remote sensing

Article

Cloud Detection Using a UNet3+ Model with a Hybrid Swin Transformer and EfficientNet (UNet3+STE) for Very-High-Resolution Satellite Imagery

Jaewan Choi [1,*], Doochun Seo [2], Jinha Jung [3], Youkyung Han [4], Jaehong Oh [5] and Changno Lee [4]

[1] Department of Civil Engineering, Chungbuk National University, Chungdae-ro 1, Seowon-Gu, Cheongju 28644, Republic of Korea
[2] Satellite Ground Station Research and Development Division, National Satellite Operation & Application Center, Korea Aerospace Research Institute (KARI), Daejeon 34141, Republic of Korea; dcivil@kari.re.kr
[3] Lyles School of Civil and Construction Engineering, Purdue University, 550 Stadium Mall Drive, West Lafayette, IN 47907, USA; jinha@purdue.edu
[4] Department of Civil Engineering, Seoul National University of Science and Technology, Seoul 01811, Republic of Korea; han602@seoultech.ac.kr (Y.H.); changno@seoultech.ac.kr (C.L.)
[5] Department of Civil Engineering, Korea Maritime and Ocean University, Busan 49112, Republic of Korea; jhoh@kmou.ac.kr
* Correspondence: jaewanchoi@chungbuk.ac.kr; Tel.: +82-43-261-2406

Abstract: It is necessary to extract and recognize the cloud regions presented in imagery to generate satellite imagery as analysis-ready data (ARD). In this manuscript, we proposed a new deep learning model to detect cloud areas in very-high-resolution (VHR) satellite imagery by fusing two deep learning architectures. The proposed UNet3+ model with a hybrid Swin Transformer and EfficientNet (UNet3+STE) was based on the structure of UNet3+, with the encoder sequentially combining EfficientNet based on mobile inverted bottleneck convolution (MBConv) and the Swin Transformer. By sequentially utilizing convolutional neural networks (CNNs) and transformer layers, the proposed algorithm aimed to extract the local and global information of cloud regions effectively. In addition, the decoder used MBConv to restore the spatial information of the feature map extracted by the encoder and adopted the deep supervision strategy of UNet3+ to enhance the model's performance. The proposed model was trained using the open dataset derived from KOMPSAT-3 and 3A satellite imagery and conducted a comparative evaluation with the state-of-the-art (SOTA) methods on fourteen test datasets at the product level. The experimental results confirmed that the proposed UNet3+STE model outperformed the SOTA methods and demonstrated the most stable precision, recall, and F1 score values with fewer parameters and lower complexity.

Keywords: analysis-ready data; cloud regions; convolutional neural networks; deep learning; Swin Transformer; UNet3+STE; very high resolution

check for updates

Citation: Choi, J.; Seo, D.; Jung, J.; Han, Y.; Oh, J.; Lee, C. Cloud Detection Using a UNet3+ Model with a Hybrid Swin Transformer and EfficientNet (UNet3+STE) for Very-High-Resolution Satellite Imagery. *Remote Sens.* **2024**, *16*, 3880. https://doi.org/10.3390/rs16203880

Academic Editors: Edoardo Pasolli, Mohamed Lamine Mekhalfi, Mawloud Guermoui and Yakoub Bazi

Received: 27 August 2024
Revised: 3 October 2024
Accepted: 17 October 2024
Published: 18 October 2024

1. Introduction

Remotely sensed satellite imagery can be applied in various applications, such as digital mapping, environmental analysis, disaster monitoring, forestry, and agriculture, by processing images of wide regions and conducting time series analyses via temporal datasets. In particular, due to the development of artificial intelligence (AI) and various image-processing techniques, preprocessing algorithms for satellite imagery have rapidly evolved. Specifically, for the operation of different satellite sensors, such as Landsat, Sentinel-2, PlanetScope, and Worldview, preprocessing steps, including orthorectification, geometric correction, and radiometric correction, have been performed to provide satellite imagery archives in the form of analysis-ready data (ARD), allowing users to utilize such satellite imagery immediately [1]. In addition, when distributing ARD, along with satellite imagery, per-pixel quality assessment information, such as unusable data masks and cloud masks, is provided to users in the raster format [2,3].

In remote sensing, cloud information within satellite imagery has been utilized in environmental and meteorological studies to forecast weather and environmental changes. Utilizing stationary-orbit satellite sensors such as the Moderate Resolution Imaging Spectroradiometer (MODIS), Visible Infrared Imaging Radiometer (VIIRS), and Geostationary Korea Multi-Purpose Satellite (Geo-KOMPSAT) with spatial resolutions of approximately 1 km, algorithms for performing cloud detection at the product level in continental-scale regions have been developed for environmental and meteorological analysis purposes. Frey et al. [4] developed a spectral-based threshold algorithm using brightness temperature differences and the ratios of channels. Stökli et al. [5] proposed a regression model based on diurnal clear sky reflectance and temperature data. Mahajan and Fataniya [6] analyzed and classified various cloud detection algorithms based on the existence of clouds and cloud types, including snow, ice, and cloud shadows. Lee and Choi [7] developed an algorithm based on automatic threshold determination for generating cloud products from images provided by the GK-2A satellite.

In the imagery provided by remotely sensed satellite sensors with medium or high spatial resolutions, such as Sentinel-2, PlanetScope, and WorldView-3, cloud regions pose challenges in effectively utilizing the imagery because clouds contaminate terrestrial areas. Extracting information about cloud regions from satellite images and providing it to users can enhance the usability of the imagery, while cloud detection algorithms based on stationary-orbit satellite sensors focus on environmental applications using cloud products. Therefore, various cloud detection algorithms have been developed to extract and identify cloud areas in medium- and high-resolution satellite images and provide them as binary data or classified information products [2]. Conventional cloud detection algorithms comprise rule-based methods, such as single or multiple thresholds for each multispectral band. The representative rule-based cloud detection algorithm is the Fmask algorithm presented by Zhu and Woodcock [8]. Based on the Fmask algorithm, Zhu et al. [9] developed a multitemporal mask (Tmask) to improve the performance of the Fmask algorithm on the Landsat archive. Qin et al. [10] developed the Fmask 4.0 algorithm based on integrating auxiliary data and forming cloud probabilities and spectral contextual features. In addition, Sen2Cor, which is a software program for generating the Sentinel-2 Level 2A product, was developed to classify clouds, shadows, water, and snow and serves as a cloud information product [11]. However, most rule-based algorithms, including Fmask, rely on the surface reflectance of each band or require various spectral features, making them difficult to reliably apply to satellite images with different seasonal and regional characteristics. Various supervised pixel classification methods have been developed to overcome such limitations. Specifically, machine learning-based techniques, such as random forests, support vector machines, and multilayer perceptrons (MLPs), have been employed to extract cloud regions [12–16].

Moreover, since the introduction of AlexNet in computer vision, various deep learning models based on convolution have been developed [17,18]. Fully convolutional networks (FCNs) based on encoder–decoder structures, such as UNet, DeepLab, and FC-DenseNet, have been proposed and used for semantic segmentation [19–21]. Recently, deep learning models based on transformers, which have shown outstanding performance in natural language, have been extended to vision transformers (ViTs) and have demonstrated a superior performance to traditional CNNs on open datasets such as ImageNet-1K [22–24]. However, compared with CNNs, ViTs are computationally expensive because of the large amounts of required training data, and they lack desirable inductive biases [25]. Considering the training efficiency and computational costs, various models that integrate ConvNets and transformers have recently been proposed [25–28]. These approaches have also been implemented in remote sensing and have all been proven to achieve an improved performance when integrating two networks relative to using either ConvNet or a transformer alone [29–32].

With the acquisition of large amounts of high-resolution satellite imagery and advancements in deep learning technology, various deep learning-based studies have been conducted to extract clouds from high-resolution images. Segal-Rozenhaimer et al. [33] developed a CNN model for cloud detection using WorldView-2 and Sentinel-2 satellite imagery. Pu et al. [34] improved the accuracy at the edges of cloud areas via self-attention and spatial pyramid pooling fusion in a CNN model. Li et al. [35] developed a deep learning model based on spectral assimilation for multiple types of satellite images to generalize the model's performance and apply it to various satellite imagery. Pasquarella et al. [36] developed a spacetime context network through a weakly supervised measure of a fined-tuned atmospheric similarity model for Sentinel-2 imagery. Specifically, numerous studies have been conducted to generate open data by compiling satellite imagery, such as Landsat-8 and Sentinel-2, along with corresponding ground-truth data. These open data enable diverse users to evaluate the performance of deep learning models and conduct benchmarking analyses. The spatial procedures for automated removal of cloud and shadow (SPARCS) dataset was generated and tested to develop a CNN model for cloud detection [37]. The SPARCS dataset comprises 80 images captured in various regions, including major terrestrial habitats, inland wetlands, rocks, and ice. In addition, the WHUS2-CD+ dataset consists of 36 Sentinel-2 satellite images depicting clear skies and cloudy conditions [38]. Utilizing WHUS2-CD+, the Cloud Detection-Fusing Multi-Scale Spectral and Spatial Features (CD-FM3SFs) model was proposed [38]. The AIR-CD open dataset, constructed using GF-2 satellite images captured in various regions of China from February to November 2017 with a spatial resolution of 4 m, was utilized by He et al. [39] to develop a deformable contextual and boundary-weighted network. López-Puigdollers et al. [40] applied deep learning models to an open dataset constructed from Landsat-8 and Sentinel-2 satellite imagery and analyzed their performance. Kim and Oh [41] produced large amounts of AIHub open datasets for cloud detection in KOMPSAT images; these data are composed of 4000 image pairs with a resolution of 1000 × 1000 and were obtained through funding from the government AI training dataset building program.

Applying a rule-based approach is challenging when conducting cloud detection on high-resolution satellite imagery because of radiometric correction difficulties and spectral constraints composed of visible and near-infrared (VNIR) wavelengths with blue, green, red, and NIR channels [38]. Existing deep learning-based cloud detection algorithms and open datasets have been applied primarily to medium- and low-resolution satellite imagery, such as Landsat and Sentinel-2, which contain multiple spectral bands with VNIR and short-wave infrared (SWIR) regions [37,38]. In particular, clouds and cloud shadows within high-resolution imagery exhibit spectral characteristics similar to those of snow and water. Difficulties in defining thin clouds and producing corresponding labeling data have led to a predominant focus on binary classification research for distinguishing between cloud and non-cloud regions [39]. Therefore, improving the performance of deep learning models is necessary to detect cloud regions within high-resolution satellite images containing VNIR spectral bands. Additionally, the training processes of deep learning models are typically performed using small image patches, but clouds in high-resolution satellite imagery could exceed the sizes of these patches, leading to ineffective spatial feature learning and potentially resulting in lower performance. Therefore, developing models capable of effectively detecting clouds with various shapes and sizes exceeding a certain threshold is necessary.

