International Journal on Advanced Science Engineering Information Technology

Vol.11 (2021) No. 6
ISSN: 2088-5334

Cloud Detection for Pleiades and SPOT 6/7 Imageries Using Modified K-means and Deep Learning

Yudhi Prabowo [a,*], Danang Surya Candra [a], Rachmat Maulana [a]

[a] *Remote Sensing Technology and Data Center, National Institute of Aeronautics and Space of Indonesia (LAPAN), Jakarta 13710, Indonesia*
Corresponding author: [*]yudhi.prabowo@lapan.go.id

***Abstract*— Cloud detection is one of the important stages in optical remote sensing activities as the cloud's existence interferes with the works. Many methods have been developed to detect the cloud, but it is still a few methods for high-resolution images, which mostly have limited multispectral bands. In this paper, a novel method of cloud detection for the images is proposed by integrating an unsupervised algorithm and deep learning. This method has three main steps: (1) pre-processing; (2) segmentation using modified K-means; and (3) cloud detection using CNN. In the segmentation step, an unsupervised algorithm, K-means is modified and used to divide pixels values into k clusters. Our modified K-means method can separate thin clouds from relative bright objects in gray clusters that will be grouped into potential cloud pixels. Afterward, a design of convolutional neural network (CNN) is used to extract the multi-scale features from each cluster and classify them into two classes: (1) cloud, which consists of thin cloud and thick cloud, and (2) non-cloud. The potential cloud area from the first step is used for guiding the result of CNN to provide accurate cloud areas. Several Pleiades and SPOT 6/7 images were used to test the reliability of the proposed method. As a result, our modified K-means has an improvement to increase the accuracy of the results. The results showed that the proposed method could detect cloud and non-cloud accurately and has the highest accuracy of the results compared to the other methods.**

***Keywords*— Cloud detection; high resolution; Pleiades; SPOT 6/7, deep learning; modified K-means; convolutional neural network.**

Manuscript received 25 Jan. 2021; revised 21 May 2021; accepted 29 Jun. 2021. Date of publication 31 Dec. 2021.

I. INTRODUCTION

High spatial resolution satellite imageries have been used for many applications such as land-use classification [1], building detection [2], forestry [3], marine [4], [5], and disaster monitoring [6]. However, the existence of a cloud on the image interferes with data information extraction. Moreover, cloud covers about two-thirds of earth's surface everyday [7] lead to difficulty in remote sensing activities. Therefore, a robust method of cloud detection is needed to develop to address this issue.

Many methods of cloud detection have been developed, but mostly the methods used for middle spatial resolution satellite imageries such as Landsat 8 and Sentinel-2 [7]–[11]. Therefore, it is challenging to develop a cloud detection method for high spatial resolution satellite images as it is still rare. Detecting cloud for high spatial resolution satellite imagery is quite challenging as most of the imagery data only have limited multispectral bands available on red, green, blue, and near-infrared bands besides the panchromatic band. It leads to difficulty in developing the method of cloud detection. In addition, compared to the medium-spatial-resolution satellite images, they provide thermal infrared band (TIRS), which usually plays a key role in detecting cloud, especially on threshold-based methods [12].

Generally, cloud detection methods can be categorized into two groups: (1) threshold-based and (2) machine-learning-based. In the threshold-based methods, a set of manual features pixel by pixel is extracted by this method. Afterward, it learns a binary classifier to identify the pixel as cloud or non-cloud [13]. It takes advantage of the difference between the reflectance of the cloud and the underlying surface. The reflectance of the surface, however, differs from the apparent reflectance because of atmospheric influence. The difference throughout these reflectivity makes it difficult to assess threshold effectively and precisely, generating rough cloud detection results based on the significance map and the suggested prime threshold despite its practical and quick calculation [14]. The temporal NDVI profile information was used [15] to detect cloud for AVHRR images. A simple two-step direct threshold technique was used [16] in detecting clouds. The first step is thresholding of near-infrared band reflectance, and the second step is thresholding of |NDVI|^b

R^(-2), where R is red band reflectance. Bi-spectral composite (BTC) threshold technique was proposed [17] to detect clouds for GOES imagery. This approach used a difference image of the 20-day composites of the 11- and the 11–3.9-μm channel for the bi-spectral cloud tests to demonstrate varying thresholds of clear-sky temporally. The speed of this process is fast, but this process depends on the sensors. In addition, it does not consider the structure and texture of the cloud.

On the other hand, machine-learning-based methods deliver more robustness in detecting clouds than threshold-based methods. SVM was used [18] to detect clouds for MODIS images. In this method, SVM was incorporated with the discriminant analysis (DA) to determine the cloud and clear sky subjectively and obtain typical cloud data without direct cloud detection. SVM was also proposed by Pengfei [19] in detecting clouds. The method divided images into small blocks and extracted characteristics of the brightness in the initial cloud detection and based on the features of the texture information of the sub-block image. Afterward, to detect cloud, the SVM uses the sub-block cloud image as learning samples. A cloud detection method was proposed [20] for Gao Fen-1 and Gao Fen-2 imageries using multi-feature fusion and machine learning. Various high-resolution satellite images which have near-infrared bands can be applied using this method. However, this method is unsuitable for images that have bare soil, desert, snow, and ice.

One of the popular (deep) machine learning approaches is convolution neural network (CNN) which is generally used for object detection [21], image classification [22], and segmentation [23] is currently used CNN for detecting cloud. Deep CNN was proposed [13] to detect multilevel clouds by using the improved simple linear iterative clustering (SLIC) in the image segmentation step. Afterward, they applied CNN in extracting the multi-scale features from each super-pixel and classify the super-pixel as thin cloud, thick cloud, or non-cloud. The proposed method can detect multilevel cloud detection accurately. On the other hand, [12] proposed the adaptive simple linear iterative clustering (A-SLIC) in the image segmentation step to produce good quality super-pixels. After that, they applied new multiple convolutional neural networks (MCNNs) to extract and identify multi-scale features from each super-pixel as a thick cloud, thin cloud, cloud shadow, or non-cloud. An algorithm of cloud and cloud shadow detection using CNN for WordView-2 and Sentinel-2 imageries developed by [24]. It eliminates the weakness of threshold-based that need to set a threshold which is usually complicated, and the spatial and spectral context of the multi-bands image are consider in this algorithm. The evaluation demonstrated that the algorithm has a higher accuracy compared to the other cloud detection algorithms.

Based on the current cloud detection methods overview, the machine-learning-based method remains challenging and has better expectations in detecting clouds for high-resolution satellite imagery. Therefore, this paper uses modified K-means clustering and deep CNN to detect clouds for high spatial resolution satellite imagery. We developed the modified K-means, an improved K-means clustering using the normalized blue indices to detect clouds better.

This paper aims to demonstrate the reliability of the proposed method. To achieve this aim, we use Pleiades and SPOT 6/7 images which are covered by various cloud types in the experiments. The images that have various land covers such as forest, open land, settlement, and water are selected to test the proposed method. In addition, we use visual and statistical assessments to evaluate the results. We compare the proposed method with SLIC + CNN as SLIC is a common approach in the segmentation step and is often used in the current studies. We also compare the proposed method with original K-means + CNN to demonstrate the improvement of our modified K-means in the segmentation step and the final results of the cloud detection process.

II. Material and Methods

A. Materials

In this paper, we used Pleiades-1A and Pleiades-1B satellite images from Airbus Defense and Space. Both are commercial twin satellites that operated at an altitude of 695 km, at a sun-synchronous orbit. The sensor has four multispectral bands, ranging from the visible to the near-infrared wave length and one panchromatic band (see Table 1).

