Hindawi
Journal of Sensors
Volume 2018, Article ID 7195432, 12 pages
https://doi.org/10.1155/2018/7195432

Research Article

Classification of Very High Resolution Aerial Photos Using Spectral-Spatial Convolutional Neural Networks

Maher Ibrahim Sameen, Biswajeet Pradhan, and Omar Saud Aziz

School of Systems, Management and Leadership, Faculty of Engineering and Information Technology, University of Technology Sydney, Building 11, Level 06, 81 Broadway, P.O. Box 123, Ultimo, NSW 2007, Australia

Correspondence should be addressed to Biswajeet Pradhan; biswajeet24@gmail.com

Received 3 March 2018; Revised 17 April 2018; Accepted 6 May 2018; Published 26 June 2018

Academic Editor: Paolo Bruschi

Classification of aerial photographs relying purely on spectral content is a challenging topic in remote sensing. A convolutional neural network (CNN) was developed to classify aerial photographs into seven land cover classes such as building, grassland, dense vegetation, waterbody, barren land, road, and shadow. The classifier utilized spectral and spatial contents of the data to maximize the accuracy of the classification process. CNN was trained from scratch with manually created ground truth samples. The architecture of the network comprised of a single convolution layer of 32 filters and a kernel size of 3×3, pooling size of 2×2, batch normalization, dropout, and a dense layer with Softmax activation. The design of the architecture and its hyperparameters were selected via sensitivity analysis and validation accuracy. The results showed that the proposed model could be effective for classifying the aerial photographs. The overall accuracy and Kappa coefficient of the best model were 0.973 and 0.967, respectively. In addition, the sensitivity analysis suggested that the use of dropout and batch normalization technique in CNN is essential to improve the generalization performance of the model. The CNN model without the techniques above achieved the worse performance, with an overall accuracy and Kappa of 0.932 and 0.922, respectively. This research shows that CNN-based models are robust for land cover classification using aerial photographs. However, the architecture and hyperparameters of these models should be carefully selected and optimized.

1. Introduction

Classifying remote sensing data (especially orthophotos of three bands—red, green, blue (RGB)) with traditional methods is a challenge even though some methods in literature have produced excellent results [1, 2]. The main reason behind is that remote sensing datasets have high intra- and interclass variability and the amount of labeled data is much smaller as compared to the total size of the dataset [3]. On the other hand, the recent advances in deep learning methods like convolutional neural networks (CNNs) have shown promising results in remote sensing image classification especially hyperspectral image classification [4–6]. The advantages of deep learning methods include learning high-order features from the data that are often useful than the raw pixels for classifying the image into some predefined labels. Other advantages of these methods are spatial learning of contextual information from data via feature pooling from a local spatial neighborhood [3].

There are several methods and algorithms that have been adopted by many researchers to efficiently classify a very high-resolution aerial photo and produce accurate land cover maps. Methods such as object-based image analysis (or OBIA) was mostly investigated because of its advantage in very high-resolution image processing via spectral and spatial features. In a recent paper, Hsieh et al. [7] applied aerial photo classification by combining OBIA with decision tree using texture, shape, and spectral feature. Their results achieved an accuracy of 78.20% and a Kappa coefficient of 0.7597. Vogels et al. [8] combined OBIA with random forest classification with texture, slope, shape, neighbor, and spectral information to produce classification maps for agricultural areas. They have tested their algorithm on two datasets, and the results showed the employed methodology

to be effective with accuracies of 90% and 96% for the two study areas, respectively. On the other hand, a novel model was presented by Meng et al. [9], where they applied OBIA to improve vegetation classification based on aerial photos and global positioning systems. Results illustrated a significant improvement in classification accuracy that increased from 83.98% to 96.12% in overall accuracy and from 0.7806 to 0.947 in the Kappa value. Furthermore, Juel et al. [10] showed that random forest with the use of a digital elevation model could achieve relatively high performance for vegetation mapping. In a most recent paper, Wu et al. [2] developed a model based on a comparison between pixel-based decision tree and object-based SVM to classify aerial photos. The object-based support vector machine (SVM) had higher accuracy than that of the pixel-based decision tree. Albert et al. [11] developed classifiers based on conditional random fields and pixel-based analysis to classify aerial photos. Their results showed that such techniques are beneficial for land cover classes covering large, homogeneous areas.

2. Related Works

The success of CNN in the fields like computer vision, language modeling, and speech recognition has motivated the remote sensing scientists to apply it in image classification. There are several works that have been done on CNN for remote sensing image classification [12–15]. This section briefly explains some of these works highlighting their findings and their limitations.

Sun et al. [16] proposed an automated model for feature extraction and classification with classification refinement by combining random forest and CNN. Their combined model could perform well (86.9%) and obtained higher accuracy than the single models. Akar [1] developed a model based on rotation forest and OBIA to classify aerial photos. Results were compared to gentle AdaBoost, and their experiments suggested that their method performed better than the other method with 92.52% and 91.29% accuracies, respectively. Bergado et al. [17] developed deep learning algorithms based on CNN for aerial photo classification in high-resolution urban areas. They used data from optical bands, digital surface models, and ground truth maps. The results showed that CNN is very effective in learning discriminative contextual features leading to accurate classified maps and outperforming traditional classification methods based on the extraction of textural features. Scott et al. [13] applied CNN to produce land cover maps from high-resolution images. Other researchers such as Cheng et al. [12] used CNN as a classification algorithm for scene understanding from aerial imagery. Furthermore, Sherrah [14] and Yao et al. [15] used CNN for semantic classification of aerial images.

This research investigates the development of a CNN model with regularization techniques such as dropout and batch normalization for classifying aerial orthophotos into general land cover classes (e.g., road, building, waterbody, grassland, barren land, shadow, and dense vegetation). The main objective of the research is to run several experiments exploring the impacts of CNN architectures and hyperparameters on the accuracy of land cover classification using aerial photos. The aim is to understand the behaviours of the CNN model concerning its architecture design and hyperparameters to produce models with high generalization capacity.

3. Methodology

This section presents the dataset, preprocessing, and the methodology of the proposed CNN model including the network architecture and training procedure.

3.1. Dataset and Preprocessing

3.1.1. Dataset. To implement the current research, a pilot area was identified based on the diversity of the land cover of the area. The study area is located in Selangor, Malaysia (Figure 1).