In this manuscript, we proposed a new deep learning model network for detecting cloud regions in VNIR multispectral satellite images with high spatial resolutions. To generate scene-level products for the cloud detection results derived from high-resolution satellite images, we integrated traditional CNN and transformer layers to detect various clouds in the images effectively. Since a typical CNN structure could be effective at extracting local spatial features from images, we modified the structure of the UNet3+ network, which is known for its excellent semantic segmentation performance. In UNet3+, residual blocks are used in the encoder, and a deep supervision structure was employed in the decoder to integrate the feature maps acquired from each depth. Then, we reconstructed the residual block via mobile inverted bottleneck convolution (MBConv), used in the EfficientNet model, to construct the model efficiently and achieve an improved performance [42].

Additionally, some parts of the encoder with a transformer block were replaced to effectively extract the global characteristics of the clouds contained within image patches. By integrating the CNN and transformer in the encoder section, we aimed to enhance the cloud detection results produced for high-resolution satellite images with VNIR bands. In the case decoder part, the skip connections and convolutional layer of the original UNet3+ model via MBConv were reorganized. Image patches consisting of four classes—clouds, thin clouds, cloud shadows, and clear skies obtained from KOMPSAT imagery—were used for model training purposes. The KOMPSAT-3 and 3A satellite imagery used in this manuscript, provided by AIHub, offers a higher spatial resolution but is limited to four bands: R, G, B, and NIR. The performance of the trained model was also evaluated on scene-level KOMPSAT images. Therefore, the contribution of this study is to develop a new deep learning model by combining a CNN and a transformer and increasing the performance of the deep learning model for cloud detection in high-spatial-resolution imagery containing VNIR bands. The outline of our manuscript is as follows. Section 2 describes the training and test datasets the deep learning model uses for cloud detection. Section 3 presents the proposed deep learning model, and Section 4 describes the experimental results and discussion. Finally, the conclusions are provided in Section 5.

2. Materials

2.1. KOMPSAT Imagery

KOMPSAT is a representative series of satellites operated by the Korea Aerospace Research Institute (KARI) in South Korea for earth observation, monitoring, and digital mapping. Currently, KOMPSAT-3 and 3A are in operation, providing VNIR satellite imagery with high spatial resolutions, including panchromatic and four-band multispectral images. The characteristics of each satellite image are summarized in Table 1.

Table 1. Specifications of the KOMPSAT-3 and 3A satellite sensors.

Satellite Sensor		KOMPSAT-3	KOMPSAT-3A
Ground sample distance	Multispectral	2.8 m	2.2 m
	Panchromatic	0.7 m	0.55 m
Spectral wavelength	Panchromatic	450–900 nm	450–900 nm
	Blue	450–520 nm	450–520 nm
	Green	520–600 nm	520–600 nm
	Red	630–690 nm	630–690 nm
	NIR	760–900 nm	760–900 nm
Orbit altitude		685 km	528 km
Swath width		>15 km (at nadir)	>12 km (at nadir)
Radiometric resolution		14 bits	14 bits

2.2. Training and Testing Datasets

To train a deep learning model, a training dataset should be constructed based on labeled data and satellite imagery suitable for the intended use. Cloud-masked data and satellite imagery for scene products, known as the AIHub open dataset, were developed in South Korea through the AI training dataset building program provided by the government [41]. The clouds in optical satellite imagery can be influenced by terrain and location characteristics. In the AIHub open dataset, labeling data for each KOMPSAT satellite image product were generated through the visual interpretation of various regions, including urban areas, mountains, coasts, agricultural lands, and deserts. The labels of the AIHub open dataset were categorized into four classes: thick clouds, thin clouds, cloud shadows, and clear skies. The AIHub open dataset was constructed using 148 images from the KOMPSAT-3 and 3A satellites. The generated data are on the project website (www.aihub.or.kr (accessed on 17 October 2024)) [41]. In high-resolution satellite imagery, thin clouds can manifest in various forms, such as haze and fog, presenting challenges when distinguishing them from other cloud types. Therefore, in this manuscript, we inspected the 148 images provided by AIHub through visual interpretation and excluded those with inaccurately labeled thin clouds and other features. Thick and thin clouds were subsequently integrated into a single cloud class within the images, and 130 images were then recategorized into three classes—clouds, cloud shadows, and clear skies—after removing 18 images that included ambiguous labeling results. Of these, we used 116 images as training data and 14 images as test data.

Preprocessing was applied to the satellite imagery to increase the performance of the deep learning model. First, the spatial resolution of the original multispectral images was reduced by half via interpolation techniques. Since the cloud regions in high-resolution satellite imagery could occupy many pixels, patches with only clouds could exist when a small patch size, such as 128 × 128, is used. These patches might be confused with other homogeneous areas, such as sand, ice, and snow-covered regions. Even when considering the increase in the receptive field of a deep learning model, this allows the constructed model to analyze a broader area. After the spatial resolution of the original satellite data was decreased, 2% linear stretching was applied to each image, and the data were normalized to the range of 0~1. Finally, they were divided into 512 × 512 image patches. Table 2 provides examples of the image patches used for training. We considered the number of patches sufficient and did not perform data augmentation.

Table 2. Specifications of the image patches used for model training.

Characteristics	Variables
Size	Satellite image: 512 × 512 × 4 Labeling data: 512 × 512
Number	Training data: 9628 Validation data: 1069

Additionally, the image patches were divided into training and validation datasets, as the validation data were used during the training process to determine the optimal model with minimal overfitting. Figure 1 provides examples of the image patches. The AIHub datasets were labeled as thick clouds, thin clouds, cloud shadows, and clear skies. Since thick and thin clouds are difficult to distinguish even with visual inspection, the thick and thin clouds were integrated into one class, as illustrated in Figure 1. The reference data were reorganized into three classes: clear skies, clouds, and cloud shadows. We selected 14 images as test data for evaluating the trained model; these images were not used for making image patches. These images also reflected the characteristics of various regions, as detailed in Figure 2.

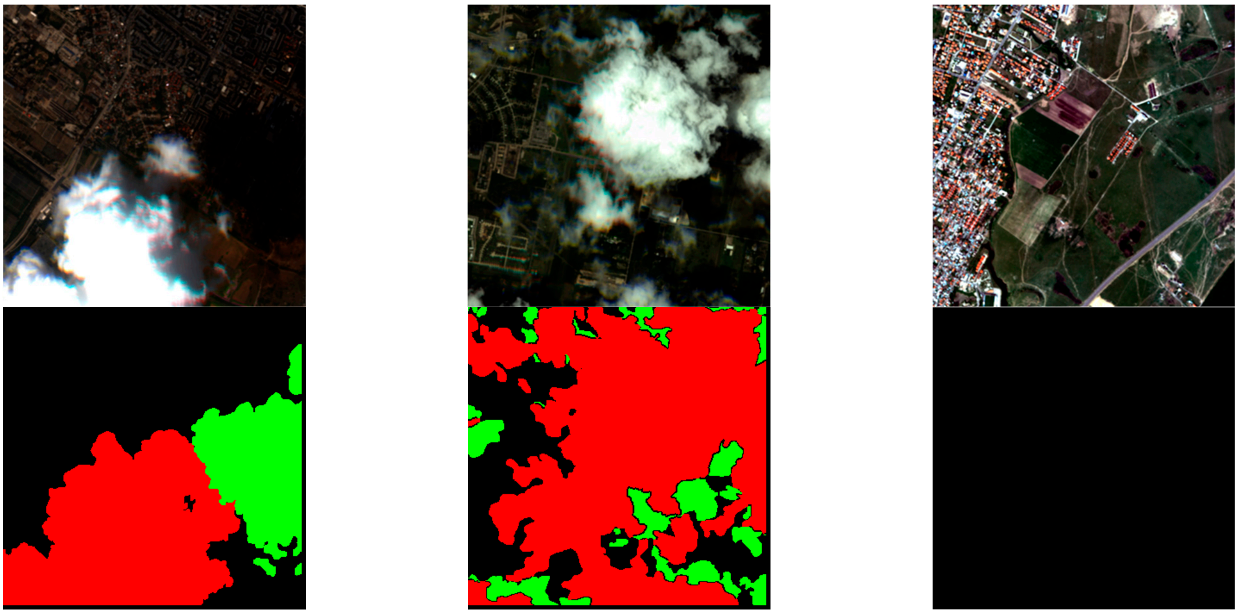


Figure 1. Examples of images contained in the training dataset: satellite images (**top**) and labeled reference data (**bottom**) (black: clear skies; red: thick and thin clouds; green: cloud shadows).

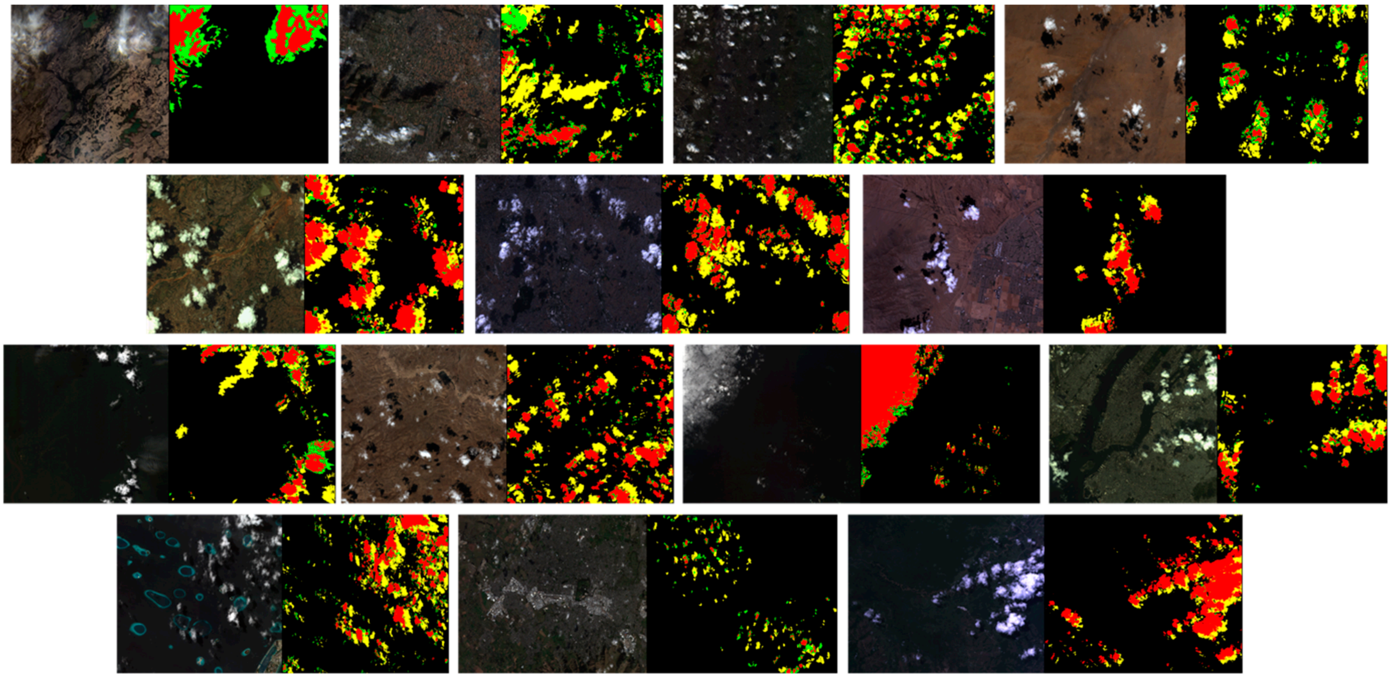


Figure 2. Test datasets for evaluating the performance of deep learning models (black: clear skies; red: thick clouds; green: thin clouds; yellow: cloud shadows).

