remote sensing

MDPI

Article

Hyperspectral Image Classification Based on Class-Incremental Learning with Knowledge Distillation

Meng Xu [1], Yuanyuan Zhao [2], Yajun Liang [3] and Xiaorui Ma [2,*]

[1] China Academy of Space Technology, Beijing 100098, China; xumeng@cast.cn
[2] School of Information Science and Technology, Dalian University of Technology, Dalian 116024, China; zhaoyy1020@mail.dlut.edu.cn
[3] Space Star Technology Co., Ltd., Chengdu 610100, China; liangyj@spacestar.com.cn
* Correspondence: maxr@dlut.edu.cn

Abstract: By virtue of its large-covered spatial information and high-resolution spectral information, hyperspectral images make lots of mapping-based fine-grained remote sensing applications possible. However, due to the inconsistency of land-cover types between different images, most hyperspectral image classification methods keep their effectiveness by training on every image and saving all classification models and training samples, which limits the promotion of related remote sensing tasks. To deal with the aforementioned issues, this paper proposes a hyperspectral image classification method based on class-incremental learning to learn new land-cover types without forgetting the old ones, which enables the classification method to classify all land-cover types with one final model. Specially, when learning new classes, a knowledge distillation strategy is designed to recall the information of old classes by transferring knowledge to the newly trained network, and a linear correction layer is proposed to relax the heavy bias towards the new class by reapportioning information between different classes. Additionally, the proposed method introduces a channel attention mechanism to effectively utilize spatial–spectral information by a recalibration strategy. Experimental results on the three widely used hyperspectral images demonstrate that the proposed method can identify both new and old land-cover types with high accuracy, which proves the proposed method is more practical in large-coverage remote sensing tasks.

Keywords: hyperspectral image; classification; class-incremental learning

Citation: Xu, M.; Zhao, Y.; Liang, Y.; Ma, X. Hyperspectral Image Classification Based on Class-Incremental Learning with Knowledge Distillation. *Remote Sens.* **2022**, *14*, 2556. https://doi.org/10.3390/rs14112556

Academic Editor: Naoto Yokoya

Received: 11 April 2022
Accepted: 24 May 2022
Published: 26 May 2022

1. Introduction

Hyperspectral image (HSI), which contains massive amounts of spectral and spatial information, makes lots of fine-grained remote sensing tasks possible [1]. HSI classification assigns each pixel of the image to a specific land-cover category according to its spectral and spatial characteristics, which is crucial for lots of earth observation applications, such as agricultural detection, ocean exploration, military defense, and land-cover mapping [2–4]. Therefore, HSI classification has attracted lots of research attention, and has become a hot topic in the field of remote sensing [5].

Up to now, lots of methods have been proposed for HSI classification. Traditional machine learning methods are firstly applied in HSI classification. Compared with visual interpretation, they can give automatic solutions. In the beginning, spectral classification methods, which regard HSI as a continuous sequence shown in the form of a two-dimensional image [6], are widely developed, support vector-machine-related methods [7,8] are representative classifiers of this period. Because of the Hughes phenomenon [9], band selection [10] and dimensionality reduction techniques have also been widely used in HSI classification, including principal component analysis [11], functional data analysis [12], and linear discriminant analysis [13]. For the improvement of HSI spatial resolution, spectral–spatial classification methods are introduced soon afterward to improve the classification performance with spatial information. Fauvel et al. [14] proposed a method

to include both spatial and spectral information in the classification process by a data fusion scheme; Patra et al. [15] utilized spectral–spatial features by extended morphological profiles, which have further increased the classification accuracy with spatial information. HSI classification methods based on traditional machine learning prove the feasibility of hyperspectral images in the land-cover investigation, and start a new era of remote sensing applications with sufficient spectral and spatial information.