3.1.2. Preprocessing

(1) Geometric Calibration. Since the orthophoto was captured by an airborne laser scanning (LiDAR) system, it was essential to calibrate it geometrically to correct the geometric errors. In this step, the data was corrected based on ground control points (GCPs) collected from the field (Figure 2). There were 34 GCPs identified from clearly identifiable points (i.e., road intersections, corners, and power lines). The geometric correction was done in ArcGIS 10.5 software. The steps of geometric correction included identification of transformation points in the orthophoto, application of the least square transformation, and calculation of the accuracy of the process. The selected points were uniformly distributed in the area. After that, the least square method (Kardoulas et al., 1996) was applied to estimate the coefficients, which are essential for the geometric transformation process. After the least square solution, the polynomial equations were used to solve for X, Y coordinates of GCPs and to determine the residuals and RMS errors between the source X, Y coordinates and the retransformed X, Y coordinates.

(2) Normalization. Since the aerial orthophotos have integer digital values and initial weights of the CNN model are randomly selected within 0-1, a z-score normalization was applied to pixel values of the orthophotos to avoid abnormal gradients. This step is essential as it improves the progress of the activation and the gradient descent optimization (LeCun et al., 2012).

$$X' = \frac{(X/\max) - \mu}{\sigma}, \tag{1}$$

where max is the maximum pixel value in the image, μ and σ are the mean and standard deviation of X/max, respectively, and X' is normalized data.

3.2. The Proposed Approach

3.2.1. Overview. An orthophoto is composed of $m \times n \times d$ digital values, where m, n, and d are the image width, length, and depth, respectively. The goal of a classification model is

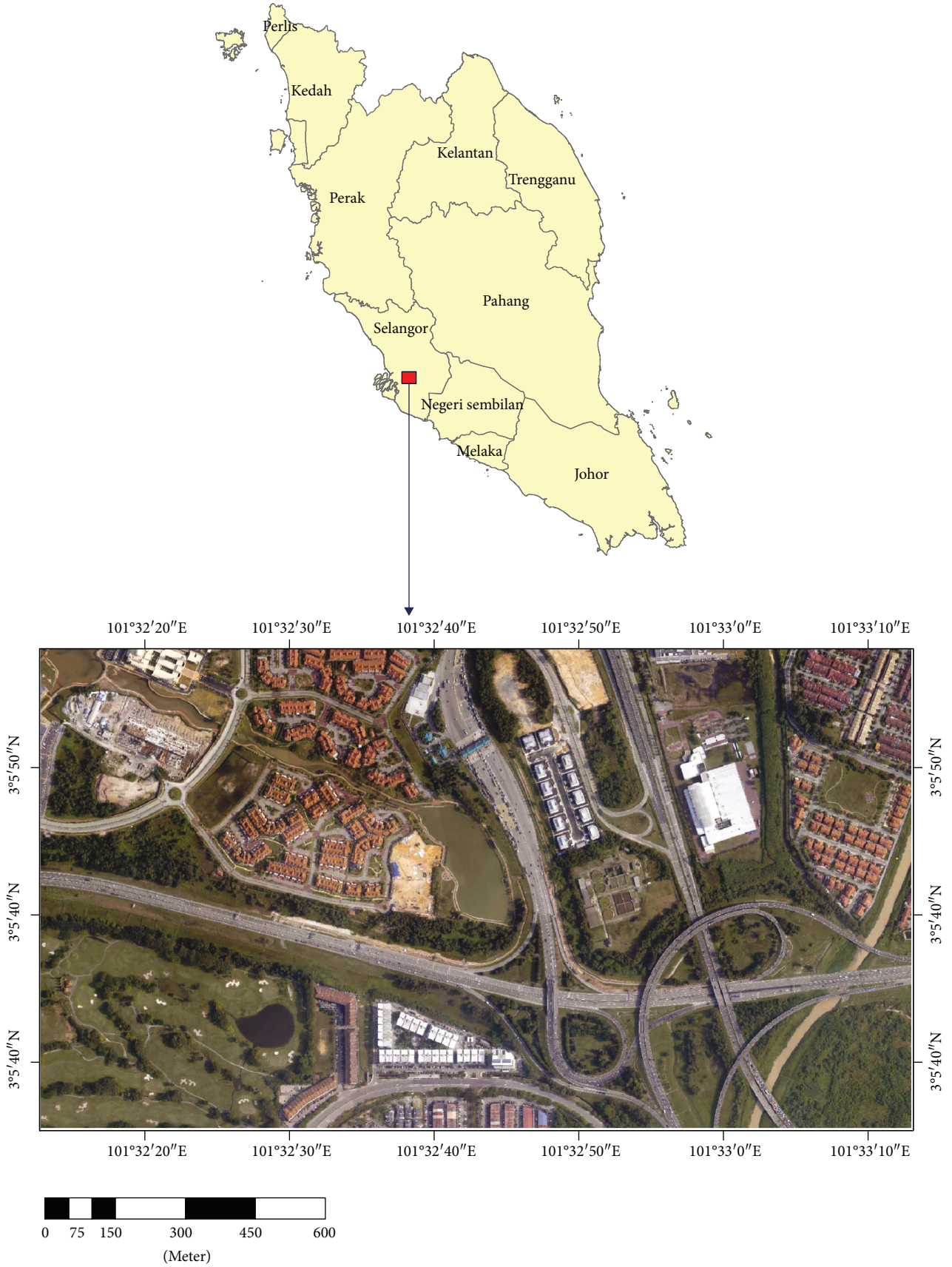


Figure 1: The study area location map.