3. Methodology

3.1. UNet3+

UNet is a fully connected network (FCN)-based approach that uses an encoder–decoder structure developed for semantic segmentation [19]. The general deep learning models for semantic segmentation employ encoder–decoder structures. The encoder, also known as the contracting path, reduces the size of the feature map or input data while extracting features, typically through a convolution unit implemented via a combination of convolution layers, activation functions, and maxpooling. The size-reduced feature map obtained from the encoder includes meaningful information for segmenting each pixel while losing spatial information. The decoder restores the feature map to the size of the input image and produces pixelwise segmentation results. In particular, the spatial information derived from the encoder process can be restored in the decoder through skip connections that utilize feature maps with the same resolution during the encoder process. However, even with skip connections, the inability to use the spatial information of each feature map stage remains an issue. To address this, UNet3+ introduced inter-connections and intra-connections to minimize the loss of spatial information while extracting sufficient information from feature maps [43]. Additionally, the training performance was improved by generating resolution-specific segmentation results from the produced feature maps and integrating these results through a deep supervision process. Figure 3 shows an overview of UNet3+ [43,44].

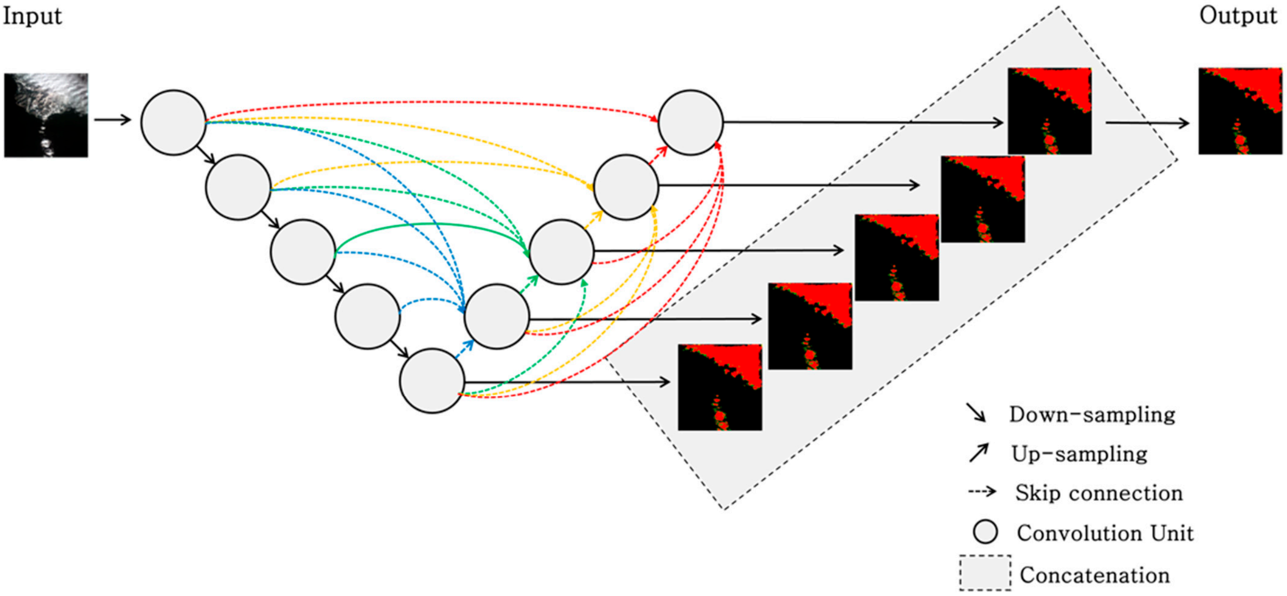


Figure 3. Architecture of UNet3+.

3.2. Proposed Network Integrating a CNN and a Transformer

This manuscript proposed a new deep learning model for detecting clouds in high-resolution satellite images by integrating convolution layers and transformers based on UNet3+. To fuse the CNN and transformer, we restructured the encoder part of UNet3+ based on MBConv, proposed by Dai et al. [25] and used in EfficientNet [42]. Figure 4 shows the overall architecture of our proposed UNet3+ with a hybrid Swin Transformer and EfficientNet (UNet3+STE). UNet3+STE is composed of an encoder and a decoder, such as UNet3+. The encoder part extracts feature maps $\boldsymbol{E} = [E_1, E_2, E_3, E_4, E_5]$ containing information about the cloud regions in high-resolution satellite images. In the typical encoder part of a CNN, the integration of a convolutional layer, an activation function, and maxpooling is repeatedly applied to reduce the size of the feature map while increasing the number of channels in the feature map to extract rich spectral information. In particular, processes based on maxpooling increase the receptive field during convolution, which enables the extraction of information across the global range of the feature map.

However, deep learning models that use convolutional layers might exhibit a poorer performance in effectively extracting global information from images than models based on transformers. ViTs, however, may face limitations, including the need for vast training data and high computational costs. To address these issues, techniques that combine CNNs and transformers have been developed, as described in the introduction [27,28,31]. Specifically, Dai et al. [25] proved that sequentially mixing CNNs and transformers is more effective from generalizability and model capacity perspectives. Utilizing this concept, CNNs and transformers were integrated to construct the encoder of the proposed model. The encoder part of the proposed model was built over five stages. The 1st through 3rd stages ($E_1 \sim E_3$)

of the encoder part consisted of the EfficientNet structure, which is based on MBConv. The Swin Transformer was employed in the 4th and 5th stages (E_4 and E_5) to generate feature maps for extracting the final global information. The detailed structure of the encoder part is described in Figure 5.

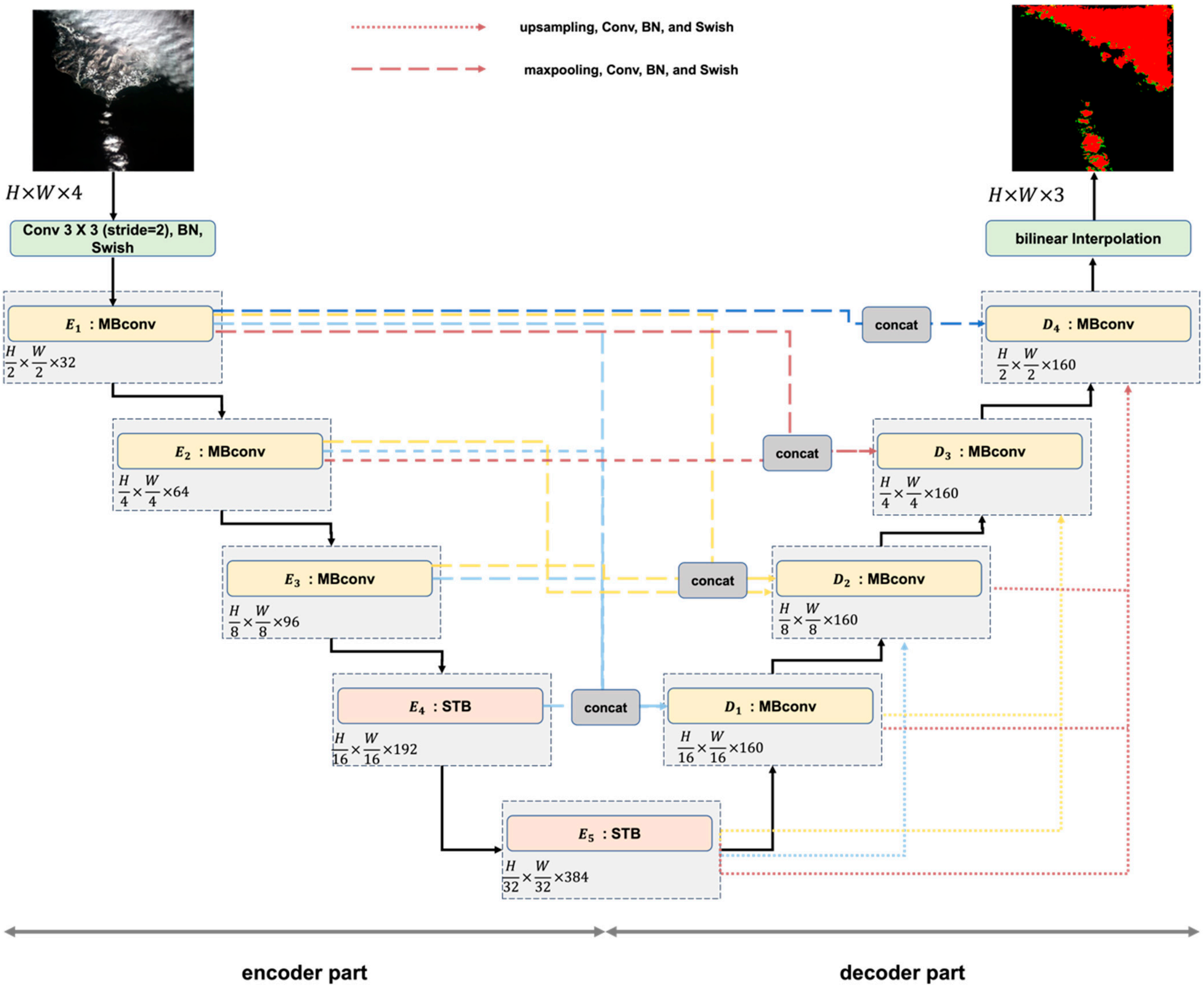


Figure 4. Architecture of the proposed UNet3+STE model (where $\boldsymbol{E} = [E_1, E_2, E_3, E_4, E_5]$ contains the feature map of each encoder stage and $\boldsymbol{D} = [D_1, D_2, D_3, D_4]$ includes the feature map of each decoder stage).

