INTERNATIONAL JOURNAL OF INTEGRATED ENGINEERING
ISSN: 2229-838X e-ISSN: 2600-7916

IJIE

Vol. 16 No. 3 (2024) 145-156
https://publisher.uthm.edu.my/ojs/index.php/ijie

Atmospheric Cloud Image Detection with Convolutional Neural Network (CNN)

Pei Yong Lo[1], Sin Liang Lim[1]*

[1] *Faculty of Engineering*
Multimedia University, Persiaran Multimedia, Cyberjaya, Selangor, 63100, MALAYSIA

*Corresponding Author: lim.sin.liang@mmu.edu.my
DOI: https://doi.org/10.30880/ijie.2024.16.03.002

Article Info

Received: 18 December 2023
Accepted: 27 March 2024
Available online: 15 May 2024

Keywords

Atmospheric cloud image detection, convolutional neural network (CNN), transfer learning, U-Net, ResNet34, VGG16

Abstract

Cloud is an aerosol consisting of visible mass of miniature liquid droplets, frozen crystal, or other particles suspended in the atmosphere. The study of atmospheric clouds is crucial for us to better understand and predict the behaviors of clouds, which has implications for climate, weather, aviation safety, agriculture, and energy production. Convolutional neural network (CNN) method is applied to train an atmospheric cloud image detection model to identify the presence of cloud and classify them. Supervised learning method is applied to train the model such that the machine is given labeled cloud image dataset to learn how to classify and predict the presence of cloud. U-Net architecture is used to train the atmospheric cloud image detection model because the architecture has the highest performance in image segmentation especially object detection in satellite images. The 38-Cloud Dataset which is used to train the model, is obtained from Landsat 8 Earth observation satellite. The dataset is randomly divided into training set (75% of the total images) and validation set (25% of the total images). Following this, the dataset is preprocessed and transformed into tensors to train the model. The training has been carried out for 50 epochs. Apart from the U-Net architecture proposed, the architecture is further modified with ResNet34 and VGG16 and the performance of each model is studied. The recognition accuracy obtained for atmospheric cloud image detection trained with the dataset achieved 97%. With this accuracy, U-Net architecture can be justified as a powerful and suitable convolutional neural network in performing atmospheric cloud image detection.

1. Introduction

Clouds have an intriguing and crucial function in meteorology and Earth's atmospheric sciences. Atmospheric clouds are observable collections of water or ice crystals suspended in the atmosphere that produce a constantly shifting skycape. These dynamic structures have an essential role in predicting weather patterns and atmospheric conditions, in addition to enhancing beauty and intrigue in our daily lives. The complex relationship between temperature, humidity, and air velocity in the atmosphere leads to clouds. Warm air rises, expands, and cools, causing water vapor to condense into microscopic droplets or ice crystals. Then, these tiny particles assemble and combine to create clear cloud patterns. The water cycle and precipitation patterns are also impacted by clouds. They act as the growth areas for the ice crystals or water droplets that eventually turn into rain, snow or other types of precipitation. In addition, clouds may store atmospheric moisture, dispersing it over the globe and assisting in controlling the water cycle. Observing and understanding atmospheric clouds is crucial for weather forecasting, climate modeling, and studying atmospheric dynamics. To understand the behavior of clouds, follow their motions, and forecast their evolution, scientists use a combination of ground-based observations, satellite photography, and sophisticated computer models. The ability to forecast meteorological conditions such as the possibility of precipitation, the emergence of severe storms, and the evolution of atmospheric disturbances is made possible by this knowledge [1].

Deep learning is a branch of machine learning focus on training artificial neural networks to recognize patterns and extract pertinent information from massive datasets in order to learn and make decisions. It draws inspiration from the design and operation of the human brain, particularly from the intricately interwoven network of neurons that makes up the brain. Deep neural networks, which are neural networks with several hidden layers, are what distinguish deep learning. The network can learn hierarchical representations of the input data using these layers, where each layer captures increasingly sophisticated and abstract properties. Recently, convolution neural network (CNN) architecture, a type of deep learning architecture, has gained prominence in practical applications. The primary factor in its appeal is its ability to automatically extract and classify features, eliminating the need for manual feature extraction and selection [2].

Transfer learning is a powerful technique in deep learning where knowledge gained from training a model on one task is leveraged to improve the performance on a different but related task. By transferring learned representations from one domain to another, transfer learning enables models to benefit from pre-existing knowledge and overcome challenges of limited labeled data [3].

The dynamics of climate, weather patterns, and the energy balance of the planet all depend heavily on atmospheric clouds. For many applications, such as weather forecasting, climate modelling, and remote sensing, accurate and fast cloud identification is crucial. Manual cloud identification techniques, however, take a lot of time, are arbitrary, and frequently only cover a small area. Therefore, there is a need for an automated cloud detection system that can efficiently and accurately identify cloud regions in atmospheric imagery.