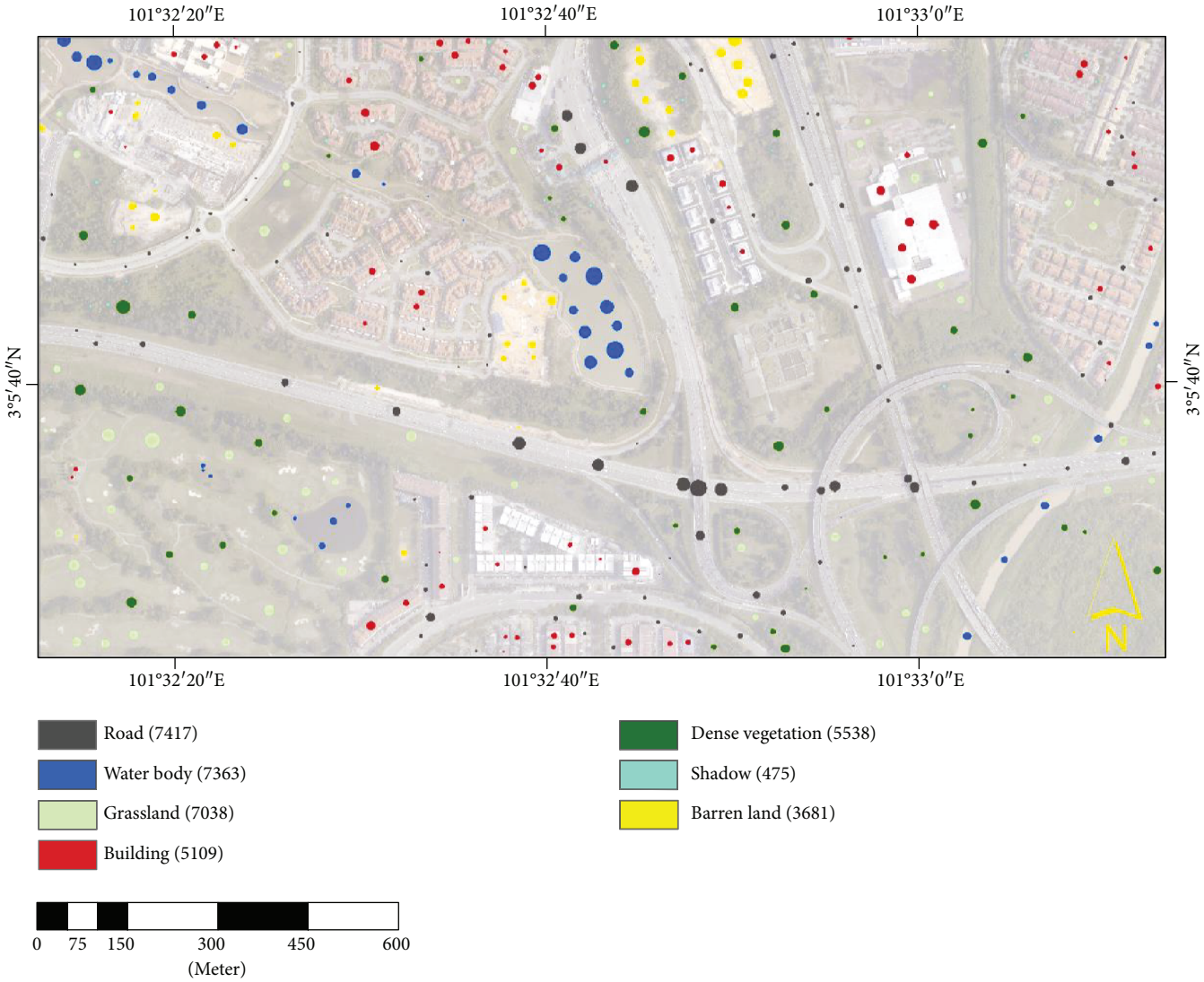


Figure 2: The ground truth samples over the study area, which were manually selected for seven land cover classes, for example, road, water body, grassland, building, dense vegetation, shadow, and barren land. The number in the brackets indicates the number of pixels in each class.

to assign a label to each pixel in the image given a set of training examples with their ground truth labels. In general, the common classification methods utilize the spectral information (image pixels across different bands) to achieve that goal. In addition, some of other techniques such as object-based image analysis (OBIA) segment the input image into several homogeneous contiguous groups before classification. This method uses additional features like spatial, shape, and texture to boost the classification performance of the classifier. However, both the methods, pixel-based and OBIA have several challenges like speckle noise in the first method and segmentation optimization in OBIA. Furthermore, both methods require careful feature engineering and band selection to obtain high accuracy of classification. More recently, classification methods using image patches and deep learning algorithms have been proposed to overcome the above challenges. Among the common methods is CNN. As a result, this study has proposed a classification method that is based on CNN and spectral-spatial feature learning for classifying very high-resolution aerial orthophotos. The following sections describe the proposed model and its components including the basics of CNN, the network architecture, and the training methodology.

The pseudocode of the proposed classification model is presented in Algorithm 1. We developed the CNN model in the current study by running several experiments with different configurations. Then, we designed the ultimate model with best hyperparameters and architecture based on some statistical accuracy metrics such as overall accuracy, Kappa index, and per-class accuracies.

3.2.2. Basics of CNN. Convolutional neural networks (CNNs) or ConvNets are a type of artificial neural networks that simulate the human vision cortical system by using local receptive field and shared weights. It was introduced by LeCun and his colleagues [18]. Figure 3 shows a typical CNN with convolutional max pooling operations. CNN is suitable for analyzing images, videos, or data in the form of

Algorithm 1: CNN for orthophoto classification
Input: RGB image (I) captured by the aerial remote sensing system, training/testing samples (D)
Output: Land cover classification map with seven classes (O)
I, D, O
Preprocessing (Section 3.1.2):
calibrate I using the available 34 GCPs
normalize pixel values using Eq. 1
Classification (CNN) (Section 3.2.2 and Section 3.2.3):
for Patch_x_axis:
  initialize sum = 0
  for Patch_y_axis:
    calculate dot product(Patch, Filter)
  result_convolution (x, y) = Dot product
for Patch_x_axis:
  for Patch_y_axis:
    calculate Max (Patch)
result_maxpool (x, y) = Dot product
update F = max(0, x)
result_cnn_model = trained model
Prediction:
apply the trained model to the whole image and get O
Mapping:
get the results of prediction
reshape the predicted values to the original image shape
convert the array to image and **write** it on the hard disk

Algorithm 1: The pseudocode of the proposed CNN developed for land cover mapping using aerial images.

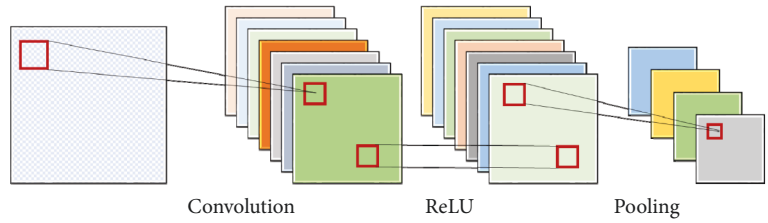


Figure 3: Illustration of typical layers of a CNN.

n-dimensional arrays that have a spatial component. This unique property makes them suitable for remote sensing image classification as well. A typical architecture of CNN consists of a series of layers such as convolution, pooling, fully connected (i.e., dense), and logistic regression/Softmax. However, additional layers like dropout and batch normalization also can be added to avoid overfitting and improve the generalization of these models. The last layer depends on the type of the problem, where for binary classification problems, a logistic regression (sigmoid) layer is often used. Instead, for multiclass classification problems, a Softmax layer is used. Each layer has its operation and is aimed in these models. For example, the convolutional layers are aimed at constructing feature maps via convolutional filters that can learn high-level features that allow taking advantage of the image properties. The output of these layers then passes through a nonlinearity such as a ReLU (rectified linear unit). Local groups of values in array data are often highly correlated, and local statistics of images are invariant to location [19]. In addition, pooling layers (or subsampling) are used to merge semantically similar features into one. The most common method of subsampling computes the maximum of a local patch of units in feature maps. Other pooling operations are averaging max pooling and stochastic pooling. In general, several convolutional and subsampling layers are stacked, followed by dense layers and a Softmax or a logistic regression layer to predict the label of each pixel in the image.

