remote sensing

MDPI

Article

Unseen Land Cover Classification from High-Resolution Orthophotos Using Integration of Zero-Shot Learning and Convolutional Neural Networks

Biswajeet Pradhan [1,2,*], Husam A. H. Al-Najjar [1], Maher Ibrahim Sameen [1], Ivor Tsang [3] and Abdullah M. Alamri [4]

1 The Centre for Advanced Modelling and Geospatial Information Systems (CAMGIS), Faculty of Engineering and IT, University of Technology Sydney, Sydney, NSW 2007, Australia; Husam.AL-NAJJAR@student.uts.edu.au (H.A.H.A.-N.); Maher.Alzuhairi@uts.edu.au (M.I.S.)

2 Department of Energy and Mineral Resources Engineering, Sejong University, Choongmu-gwan, 209 Neungdong-ro, Gwangjingu, Seoul 05006, Korea

3 Center for Artificial Intelligence, Faculty of Engineering and IT, University of Technology Sydney, Sydney 2007, Australia; Ivor.Tsang@uts.edu.au

4 Department of Geology & Geophysics, College of Science, King Saud Univ., P.O. Box 2455, Riyadh 11451, Saudi Arabia; amsamri@ksu.edu.sa

* Correspondence: Biswajeet.Pradhan@uts.edu.au or biswajeet24@gmail.com; Tel.: +61-2-9514-7937

Received: 13 March 2020; Accepted: 21 May 2020; Published: 23 May 2020

Abstract: Zero-shot learning (ZSL) is an approach to classify objects unseen during the training phase and shown to be useful for real-world applications, especially when there is a lack of sufficient training data. Only a limited amount of works has been carried out on ZSL, especially in the field of remote sensing. This research investigates the use of a convolutional neural network (CNN) as a feature extraction and classification method for land cover mapping using high-resolution orthophotos. In the feature extraction phase, we used a CNN model with a single convolutional layer to extract discriminative features. In the second phase, we used class attributes learned from the Word2Vec model (pre-trained by Google News) to train a second CNN model that performed class signature prediction by using both the features extracted by the first CNN and class attributes during training and only the features during prediction. We trained and tested our models on datasets collected over two subareas in the Cameron Highlands (training dataset, first test dataset) and Ipoh (second test dataset) in Malaysia. Several experiments have been conducted on the feature extraction and classification models regarding the main parameters, such as the network's layers and depth, number of filters, and the impact of Gaussian noise. As a result, the best models were selected using various accuracy metrics such as top-k categorical accuracy for k = [1,2,3], Recall, Precision, and F1-score. The best model for feature extraction achieved 0.953 F1-score, 0.941 precision, 0.882 recall for the training dataset and 0.904 F1-score, 0.869 precision, 0.949 recall for the first test dataset, and 0.898 F1-score, 0.870 precision, 0.838 recall for the second test dataset. The best model for classification achieved an average of 0.778 top-one, 0.890 top-two and 0.942 top-three accuracy, 0.798 F1-score, 0.766 recall and 0.838 precision for the first test dataset and 0.737 top-one, 0.906 top-two, 0.924 top-three, 0.729 F1-score, 0.676 recall and 0.790 precision for the second test dataset. The results demonstrated that the proposed ZSL is a promising tool for land cover mapping based on high-resolution photos.

Keywords: land cover classification; deep-learning; CNN; Zero-Shot Learning; remote sensing; orthophotos

1. Introduction

Remote sensing has been a standard tool for producing land use and land cover maps for decades [1]. These products play an important role in various applications, such as natural disaster management [2,3], environmental monitoring [4,5], urban planning [6,7] precision farming [8–11] and vegetation monitoring [12]. With advances in sensor engineering science, a deluge of high-resolution image data has been generated using various sensing technologies [13]. However, processing such a huge amount of data requires extensive training samples and advanced image-processing techniques, which are not always accessible [14]. One approach to overcome this problem is the use of classification models based on Zero-Shot Learning (ZSL). Such an approach is used to construct classification models for unseen classes that have not been labelled for training. It uses a semantic representation of classes (e.g., attributes, word vectors) as aside information and transfers knowledge from source (seen) classes to unseen classes. The applications of ZSL are becoming popular among researchers, both for classification and object recognition problems, and recently have received attention from the remote sensing community.

Before ZSL, a wide range of object recognition and classification models have been proposed and implemented, such as pixel-based and object-based approaches. Nevertheless, these common classification models can only classify objects that are seen during the training stage. These methods, for example, object-based image analysis (OBIA), have a powerful ability of classification, but they failed to detect and classify new (unseen) objects. In addition, they have their limitations such as finding an ideal segmentation scale, complex workflow, and carefully optimisation process [15–19]. Convolutional Neural Networks (CNNs) have also shown to be successful for land cover classification and other related applications [20]. Zhang et al. [20] used CNNs for high-resolution image classification with different feature learning strategies. Sumbul et al. [21] presented a CNN model to classify high-resolution aerial photographs for land cover classification. Al-Najjar et al. [1] used CNN for land use mapping from drone images. Chen et al. [22] detected various species of forest mangrove with a patch-based CNN and found that CNN is superior to support vector machine (SVM). However, these models are only successful if the test images do not contain unseen classes in the training stage [14]. Other limitations of these methods include the requirement of a large number of labelled samples to train the models efficiently and over-fitting issues. Considering that sufficient training samples with labels are expensive to collect, these traditional classification models are still inefficient for many real-world applications, including remote sensing image classifications.

To overcome these limitations, the present study proposes a classification framework based on ZSL for unseen land cover classes from high-resolution orthophotos. ZSL is a classification approach that employs the semantic word vector of each class as a bridge to the semantic word vector of the nearest class name, to learn the model from known samples to obtain unknown (unseen) classes. This technique has recently been applied to a few remote sensing applications such as scene classification [14], street tree classification [21], vehicle recognition [22], and land cover mapping [23,24]. Further explanation will be presented in the following section of related studies.

The current framework consists of (i) a Word2Vec model [25] for label embedding, (ii) a CNN model for image feature learning, and (iii) a K-nearest neighbour (KNN) model that provides label predictions for unseen classes that are not included in the training data.

In summary, although numerous ZSL works have been carried out in the field of computer vision, only a few works have been implemented in the remote sensing domain. Therefore, the main contribution of this research is to integrate the recent techniques in computer vision such as ZSL, Word2Vec models and CNN to address the existing problem of traditional classification systems in the field of remote sensing, such as the lack of sufficient training data for every class. The remainder of this article is organised as follows: Section 2 summarises the related work. Section 3 describes the proposed framework and methods. Section 4 presents the experiments and results. Section 5 provides the discussion. Lastly, Section 6 explains the conclusion and future works.

2. Related Studies

This section presents a summary of the previous works on ZSL, followed by a review of ZSL exclusively to remote sensing field applications.

2.1. Zero-Shot Learning (ZSL)

ZSL has progressed positively over the past few years due to its capability to recognise unseen classes with no training samples by transferring knowledge from seen classes to unseen classes [24]. Both seen and unseen classes are related in a high dimensional vector space, called semantic space, where the knowledge from seen classes can be transferred to unseen classes. In ZSL, the set of seen and unseen classes are disjointed, and it is assumed that the semantic representations are class specific. The relationship between the seen and unseen classes is often established by using semantic spaces. Examples of semantic representations used in ZSL studies include attributes, word vector, and text description. As traditional machine learning, ZSL consists of training and inference stages. In the training stage, the relationship between seen data and its corresponding semantic representation (attributes, word vectors) are learned. In the inference stage, the target image is parsed from an unseen class into its semantic representation to predict the corresponding label. The classification is simply performed via the nearest neighbour or via a more elaborate scheme like label propagation. This phase recognises new inputs and again leads to newer classes, regardless of the training data.

The common ZSL methods comprise three main stages, including extraction of visual instances (features), middle-class semantic information retrieval from training and testing classes by word embedding, which is a geometric approach for extraction of the semantic meaning using vectors, and a model that maps the features to the semantic meaning of the class labels (vectors) [14]. For the extraction of visual instances (features), models such as CNNs or morphological profiles are often used. These models extract more abstract features from the image pixels [20]. From the class labels, the semantic information is often retrieved as vectors using models like Word2Vec. This step is important as it provides a way to link the image features with their corresponding class semantic information statistically. Finally, any machine learning or statistical models can be applied to establish the relationship between the image features and the class vectors. This will enable the prediction of novel class labels for the new images, as the model requires only image features as input [15].

In recent work, the authors in [26] introduced an attribute-based ZSL technique to identify different types of animal image recognition. The experimental results on a different dataset showed that the model can recognise images without training samples of the target class. The authors in [27] proposed a visual abstraction ZSL technique to learn the concept of humans and their interactions. The experimental results unveiled a reasonable recognition of the human pose dataset. The authors in [28] introduced linearly combining classifiers for seen objects. To solve the problem due to a mixture of seen class proportions, the paper [16] introduced a semantic similarity embedding for the ZSL approach and obtained satisfactory results based on five datasets. In paper [29], the authors developed a hybrid relative attribute-based ZSL approach based on sparse coding to recognise images of faces and natural scenes. The model had strong grading and classification capability along the distinguishability of the non-semantic relative attributes in classification. The authors in [17] proposed a framework for zero-shot classification based on higher relations of multi-task mixed attributes and specific features of attributes. The authors utilised the AWA and PubFig dataset for their study and could achieve a robust accuracy over the traditional MTL attribute learning. However, they did not employ deep learning methods in their research due to the presence of low-level features in their case study. In another study [18], the concept of ZSL was utilised in the prediction for transportation networks utilising a fixed set of trains and unknown trains by employing knowledge transferring. The authors in [16] proposed semantic similarity embedding learning for zero-shot unseen objects using CNN for five benchmarks including SUN, CIFAR-10, aPascal and aYahoo, animals with attributes, and Caltech-UCSD Birds-200-2011 dataset. They labelled the semantics in a way that the empirical mean embeddings of the seen class data distributions are aligned with their corresponding source

domain embeddings. Considering the good performance of their method, they suggested using it for person recognition. In another study, the authors in [19] combined image matching, object detection, image retrieval and ZSL to overcome the semantic matching problem. They utilised the ILSVRC 2014 dataset for a single-shot semantic matcher with CNN architecture based on GoogleNet and YOLO/DetectNet architectures. They concluded that their semantic matcher approach is beneficial for real-time multi-class object recognition.

Despite the success of the previous ZSL models using standard datasets, most of these models performed the prediction by the absence of the seen label in the testing stage [24,30]. In these ZSL models, the research gap of the unseen class was exclusively narrow to the test label, whereas the problem sets wherein both train and test labels are concurrent, which is called transductive or generalised ZSL (GZSL), that is more challenging [24]. Additional details on this approach, including the positive and negative sides, can be found in the reference [15].

2.2. Transductive or GZSL

ZSL assumes that only the classes from testing samples are categorised into potential unseen classes. This assumption is invalid in the real-world because the samples from seen classes could be present in the testing set [15]. Nevertheless, GZSL can allocate (classify) the testing samples whether into seen and unseen classes [15,30]. Alternatively, GZSL is more complicated than ZSL because the information must be relocated from the main domain to the target. In addition, GZSL should be able to differentiate between seen and unseen classes. This is challenging as the majority of testing samples can be classified into a seen family, not a true unseen family. ZSL techniques can be used in GZSL, but they have a considerably lower accuracy [31]. Few studies have established a standard guideline for GZSL [32].

