Hindawi
Journal of Sensors
Volume 2018, Article ID 7212307, 12 pages
https://doi.org/10.1155/2018/7212307

Research Article

Deep Learning Approach for Building Detection Using LiDAR–Orthophoto Fusion

Faten Hamed Nahhas,[1] Helmi Z. M. Shafri,[1,2] Maher Ibrahim Sameen,[3] Biswajeet Pradhan,[3] and Shattri Mansor[1,2]

[1]*Department of Civil Engineering, Faculty of Engineering, Universiti Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia*
[2]*Geospatial Information Science Research Center (GISRC), Faculty of Engineering, Universiti Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia*
[3]*Centre for Advanced Modelling and Geospatial Information Systems (CAMGIS), Faculty of Engineering and IT, University of Technology Sydney, Sydney, NSW, Australia*

Correspondence should be addressed to Helmi Z. M. Shafri; helmi@upm.edu.my

Received 21 November 2017; Accepted 26 June 2018; Published 5 August 2018

Academic Editor: Antonio Lazaro

This paper reports on a building detection approach based on deep learning (DL) using the fusion of Light Detection and Ranging (LiDAR) data and orthophotos. The proposed method utilized object-based analysis to create objects, a feature-level fusion, an autoencoder-based dimensionality reduction to transform low-level features into compressed features, and a convolutional neural network (CNN) to transform compressed features into high-level features, which were used to classify objects into buildings and background. The proposed architecture was optimized for the grid search method, and its sensitivity to hyperparameters was analyzed and discussed. The proposed model was evaluated on two datasets selected from an urban area with different building types. Results show that the dimensionality reduction by the autoencoder approach from 21 features to 10 features can improve detection accuracy from 86.06% to 86.19% in the working area and from 77.92% to 78.26% in the testing area. The sensitivity analysis also shows that the selection of the hyperparameter values of the model significantly affects detection accuracy. The best hyperparameters of the model are 128 filters in the CNN model, the Adamax optimizer, 10 units in the fully connected layer of the CNN model, a batch size of 8, and a dropout of 0.2. These hyperparameters are critical to improving the generalization capacity of the model. Furthermore, comparison experiments with the support vector machine (SVM) show that the proposed model with or without dimensionality reduction outperforms the SVM models in the working area. However, the SVM model achieves better accuracy in the testing area than the proposed model without dimensionality reduction. This study generally shows that the use of an autoencoder in DL models can improve the accuracy of building recognition in fused LiDAR–orthophoto data.

1. Introduction

Buildings are a fundamental element in forming a city and are essential for urban mapping [1]. The extraction of accurate building objects from remote sensing data has become an interesting topic and has received increasing attention in recent years. Building information is important in several geospatial applications, such as urban planning, risk and damage assessment of natural hazards, 3D city modeling, and environmental sciences. Building objects can be delineated from many data sources, such as satellite images, aerial photos, radar images, and laser scanning data. In particular, Light Detection and Ranging (LiDAR) offers an accurate and efficient approach for obtaining elevation data, which can be used to extract ground objects, such as buildings [2]. The advantages of using LiDAR over traditional photogrammetry include the capability to collect high-density point clouds at a relatively short time, high vertical accuracy,

and low cost. However, the accurate extraction of buildings in urban areas with precise boundaries is a difficult task due to the presence of nearby objects, such as trees, which frequently have the same elevations as buildings. Therefore, the fusion of LiDAR point clouds and aerial images can be an important step toward improving the quality of building detection.

Numerous methods have been proposed for building detection in the past decades by using LiDAR data and by fusing other remote sensing data with LiDAR data to improve accuracy and quality. Li et al. [3] proposed a series of novel algorithms for detecting building boundaries from the fusion of LiDAR and high-resolution images. Their results indicate that the fusion of LiDAR and high-resolution images is a promising approach for the accurate detection of building boundaries (correctness = 98% and completeness = 95%). Li et al. [4] proposed an improved building extraction method based on the fusion of optical imagery and LiDAR data. The aforementioned method comprises four steps: filtering, building detection, wall point removal, and roof patch detection. Their results suggest that the proposed method can automatically extract building objects with complex shapes. Saeidi et al. [5] also applied a data-driven method based on Dempster–Shafer theory to fuse LiDAR and SPOT (Satellite Pour l'Observation de la Terre) data for building extraction. These researchers examined the potential of slope and height information extracted from the LiDAR-based digital elevation model (DEM) and digital surface model (DSM), as well as from the normalized difference vegetation index (NDVI) created from SPOT images. Their results show that NDVI/normalized DSM (nDSM) fusion performs better than NDVI/slope for building extraction.

Uzar and Yastikli [6] developed an automatic building detection method based on LiDAR data and aerial photographs. This method includes segmentation and classification with object-based image analysis. The accuracy assessment shows an overall accuracy of approximately 93%, a completeness of 96.73%, and a correctness of 95.02% for building extraction. Uzar [7] developed an automatic approach for building extraction based on multisensor data (LiDAR and aerial photographs) and rule-based classification. He applied fuzzy classification to improve building extraction results. His method achieved a completeness of 81.71% and a correctness of 87.64% based on a comparison between the extracted buildings and reference data. Furthermore, Awrangjeb et al. [8] proposed an automatic building detection technique using LIDAR data and multispectral imagery. They utilized the normalized difference vegetation index to separate the buildings from trees and extract the residential buildings in the area. Awrangjeb et al. [9] also developed a building detection technique for complex scenes. In their method, a rule-based procedure was established to utilize the normalized digital surface model extracted from LiDAR data for the task in hand effectively.

Recently, Wang et al. [10] presented an automatic method for building boundary extraction from LiDAR data. This method includes height-based segmentation, shape recognition by shape indices, and boundary reconstruction using Hough transformation and a sequential linking technique. Their findings show that the proposed method can achieve accurate extraction of building boundaries at rates of 97%, 85%, and 92% for three LiDAR datasets with different scene complexities. Prerna and Singh [11] assessed a building detection method based on the segmentation of LiDAR and high-resolution photographs. These researchers determined that an object-based-oriented classification yielded the best accuracy ($R^2 = 0.86$) compared with using only LiDAR. Zhao et al. [12] presented a building extraction method using LiDAR data and connected operators. Their results demonstrate that the proposed method performs effectively. The efficient and average offset values of simple and complex building boundaries are 0.2 m to 0.4 m and 0.3 m to 0.6 m, respectively. Tomljenovic et al. [13] applied object-based analysis for building extraction from LiDAR data. Their obtained results exhibit high accuracies for the initial study area and on the International Society for Photogrammetry and Remote Sensing benchmark without any modification.

Tomljenovic et al. [2] reviewed building extraction methods based on LiDAR data. Their analysis shows that the main limitations of current building detection methods are their application to wide-area datasets and the lack of transferability studies and measures. Other challenges in building detection from LiDAR include point cloud sparsity, high spectral variability, differences of urban objects, surrounding complexity, and data misalignment [14]. Gilani et al. [14] proposed a methodology that extracts and regularizes buildings using features from LiDAR data and orthoimagery to overcome some of the aforementioned limitations. Their results demonstrate the robustness of their approach. However, this method is affected by the registration error between LiDAR data and orthoimagery, which requires a further validation on different datasets. The lack of transferability of current methods is mainly due to the use of rule-based classification.

