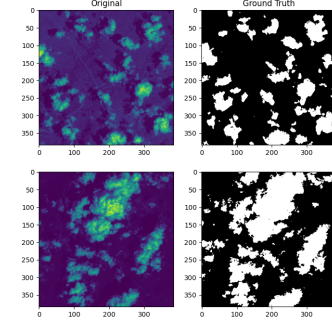


Figure 2 Example for unchallenging train images

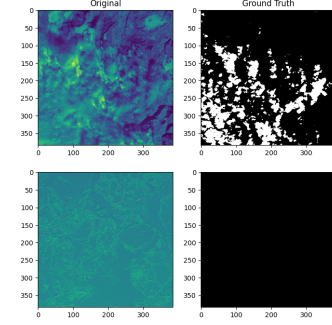


Figure 3 Example for challenging train images

#### B. Cloud Detection Algorithm

Deep learning is a machine learning discipline that focuses on training artificial neural networks to recognize information and make predictions. It utilizes algorithms that can identify patterns and extract relevant information from large datasets. Deep learning models are designed to learn hierarchical structures in data by gradually acquiring more complex and abstract features. This learning process, known as training, involves adjusting the internal parameters of the neural network based on input data and desired output. The key component of deep learning is the artificial neural network, which consists of multiple layers of interconnected nodes. Each node performs computations and shares the results with other nodes in the network. The connections between nodes, represented by weights, are adjusted during training to optimize the network's performance.

In the context of image segmentation, a project was created using a U-Net-based software implementation of a Convolutional Neural Network. U-Net is a popular deep learning architecture known for its effectiveness in image segmentation tasks. Its U-shaped structure, consisting of an encoder and decoder network, enables efficient feature extraction and spatial information reconstruction.

The encoder network, similar to a convolutional neural network, employs convolutional and pooling layers to extract high-level features and capture context from input images. This reduces the spatial dimensions and increases the number of feature channels. On the other hand, the decoder network utilizes upsampling, concatenation operations, and convolutional layers to improve spatial resolution and generate a segmentation map. It combines

feature maps and enhances spatial resolution to provide precise localization information. U-Net incorporates skip connections that connect corresponding levels of the encoder and decoder networks.

The final layer of the U-Net architecture consists of a 1x1 convolutional layer followed by a softmax activation function. This layer produces a pixel-by-pixel prediction probability map, where each pixel represents the likelihood of a particular class being present in the image.

The representation of the U-NET Architecture is given below Figure 4.

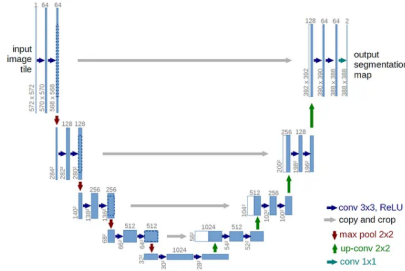


Figure 4 U-NET Architecture

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 5.

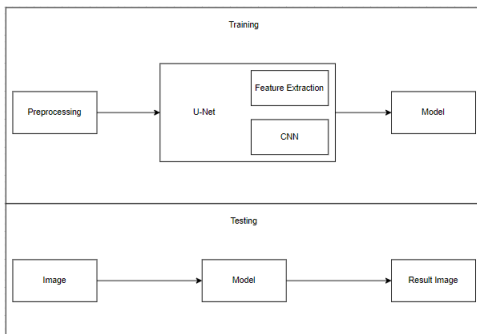


Figure 5 Block Diagram

#### IV. 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. An overview 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 (Intersection Over Union) 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 6 and Figure 7.

