forests

MDPI

Article

Evaluation of Hyperparameter Combinations of the U-Net Model for Land Cover Classification

Yongkyu Lee [1], Woodam Sim [1], Jeongmook Park [2] and Jungsoo Lee [1,*]

[1] Department of Forest Management, Kangwon National University, Chuncheon 24341, Korea
[2] Forest ICT Research Center, National Institute of Forest Science, Seoul 02455, Korea
* Correspondence: jslee72@kangwon.ac.kr

Abstract: The aim of this study was to select the optimal deep learning model for land cover classification through hyperparameter adjustment. A U-Net model with encoder and decoder structures was used as the deep learning model, and RapidEye satellite images and a sub-divided land cover map provided by the Ministry of Environment were used as the training dataset and label images, respectively. According to different combinations of hyperparameters, including the size of the input image, the configuration of convolutional layers, the kernel size, and the number of pooling and up-convolutional layers, 90 deep learning models were built, and the model performance was evaluated through the training accuracy and loss, as well as the validation accuracy and loss values. The evaluation results showed that the accuracy was higher with a smaller image size and a smaller kernel size, and was more dependent on the convolutional layer configuration and number of layers than the kernel size. The loss tended to be lower as the convolutional layer composition and number of layers increased, regardless of the image size or kernel size. The deep learning model with the best performance recorded a validation loss of 0.11 with an image size of 64 × 64, a convolutional layer configuration of C→C→C→P, a kernel size of 5 × 5, and five layers. Regarding the classification accuracy of the land cover map constructed using this model, the overall accuracy and kappa coefficient for three study cities showed high agreement at approximately 82.9% and 66.3%, respectively.

Keywords: deep learning; land cover classification; U-net; hyperparameter

Citation: Lee, Y.; Sim, W.; Park, J.; Lee, J. Evaluation of Hyperparameter Combinations of the U-Net Model for Land Cover Classification. *Forests* **2022**, *13*, 1813. https://doi.org/10.3390/f13111813

Academic Editors: Sang-Kyun Han, Heesung Woo, Jae-Heun Oh and Rodolfo Picchio

Received: 9 September 2022
Accepted: 25 October 2022
Published: 31 October 2022

1. Introduction

The use of remote sensing technology to acquire large-scale and periodic data for forest resource monitoring has become widespread [1,2]. In particular, remote sensing-based research in forestry is being actively conducted on forest carbon estimation and land-use change prediction with medium resolution satellite images and high resolution images.

The classification of land cover involves pixel-based or segmentation-based supervised classification, unsupervised classification, and object-based classification. Since 2010, land cover classification research has also employed artificial intelligence techniques such as machine learning and deep learning, with the former involving algorithms such as deep learning, support vector machine, and random forest. The image processing using deep learning is largely divided into the classification, localization, and segmentation of images. Classification involves identifying what the image shows, whereas localization focuses on detecting what is in the image and where it is located. Segmentation describes the process of identifying and classifying objects within an image (in units of pixels) in a more precise manner than that afforded by simple detection; this process is divided into semantic segmentation and instance segmentation according to the presence or absence of object separation [3]. Representative deep learning models that employ the semantic segmentation method (e.g., U-Net, DeepLab, FCN, SegNet, and PSPNet) are configured in the form of an encoder-decoder network, which exhibits high performance in the field of

land cover classification. Here, an encoder and a decoder are connected as a pair, where the encoder compresses the input data and summarizes the information, and the decoder returns the compressed information passed by the encoder [4,5].

When determining land-use changes for the purpose of calculating national greenhouse gas emissions, it is necessary to employ the statistical calculation methods suggested by the Intergovernmental Panel on Climate Change to ensure a consistent representation of land use. To achieve spatial and temporal consistency, sample-based and wall-to-wall methods can be used. The former calculates the land cover area by sampling at regular intervals, whereas the latter calculates the land cover area in units of space. Park et al. [6] suggested that wall-to-wall methods, such as semantic segmentation, are effective for advanced forest inventory calculation because of their spatial accuracy.

To ensure high accuracy in CNN models, it is necessary to set optimal hyperparameters. The hyperparameter values must be precisely determined for each dataset. However, previous studies have mainly focused on investigating hyperparameters related to learning processes, except some cases related network architecture of deep learning models. In addition, network architecture search (NAS) was designed as a new network architecture to solve classification problems. Study on NAS is mainly conducted in the field of classification. Application of NAS has been limited in the field of semantic segmentation [7–10]. In this context, this research aimed to search hyperparameters for constructing an optimal network architecture based on the U-net model and map a land cover map using the optimal model.

2. Materials and Methods

2.1. Study Site

Three cities in three different provinces of South Korea were chosen as the study sites: Chuncheon, Gimcheon, and Suncheon. Chuncheon, Gimcheon, and Suncheon have a population of 285,575, 140,065, and 280,478, respectively [11]. In terms of the distribution of land use by city, based on the sub-divided land cover map (Sd LCM) provided by the Ministry of Environment (2018), forest occupied the largest area of Chuncheon (approximately 73.3% of the total area), followed by grass (approximately 7.8%) and agricultural land (approximately 6.1%). Owing to the abundance of lakes and rivers, water occupied approximately 5.5% of Chuncheon, representing the highest proportion of water among the three cities. In Gimcheon, forest occupied the largest area (approximately 66.0%), followed by agricultural land (approximately 17.2%), and grass (approximately 8.1%); Gimcheon had the highest proportion of agricultural land among the three cities. In Suncheon, forest occupied the largest area (approximately 61.2%), followed by agricultural land (approximately 14.5%), and grass (approximately 13.3%); Suncheon had the highest proportion of grass among the three cities (Table 1).

Table 1. Current land cover distribution in the three study cities based on Sd LCM (unit: ha).