Therefore, the current paper reports on a building detection method based on the fusion of LiDAR data and orthophotos using a deep learning (DL) approach. At present, DL has gone beyond multilevel perceptrons and comprises a collection of techniques and computational methods for building compassable differentiable architecture. In particular, this study develops a framework based on an autoencoder to reduce feature dimensionality and a convolutional neural network (CNN) to distinguish building objects from non-building objects after segmentation is performed on LiDAR and orthoimage data.

2. Methodology

This section describes the proposed model and explains its components that have been designed to detect buildings from LiDAR and orthophotos based on a DL approach. It describes the overall workflow, data preprocessing and preparation, feature extraction through multiresolution and spectral difference segmentations, feature fusion and abstraction using autoencoders and CNN, and building detection that applies fully connected layers with sigmoid activation to the final layer.

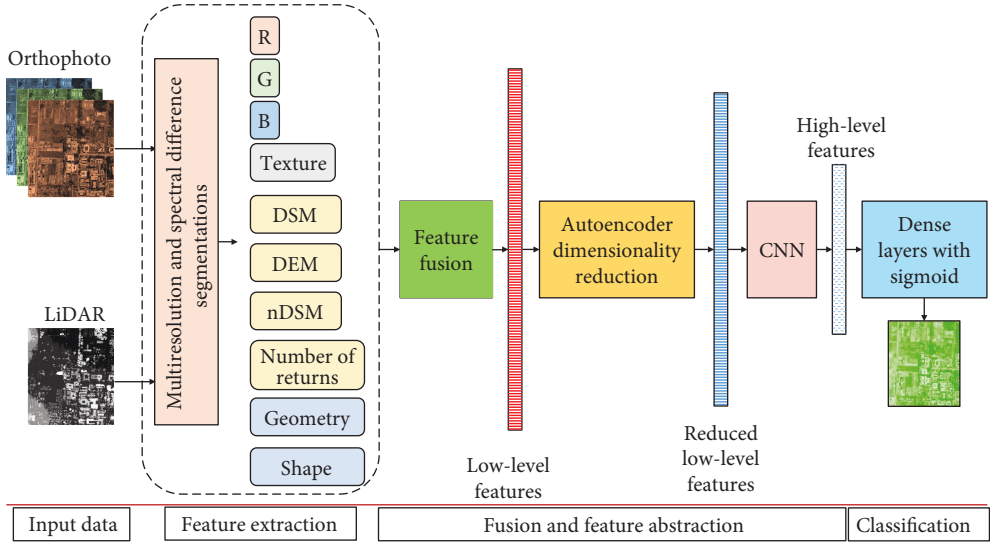

Figure 1: Architecture of the proposed building detection method using DL and LiDAR–orthophoto fusion.

2.1. Overall Architecture. This study proposes a DL model for detecting buildings from fused LiDAR and orthophoto data. The overall workflow of this model is presented in Figure 1. The proposed pipeline encompasses four main components: preprocessing and preparation of input data, feature extraction, fusion and feature abstraction, and classification. The first component, that is, data preparation, includes the geometric correction and registration of LiDAR point clouds with orthophotos. The point clouds were filtered to create DSM, DEM, and nDSM samples. DSM was created by interpolating point clouds using the inverse weighted distance (IDW) method. DEM was created by filtering nonground points using the multiscale curvature algorithm of ArcGIS [15]. nDSM was created by subtracting DEM from DSM. The LiDAR-derived DSM, DEM, nDSM; number of returns; and orthophoto bands (i.e., red, green, and blue) were then composted at 0.3 m spatial resolution and prepared for segmentation.

The second component, that is, feature extraction, was implemented to extract the spectral and texture features from the orthophotos and DSM, DEM, number of returns, and geometry and shape features from LiDAR data. The third component includes feature fusion and abstraction using an autoencoder DL model to reduce features and a CNN model to transform low-level features into high-level features. The last component adopted fully connected layers and a sigmoid layer to classify image objects into background and buildings. The details of these processing steps are explained in the following sections.

2.2. Feature Extraction. A total of 21 features, including spectral, shape, textural, and LiDAR-based features, were initially extracted to detect building objects in the LiDAR and orthophoto data. Spectral features were used to evaluate the mean pixel values in the orthophoto bands. The shape features refer to the geometric information of meaningful objects, which is calculated from the pixels that form these objects. An accurate segmentation of the map is necessary to ensure the successful use of these features. Texture features were also derived from the Haralick texture features based on the gray-level cooccurrence matrix (GLCM) or the gray-level difference vector. Alternatively, the LiDAR-based features were used to describe the topography and height of objects.

The low-level features (Table 1) were calculated based on the image objects created via multiresolution and spectral difference segmentations. The features extracted from the LiDAR data and orthophoto were fused at the feature level. The features were then reduced by applying an autoencoder-based dimensionality reduction approach. The reduced low-level features were then fed into the CNN model to extract the high-level features for classification. The following sections describe the aforementioned processes.

2.3. Fusion and Feature Abstraction. Building detection and description are important steps in reconstructing building objects. The former refers to the process of identifying building objects among other objects [20], whereas the latter refers to the process of delineating the geometric boundary of building objects to describe their geometry and extract information as attributes linked to the objects in a geographic information system (GIS). On the one hand, orthophotos have a significant capacity in spatial resolution and exhibit strong reflectance around building boundaries. However, the spectral similarity of different ground objects generates difficulties in extracting buildings from orthophotos. On the other hand, extracting building edges with height discontinuity is difficult in LiDAR due to the relatively small footprint size of the laser beam and disadvantageous backscattering from illuminated targets [20]. Thus, the fusion of orthophotos and LiDAR can improve the accuracy of building detection and description processes.

Data fusion is defined as the process of using or combining data from multiple sources to form a new dataset and accomplish a particular objective [21]. The three fusion levels that can combine data from different sources are classified as pixel, feature, and decision fusions [22]. The present study

Table 1: Extracted features from the LiDAR and orthophoto data [16–19].