TABLE I
THE SPECTRAL RANGE OF PLEIADES AND SPOT 6/7 IMAGES

Band	Pleiades		SPOT 6/7	
	Wavelength (μm)	Spatial resolution (m)	Wavelength (μm)	Spatial resolution (m)
Pan	0.48 – 0.83	0.5	0.45 – 0.75	1.5
Red	0.60 – 0.72	2	0.45 – 0.52	6
Green	0.49 – 0.61	2	0.53 – 0.59	6
Blue	0.43 – 0.55	2	0.62 – 0.69	6
NIR	0.75 – 0.95	2	0.76 – 0.89	6

The spatial resolution is 0.5 m for the panchromatic band and 2 m for multispectral bands. We also used SPOT 6/7 satellite images in the experiments to demonstrate that the proposed method can be used for many satellite images. We selected the several Pleiades and SPOT6/7 images with various cloud types and land-covers and have been radiometrically corrected (see Figure 1).

One of the important things to develop an accurate deep learning model is training with an abundance of image datasets [25]. It is because the classification with relatively few datasets may lead to overfitting [26]. The dataset used for training is collected from some Pleiades and SPOT 6/7 images intentionally devoted to training purposes. Beforehand, the Pleiades images have been resampled to 6-meter spatial resolution. The dataset is separated into two categories: positive images and negative images. The positive images contain cloud areas, whereas the negative images represent clear areas (non-cloud). Both positive images and negative images are cropped by 15x15 pixels based on visual interpretation. The input image in our CNN model contains three channels. These three channels are red, green, and blue, scaled from 12-bits to 8-bits unsigned integer. The positive images are mostly selected in the edge of the cloud area rather than the center of the cloud area. The total number of collected training dataset reaches to 15000 images with the ratio of positive images and negative images is 40% and 60%, respectively. We also applied data augmentation process by flipping the images horizontally and vertically and then

rotating by angle 90°, 180°, and 270° before the training process running.

B. Methods

The proposed method used to detect clouds for high spatial resolution imageries in this paper is object-based. It combines modified K-means, unsupervised classification, and deep learning, using CNN, supervised classification. In this method, the modified K-means and CNN are working separately. Afterward, the results of each process are combined to detect the cloud.

In the image clustering, modified K-means are used to detect the cloud. This process will be used to guide the results of CNN in the delineation of cloud areas. On the other hand, CNN in the proposed method has a feature extraction process that produces a features vector. These features vector will be treated as an input layer in the classification process. The result of CNN is a class prediction that will be gained by merging it with modified K-means. The aim is to improve the accuracy of the cloud detection results.

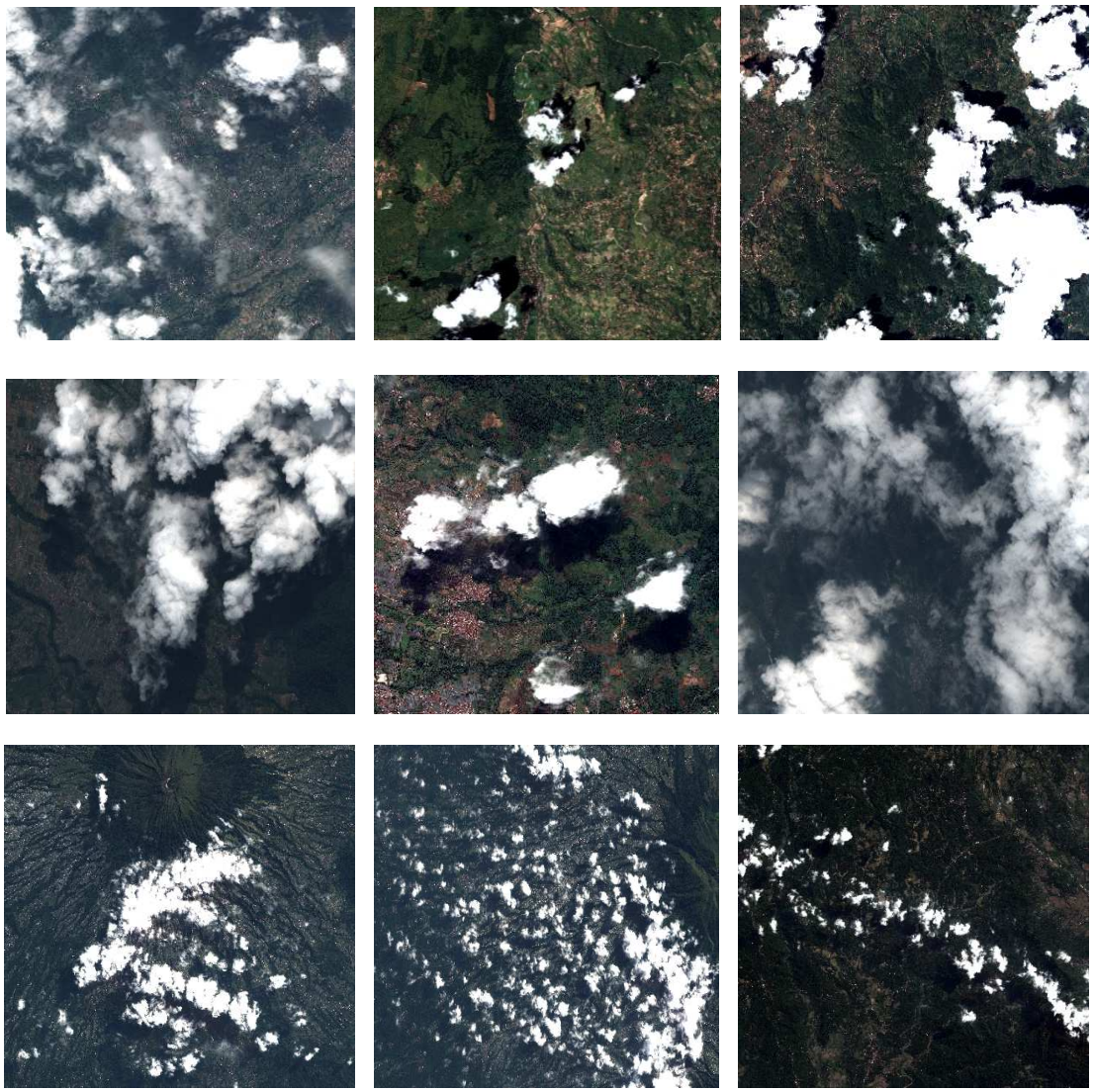


Fig. 1 Pleiades and SPOT 6/7 images were selected with various cloud types and land covers

There are three main steps of the proposed method: (1) pre-processing; (2) image clustering using modified K-means; and (3) cloud detection using CNN. Each step is detailed as follows.

1) Pre-processing: In this paper, a contrast enhancement of the image is conducted before the clustering and classification process. The aim of this process is to enhance the contrast of the image by scaling histogram input based on a particular range value. In this process, we use left and right 2% of percentile of the histogram as left and right borders of the scaling process. Afterward, each pixel is scaled based on the borders to range 0-255.

To make the computation process in the classification step faster, the image is downscaled to 6 meters using the averaging method. On the other hand, the original image which has not been downscaled is used in the clustering process. We need the original image with the original spatial resolution in this process as the border of the cloud mask is still depends on the results of this process.