3.2.3. Network Architecture. The architecture of the CNN model was built with a single convolutional layer followed by a max pooling operation, batch normalization, and two dense layer classifiers (Figure 4). This architecture yielded 3527 total parameters where 96 parameters are not trainable. The convolutional kernels were kept as 3×3, and the pooling size in the max pooling layer was kept at 2×2. Dropout was performed in the convolutional layer and the first dense layer with a drop probability of 0.5 to avoid overfitting. The

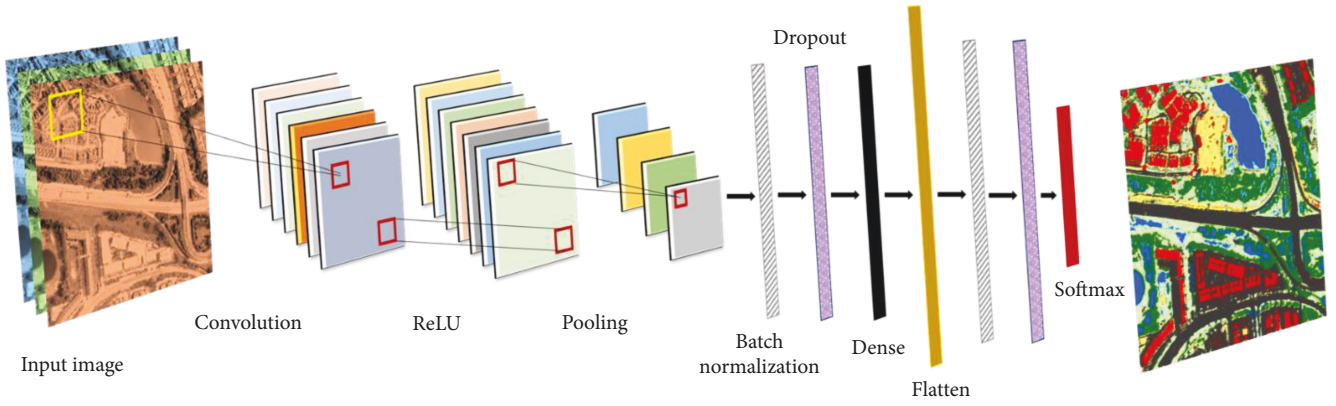


Figure 4: The architecture of the proposed CNN for aerial orthophoto classification.

Table 1: The summary of the CNN model layers.

Layer (type)	Output shape	Number of parameters
Input	(None, 3, 7, 7)	0
2D convolution	(None, 1, 5, 32)	2048
Max pooling	(None, 1, 2, 16)	0
Batch normalization	(None, 1, 2, 16)	64
Dropout	(None, 1, 2, 16)	0
Flatten	(None, 32)	0
Dense	(None, 32)	1056
Batch normalization	(None, 32)	128
Dropout	(None, 32)	0
Dense (Softmax)	(None, 7)	231

minibatch of stochastic gradient descent (SGD) was set to 32 images. Under the framework of Keras with Tensorflow backend, the whole process was run on a CPU Core i7 2.6 GHz and memory ram (RAM) of 16 GB. In the experiments, 60% of the total samples were randomly chosen for training, and the rest were chosen for testing, and overall accuracy (OA), average accuracy (AA), Kappa coefficient (κ), and per-class accuracy (PA) are used to evaluate the performance of the CNN classification method (Congalton and Green, 2008). The summary of the model's layers is shown in Table 1.

3.2.4. Training the Model. The CNN model was trained with backpropagation algorithm and stochastic gradient descent (SGD). It uses the minibatch's backpropagation error to approximate the error of all the training samples, which accelerates the cycle of the weight update with smaller back propagation error to speed up the convergence of the whole model. The optimization was run to reduce the loss function (J) (i.e., categorical cross entropy) of CNN expressed as the following:

$$J\left(X', W, b, \theta\right) = -\frac{1}{N}\left[\sum_{i=1}^{N}\sum_{j=1}^{k} 1\left\{y^i = t\right\} \cdot y_t^i\right], \quad (2)$$

where X' is normalized features, W and b are parameters of CNN, θ is the parameters of Softmax layer, N is the number of samples, k is the number of land cover classes, $\boldsymbol{y}^i = (y_1^i, y_2^i, \ldots, y_k^i)$ is the prediction vector geo by the Softmax classifier (3), and y_t^i represents the possibility of the ith sample label being t and is computed by (3).

$$y_t^i = \frac{\exp\left(\theta_t^T c\right)}{\sum_{j=1}^{k}\exp\left(\theta_t^T c\right)}. \quad (3)$$

During back propagation, (4) are adapted to update W and b in every layer, where λ is the momentum which help accelerate SGD by adding a fraction of the update value of the past time step to the current update value, α is the learning rate, ∇W and ∇b are the gradients of $J(\cdot)$ with respect to W and b, respectively, and t just stands for the number of epoch during SGD:

$$\begin{aligned} W_{t+1} &= W_t - \lambda V_t - \alpha \nabla W, \\ b_{t+1} &= b_t - \lambda U_t - \alpha \nabla b. \end{aligned} \quad (4)$$

3.2.5. Evaluation. This study uses several statistical accuracy measures to evaluate different models and compare them under various experimental configurations. These metrics are overall accuracy (OA), average accuracy (AA), per-class accuracy (PA), and Kappa index (κ). They are calculated using the following equations [20]:

$$\begin{aligned} \mathrm{OA} &= \frac{\sum D_{ii}}{N}, \\ \mathrm{AA} &= \frac{\sum_1^m PA_m}{m}, \\ \mathrm{PA} &= \frac{D_{ij}}{R_i}, \\ k &= \frac{N\sum_{i,j=1}^{m} D_{ij} - \sum_{i,j=1}^{m} R_i \cdot C_j}{N^2 - \sum_{i,j=1}^{m} R_i \cdot C_j}, \end{aligned} \quad (5)$$

where $\sum D_{ii}$ is the total number of correctly classified pixels, N is total number of pixels in the error matrix, m is the