The objective of this study is to develop an automated atmospheric cloud detection system which is reliable, efficient, and accurate in cloud identification based on satellite or aerial imagery. The outcome of the study will have significant implications for weather analysis, climate modeling, and environmental monitoring which enable more precise predictions and a better understanding of atmospheric processes.

2. Literature Review

Some background studies on atmospheric cloud image detection done by other researchers is discussed in this section. In [4], a novel cloud detection method considering both spectral and spatial information is proposed for the multispectral remote sensing images. Firstly, the image is divided into spectrally homogeneous segments, namely superpixels, by Simple Linear Iterative Cluster (SLIC) method. A two-step classification strategy is used to divide these superpixels into cloud and non-cloud. The second step further classifies the potential cloud as cloud and non-cloud using the PCA Network (PCANet) combined with Support Vector Machine (SVM). Finally, a fully connected Conditional Random Field (CRF) model is employed to refine the cloud detection result and accurate cloud borders are obtained. The experimental images of this paper are obtained from the Landsat 8 Cloud Cover Assessment (L8 CCA) dataset which contains a total of 96 images with image size of approximately 6400 × 6400. The images are divided into two parts: 24 for training and remaining 72 for testing.

Table 1 *Right rate (RR), error rate (ER), and ratio of RR to ER (RER) of each method*

Methods	RR	ER	RER
Single-branch PCANet(9D)	0.9023	0.0829	10.8854
Single-branch PCANet(5D)	0.8010	0.1164	6.8828
Double-Branch PCANet	0.9138	0.0729	12.5289

The statistical result obtain by different network architectures are shown in Table 1. It can be seen that in two single-branch PCANets, the 5-dimensional PCANet has lower performance than the 9-dimensional one due to the incomplete spectral information input. In addition, when combining the two single-branch PCANets together, it is exactly the final double-branch PCANet which produces a better cloud detection result than both of the two single-branch PCANets.

The authors in [5] designed a novel Multilevel Convolutional Neural Network (MCNNs) architecture for multilevel cloud detection. The MCNN model used three different-sized patches (128 × 128, 64 × 64, and 32 × 32) as the input data to extract features from remote sensing imagery and the output is a 1024-dimensional vector, which is reshaped into four 16 × 16 channels to perform cloud detection.

Three categories of different spatial resolutions satellite imagery, GaoFen-1 (GF-1), GaoFen-2 (GF-2), and ZiYun-3 (ZY-3), were used for multilevel cloud detection. The ratio of the training set used are (10 ZY-3: 7 GF-1: 9 GF-2) and (2 ZY-3: 1 GF-1: 1 GF-2) for testing.

The impact of the superpixel segmentation on the performance of multilevel cloud detection has been studied by comparing the cloud detection accuracy using Adaptive-SLIC (A-SLIC) + MCNNs, SLIC + MCNNs, and Pixel +MCNNs methods. The statistical result obtained are shown in Table 2.

Table 2 *Statistics of different superpixel algorithms*

Parameter (%)	A-SLIC	SLIC	Pixel
OA	98.27	94.34	92.14
Kappa	92.34	88.31	87.31
EOA	97.36	92.13	90.38
EOE	0.94	2.61	4.24
ECE	1.70	4.26	5.38

**Overall Accuracy (OA), Edge Overall Accuracy (EOA), Edge Omission Error (EOE), and Edge Commission Error (ECE)

From Table 2, it is observed that the proposed method yields the best OA and EOA, and its EOE was lower than that of the other methods. This is because the superpixel segmentation method introduced the idea of affinity propagation, adaptively determining the number of superpixels and the center of clustering, the superpixel can contain the cloud boundary well.

The proposed method is then compared with other CNN approaches using spatial average pooling (SAP), max pooling (MP) and average pooling (AP). The statistical result obtained are shown in Table 3.

Table 3 *Statistics of different CNN architectures*

Parameter (%)	PA	SAP	MP	AP
OA	98.64	96.17	89.07	84.13
Kappa	95.27	88.34	89.34	87.81
EOA	97.37	94.01	87.34	82.41
EOE	1.02	2.28	4.81	8.17
ECE	1.61	3.71	7.85	9.42

**PA: Proposed Approach

Table 3 demonstrate that SAP+MCNNs is more effective at extracting cloud features and detects thin and thick cloud effectively on different underlying surfaces.

In addition, the proposed CDnet in [6] is a modified ResNet-50 network in order to extract features. The authors use three small filters with 3 × 3 receptive fields with stride 1. In addition, the incorporated three nonlinear 3 × 3 convolutional layers also make the modified ResNet-50 deeper and more discriminative than ResNet-50. Furthermore, ZY-3 Satellite Cloud Cover dataset that consists of 475 images are used in this paper. The data set is divided into three parts: 200 for training, 80 for testing and 195 for validation. The input size for CDnet is 321 × 321 pixel and hence around 46k of sub images are used at the training stages. The result of obtained by the CDnet are shown in Table 4.