2) Image Clustering using Modified K-means: K-means is a kind of unsupervised algorithm that has been widely used in many applications because of its simplicity [27]. The goal of K-means clustering in image processing is to divide pixels values into k clusters. Each pixel is set to the cluster

considering the nearest mean [28]. Firstly, each cluster should be defined as an initial random value as a cluster centroid that is distinct from each other. After defining the k centroids, the next step is to calculate the spectral similarity from each pixel to each centroid. So that each pixel will have k similarity distances as many as the number of the cluster that is desired, this study used Euclidean distance to measure the spectral similarity between each pixel and each centroid. Each pixel is grouped into a cluster where the similarity distance is a minimum. Pixels are finished to be divided, the value of each centroid needs to be updated by using the mean value of each cluster. The process of calculating spectral similarity is performed again. This iterative process stops when the centroids are no more changes.

In this study, the initial number of K-means clusters is set by three clusters (k=3). They are colored white, gray, and black under their brightness. The white cluster covers the thick cloud areas and the gray cluster covers thin clouds and some relative bright objects on the surface such as settlement, barren, grassland, etc. On the other hand, the black cluster covers the remaining relative dark objects such as shadow, forest, river, etc. However, the result of K-means still has a lot of errors so that it needs to be improved. We improved the result of K-means using the normalized blue indices (BI) as following:

$$BI = \frac{B}{R+G+B} \quad (1)$$

$$k' = \begin{cases} 1, & k = 1,2 \; and \; 0.3 < BI < 0.4 \\ 0, & k = 0,1 \; and \; BI \leq 0.3 \; and \; BI \geq 0.4 \end{cases} \quad (2)$$

R, G, and B are the value of red, green, and blue bands after pre-processing step. k = 0,1,2 represents black, gray, and white clusters from original K-means, respectively, and k' is our modified K-means output cluster. We performed pre-detection of potential cloud pixels by filtering the result of original K-means with the normalized blue indices. The potential cloud pixel is selected from two clusters, white and gray clusters, where the blue indices (BI) value is greater than 0.3 and less than 0.4. On the other hand, the non-cloud pixel is selected from gray and black clusters with the BI value is less than or equal to 0.3 and greater than or equal to 0.4. This modification of K-means suppresses the number of clusters from three clusters to be binary clusters labeled with potential cloud clusters and non-cloud clusters.

Our modified K-means method can separate thin clouds from relative bright objects in gray clusters so that some pixels in gray clusters that are considered thin clouds will be grouped into potential cloud pixels. In comparison, a relatively bright object such as barren, grassland, road, and settlements will be classified as non-cloud. The result of K-means clustering is used as a pre-detection of cloud areas. The potential cloud area is used for guiding the result of CNN to delineate the cloud areas accurately.

3) Cloud Detection using CNN: Deep neural network is one of the supervised classification methods that have become popular in the decades. A neuron is used to connect computational nodes to represent linear or non-linear functions that obtain an output as an object class based on the input. A set of layers containing input layers, hidden layers, and output layers build a neural network architecture called MLP (Multi-Layer Perceptron). The input layer is built from a series of neurons as a 1-dimensional array. The neurons in the input layer are individually mapped to each neuron in the hidden layer until the output layer. In remote sensing studies, the input layer can be concatenated from various parameters such as multi-channel intensity [29], spectral indices value [30], the statistical value of an image [31], etc.

- *Training Deep Learning Model:* The objective of training the neural network model is to tune its parameters that can minimize the classification error from the output layer. There are two main parts: (1) feed forwarding and (2) backpropagation. The feed-forward process is propagating the input to the output layer until resulting in the output prediction. The backpropagation is an iterative process using all training datasets, resulting in the network's updated parameters that fit the classification task. From the output prediction, the loss error can be calculated by using some loss function methods. The loss error is then propagated back to the network to update the weights and biases in every layer. The gradients from the loss function to all parameters are calculated by using the partial derivative function to update the network parameters. Once we have the gradients, we can update the parameters of weights and biases in the network. We used the binary cross entropy function to measure the loss error and the stochastic gradient descent method for the optimizer in this training process. The initial learning rate for the training is set to 0.0001.
- *Convolutional Neural Network:* Convolutional Neural Network fundamentally develops a neural network model that has revolutionized image detection and even images recognition [32]. It has a different architecture than a regular neural network. It simply consists of two consecutive processes: (1) feature extraction process and (2) classification process. In the first process, the input image is extracted with a convolutional and pooling operation series to produce a different form called feature maps. The feature map commonly has a smaller dimension than the input image, which summarizes detected features in the input image [33]. The convolution uses a kernel which is simply a small matrix of weights. The kernel slides over the image performing an element-wise multiplication with the pixel value of the image and then summing up the result into a single output pixel. The output of convolution will be passed through the activation function, making the output non-linear [34]. The activation functions usually used in this process are rectified linear unit (ReLU), sigmoid, tanh, and softmax [35]. After a convolution layer, it is common to add a pooling layer in between the feature extraction process. A pooling layer is used to reduce the size of the feature map without losing its significant information [36]. The final feature maps are then flattened from 3-dimensional into a 1-dimensional form which is called as features vector.

The second process of CNN is the classification process which uses a series of fully connected layers. In the fully connected layer, neurons fully connect to all neurons in the next layer. In principle, this part is the same as regular MLP

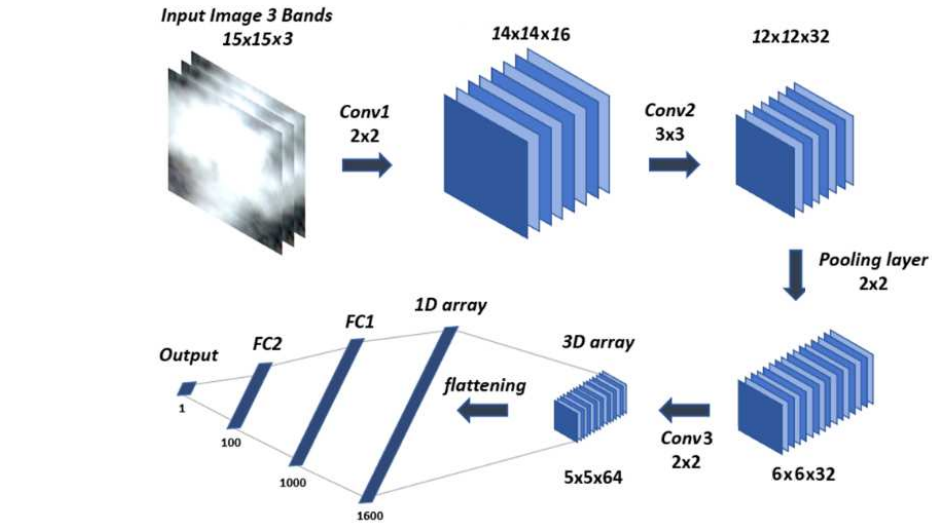
architecture, so that it can only accept data in 1-dimensional form. In this part, the features vector results from the convolutional process will be treated as an input layer. It will be fed to the fully connected layers to predict the class which has a higher confidence level.

In this paper, CNN is designed for binary classification that is classifying input images as cloud or non-cloud. Cloud class consists of thin and thick clouds. In practice, CNN uses a sliding window to scan the location of the cloud in the image. The sliding window in this paper has a fixed size which moves throughout the image from the top left to the bottom right. A window image is extracted at each window step and then classified to the cloud or non-cloud class. Our CNN architecture uses three convolutional layers, one pooling layer, and two hidden layers in fully connected layers. It ends up with one neuron in the output layer. Both the convolutional layers and the hidden layers, the outputs are passed through the ReLU activation function to convert them become non-linear. In addition, the activation function in the last layer of the network is a sigmoid, which the result will be in the confidence level of the cloud. The results with a confidence level higher than 0.5 are classified as cloud otherwise are classified as non-cloud.