In recent years, inspired by the superiority of deep learning in traditional computer vision tasks, a large number of HSI classification methods based on deep networks have been proposed and greatly improved the classification performance. Chen et al. [16] first introduced the concept of deep learning into HSI classification, which proposed a new learning framework to merge stacked auto-encoders and spatial-dominated information to get higher accuracy. Recently, Cui et al. [17] proposed a novel classification method based on super-pixel and multi-classifier fusion; Sk et al. [18] proposed a new convolutional neural network to use different derivatives to train different layers; and Li [19] proposed a deep network with channel and position global context attention to capture discriminative features. Their research has further improved the classification performance and represents the development trend of HSI classification. However, there are some issues that need to be considered. The hyperspectral imagers obtain images continuously over time, and the different land-cover types are found sequentially. The traditional classification model can only obtain a good classification performance on predefined classes; otherwise, there will be catastrophic forgetting [20], i.e., the model quickly updates to ensure adaptation to new data streams, but rapidly forgets old knowledge, which can be seen in Figure 1 (top). When learning a new classification task, the previous decision boundary could be changed significantly, resulting in catastrophic forgetting. However, traditional methods rely on saving all training samples and training the network from scratch to solve the catastrophic forgetting problem, which is not feasible in large-coverage remote sensing applications. Therefore, when faced with different land-cover types, the classification method should learn the knowledge of new land-cover types without forgetting the old ones, and classify all land-cover types with one final classification model.

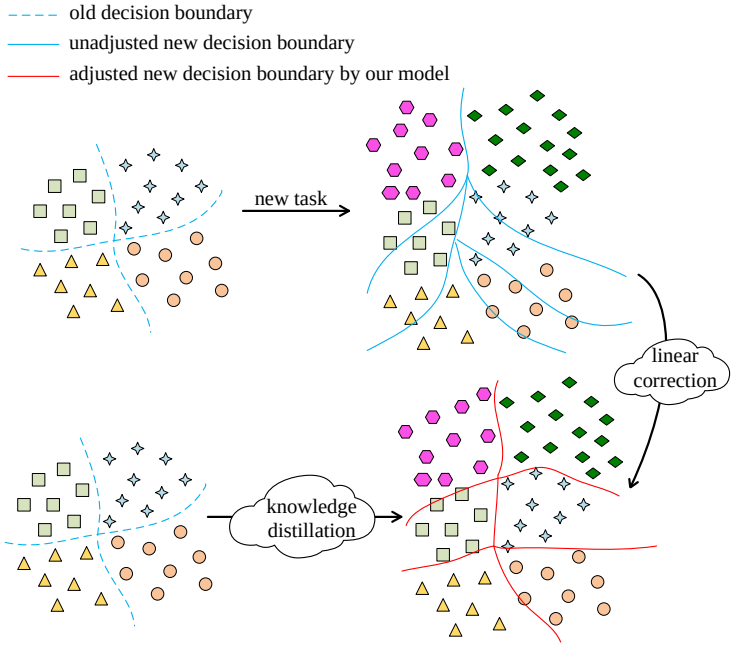


Figure 1. Illustration comparing the traditional learning approach and the proposed scheme of our model: Traditional classification methods suffer from catastrophic forgetting; the old decision boundary will change significantly (**top**). The proposed model utilizes knowledge distillation to transfer old knowledge to the new network and utilizes linear correction unit to adjust and correct the decision boundary (**bottom**).

In order to address the aforementioned issue, this paper studies a class-incremental learning method for hyperspectral image classification. In order to maintain the classification performance for the old ones, the proposed method introduces knowledge distillation into hyperspectral image classification, which is able to learn incremental knowledge of new classes without forgetting the old ones [21]. Moreover, since the imbalanced number of samples in old and new classes makes the classifier biased towards new classes and results in terrible classification performance, the proposed method designs a linear correction layer to balance between different classes and relax the heavy bias toward new classes (shown in Figure 1). Furthermore, the proposed method utilizes a channel attention structure with a feature recalibration strategy to calibrate the importance of each feature by assigning different weights to each channel. With the aforementioned strategies, the proposed model is able to correctly recognize the new classes while maintaining the ability to recognize the old ones, and realizes class-incremental learning for HSI classification. In summary, this paper proposes an incremental learning method for HSI classification. The proposed method learns new classes with a channel attention network, recalls the old classes with a knowledge distillation strategy, and recalibrates all classes with a linear correction layer. All these modules work together to classify all classes with one final model.