Division	Used Area	Agricultural Land	Forest	Grass	Wet Land	Barren	Water	Total
Chuncheon	3991 (3.6%)	6797 (6.1%)	81,739 (73.3%)	8662 (7.8%)	907 (0.8%)	3285 (2.9%)	6177 (5.5%)	111,559
Gimcheon	4266 (4.2%)	17,306 (17.2%)	66,367 (66.0%)	8104 (8.1%)	905 (0.9%)	2836 (2.8%)	770 (0.8%)	100,554
Sumcheon	5103 (5.6%)	13,190 (14.5%)	55,632 (61.2%)	12,120 (13.3%)	915 (1.0%)	1601 (1.8%)	2407 (2.6%)	90,968

2.2. Materials

RapidEye images with a spatial resolution of 6.5 m (Chuncheon: 21 May 2018, Gimcheon: 2 August 2018, Suncheon: 22 May 2019) were used as the satellite image data. RapidEye images included the red-edge band as well as four other bands: red, blue, green, and near infrared (NIR), which can be effectively used to acquire data for forestry and crop monitoring [12]. RapidEye images are classified into Level 1B, 3A, and 3B according to the

preprocessing level. In this study, we used 3A_Analytic image data (spatial resolution of 5 m), which were corrected with orthographic images according to the following processes: radiation correction, sensor correction, and geometric correction.

To construct a deep learning model for land cover classification, labeled image data that can be used as ground truth data for land cover are required. In this study, the Sd LCM was used. Sd LCM (spatial resolution of 1 m) is a thematic map constructed by visual interpretation of aerial orthographic images (spatial resolution of 0.25 m), which provides information on 41 items.

2.3. Methods

For land cover classification using deep learning, the U-Net model was used as the base model, and a deep learning algorithm was built by adjusting the convolutional layer configuration, kernel size, and number of layers in the U-Net basic model. We then constructed, compared, and evaluated a total of 90 deep learning models with different combinations of input data size and deep learning architecture (Figure 1). These steps are described in the following subsections.

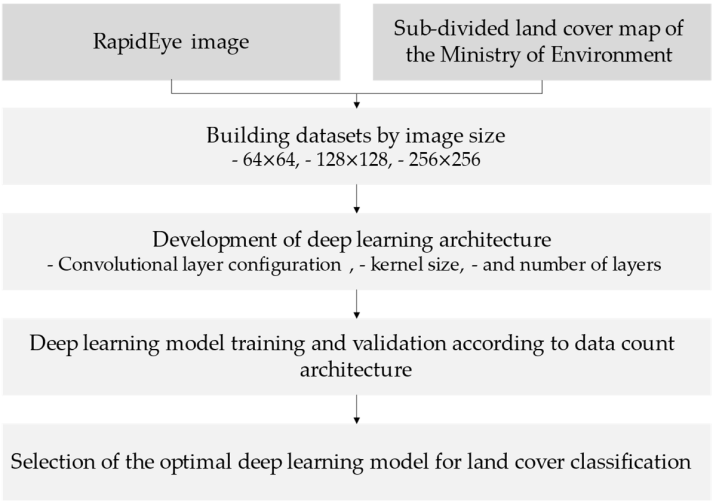

Figure 1. Optimal algorithm development process for land-use and land cover maps.

2.3.1. Base Model Selection

Among the major deep learning models that employ semantic segmentation for land cover and forest classification, U-Net was selected as the base model in this study. Unlike general CNN networks, U-Net models are trained by cutting images into patch units rather than using a sliding window method to learn the images. As spatial pattern information can be used for classification, this model has the advantage of being effective for high-resolution video image classification [13].

A U-Net is a structure configured for the purpose of image segmentation, whose network configuration resembles the letter "U". The left side of the U-Net architecture is divided into an encoder layer consisting of a convolution layer and a max pooling layer, and a decoder layer consisting of a convolution layer and an up-convolution layer [14,15]. In addition, the skip-connection method and weighted cross entropy method used in the U-Net model were judged to effectively extract the characteristics of objects in land cover and forest classification. Therefore, we established an optimal deep learning model for land cover and forest classification by selecting U-Net as the base model and adjusting the hyperparameters.

2.3.2. Dataset Construction According to Spatial Resolution

The U-Net model is a supervised learning method. The training data for the U-Net model must be built as a dataset that includes the ground truth value, called a labeled

image, in the input image data [16]. RapidEye images were used as the input image data, and the Sd LCM was used to set the labeled images. Forty-one items of the Sd LCM were regrouped into major land cover classification categories according to the spatial resolution of the training data. That is, the labeled images were configured as used area, agricultural land, forest, grass, wet land (waterside vegetation), barren, and water images. The labeled images were rasterized with the same resolution (5 m) and the same pixel size as the RapidEye images. Binary values were set for each classification category, with seven bands equal to the number of categories.

Finally, three datasets with sizes of 64 × 64 pixels, 128 × 128 pixels, and 256 × 256 pixels were constructed to evaluate the effect of the deep learning model according to image size. The model was trained using 7% of the constructed dataset and validated using 3% of the dataset, determined by random sampling.

2.3.3. Hyperparameter Selection for Deep Learning Model Architecture

Deep learning CNN models have multiple hyperparameters that need to be defined [17], including parameters that affect learning, such as the number of training times, training rate, and batch size, and parameters that affect the architecture of the model, such as kernel size and convolutional layer configuration. As it was difficult to consider all hyperparameters, a total of 30 deep learning architectures were constructed in this study by combining three convolutional layer configurations (C→P, C→C→P, and C→C→C→P), two kernel sizes (3 × 3 and 5 × 5), and five pooling and up-convolutional layers (1, 2, 3, 4, and 5). We then compared the accuracy of the model according to the parameters that affect the model architecture. As for the parameters that affect the learning of the model, the number of training times was fixed at 200 times, the training rate was fixed at 0.01, and the pooling and up-convolution filter size was fixed at 2 × 2. The default U-Net value was used for the number of channels per layer (Figure 2).

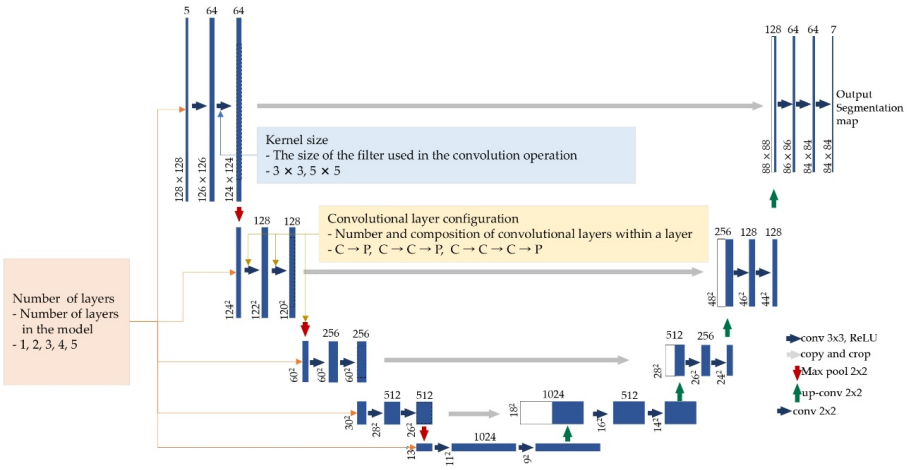

Figure 2. Construction of the deep learning model by adjusting the input image size and hyperparameters.