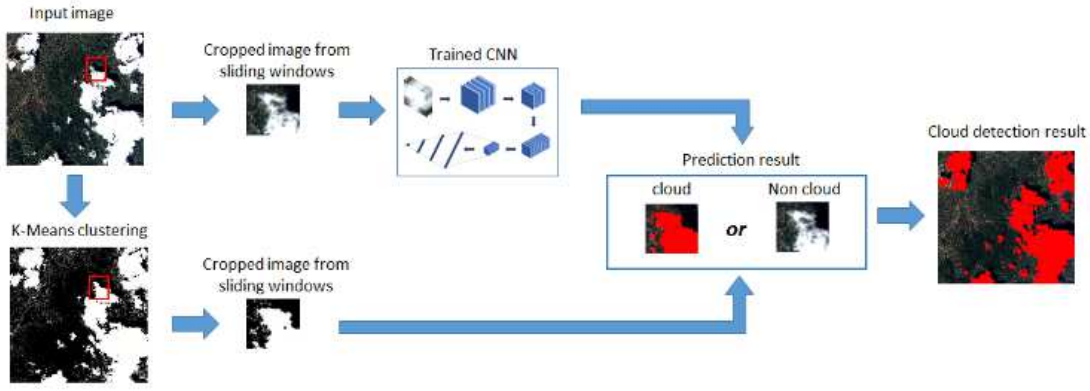
The input image of our CNN model is 15×15 pixels with three channels of RGB. The first layer is a convolutional process that uses kernels with the size of 2x2, and it generates 16 feature maps with the size of 14x14 pixels. The total number of trainable weights and biases in this layer is 208. Moreover, in the second layer, the convolutional with 3x3 kernel size generates 32 feature maps with 12x12 pixels. The total number of weights and biases is 4640. In the pooling layer, we used max-pooling method to down-sample 32 feature maps into 6x6 with the kernel size of 2x2 without padding. This pooling layer has no trainable parameters.

Lastly, the convolutional layer with 2x2 kernel size ends up the process of feature extraction. It generates 64 feature maps with the size of 5x5 and has 8256 trainable parameters of weights and biases. Then these feature maps are converted into a feature vector that consists of 1600 neurons as an input layer. Moreover, the feature vector passes through the fully connected layers to predict the output probabilities. Each neuron in the input layer connects to 1000 neurons in the first fully connected layer, and then in the next layer connects to 100 neurons. The total trainable parameter of weights and biases in fully connected layers is 1701201. The output layer has one neuron that indicates the input window image class, which belongs to cloud or non-cloud. The architecture of our CNN can be seen in Figures 2(a).

The delineation of the cloud mask area is obtained by merging the results of CNN classification and modified K-means. The result of CNN is only a class prediction for a subset window image in every step of the sliding window. It means that all pixels inside each window image have a one-class represented by their window class.



(a)



(b)

Fig. 2 (a) Architecture of our designed CNN and (b) Cloud detection processing flow in the proposed framework

However, when the window shifts to the next step, these associated pixels may have a different class from the current window class. The process of scanning images by sliding window records the position of windows, which are classified as a cloud. The boundary of cloud area is gained by reclassifying the pixels inside these selected windows with K-means clustering. The pixels which are outside the cloud cluster will not be classified as a cloud. The illustration of the cloud detection process can be seen in Figure 2(b).

4) Assessments: In this paper, visual and statistical assessments are applied to the resultant images to evaluate the reliability of the proposed method. In the statistical assessment, precision and recall were used in this paper as they are commonly used to assess the cases in machine learning.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (3)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (4)$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

Precision and recall are an assessment of classification results into positive and negative classes. In addition, a false positive is a predicted class that failed to meet the expected criteria, and a false negative is a true class that failed to be identified by a classifier. In a confusion matrix, the same terminology can be used to define precision and recall [37]. To redefine these measures in the confusion matrix, they can be categorized into two classes: (1) positive class and (2) negative class (see Table 2).

TABLE II
PRECISION AND RECALL FOR AN ASSESSMENT OF THE CLOUD DETECTION RESULTS

		Predicted	
		Negative (Non-cloud)	**Positive (Cloud)**
Actual	**Negative (Non-cloud)**	True Negative (Non-cloud)	False Positive (Non-cloud)
	Positive (Cloud)	False Negative (Cloud)	True Positive (Cloud)

The ratio of the total number of true positives to the total number of predicted positives is defined as precision (see Eq.3). On the other hand, the ratio of the total number of true positive to the total number of actual positives is defined as recall (see Eq.4). We used the F1 score to calculate the accuracy, which is the weighted average of Precision and Recall (see Eq.5) [38]. It is better than conventional accuracy, especially if there is an uneven class distribution. The flowchart of the cloud detection process using the proposed method can be seen in Fig. 3.

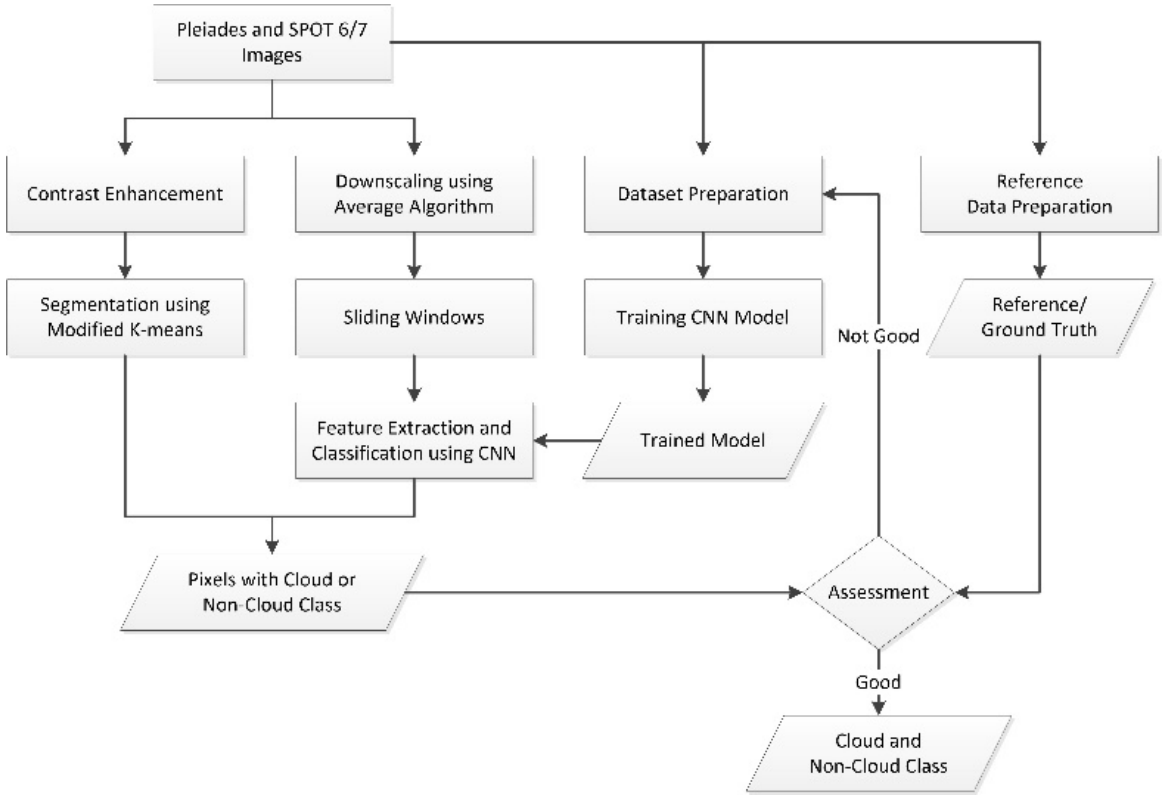


Fig. 3 Flowchart of cloud detection process using Modified K-means + CNN

III. RESULTS AND DISCUSSION

In the clustering step, the original K-means which clusters the images into three classes: thick cloud, thin cloud, and non-cloud overconfidence to classify open land and settlement, become cloud class (see the red circles in Figure 4). On the contrary, modified K-means with the normalized blue indices (BI) were used to cluster the images into two classes: cloud and non-cloud were successfully detected open land and settlement become a non-cloud class. Moreover, the original K-means did not detect thin clouds at some spots, whereas the modified K-means identified thin clouds accurately (see green circles in Figure 4). Thus, the modified K-means used in this