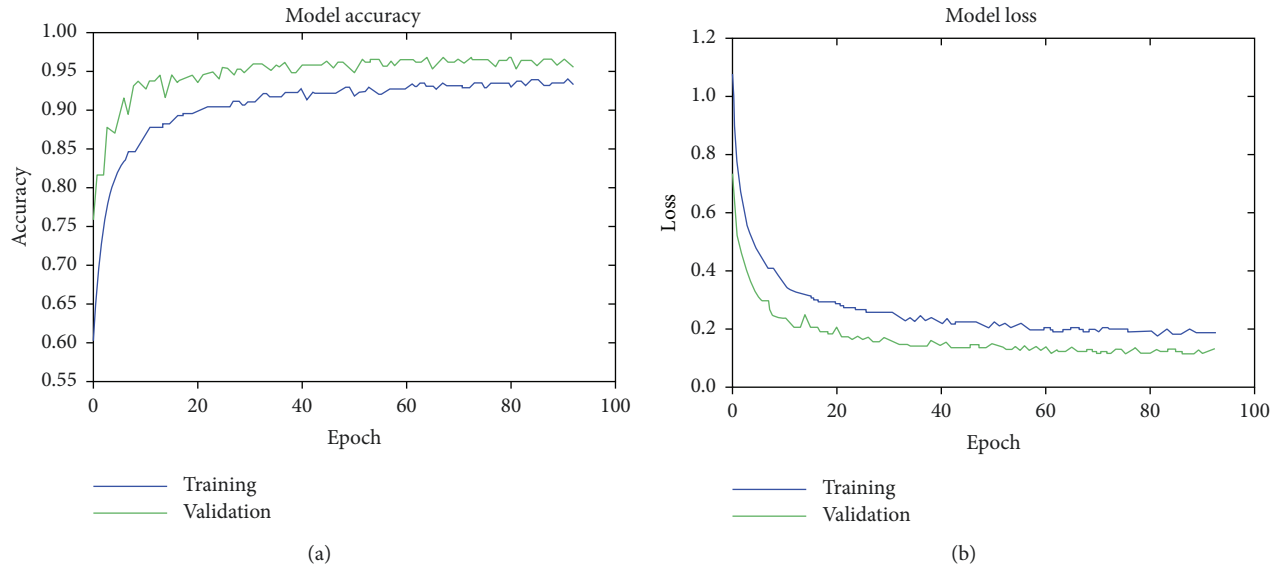


Figure 5: Performance of the CNN model with optimum parameters set, (a) model accuracy and (b) model loss for 93 epochs (early stopping).

Table 2: PA of the CNN model.

Class	PA
Road	0.971
Waterbody	0.944
Grassland	0.972
Building	0.995
Dense vegetation	0.999
Shadow	0.894
Barren land	0.980

number of classes, D_{ij} is the number of correctly classified pixels in row i (in the diagonal cell), R_i is the total number of pixels in row i, and C_j is the total number of pixels in column j.

4. Experimental Results

4.1. Performance of the Proposed Model

4.1.1. CNN with Dropout and Batch Normalization. Figure 5 shows the accuracy performance of the CNN model with dropout and batch normalization for 93 epochs on both training and validation datasets. The increment in model accuracy and reduction in model loss over time indicates that the model has learned useful features to classify the image pixels into the different class labels. The fluctuations in the accuracy from one epoch to another are because of using dropout that yielded a slightly different model at each epoch. The OA, AA, κ of this model on validation dataset was 0.973, 0.965, 0.967, respectively. In addition, Table 2 shows the per-class accuracy (PA) achieved by the model. The results suggest that the CNN model could classify almost all the classes with relatively high accuracy. The minimum accuracy was 0.894 for the shadow class. While examining the confusion matrix (Table 3), the results indicate that several (~11) samples of this class were misclassified as dense vegetation affecting its PA. The confusion matrix also shows that there were several samples of water body class misclassified as grassland.

4.1.2. CNN Model with Other Configurations. The CNN model was also trained without dropout and batch normalization to see their impacts on the accuracy of the classification map. Table 4 summarizes the results of comparing CNN models with different configurations (i.e., CNN + dropout + batch normalization, CNN + dropout, CNN + batch normalization, and CNN). The results suggest that the use of dropout and batch normalization could improve the accuracy (OA, AA, and κ) of the classification by almost 4%. The use of batch normalization slightly performed better (OA = 0.964, AA = 0.956, κ = 0.961) than just using dropout (OA = 0.958, AA = 0.956, κ = 0.954). Nevertheless, the use of either dropout or batch normalization could improve the accuracy of the classification compared to not using any of these techniques with the CNN model. The CNN model without these techniques achieved the following accuracies: OA = 0.932, AA = 0.922, κ = 0.922 indicating the importance of such regularization methods for aerial orthophoto classification. The classified maps produced by these methods are shown in Figure 6. Furthermore, the performance plot (Figure 7) of the CNN model without dropout and batch normalization shows that this model overfits the training data and performs worse when applied to new data. Overall, the experimental results on both training and validation data sets infer that the proposed CNN architecture is a robust and efficient model, while the use of dropout and batch normalization

Table 3: The confusion matrix calculated for the CNN model.

	Road	Waterbody	Grassland	Building	Dense vegetation	Shadow	Barren land
Road	**1474**	0	0	23	0	0	21
Water body	0	**1463**	85	0	0	1	0
Grassland	0	10	**1323**	0	27	0	0
Building	4	0	0	**991**	0	0	0
Dense vegetation	0	0	0	0	**1070**	1	0
Shadow	0	0	0	0	11	**93**	0
Barren land	6	0	0	8	0	0	**716**

Table 4: Performance of CNN model with different configurations.

Model	OA	AA	κ
CNN + dropout + batch normalization	**0.973**	**0.965**	**0.967**
CNN + dropout	0.958	0.956	0.954
CNN + batch normalization	0.964	0.956	0.961
CNN	0.932	0.922	0.922

techniques as regularization methods is essential to obtain high accuracy of classification for the entire area rather than just predicting the labels of the training samples.

4.2. Sensitivity Analysis. The performance of CNN while classifying orthophotos is highly dependent on its architecture and hyperparameters. Thus, the sensitivity analysis could serve as an essential step in finding a good set of parameters and architecture configurations in addition to an understanding of the model behavior. Figure 8 shows the impact of different parameters (e.g., number of convolutional filters, activation function, drop probability, optimizer, batch size, and patch size) on the validation accuracy of CNN.