2.3.4. Training and Validation of Deep Learning Models

A total of 90 deep learning models, constructed by combining three datasets and 30 model architectures, were trained and validated. In the training process, gray-scale images were used in the input layer of the original model, whereas five bands (red, green, blue, red-edge, and NIR) were used in this study, requiring the number of input channels to be changed from one to five. Deep learning models were trained using training images and labeled data. When the first training image was input to the deep learning architecture, the image was classified through a convolutional layer, and the parameters were adjusted by comparing the classification results with the labeled data, which contained the ground truth value. Training images were sequentially input to the deep learning architecture, and

parameters were adjusted repeatedly until all training images were input. Once all training images were input, the parameters were fixed. When the validation images were input, the model output the classification results, and the accuracy and loss were calculated through a comparison with the labeled data of the validation image to evaluate model performance. Each cycle of the training and validation process is called an epoch; to develop the deep learning model, model training and validation were performed for up to 200 epochs. The performance of the trained model was evaluated by calculating the training accuracy, training loss, validation accuracy, and validation loss.

2.3.5. Evaluation of Deep Learning Model for Land Cover Mapping

After training, the training rate and training loss value of the deep learning model were calculated. Then, the validation accuracy and validation loss value were calculated using the validation dataset. Evaluating the performance of a deep learning model based on the training rate and training loss may lead to selection of a model that is overfitted to the training data; however, evaluating model performance based on the validation accuracy and validation loss may lead to selection of a model with a low training rate. Therefore, in this study, the optimal deep learning model was selected by considering all four parameters. Models in the top 10% of all 90 deep learning models in terms of training accuracy, training loss, validation accuracy, and validation loss were selected as the optimal models.

2.3.6. Consistency between Deep Learning-Based Land Cover Map and Sd LCM

To evaluate the classification accuracy of the deep learning-based land cover map, we evaluated its consistency with the Sd LCM. Specifically, the accuracy was evaluated by calculating the overall accuracy (OA) and kappa coefficient using a confusion matrix (Table 2) [18,19].

Table 2. Summary of accuracy metrics used in the object detection, where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

Accuracy Metrics	Equation
Overall accuracy (OA)	$\frac{TP+TN}{TP+FP+TN+FN}$
Precision (P)	$\frac{TP}{TP+FP}$
Recall (R)	$\frac{TP}{TP+FN}$
F1	$2 \times \frac{P \times R}{P+R}$
Kappa	$\frac{TA-P_e}{1-P_e}$, *where* $p_e = \frac{(TP+FN)\times(TP+FP)(FP+TN)\times(FN+TN)}{(TP+FN+TF+FP)^2}$

3. Results and Discussion

3.1. Deep Learning Model Architecture

Thirty deep learning model architectures were developed according to the convolutional layer configuration, kernel size, and number of layers. According to the U-Net base model, the models were composed of 64 layers with 23,382,663 parameters. The model with the shallowest structure among the developed architectures was that with a convolutional layer configuration of C→P, a kernel size of 3 × 3, and one layer, which consisted of nine layers with 127,687 parameters. The model with the deepest structure among the developed architectures was that with a convolutional layer configuration of C→C→C→P, a kernel size of 5 × 5, and five layers, which consisted of 87 layers with 79,860,999 parameters.

3.2. Evaluation of Training and Validation Accuracies

3.2.1. Comparison of Training Accuracy and Training Loss

The training accuracy increased as the image size and kernel size decreased, but was more affected by the convolutional layer configuration and number of layers than the

kernel size. The training accuracy was greatest with an image size of 64 × 64, whereas the training accuracies for image sizes of 128 × 128 and 256 × 256 were similar. When the overall convolutional layer composition was C→C→C→P or C→P, the training accuracy increased as the number of layers increased. The influence of the number of layers seemed greater than that of the convolutional layer configuration. Similar to the training accuracy, the training loss increased as the image size and kernel size decreased. Moreover, the training accuracy was greatest with an image size of 64 × 64, and approximately 0.2 higher than that for image sizes of 128 × 128 and 256 × 256. When the overall convolutional layer composition was C→P, the training loss decreased as the number of layers increased. The difference in training loss according to the number of layers was greater than that according to the convolutional layer configuration.

Sameen et al. [20] reported that the classification performance increased as the input data size increased. However, the opposite trend was found in this study. Sameen et al. [20] used high resolution aerial photos as the input images, which had relatively small sizes (3 × 3, 5 × 5, 7 × 7, 9 × 9, 11 × 11, and 13 × 13). When training a model based on high resolution aerial photos, the features of each object in images as small as 3 × 3 may not be extracted well. Conversely, the 64 × 64 input images used in this study, with a spatial resolution of 5 m, corresponded to a width of approximately 300 m; therefore, they were more effective for extracting the features of each object in the training images than the 256 × 256 images, which corresponded to a width of approximately 1.3 km. The larger the image size, the more likely it is to contain the forest area object with the highest distribution within the study site. If the forest object is widely distributed in the image, the amount of training for each object could differ as the forest will be learned to a much greater extent than objects in the other land categories. Therefore, the best training was observed in this study when the size of the image was 64 × 64.

3.2.2. Comparison of Verification Accuracy and Verification Loss between Hyperparameter Combinations

Similar to the training accuracy, the verification accuracy tended to increase as the image size and kernel size decreased; however, the verification accuracy was lower than the training accuracy. The average difference between the training accuracy and validation accuracy of models with an image size of 64 × 64 and five layers was 0.02, whereas the average difference for models with an image size of 256 × 256 and one layer was approximately 0.15, that is, approximately seven times greater.