paper were done well and improved the accuracy results compared to the original K-means.

This paper tested the proposed method in detecting cloud for selected Pleiades and SPOT 6/7 images with various land covers such as settlement, open land, forest, mountainous area, and water. Cloud is not a white body. Therefore, land covers beneath the cloud influence its reflectance values, especially for thin clouds. We also evaluated the proposed method for the images that have a variety of cloud types to show the method's reliability.

We evaluated the resultant images by using visual assessment to investigate the proposed method of detecting cloud visually. This assessment helps us to know the ability of the proposed method quickly. The following assesses the proposed method in detecting clouds for thick clouds and thin clouds over heterogeneous land cover.

The most difficult method of detecting clouds is to separate clouds and bright objects, especially thick clouds, as they have a similar spectral response. This issue increases as the Pleiades and SPOT 6/7 have only four multispectral bands (blue, green, red, and near-infrared bands) with similar spectral responses to the cloud in bright objects such as open settlement land, road, etc. We can see in the resultant images in Figure 5(b,d,f,j) that the proposed method can detect thick clouds in settlement and open land areas. It can separate cloud to settlement, open land, and road properly. Thus, the proposed method works well to detect all thick clouds in this area accurately.

In the forest area (see Figure 5(d,f,h,l,n,p,r)), it can be seen that thick clouds can be detected accurately by using the proposed method. The spectral response of thick clouds is different from the forest. Therefore, there is no issue in detecting the cloud in this area. This also applies to dark objects such as water and mountainous areas (see Figure 5(h,n,p)). In these areas, the thick cloud is easy to detect as the spectral response of thick cloud is quite different to water and mountainous areas. As a result, we concluded that the proposed method could identify thick clouds properly in heterogeneous land covers. It is usually difficult to detect the borders of thick cloud region, but the proposed method works well to detect them accurately.

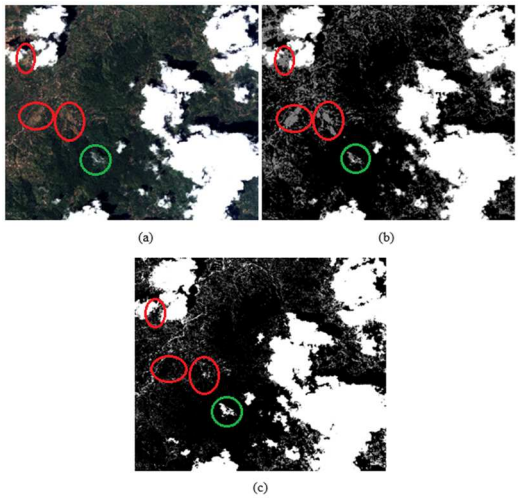
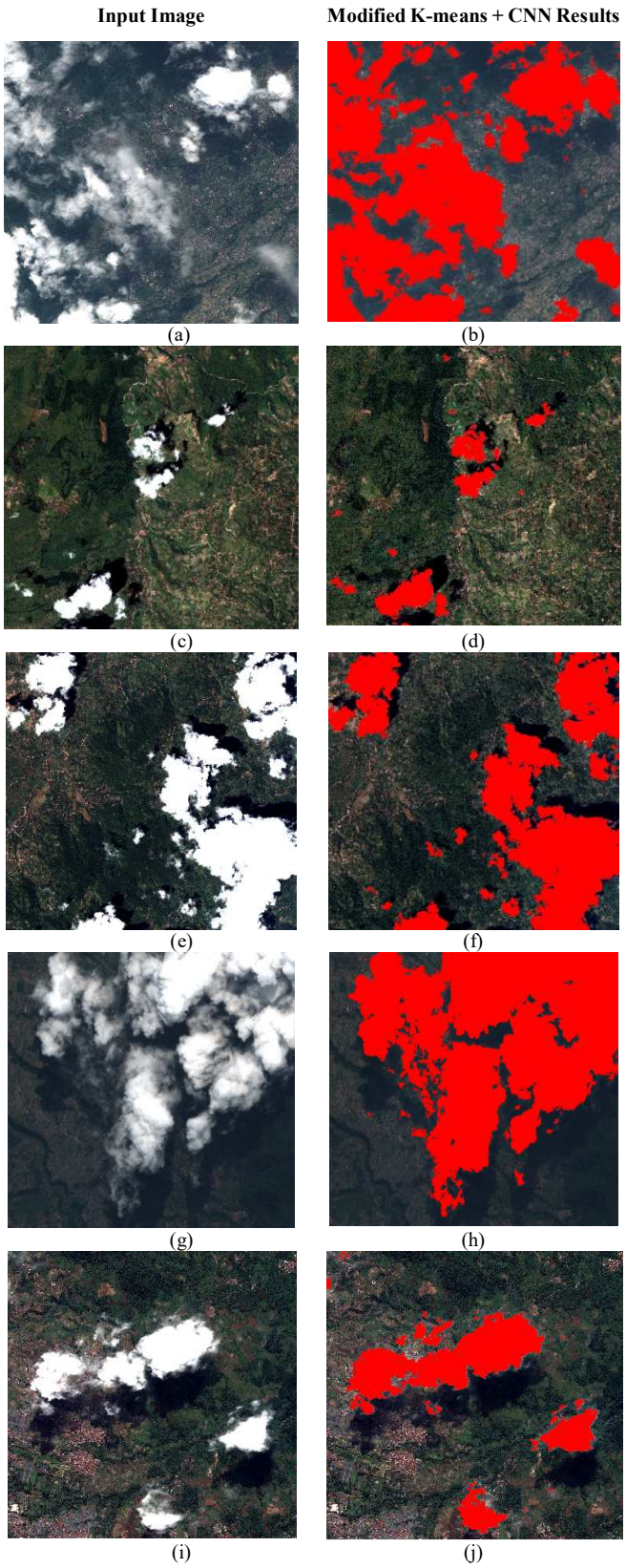
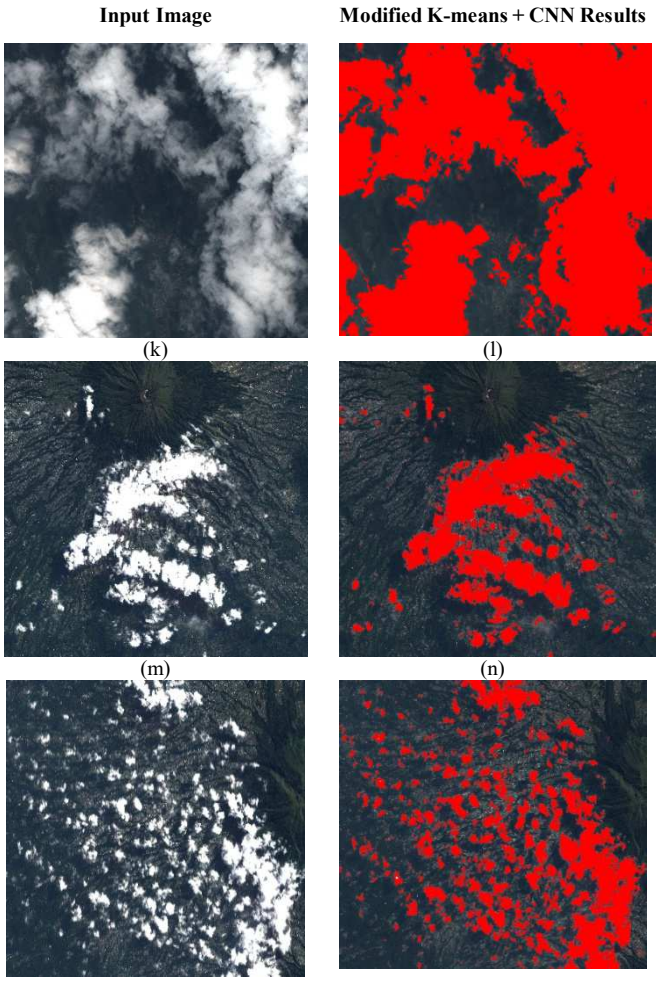