2.3. ZSL on Remote Sensing Data

Although notable developments of ZSL models have been published for various computer vision tasks, these models are rarely applied to remote sensing applications. The authors in [14] developed a novel zero-shot scene classification approach to recognise images from unseen scene classes. The authors used the Word2Vec model to map labels of seen/unseen scene classes to semantic vectors for describing the relationships between seen and unseen classes. Then, to transfer knowledge from seen classes to unseen classes, they used a label-propagation algorithm incorporating the semantic-directed graph and an unsupervised domain adaptation model. They applied the method on the UC Merced benchmark dataset and remote sensing data. The authors in [23] proposed the integration of ZSL into a dual-memory LSTM framework for land cover prediction. The method contains two memories that can capture both long-term and short-term variation patterns, which can effectively resolve the temporal variation. In another ZSL remote sensing application [33], synthetic aperture radar (SAR) was employed for target recognition with a ratio of unseen per seen classes of 1/7. Authors in [21] performed street trees classification using a multi-source region attention network for fine-grained object recognition tested on RGB, multispectral (MS), and LiDAR (light detection and ranging) data. The best performance of 17.7% was obtained by the proposed model trained using RGB and MS data. The more complex models that use all three sources had slightly lower performances due to a limited number of training samples. Chen et al. [22] utilised generalised ZSL for vehicle detection via a coarse-to-fine framework with latent attributes using ISPRS Potsdam 2D Semantic data. Their method showed the effectiveness of GZSL for unseen vehicles. Unseen object classification remains challenging, especially for remote sensing applications, because remote sensing data have particularly diverse structures and huge volumes compared to the data generally used in other fields such as computer science [20,34]. Therefore, in this study, a classification framework based on ZSL is proposed to detect and classify unseen objects by utilising seen classes and the success of CNN for feature learning.

3. Methods

ZSL enables land cover mapping in areas that contain novel (unseen) objects, which are useful in real-world applications. This research proposed a framework for ZSL-based land cover classification using orthophotos. The framework employed three main techniques, including Word2Vec, CNNs, and KNN. The models above were used for three main functions, namely class embedding, feature extractions, and classification of unseen objects based on their semantic similarity, respectively.

3.1. Theoretical Background

3.1.1. Theory of ZSL

ZSL is part of transfer learning due to heterogeneous transfer learning [15]. The main goal of ZSL is to perform a task without using any sample of that task at training data. For example, the land cover classification task in areas that contain novel classes not included in the training data can be regarded as an example of ZSL. In a simple term, ZSL allows classifying and recognising the unseen objects. ZSL consists of seen and unseen classes, which refer to labelled training and unlabelled testing samples, respectively. Each sample is assumed to be a part of one class and represented by a vector in feature space. Feature space is generally a real number space. $S = \left\{c_i^s | i = 1, \ldots, N_s\right\}$ represents the group of seen classes, where each c_i^s is a seen class. $U = \left\{c_i^u | i = 1, \ldots, N_u\right\}$ denoted the group of unseen classes, where each c_i^u is an unseen class. Note that $S \cap U = \varnothing$. $\mathbb{X}$ indicates the feature space, which is a real number space R^D. $G^{train} = \left\{\left(x_i^{train}, y_i^{train}\right) \in \mathbb{X} \times S\right\}_{i=1}^{N_{train}}$ represents the group of labeled training data for seen classes; for each labelled sample $\left(x_i^{train}, y_i^{train}\right)$, x_i^{train} is the sample in the feature space, and y_i^{train} is the corresponding class label. $X^{test} = \{(x_i^{test} \in \mathbb{X}\}_{i=1}^{N_{test}}$ denotes the group of testing samples, where x_i^{test} is a testing sample in the feature space. $Y^{test} = \{(y_i^{test} \in U\}_{i=1}^{N_{test}}$ indicates the corresponding class label for X^{test}, which is to be predicated. ZSL attains to learn the zero-shot classifier $f^u(0) : \mathbb{X} \rightarrow U$, which categorises testing samples X^{test} (i.e., to predict Y^{test}) corresponding to the unseen classes U. Thus, training and zero-shot classes exist. No samples from zero-shot classes are used during training.

3.1.2. Theory of CNN

CNN is a special type of neural network that was designed for image (or array-like) data under the concept of convolution and has shown significant success in the field of computer vision and recently in the remote sensing domain. CNN was first introduced by the paper [35] and was improved further by other researchers through advances in computing and software technologies. CNN utilises local connections, shared weights and a wide range of computing layers [36]. Consequently, CNN can efficiently extract features from the input image data without considerable intervention from humans. In comparison with feed-forward neural networks and other machine learning models, CNN has shown a strong predictive capability given that adequate training samples are available. CNN is also computationally efficient because it can compute convolutions in parallel with multiple GPU cores [36].

CNN consists of a series of convolutional and pooling layers, followed by a classification layer (e.g., softmax). Other layers, such as dense (fully connected), dropout and batch-normalisation layers, were optionally added to the model to improve its generalisation and predictive capacity. However, adding these layers does not necessarily improve the model's predictions unless proper optimisation is considered. The main component of CNN is the convolutional layer, which is composed of several convolutional kernels. The convolutional layer is associated with a small area of the input data—also known as an image patch. If $I_{x,y}$ is the given image, then the convolution is performed through $F_l^k = \left(I_{x,y}. K_l^k\right)$, where x, y shows the spatial locality and K_l^k represents the l^{th} convolutional kernel of the k^{th} layer. Pooling is another important component of CNN, which summarises information across adjacent spatial regions. For each patch in the feature map, pooling calculates the average (or other functions, e.g., max and min) value. Global pooling can be performed to sample the entire feature map

into a single value. Pooling is expected to improve model invariance to local translation and reduce model parameters. These layers are calculated using $Z_l = f_p\left(F^l_{x,y}\right)$, where Z_l represents the l^{th} output feature map, $F^l_{x,y}$ is the l^{th} input feature map and $f_p(.)$ defines the type of pooling operation [1,37].

3.1.3. KNN Model

K-nearest neighbour (KNN) is one of the simplest classification algorithms that stores all available cases and is often utilised as the first choice for a classification task when there is little or no previous knowledge about the distribution of the data. It is simple with no assumptions about data, relatively accurate and easy to implement, which makes it more attractive than other approaches. Moreover, In ZSL studies, the interpretation is generally performed based on the nearest neighbour scheme [15,24]. Therefore, KNN was used in the current study due to the aforementioned advantages.

KNN search is performed in the embedding space to match the projection of an image feature vector against that of an unseen class. In KNN, the output is the participation of a certain class. An object is categorised by a majority vote of its neighbours, with the object being designated to the class with the most frequent one among its K-nearest neighbours, in which the K value is a positive integer and often small [38,39].

3.1.4. Class Embedding

In ZSL, the label of each class is a text (word or phrase). The class labels are assumed to provide semantic side information about the classes. This aside information is considered by learning the semantic (numeric) vector representation of the class labels. These semantic representations will then play a key role in transferring knowledge from the seen to unseen classes. For this purpose, we use the commonly used Word2Vec model published by Google. Word2Vec is an efficient model for learning semantic vector representations of words and phrases. Similar words or phrases are embedded as nearby vectors, whereas dissimilar words or phrases are embedded as far vectors.

In this research, we use the pre-trained Word2Vec model published by Google to create vector representations for class labels. This contains 300-dimensional representations of the vectors for around 3 million terms and phrases. Since this pre-trained model is case-sensitive, we cast the text label in a lower case. For the single-word labels, we use the semantic word vector as the class label embedding for a class. For multi-word labels, an embedding of a class label is obtained by calculating the mean of the individual semantic term vectors. For more information about this model, a recent work for label embedding can be found in reference [40].

3.2. Overall Workflow

The overall workflow of the proposed ZSL framework for the land cover mapping from orthophotos is shown in Figure 1. In the training phase, the training image with training classes (e.g., agricultural land, urban areas, barren land, forest lands, road, and water body) is fed into CNN-1 for feature extraction. The output of this step is image vectors. The training classes are fed into the Word2Vec model to convert the class labels into vectors. These word vectors are trained based on Google News data, which is prepared by Google, with a dimension of 300 unique numbers [25]. The output of this step is a class signature with a size of 300, multiplied by the number of training classes.

Then, CNN-2 is used to map the image vectors to the class signature. In the test area, first, the image is fed to the trained CNN-1 to extract the image features. The ZSL classes are then fed into another Word2Vec model (which acts here as an auxiliary label's bank) to extract the class signature. To avoid predicting labels for unseen classes that are not relevant to land cover applications, we used the KNN model to find labels from a label bank created for the purpose of land cover mapping. It contains labels only suitable for land cover e.g., agricultural land, barren land, road, forest, and urban area etc. Thereafter, the CNN-2, which is trained on the training dataset, is used along with the image features to predict the class signature from the label's bank [41]. Once the class signature is predicted, the class

labels are determined by the KNN algorithm to measuring the similarity between the predicted class signature and the class signature, which was pre-computed from the label's bank for all possible class labels. Lastly, the predicted class labels are used to produce the land cover map for the area.

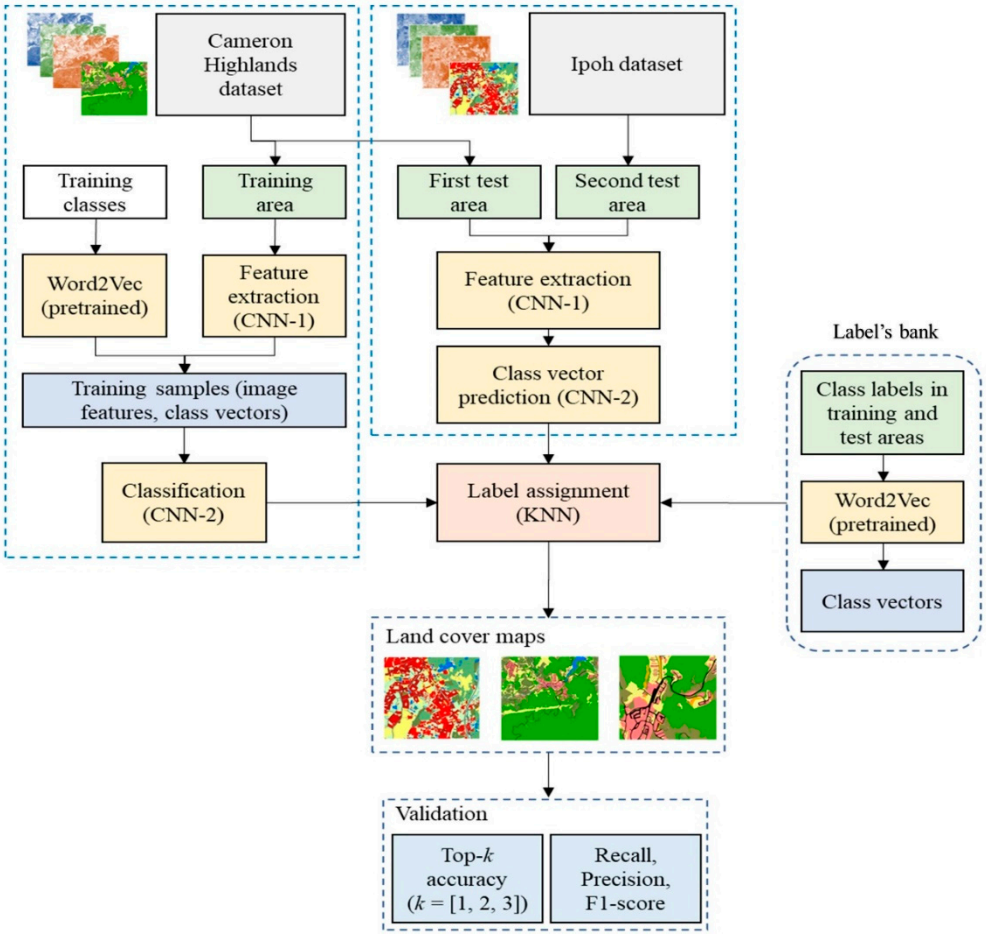

Figure 1. The overall workflow of the proposed ZSL framework.