The rest of this paper is organized into four sections. Section 2 introduces the related work on incremental learning and HSI classification based on deep learning in recent years; Section 3 systematically presents the principle of the proposed class-incremental learning for HSI classification; Section 4 is the experimental description and results; Section 5 is the further discussion of the effectiveness of the model; and Section 6 is the conclusion and future work.

2. Related Works

2.1. HSI Classification

In recent years, most research on HSI classification has relied on deep learning. Compared with traditional machine learning based methods, deep learning is able to extract more abstract and representative features with deep and hierarchical structures. Many deep networks have been introduced and improved for HSI classification, such as Convolutional Neural Networks (CNN) [22–27], Stacked Auto-Encoder (SAE) [16,28–31], and Deep Belief Network (DBN) [32–36].

CNN is currently the most widely used deep model for spatial feature learning in HSI classification. Due to the complexity of computing, dimensionality reduction methods are usually used before spatial feature extraction. Chen et al. [25] provided the idea of automatic CNN, which is able to select the best CNN architecture by using neural architecture search techniques for the HSI classification. Wang et al. [26] proposed a 3D–2D CNN model to solve the problem of few-shot classification, which is able to make better use of hierarchical spectral–spatial features with dense blocks. Zhang et al. [27] proposed a spectral—spatial fractal residual CNN with data-balance augmentation to solve the problems of limited labeled data and imbalanced categories.

SAE is built by multiple unsupervised pretrained auto-encoders. Some methods utilize SAE for spectral feature extraction or feature optimization. Tao et al. [30] compressed HSI to one dimension using principal component analysis, and then used SAE for feature extraction. Liu et al. [37] and Chen et al. [38] stacked denoised auto-encoders to extract spectral features and perform classification based on the deep spectral features. In order to better extract spectral and spatial information, Zhao et al. [31] proposed a combination network for HSI classification based on a stacked auto-encoder and 3D deep residual network.

DBN is also a stacked network model with Boltzmann machine. Chen et al. [32] introduced sparse restriction and Gaussian noise in DBN for HSI classification. To deal with limited number of samples of HSI, Zhong et al. [33] proposed unsupervised pretraining over unlabeled samples by DBN and then a supervised fine-tuning over labeled samples, which improves the efficiency of classification. Chen et al. [36] proposed deep belief networks based on the conjugate gradient update algorithm, which includes two processes: unsupervised pretraining and supervised fine-tuning.

The aforementioned deep-learning-based HSI classification methods improved the deep architectures according to the specialty of HSI, and proved the advance of deep networks in HSI classification. Therefore, this paper studies the incremental learning method based on the deep network for HSI classification and designs a deep architecture to classify all land-cover types with one final deep network.

2.2. Class-Incremental Learning

Research related to incremental learning includes lifelong learning and continuous learning, which can be divided into the following three main categories: regularization-based methods [39–42], architecture-based methods [43–46], and replay-based methods [21,47–49].

Regularization-based methods try to protect the old knowledge from being overwritten by imposing constraints on the loss function of the new task. This type of method usually does not need old data to allow the model to review the learned task. Li et al. [40] proposed a typical regularization-based method, i.e., Learning without Forgetting (LwF), which makes the prediction of the new model on the new task similar to the prediction of the old model on the new task. However, the disadvantage of this method is that it highly depends on the correlation between the new and old tasks. Rannen et al. [41] proposed an Encoder-Based Lifelong Learning algorithm based on low-dimensional feature mapping to improve the strategy for LwF. Recently, Zhu et al. [42] employed self-supervised learning to learn more generalizable and transferable features for other tasks and adopted prototype augmentation to maintain the decision boundary of old tasks.