Fig. 4 (a) Pleiades image, (b) The result of K-means (3 classes), (c) The result of our modified K-means (2 classes)

It is difficult to detect a thin cloud as its transparency. It makes reflectance values of land covers beneath it influence their reflectance values. The difficulty of detecting thin cloud increases as Pleiades and SPOT 6/7 does not have a cirrus band. It is also difficult to assess the proposed method for thin clouds statistically as the border of the thin cloud is not distinct. Hence, visual assessment is very useful to evaluate the reliability of the proposed method in detecting thin clouds.





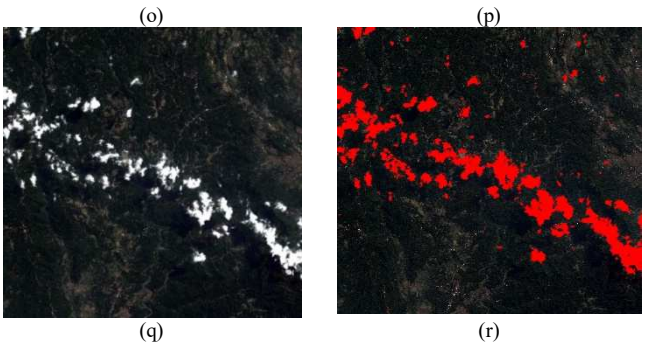


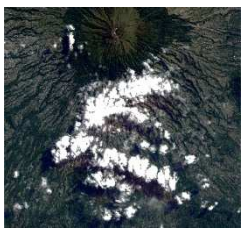
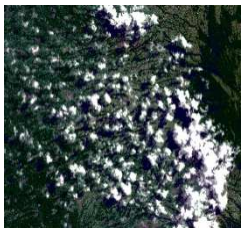
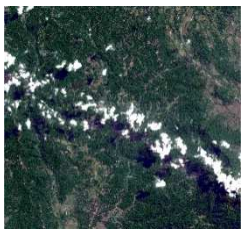
Fig. 5 The results of detecting cloud for Pleiades images by using the proposed method. Input images (left side) and resultant images (right side) of proposed method

We can see in Figure 5(a) that the image has a large amount of thin cloud. The resultant image in Figure 5(b) shows that the proposed method can detect them accurately. Moreover, the border of thin clouds, which are usually very thin (e.g., thin cloud in the bottom left of image) can also be detected by the proposed method. Figure 5(k) shows that a thin cloud spread in the whole image. It is not easy to detect very thin cloud in the settlement area. However, the proposed method works well to detect thin and very thin cloud accurately (see Figure 5(l)).

The proposed method works well in detecting thick and thin clouds for Pleiades images. In addition, it can be used to identify clouds over heterogeneous land covers. The lack of cirrus band does not make the proposed method fail to detect cloud, especially thin cloud.

TABLE III
PRECISION, RECALL, AND OF THE CLOUD DETECTION RESULTS

No	Image	SLIC+CNN	K-means + CNN	Modified K-means + CNN
1		Precision = 0.97 Recall = 0.91 Overall accuracy = 0.96 Kappa score = 0.92 F1 score = 0.94	Precision = 0.99 Recall = 0.87 Overall accuracy = 0.96 Kappa score = 0.90 F1 score = 0.93	Precision = 0.94 Recall = 0.95 Overall accuracy = 0.97 Kappa score = 0.93 F1 score = 0.95
2		Precision = 0.94 Recall = 0.68 Overall accuracy = 0.98 Kappa score = 0.78 F1 score = 0.79	Precision = 0.94 Recall = 0.79 Overall accuracy = 0.98 Kappa score = 0.85 F1 score = 0.86	Precision = 0.86 Recall = 0.90 Overall accuracy = 0.98 Kappa score = 0.88 F1 score = 0.88
3		Precision = 0.77 Recall = 0.84 Overall accuracy = 0.82 Kappa score = 0.64 F1 score = 0.80	Precision = 0.99 Recall = 0.46 Overall accuracy = 0.76 Kappa score = 0.48 F1 score = 0.63	Precision = 0.91 Recall = 0.92 Overall accuracy = 0.92 Kappa score = 0.85 F1 score = 0.91

4		Precision = 0.93 Recall = 0.92 Overall accuracy = 0.93 Kappa score = 0.86 F1 score = 0.92	Precision = 0.99 Recall = 0.59 Overall accuracy = 0.80 Kappa score = 0.60 F1 score = 0.74	Precision = 0.97 Recall = 0.94 Overall accuracy = 0.96 Kappa score = 0.92 F1 score = 0.96
5		Precision = 0.92 Recall = 0.64 Overall accuracy = 0.95 Kappa score = 0.73 F1 score = 0.75	Precision = 0.99 Recall = 0.55 Overall accuracy = 0.95 Kappa score = 0.69 F1 score = 0.71	Precision = 0.95 Recall = 0.88 Overall accuracy = 0.98 Kappa score = 0.90 F1 score = 0.91
6		Precision = 0.99 Recall = 0.82 Overall accuracy = 0.87 Kappa score = 0.73 F1 score = 0.90	Precision = 0.99 Recall = 0.64 Overall accuracy = 0.75 Kappa score = 0.52 F1 score = 0.78	Precision = 0.98 Recall = 0.92 Overall accuracy = 0.93 Kappa score = 0.86 F1 score = 0.95
7		Precision = 0.65 Recall = 0.83 Overall accuracy = 0.91 Kappa score = 0.67 F1 score = 0.73	Precision = 0.98 Recall = 0.65 Overall accuracy = 0.94 Kappa score = 0.75 F1 score = 0.78	Precision = 0.90 Recall = 0.88 Overall accuracy = 0.96 Kappa score = 0.87 F1 score = 0.89
8		Precision = 0.82 Recall = 0.69 Overall accuracy = 0.89 Kappa score = 0.68 F1 score = 0.75	Precision = 0.99 Recall = 0.55 Overall accuracy = 0.89 Kappa score = 0.65 F1 score = 0.71	Precision = 0.95 Recall = 0.82 Overall accuracy = 0.94 Kappa score = 0.85 F1 score = 0.88
9		Precision = 0.97 Recall = 0.54 Overall accuracy = 0.96 Kappa score = 0.68 F1 score = 0.70	Precision = 0.98 Recall = 0.57 Overall accuracy = 0.96 Kappa score = 0.71 F1 score = 0.72	Precision = 0.83 Recall = 0.91 Overall accuracy = 0.97 Kappa score = 0.85 F1 score = 0.87
	Total	Precision = 0.89 Recall = 0.83 Overall accuracy = 0.91 Kappa score = 0.80 F1 score = 0.86	Precision = 0.99 Recall = 0.62 Overall accuracy = 0.88 Kappa score = 0.69 F1 score = 0.76	Precision = 0.95 Recall = 0.91 Overall accuracy = 0.96 Kappa score = 0.90 F1 score = 0.93