Table 4 *Cloud extraction accuracy by CDnet*

Method	CDnet	SVM	PSPnet
OA	96.47	78.21	94.24
MIoU	91.70	66.79	88.37
Kappa	85.06	54.87	81.41
PA	89.75	91.77	86.67
UA	90.41	56.37	89.17

** Overall Accuracy (OA), Mean value across all the classes (MIoU), Product Accuracy (PA) and User Accuracy (UA)

From Table 4, it is observed that CDnet consistently outperforms compared to other models.

Luotamo et al. proposed a novel two-phase semantic segmentation framework for cloud detection from high-resolution optical remote sensing images, drawing on state-of-the-art CNN architectures [7]. To alleviate the drawbacks of either approach, a model architecture that combines coarse analysis of under sampled images with fine-grained analysis for a small number of patches selected by the coarse model are used. This allows fast and memory-efficient analysis of global features while retaining a capability for full-resolution segmentation.

478 Sentinel-2 (S2) Multispectral Instrument (MSI) LIC images from years 2016 to 2017 are used as data set for training and evaluating the model. 454 of the original 478 MSI images are used to train the model and the remaining 24 images were used for model testing evaluation. The modified architecture is applied to some of the model and the results are shown in Table 5 and Table 6.

Table 5 *Comparison of encoder backbones used with a semantic segmentation architecture fixed to U-NET*

Encoder	Acc	Pre	Rcal
EfficientNet	0.5515	0.5112	0.8308
ResNet-50	0.7158	0.7044	0.6422
VGG16	0.7299	0.7219	0.6122
InceptionV3	0.7336	0.6936	0.6907
SEResNeXt-50	0.7345	0.6898	0.6703

Table 6 *Comparison of encoder-decoder semantic segmentation architectures, with the encoder backbone fixed to SEResNeXt-50*

Segment	Acc	Pre	Rcal
U-Net	0.7345	0.6898	0.6703
LinkNet	0.6346	0.5562	0.6898
PSPNet	0.6666	0.5841	0.7096
FPN	0.737	0.7184	0.6426

**Accuracy (Acc), Precision (Pre), and Recall (Rcal)

Next, the segmentation architecture is modified, to evaluate with U-Net, Linknet, FPN, and PSPNet, by fixing the encoder to SEResNeXt-50 that initially performed best with U-Net. Table 5 indicates that in terms of accuracy, the best results for the considered encoder–decoder architectures are obtained with a combination of FPN and SEResNext-50.

Among the papers studied, the MCNN model shows the highest overall accuracy but the training time and the memory used to train the model are still unsatisfactory. Therefore, the aim of this paper is to propose a solution to build a more simplified cloud detection model which has higher accuracy, lower training time and lower memory usage.

3. Methodology

3.1 Problem Formulation

Atmospheric cloud detection plays a crucial role in various fields, including weather forecasting, climate research, and satellite imagery analysis. However, because they come in a variety of sizes, shapes, and textures, it can be difficult to correctly identify clouds in complex weather conditions. Transfer learning using the U-Net architecture is a technique that has attracted a lot of attention and produced outstanding outcomes in cloud identification.

Transfer learning is a machine learning technique where knowledge gained from solving one task is applied to a different but related task. Transfer learning enables us to make use of previously trained models on massive datasets, such as natural photos, and modify them to precisely recognize clouds in atmospheric imaging [3]. The U-Net architecture, originally introduced for biomedical image segmentation, has proven to be highly effective in various computer vision tasks. It is a CNN that consists of an encoder path and a decoder path, enabling it to capture both global and local context information. The encoder path captures high-level features through downsampling, while the decoder path reconstructs the output at the original resolution using upsampling.

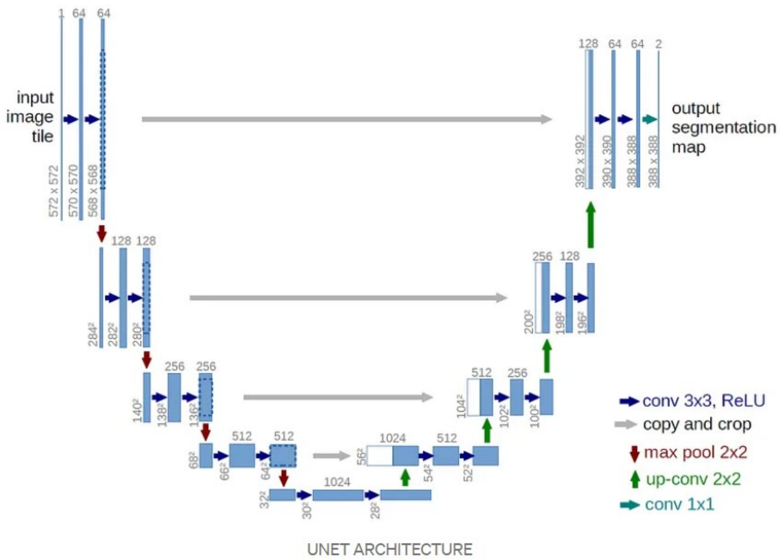

Fig. 1 *U-Net Architecture [8]*

As shown in Fig. 1, U-Net is a U-shaped encoder-decoder network architecture, which consists of four encoder blocks and four decoder blocks that are connected via a bridge. The encoder network (contracting path) halves the spatial dimensions and doubles the number of filters (feature channels) at each encoder block. Likewise, the decoder network doubles the spatial dimensions and halves the number of feature channels [8].