3.3. Details of CNN-1 Used for Image Feature Extraction

In this research, CNN-1 is designed to extract features from input orthophotos. It is trained on image pixels (patches) with the corresponding pixel-level class labels (from the ground truth data). In this research, CNN is used other than any machine learning model because of its capability to utilise spectral and spatial information effectively [20]. The spectral content can be obtained by three visible bands (RGB) of the orthophotos. Instead, the spatial content is attainable in considering neighbouring pixels [42]. The weights in CNNs are adjusted during training to result in learned filters or nodes, which are equivalent to handmade filters, the outputs of which during the feed-forward stage are the feature maps. Based on the experiments above, the best model then was selected among the other models. Thus, after conducting several experiments (i.e., explained in detail in the experimental set-up of CNN-1 section), the network was established based on a single convolutional layer (Conv2D) with a kernel size of 3 × 3 and a filter size of 64. After Conv2D, we used batch normalisation with ReLU activation to accelerate convergence and introduce regularisation within the network. A dropout of 0.3 is utilised to regularise the network further. Then, a flatten layer is adopted to convert the features from 2D to 1D shape. A dense layer is added on top of the dropout layer with ReLU activation. The number of units used is 32. Finally, a softmax layer is used to perform classification. We train the model with an Adam optimiser (initial learning rate = 0.001) with a batch size of 2048 for 100 iterations.

The CNN-1 model is trained on image pixels (patches) with the corresponding pixel-level class labels (from the ground truth data). It aimed at extracting image features from a given image pixel. Figure 2 presents the architecture of the model.

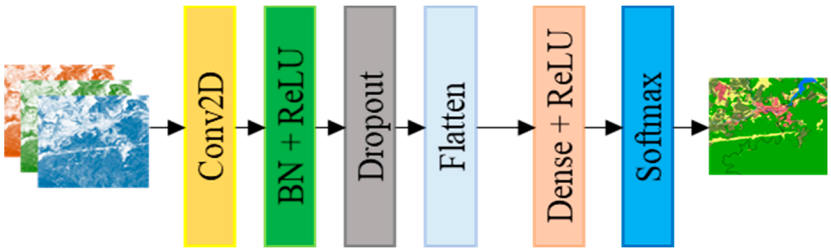

Figure 2. The architecture of CNN-1 used for feature extraction in the proposed ZSL framework.

3.4. Details of CNN-2 Used for Class Signature Prediction

Figure 3 shows the architecture of CNN-2 used for class signature prediction given the image vectors. This model was chosen due to the general robustness of CNNs over the traditional classification techniques in land cover classification [20]. The model consists of a convolutional layer with ReLU activation. The number of filters in this layer was 128. We included a batch normalisation with ReLU to expedite convergence and introduce regularisation within the network. A dropout layer is then added to avoid over-fitting with a drop probability of 0.3. Thereafter, we added two dense layers with ReLU activations. The number of units in the first dense layer was 64. The second dense layer takes the number of attributes (300), which is pre-trained by Google News. The target kernel is initialised with the class signature pre-computed by the Word2Vec model. The final layer is softmax with aims at classification. The model is trained with an Adam optimiser (initial learning rate = 0.001) with a batch size of 2048 for 100 iterations. Table 1 presents the parameters of the CNNs set in this study. Different and several hyper-parameters were tested during the design and optimisation process for the CNNs configuration. The best result was obtained using the aforementioned parameters. Moreover, by adding more layers, no constructive progress was observed (experiments with details are provided in the experimental set-up of CNN-2).

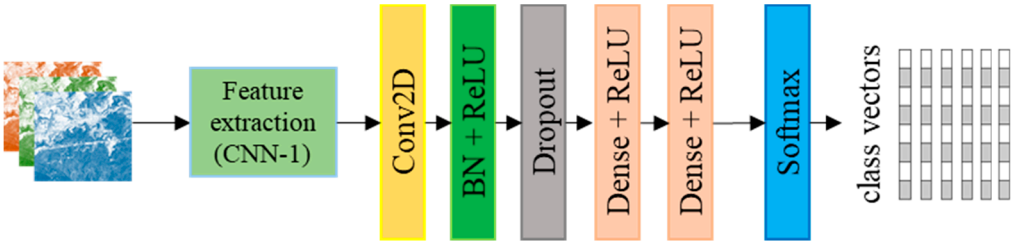

Figure 3. The architecture of CNN-2 used for class signature prediction in the proposed ZSL framework.

3.5. KNN Algorithm for Finding the Nearest Class/Label

In this algorithm, the assumption is based on the idea that the adjacent instances should have a similar label. Considering the initial clustering centres, K-means clustering is performed on the testing instances. Then, the one-to-one similarity between clustering centres and unseen class samples are calculated using linear programming. Occurrences in each cluster are categorised into the related unseen class [39]. In the current research, KNN was separately applied to six scenarios of different unseen classes to find the closest class (one unseen class for each scenario).

Table 1. Details of the selected CNN-1 and CNN-2 architectures.

Details of Layers	**CNN-1**	**CNN-2**
Duty	Feature extraction	Class signature prediction
Kernel size	3 × 3	3x3
Number of filters	64	128
Activation function	ReLU	ReLU
Drop out	0.3	0.3
Flatten layers	1	None
Dense layers	1	2
Size of dense layers	32	64/300 (# attributes)
Classifier	Softmax	Softmax
Optimiser	Adam (L. Rate = 0.001)	Adam (L. Rate = 0.001)
Batch size	2,048	2,048
Iterations	100	100

3.6. Evaluation Metrics (Precision, Recall and F-measure)

The F-measure is the weighted normal or symphonious mean of two proportions, known as exactness or precision (p). The recall (r) metric is another presentation measure used to evaluate the class-specific exactness accuracy from recovered information/data. Its computation is based on Equation (1), based on the average of p and r. The F-measure esteem ranges from Zero (0) as lowest to one (1) as the highest value [33,43].

$$F_{measure} = 2 \times \frac{p \times r}{p + r} \quad (1)$$

The p or the certainty of an unseen land cover class is defined by dividing the true positives values (number of pixels having a place with the real class) with the total number of objects categorised as the positive class (for example, the sum of true positives and false positives, which are objects/pixels erroneously categorised as having a place with the class). The sensitivity (r) represents the extent of true positive objects/pixels that are correctly predicted and distinguished, and can be characterised as the number of true positives divided by the total number of objects/pixels that belong to the positive class (for example, the sum of true positives and false negatives). The determination of p and r can be performed utilising the Equations (2) and (3). An ideal predictor's value for p and r would be depicted as 1 [43].

$$p = \frac{\text{True positives}}{\text{True positives} + \text{False positives}} \quad (2)$$

$$r = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}} \quad (3)$$

4. Experiments and results

4.1. Datasets

In this research, orthophotos acquired over the Cameron Highlands, Malaysia, on January 15, 2015, by using an airborne system (RIEGL) with an RGB camera, were used as the first dataset. The average height of the system whilst collecting data was 1510 m. The spatial resolution of the data was 1 m. Two subset areas were selected for training and testing the proposed ZSL framework. The training area consisted of six types of land covers, namely, agricultural land, barren land, road, forest, urban area and water body. The test area included five types of land covers (agricultural land, barren land, road, forest and urban area) with an additional class (croplands). Figure 4 presents details about the number of pixels for each class in the first training and test areas. For the training area, agricultural land contained 2,021,470 pixels, followed by barren land with 1,142,705 pixels. Road, forest lands, urban areas and water body contained 395,206, 7,819,007, 558,300 and 142,896 pixels, respectively. In total, the number

of training pixels was reported at 12,079,584 for all classes with a dimension of 4464 × 2706 × 3. The test area contained 1,423,087 pixels with dimensions of 863 × 1649 × 3. Agricultural land contained 229,060 pixels, followed by barren land with 113,040 pixels. Road, forest lands, urban areas and croplands contained 76,436, 829,964, 152,786 and 21,801 pixels, respectively. Figure 5 exhibits the maps of the first training and test areas, including the orthophotos and ground truth data.

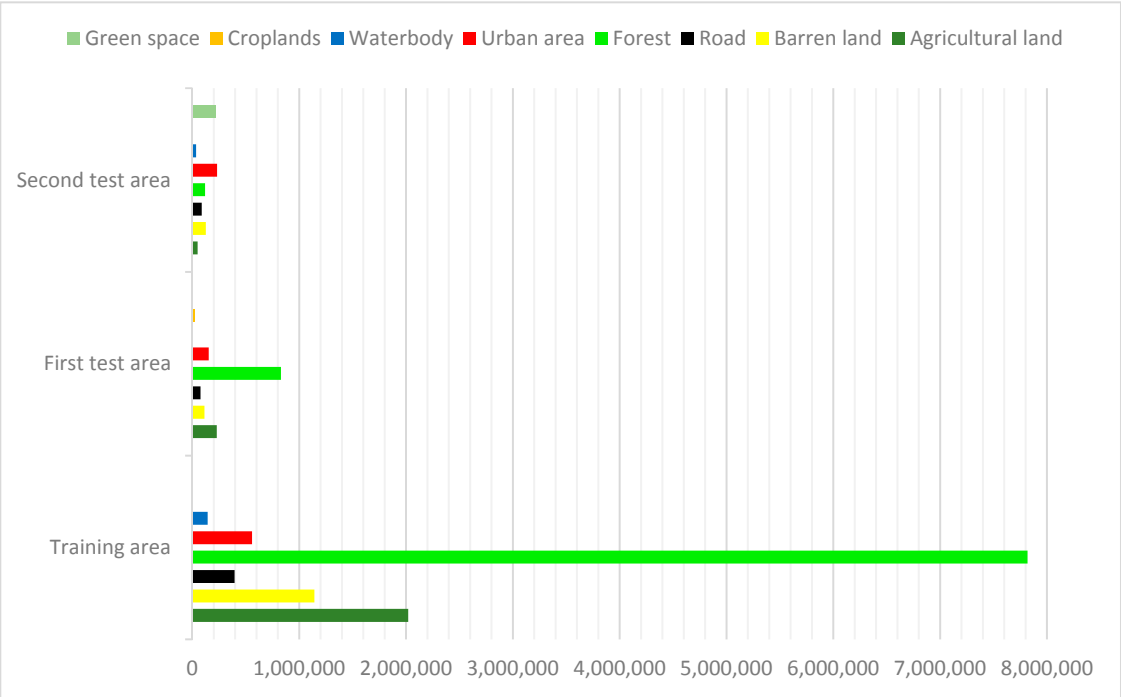

Figure 4. The number of pixels per land cover class in the ground truth dataset for training and test areas.