To show the reliability of the proposed method, statistical assessment was used. In this assessment, we used Precision, Recall and F1. The matrix described how cloud and non-cloud classes were classified in comparison to the true ordering. To evaluate the accuracy of the proposed method, we need a reference of the cloud for reference data. The references of cloud polygon were built by using manual digitizing on-screen. To show the improvement of our proposed method, we compared the proposed method to other approaches: SLIC + CNN and K-means + CNN. Many studies use simple linear iterative clustering (SLIC) to cluster a particular object such as a cloud.

The SLIC approach is used to segment the image into good-quality super-pixels that are roughly equally sized. SLIC divides the image into several equal-size grids to create initial cluster centers. In searching space, it has a limitation to a local region. Therefore, the smooth thick cloud region will be over-segmented (e.g., see figure in Table 5(3)), and it obtained the precision of SLIC + CNN, in this case, is quite low at 0.77.

On the contrary, the original K-means failed to detect this cloud, and it makes the recall of K-means + CNN is quite low at 0.46. On the other hand, our proposed method can handle this issue in the same cases to detect the smooth thick cloud accurately with precision and 0.91 and recall is 0.91 and 0.92, respectively.

The intensity of forest tends to low intensity as it has dense tree canopies. This low intensity of forest influences the reflectance of thin cloud. It makes thin cloud more difficult to identify and commonly makes any classifiers failed to identify this cloud. It occurred in figures in the Table 5(8,9) which these figures have thin cloud over forest area. These circumstances make the SLIC + CNN and K-means + CNN obtained low recall values for Figure 5(8) at 0.69 and 0.55, respectively and for figure in the Table 5(9) at 0.54 and 0.57, respectively.

However, the modified K-means + CNN has a bit higher recall value than these two classifiers. The recall value for figures in the Table 5(8) and 5(9) of the proposed method is 0.82 and 0.91, respectively. For nine test images, the total precision, recall and F1 score of the SLIC + CNN is 0.89, 0.83, and 0.86, respectively. On the other hand, the total precision, recall and F1 score of K-means + CNN is 0.99, 0.62, and 0.76, respectively. Compared to these two classifiers, the precision, recall and F1 score of modified K-means + CNN has a bit higher, i.e., 0.95, 0.91, and 0.93, respectively. These accuracy results showed that the proposed method works well and has improvement in terms of detecting cloud for high spatial resolution imageries.

IV. Conclusions

Pleiades and SPOT 6/7, high spatial resolution imageries, is generally difficult in detecting cloud as it has limited bands availability including visible and near-infrared spectral bands. This paper proposes a novel method of cloud detection to detect cloud for the images by integrating unsupervised algorithm and deep learning. K-means, an unsupervised algorithm, is modified to segment the images into good quality clusters. In addition, the deep CNN is designed to extract the multi-scale features from each cluster and classify them as cloud and non-cloud.

To provide accurate cloud class, the potential cloud from the segmentation step is used to guide the CNN in predicting the class. The nine selected Pleiades and SPOT 6/7 images with various land covers and cloud types were used to test the proposed method. As a result, the original K-means failed to identify thin clouds at some spots whereas the modified K-means successfully detected them. Hence, compared to the original K-means, the modified K-means proposed in this paper improve the accuracy of the results.

Moreover, the SLIC over segmented in detecting the smooth thick cloud region, so that the SLIC + CNN has low precision value. On the other hand, the original K-means failed to detect this cloud, so that the K-means + CNN approach has low recall value in this case. In the same case, on the other hand, the modified K-means, which is proposed in this paper, can handle this issue so that the precision and recall values are quite high. In the overall tests, compared to SLIC + CNN and K-means + CNN, the proposed method has the highest precision, recall, and F1 score. The experiment results showed that the proposed method could accurately detect clouds for Pleiades and SPOT 6/7 images. The limitation of the proposed method is still using manual training sample selection. Therefore, the approach for automatic training sample selection will be considered in further research.

Acknowledgment

The authors would like to thank and appreciate the anonymous reviewers. The authors also would like to thank Remote Sensing Technology and Data Center, National Institute of Aeronautics and Space of Indonesian (LAPAN) for providing Pleiades and SPOT 6/7 images.

References

[1] W. Zhang, P. Tang, and L. Zhao, "Remote Sensing Image Scene Classification Using CNN-CapsNet," *Remote Sens.*, vol. 11, no. 5, 2019, doi: 10.3390/rs11050494.

[2] Y. You *et al.*, "Building Detection from VHR Remote Sensing Imagery Based on the Morphological Building Index," *Remote Sens.*, vol. 10, no. 8, pp. 1–22, 2018, doi: 10.3390/rs10081287.

[3] L. Piermattei *et al.*, "Impact of the Acquisition Geometry of Very High-Resolution Pl é iades Imagery on the Accuracy of Canopy Height Models over Forested Alpine Regions," *Remote Sens.*, vol. 10, no. 10, 2018, doi: 10.3390/rs10101542.

[4] G. Marmorino and W. Chen, "Use of WorldView-2 Along-Track Stereo Imagery to Probe a Baltic Sea Algal Spiral," *Remote Sens.*, vol. 11, no. 7, pp. 1–9, 2019, doi: 10.3390/rs11070865.

[5] J. Marcello, F. Eugenio, J. Martin, and F. Marques, "Seabed Mapping in Coastal Shallow Waters Using High Resolution Multispectral and Hyperspectral Imagery," *Remote Sens.*, vol. 10, no. 8, 2018, doi: 10.3390/rs10081208.

[6] D. Amitrano *et al.*, "Long-Term Satellite Monitoring of the Slumgullion Landslide Using Space-Borne Synthetic Aperture Radar Sub-Pixel Offset Tracking," *Remote Sens.*, vol. 11, no. 3, pp. 1–13, 2019, doi: 10.3390/rs11030369.

[7] D. S. Candra, S. Phinn, and P. Scarth, "Automated cloud and cloud-shadow masking for Landsat 8 using multitemporal images in a variety of environments," *Remote Sens.*, vol. 11, no. 17, 2019, doi: 10.3390/rs11172060.

[8] D. S. Candra, S. Phinn, and P. Scarth, "Cloud and cloud shadow masking for Sentinel-2 using multitemporal images in global area," *Int. J. Remote Sens.*, vol. 41, no. 8, p. 2020, 2020, doi: https://doi.org/10.1080/01431161.2019.1697006.

[9] Q. Shi, H. Binbin, Z. Zhe, L. Zhanmang, and Q. Xingwen, "Improving Fmask cloud and cloud shadow detection in mountainous area for Landsats 4-8 images," *Remote Sens. Environ.*, vol. 199, no. September 2017, pp. 107–119, 2017.