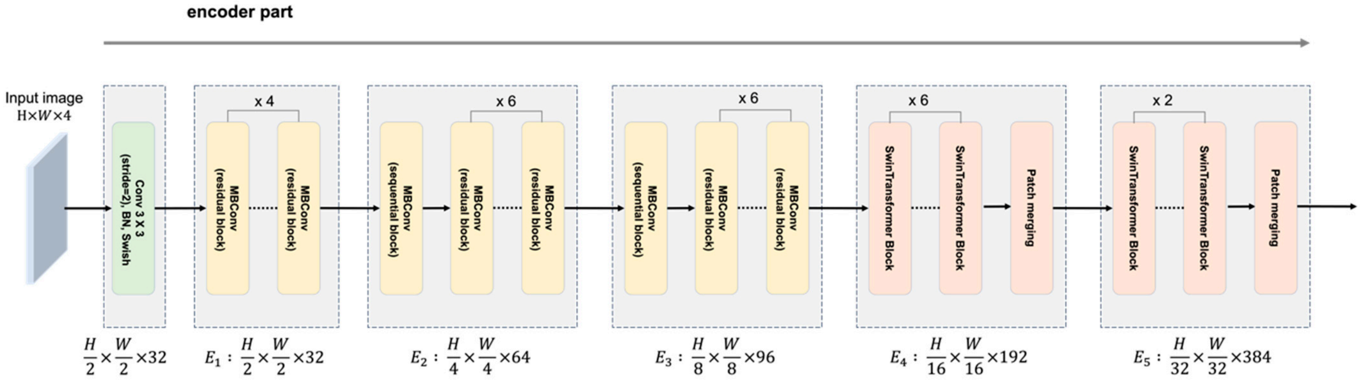
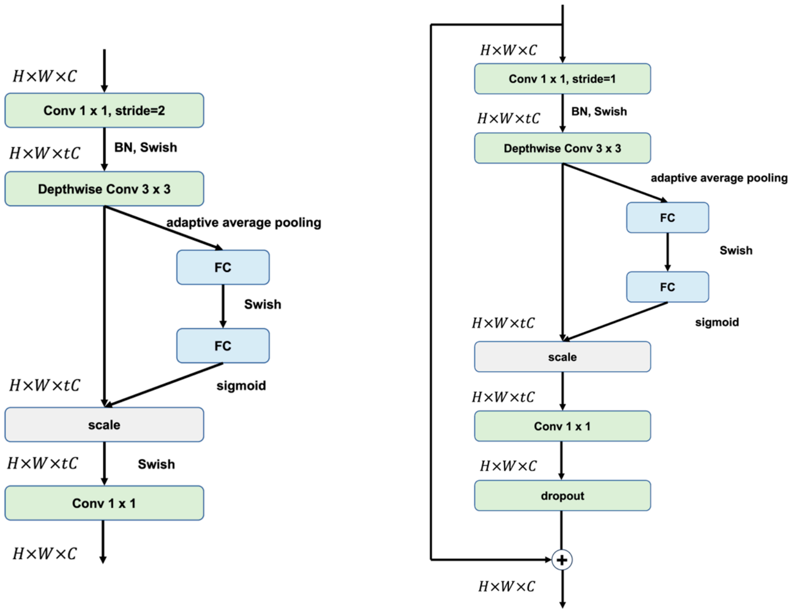


Figure 5. Structure of the encoder part.

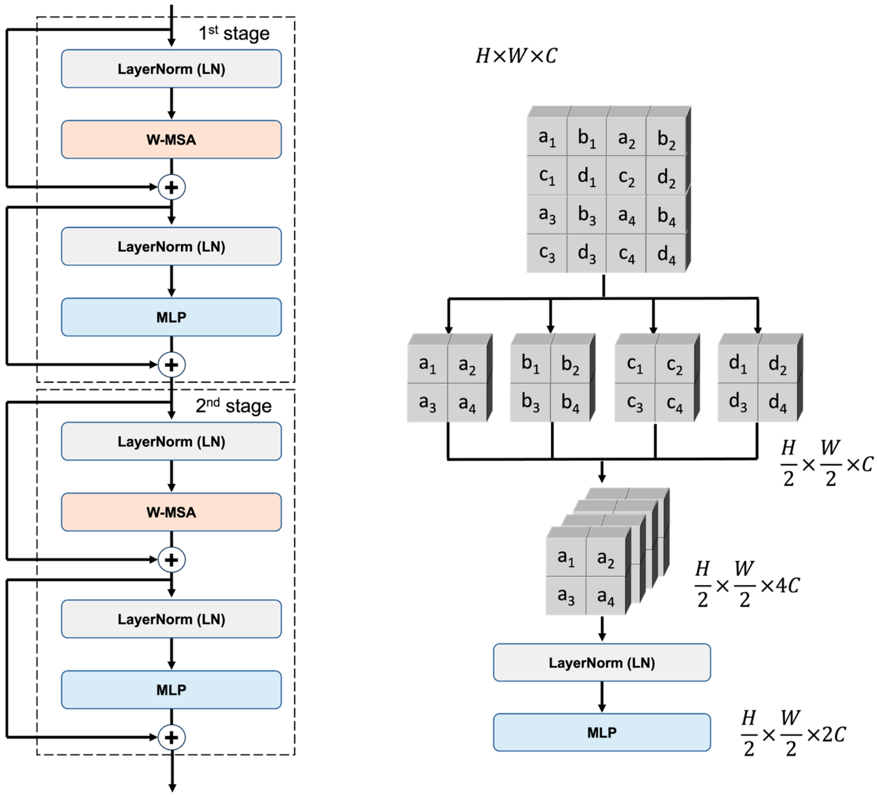
As shown in Figures 4 and 5, a convolutional layer consisting of a 3 × 3 convolution with a stride of 2, batch normalization, and a Swish activation function was used to create the initial feature map from the input data. By utilizing the strided convolution, the size of the input data was reduced by half, generating a feature map E_1 with dimensions of $\frac{H}{2} \times \frac{W}{2} \times 32$. Based on this feature map, MBConv was applied in the 1st through 3rd stages of the encoder. The MBConv used in the encoder part, as shown in Figure 6a,b, was composed of a 1 × 1 strided convolution, a 3 × 3 depthwise convolution, and a squeeze-and-excitation module [45,46]. In the first 1 × 1 convolution, the expansion ratio increased the number of channels. After the 1 × 1 convolution, a 3 × 3 depthwise convolution was applied. The squeeze-and-excitation module, composed of adaptive average pooling and two fully connected layers, assigned additional weights to each feature map during the two-stage convolution process [47]. Finally, the 1 × 1 convolution adjusted the number of channels to match the number of input feature maps, forming a bottleneck. In the proposed model, MBConv was iteratively applied within each stage. When the size of the feature map was reduced via basic convolution (stride = 1) in the initial 1 × 1 convolution, a sequential block structure was used (Figures 5 and 6a). However, when the size of the feature map was decreased (1 × 1 convolution with stride = 2), MBConv was configured as a residual block to enhance the performance of the network (Figures 5 and 6b). Therefore, the MBConv of each stage utilized a combination of sequential and residual block structures depending on the stride of the 1 × 1 convolution. In addition, these methods were iteratively applied according to the structure of each CNN stage, as shown in Figure 5.



(**a**) Sequential block (**b**) Residual block

Figure 6. Structures of the MBConvs in UNet3+STE.

The 4th and 5th stages of the encoder were composed of Swin Transformer blocks to enforce the global information of each feature via the convolutional layers of the 1st~3rd stages [23]. Various attention-based techniques have been developed for applying transformers because of the differences between the resolutions and spatial characteristics of images and those of natural language processing. In particular, transformer-based methods for image processing divide an image into patches and apply window-based attention to each patch. However, since the diverse information in images has spatial characteristics, the existing window-based attention methods have difficulty reflecting the spatial information among patches. The Swin Transformer utilizes cross-window connections via a shifted window-based attention model to preserve the advantages of nonoverlapping windows [23]. Figure 7a shows the structure of the Swin Transformer block. In the first step, the feature map was divided into windows of size M to create a regular partition, and then, window-multihead self-attention (W-MSA) and an MLP were applied to each partition.



(**a**) Swin Transformer block (**b**) Patch merging module

Figure 7. Structure of the Swin Transformer layer.

A layer normalization (LN) layer was added before the MSA and MLP, and residual connections were employed for both modules. Because each partition remained fixed during the application of MSA, no connections occurred between the local windows. Therefore, before the feature map was divided into windows of size M, the feature map was shifted by $\frac{M}{2}$ and then partitioned. A cyclic shifting method was used to move the windows to the top left to create shifted windows, enabling a more efficient batch calculation method. The attention process applied to the shifted window partition is defined as SW-MSA. In the second module, the SW-MSA and MLP were configured as a residual block, allowing the model to learn about the connections between local windows. The feature map generated through the Swin Transformer block was reconstructed through the patch merging process. As shown in Figure 7b, by splitting the pixels of neighboring 2×2 patches into four separate pieces to create individual feature maps that were half the original size, the size was reduced by half. At the same time, the number of channels was doubled, transforming the patches into new patches through the concatenation process and the MLP.

The decoder integrated the feature maps generated at each stage of the encoder with the feature maps generated during the decoding process via MBConv. The decoder consists of four stages, and Figure 8 shows an example of D_2 using the 2nd MBConv process in the decoder part. First, the feature maps (e.g., D_1 in Figures 4 and 8), which were acquired from the previous stage and were smaller than the feature map D_2 of the decoder part, and the final-stage feature map of the encoder part, E_5, were transformed into feature maps with the same size as that of D_2 through bilinear resampling. Moreover, the feature maps of the encoder part that were equal to or larger than that of D_2 in terms of size were reduced to the same size as that of D_2 via maxpooling. The feature maps derived from the encoder and decoder parts, transformed to the same size as that of D_2, were integrated into a single feature map through skip connections.