The verification loss decreased as the number of layers increased, regardless of the image size or kernel size, and the verification loss was higher than the training loss. In particular, when the number of layers was three or less, the verification loss increased by an average value of 0.2 or more. Thus, to ensure consistent accuracy, four or more layers should be used in the model.

3.2.3. Comparison of Accuracy and Analysis Time for Each Model

In terms of the validation loss, which is widely used for model evaluation and selection, the lowest value (0.11) was obtained for the model with an image size of 64 × 64, convolutional layer configuration of C→C→C→P, kernel size of 5 × 5, and five layers, and the highest value was obtained for the model with an image size of 128 × 128, convolutional layer configuration of C→P, kernel size of 3 × 3, and two layers (Table 3). According to a comparison of the training time of each model, the training time required for construction was shortest for the model with an image size of 256 × 256, convolutional layer configuration of C→P, and kernel size of 3 × 3, and approximately 23% shorter than that for the model with an image size of 64 × 64, convolutional layer configuration of C→C→C→P, and kernel size of 5 × 5 (Table 4).

Table 3. Distribution of accuracy and loss values for the training and validation data of the deep learning model.

Image Size	Kernel Size	Convolutional Layer Configuration	Training Accuracy					Training Loss					Verification Accuracy					Verification Loss				
			Number of Layers					Number of Layers					Number of Layers					Number of Layers				
			1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
64 × 64	3 × 3	C→P	0.83	0.86	0.89	0.91	0.91	0.12	0.10	0.08	0.06	0.07	0.81	0.83	0.86	0.87	0.88	0.33	0.18	0.26	0.19	0.16
		C→C→P	0.85	0.87	0.89	0.88	0.88	0.11	0.09	0.08	0.09	0.08	0.82	0.85	0.86	0.84	0.86	0.19	0.15	0.18	0.23	0.17
		C→C→C→P	0.86	0.87	0.87	0.87	0.87	0.10	0.09	0.09	0.09	0.09	0.71	0.82	0.84	0.84	0.84	0.80	0.21	0.19	0.16	0.17
	5 × 5	C→P	0.84	0.86	0.91	0.92	0.91	0.12	0.10	0.07	0.06	0.07	0.82	0.85	0.87	0.86	0.87	0.15	0.15	0.21	0.25	0.19
		C→C→P	0.85	0.88	0.88	0.87	0.87	0.10	0.08	0.08	0.09	0.09	0.84	0.86	0.85	0.80	0.85	0.14	0.15	0.20	0.25	0.14
		C→C→C→P	0.86	0.88	0.88	0.87	0.86	0.10	0.09	0.08	0.09	0.10	0.77	0.83	0.86	0.83	0.86	0.30	0.21	0.13	0.18	0.11
128 × 128	3 × 3	C→P	0.83	0.85	0.85	0.87	0.86	0.12	0.11	0.10	0.09	0.10	0.77	0.63	0.63	0.84	0.82	0.26	1.41	1.22	0.18	0.26
		C→C→P	0.84	0.85	0.86	0.85	0.86	0.11	0.10	0.10	0.11	0.10	0.71	0.83	0.81	0.82	0.80	0.33	0.19	0.22	0.32	0.24
		C→C→C→P	0.85	0.86	0.85	0.86	0.85	0.10	0.10	0.10	0.10	0.10	0.84	0.83	0.71	0.75	0.83	0.26	0.17	0.53	0.33	0.16
	5 × 5	C→P	0.84	0.84	0.86	0.87	0.86	0.12	0.11	0.10	0.09	0.10	0.72	0.65	0.79	0.81	0.85	0.43	0.65	0.26	0.22	0.13
		C→C→P	0.84	0.86	0.86	0.86	0.85	0.12	0.10	0.10	0.10	0.11	0.61	0.82	0.81	0.83	0.82	0.80	0.18	0.22	0.24	0.19
		C→C→C→P	0.85	0.85	0.85	0.85	0.85	0.11	0.10	0.10	0.10	0.11	0.81	0.83	0.69	0.75	0.84	0.19	0.16	0.47	0.29	0.13
256 × 256	3 × 3	C→P	0.83	0.85	0.86	0.86	0.88	0.12	0.10	0.09	0.09	0.08	0.81	0.84	0.82	0.81	0.85	0.36	0.15	0.24	0.23	0.16
		C→C→P	0.85	0.85	0.86	0.87	0.86	0.10	0.10	0.10	0.09	0.10	0.64	0.8	0.78	0.84	0.82	0.45	0.34	0.28	0.15	0.19
		C→C→C→P	0.85	0.85	0.86	0.86	0.85	0.10	0.10	0.09	0.10	0.10	0.66	0.83	0.84	0.81	0.82	0.36	0.14	0.14	0.25	0.15
	5 × 5	C→P	0.83	0.86	0.87	0.87	0.86	0.12	0.10	0.09	0.09	0.09	0.81	0.67	0.84	0.83	0.83	0.17	0.43	0.15	0.17	0.2
		C→C→P	0.85	0.85	0.86	0.85	0.86	0.10	0.10	0.09	0.10	0.09	0.65	0.76	0.82	0.81	0.84	0.64	0.48	0.21	0.19	0.14
		C→C→C→P	0.85	0.85	0.85	0.85	0.85	0.10	0.10	0.10	0.10	0.10	0.62	0.83	0.79	0.61	0.83	0.6	0.19	0.2	0.57	0.15

Table 4. Analysis of training time in deep learning model with hyperparameter combinations (unit: min).