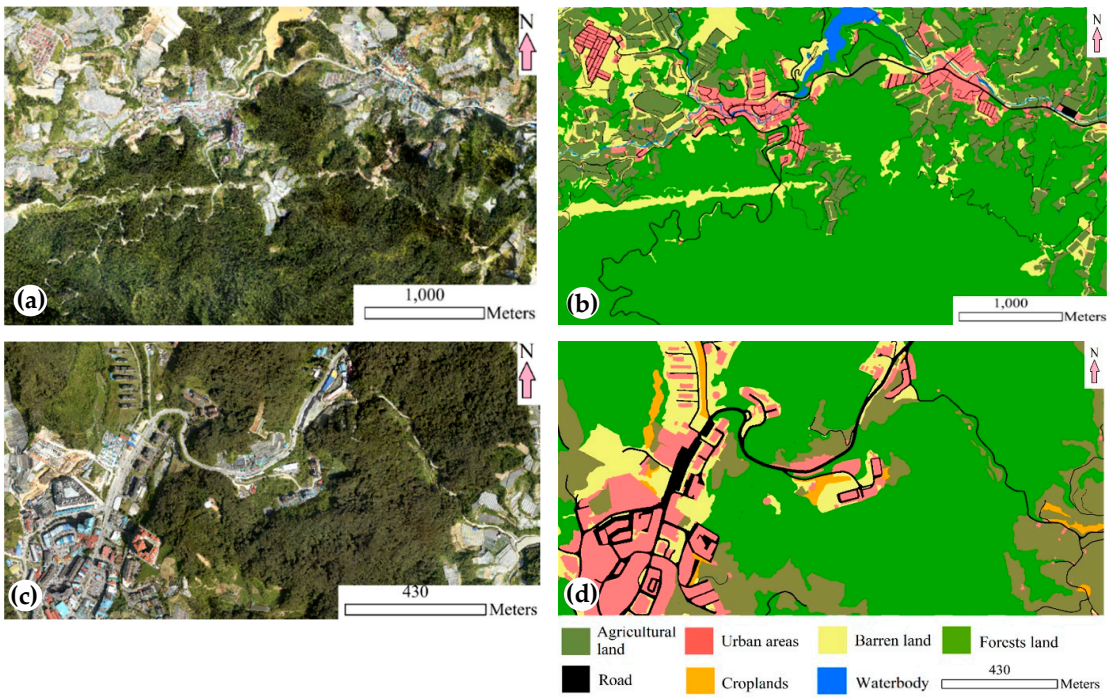

Figure 5. (**a**) Training of first dataset and (**b**) Ground truth map of the first dataset. (**c**) First test dataset and (**d**) Ground truth map of the first test dataset.

To further assess the performance and robustness of the current ZSL framework, a second dataset also was tested. The new dataset was taken from the Ipoh area (in Peninsular Malaysia) on January 15, 2015, by using an airborne system (RIEGL) with an RGB camera and the average height of the system whilst collecting data was 1000 m. The area contained seven land cover types including green space, road, urban area, barren land, forest, water body and agricultural land. Figure 4 presents the number of ground truth pixels for each class. The maps of the second test dataset, including the orthophotos and ground truth data, are shown in Figure 6.

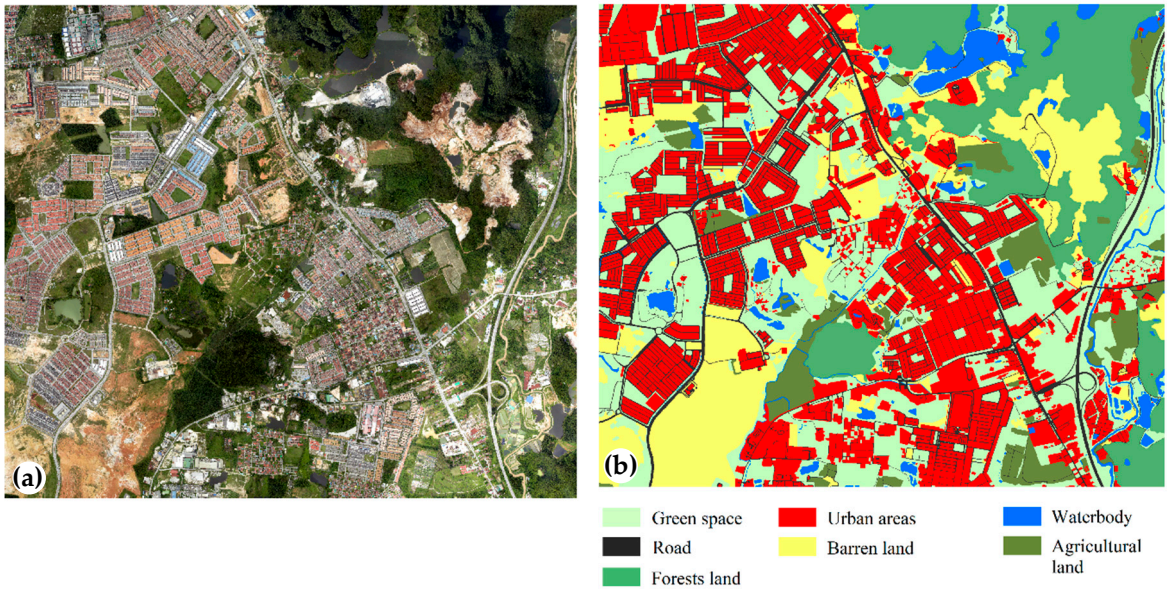

Figure 6. (**a**) Second test dataset and (**b**) Ground truth map of the second test dataset.

4.2. Results of Feature Extraction (CNN-1)

CNN-1 was used for feature extraction, which is an important step in the proposed ZSL framework. These features were extracted from the second last layer (dense) before the softmax layer. The number of extracted features was 32 per pixel. The high-level features performed better than the features extracted from shallow layers [42]. This model was tested separately using the ground truth datasets for the training and test areas. The model accuracy was measured by the recall, precision, F1-score and top-k categorical accuracy on training and test areas. For the training area, the model achieved a 0.882 recall, 0.941 precision, 0.953 F1-score and 0.760 top-one, 0.892 top-two, 0.963 top-three. For the first test area, the model performed well with a 0.949 recall, 0.869 precision, 0.904 F1-score and 0.805 top-one, 0.911 top-two, 0.963 top-three. The evaluation of the model over the second test area also showed a satisfactory performance with a 0.838 recall, 0.870 precision, 0.898 F1-score, 0.708 top-one, 0.886 top-two and 0.974 top-three, respectively.

4.2.1. Impact of Network Architectures

CNN-1 model was used for feature extraction, which is an important step in the proposed ZSL framework. These features were extracted from the second last layer (dense) before the softmax layer. For this purpose, we developed and examined various CNNs architectures applied to the first and second datasets. The results are demonstrated in Tables 2–5.

Table 2. The impact of various architectures applied to the first and second datasets.

Model	Metric	Training Area	First Test Area	Second Test Area
CNN without Batch Normalisation layer	Recall	0.780	0.748	0.733
	Precision	0.927	0.885	0.826
	F1	0.882	0.845	0.815
	Top-three	0.947	0.897	0.924
	Top-two	0.857	0.789	0.846
	Top-one	0.696	0.673	0.649
CNN without Pooling layer	Recall	0.882	0.949	0.838
	Precision	0.941	0.869	0.870
	F1	0.953	0.904	0.898
	Top-three	0.963	0.963	0.947
	Top-two	0.892	0.911	0.886
	Top-one	0.760	0.805	0.708
CNN without Batch normalisation and pooling layers	Recall	0.834	0.807	0.777
	Precision	0.936	0.860	0.844
	F1	0.921	0.874	0.850
	Top-three	0.957	0.905	0.938
	Top-two	0.877	0.804	0.866
	Top-one	0.733	0.684	0.675
CNN with batch normalisation and pooling layers	Recall	0.859	0.835	0.809
	Precision	0.941	0.877	0.857
	F1	0.939	0.899	0.875
	Top-three	0.959	0.904	0.939
	Top-two	0.882	0.812	0.875
	Top-one	0.747	0.709	0.691

Table 3. Impact of the number of convolutional filters on the feature extraction (CNN-1) for the first and second datasets.

Number of Filters in Conv. Layer	Metric	Training Area	First Test Area	Second Test Area
128	Recall	0.896	0.853	0.764
	Precision	0.946	0.854	0.881
	F1	0.964	0.900	0.817
	Top-three	0.966	0.889	0.931
	Top-two	0.899	0.810	0.858
	Top-one	0.770	0.704	0.630
64	Recall	0.882	0.949	0.838
	Precision	0.941	0.869	0.870
	F1	0.953	0.904	0.898
	Top-three	0.963	0.963	0.947
	Top-two	0.892	0.911	0.886
	Top-one	0.760	0.805	0.708
32	Recall	0.871	0.931	0.810
	Precision	0.941	0.963	0.858
	F1	0.946	0.971	0.876
	Top-three	0.962	0.959	0.940
	Top-two	0.887	0.905	0.875
	Top-one	0.753	0.793	0.692

Table 2 shows the impact of four architectures applied to the first and second datasets, including: (1) CNN without batch normalisation layer, (2) CNN without pooling layer, (3) CNN without batch normalisation and pooling layers, and (4) CNN with batch normalisation and pooling layers. The performance of CNN without the pooling layer was superior, with an F1-score of 0.953 for the training dataset, 0.904 for test dataset and 0.898 for the second test dataset, respectively. This was

followed by the architecture of CNN using both batch normalisation and pooling layers with an F1-score of 0.939 for the training dataset, 0.899 test dataset, and 0.875 for the second test dataset. Therefore, the layer with superior performance was used in the current research.

Table 4. Impact of the network's depth on feature extraction for the first and second datasets.

Number of Conv. Layers	Metric	Training Area	First Test Area	Second Test Area
1 Conv2D layer	Recall	0.882	0.949	0.838
	Precision	0.941	0.869	0.870
	F1	0.953	0.904	0.898
	Top-three	0.963	0.963	0.947
	Top-two	0.892	0.911	0.886
	Top-one	0.760	0.805	0.708
2 Conv2D layers	Recall	0.805	0.882	0.705
	Precision	0.939	0.971	0.824
	F1	0.903	0.965	0.795
	Top-three	0.948	0.951	0.913
	Top-two	0.863	0.890	0.836
	Top-one	0.719	0.770	0.634
3 Conv2D layers	Recall	0.791	0.827	0.761
	Precision	0.941	0.901	0.850
	F1	0.894	0.939	0.842
	Top-three	0.944	0.938	0.930
	Top-two	0.860	0.870	0.855
	Top-one	0.712	0.742	0.669

Table 5. Impact of Gaussian noise (mu = 0, sigma = 0.1) on feature extraction for first and second dataset.

Noise Impact	Metric	Training Area	First Test Area	Second Test Area
Without noise	Recall	0.882	0.949	0.838
	Precision	0.941	0.869	0.870
	F1	0.953	0.904	0.898
	Top-three	0.963	0.963	0.947
	Top-two	0.892	0.911	0.886
	Top-one	0.760	0.805	0.708
With noise	Recall	0.858	0.933	0.816
	Precision	0.931	0.864	0.861
	F1	0.934	0.893	0.881
	Top-three	0.959	0.960	0.941
	Top-two	0.882	0.906	0.877
	Top-one	0.742	0.795	0.695

4.2.2. Impact of Filters Size

Moreover, the influence of filter size in convolutional layers was also examined using several configurations. As mentioned in Table 3, the experiment with 64 filters demonstrated better performance, with an F1-score of 0.953 and 0.904 for the training area and first test area, as well as an F1-score of 0.898 for the second test area. Therefore, the filter size of 64 was set for CNN-1's network.