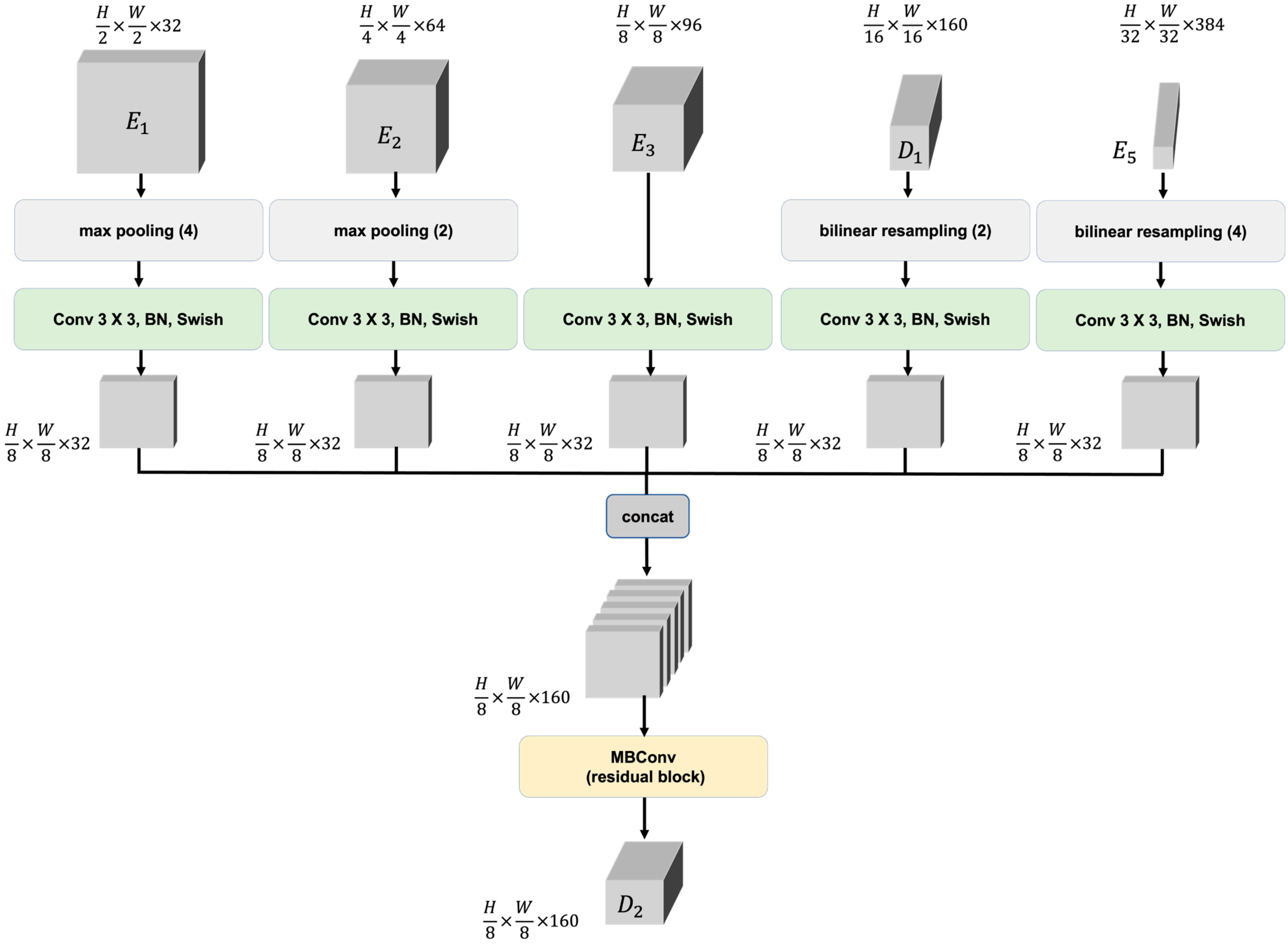

$\frac{H}{2}\times\frac{W}{2}\times32$
$\frac{H}{4}\times\frac{W}{4}\times64$
$\frac{H}{8}\times\frac{W}{8}\times96$
$\frac{H}{16}\times\frac{W}{16}\times160$
$\frac{H}{32}\times\frac{W}{32}\times384$
E_1
E_2
E_3
D_1
E_5
max pooling (4)
max pooling (2)
bilinear resampling (2)
bilinear resampling (4)
Conv 3 X 3, BN, Swish
Conv 3 X 3, BN, Swish
Conv 3 X 3, BN, Swish
Conv 3 X 3, BN, Swish
Conv 3 X 3, BN, Swish
$\frac{H}{8}\times\frac{W}{8}\times32$
$\frac{H}{8}\times\frac{W}{8}\times32$
$\frac{H}{8}\times\frac{W}{8}\times32$
$\frac{H}{8}\times\frac{W}{8}\times32$
$\frac{H}{8}\times\frac{W}{8}\times32$
concat
$\frac{H}{8}\times\frac{W}{8}\times160$
MBConv
(residual block)
$\frac{H}{8}\times\frac{W}{8}\times160$
D_2

Figure 8. Examples of structures for calculating D_2 in the decoder part.

The main reason for performing skip connections on the feature maps of the encoder part was to reflect the spatial information of that stage. Then, the feature map of the decoder part was integrated to utilize the global information of the feature map acquired from the previous stage. After each feature map extracted from the decoder and encoder parts was adjusted to have 32 channels through convolution, it was integrated into one map through concatenation. Finally, D_2 was generated through the MBConv operation. This feature map was reused as the input data in the next decoder stage. Ultimately, the feature map D_4 generated in the 4th stage was computed as a feature map for the three classes through convolution, and an output with the same size as that of the input image was produced through the interpolation process.

3.3. Deep Supervision for Efficiently Conducting Training

To train the proposed model, the deep supervision techniques used in UNet3+ were employed [43]. For the optimized training process of the model, feature maps of various sizes generated from each decoder part are needed to effectively reflect each class's characteristics. To achieve this, as shown in Figure 9, deep supervision transformed all the feature maps of the decoder part into outputs for the three classes, and a loss function was calculated for these outputs to proceed with the training procedure. A 3 × 3 convolution was performed on each feature map to generate a feature map with three channels, and interpolation was applied to the ground-truth data to match the sizes of the feature maps at different stages before applying loss functions. The loss functions $L_1 \sim L_5$ for each feature map and ground truth were calculated, as shown in Figure 9, and the final loss function was calculated by adding them together. The size of the ground-truth data was reduced to shorten the computational time, and larger feature maps were adjusted to have higher weights.

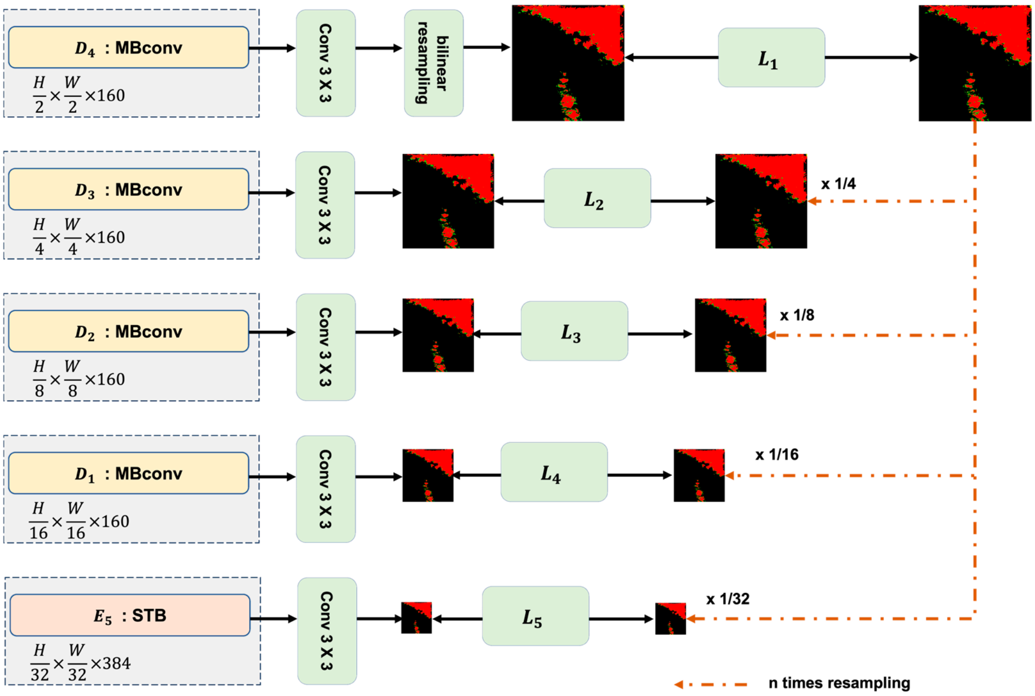

D_4 : MBconv
$\frac{H}{2}\times\frac{W}{2}\times160$
D_3 : MBconv
$\frac{H}{4}\times\frac{W}{4}\times160$
D_2 : MBconv
$\frac{H}{8}\times\frac{W}{8}\times160$
D_1 : MBconv
$\frac{H}{16}\times\frac{W}{16}\times160$
E_5 : STB
$\frac{H}{32}\times\frac{W}{32}\times384$
Conv 3 X 3
bilinear resampling
L_1
L_2
L_3
L_4
L_5
x 1/4
x 1/8
x 1/16
x 1/32
n times resampling

Figure 9. Deep supervision structures in the decoder part.

4. Experimental Results

4.1. Training the Deep Learning Model

The proposed UNet3+STE and state-of-the-art (SOTA) models were trained via the same hardware and hyperparameters presented in Table 3. The deep learning models were implemented via PyTorch on Linux, and training was performed using an NVIDIA RTX A5000 GPU. The batch size was set to 3 for training the models, and the weighted adaptive moment estimation (AdamW) optimizer was used. CycleLR was used to adjust the learning rate when the optimizer was applied to increase the efficiency of the training process. The utilized loss function was categorical cross-entropy. In the case of the proposed model, deep supervision was used, so the loss function was composed of the average of the loss functions produced at each stage, as shown in Figure 9. The UNet3+STE and SOTA models were trained using the same training and validation data presented in Table 2.

Table 3. Hardware and hyperparameters for training the deep learning model.

Hardware and Hyperparameters		Value
Hardware	CPU	Intel Xeon W-2235 (3.8 GHz)
	GPU	NVIDIA Quadro RTX A5000
	RAM	128 GB
	OS	Linux
	Framework	PyTorch
	Batch size	3
	Optimizer	AdamW
	Number of epochs	100
	Loss function	Categorical cross-entropy

Table 3. *Cont.*

Hardware and Hyperparameters		Value
Hyperparameter	Base learning rate	0.0001
	Max learning rate	0.001
	Step size	5
	Gamma	0.995
	Mode	Exponential range

4.2. Assessment of the Proposed Deep Learning Model

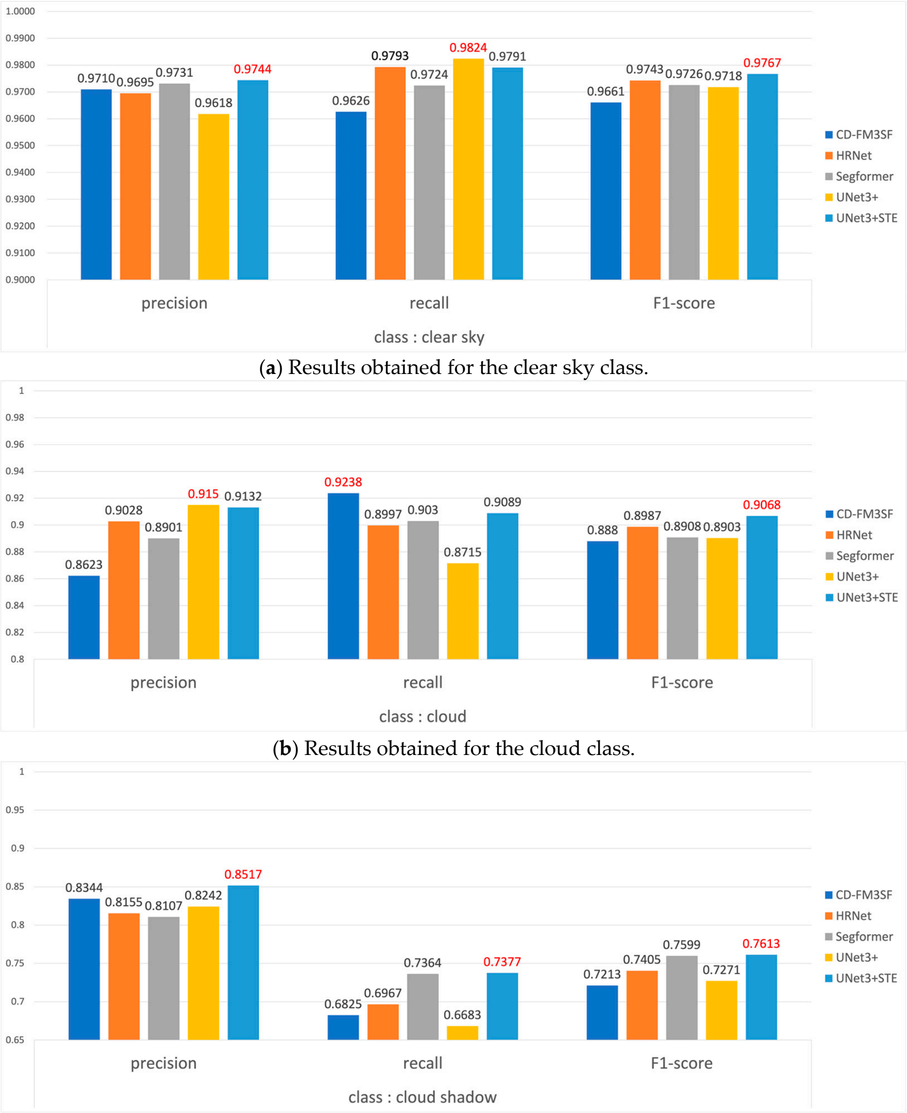
We conducted a comparative evaluation with the SOTA techniques to evaluate the performance of the proposed UNet3+STE method. CD-FM3SF, HRNet, UNet3+, and the Segformer were selected as these SOTA techniques [24,38,43,48]. UNet3+ was chosen as the benchmark for the proposed method, and the Segformer was used to conduct a comparative evaluation with transformer-based techniques [24,43]. HRNet is a convolution-based segmentation method, whereas CD-FM3SF is a representative method proposed for cloud detection [38,48]. The performance of the models was assessed in terms of the precision, recall, and F1 score values achieved for each class, as well as the average F1 score. The test data comprised 14 KOMPSAT-3 and 3A satellite images were not included in the training process. Since the KOMPSAT-3 and 3A imagery used as the test data were much larger than the image patch size, the test images were divided into smaller patches for applying the deep learning models due to GPU memory limitations. Specifically, as this study aimed to apply the developed deep learning model to satellite imagery at the product level, overlapping image patches were used to minimize the errors induced at the boundaries of the image patches. Each deep learning model was applied to a 512 × 512 image patch. The test images were divided into patches, with each patch overlapping with the adjacent patches by 32 × 32 patches. The final output was generated by integrating the results obtained for the 480 × 480 pixels in each 512 × 512 image patch, excluding the overlapping 32 × 32 pixels at the boundaries. The average F1 score results produced by each deep learning model for each image are shown in Table 4.