Data source	Feature group	Feature	Description
Orthophoto	Spectral	Mean red	Average value of the pixels that cover the segment in the red band
		Mean green	Average value of the pixels that cover the segment in the green band
		Mean blue	Average value of the pixels that cover the segment in the blue band
	Texture	GLCM angular	$\sum_{i,j=0}^{N-1} P_{i,j}^2$
		GLCM contrast	$\sum_{i,j=0}^{N-1} P_{i,j}\lvert i-j\rvert^2$
		GLCM correlation	$\sum_{i,j=0}^{N-1} [((i-\mu_i)(j-\mu_j))(\sigma_i^2+\sigma_j^2)^{-1/2}]$
		GLCM dissimilarity	$\sum_{i,j=0}^{N-1} P_{i,j}\lvert i-j\rvert$
		GLCM entropy	$\sum_{i,j=0}^{N-1} P_{i,j}(-\ln P_{i,j})$
		GLCM homogeneity	$\sum_{i,j=0}^{N-1} \frac{P_{i,j}}{(1+(i-j)^2)}$
		GLCM mean	$f_{\mu_i} = \mu_i \sum_{i,j=0}^{N-1} i(P_{i,j}), f_{\mu_i} = \mu_j \sum_{i,j=0}^{N-1} j(P_{i,j})$
		GLCM variance	$\sum_{i,j=0}^{N-1} P_{i,j}(i-\mu)^2$
LiDAR	Shape	Area	Total area of segment without holes
		Compactness	Ratio of the area of a polygon to the area of a circle with the same perimeter
		Density	Distribution in space of the pixels of an image object
		Length/width	Length-width ratio of the envelope rectangle
		Rectangular fit	Goodness of a building that fits into a rectangle
		Roundness	Area of the segment to the square of the maximum diameter of the referred segment
		Shape index	Border length of the segment divided by four times the square root of its area
	LiDAR	DEM	Digital elevation model
		DSM	Digital surface model
		nDSM	Object height by subtracting DEM from DSM

In the equations above, i is the row number of the cooccurrence matrix, j is the column number of the cooccurrence matrix, and $P_{i,j}$ is the normalized value in cell i, j ($P_{i,j} = V_{i,j}/\sum_{i,j=0}^{N-1} V_{i,j}$), where $V_{i,j}$ is the value in cell i, j of the cooccurrence matrix and N is the number of rows or columns of the cooccurrence matrix.

adopts the feature level because building detection and description with object-based analysis are easier and more efficient. Orthophoto features (e.g., spectral and textural features) and LiDAR features (e.g., DSM, DEM, nDSM, and spatial features) are combined to form low-level features for building detection (Table 1).

Many features that are related to spectral, textural, topographical, and shape groups can be extracted from orthophotos and LiDAR data. The use of many features can cause overfitting, particularly when the training samples are relatively small. The other disadvantages of using a large number of features are noise, redundant information, and increasing computing time. The current study introduces an autoencoder-based approach that reduces feature space dimensionality and improves low-level features by transforming them into fewer features (i.e., reduced low-level features) to address the aforementioned issue. The transformed features are expected to be more informative than the raw features and to improve the performance of the overall methodology workflow of building detection. A CNN model is also developed to select the relevant features for detecting buildings and to transform the reduced low-level features into high-level features by applying a set of convolution and pooling operations. The process of reducing (or abstracting) low-level features by using the autoencoder and CNN models is described in the following sections.

2.3.1. Autoencoders. Autoencoders (Figure 2) are neural networks that attempt to reconstruct their inputs without using labels (unsupervised); they have two logical parts, that is, the

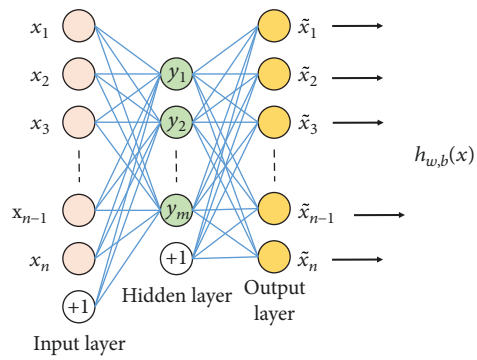

Figure 2: Simple structure of an autoencoder (adapted from [24]).

encoder and the decoder [23]. The former comprises the network layers that create a hidden representation of the input data, whereas the latter comprises the network layers that take the hidden representation from the encoder and create an output that is similar to the input data of the encoder. Thus, the last layer in autoencoder networks has the same size as the input of the first input layer. This process allows the network to learn features regarding the input data and regularization parameters. Hidden representation can be smaller than the input data; hence, the major benefit of autoencoders is dimensionality reduction.

Autoencoders adopt the backpropagation algorithm for training [23]. In an autoencoder, the output $h_{w,b}(x) = (\tilde{x_1}, \tilde{x_2}, \ldots, \tilde{x_n})^T$ is equal to the input $x = (x_1, x_2, \ldots, x_n)^T$.

$$h_{w,b}(x) = g(f(x)) \approx x,$$
$$J(W, b; x, y) = \frac{1}{2}\left\|h_{w,b}(x) - y\right\|^2, \quad (1)$$

where x is an input that belongs to the n-dimensional space, y is a new representation that belongs to the m-dimensional space, and J is the reconstruction error.

A standard autoencoder comprises three layers. The first layer to the second layer amounts to an encoder f, and the second layer to the third layer amounts to a decoder g. Then, the algorithm minimizes J by adjusting the parameters in the encoder and the decoder to obtain the reconstructed input. The number of hidden layer nodes m is restricted to less than the number of original input nodes n to utilize the autoencoder as a dimensionality reduction algorithm.

The proposed autoencoder architecture includes 21 input features, 3 hidden layers with 128, 50, and 30 nodes, and a central layer with dimensions 5, 10, and 15 that are evaluated iteratively. Several hidden layers and their associated number of nodes were selected using a grid search method [25] and evaluated based on the similarity between the input and reconstructed data measured via the mean squared error. The network was trained through the Adamax optimization method [26] with its default parameters in Keras (TensorFlow backend) [27] and a batch size of 32. A sparsity constraint (L1 activity regularizer) was also added to the encoded representations to avoid overfitting and reduce model complexity.

2.3.2. CNNs. CNN [28] is a method that simulates a multilayer structure of the human brain. It can extract the features of input data from a low to a high layer incrementally to improve classification or prediction processes. It abstracts the relationships among data and improves optimization performance with a reduction in training parameters. The structure of CNN consists of three layers that can be described as the convolution, subsample (pooling), and fully connected layers (Figure 3).

The proposed CNN architecture is encompassed by two stacked feature stages. Each stage contains a convolution layer followed by a pooling layer. A 2D convolution with 128 filters and maximum pooling were used. The high-level features were produced by flattening the 2D features estimated via the convolution and pooling operations. The network was also trained using the Adamax optimization method with a batch size of 8. Once the high-level features were obtained, a fully dense layer with 10 nodes and a dropout rate of 0.2 were used to classify features into building or background classes. The trained CNN model was then adopted to predict the class of test data, and the outputs were utilized to create the final building maps in GIS. The CNN network was optimized using the grid search method, which is explained in the next section.

2.3.3. Optimization Procedure. The optimization of hyperparameters is a crucial step in developing an efficient object detection model through DL methods, which are easy to use. Optimization can improve the overall performance, prediction accuracy, and generalization capacity of models, particularly when they are used to predict unseen data. The current study utilizes the grid search method to determine the optimal hyperparameters among specific search spaces of the CNN model. The grid search typically identifies a better set of hyperparameters than a manual search within the same amount of time. The optimized parameters, their search spaces, and their determined optimal values are shown in Table 2. Five hyperparameters, namely, the optimizer, number of filters, number of hidden units of the dense layer, dropout rate, and batch size, were optimized. The search spaces of the hyperparameters (excluding the optimizer) were manually selected after several random experiments.