Image Size	Kernel Size	Convolutional Layer Configuration	Number of Layers				
			1	**2**	**3**	**4**	**5**
64 × 64	3 × 3	C→P	183	185	187	189	193
		C→C→P	185	196	200	203	204
		C→C→C→P	185	204	209	214	215
	5 × 5	C→P	178	185	189	193	204
		C→C→P	181	196	203	213	218
		C→C→C→P	186	209	213	233	243
128 × 128	3 × 3	C→P	174	177	179	180	180
		C→C→P	177	179	182	184	184
		C→C→C→P	178	182	185	188	188
	5 × 5	C→P	172	175	178	180	182
		C→C→P	174	178	184	188	189
		C→C→C→P	177	182	192	197	200
256 × 256	3 × 3	C→P	161	167	168	169	170
		C→C→P	167	172	174	176	175
		C→C→C→P	169	175	181	184	187
	5 × 5	C→P	163	173	174	177	181
		C→C→P	171	179	183	189	189
		C→C→C→P	175	185	190	210	223

Compared to the verification loss for an image size of 64 × 64, the average verification loss for image sizes of 256 × 256 and 128 × 128 increased by 0.68 times and 1.28 times, respectively. Moreover, the time required for analysis decreased by approximately 19% for an image size of 256 × 256. The average verification loss according to the kernel size was 0.03, showing no significant difference in accuracy. Kim et al. [21] conducted a crop classification analysis by constructing deep learning CNN models according to kernel size (2 × 2, 3 × 3, and 5 × 5), and reported that classification accuracy increased as the kernel size decreased. Moreover, in their study, more detailed feature information could be extracted from the input data when the kernel size was smaller. However, in our study, the kernel size did not affect the classification performance in the 5-m images. Considering that the required time increased by approximately 1.04 times when the kernel size was 5 × 5, a kernel size of 3 × 3 is most appropriate in terms of cost and efficiency. The average validation loss according to the convolutional layer configuration showed a difference of up to 0.05. With each convolution layer added, the average loss decreased by approximately 1.1 times. When the convolutional layer configuration was C→C→C→P, the required time increased by approximately 1.1 times compared to that of C→P. Furthermore, the average validation loss decreased as the number of layers increased. With each additional layer, the verification loss was approximately 0.2 smaller than that with one layer applied. However, when five layers were applied, the time required increased by approximately 1.12 times from that with one layer applied. Therefore, considering both the efficiency and accuracy, the optimal number of layers is four or more (Figure 3).

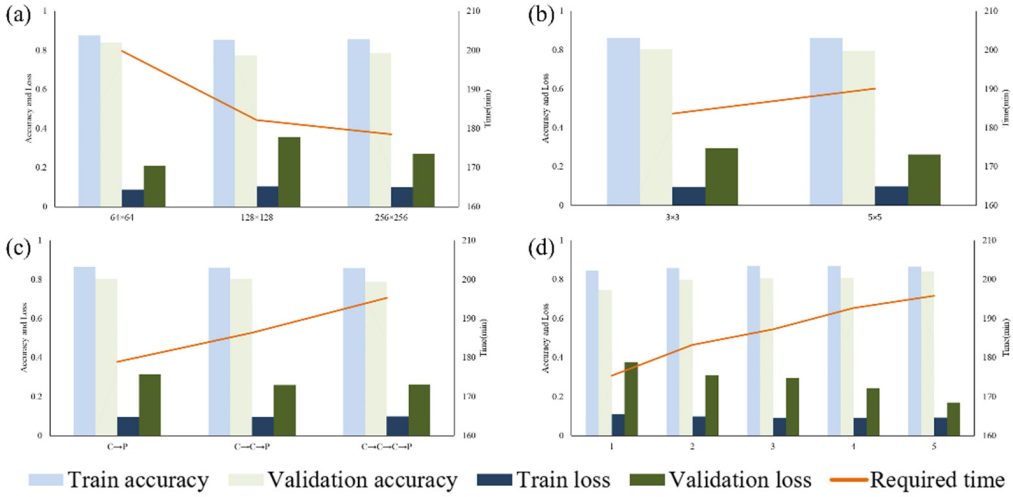

Figure 3. Comparison of the average accuracy and training time according to (**a**) image size, (**b**) kernel size, (**c**) convolutional layer configuration, and (**d**) number of layers.

3.2.4. Consistency between the Deep Learning-Based Land Cover Maps and Sd LCM

A land cover map was constructed by applying the model with the best performance among the evaluated model with an image size of 64 × 64, convolutional layer configuration of C→C→C→P, kernel size of 5 × 5, and five layers. According to this map, in Chuncheon, forest occupied approximately 74.9% of the total area, followed by grass (approximately 7.0%) and agricultural land (approximately 5.8%). A comparison with the Sd LCM and the area distribution by category indicated similar distribution results for each category, with overestimation of the forest area by up to approximately 1.6% by the deep learning land cover map. As for Gimcheon, forest occupied the largest area (approximately 70.4%), followed by agricultural land (approximately 14.9%) and grass (approximately 5.5%). The forest area was overestimated by up to approximately 4.4%, representing the largest difference among the three cities. In Suncheon, forest occupied the largest area (approximately 61.2%), followed by grass (approximately 15.3%) and agricultural land (approximately 14.2%), and the area of grass was overestimated by approximately 2.0% (Figure 4).

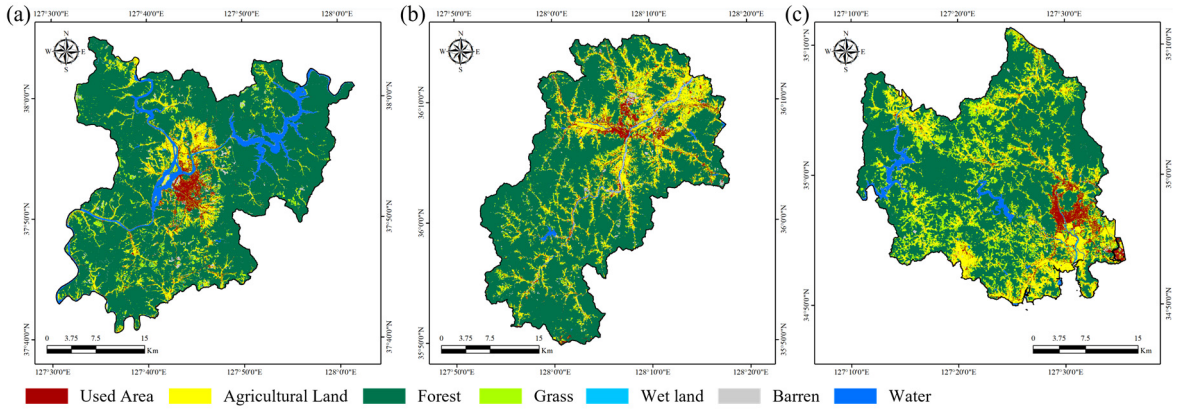

Figure 4. The deep learning—based land cover map in this study for (**a**) Chuncheon, (**b**) Gimcheon, and (**c**) Suncheon.