Table 4. Quantitative evaluation results produced by the proposed deep learning model and the SOTA models.

Test Image ID	CD-FM3SF	HRNet	Segformer	UNet3+	UNet3+STE
1	0.9421	0.9541	0.9433	0.9527	***0.9555***
2	0.9187	0.9213	0.9252	0.9141	***0.934***
3	0.9244	0.9232	***0.9339***	0.9193	0.9331
4	0.9707	0.9735	0.9725	0.9713	***0.975***
5	0.9543	0.9569	0.957	0.9527	***0.9587***
6	0.9465	0.949	***0.9541***	0.9413	0.9529
7	0.9876	0.9894	***0.9915***	0.9882	0.99
8	0.9683	0.9727	***0.9754***	0.9686	0.9733
9	0.9284	0.9346	0.9325	0.9284	***0.9383***
10	0.9552	0.9554	***0.9592***	0.9539	0.9587
11	0.9147	0.9754	0.9418	0.9581	***0.978***
12	0.9626	0.9628	0.9732	0.964	***0.9733***
13	0.8828	0.9161	0.9085	0.9091	***0.9235***
14	0.9776	0.9778	0.978	0.9787	***0.9813***
Mean	0.9453	0.9544	0.9533	0.95	***0.959***

As shown in the experimental results of Table 4, the Segformer method had the highest values for a total of five images, whereas the proposed UNet3+STE method yielded the best results for nine images. The average F1 score of the proposed model was the highest across the fourteen images used for the test. Therefore, it was confirmed that the proposed model is stable and exhibits excellent performance compared with that of the SOTA algorithms.

Figure 10 shows the average precision, recall, and F1 score values produced for each class of the test dataset. As shown in Figure 10, the F1 score values ranked from high to low are listed in the following order: clear skies, clouds, and cloud shadows. While the clear sky class accounted for the greatest proportion of high-resolution satellite images in our training dataset, clouds and cloud shadows constituted the lowest proportions. Therefore, the results reflected these characteristics of imbalanced training data. Nevertheless, since cloud areas were distinguishable from the clear sky class, most of the results produced by the deep learning models yielded F1 scores of 0.88 or higher for the cloud class. As shown in Figure 10a, the results of the proposed method presented the highest precision and F1 score for the clear sky areas. In the case of UNet3+, although it had the highest recall, it presented the lowest precision, indicating a tendency to overestimate clear skies. This tendency was related to the results of the UNet3+ method, as shown in Figure 10b,c, which had very low recall values for clouds and cloud shadows and underestimated these classes in contrast to clear skies. Additionally, the CD-FM3SF results also overestimated the cloud class. However, the proposed UNet3+STE method yielded the highest F1 scores for the cloud and cloud shadow areas, with the smallest differences between the precision and recall values. Therefore, the proposed method could detect clouds in high-resolution satellite images with the highest level of stability.



(**a**) Results obtained for the clear sky class.

(**b**) Results obtained for the cloud class.

(**c**) Results obtained for the cloud shadow class.

Figure 10. Precision, recall, and F1 scores for each class.

Figures 11–13 illustrate the results obtained after applying each model to the test dataset. Figures 11–13 show that the deep learning models efficiently detected the overall cloud regions. First, Figure 11 represents an entire image processed at the product level with a size of 5965 × 6317. As shown in Figure 11a, clouds are widely distributed over the entire area in the upper left. The proposed UNet3+STE effectively detected the overall clouds, whereas CD-FM3SF, HRNet, and the Segformer detected some coastal areas as cloud shadows. Additionally, compared with UNet3+STE, the Segformer and UNet3+ detected some of the widely distributed cloud regions on the left as clear skies. Therefore, the proposed method could effectively detect large cloud regions. Figures 12 and 13 are examples of 2000 × 2000 subset areas of the cloud detection results produced for the product-level image. As shown in Figure 13, all the SOTA methods and proposed techniques effectively detected the cloud and cloud shadow areas within the urban area. This finding indicates that AIHub data could be used to build an effective cloud detection model for satellite images with high spatial resolutions. However, some techniques detected clouds as clear skies (Figure 12c,e) or failed to distinguish dark areas with low reflectance values from cloud shadows (Figure 12f). On the other hand, the proposed technique stably detected clouds and cloud shadow areas compared with the SOTA techniques, indicating that the proposed approach, which integrates a CNN and a transformer, could be more effective for detecting clouds.

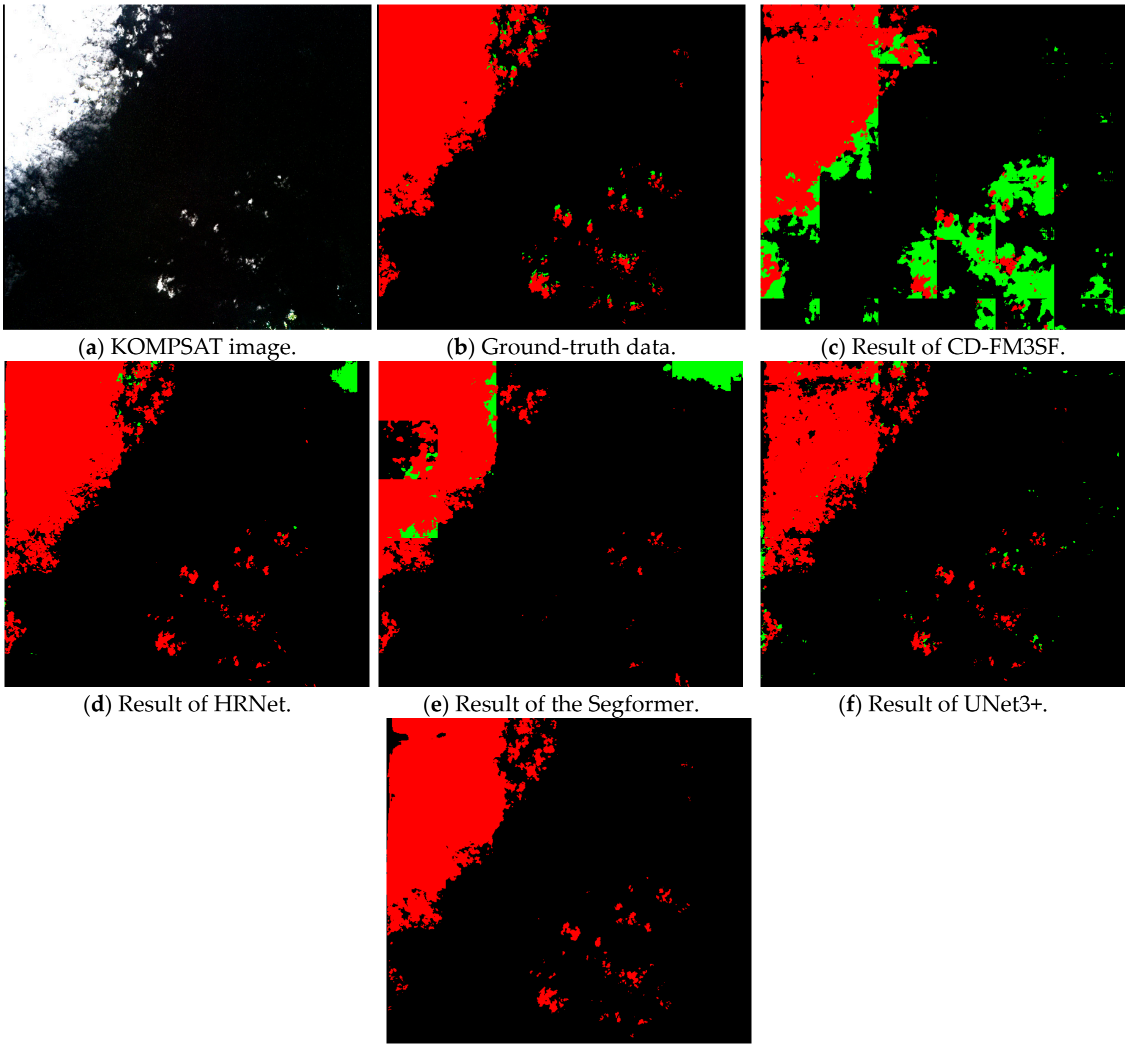
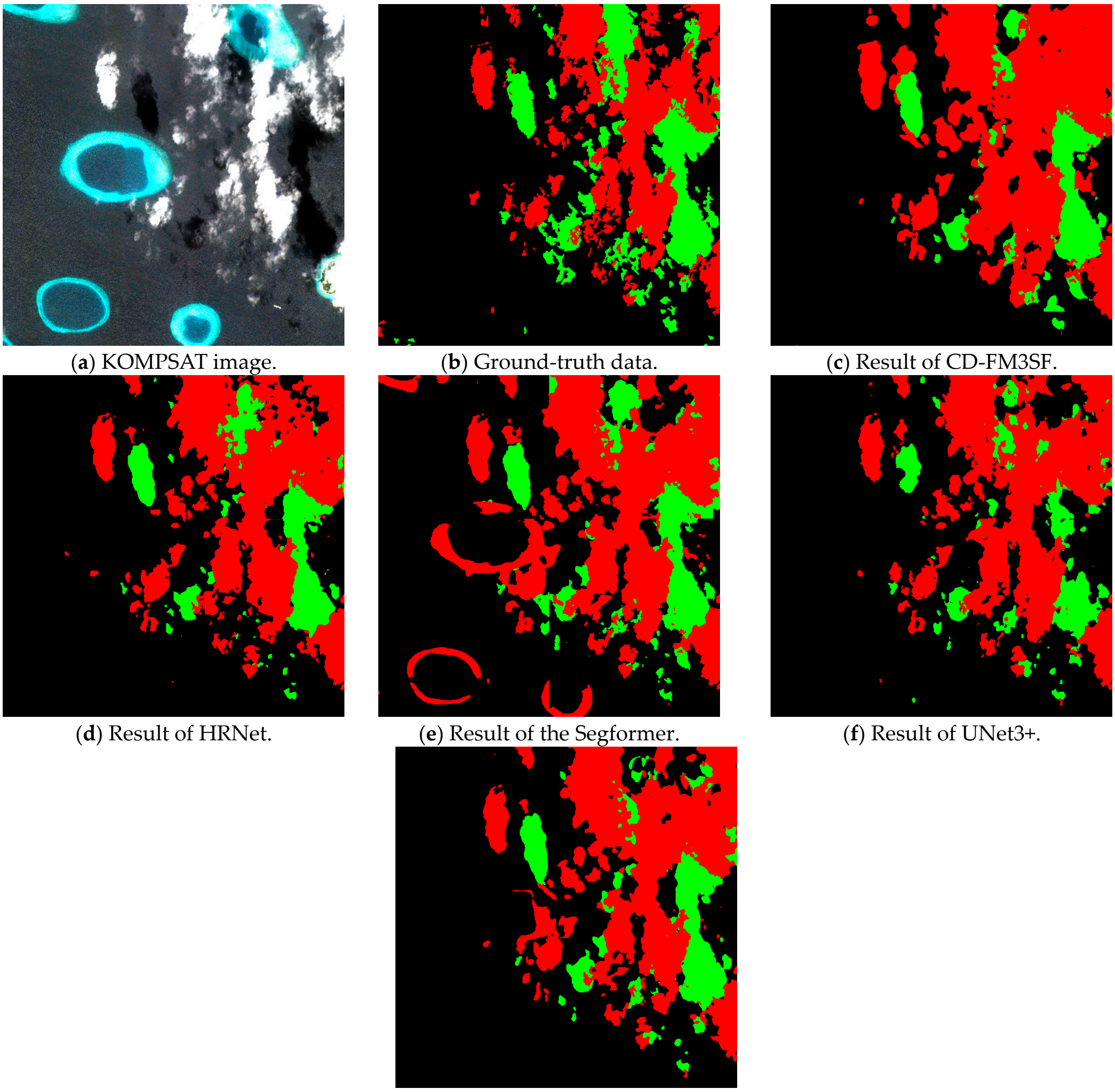