For convolutional filters, the sensitivity analysis shows that larger number of filters can lead to an increase in performance. However, use of larger number of filters can increase training time and overfit the training data if the model is not regularized properly. Thus, this parameter was set to 32 as an optimal setting and not exploring a larger number of filters. With this configuration, the model could achieve the following accuracies: OA = 0.956, AA = 0.945, and $\kappa = 0.947$. In addition, this analysis shows that the activation function "ReLU" outperformed the other two functions ("Sigmoid" and "ELU"). By using this activation, the CNN model could achieve an OA of 0.956 higher than the second best activation "Sigmoid" by ~4.4%. ReLU also facilitates faster training and reduced likelihood of vanishing gradient. The experiments on drop probability showed that different parametric values can improve the performance of CNN depending on the accuracy metric. For example, results showed that the use of drop probability as 0.2 could optimize the model for OA and κ, where the model achieved an OA and κ of 0.975, 0.970, respectively. However, drop probability of 0.3 could perform better than the value of 0.2 for this parameter regarding AA. Furthermore, performances of CNN with different optimizers have been investigated, and the results indicated that "Adam" could be effective in training compared to other optimizers. The highest OA (0.975) and κ (0.970) were achieved by the CNN model that was trained with "Adam." However, when the optimizer "Nadam" was used to train CNN, the model could achieve the highest AA (0.974). The worst performance of CNN (OA = 0.945, AA = 0.949, and $\kappa = 0.934$) was found to be when the model was trained with SGD. Moreover, the efficiency of CNN was compared with different batch sizes such as 4, 8, 16, 32, and 64. The batch size of 32 was found the best considering OA (0.975) and κ (0.970), while the batch size of 64 achieved the highest AA (0.975).

Another important parameter in the proposed CNN is the patch size, which is the local neighborhood area that forms with the size $(n \times n)$. The advantage of using patch-based learning for orthophoto classification is sourced from the benefits of spectral and spatial information of the data that can improve the accuracy compared to just using the individual pixels (only spectral information). To understand this parameter and find its suboptimal value, several experiments were conducted with different patch sizes ($n = [3, 5, 7, 9, 11, 13]$). The statistical analysis in terms of model accuracy indicates that using larger n yields higher accuracy (Figure 8). However, when analyzing the classification map visually, the use of larger n reduces the spatial quality of the features in the classification map (Figure 9). As a result, we considered $n = 7$ as an effective value for this parameter as it achieved relatively high accuracy measured by OA, AA, and κ as well as high spatial quality features.

4.3. Training Time Analysis. The computing performance of the CNN model was dependent on the use of dropout and batch normalization layers in the network architecture in addition to other hyperparameters such as a number of convolutional filters and image patch size. Table 5 shows the training time of the CNN model with different configurations. When early stopping was applied, the training of CNN with dropout and batch normalization took about 124 seconds on a CPU. Removing the batch normalization from the architecture yielded a training time of 150 seconds, whereas CNN with dropout took 75 seconds to be trained. The CNN model without the use of dropout and batch normalization took the shortest time (58.4 seconds)

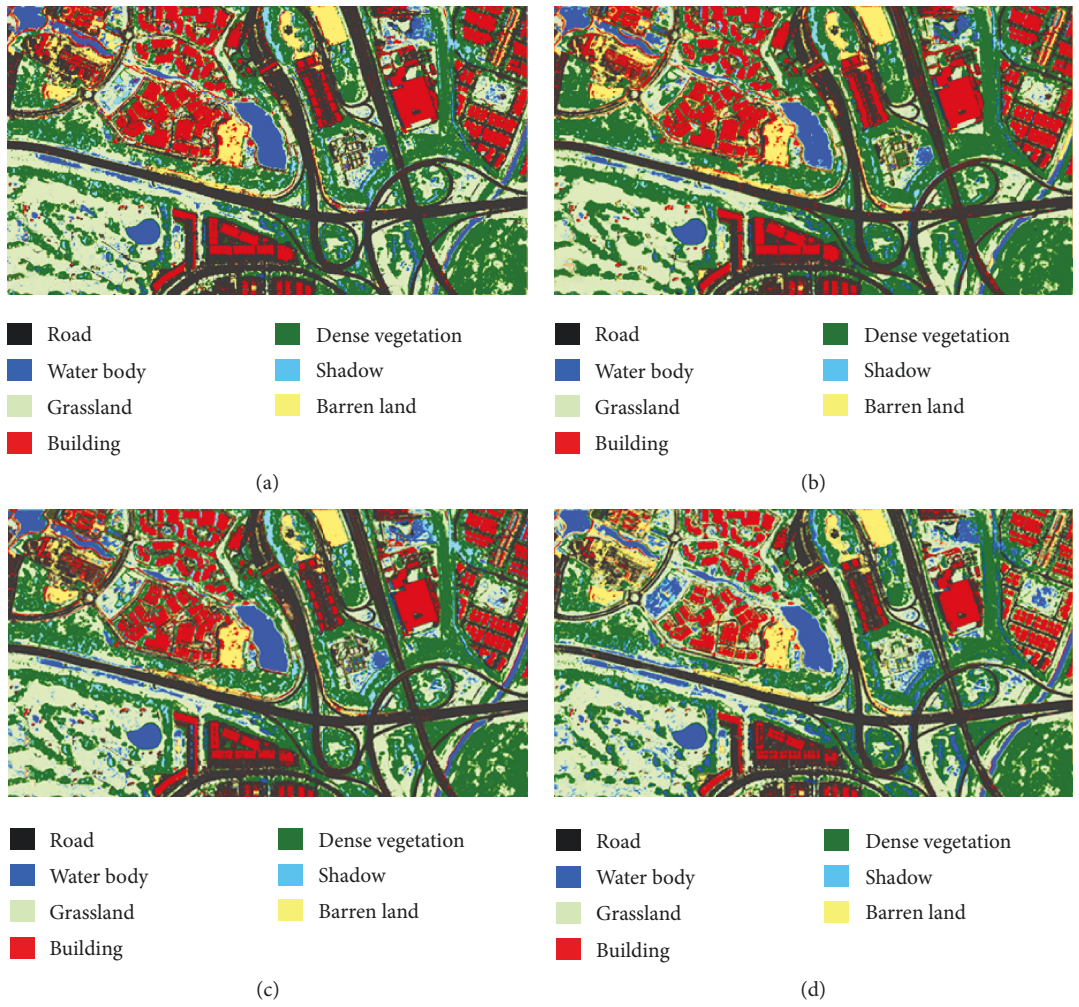


Figure 6: Classification maps produced by CNN models, (a) CNN + dropout + batch normalization, (b) CNN + dropout, (c) CNN + batch normalization, and (d) CNN.