By employing transfer learning with U-Net, we can harness the power of pre-trained models like VGG16 or ResNet50, which have been trained on vast amounts of natural images, and fine-tune them for cloud detection. The lower layers of these models learn general features like edges and textures, which are valuable for cloud identification. The higher layers of the pre-trained model are retrained using transfer learning on a smaller dataset of annotated atmospheric photos. With the help of this fine-tuning, the model may focus on identifying features unique to clouds while still maintaining its broad comprehension of visual patterns from the pre-training stage. As a result, the model can now accurately separate clouds under various meteorological circumstances, such as those with diverse lighting, cloud thickness, and cloud kinds.

The combination of transfer learning and U-Net architecture provides a powerful and efficient solution for atmospheric cloud detection. By leveraging pre-existing knowledge and effectively capturing the unique features of clouds, this approach offers enhanced accuracy and robustness, which is vital for applications such as weather prediction, climate monitoring, and environmental analysis.

3.2 Data Preparation

For training and evaluating the model, we use a data set of 38 Landsat 8 scene images which are manually extracted pixel-level ground truths for cloud detection. 38-Cloud dataset is introduced in [9], and yet it was a further modification of the dataset in [10]. The entire images of these scenes are pre-processed and cropped into multiple 384×384 patches to be proper for deep learning-based semantic segmentation algorithms. There are 8400 patches of image for training. Each patch has 4 corresponding spectral channels which are Red (band 4), Green (band 3), Blue (band 2), and Near Infrared (band 5).

In order to prepare the dataset to be suited for the model, a dataset class that reads the Red, Green, Blue and Nir patches and stack them all into a tensor is created. The 4 channels are combined into a single tensor for a given index, and return a tuple (x,y) with a sample from the dataset where x is the 4 channels tensor and y is the ground truth mask. Besides, the ground truth values are modified to 0 and 1 from 0 to 255 to fit correctly in the model. Next, the data inside the dataset is split into train and validation sets. In addition, a data loader which is responsible for preparing the batches of data to pass them to the model is also created.

3.3 Optimizer and Hyperparameters

The choice of optimizer and hyperparameters are crucial since they can greatly impact the performance and convergence of the model. For training the U-Net model, the Adam optimization algorithm is chosen because it combines the advantage of adaptive learning rates and momentum. Based on previous gradients, it adjusts the learning rate for each parameter and applies momentum to smooth out updates. Moreover, the algorithm is straightforward to implement, has a faster running time, low memory requirements, and requires less tuning than any other optimization algorithm.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\left[\frac{\delta L}{\delta \omega_t}\right] = \beta_2 v_{t-1} + (1-\beta_2)\left[\frac{\delta L}{\delta \omega_t}\right]^2 \tag{1}$$

Equation (1) represents the working of Adam optimizer, where β1 and β2 represent the decay rate of the average gradients [11]. Next, the learning rate determines the step size taken during gradient descent to update the parameters of the model. The learning rate for the Adam optimizer to train the model is 0.01 to achieve a better a result.

Hyperparameters are parameters that are set prior to the training of a model and define its architecture and behaviour. They have a significant impact on the model's performance, convergence speed, and generalization ability. In the building of U-Net model, cross-entropy loss function is used for image segmentation. The function is well-suited for multi-class classification tasks, where each pixel can belong to one of several classes. It combines a SoftMax activation function with the negative log-likelihood loss, effectively measuring the dissimilarity between the predicted class probabilities and the true class labels. By minimizing the cross-entropy loss function during the training process, the U-Net model learns to assign higher probabilities to the correct classes for each pixel, thereby improving its ability to segment and classify objects accurately in the input images. The cross-entropy is defined as:

$$L_{CE} = -\sum_{i=1}^{n} t_i \log(p_i)\,, \qquad for\ n\ classes \tag{2}$$

where t_i is the truth label and p_i is the Softmax probability for the i^{th} class [12]. In addition, the batch size used to train the U-Net model is equal to 12. This means that 12 training examples are processed together in each iteration of the training algorithm.

3.4 Model Architecture

The Kaggle is one of the largest data science communities for open-source data and collaboration. Kaggle notebooks function the same way as Colab or Juptyer notebooks. The most frequently used packages are already preinstalled in the Kaggle environment [13]. In this paper, the U-net model is trained in the Kaggle notebooks.

Our proposed U-Net model consists of 3 contracting blocks and 3 expanding blocks. The first part of the U-Net model (contraction block) is a set of convolutions with pooling that downsizes the image resolution and creates additional layers to extract the features from the image.