4.2.3. Impact of Network's Depth

To assess the impact of the network's depth, three depths (1, 2, 3 Conv2D layers) were examined based on the training and test datasets. The results demonstrated that using one Conv2D layer outperformed the other structures with an F1-score of 0.953 and 0.904 for the training and first test dataset, and 0.898 for the second test dataset (Table 4).

4.2.4. Impact of Gaussian Noise

In this work, to investigate the sensitivity to the quality of training samples, the Gaussian noise technique was employed. This additive noise can have a regularising effect and reduce over-fitting [44–46]. The performance analysis from Table 5 shows that the model performed well for the case without noise by (0.953 F1-score for the training dataset, 0.904 for the first test dataset, 0.898 for the second test dataset). The results also indicated that the network's performance was slightly dropped by adding the Gaussian noise to the input samples of the first and second datasets. This could imply that the network was not very sensitive by additional noise, inferring that the quality of the training data was good.

4.3. Results of Classification (CNN-2)

The CNN-2 model was utilised as a classifier for signature prediction, which is an essential stage in the proposed ZSL framework. These signature classes (vectors) were extracted from the softmax layer. The second dense layer before the softmax layer takes the number of attributes (300), which is pre-trained by Google News. The target kernel is initialised with the class signature pre-computed by the Word2Vec model. The softmax as a final layer was responsible for the classification (signature prediction). To this end, we conducted diverse experiments on CNN's architectures applied to the training and test dataset. The following subsections describe the experiments in detail.

The proposed ZSL model was used to generate land cover maps of the training and testing areas. For this purpose, six scenarios were applied to the testing area of the first dataset by considering different unseen classes separately. In other words, only one class was considered an unseen class in each scenario. Figure 7c–h displays the classification results obtained from the unseen class of road, urban areas, barren land, agricultural land, forest lands, and croplands for the first testing area, respectively (each singular unseen case was shown by a reddish box around the corresponding unseen class in the legend). As shown in Figure 7c, the road was the unseen class that was detected in the test image; the other testing classes consisted of urban areas, barren land, agricultural land, forest lands, and croplands. The next scenario was the detection of the urban area as an unseen class which is shown in Figure 7d. Likewise, the remaining unseen scenarios for the barren land, agricultural land, forests land, and croplands are shown in Figure 7e–h, respectively. Table 6 shows the accuracy of each corresponding unseen scenario.

Table 6. Retrieved labels for unseen classes for the first test dataset (Cameron Highlands).

Unseen Class	F1-Score	Recall	Precision	Top-One	Top-Two	Top-Three
Agricultural land	0.862	0.827	0.899	0.831	0.918	0.952
Barren land	0.842	0.809	0.877	0.813	0.914	0.952
Urban areas	0.825	0.782	0.874	0.796	0.897	0.944
Road	0.819	0.872	0.772	0.790	0.894	0.943
Forest lands	0.803	0.750	0.863	0.775	0.884	0.938
Croplands	0.639	0.558	0.747	0.665	0.837	0.927
Average	0.798	0.766	0.838	0.778	0.890	0.942

In general, the average accuracy of the proposed model for prediction of the unseen classes based on the top-K categorical accuracy was 0.778 top-one, 0.890 top-two, 0.942 top-three, as well as 0.798 F1-score, 0.766 recall, 0.838 precision, respectively.

As illustrated in Figure 7d, it can be seen that the highest accuracy is achieved when the unseen class is designated as an agricultural land with 0.862 F1-score, 0.827, 0.899 precision, 0.831 top-one, 0.918 top-two and 0.952 top-three. This result might be attributed due to the good semantic matching of agricultural lands in the pre-trained Word2Vec model and good detection of this class in the feature extraction phase. However, the lowest accuracy based on the top-one belongs to the scenario wherein the unseen class was croplands with an accuracy of top-one 0.665 (Figure 7h). That is because the areas of croplands could be misclassified with other spectrally closed classes such as agricultural and forest lands. Likewise, the lowest accuracy of croplands could be observed considering the other

metrics including the F1-score, precision, recall, top-two and top-three with values of 0.639, 0.747, 0.558, 0.837 and 0.927, respectively. All the test cases retrieved the orange colour for the croplands in the map. However, as the cropland's pixels are limited and highly scattered within the map, for better visualisation, we pointed out the croplands class by a yellowish rectangular box in Figure 7h.

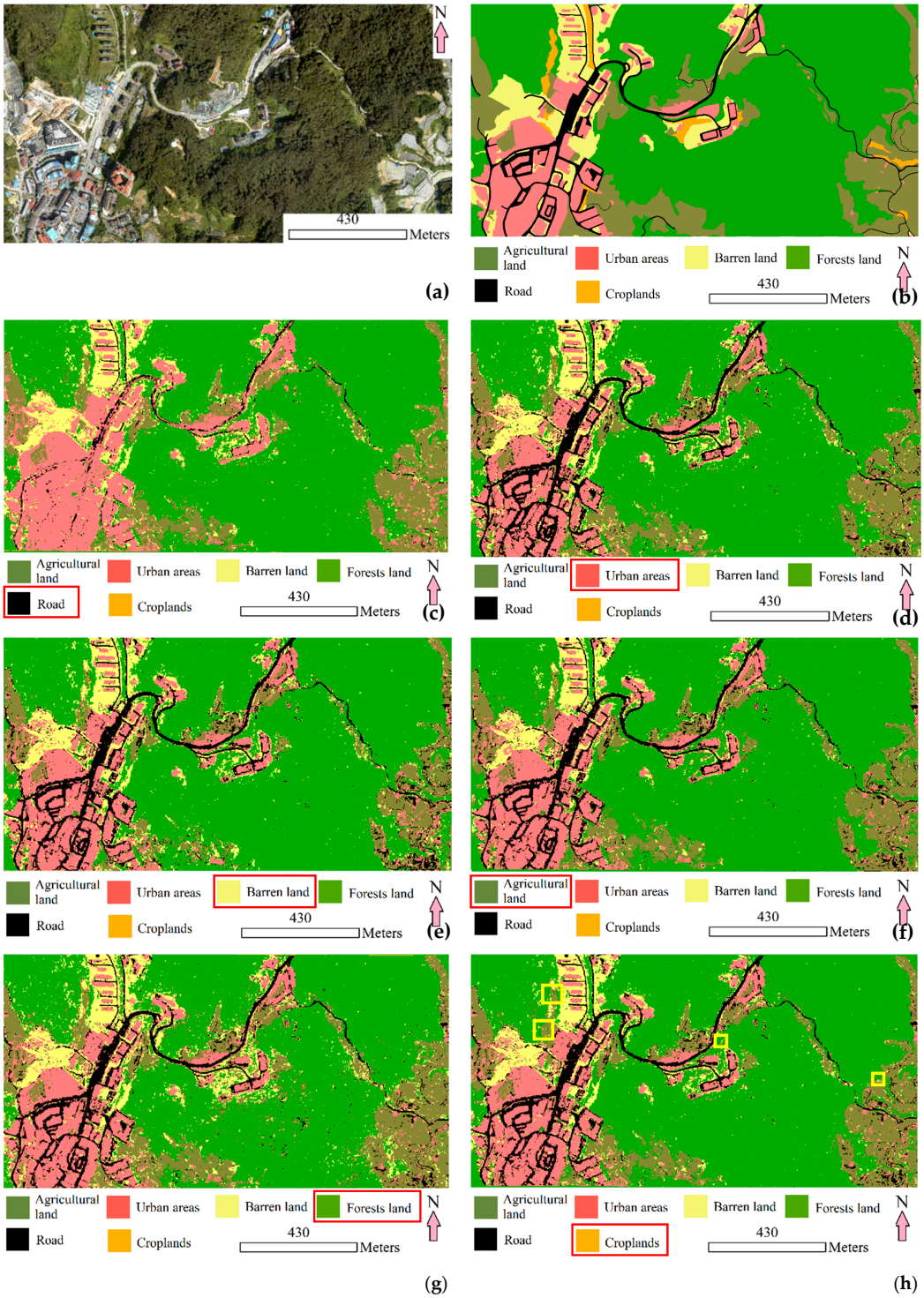

Figure 7. (**a**) Testing of the first dataset (Cameron Highlands area), (**b**) Ground truth map of testing dataset, and classification results of the testing area with different unseen classes: (**c**) road, (**d**) urban areas, (**e**) barren land, (**f**) agricultural land, (**g**) forest lands, and (**h**) croplands.

Overall, although the average top-one accuracy for all unseen classes did not reach 80%, the top-two and top-three approximately raised to 90% and above, which is a promising result given that the same probability in the traditional classification systems is almost nil due to absence of the label in the training stage. This can infer that two potential classes could be designated as correct classes (top-two), and three potential correct classes based on (top-three). Thus, this outcome could help experts in the geoscience field to select and advise the most appropriate class among two or three possible classes.

4.3.1. Impact of Batch Normalisation

Batch normalisation is a technique used in the training of CNN. Its function is to standardise the input to a layer for every mini-batch by adjusting and scaling the activation. It stabilises the training task which causes a reduction in the number of training epochs [1,47].

We evaluated the CNN-2 network to inspect whether the input layer is benefiting from batch normalisation during the training and prediction. To this end, two architectures were developed by considering the presence and absence of batch normalisation in the network.

Table 7 shows the comparative experiment of including batch normalisation in the network for the first training dataset. It appears that using the batch normalisation layer slightly enhanced the performance of the network during the training stage. The improvement was subsequently represented by mean F1-score of 0.806 reaching 0.809, mean recall of 0.752 reaching 0.757, mean precision of 0.871 reaching 0.872, mean top-one 0.812 reaching 0.814, mean top-two 0.21 reaching 0.923 and mean top-three 0.972 reaching 0.973.

Table 7. Accuracy of the model without and with batch normalisation for the training area.

Unseen Class	F1-Score	Recall	Precision	Top-One	Top-Two	Top-Three
Without batch normalisation						
Urban areas	0.831	0.776	0.893	0.831	0.925	0.976
Barren land	0.863	0.824	0.906	0.862	0.943	0.979
Agricultural land	0.874	0.841	0.910	0.873	0.948	0.978
Forest lands	0.612	0.527	0.731	0.646	0.862	0.952
Road	0.829	0.774	0.893	0.830	0.924	0.975
Water body	0.830	0.775	0.893	0.830	0.924	0.975
Mean	0.806	0.752	0.871	0.812	0.921	0.972
With batch normalisation						
Urban areas	0.832	0.780	0.892	0.833	0.927	0.976
Barren land	0.865	0.827	0.907	0.864	0.944	0.980
Agricultural land	0.876	0.844	0.911	0.875	0.950	0.979
Forest lands	0.621	0.536	0.738	0.652	0.866	0.953
Road	0.832	0.780	0.892	0.833	0.927	0.976
Water body	0.832	0.780	0.892	0.832	0.926	0.975
Mean	0.809	0.757	0.872	0.814	0.923	0.973

The same experiment was performed on the first test dataset. The results from Table 8 show that batch normalisation also improved the network by mean values of F1-score from 0.791 to 0.798, recall from 0.740 to 0.751, top-one from 0.773 to 0.778, top-two from 0.889 to 0.890 and top-three from 0.930 to 0.942, respectively.