3. Results and Discussion

This section describes the experimental datasets, the results of building detection with the accuracy assessment, and the sensitivity analysis of the proposed model. The proposed model was developed in Python using Google's TensorFlow library. It was then implemented in a personal computer with an Intel® Core i7 at 2.00 GHz and a memory (RAM) of 16 GB.

3.1. Experimental Datasets. The proposed building detection model was evaluated on two datasets (i.e., working and testing) selected from the Universiti Putra Malaysia campus located in the state of Selangor, Malaysia (Figure 4). The selected areas are geographically located between latitudes 7°11′00″E and 7°14′00″E and longitudes 3°00′00″N and

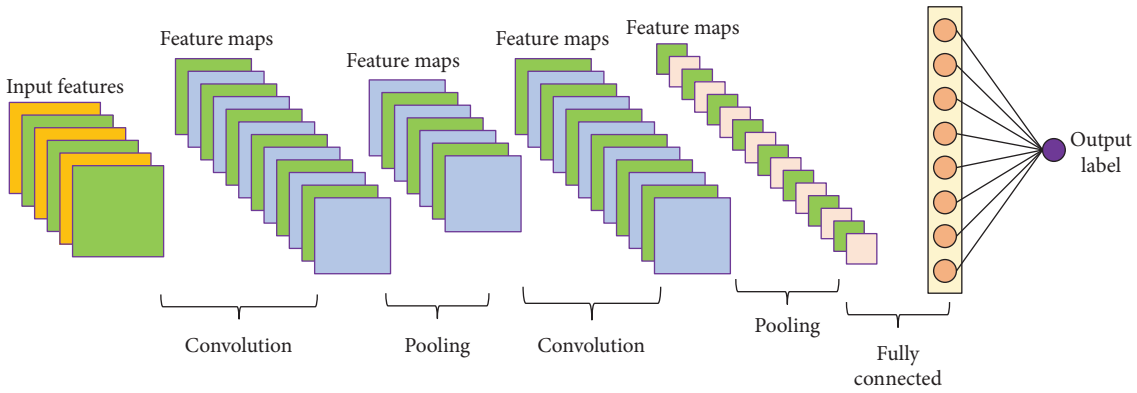

Figure 3: Typical CNN architecture.

Table 2: Optimization of CNN hyperparameters.

Parameter	Search space	Optimal value
Optimizer	(rmsprop, adam, nadam, Adamax, Adadelta, sgd)	Adamax
Number of filters	(16, 32, 64, 128)	128
Number of hidden nodes	(3, 10, 50, 100)	10
Dropout rate	(0, 0.2, 0.3, 0.5)	0.2
Batch size	(4, 8, 16, 32, 64)	8

3°40′00″N of the Kertau RSO Malaya coordinate system. The areas were selected because they include a mixture of urban features, such as asphalt roads, trees, dense vegetation, water bodies, and buildings. The buildings have different roofing materials, shapes, sizes, and heights.

The LiDAR data used in this study was obtained with a laser scanning system (Riegl LM Q5600 and Camera Hasselblad 39 Mp) on March 8, 2015. The systems had a scanning angle of 60° and a camera angle of ±30°. The average point density of the LiDAR data was 4 points/m^2 with an average point space of 0.43 m. Overall, the LiDAR data contained 9.24 million points in the working and testing areas. The minimum and maximum elevations are in the working area at 37.65 m and 79.83 m, respectively. The elevations in the testing area range from 36.86 m to 100.36 m. Three different products were derived from the raw LiDAR point clouds, namely, DEM, DSM, and height feature or nDSM. Furthermore, the laser scanning system also collected RGB images along the point clouds. The spatial resolution of the collected orthophotos is 13 cm.

DSM was derived with IDW interpolation at 0.5 m spatial resolution. Meanwhile, DEM was derived using an ArcGIS filtering algorithm called multiscale curvature classification (MCC) [15]. The validations of this filtering method exhibit improvement in removing understory vegetation, which addresses topological differences across scales [15]. The other advantages of this approach include a built-in function in ArcGIS software, which makes its implementation easy and enables its integration into an automatic processing pipeline. The MCC algorithm filters LiDAR point clouds by classifying LiDAR returns as ground and nonground points. This algorithm combines curvature filtering with a scale component and variable curvature tolerance [15]. MCC then interpolates a surface at different resolutions through the thin-plate spline method, and points are classified based on a progressive curvature threshold parameter (0.78 in this study). Other LiDAR data filtering methods are presented in the works of [29, 30].

3.2. Results of Building Detection. The image objects created via multiresolution segmentation were classified into buildings and backgrounds using the proposed DL model. Classification was applied with the complete set of features (21) and the best number of features obtained by the autoencoders (10 features). The building detection results are shown in Figure 5. Figure 5(a) shows the buildings detected by the model in the working area without reducing the dimensionality of the input features. The total number of buildings detected is 2808, which is higher by 8% than the real number in the reference dataset. The reason for this misclassification is mainly due to noise, which leads to small objects being incorrectly detected as buildings. With regard to the geometry of the detected buildings, Figure 5(c) shows that the detected buildings were affected by nearby objects, such as roads and trees. These objects create a problem in accurately describing the buildings. For example, a single building is composed of several objects, which cannot offer accurate building counting in the study area. Additional nearby objects attached to the detected buildings also create an issue in describing building objects, such as estimating their roofing geometry, floor area, and even their height. By contrast, the results of the model with reduced features show better building detection with less misclassification and better boundary delineation (Figures 5(b) and 5(d)). The number of buildings calculated using this method is 281, which is 0.86% lower than the reference number of buildings. Reducing the number of features using the autoencoder model may contribute to the removal of features that create overfitting in the model and offer better building detection results. Figure 5(d) shows an example of how reducing the features used for building detection can also contribute to improving the boundary delineation of objects. This property is extremely useful in counting the buildings in the study area with better accuracy. Furthermore, building detection with

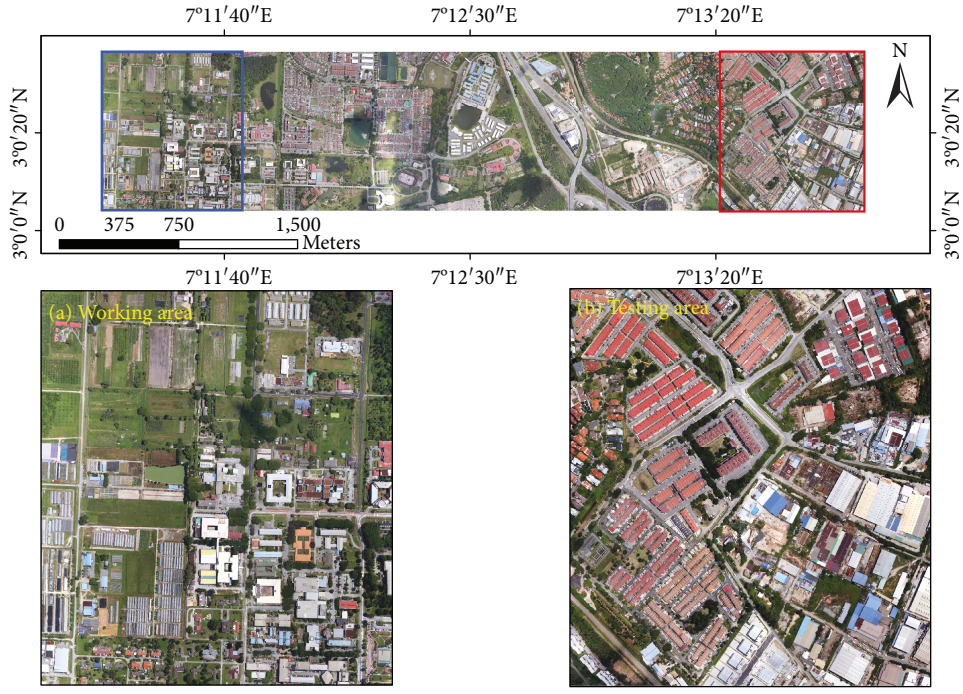

Figure 4: Location of the study area and experimental datasets: (a) working area and (b) testing area.