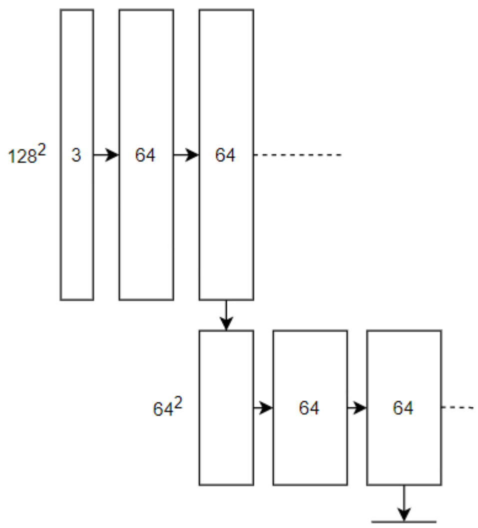

Fig. 2 *Contracting phase architecture*

Fig. 2 shows the first and second contraction blocks, each of them is composed of two convolutions with a 3 × 3 kernel and a max pooling with a stride of 2 × 2 that cuts in half the size of the input image. Hence, from here the image will be modified from 128 × 128 to 64 × 64. In addition, the number of layers increase from 3 to 64, indicating the design decision of the number of kernels used in the first convolution.

Next, the expanding block in the U-Net model complements the contracting block by progressively increasing the spatial dimensions while reducing the number of channels. It seeks to recover the compact representation produced by the contracting block from the high-resolution segmentation map. The main component inside the expanding block is the transpose convolutions which perform an inverse operation to the convolution by expanding the spatial dimensions while reducing the number of channels.

The combination of the contracting blocks and the expanding blocks is the final architecture of the U-Net model. The U-Net model proposed consists of 3 contracting blocks and 3 expanding blocks to achieve exactly the same resolutions and maintain two channels, as shown in Fig. 3.

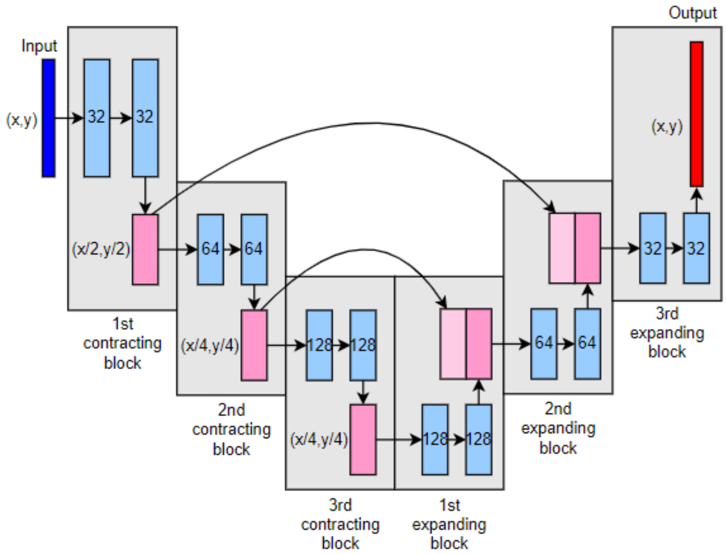

Fig. 3 Proposed U-Net model

In other words, a U-Net model that receives 4 channels as input and returns 2 channels is constructed as shown in the architecture in Fig. 3.

3.5 Model Architecture

Train function that receives the mode, the loss function, the optimizer, an accuracy function, the number of epochs, the train and the validation data loaders is created to train the U-Net model. The PyTorch Cross Entropy function is used as the loss function. In addition, the accuracy of the model is defined as equation (3) below:

$$Accuraccy = \frac{Number\ of\ matched\ pixels}{Total\ number\ of\ pixels\ in\ a\ batch} \quad (3)$$

Moreover, the Adam optimizer with a fixed learning rate of 0.01 is applied. Meanwhile, the number of epochs to train the U-Net model is 50 in order to increase the accuracy of the model.

In order to achieve a better performance for the U-Net model constructed, the model is further modified with other pre-trained model concept. In this work, the U-Net model is further modified with ResNet34 model and VGG16 model.

ResNet34 (Residual Network-34) is a popular CNN architecture that belongs to the ResNet family. This architecture was introduced in [14] back in 2016. ResNet34 has shown impressive performance in various computer vision tasks including image classification, object detection, and image segmentation. Its deep architecture and residual connections enable it to capture intricate patterns and features, making it highly effective in handling complex visual data. To modify the U-Net architecture with ResNet34, the initial contracting blocks are replaced with the layers of ResNet34 model, whereas the expansive path of the model remains.

VGG16 is a widely recognized CNN architecture that was developed by the Visual Geometry Group (VGG) at the University of Oxford. This architecture was introduced in [15]. The VGG16 architecture is significantly advanced in the field of deep learning and has emerged as a standard for image classification problems. It has sparked additional investigation and advancements in the construction of deep neural networks and served as the basis for many future CNN architectures. For this modification, the contracting path is replaced with VGG16 model.