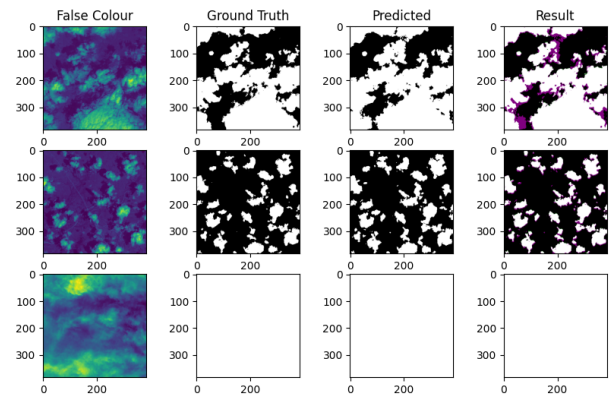


Figure 6 Result Table 1

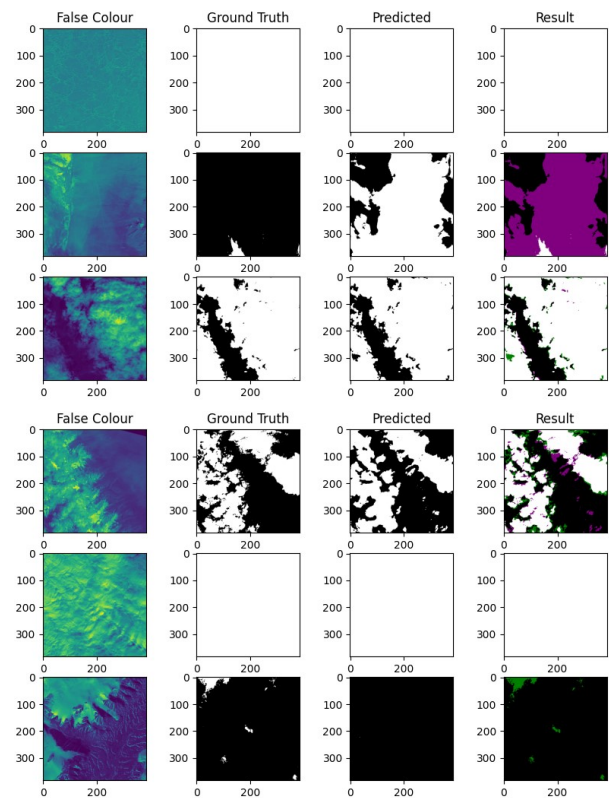


Figure 7 Result Table 2

The success measurements obtained from training and

evaluating the model are provided in Table 1.

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

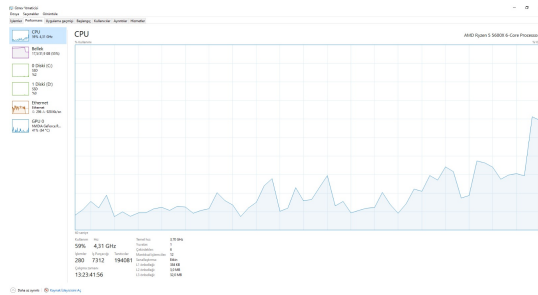
**Table 1** Performance Criteria - Value

## V. 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.

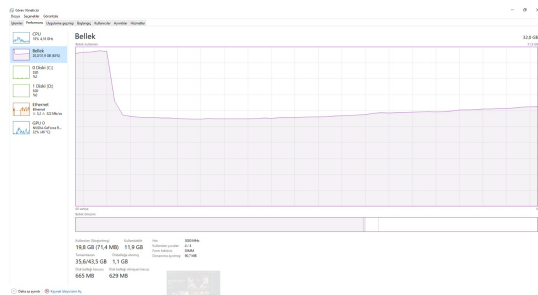
The checkpoint callback is utilized during training, and the model is trained on a train dataset for 20 epochs while utilizing a different validation dataset for validation. When the model gets its best IOU score on the validation data, the weights will be saved.

While training the model, the CPU performance is shown in below Figure 8.



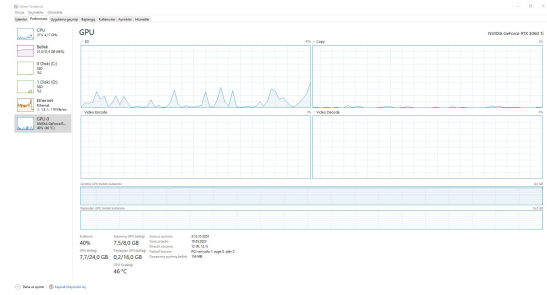
**Figure 8** CPU Usage

While training the model, the memory performance is shown in below Figure 9.



**Figure 9** Memory Usage

While training the model with, the GPU performance is shown in below Figure 10.



**Figure 10** GPU Usage

## VI. 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] 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.
- [2] 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.
- [3] B. Grabowski, M. Ziata, 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.
- [4] 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.
- [5] 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.