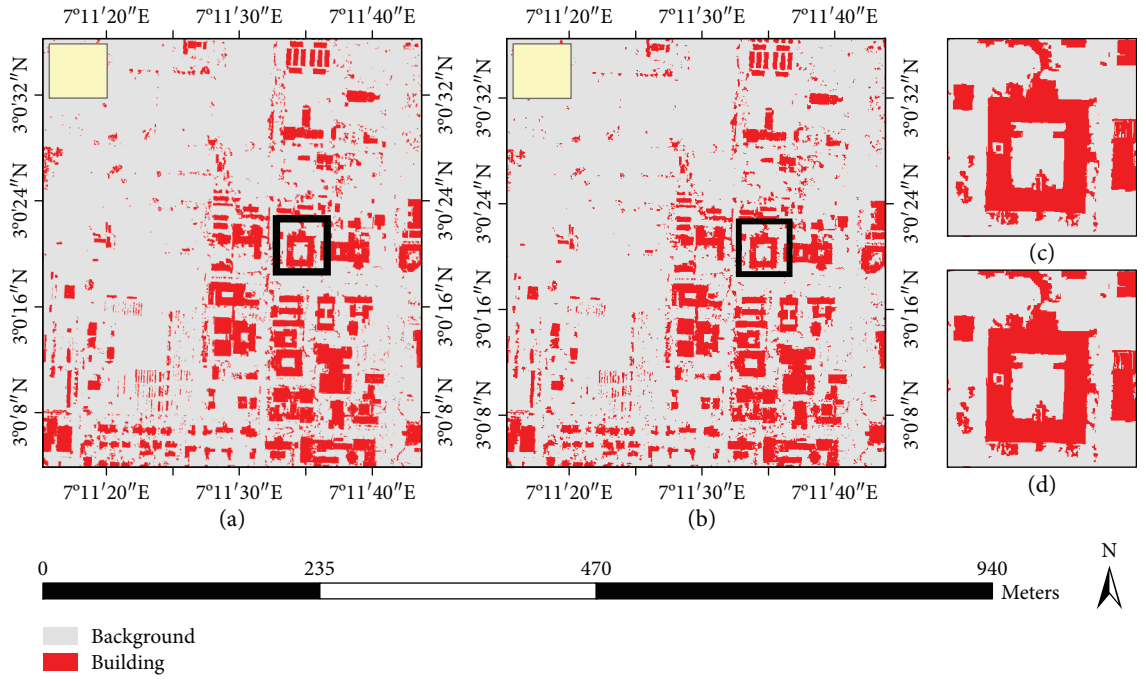

Figure 5: Results of building detection using the proposed method in the working area: (a) without feature dimensionality reduction, (b) with dimensionality reduction (10 features), (c) example of a detected building through a complete set of input features, and (d) example of building detection after feature reduction by the autoencoder dimensionality reduction approach.

accurate boundary can calculate several spatial and geometric attributes of the objects with high precision. The model outputs that apply autoencoders for feature fusion and abstraction allow exporting of building information in the study area that can be useful for decision-making and urban planning, among other applications.

Furthermore, the proposed model was also used to detect buildings in the testing area, and the results are shown in Figure 6. The model was applied with and without feature reduction. Figure 6(a) shows the buildings in the testing area obtained by the model without using autoencoders. The number of buildings in this map is 1029, which is 4.47%