4. Results and Discussions

In this paper, the proposed cloud detection model is implemented in Kaggle on a computer with AMD Ryzen 5 3500U CPU at 4GHz and 13GB RAM. The training process is enhanced with the GPU P100 (NVIDIA Tesla P100) which contains 16GB of memory to accelerate the computing power and improve the efficiency of the machine learning workflows. The loss plots of the training atmospheric cloud detection model for U-Net, modified U-Net with ResNet34, and U-Net modified with VGG16 are obtained and shown in Figs. 4, 5, and 6, respectively.

4.1 Loss Plots

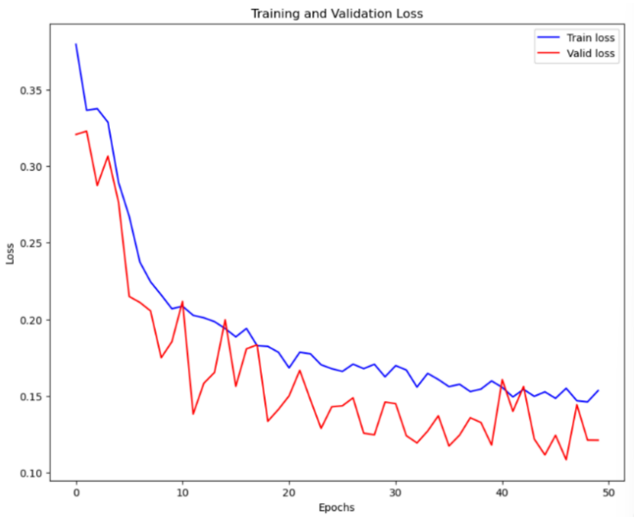

Fig. 4 *Loss plot of the U-Net model*

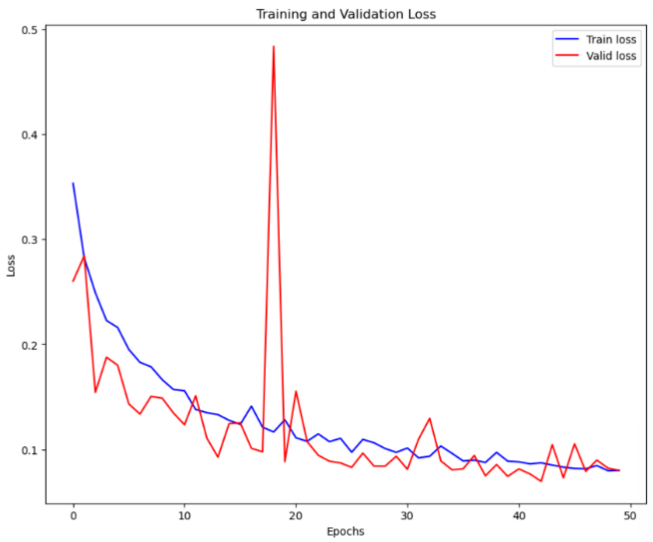

Fig. 5 *Loss plot of the U-Net model modified with ResNet34*

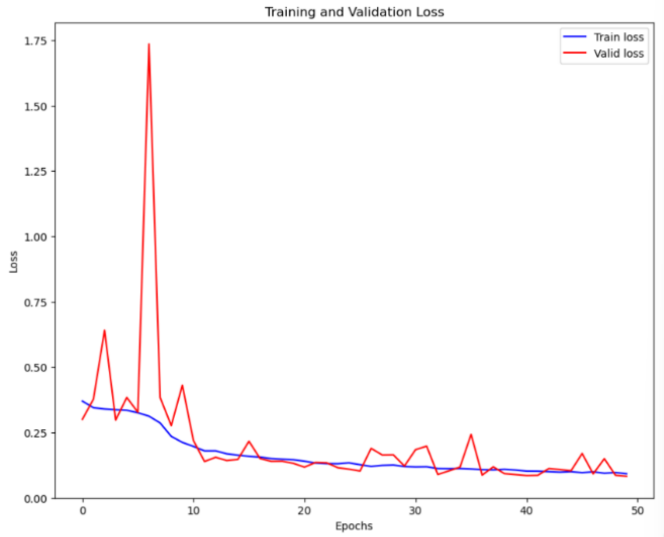

Fig. 6 *Loss plot of the U-Net model modified with the VGG16*

From the loss plots obtained, the blue lines which represent the training loss of the 3 models decrease until they reach a point of stability smoothly. Whereas, the red lines represent the validation loss fluctuate throughout the model training processes. Fig. 7 and Fig. 8 show the training loss plot and the validation loss plot of the 3 models respectively.