(**a**) KOMPSAT image.
(**b**) Ground-truth data.
(**c**) Result of CD-FM3SF.
(**d**) Result of HRNet.
(**e**) Result of the Segformer.
(**f**) Result of UNet3+.
(**g**) Result of UNet3+STE.

Figure 11. Cloud detection results produced for high-spatial-resolution (5965 × 6317) images at the product level (black: clear skies; red: thick and thin clouds; green: cloud shadows).



(**a**) KOMPSAT image.

(**b**) Ground-truth data.

(**c**) Result of CD-FM3SF.

(**d**) Result of HRNet.

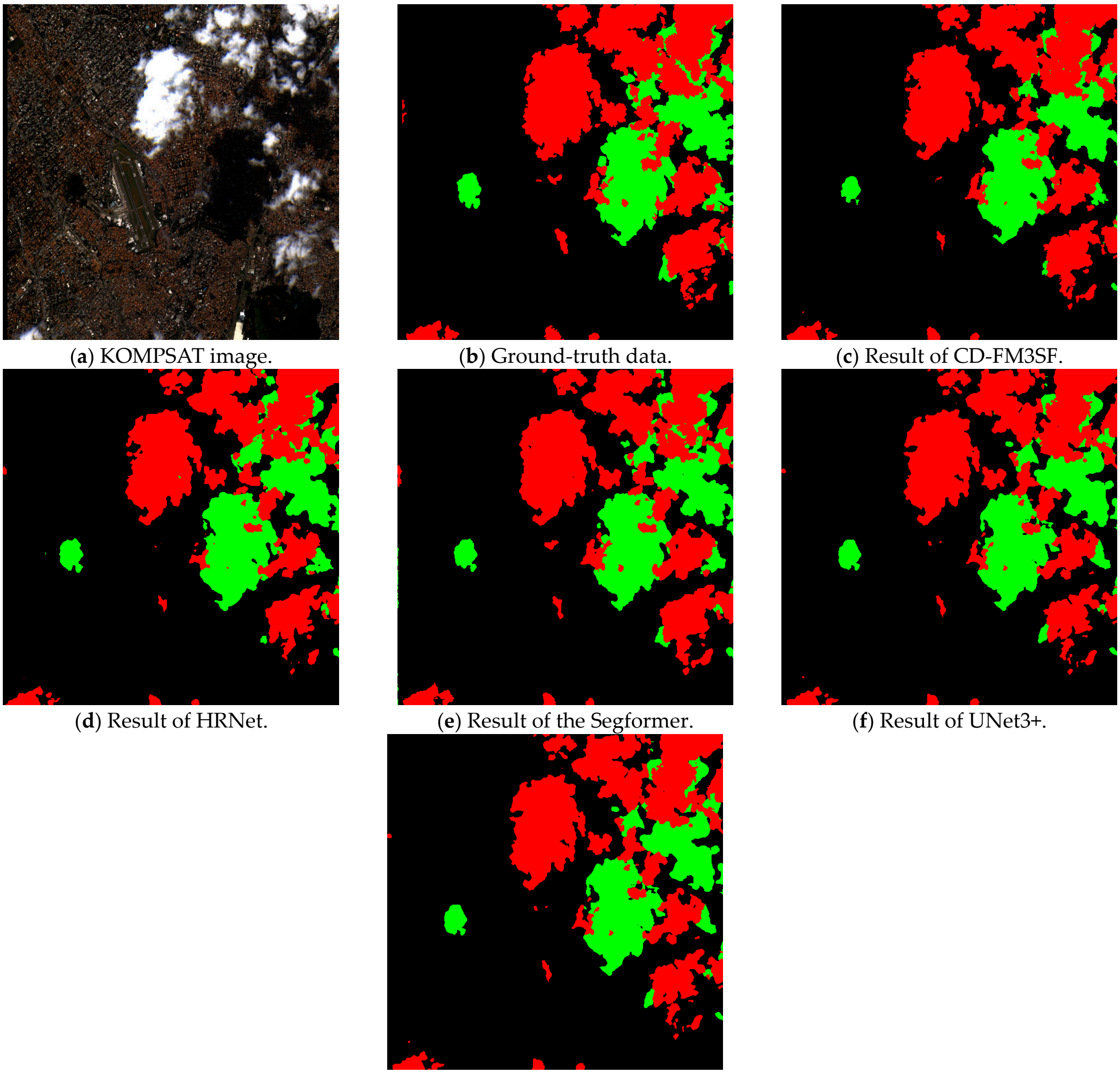
(**e**) Result of the Segformer.

(**f**) Result of UNet3+.

(**g**) Result of UNet3+STE.

Figure 12. First-subset images (2000 × 2000) of the cloud detection results produced for high-spatial-resolution (5965 × 5720) images at the product level.

Additionally, an analysis of the model complexity and computational efficiency was conducted. As shown in Table 5, the proposed method in this manuscript had the fewest FLOPs and parameters, excluding the CD-FM3SF model, which achieved the lowest cloud detection performance. By utilizing MBconv in the encoder, the proposed method demonstrated a superior model complexity and computational efficiency compared with those of UNet3+. Additionally, by combining a CNN with a transformer, it was confirmed that the model demonstrated superior performance with fewer parameters and lower complexity than a model using only a transformer did. Thus, the proposed model outperformed the SOTA algorithms in terms of accuracy and efficiency.



(**a**) KOMPSAT image.

(**b**) Ground-truth data.

(**c**) Result of CD-FM3SF.

(**d**) Result of HRNet.

(**e**) Result of the Segformer.

(**f**) Result of UNet3+.

(**g**) Result of UNet3+STE.

Figure 13. Second-subset images (2000 × 2000) of the cloud detection results produced for high-spatial-resolution (5965 × 5073) images at the product level.

Table 5. Complexity and computational efficiency of the SOTA and proposed algorithms.

Model	FLOPs	Parameters
CD-FM3SF	23.0 G	0.66 M
HRNet	183.6 G	137.55 M
Segformer	99.8 G	84.60 M
UNet3+	203.0 G	6.75 M
UNet3+STE	65.7 G	5.93 M

4.3. Ablation Study

Additionally, we evaluated UNet3+STE without deep supervision to verify the efficacy of the proposed method. Table 6 shows the performance evaluation results obtained for each class with the application of deep supervision. As shown in Table 6, except for the recall achieved for the clear sky class, the proposed UNet3+STE method outperformed the model without deep supervision in terms of the other metrics. It was confirmed that generating stagewise outputs in the decoder part based on deep supervision and calculating the loss function for each output during training was a much more effective strategy. Moreover, in this experiment, categorical cross-entropy was used as the loss function for the comparison with the SOTA method, and the performance of the model could be improved by applying an optimal loss function that is effective in cloud detection cases with imbalanced data. However, this study focused on improving the UNet3+ model for conducting cloud detection in high-resolution satellite images, so the impact of the loss function was not analyzed.

Table 6. Quantitative evaluation results obtained by UNet3+STE when deep supervision was applied.

Method	**Clear Sky**			**Cloud**			**Cloud Shadow**			**Mean F1 Score**
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	
UNet3+STE without deep supervision	0.9726	***0.9810***	0.9766	0.9077	0.9081	0.9056	0.8380	0.7224	0.7576	0.9585
UNet3+STE	***0.9744***	0.9791	***0.9767***	***0.9132***	***0.9089***	***0.9068***	***0.8517***	***0.7377***	***0.7613***	***0.9590***

5. Conclusions

In this manuscript, we developed a deep learning model to detect cloud regions in very-high-resolution (VHR) satellite images at the product level. The proposed UNet3+STE model uses an encoder that sequentially combines a CNN structure composed of MBConv and a transformer structure composed of a Swin Transformer. Sequentially combining the CNN and transformer enables the transformer to be effectively trained with a small amount of training data and generate feature maps containing information about the local and global cloud areas in the input image by reflecting the characteristics of both the CNN and transformer. Additionally, when deep supervision was used, the model's overall performance improved. The experimental results on the AIHub datasets acquired from KOMPSAT-3 and 3A indicated that UNet3+STE produced the highest F1 scores across the entire test dataset. Specifically, the SOTA techniques tended to overestimate or underestimate the results for the clear sky, cloud, and cloud shadow classes, but the proposed method demonstrated stable precision and recall values. Since the experiments conducted in this study were performed on KOMPSAT-3 and 3A images at the product level, the proposed method yielded a performance directly applicable to actual VHR satellite images. The UNet3+STE model developed in this manuscript could be used as a preprocessing step in various remote sensing application fields for creating ARD for VHR satellite imagery and for improving data usability. Moreover, in high-resolution satellite imagery, there is often an imbalance between the classes of clouds, clear skies, and shadows in the training data. In future work, to minimize performance degradation due to this imbalance, it may be necessary to optimize and refine the loss function and hyperparameters during training.