Figure 5 is more detailed information on the results of this study. Each detailed land cover map is generated by merging 320 of the classified results. As seen Figure 5, the performance of the land cover map was well generated except for some misclassification, including small object classification, such as a narrow road within the agricultural land. This misclassification was occluded due to the difference in resolution between the labeled

data and the satellite image. Thus, pre-processing is highly required. On the other hand, there was a part where the boundary part did not match in the process of merging the images. Therefore, we need additional research about a post-processing method in the process of merging output images using a deep learning model.

As for the consistency between the land cover map constructed by applying the optimal model and the Sd LCM, the OA and kappa coefficient for the three cities indicated high agreement of approximately 82.9% and 66.3%, respectively (Figure 6). Kim et al. [22] reported an accuracy of 65% when single-temporal images were applied to the U-Net model, which was improved by up to 82.6% by applying multi-temporal images. In this study, the results were obtained by using multi-temporal images after performing model training with single-temporal images of three regions to improve the architecture of the model. The classification performance was higher than the classification accuracy standard based on land cover map preparation guidelines (75%). However, the accuracy of the model varied with the study area, with Chuncheon exhibiting higher classification accuracy than the other two cities. Thus, the distribution of land cover by category, along with the difference in area by region, clearly affects the model construction.

As for model accuracy according to land cover category, the model classified the largest forest area with a high accuracy (F1-Score over 90%), regardless of the area. The F1-score of the entire water area also showed high accuracy of approximately 89.5%; however, the F1-score for wet land area was only 16.2%, representing the lowest accuracy among the seven land cover categories. The wet land category includes mosses, lichens, and riparian vegetation around rivers, as well as small streams, with similar spectral characteristics to the water category. The accuracy was likely low because the distribution of lichens may vary depending on the time of each image. The agricultural land and used area categories showed an accuracy score of approximately 67.5% and 59.4%, respectively, with similar F1-scores among the three cities (within approximately 3%), indicating high consistency. However, the grass and barren classification showed lower accuracy than the other categories, of approximately 43.4% and 32.3%, respectively. Gimcheon showed the lowest classification accuracy. Here, river side and riparian vegetation, which would be classified as wet land areas in the other cities, were classified as grassland or barren in the Sd LCM, which was used as the label image. Moreover, logging areas corresponding to forest were classified as grass, contributing to the low accuracy for grass or barren.

Karra et al. [23] conducted land cover classification analysis using a U-Net model based on a Sentinel-2 image with a spatial resolution of 10 m. In the study site with the highest classification accuracy, they reported high accuracy of approximately 85% or more for used area, forest, agricultural land, and water categories. Conversely, grass, wet land, and barren categories showed low accuracy of less than approximately 20%. In addition, Lee and Lee [24] constructed a classification map using the U-Net model, which classified coniferous forests and rice paddies/fields with a high accuracy of approximately 87.4% and 77.3%, respectively, but classified buildings, cemeteries, grass, and wet lands with an accuracy of less than 50%. Therefore, in this study, the classification accuracy for forest and water categories were similarly high compared to previous studies, whereas that for agricultural land and used areas, such as buildings and streets, was lower than that of previous studies. Agricultural land classified on the Sd LMC (the labeled image) included facilities such as plantation sites and greenhouses; however, the deep learning model classified these regions as used areas, which reduced the classification accuracy. Also, the classification accuracy for grass and barren is generally low in previous studies. Moreover, as cultivated land with active growth after rice planting has similar spectral characteristics to herbaceous plants in the growing season image, the classification accuracy for grass was low. There have also been many previous cases of misclassifying barren as shrubland and grass. Similarly, in this study, barren was the most likely land cover category to be misclassified as grass or forest. Specifically, according to the Sd LCM, riverside and wet land areas were classified as barren, and areas where buildings were being built were classified as used areas or barren. There were also many cases in which barren, such as

rocky and stony fields, were classified as forests; therefore, the training data should be carefully reviewed when using this model.