The experiment on the second dataset (Table 9) showed a similar result compared to the first dataset. Specifically, when using batch normalisation, the model performed slightly better than the case without batch normalisation with subsequent mean values of F1-score from 0.722 reaching 0.729, recall from 0.668 reaching 0.676, precision from 0.786 reaching 0.790 and top-one from 0.731 reaching 0.737, top-two from 0.901 reaching 0.906 and top-three from 0.920 reaching 0.924.

Table 8. Accuracy of the model without and with batch normalisation for the first test area (Cameron Highlands).

Unseen Class	F1-Score	Recall	Precision	Top-One	Top-Two	Top-Three
Without batch normalisation						
Urban areas	0.819	0.773	0.871	0.791	0.891	0.931
Barren land	0.837	0.802	0.876	0.808	0.910	0.941
Agricultural land	0.859	0.822	0.898	0.828	0.905	0.940
Forest lands	0.797	0.741	0.863	0.770	0.879	0.925
Road	0.814	0.765	0.870	0.786	0.889	0.931
Croplands	0.624	0.539	0.742	0.655	0.860	0.914
Mean	0.791	0.740	0.853	0.773	0.889	0.930
With batch normalisation						
Urban areas	0.825	0.782	0.874	0.796	0.897	0.944
Barren land	0.842	0.809	0.877	0.813	0.914	0.952
Agricultural land	0.862	0.827	0.899	0.831	0.918	0.952
Forest lands	0.803	0.750	0.863	0.775	0.884	0.938
Road	0.819	0.872	0.772	0.790	0.894	0.943
Croplands	0.639	0.558	0.747	0.665	0.837	0.927
Mean	0.798	0.751	0.838	0.778	0.890	0.942

Table 9. Accuracy of the model without and with batch normalisation for the second test area (Ipoh).

Unseen Class	F1-Score	Recall	Precision	Top-One	Top-Two	Top-Three
Without batch normalisation						
Green space	0.749	0.694	0.813	0.754	0.907	0.924
Urban areas	0.729	0.679	0.787	0.737	0.903	0.928
Barren land	0.709	0.653	0.776	0.721	0.900	0.928
Agricultural land	0.724	0.673	0.783	0.732	0.905	0.918
Forest lands	0.713	0.656	0.781	0.723	0.894	0.905
Road	0.728	0.675	0.790	0.737	0.903	0.922
Water body	0.705	0.648	0.773	0.716	0.897	0.917
Mean	0.722	0.668	0.786	0.731	0.901	0.920
With batch normalisation						
Green space	0.755	0.702	0.817	0.760	0.912	0.937
Urban areas	0.737	0.688	0.792	0.744	0.908	0.921
Barren land	0.717	0.664	0.779	0.728	0.905	0.922
Agricultural land	0.729	0.678	0.788	0.737	0.909	0.923
Forest lands	0.721	0.664	0.787	0.730	0.900	0.918
Road	0.735	0.684	0.794	0.743	0.907	0.920
Water body	0.712	0.655	0.778	0.722	0.901	0.929
Mean	0.729	0.676	0.790	0.737	0.906	0.924

4.3.2. Impact of the Number of Neurons

Additionally, to enrich the robustness of the network, a different number of neurons were examined over the first training and testing areas as well as the second test area. Table 10 represents the impact of diverse configurations of neurons with sizes of 128, 64, and 32.

The results from the experiment showed that the network with a filter size of 128 outperformed the other configurations of all datasets, including the training dataset (with F1-score of 0.832, recall of 0.780, the precision of 0.892, 0.833 top-one, 0.927 top-two and 0.976 top-three), first test dataset (with F1-score of 0.825, recall of 0.782, the precision of 0.874, 0.796 top-one, 0.897 top-two and 0.944 top-three) and second test data (with F1-score of 0.737, recall of 0.688, the precision of 0.792, 0.744 top-one, 0.908 top-two and 0.921 top-three), respectively.

Table 10. Impact of the number of neurons on the CNN-2 network.

Number of Neurons	Metric	Training Area	First Test Area	Second Test Area
128	Recall	0.780	0.782	0.688
	Precision	0.892	0.874	0.792
	F1	0.832	0.825	0.737
	Top-three	0.976	0.944	0.921
	Top-two	0.927	0.897	0.908
	Top-one	0.833	0.796	0.744
64	Recall	0.775	0.772	0.677
	Precision	0.892	0.870	0.784
	F1	0.829	0.818	0.727
	Top-three	0.975	0.941	0.927
	Top-two	0.924	0.891	0.872
	Top-one	0.829	0.790	0.735
32	Recall	0.766	0.763	0.661
	Precision	0.892	0.868	0.775
	F1	0.824	0.812	0.714
	Top-three	0.973	0.938	0.913
	Top-two	0.920	0.896	0.865
	Top-one	0.826	0.814	0.724

4.3.3. Impact of Gaussian Noise

The experiments of adding Gaussian noise to the training and test samples were conducted to investigate the sensitivity of the models to the noise in the samples. The parameters were set as (mu = 0, sigma = 0.1). Table 11 demonstrated that the network with the absence of additional noise outperformed the case with extra noise with (0.832 F1-score, 0.892 precision, 0.780 recall) for the training of the first dataset, (0.825 F1-score, 0.874 precision, 0.872 recall) for the first test and (0.737 F1-score, 0.792 precision, 0.688 recall) for the second test dataset. This outcome shows that the models trained with noisy samples also achieved significant results.

Table 11. Impact of Gaussian noise (mu = 0, sigma = 0.1).

	Metric	Training Area	First Test Area	Second Test Area
Without noise	Recall	0.780	0.782	0.688
	Precision	0.892	0.874	0.792
	F1	0.832	0.825	0.737
	Top-three	0.976	0.944	0.921
	Top-two	0.927	0.897	0.908
	Top-one	0.833	0.796	0.744
With noise	Recall	0.762	0.779	0.679
	Precision	0.873	0.882	0.785
	F1	0.810	0.811	0.719
	Top-three	0.957	0.929	0.901
	Top-two	0.917	0.875	0.886
	Top-one	0.820	0.787	0.723

4.4. Results of Transferability

To evaluate the transferability of the proposed framework, the entire process was applied to the second test dataset. The visual interpretation illustrated that the area consisted of diverse land cover types, including green space, urban areas, road, agricultural land, forest lands, barren land, and water body. The image was taken from a similar geographical environment to the first dataset; therefore, the same set of hyper-parameters was employed for CNNs. Figure 8 demonstrates the classification result for seven scenarios of unseen classes.

Table 12 demonstrates the results of the singular unseen class for each scenario and their mean accuracies. In each scenario, the unseen class did not exist in the training set. Overall, the average categorical accuracy of unseen classes showed a promising result by scoring 0.737 top-one, 0.906 top-two, 0.924 top-three, 0.729 F1-score, 0.676 recall and 0.790 precision, respectively. This can infer that the model excellently works for the second dataset as well.

Table 12. Retrieved labels for the unseen classes for the second test area (Ipoh).

Unseen Class	F1-Score	Recall	Precision	Top-One	Top-Two	Top-Three
Green space	0.755	0.702	0.817	0.760	0.912	0.937
Urban areas	0.737	0.688	0.792	0.744	0.908	0.921
Road	0.735	0.684	0.794	0.743	0.907	0.920
Agricultural land	0.729	0.678	0.788	0.737	0.909	0.923
Forest lands	0.721	0.664	0.787	0.730	0.900	0.918
Barren land	0.717	0.664	0.779	0.728	0.905	0.922
Water body	0.712	0.655	0.778	0.722	0.901	0.929
Average	0.729	0.676	0.790	0.737	0.906	0.924