Author Contributions: Conceptualization, J.C. and D.S.; methodology, J.C.; software, J.C. and Y.H.; validation, J.C., D.S. and J.J.; formal analysis, J.C. and J.J.; data curation, J.C., C.L. and J.O.; writing—original draft preparation, J.C.; writing—review and editing, J.C., J.J. and Y.H.; visualization, Y.H.; supervision, J.C., C.L. and J.O. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2021M1A3A4A11032016). In addition, this work was supported by funding from the academic research program of Chungbuk National University in 2024.

Data Availability Statement: The AIHub dataset is available online upon request at https://aihub.or.kr (accessed on 6 August 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dwyer, J.L.; Roy, D.P.; Sauer, B.; Jenkerson, C.B.; Zhang, H.K.; Lymburner, L. Analysis Ready Data: Enabling Analysis of the Landsat Archive. *Remote Sens.* **2018**, *10*, 1363. [CrossRef]
2. Foga, S.; Scaramuzza, P.L.; Guo, S.; Zhu, Z.; Dilley, R.D.; Beckmann, T.; Schmidt, G.L.; Dwyer, J.L.; Joseph Hughes, M.; Laue, B. Cloud Detection Algorithm Comparison and Validation for Operational Landsat Data Products. *Remote Sens. Environ.* **2017**, *194*, 379–390. [CrossRef]
3. Frantz, D.; Haß, E.; Uhl, A.; Stoffels, J.; Hill, J. Improvement of the Fmask algorithm for Sentinel-2 images: Separating clouds from bright surfaces based on parallax effects. *Remote Sens. Environ.* **2018**, *215*, 471–481. [CrossRef]
4. Frey, R.A.; Ackerman, S.A.; Liu, Y.; Strabala, K.I.; Zhang, H.; Key, J.R.; Wang, X. Cloud Detection with MODIS. Part I: Improvements in the MODIS Cloud Mask for Collection 5. *J. Atmos. Ocean. Technol.* **2008**, *25*, 1057–1072. [CrossRef]
5. Stöckli, R.; Bojanowski, J.S.; John, V.O.; Duguay-Tetzlaff, A.; Bourgeois, Q.; Schulz, J.; Hollmann, R. Cloud Detection with Historical Geostationary Satellite Sensors for Climate Applications. *Remote Sens.* **2019**, *11*, 1052. [CrossRef]
6. Mahajan, S.; Fataniya, B. Cloud detection methodologies: Variants and development—A review. *Complex Intell. Syst.* **2019**, *6*, 251–261. [CrossRef]
7. Lee, S.; Choi, J. Daytime Cloud Detection Algorithm Based on a Multitemporal Dataset for GK-2A Imagery. *Remote Sens.* **2021**, *13*, 3215. [CrossRef]
8. Zhu, Z.; Woodcock, C.E. Automated cloud, cloud shadow, and snow detection in multitemporal landsat data: An algorithm designed specifically for monitoring land cover change. *Remote Sens. Environ.* **2014**, *152*, 217–234. [CrossRef]
9. Zhu, Z.; Wang, S.; Woodcock, C.E. Improvement and expansion of the Fmask algorithm: Cloud, cloud shadow, and snow detection for Landsats 4–7, 8, and Sentinel 2 images. *Remote Sens. Environ.* **2015**, *159*, 269–277. [CrossRef]
10. Qiu, S.; Zhu, Z.; He, B. Fmask 4.0: Improved cloud and cloud shadow detection in Landsats 4–8 and Sentinel-2 imagery. *Remote Sens. Environ.* **2019**, *231*, 111205. [CrossRef]
11. Main-Knorn, M.; Pflug, B.; Louis, J.; Debaecker, V.; Müller-Wilm, U.; Gascon, F. Sen2Cor for sentinel-2. In Proceedings of the Image and Signal Processing for Remote Sensing XXIII, Warsaw, Poland, 4 October 2017; p. 1042704.
12. Bai, T.; Li, D.; Sun, K.; Chen, Y.; Li, W. Cloud Detection for High-Resolution Satellite Imagery Using Machine Learning and Multi-Feature Fusion. *Remote Sens.* **2016**, *8*, 715. [CrossRef]
13. Chen, X.; Liu, L.; Gao, Y.; Zhang, X.; Xie, S. A Novel Classification Extension-Based Cloud Detection Method for Medium-Resolution Optical Images. *Remote Sens.* **2020**, *12*, 2365. [CrossRef]
14. Wei, J.; Huang, W.; Li, Z.; Sun, L.; Zhu, X.; Yuan, Q.; Liu, L.; Cribb, M. Cloud Detection for Landsat Imagery by Combining the Random Forest and Superpixels Extracted via Energy-Driven Sampling Segmentation Approaches. *Remote Sens. Environ.* **2020**, *248*, 112005. [CrossRef]
15. Yao, X.; Guo, Q.; Li, A.; Shi, L. Optical remote sensing cloud detection based on random forest only using the visible light and near-infrared image bands. *Eur. J. Remote Sens.* **2022**, *55*, 150–167. [CrossRef]
16. Pirinen, A.; Abid, N.; Paszkowsky, N.A.; Timoudas, T.O.; Scheirer, R.; Ceccobello, C.; Kovács, G.; Persson, A. Creating and Leveraging a Synthetic Dataset of Cloud Optical Thickness Measures for Cloud Detection in MSI. *Remote Sens.* **2024**, *16*, 694. [CrossRef]
17. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; Volume 1, pp. 1097–1105.
18. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6999–7019. [CrossRef]
19. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv* **2015**, arXiv:1505.04597[cs].
20. Jégou, S.; Drozdzal, M.; Vazquez, D.; Romero, A.; Bengio, Y. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1175–1183.
21. Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef]
22. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
23. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
24. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. Segformer: Simple and efficient design for semantic segmentation with transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 12077–12090.
25. Dai, Z.; Liu, H.; Le, Q.V.; Tan, M. CoAtNet: Marrying convolution and attention for all data sizes. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 3965–3977.
26. Yan, H.; Li, Z.; Li, W.; Wang, C.; Wu, M.; Zhang, C. ConTNet: Why not use convolution and transformer at the same time? *arXiv* **2021**, arXiv:2104.13497.

27. Jin, Y.; Han, D.; Ko, H. TrSeg: Transformer for semantic segmentation. *Pattern Recognit. Lett.* **2021**, *148*, 29–35. [CrossRef]
28. Zhang, X.; Zhang, Y. Conv-PVT: A fusion architecture of convolution and pyramid vision transformer. *Int. J. Mach. Learn. Cyber.* **2023**, *14*, 2127–2136. [CrossRef]
29. Gao, L.; Liu, H.; Yang, M.; Chen, L.; Wan, Y.; Xiao, Z.; Qian, Y. Stransfuse: Fusing swin Transformer and convolutional neural network for remote sensing image semantic segmentation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 10990–11003. [CrossRef]
30. Wang, L.; Li, R.; Wang, D.; Duan, C.; Wang, T.; Meng, X. Transformer Meets Convolution: A Bilateral Awareness Network for Semantic Segmentation of Very Fine Resolution Urban Scene Images. *Remote Sens.* **2021**, *13*, 3065. [CrossRef]
31. Zhang, W.; Tan, Z.; Lv, Q.; Li, J.; Zhu, B.; Liu, Y. An Efficient Hybrid CNN-Transformer Approach for Remote Sensing Super-Resolution. *Remote Sens.* **2024**, *16*, 880. [CrossRef]
32. Yao, M.; Zhang, Y.; Liu, G.; Pang, D. SSNet: A Novel Transformer and CNN Hybrid Network for Remote Sensing Semantic Segmentation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 3023–3037. [CrossRef]
33. Segal-Rozenhaimer, M.; Li, A.; Das, K.; Chirayath, V. Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks (CNN). *Remote Sens. Environ.* **2020**, *237*, 111446. [CrossRef]
34. Pu, W.; Wang, Z.; Liu, D.; Zhang, Q. Optical Remote Sensing Image Cloud Detection with Self-Attention and Spatial Pyramid Pooling Fusion. *Remote Sens.* **2022**, *14*, 4312. [CrossRef]
35. Li, K.; Ma, N.; Sun, L. Cloud Detection of Multi-Type Satellite Images Based on Spectral Assimilation and Deep Learning. *Int. J. Remote Sens.* **2023**, *44*, 3106–3121. [CrossRef]
36. Pasquarella, V.J.; Brown, C.F.; Czerwinski, W.; Rucklidge, W.J. Comprehensive Quality Assessment of Optical Satellite Imagery Using Weakly Supervised Video Learning. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, 17–24 June 2023; pp. 2125–2135.
37. Hughes, M.J.; Hayes, D.J. Automated Detection of Cloud and Cloud Shadow in Single-Date Landsat Imagery Using Neural Networks and Spatial Post-Processing. *Remote Sens.* **2014**, *6*, 4907–4926. [CrossRef]
38. Li, J.; Wu, Z.; Hu, Z.; Jian, C.; Luo, S.; Mou, L.; Zhu, X.X.; Molinier, M. A lightweight deep learning-based cloud detection method for Sentinel-2A imagery fusing multiscale spectral and spatial features. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–19. [CrossRef]
39. He, Q.; Sun, X.; Yan, Z.; Fu, K. DABnet: Deformable contextual and boundary-weighted network for cloud detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5601216. [CrossRef]
40. López-Puigdollers, D.; Mateo-García, G.; Gómez-Chova, L. Benchmarking Deep Learning Models for Cloud Detection in Landsat-8 and Sentinel-2 Images. *Remote Sens.* **2021**, *13*, 992. [CrossRef]
41. Kim, B.; Oh, H. AI Training Dataset for Cloud Detection of KOMPSAT Images. *GEO DATA* **2020**, *2*, 56–62. [CrossRef]
42. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning. PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
43. Huang, H.M.; Lin, L.F.; Tong, R.F.; Hu, H.J.; Zhang, Q.W.; Iwamoto, Y.; Han, X.H.; Chen, Y.W.; Wu, J. Unet 3+: A Full-Scale Connected Unet for Medical Image Segmentation. In Proceedings of the ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, Virtual, 4–8 May 2020; pp. 1055–1059.
44. Mo, J.; Seong, S.; Oh, J.; Choi, J.J.I.A. SAUNet3+ CD: A Siamese-attentive UNet3+ for change detection in remote sensing images. *IEEE Access* **2022**, *10*, 101434–101444. [CrossRef]
45. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
46. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. Mnasnet: Platform-Aware Neural Architecture Search for Mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2820–2828.
47. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
48. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3349–3364. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.