[10] D. Frantz, E. Haß, A. Uhl, J. Stoffels, and J. Hill, "Improvement of the Fmask algorithm for Sentinel-2 images: Separating clouds from bright surfaces based on parallax effects," *Remote Sens. Environ.*, vol. 215, no. April 2017, pp. 471–481, 2018, doi: 10.1016/j.rse.2018.04.046.

[11] G. Nafiseh and A. Mehdi, "Introducing two Random Forest based methods for cloud detection in remote sensing images," *Adv. Sp. Res.*, vol. 62, no. July 2018, pp. 288–303, 2018, doi: https://doi.org/10.1016/j.asr.2018.04.030.

[12] Y. Chen, R. Fan, M. Bilal, X. Yang, J. Wang, and W. Li, "Multilevel cloud detection for high-resolution remote sensing imagery using multiple convolutional neural networks," *ISPRS Int. J. Geo-Information*, vol. 7, no. 5, 2018, doi: 10.3390/ijgi7050181.

[13] X. Fengyin, S. Mengyun, S. Zhenwei, Y. Jihao, and Z. Danpei, "Multilevel Cloud Detection in Remote Sensing Images Based on Deep Learning," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 10, no. 8, pp. 1–10, 2017.

[14] L. Sun *et al.*, "A cloud detection algorithm-generating method for remote sensing data at visible to short-wave infrared wavelengths," *ISPRS J. Photogramm. Remote Sens.*, vol. 124, pp. 70–88, 2017, doi: 10.1016/j.isprsjprs.2016.12.005.

[15] J. Cihlar and J. Howarth, "Detection and removal of cloud contamination from AVHRR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 3, pp. 583–589, 1994, doi: 10.1109/36.297976.

[16] B. Lee, L. Di Girolamo, G. Zhao, and Y. Zhan, "Three-dimensional cloud volume reconstruction from the Multi-Angle Imaging

SpectroRadiometer," *Remote Sens.*, vol. 10, no. 11, 2018, doi: 10.3390/rs10111858.
[17] J. Gary J, H. Stephanie L, and L. Frank J, "Spatial and Temporal Varying Thresholds for Cloud Detection in GOES Imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1705–1717, 2008.
[18] I. Haruma, O. Yu, M. Keitaro, M. Keigo, and Y. N. Takashi, "Development of a support vector machine based cloud detection method for MODIS with the adjustability to various conditions," *Remote Sens. Environ.*, vol. 205, no. February 2018, pp. 390–407, 2018, doi: https://doi.org/10.1016/j.neucom.2014.09.102.
[19] L. Pengfei, D. Limin, X. Huachao, and X. Mingliang, "A cloud image detection method based on SVM vector machine," *Neurocomputing*, vol. 169, no. December 2015, pp. 34–42, 2015.
[20] T. Bai, D. Li, K. Sun, Y. Chen, and W. Li, "Cloud detection for high-resolution satellite imagery using machine learning and multi-feature fusion," *Remote Sens.*, vol. 8, no. 9, pp. 1–21, 2016, doi: 10.3390/rs8090715.
[21] W. Li, R. Dong, H. Fu, and L. Yu, "Large-scale oil palm tree detection from high-resolution satellite images using two-stage convolutional neural networks," *Remote Sens.*, vol. 11, no. 1, 2019, doi: 10.3390/rs11010011.
[22] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep & Dense convolutional neural network for hyperspectral image classification," *Remote Sens.*, vol. 10, no. 9, pp. 1–28, 2018, doi: 10.3390/rs10091454.
[23] K. A. Korznikov, D. E. Kislov, J. Altman, J. Doležal, A. S. Vozmishcheva, and P. V. Krestov, "Using u-net-like deep convolutional neural networks for precise tree recognition in very high resolution rgb (Red, green, blue) satellite images," *Forests*, vol. 12, no. 1, pp. 1–17, 2021, doi: 10.3390/f12010066.
[24] M. Segal-Rozenhaimer, A. Li, K. Das, and V. Chirayath, "Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks (CNN)," *Remote Sens. Environ.*, vol. 237, 2020, doi: 10.1016/j.rse.2019.111446.
[25] S. Almabdy and L. Elrefaei, "Deep convolutional neural network-based approaches for face recognition," *Appl. Sci.*, vol. 9, no. 20, 2019, doi: 10.3390/app9204397.
[26] K. Kamycki, T. Kapuscinski, and M. Oszust, "Data augmentation with suboptimal warping for time-series classification," *Sensors*, vol. 20, no. 1, 2020, doi: 10.3390/s20010098.
[27] J. Qi, Y. Yu, L. Wang, J. Liu, and Y. Wang, "An effective and efficient hierarchical K-means clustering algorithm," *Int. J. Distrib. Sens. Networks*, vol. 13, no. 8, pp. 1–17, 2017, doi: 10.1177/1550147717728627.
[28] S. Mehta, X. Shen, J. Gou, and D. Niu, "A new nearest centroid neighbor classifier based on k local means using harmonic mean distance," *Information*, vol. 9, no. 9, 2018, doi: 10.3390/info9090234.
[29] L. Hu, M. Qin, F. Zhang, Z. Du, and R. Liu, "RSCNN: A cnn-based method to enhance low-light remote-sensing images," *Remote Sens.*, vol. 13, no. 1, pp. 1–13, 2021, doi: 10.3390/rs13010062.
[30] R. Ba, W. Song, X. Li, Z. Xie, and S. Lo, "Integration of multiple spectral indices and a neural network for burned area mapping based on MODIS data," *Remote Sens.*, vol. 11, no. 3, 2019, doi: 10.3390/rs11030326.
[31] J. I. Hwang and H. S. Jung, "Automatic ship detection using the artificial neural network and support vector machine from X-Band Sar satellite images," *Remote Sens.*, vol. 10, no. 11, 2018, doi: 10.3390/rs10111799.
[32] R. Waseem and W. Zenghui, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Neural Comput.*, vol. 29, no. 9, pp. 1–98, 2017, doi: 10.1162/NECO.
[33] W. Li, H. Fu, L. Yu, and A. Cracknell, "Deep learning based oil palm tree detection and counting for high-resolution remote sensing images," *Remote Sens.*, vol. 9, no. 1, 2017, doi: 10.3390/rs9010022.
[34] C. Yu, W. Duo, Z. Pan, and Z. Tao, "Model compression and acceleration for deep neural networks: The principles, progress, and challenges," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 126–136, 2018, doi: 10.1109/MSP.2017.2765695.
[35] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Appl. Sci.*, vol. 10, no. 5, 2020, doi: 10.3390/app10051897.
[36] Y. Hu, Q. Zhang, Y. Zhang, and H. Yan, "A deep convolution neural network method for land cover mapping: A case study of Qinhuangdao, China," *Remote Sens.*, vol. 10, no. 12, 2018, doi: 10.3390/rs10122053.
[37] A. Elmes *et al.*, "Accounting for training data error in machine learning applied to earth observations," *Remote Sens.*, vol. 12, no. 6, pp. 86–88, 2020, doi: 10.3390/rs12061034.
[38] F. Mohammadimanesh, B. Salehi, M. Mahdianpari, E. Gill, and M. Molinier, "A new fully convolutional neural network for semantic segmentation of polarimetric SAR imagery in complex land cover ecosystem," *ISPRS J. Photogramm. Remote Sens.*, vol. 151, no. April, pp. 223–236, 2019, doi: 10.1016/j.isprsjprs.2019.03.015.