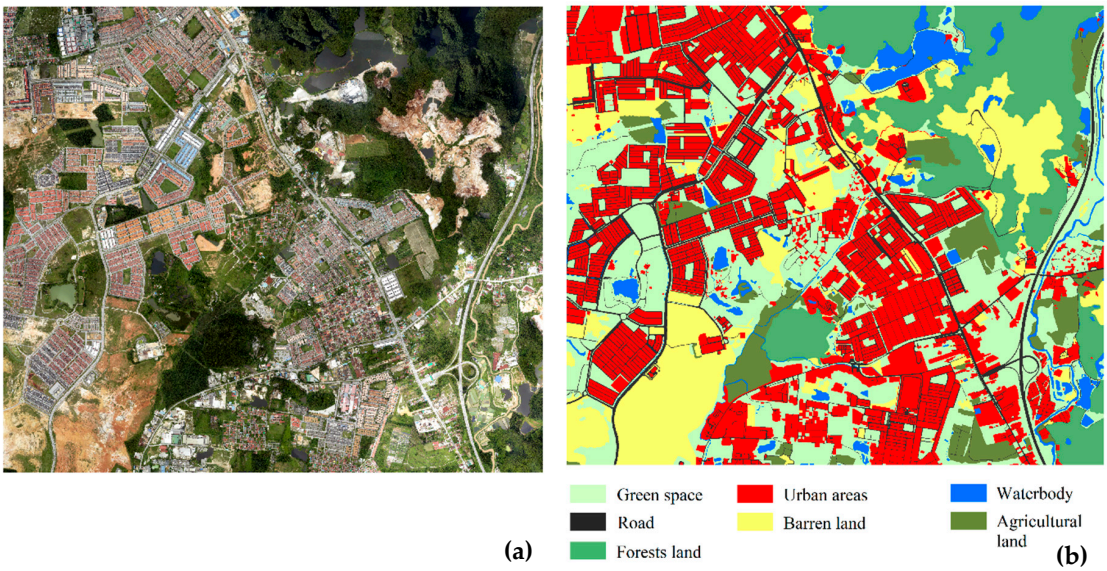

Figure 8. *Cont.*

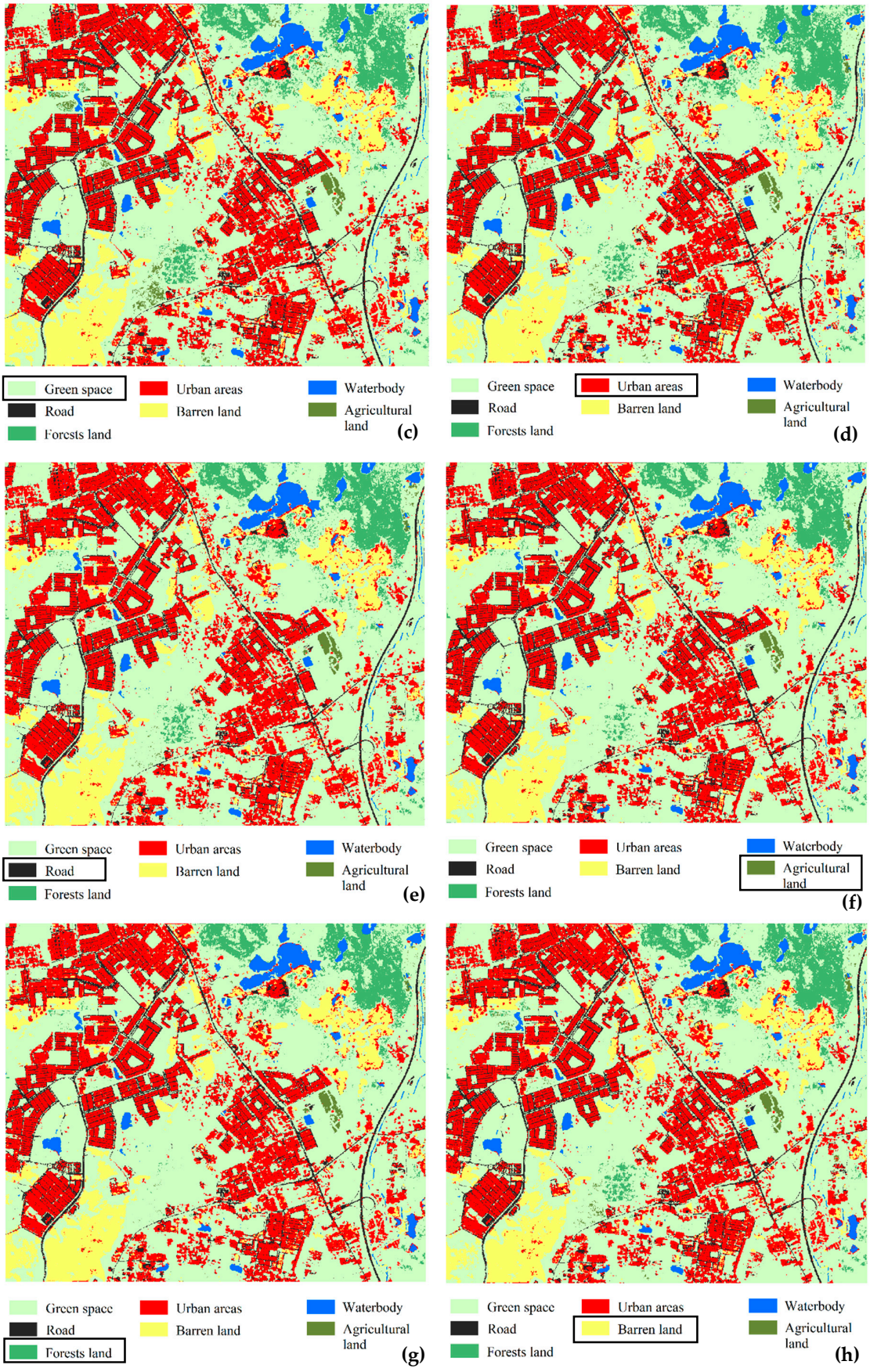

Figure 8. *Cont.*

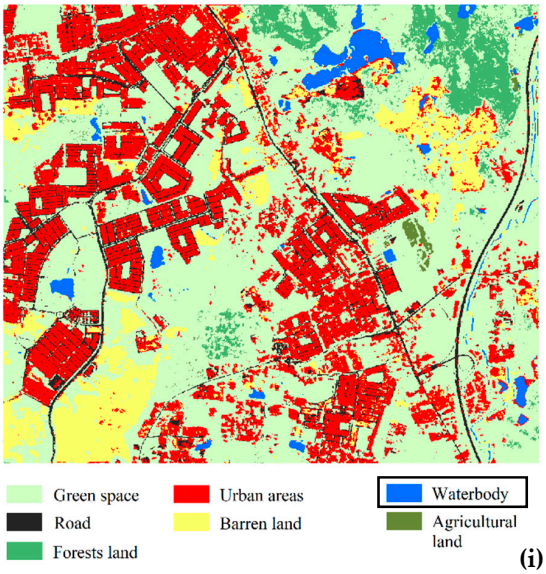

Figure 8. (**a**) Second test dataset (Ipoh area), (**b**) Ground truth map of the second test area, and classification results of the testing area with different unseen classes, (**c**) green space, (**d**) urban areas, (**e**) road, (**f**) agricultural land, (**g**) forest lands, (**h**) barren land, and (**i**) waterbody.

5. Discussion

Typical classification models in remote sensing can only classify objects that are seen during the training stage. These methods are unsuccessful in the classification of unseen objects in the testing stage. Unseen object classification is a challenging topic, in which plenty of studies have attempted to develop models to address this problem. For example, in recent years, ZSL has been widely implemented in computer science due to its potential to identify unseen objects without obtaining training samples by assistance from semantic information [14]. These models extract abstract features from the image pixels. From the class labels, the semantic information is often retrieved as vectors using models like Word2Vec. Unseen object classification is a hot topic, especially for the remote sensing field, due to its data variety and scalability [20,34]. To the best of our knowledge, these models are rarely applied to geoscience applications, especially for high-resolution land cover classification from aerial photos.

Therefore, this paper presented a ZSL framework based on CNN and Word2Vec for unseen land cover mapping. The CNN was found to be a robust feature extraction technique that achieved relatively high accuracies on training (0.953 F1-score, 0.941 precision, 0.882 recall), testing (0.904 F1-score, 0.869 precision, 0.949 recall) and second test dataset (0.898 F1-score, 0.870 precision, 0.838 recall). For the robustness of the network, various cases and models were tested. Despite including the Gaussian noise to input samples, no improvement was observed in the networks. This implies that the training strategies used in this experiment were good enough, and also, no over-fitting effects were observed. In the feature extraction phase, the best performance status was recorded, when the pooling layer was not included in the architecture. Therefore, we hypothesised that the dimensionality or complexity of the dataset was not an obstacle to train the network. Nevertheless, the case study including CNN with batch normalisation and pooling layers had comparable accuracy with the superior case (without pooling). Given that the feature extraction is a separate step in our ZSL, the proposed framework is flexible and can be further improved and customised for other applications. This process can be achieved by replacing the proposed CNN model with other deep-learning methods, depending on a new problem.

In the second phase, Word2Vec with 300 dimensions is used as word embedding to gather class attributes. Our ZSL approach could obtain (0.798 F1-score, 0.766 recall, 0.838 precision, 0.778 top-one, 0.890 top-two and 0.942 top-three) mean accuracies for different unseen classes on the first test area, accompanied by (0.729 F1-score, 0.676 recall, 0.790 precision, 0.737 top-one, 0.906 top-two and 0.924

top-three) mean accuracies for the second test area; however, a standard terminology or word model (exclusively for remote sensing domain) might further improve the results. Moreover, the obstacle of distance structure distinction between the word vectors and visual models of remote sensing image classification seriously impacts the operation and efficiency of the ZSL image classification [47]. Thus, special embedding attributes for remote sensing data could positively affect the model's performance.

In a previous generalised ZSL application on land cover classification with 8 m PolSAR images, the overall accuracy of 73% and the ratio of 1/3–3/3 unseen/seen classes were reported; however, the exact category missing with attributes in the SUN database of rural areas, wetland, and agricultural land was reported [24]. In another application utilising label propagation and label refinement approaches [14], the overall accuracy was reported as 58% and 70.4% for 5/16 and 1/7 unseen/seen ratio, respectively. In another work of street tree detection from areal images [21], CNN structures were employed and the accuracy of 14.3% for 16 tree classes was reported. While the overall accuracy evaluation metric was the main performance evaluation used in the previous remote sensing studies on zero-shot unseen land cover mapping, we further evaluated the robustness of the current framework considering the imbalanced class distribution via six evaluation metrics, namely F1-score, precision, recall and top-k categorical accuracy for k = [1,2,3] with 1/6 unsee/seen ratio. Table 13 presents a comparison among some related studies that have a similar scope with the current study in the remote sensing domain.

Table 13. Comparison of results from the previous studies and the current study.

ZSL Applications On Remote Sensing	Data	Approach	Accuracy	Unseen/Seen Ratio
Land cover classification (Current study)	Aerial images	2 CNNs-based structure	Average: F1-score: 0.798 Precision: 0.838 Recall: 0.766 Top-one: 0.778 Top-two:0.890 Top-three: 0.942	1/6
Land cover classification	PolSAR image with 8-meters resolution	Embedding space and latent embedding	Overall: 73%	1/3–3/3
Object detection	High-resolution satellite image	Label propagation and label refinement	Overall: 58.7% Overall: 70.4%	5/16 1/7
Street trees detection	Aerial images	CNN-based structure	Average: 14.3%	16/40

Although ZSL models achieve relatively high accuracy in computer vision, it is not fully explored in remote sensing applications. For this purpose, we need further specific word embedding models that can be trained on geospatial data. This case can introduce new research areas of ZSL and other fields requiring word embedding.

The most influential step in the current proposed ZSL framework is the feature extractor, which is based on CNN models. It has shown a promising result that could be extended to other geospatial applications. Accordingly, more robust CNN architectures like graphical CNNs, capsule-based CNNs, and others can help to improve the accuracy of the ZSL classification. A second model (CNN) is required to perform class signature prediction. This model can also be replaced with any other machine learning methods, such as SVMs or random forests. The class label is predicted by mapping the output vector to the nearest one and matching it with the vectors of all classes. Nevertheless, as part of the limitations that exist in CNN and deep learning models, they often require a large number of training samples for their learnings, and also their architectures contain enormous tuneable parameters to train the classifiers [36]. Another limitation is that the classifiers usually employed as black boxes. To solve these drawbacks, tensor-based learning models (tensor algebraic operations) [48], could provide promising solutions including a reduction in weight parameters (especially for high-dimensional/noisy data), allow physical interpretation and preserve the spatial and spectral coherency of input samples. This technique can generally be applied to CNN's structure by replacing fully-connected layers with tensor contractions [49].

In future works, several subjects should be additionally contemplated. First, a more efficient semantic data related to the land cover mapping in different remote sensing utilisation should be

investigated, including orthophotos, multispectral/hyperspectral images, synthetic aperture radar (SAR), and other remote sensing products. Second, the potential of different semantic models for land cover mapping assisted with diverse machine learning methods need to be further expanded. Third, there is no specific standard or agreed upon ZSL benchmark in the geospatial field [27,32]. Thus, the potential application of ZSL in a variety of popular applications such as change detection, land use/land cover mapping, detecting the types of landslides and more operations can be explored.

In this research, although some degrees of confusion are still existing among classes (e.g., croplands, forest lands and agricultural lands), the overall performance is still satisfactory. This confusion could be attributed to the likeness of the spectral and spatial properties of these classes. In such a case, using data with additional spectral bands (e.g., hyperspectral) could relatively improve the result. Besides, it is expected that establishing special standard embedding attributes for remote sensing data could decrease such confusion effects among classes.

Overall, the applicability of the adopted framework in the second study area showed that our ZSL scheme exhibited relatively similar classification results. Most of the unseen classes in different scenarios are excellently detected, given that such detection of unseen classes is impossible via traditional classification methods due to the absence of specific samples in training data. We performed various CNN models, techniques and sensitivity analysis (e.g., Gaussian noise additive) to our networks to assure its robustness using two different datasets. Both experiments on the datasets demonstrated their agreement to the framework's performance.

6. Conclusions

ZSL is an important concept that has been recently developed because of its potential to solve classification problems that lack sufficient training data for every class. In this paper, we present a ZSL framework for land cover mapping using orthophotos. The framework is built based on CNN and Word2Vec. The former is applied for the feature extraction process, whereas the latter is used to learn class attributes from the class labels.