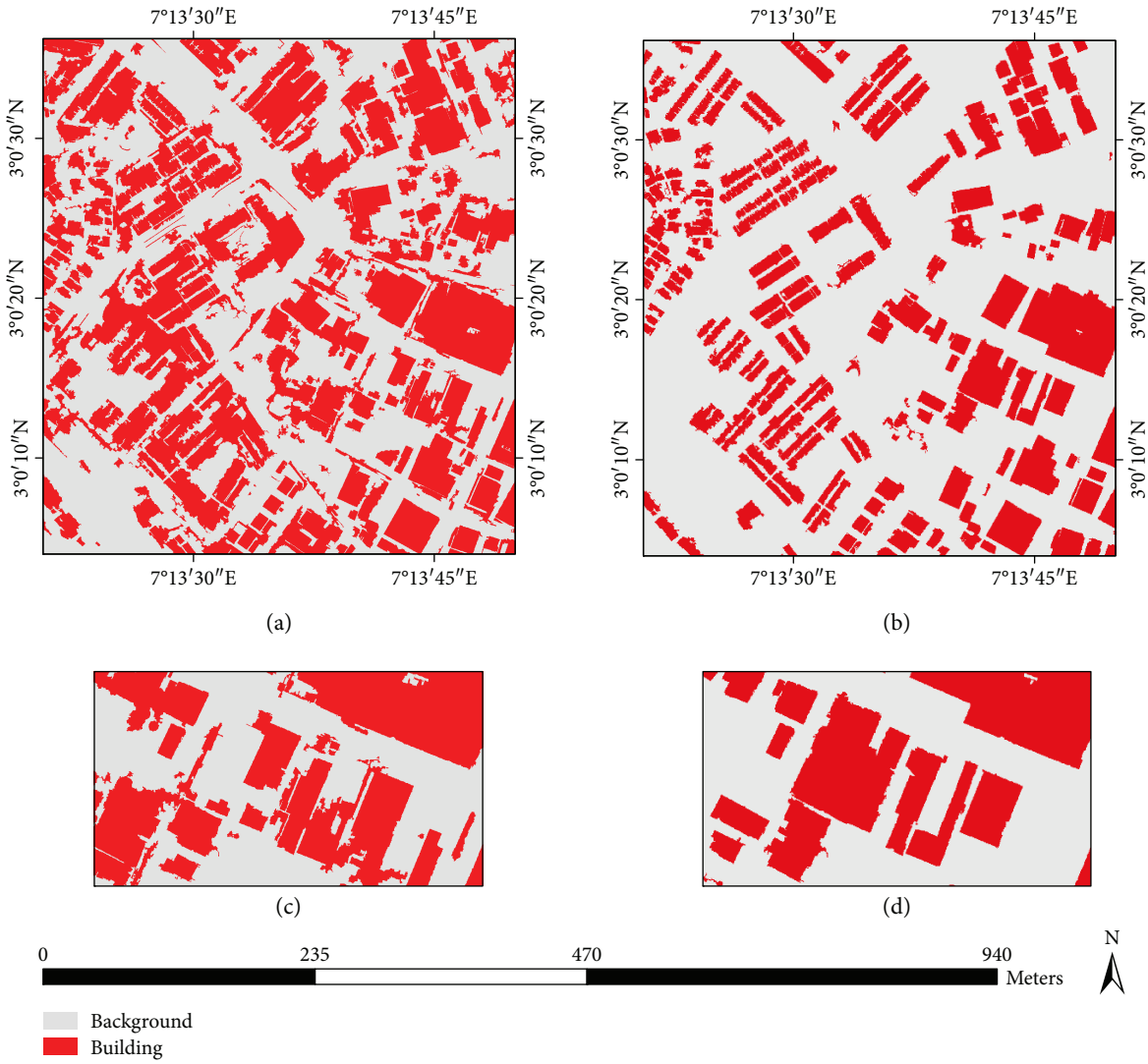

Figure 6: Results of building detection using the proposed method in the testing area: (a) without feature dimensionality reduction, (b) with dimensionality reduction (10 features), (c) example of a detected building through a complete set of input features, and (d) example of building detection after feature reduction by the autoencoder dimensionality reduction approach.

higher than the ground truth number. The geometry of the detected objects also shows noisy boundaries and additional nearby objects, such as trees combined with building objects (Figure 6(c)). The noisy boundaries limit the applications of the produced building map due to insufficient accuracy regarding counting and geometry. By contrast, the model that uses autoencoders presents better results (Figures 6(b) and 6(d)). The number of buildings calculated in the map is 256 (1.11% higher than the ground truth). Similarly, the results indicate that reducing the number of features by using the autoencoder approach can improve building detection accuracy and its boundary delineation.

3.3. Sensitivity Analysis. The proposed model has several hyperparameters with significant effects on the accuracy of building detection from the fusion of LiDAR–orthophoto data. Thus, this section presents a sensitivity analysis of these hyperparameters.

Table 3: Effects of dimensionality reduction on building detection accuracy.

Number of features	Accuracy (%)	
	Working area	Testing area
21	86.06	77.92
15	85.90	76.71
10	86.19	81.86
5	84.77	78.26

3.3.1. Effects of Dimensionality Reduction. Autoencoders can reduce the dimensionality of input features to a lower number of features by specifying the dimension of its central layer. Table 3 shows the different experiments applied to detect buildings in the input data with different dimensions of the middle layer of the autoencoder model. The dimensions explored were 15, 10, and 5. The model with the complete set of features achieved an accuracy of 86.06% and

77.92% in the working and testing areas, respectively. After reducing the number of features to 15, the model detected the buildings in the areas with accuracies of slightly less than those using the complete set of features. The accuracy in the working and testing areas with 15 features were 85.90% and 76.71%, respectively. In addition, the use of 10 features offered the best results in the working and testing areas with overall accuracies of 86.19% and 81.86%, respectively. By contrast, when the number of features was reduced to 5, the overall accuracy of building detection decreased by 1.29% in the working area and slightly improved in the testing area compared with using the complete set of features.

Autoencoders learn a compressed representation of the input; thus, using the transformed features instead of the complete set of features can reduce noise and redundant information in the features. Although the use of autoencoders in DL models can exhibit lower performance in the training data, better generalization power can still be attained. Moreover, reducing the number of features improves the computing performance of the model while keeping accuracy as high as possible. The use of autoencoders can be more efficient than the multiclass recognition problem or the one-class classification problem for building detection. The main reason for this finding is that detecting one feature type frequently requires relatively fewer significant features than using many features, wherein some of the features may be irrelevant to the task. In the case of multiclass recognition problems, features that are irrelevant to a specific class may be significant for others, and vice versa.

3.3.2. Effects of the CNN Model. The CNN model has several hyperparameters, such as the number of filters, the optimizer, the number of hidden units in the fully connected layer, batch size, and dropout rate. The selection of hyperparameter values significantly affects detection accuracy; therefore, the parameters were carefully analyzed and optimized. Figure 7 shows the results of the sensitivity analysis of these parameters evaluated based on the 10-fold cross-validation accuracy achieved for building detection in the testing area. With regard to the number of filters, the results show that the best number of filters is 128, which achieves an accuracy of 81.86%. The lowest accuracy (15.5%) was obtained with 64 filters. The analysis also shows that the best optimizer is Adamax, which realized an accuracy of 81.41% and is significantly better than other methods. By contrast, the number of hidden units in the dense layer has disregarded effects. The highest accuracy (81.86%) was attained by using 10 units or 100 units. The use of 3 units and 50 units obtained slightly lower accuracy (81.61%). The use of a lower number of units in the fully connected layer improves the computing performance of the model; therefore, the optimal value of these parameters is regarded as 10. Furthermore, the sensitivity analysis results show that the best batch size is 8, which achieved an accuracy of 81.86%. The use of a batch size of 4 also attained a slightly similar accuracy (81.32%). However, the use of batch sizes larger than 8 shows a reduction in accuracy of nearly 50%. Finally, the analysis indicates that the dropout rate can have direct effects on the accuracy of building recognition. The best dropout rate is 0.2, which achieved an accuracy of 81.73%. The combination of the best parameter values is considered the best set of parameters and thus is used to produce the final maps (Figures 5 and 6).

3.4. Comparison with Support Vector Machine (SVM). The proposed model was compared with the traditional machine learning method of SVM. Table 4 shows the accuracy assessment of the different methods applied to detect buildings in the working and testing areas. The results of the comparison experiments in the working area show that the best accuracy (86.19%) was obtained using the proposed model with a lower number of features selected by the autoencoder model. The proposed model without dimensionality reduction also obtained higher accuracy than the SVM models. The results show that the SVM model can achieve relatively good accuracy when its hyperparameters are optimized. However, the SVM model with default parameters can attain the lowest accuracy (76.56%). The experiments in the testing area similarly show that the best accuracy (81.86%) can be obtained using the proposed model with dimensionality reduction. However, the SVM model with optimized hyperparameters outperforms the proposed model without dimensionality reduction. The accuracy of the SVM model with optimization is 79.27%, whereas the DL model without using autoencoders for dimensionality reduction achieved 77.92% building detection accuracy. The SVM model with default parameters obtained the lowest accuracy (74.11%).

Figure 8 presents an example of the building detection results from the testing area for the proposed model and SVM method. Figure 8(a) shows the study subset that contains different building types with various geometric and roofing characteristics. The results of the proposed model without dimensionality reduction are presented in Figure 8(b), whereas those with dimensionality reduction are presented in Figure 8(c). The results that used autoencoders are more accurate, with less noise in nearby building boundaries. For example, the results show that the proposed model with a lower number of features can obtain results that are more precise with regard to building geometry. The buildings in Figure 8(b) were combined, and the model could not detect the features between the buildings. By contrast, the results of the proposed model (with 10 features) present better building separation compared with those using the complete set of features. Furthermore, using the transformed features instead of the original set of features can better distinguish between buildings and nearby trees. The SVM models present relatively similar results. However, the optimized SVM exhibits better detection accuracy and less misclassification between buildings and nearby trees. Moreover, the results of the optimized SVM show better building separation as highlighted by the green circles in Figure 8(d). Overall, the accuracy assessment and visual interpretation of the classification results show that the proposed model is more accurate than the SVM model.