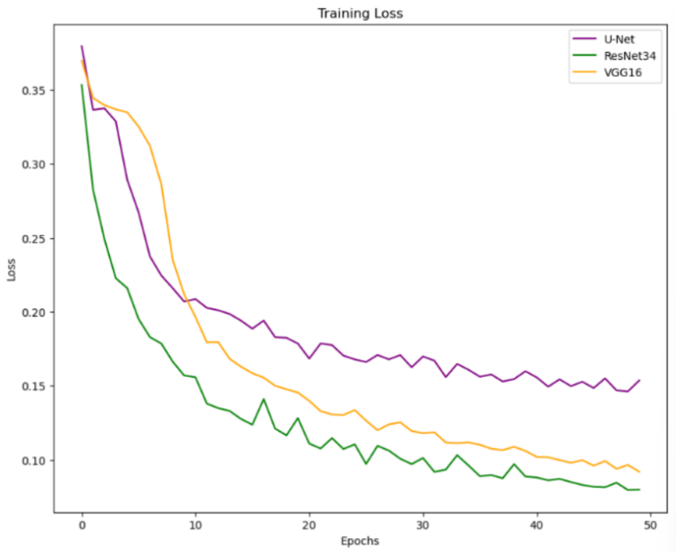

Fig. 7 *Training loss plot of the 3 different models*

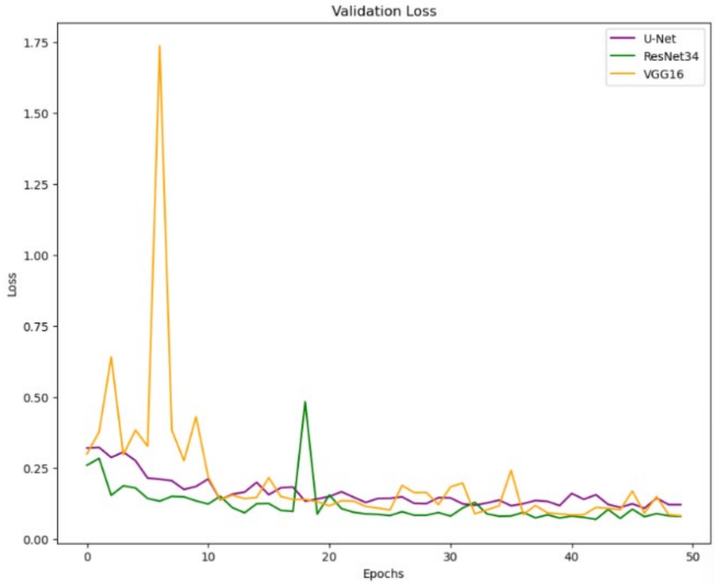

Fig. 8 *Validation loss plot of the 3 different models*

Table 7 *The validation loss of the models during training*

Model	Validation Loss
U-Net	0.1212
U-Net with ResNet34	0.0800
U-Net with VGG16	0.0822

From Table 7, the U-Net model modified with ResNet34 has the lowest average validation loss, followed by the U-Net model modified with VGG16 and the original U-Net model.

4.2 Validation Accuracy

The validation accuracy plots of the training atmospheric cloud detection model for U-Net, modified U-Net with ResNet34, and U-Net modified with VGG16 are generated and shown in Fig. 9.

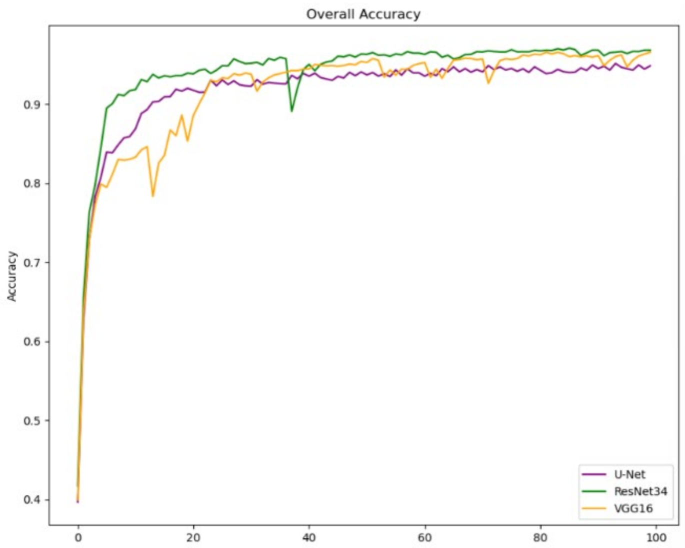

Fig. 9 *Overall accuracy plot of the 3 models*

From the accuracy plot obtained in Fig. 9, it is observed that the accuracy of training for the 3 models increase until they reach a point of stability smoothly.

Table 8 *The validation accuracy of the models*

Model	Validation Accuracy
U-Net	0.9523
U-Net with ResNet34	0.9695
U-Net with VGG16	0.9633

Based on Table 8, the U-Net model modified with ResNet34 has the highest validation accuracy which is close to 97%, whereas the U-Net model modified with VGG16 and the original U-Net model show the validation accuracy around 95%.

4.3 Training Efficiency

The model training efficiency is a critical aspect of machine learning, and the time used for model training is significant in determining the efficiency of the training process.