The proposed models and the framework are tested on two subset datasets obtained for the Cameron Highlands as the first dataset and the Ipoh area as the second dataset, in Malaysia. The results show that the proposed feature extraction model achieves high accuracies on the training of the first dataset (0.953 F1-score, 0.941 precision, 0.882 recall), the first test dataset (0.904 F1-score, 0.869 precision, 0.949 recall) and 0.898 F1-score, 0.870 precision, 0.838 recall for the second test dataset. The ZSL model achieves accuracies of 0.778 top-one, 0.890 top-two and 0.942 top-three, 0.798 F1-score, 0.766 recall and 0.838 precision on average for different unseen classes on the test area of the first dataset and 0.737 top-one, 0.906 top-two, 0.924 top-three, 0.729 F1-score, 0.676 recall and 0.790 precision average accuracies for the second dataset. This outcome could help the experts in the remote sensing field, supporting them in recognising the correct class among two or three possible classes, especially when those classes are not included in the training set.

Transforming remote sensing imagery to a new embedding and using this strategy to predict seen and unseen classes could be a useful approach to ZSL in remote sensing data. Further developments will be considered, making the proposed framework more efficient for learning to predict unseen classes by using novel Word2Vec models specifically for remote sensing applications and various types of CNN models, including residual and graph CNNs. Moreover, the same number of samples for the seen and unseen classes will be considered for further assessment.

Author Contributions: B.P. collected field data; B.P., H.A.H.A.-N. and M.I.S. performed experiments; H.A.H.A.-N., and B.P. wrote the manuscript. B.P. supervised including the funding acquisition; B.P., I.T. and A.M.A. edited, restructured, and professionally optimised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: The research is funded by the Centre for Advanced Modelling and Geospatial Information Systems (CAMGIS), Faculty of Engineering and IT, the University of Technology Sydney under Blue Sky grant numbers: 323930, 321740.2232335; 321740.2232424 and 321740.2232357. This research was also partially supported by Researchers Supporting Project number RSP-2019/14, King Saud University, Riyadh, Saudi Arabia.

Acknowledgments: Thanks to the three anonymous reviewers for their valuable comments which helped us to revise the manuscript. Authors would like to thank the Faculty of Engineering and IT, the University of Technology Sydney for providing all facilities during this research.

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

References

1. Al-Najjar, H.A.H.; Kalantar, B.; Pradhan, B.; Saeidi, V.; Halin, A.A.; Ueda, N.; Mansor, S. Land cover classification from fused DSM and UAV images using convolutional neural networks. *Remote Sens.* **2019**, *11*, 1461. [CrossRef]
2. Novellino, A.; Jordan, C.; Ager, G.; Bateson, L.; Fleming, C.; Confuorto, P. *Geological Disaster Monitoring Based on Sensor Networks*; Springer: Singapore, 2019; ISBN 978-981-13-0991-5.
3. Pradhan, B.; Sezer, E.A.; Gokceoglu, C.; Buchroithner, M.F. Landslide susceptibility mapping by neuro-fuzzy approach in a landslide-prone area (Cameron Highlands, Malaysia). *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 4164–4177. [CrossRef]
4. Wang, C.; Myint, S.W.; Wang, Z.; Song, J. Spatio-temporal modeling of the urban heat island in the phoenix metropolitan area : Land use change implications. *Remote Sens.* **2016**, *8*, 185. [CrossRef]
5. Ashby, J.; Moreno-Madriñán, M.J.; Yiannoutsos, C.T.; Stanforth, A. Niche modeling of dengue fever using remotely sensed environmental factors and boosted regression trees. *Remote Sens.* **2017**, *9*, 328. [CrossRef]
6. Lehner, A.; Blaschke, T. A generic classification scheme for urban structure types. *Remote Sens.* **2019**, *11*, 173. [CrossRef]
7. Neuville, R.; Pouliot, J.; Poux, F. 3D Viewpoint management and navigation in urban planning: Application to the exploratory phase. *Remote Sens.* **2019**, *11*, 236. [CrossRef]
8. Cheng, J.; Kustas, W.P. Using very high resolution thermal infrared imagery for more accurate determination of the impact of land cover differences on evapotranspiration in an irrigated agricultural area. *Remote Sens.* **2019**, *11*, 613. [CrossRef]
9. Coops, N.C.; Waring, R.H.; Plowright, A.; Lee, J.; Dilts, T.E. Using remotely-sensed land cover and distribution modeling to estimate tree species migration in the Pacific Northwest Region of North America. *Remote Sens.* **2016**, *8*, 65. [CrossRef]
10. Tehrany, M.S.; Pradhan, B.; Jebuv, M.N. A comparative assessment between object and pixel-based classification approaches for land use/land cover mapping using SPOT 5 imagery. *Geocarto Int.* **2014**, *29*, 351–369. [CrossRef]
11. Rizeei, H.M.; Shafri, H.Z.M.; Mohamoud, M.A.; Pradhan, B.; Kalantar, B. Oil palm counting and age estimation from WorldView-3 imagery and LiDAR data using an integrated OBIA height model and regression analysis. *J. Sens.* **2018**, *2018*, 2536327. [CrossRef]
12. Huete, A.R.; Miura, T.; Gao, X. Land cover conversion and degradation analyses through coupled soil-plant biophysical parameters derived from hyperspectral EO-1 Hyperion. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1268–1276. [CrossRef]
13. Quan, J.; Wu, C.; Wang, H.; Wang, Z. Structural alignment based zero-shot classification for remote sensing scenes. In Proceedings of the 2018 IEEE International Conference on Electronics and Communication Engineering, ICECE 2018, Xi'an, China, 10–12 December 2018; pp. 17–21.
14. Li, A.; Lu, Z.; Wang, L.; Xiang, T.; Wen, J.R. Zero-shot scene classification for high spatial resolution remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4157–4167. [CrossRef]
15. Wang, W.; Zheng, V.W.; Yu, H.; Miao, C. A survey of zero-shot learning: Settings, methods, and applications. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 1–37. [CrossRef]
16. Zhang, Z.; Saligrama, V. Zero-shot learning via semantic similarity embedding. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4166–4174.
17. Gong, P.; Wang, X.; Cheng, Y.; Fellow, Z.J.W. Zero-shot classification based on multi-task mixed attribute relations and attribute-specific features. *IEEE Trans. Cogn. Dev. Syst.* **2019**, *12*, 1-1.
18. Gaurav, R.; Srivastava, B. Estimating train delays in a large rail network using a zero shot markov model. In Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 1221–1226.

19. Gorbatsevich, V.; Vizilter, Y.; Knyaz, V.; Moiseenko, A. Single-shot semantic matcher for unseen object detection. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. ISPRS Arch.* **2018**, *42*, 379–384. [CrossRef]
20. Zhang, X.; Han, L.; Han, L.; Zhu, L. How Well Do Deep Learning-Based Methods for Land Cover Classification and Object Detection Perform on High Resolution Remote Sensing Imagery? *Remote Sensing* **2020**, *12*, 417. [CrossRef]
21. Sumbul, G.; Cinbis, R.G.; Aksoy, S. Multisource region attention network for fine-grained object recognition in remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4929–4937. [CrossRef]
22. Chen, H.; Luo, Y.; Cao, L.; Zhang, B.; Guo, G.; Wang, C.; Li, J.; Ji, R. Generalized zero-shot vehicle detection in remote sensing imagery via coarse-to-fine framework. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 687–693.
23. Jia, X.; Khandelwal, A.; Nayak, G.; Gerber, J.; Carlson, K.; West, P.; Kumar, V. Incremental dual-memory LSTM in land cover prediction. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; Part F1296. pp. 867–876.
24. Gui, R.; Xu, X.; Wang, L.; Yang, R.; Pu, F. A generalized zero-shot learning framework for PolSAR land cover classification. *Remote Sens.* **2018**, *10*, 1307. [CrossRef]
25. Mikolov, T.; Ilya, S.; Kai, C.; Greg, C.; Jeffrey, D. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, 1389–1399.
26. Lampert, C.H.; Nickisch, H.; Harmeling, S. Attribute-based classification for zero-shot visual object categorizationa. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 453–465. [CrossRef]
27. Antol, S.; Zitnick, C.L.; Parikh, D. Zero-shot learning via visual abstraction. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; ISBN 9783319105925.
28. Mensink, T.; Gavves, E.; Snoek, C.G.M. COSTA: Co-occurrence statistics for zero-shot classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2441–2448.
29. Lu, N.; Sun, Y.; Yun, X. Hybrid Relative attributes based on sparse coding for zero-shot image classification. *Math. Probl. Eng.* **2019**, *2019*, 7390327. [CrossRef]
30. Rusk, N. *Computer Vision—ECCV 2016. Part II*; Springer: Cham, Switzerland, 2016; Volume 9905, ISBN 978-3-319-46447-3.
31. Xian, Y.; Lampert, C.H.; Schiele, B.; Akata, Z. Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 2251–2265. [CrossRef] [PubMed]
32. Fu, Y.; Sigal, L. Semi-supervised vocabulary-informed learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5337–5346.
33. Li, W.; Guo, Q. A new accuracy assessment method for one-class remote sensing classification. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 4621–4632.
34. Liu, P. A survey of remote-sensing big data. *Front. Environ. Sci.* **2015**, *3*, 1–6. [CrossRef]
35. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
36. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
37. 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]
38. Xian, Y.; Akata, Z.; Sharma, G.; Nguyen, Q.; Hein, M.; Schiele, B. Latent embeddings for zero-shot classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 69–77.
39. Wang, Q.; Chen, K. Zero-shot visual recognition via bidirectional latent embedding. *Int. J. Comput. Vis.* **2017**, *124*, 356–383. [CrossRef]
40. Shi, Y.; Xu, D.; Pan, Y.; Tsang, I.W.; Pan, S. Label embedding with partial heterogeneous contexts. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-19), Honolulu, HI, USA, 27 January–1 February 2019; pp. 4926–4933.
41. Xu, D.; Shi, Y.; Tsang, I.W.; Ong, Y.; Gong, C.; Shen, X. A survey on multi-output learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, 1–21. [CrossRef]
42. Sumbul, G.; Cinbis, R.G.; Aksoy, S. Fine-grained object recognition and zero-shot learning in remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 770–779. [CrossRef]

43. Zerrouki, N.; Bouchaffra, D. Pixel-based or object-based: Which approach is more appropriate for remote sensing image classification? In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA, USA, 5–8 October 2014; pp. 864–869.
44. Hagiwara, K.; Hayasaka, T.; Toda, N.; Usui, S.; Kuno, K. Upper bound of the expected training error of neural network regression for a Gaussian noise sequence. *Neural Networks* **2001**, *14*, 1419–1429. [CrossRef]
45. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [CrossRef] [PubMed]
46. Gu, S.; Rigazio, L. Towards deep neural network architectures robust to adversarial examples. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015; pp. 1–9.
47. Chen, W.; Guang, Y.; Fengjing, Z.; Liu, Y.; Yuan, Y.; Jicheng, Q. Zero-Shot Classification Method for Remote-Sensing Scenes Based on Word Vector Consistent Fusion. *Acta Optica Sinica* **2019**, *39*, 0828002. [CrossRef]
48. Makantasis, K.; Doulamis, A.D.; Doulamis, N.D.; Nikitakis, A. Tensor-based classification models for hyperspectral data analysis. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6884–6898. [CrossRef]
49. Kossaifi, J.; Lipton, Z.C.; Khanna, A.; Furlanello, T.; Anandkumar, A. Tensor Regression Networks. *arXiv* **2017**, arXiv:1707.08308.