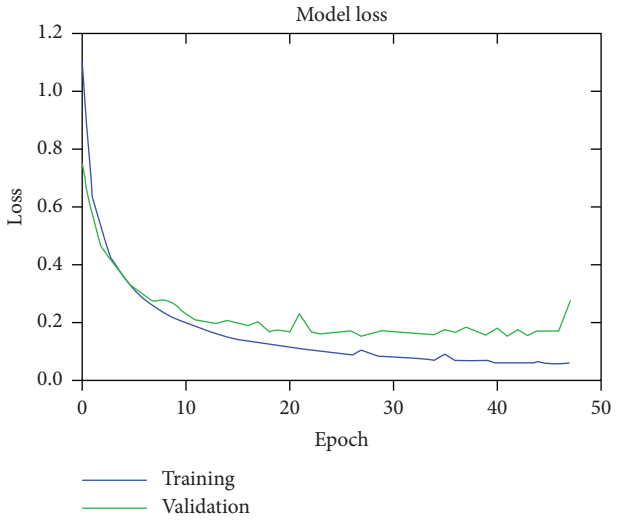
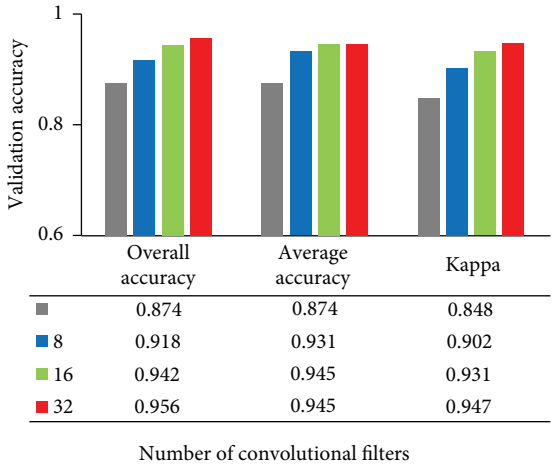


Figure 7: The loss of the CNN model without dropout and batch normalization.

to be trained. On the other hand, when the model was trained with 200 epochs without early stopping, the model (CNN + dropout + batch normalization) took about 230 seconds longer than that with early stopping by 106 seconds. In addition, the other models (CNN + dropout, CNN + batch normalization, and CNN) were also required a longer time to train as it was expected due to more number of epochs run. Overall, the computing performance of the proposed model is efficient for the investigated data. However, for larger datasets, the training of such models will require longer time, and as a result, graphical processing units will be essential.

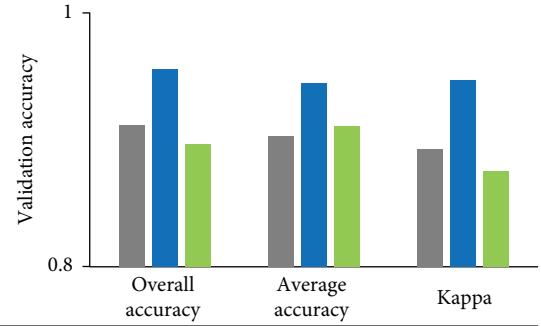
5. Conclusion

In this paper, a classification model based on CNN and spectral-spatial feature learning has been proposed for aerial photographs. With the utilization of advanced regularization techniques such as dropout and batch normalization, the proposed model could balance generalization ability and training efficiency. Use of such methods to improve the CNN model along with other techniques like preprocessing



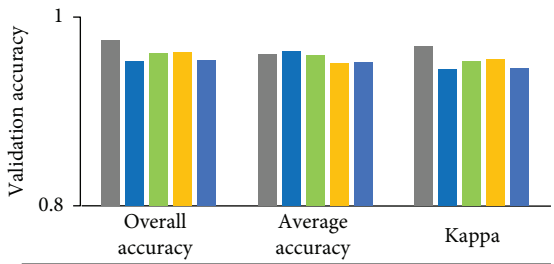
	Overall accuracy	Average accuracy	Kappa
	0.874	0.874	0.848
8	0.918	0.931	0.902
16	0.942	0.945	0.931
32	0.956	0.945	0.947

Number of convolutional filters



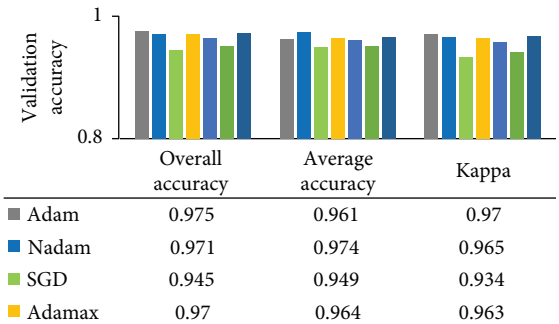
	Overall accuracy	Average accuracy	Kappa
Sigmoid	0.912	0.903	0.893
ReLU	0.956	0.945	0.947
ELU	0.897	0.911	0.876

Activation



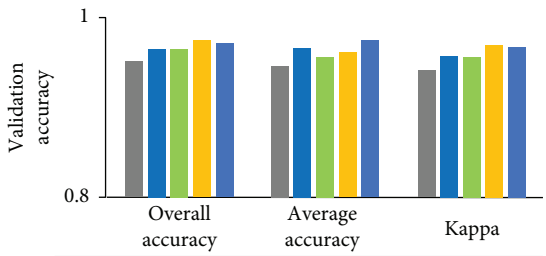
	Overall accuracy	Average accuracy	Kappa
0.2	0.975	0.961	0.97
0.3	0.954	0.964	0.945
0.4	0.962	0.959	0.954
0.5	0.963	0.951	0.956
0.6	0.955	0.953	0.946

Drop probability



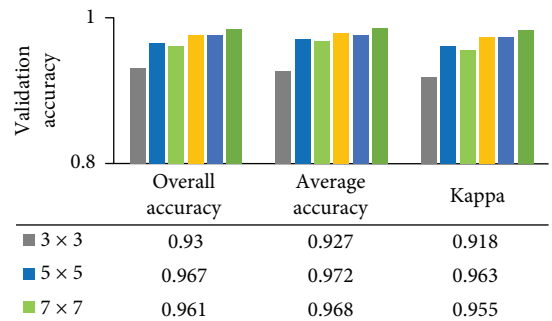
	Overall accuracy	Average accuracy	Kappa
Adam	0.975	0.961	0.97
Nadam	0.971	0.974	0.965
SGD	0.945	0.949	0.934
Adamax	0.97	0.964	0.963
Adadelta	0.964	0.96	0.956
Adagrad	0.951	0.951	0.941
RMSProp	0.972	0.965	0.966

Optimizer



	Overall accuracy	Average accuracy	Kappa
4	0.951	0.946	0.941
8	0.964	0.966	0.957
16	0.964	0.955	0.956
32	0.975	0.961	0.97
64	0.972	0.975	0.967

Batch size



	Overall accuracy	Average accuracy	Kappa
3 × 3	0.93	0.927	0.918
5 × 5	0.967	0.972	0.963
7 × 7	0.961	0.968	0.955
9 × 9	0.977	0.98	0.974
11 × 11	0.977	0.977	0.975
13 × 13	0.985	0.986	0.984

Patch size

Figure 8: The influence of hyperparameters, the number of convolutional filters, activation function, drop probability, optimizer, batch size, and patch size.