4. Conclusion

This study developed a DL approach based on autoencoders and CNN models to detect buildings in a fused LiDAR–

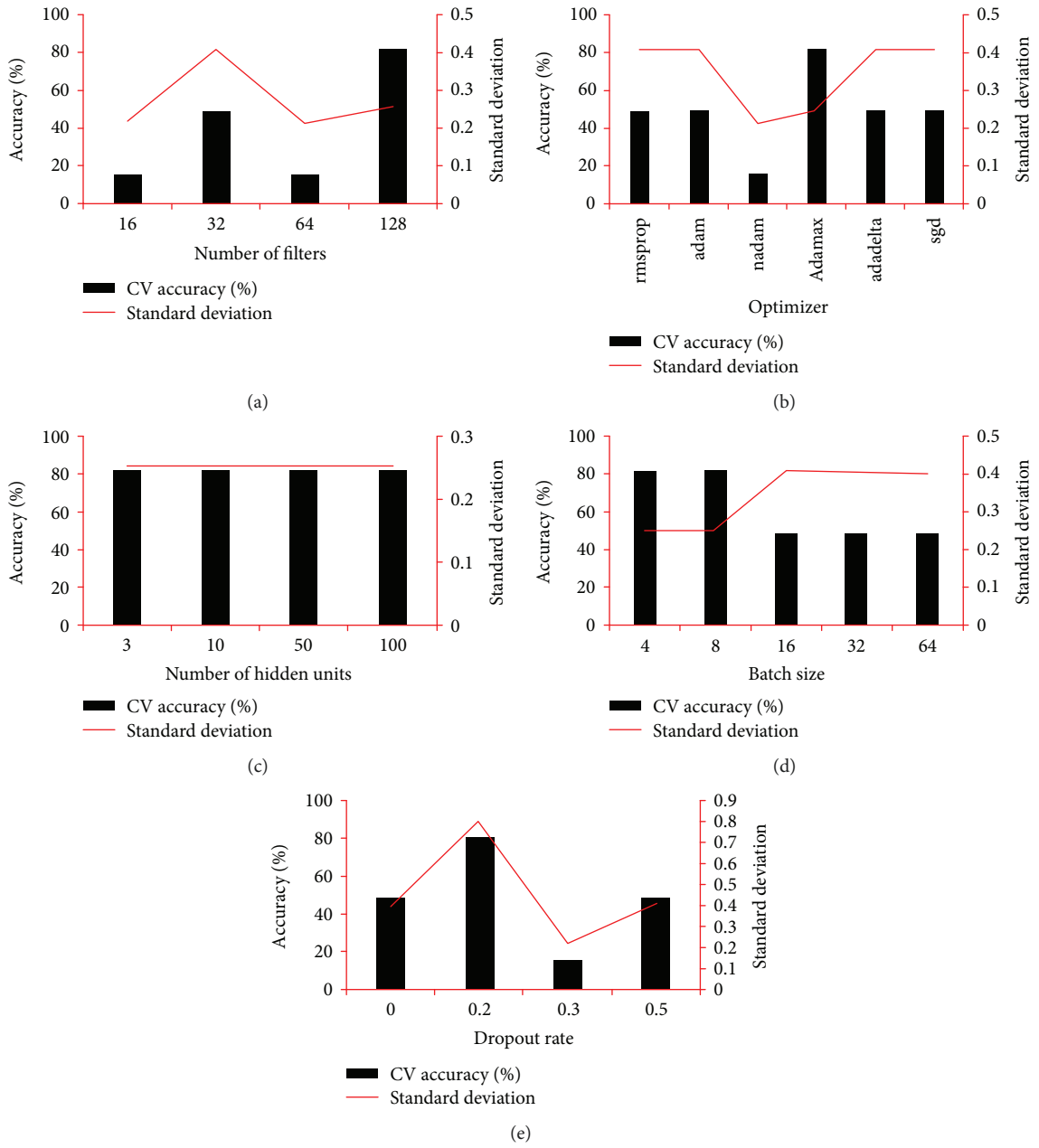

Figure 7: Effects of CNN hyperparameters on building detection accuracy.

Table 4: Accuracy assessment of the testing area.

Dataset	Model	Properties	Accuracy (%)
Working area	Proposed model	Without dimensionality reduction	86.06
	Proposed model	With dimensionality reduction	86.19
	SVM	Without optimization	76.56
	SVM	With optimization	82.34
Testing area	Proposed model	Without dimensionality reduction	77.92
	Proposed model	With dimensionality reduction	81.86
	SVM	Without optimization	74.11
	SVM	With optimization	79.27

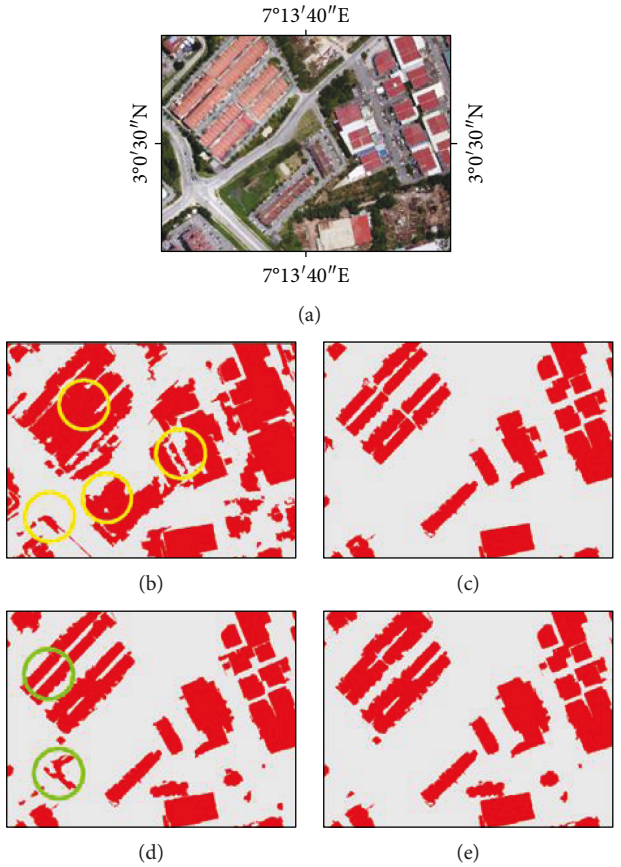

Figure 8: Examples of the results of the proposed model and the SVM method: (a) orthophoto of the area, (b) results of the proposed model without dimensionality reduction, (c) results of the proposed model with dimensionality reduction, (d) SVM results without optimization, and (e) SVM results with optimization.

orthophoto dataset. The proposed architecture includes multiresolution and spectral difference segmentations to create objects by grouping the image pixels according to their shape and spectral properties. A total of 21 features from spectral, textural, LiDAR, and spatial features were identified for building detection. These low-level features were then fused at the feature level and compressed into 10 features using the autoencoder model. The compressed features were transformed into high-level features, which were then used to classify the objects into buildings and nonbuildings. The main advantages of applying such architecture to building detection include automatic feature selection and removal of redundant features for improved building detection in datasets.