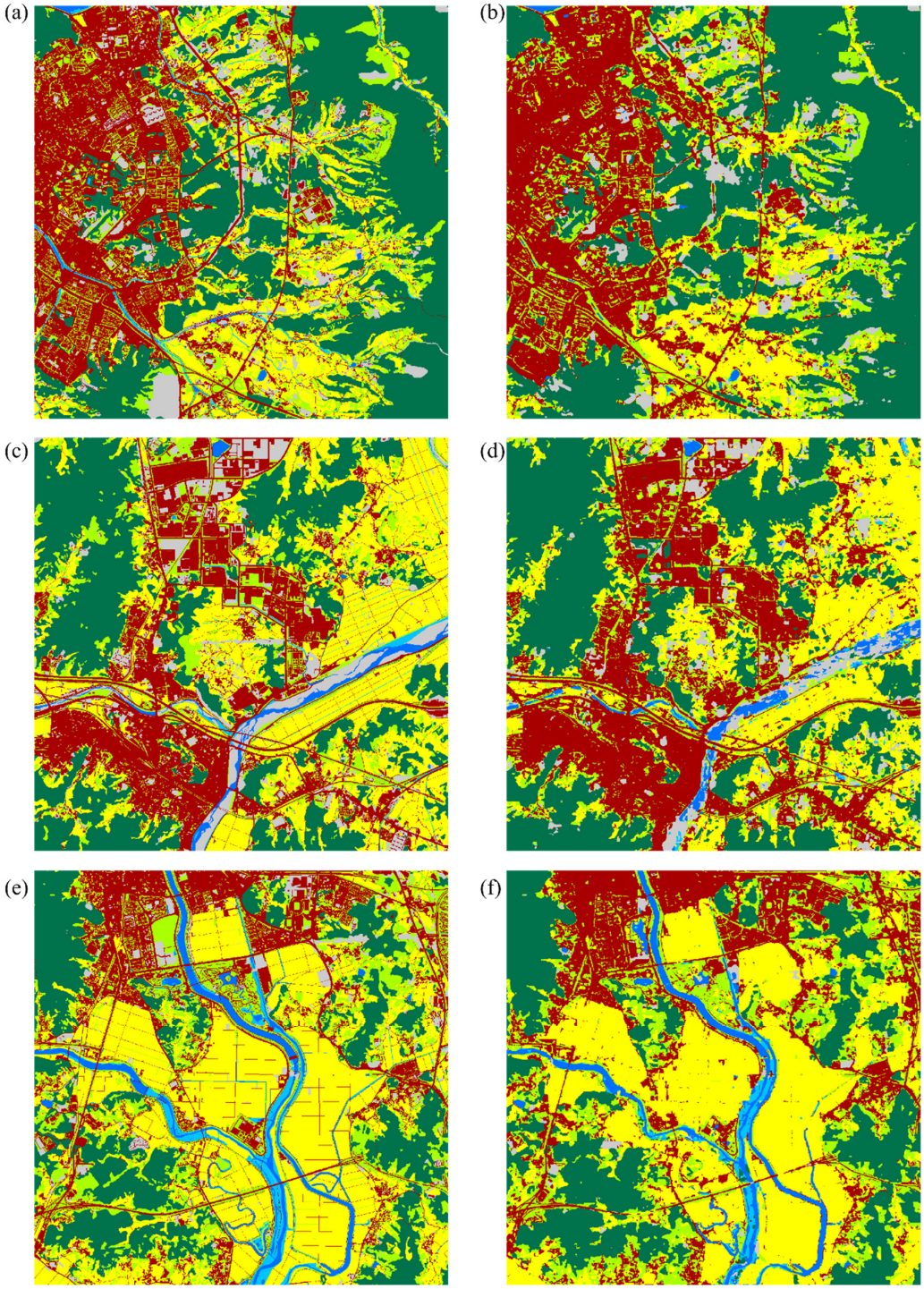

Figure 5. The detail land cover map in the study area for (**a**) label image in Chuncheon, (**b**) output image of deep learning model in Chuncheon, (**c**) Label image in Gimcheon, (**d**) output image of deep learning model in Gimcheon, (**e**) Label image in Suncheon, (**f**) output image of deep learning model in Suncheon.

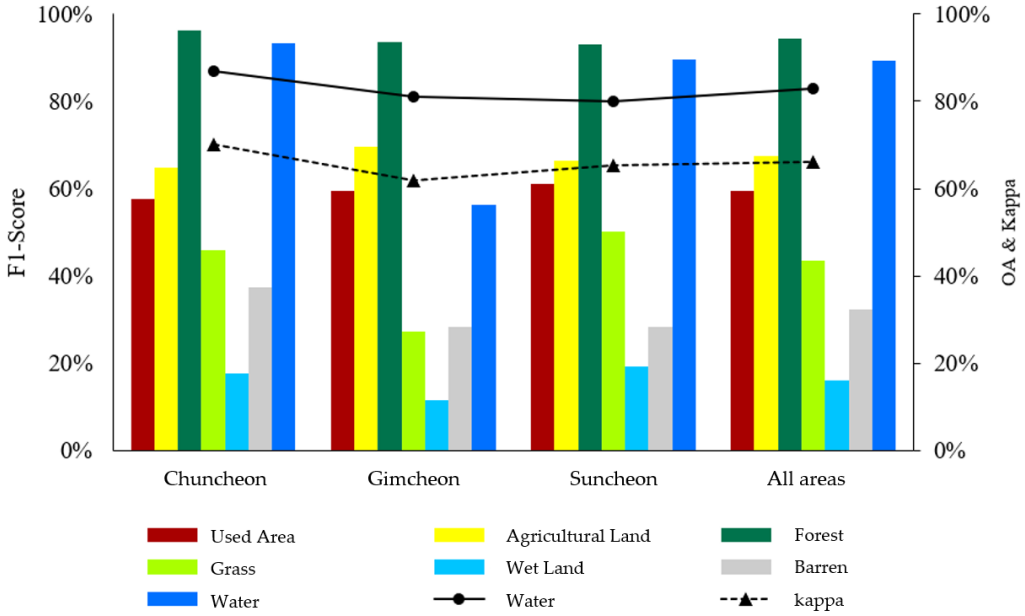

Figure 6. Evaluation of consistency of land cover categories in deep learning-based land cover map.

3.2.5. Effectiveness of Training Data for Deep Learning Model Applications

As the training data used in this study were reclassified into major land cover classification categories, a Sd LCM was used. The forest practice areas were then divided into various categories, such as barren, grass, and forest. As the forest practice areas used as the training data had the most labeled data classified as forest land, the deep learning model was deemed to have effectively classified these forest practice areas as forest. In future, it will be necessary to clearly define the training data categories and construct correct labeled images. Moreover, grass areas such as urban trees, small gardens in apartment complexes, and fences made of shrubs may have been incorrectly classified as used areas in this study. The Sd LCM used as the label image in this study was constructed by visual interpretation of a 25-cm high-resolution aerial photograph; thus, it was capable of classifying street trees, fences, and gardens as grass. However, for the RapidEye images (spatial resolution of 5 m) used as the input image, where these areas were distributed as mixed pixels, they may have been classified as used areas.

4. Conclusions

In this study, we explored the optimal parameters of a U-Net-based deep learning model for land-use classification using RapidEye images. Determining the classification accuracy and required time of the deep learning model according to the hyperparameter combination enabled us to select an optimal model that simultaneously considers the cost, efficiency, and accuracy. The optimal model exhibited high classification accuracy for forest land; thus, it could be used to prepare a land-use change matrix that ensures a spatially and temporally consistent representation of land use according to the wall-to-wall method of the Intergovernmental Panel on Climate Change.

As well as U-Net, various other deep learning models have been developed to solve semantic segmentation problems in land cover mapping. Thus, when constructing a land cover map based on the deep learning model developed in this study, the following specific points should be considered. First, pre-processing is required when using datasets from pre-built data. In U-Net-based models, the labeled image required for training, called the ground truth, was a major factor affecting model performance. In this study, the Sd LCM was used as the ground truth and satellite images were used as the input data. With satellite images, the learning training accuracy will be reduced unless the clouds or shadows are removed through preprocessing. Moreover, as the training data must be acquired easily for each land cover category, even if the training data are randomly

selected, applying the hierarchical sampling method by region and category is expected to improve model performance. Second, when constructing a land cover map based on deep learning, a loss of location information occurs because of the nature of the deep learning algorithm. As training is performed in units of images of a certain size, the accuracy may change at a specific location in the image. Therefore, applying the smoothing method when synthesizing structural changes of the convolutional layer and the classification result output as an image of a certain size should improve the results of land cover mapping. Third, land-use categories with similar reflective characteristics, such as forest practice areas, grasslands, and agricultural land, were difficult to classify accurately, even when using the deep learning model. In this model, only digital number values were used as the input data; however, it may be beneficial to enhance the model using texture information or topographic information. Furthermore, with the recent development of ensemble models, it will be necessary to review combinations of models with high classification accuracy according to land-use categories.