(geometric calibration and feature normalization) and sensitivity analysis could make these models robust for classifying the given dataset. The CNN model acts as a feature extractor, and a classifier could be trained end-to-end given training samples. The network architecture can effectively handle the inter- and intraclass complexity inside the scene. The best model achieved OA = 0.973, AA = 0.965, and $\kappa = 0.967$ outperforming the traditional CNN model by ~4% in all the accuracy indicators. The short training time (124 seconds) confirmed the robustness of the proposed model for small and medium scale remote sensing datasets. The future work should focus on scaling this architecture for large remote sensing datasets and other data sources such as satellite images and laser scanning point clouds.

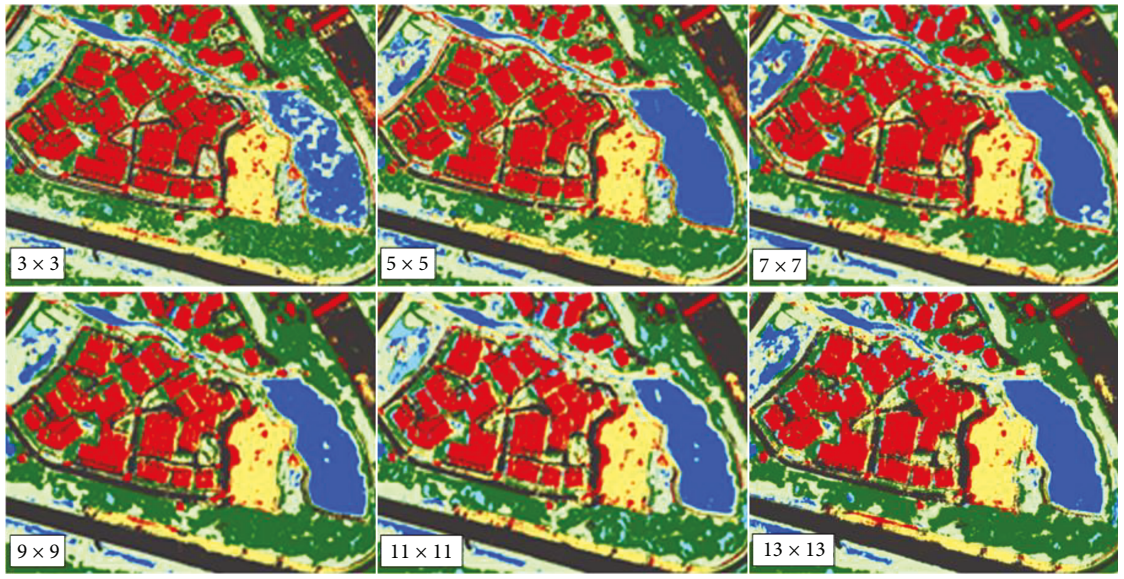


Figure 9: Effects of patch size on the quality of classified maps.

Table 5: The training time in seconds of CNN with different configurations for 200 epochs.

Model	Time (seconds)—with early stopping	Time (seconds)—full training
CNN + dropout + batch normalization	124	230
CNN + dropout	150	168
CNN + batch normalization	75	219
CNN	58.4	158

Data Availability

These data were used from a research project lead by Professor Biswajeet Pradhan. Very high resolution aerial photos were used in this research. The data can be made available upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Ö. Akar, "The Rotation Forest algorithm and object-based classification method for land use mapping through UAV images," *Geocarto International*, vol. 33, no. 5, pp. 538–553, 2017.

[2] Q. Wu, R. Zhong, W. Zhao, H. Fu, and K. Song, "A comparison of pixel-based decision tree and object-based Support Vector Machine methods for land-cover classification based on aerial images and airborne lidar data," *International Journal of Remote Sensing*, vol. 38, no. 23, pp. 7176–7195, 2017.

[3] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "DeepSat: a learning framework for satellite imagery," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, p. 37, New York, NY, USA, November 2015.

[4] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[5] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 844–853, 2017.

[6] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: a dimension reduction and deep learning approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4544–4554, 2016.

[7] Y. T. Hsieh, C. T. Chen, and J. C. Chen, "Applying object-based image analysis and knowledge-based classification to ADS-40 digital aerial photographs to facilitate complex forest land cover classification," *Journal of Applied Remote Sensing*, vol. 11, no. 1, article 015001, 2017.

[8] M. F. A. Vogels, S. M. De Jong, G. Sterk, and E. A. Addink, "Agricultural cropland mapping using black-and-white aerial photography, object-based image analysis, and random forests," *International Journal of Applied Earth Observation and Geoinformation*, vol. 54, pp. 114–123, 2017.

[9] X. Meng, N. Shang, X. Zhang et al., "Photogrammetric UAV mapping of terrain under dense coastal vegetation: an object-oriented classification ensemble algorithm for classification and terrain correction," *Remote Sensing*, vol. 9, no. 11, p. 1187, 2017.

[10] A. Juel, G. B. Groom, J. C. Svenning, and R. Ejrnaes, "Spatial application of random forest models for fine-scale coastal vegetation classification using object based analysis of aerial orthophoto and DEM data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 42, pp. 106–114, 2015.

[11] L. Albert, F. Rottensteiner, and C. Heipke, "A higher order conditional random field model for simultaneous classification of land cover and land use," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 130, pp. 63–80, 2017.

[12] G. Cheng, C. Ma, P. Zhou, X. Yao, and J. Han, "Scene classification of high resolution remote sensing images using convolutional neural networks," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 767–770, Beijing, China, July 2016.

[13] G. J. Scott, M. R. England, W. A. Starms, R. A. Marcum, and C. H. Davis, "Training deep convolutional neural networks for land-cover classification of high-resolution imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 4, pp. 549–553, 2017.

[14] J. Sherrah, "Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery," 2016, http://arxiv.org/abs/1606.02585.

[15] W. Yao, P. Poleswki, and P. Krzystek, "Classification of urban aerial data based on pixel labelling with deep convolutional neural networks and logistic regression," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B7, pp. 405–410, 2016.

[16] X. Sun, X. Lin, S. Shen, and Z. Hu, "High-resolution remote sensing data classification over urban areas using random forest ensemble and fully connected conditional random field," *ISPRS International Journal of Geo-Information*, vol. 6, no. 8, p. 245, 2017.

[17] J. R. Bergado, C. Persello, and C. Gevaert, "A deep learning approach to the classification of sub-decimetre resolution aerial images," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 1516–1519, Beijing, China, July 2016.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[20] A. Bogoliubova and P. Tymków, "Accuracy assessment of automatic image processing for land cover classification of St. Petersburg protected area," *Acta Scientiarum Polonorum. Geodesia et Descriptio Terrarum*, vol. 13, no. 1-2, 2014.