The main findings of the study suggest that using autoencoders as a dimensionality reduction step can improve the accuracy of building recognition and improve the computing performance of the model. The proposed model achieved the best accuracy of 86.19% in the working area and 81.86% in the testing area. The comparative study shows that the proposed model outperforms the SVM model in the working and testing areas. Furthermore, the sensitivity analysis indicates that the hyperparameters of the DL model and SVM method should be fine-tuned to obtain better accuracy levels in building detection. Although the proposed method that was determined to be useful for building detection achieves better results than the SVM model, several points still have to be considered in the future. Further research should be performed to improve the proposed model for large-scale building mapping and testing. Future studies should also test whether using satellite images instead of orthophotos can improve accuracy or will only increase the cost of data.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank Universiti Putra Malaysia (UPM) (Grant GP-IPS/2016/9491800) for funding the project.

References

[1] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny, "Automatic building extraction from DEMs using an object approach and application to the 3D-city modeling," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 63, no. 3, pp. 365–381, 2008.

[2] I. Tomljenovic, B. Höfle, D. Tiede, and T. Blaschke, "Building extraction from airborne laser scanning data: an analysis of the state of the art," *Remote Sensing*, vol. 7, no. 4, pp. 3826–3862, 2015.

[3] Y. Li, H. Wu, R. An, H. Xu, Q. He, and J. Xu, "An improved building boundary extraction algorithm based on fusion of optical imagery and LiDAR data," *Optik - International Journal for Light and Electron Optics*, vol. 124, no. 22, pp. 5357–5362, 2013.

[4] H. Li, C. Zhong, X. Hu, L. Xiao, and X. Huang, "New methodologies for precise building boundary extraction from LiDAR data and high resolution image," *Sensor Review*, vol. 33, no. 2, pp. 157–165, 2013.

[5] V. Saeidi, B. Pradhan, M. O. Idrees, and Z. Abd Latif, "Fusion of airborne LiDAR with multispectral spot 5 image for enhancement of feature extraction using Dempster–Shafer theory," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 10, pp. 6017–6025, 2014.

[6] M. Uzar and N. Yastikli, "Automatic building extraction using LiDAR and aerial photographs," *Boletim de Ciências Geodésicas*, vol. 19, no. 2, pp. 153–171, 2013.

[7] M. Uzar, "Automatic building extraction with multi-sensor data using rule-based classification," *European Journal of Remote Sensing*, vol. 47, no. 1, pp. 1–18, 2014.

[8] M. Awrangjeb, M. Ravanbakhsh, and C. S. Fraser, "Automatic detection of residential buildings using LIDAR data and multispectral imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 5, pp. 457–467, 2010.

[9] M. Awrangjeb, C. Zhang, and C. S. Fraser, "Building detection in complex scenes thorough effective separation of buildings from trees," *Photogrammetric Engineering & Remote Sensing*, vol. 78, no. 7, pp. 729–745, 2012.

[10] R. Wang, Y. Hu, H. Wu, and J. Wang, "Automatic extraction of building boundaries using aerial LiDAR data," *Journal of Applied Remote Sensing*, vol. 10, no. 1, article 016022, 2016.

[11] R. Prerna and C. K. Singh, "Evaluation of LiDAR and image segmentation based classification techniques for automatic building footprint extraction for a segment of Atlantic County, New Jersey," *Geocarto International*, vol. 31, no. 6, pp. 694–713, 2016.

[12] Z. Zhao, Y. Duan, Y. Zhang, and R. Cao, "Extracting buildings from and regularizing boundaries in airborne lidar data using connected operators," *International Journal of Remote Sensing*, vol. 37, no. 4, pp. 889–912, 2016.

[13] I. Tomljenovic, D. Tiede, and T. Blaschke, "A building extraction approach for airborne laser scanner data utilizing the object based image analysis paradigm," *International Journal of Applied Earth Observation and Geoinformation*, vol. 52, pp. 137–148, 2016.

[14] S. Gilani, M. Awrangjeb, and G. Lu, "An automatic building extraction and regularisation technique using LiDAR point cloud data and orthoimage," *Remote Sensing*, vol. 8, no. 3, p. 258, 2016.

[15] J. S. Evans and A. T. Hudak, "A multiscale curvature algorithm for classifying discrete return LiDAR in forested environments," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 4, pp. 1029–1038, 2007.

[16] R. M. Haralick and K. Shanmugam, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, pp. 610–621, 1973.

[17] L. Ma, T. Fu, T. Blaschke et al., "Evaluation of feature selection methods for object-based land cover mapping of unmanned aerial vehicle imagery using random forest and support vector machine classifiers," *ISPRS International Journal of Geo-Information*, vol. 6, no. 2, p. 51, 2017.

[18] M. A. Aguilar, R. Vicente, F. J. Aguilar, A. Fernández, and M. M. Saldaña, "Optimizing object-based classification in urban environments using very high resolution GeoEye-1 imagery," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. I-7, pp. 99–104, 2012.

[19] K. Charoenjit, P. Zuddas, P. Allemand, S. Pattanakiat, and K. Pachana, "Estimation of biomass and carbon stock in Para rubber plantations using object-based classification from Thaichote satellite data in Eastern Thailand," *Journal of Applied Remote Sensing*, vol. 9, no. 1, article 096072, 2015.

[20] G. Sohn and I. Dowman, "Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 62, no. 1, pp. 43–63, 2007.

[21] M. I. Sameen, F. H. Nahhas, F. H. Buraihi, B. Pradhan, and A. R. B. M. Shariff, "A refined classification approach by integrating Landsat Operational Land Imager (OLI) and RADARSAT-2 imagery for land-use and land-cover mapping in a tropical area," *International Journal of Remote Sensing*, vol. 37, no. 10, pp. 2358–2375, 2016.

[22] J. Zhang, "Multi-source remote sensing data fusion: status and trends," *International Journal of Image and Data Fusion*, vol. 1, no. 1, pp. 5–24, 2010.

[23] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[24] Y. Wang, H. Yao, and S. Zhao, "Auto-encoder based dimensionality reduction," *Neurocomputing*, vol. 184, pp. 232–242, 2016.

[25] G. Hinton, "A practical guide to training restricted Boltzmann machines," *Momentum*, vol. 9, p. 926, 2010.

[26] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[27] F. Chollet, "Keras," 2015, https://keras.io.

[28] Y. LeCun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[29] D. Mongus and B. Žalik, "Parameter-free ground filtering of LiDAR data for automatic DTM generation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, pp. 1–12, 2012.

[30] Z. Chen, B. Devereux, B. Gao, and G. Amable, "Upward-fusion urban DTM generating method using airborne Lidar data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 72, pp. 121–130, 2012.