Author Contributions: Conceptualization, J.P. and J.L.; methodology, Y.L. and W.S.; software, Y.L. and W.S.; validation, J.L.; formal analysis, Y.L.; investigation, Y.L.; resources, Y.L.; data curation, Y.L.; writing—original draft preparation, Y.L., W.S. and J.L.; writing—review and editing, J.L.; visualization, W.S.; supervision, J.L.; project administration, J.P. and J.L.; funding acquisition, J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Institution of Forest Science (grant number FM0103-2021-04-2022).

Data Availability Statement: Not applicable.

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

References

1. Santos, A.A.D.; Marcato Junior, J.; Araújo, M.S.; Di Martini, D.R.; Tetila, E.C.; Siqueira, H.L.; Gonçalves, W.N. Assessment of CNN-based methods for individual tree detection on images captured by RGB cameras attached to UAVs. *Sensors* **2019**, *19*, 3595. [CrossRef] [PubMed]
2. Zhang, M.; Liu, N.; Harper, R.; Li, Q.; Liu, K.; Wei, X.; Liu, S. A global review on hydrological responses to forest change across multiple spatial scales: Importance of scale, climate, forest type and hydrological regime. *J. Hydrol.* **2017**, *546*, 44–59. [CrossRef]
3. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A review on deep learning techniques applied to semantic segmentation. *arXiv* **2017**, arXiv:1704.06857.
4. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
5. Solórzano, J.V.; Mas, J.F.; Gao, Y.; Gallardo-Cruz, J.A. Land use land cover classification with U-Net: Advantages of combining Sentinel-1 and Sentinel-2 imagery. *Remote Sens.* **2021**, *13*, 3600. [CrossRef]
6. Park, E.B.; Song, C.H.; Ham, B.Y.; Kim, J.W.; Lee, J.Y.; Choi, S.E.; Lee, W.K. Comparison of sampling and wall-to-wall methodologies for reporting the GHG inventory of the LULUCF sector in Korea. *J. Clim. Chang. Res.* **2018**, *9*, 385–398. [CrossRef]
7. Ma, A.; Wan, Y.; Zhong, Y.; Wang, J.; Zhang, L. SceneNet: Remote sensing scene classification deep learning network using multi-objective neural evolution architecture search. *ISPRS J. Photogramm. Remote Sens.* **2021**, *172*, 171–188. [CrossRef]
8. Wang, J.; Huang, R.; Guo, S.; Li, L.; Zhu, M.; Yang, S.; Jiao, L. NAS-guided lightweight multiscale attention fusion network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 8754–8767. [CrossRef]
9. Wan, Y.; Zhong, Y.; Ma, A.; Wang, J.; Feng, R. RSSM-Net: Remote Sensing Image Scene Classification Based on Multi-Objective Neural Architecture Search. In Proceedings of the IGARSS 2020—2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020; pp. 1369–1372.
10. Jing, W.; Ren, Q.; Zhou, J.; Song, H. AutoRSISC: Automatic design of neural architecture for remote sensing image scene classification. *Pattern Recognit. Lett.* **2020**, *140*, 186–192. [CrossRef]
11. KOSIS (Korean Statistical Information Service). 2022. Available online: https://kosis.kr/index/index.do (accessed on 1 December 2021).
12. Kross, A.; McNairn, H.; Lapen, D.; Sunohara, M.; Champagne, C. Assessment of RapidEye vegetation indices for estimation of leaf area index and biomass in corn and soybean crops. *Int. J. Appl. Earth Obs. Geoinf.* **2015**, *34*, 235–248. [CrossRef]
13. Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 24–49. [CrossRef]
14. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015; pp. 234–241.

15. Song, C.; Wahyu, W.; Jung, J.; Hong, S.; Kim, D.; Kang, J. Urban change detection for high-resolution satellite images using U-Net based on SPADE. *Korean J. Remote Sens.* **2020**, *36*, 1579–1590.
16. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2019.
17. Sinha, A.; Lee, J.; Li, S.; Barbastathis, G. Lensless computational imaging through deep learning. *Optica* **2017**, *4*, 1117–1125. [CrossRef]
18. Huang, M.H.; Rust, R.T. Artificial intelligence in service. *J. Serv. Res.* **2018**, *21*, 155–172. [CrossRef]
19. Rouhi, R.; Jafari, M.; Kasaei, S.; Keshavarzian, P. Benign and malignant breast tumors classification based on region growing and CNN segmentation. *Expert Syst. Appl.* **2015**, *42*, 990–1002. [CrossRef]
20. Sameen, M.I.; Pradhan, B.; Aziz, O.S. Classification of very high resolution aerial photos using spectral-spatial convolutional neural networks. *J. Sens.* **2018**, *2018*, 7195432. [CrossRef]
21. Kim, Y.; Kwak, G.H.; Lee, K.D.; Na, S.I.; Park, C.W.; Park, N.W. Performance evaluation of machine learning and deep learning algorithms in crop classification: Impact of hyper-parameters and training sample size. *Korean J. Remote Sens.* **2018**, *34*, 811–827.
22. Kim, J.; Song, Y.; Lee, W.K. Accuracy analysis of multi-series phenological landcover classification using U-Net-based deep learning model-focusing on Seoul, Republic of Korea. *Korean J. Remote Sens.* **2021**, *37*, 409–418.
23. Karra, K.; Kontgis, C.; Statman-Weil, Z.; Mazzariello, J.C.; Mathis, M.; Brumby, S.P. Global land use/land cover with Sentinel 2 and deep learning. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; pp. 4704–4707.
24. Lee, S.H.; Lee, M.J. A study on deep learning optimization by land cover classification item using satellite imagery. *Korean J. Remote Sens.* **2020**, *36*, 1591–1604.