Table 9 *Training time of the models*

Model	Training Time (min)
U-Net	158.55
U-Net with ResNet34	117.93
U-Net with VGG16	135.10

From Table 9, the training time of the U-Net model modified with ResNet34 is the lowest, followed by the original U-Net model and the U-Net model modified with VGG16.

5. Conclusions

In conclusion, this project has successfully developed an atmospheric cloud image detection model. The model utilizes advanced techniques such as transfer learning with the U-Net architecture to accurately identify and segment clouds in complex atmospheric conditions. By leveraging pre-trained models and fine-tuning them on a specific dataset of annotated atmospheric images, the model demonstrates robust performance in detecting cloud-specific features.

The combination of transfer learning and the U-Net architecture allow the model to capture both local and global contextual information, enabling accurate cloud segmentation across various lighting conditions, cloud types, and cloud thickness. The constructed model has enormous potential for use in a variety of fields, such as weather forecasting, climatology, and satellite image analysis. For bettering weather forecasts, comprehending climatic patterns, and evaluating environmental effects, accurate cloud identification is essential. In order to improve the performance of the model, it is suggested to explore the integration of different data sources, such as satellite imagery, weather radar data, or ground-based sensors, to enhance cloud detection accuracy.

Acknowledgement

The authors would like to acknowledge Multimedia University (MMU) for funding the registration fee to MECON 2023 (Multimedia University Engineering Conference 2023) for this paper, which leads eventually to its publication in IJIE. The authors are also grateful to the reviewers for their constructive comments which strengthen the content of this paper.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: **study conception and design:** PY Lo and SL Lim; **data collection:** *PY Lo; analysis and interpretation of results: PY Lo and SL Lim;* ***draft manuscript preparation:*** *PY Lo and SL Lim. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] N. A. a. S. Administration. [Online]. Available: https://www.nasa.gov/pdf/135641main_clouds_trifold21.pdf.

[2] Shaheen, F., Verma, B., & Asafuddoula, M. (2016). Impact of Automatic Feature Extraction in Deep Learning Architecture. 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA).

[3] Dilmegani, C. (2023).Transfer Learning in 2023: What It Is & How It Works," 20 January 2023. [Online]. Available: https://research.aimultiple.com/transfer-learning/.

[4] Zi, Y., Xie, F., & Jiang, Z. (2018). A Cloud Detection Method for Landsat 8 Images Based on PCANet. Remote Sensing, 10, 6.

[5] Chen, Y., Fan, R., Bilal, M., Yang, X., Wang, J., & Li, W. (2018). Multilevel Cloud Detection for High-Resolution Remote Sensing Imagery Using Multiple Convolutional Neural Networks. ISPRS, 7, 5, pp. 181.

[6] Yang, J., Guo, J.,, Yue, H., Liu, Z., Hu, H., & Li, K. (2019). CDnet: CNN-Based Cloud Detection for Remote Sensing Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 57, 8, pp. 6195-6211.

[7] Luotamo, M., Metsämäki, S., & Klami, A. (2021). Multiscale Cloud Detection in Remote Sensing Images Using a Dual Convolutional Neural Network. *IEEE Transactions on Geoscience and Remote Sensing*, 59, 6, pp. 4972-4983.

[8] Tomar, N. (2021). What is UNET? 19 Jan 2021. [Online]. Available: https://medium.com/analytics-vidhya/what-is-unet-157314c87634.

[9] Mohajerani, S., & Saeedi, P. (2019). Cloud-Net: An End-To-End Cloud Detection Algorithm for Landsat 8 Imagery. *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 1029-1032.

[10] Mohajerani, S., Krammer, T. A., & Saeedi, P. (2018). A Cloud Detection Algorithm for Remote Sensing Images Using Fully Convolutional Neural Networks. 2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP), pp. 1-5.

[11] Gupta, A. (2017). A Comprehensive Guide on Optimizers in Deep Learning. 7 October 2017. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/.

[12] Koech, K. E. (2020). Cross-Entropy Loss Function. 3 October 2020. [Online]. Available: https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e.

[13] Fernades, K. (2021). Beginners Guide To Kaggle. 27 December 2021. [Online]. Available: *https://medium.com/mlearning-ai/beginners-guide-to-kaggle-3d1b07e8715e.*

[14] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[15] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd International Conference on Learning Representations (ICLR 2015).

[16] Jiang Y., Cheng, W., Gao, F., Zhang, S., Wang, S., Liu, C., and Liu, J. (2022). A Cloud Classification Method Based on a Convolutional Neural Network for FY-4A Satellites. Remote Sensing, 14, 10.

[17] Winthrop, R., Ziegler, L., Handa, R., & Fakoya, F. (2019). *How playful learning can help leapfrog progress in education*. Center for Universal Education at Brookings. https://www.brookings.edu/wp-content/uploads/2019/04/how_playful_learning_can_help_leapfrog_progress_in_education.pdf