Architecture-based methods generate a sub-network for each classification task to prevent catastrophic forgetting from modifying network parameters using network pruning, dynamic expansion, or parameter shielding. Piggyback [44] learns binary masks for each task in an end-to-end differentiable fashion by building upon ideas from network quantization and pruning. Serra et al. [45] proposed a task-based hard attention mechanism, which learns the attention masks for old tasks through stochastic gradient descent and uses them to constrain the parameters when learning the new task. Achituve et al. [46] developed a tree-based hierarchical model in which each internal node of the tree fits a Gaussian process classifier to the data. They proposed building a separate tree for the new and old classes and connecting them with a shared node of root.

Replay-based methods retain part of the representative old data for the model to review the old knowledge. Therefore, the main issue is which part of the old data should be retained and how to use all data to train the model. There are two strategies; one is storing a small number of old training samples, and the other is using a generation mechanism to generate pseudo data. The model proposed in [48] is the most classic incremental learning model based on replay. It stores some old data and introduces distillation loss to update the model parameters, which improves the incremental learning ability of the deep network. Similar algorithms include [21]; these algorithms alleviate the catastrophic forgetting problem from different angles, and the loss function adopts knowledge distillation loss. Liu et al. [49] proposed an Adaptive Aggregation Network by building two types of residual blocks at each residual level to resolve the dilemma between plasticity and stability.

Although these works have made great progress in incremental learning, there are few studies on incremental learning beyond the image level, i.e., pixel level.

A few works in this direction include [50–54]. Yan et al. [50] proposed a unified learning strategy based on the Expectation-Maximization framework, which integrates an iterative relabeling strategy to balance the stability–plasticity dilemma. Tasar et al. [54] proposed an approach to adapt the network to learn new as well as old classes on the new training data for semantic segmentation of large-scale remote sensing data. Fabio et al. [51] first proposed the semantic shift issue of the background class in incremental learning for semantic segmentation, and introduced a new distillation-based framework and a

specific classifier initialization strategy to explicitly cope with the evolving semantics of the background class, which greatly alleviates the catastrophic forgetting.

Despite this progress, none of these studies involved hyperspectral images. Bai et al. [55] proposed an incremental learning method for HSI classification based on a linear programming incremental learning classifier (LPILC), which enables the model to quickly recognize new classes within a few samples. Unlike LPILC, which has only one incremental learning phase and one new class, the model we proposed contains more incremental phases and more new classes. Since storing a few old class exemplars, our model belongs to the third category, i.e., replay-based method of incremental learning.

3. Methods

In this section, we present the proposed class-incremental learning method for HSI classification. The framework of the proposed method is described in Section 3.1, which consists of four key parts, i.e., feature extraction with channel attention, exemplars management, knowledge distillation, and linear correction unit; all of them are introduced in the following Sections 3.2–3.5.

3.1. Framework

At the beginning, we give a brief introduction of some key symbols used in this paper. Suppose that there are $L+1$ learning phases, i.e., 1 initial phase and L incremental phases. Usually, most of the classes are in the initial phase, and we can set different values for L. The corresponding settings are called L-phase incremental learning. We consider that incremental learning is built on data set $\mathcal{D} = \{D_l\}_{l=0}^{L}$, where l is the l-th learning phase. The set D_0 is used to train the base model in the initial phase, and $D_l (1 \leq l \leq L)$ is used to train the incremental model in the l-th learning phases. In the l-th learning phase, suppose there are N classes in total, M of them are old classes, and the rest are new classes. The selected exemplars from the old class are denoted as $D_l^0 = \{(x_i, y_i)\}_{i=0}^{P}$, where $y_i \in \{1, ..., M\}$, P is the number of selected old exemplars. The samples from new classes are denoted as $D_l^1 = \{(x_j, y_j)\}_{j=0}^{Q}$, where $y_j \in \{N-M+1, ..., N\}$, Q is the number of new class training samples, and it satisfies $(\mathrm{P}/M << Q/(N-M))$. Our goal is to train a model to correctly distinguish all the N classes.

The framework of the proposed method is shown in Figure 2, which contains four basic structures, i.e., a feature extraction module, a classifier, knowledge distillation, and a linear correction unit. The feature extraction is a deep network with channel attention, which is initialized by multiple convolution and pooling layers. The classifier is built by a fully connected (FC) layer. The feature extraction module and classifier are the basic functional modules of HSI classification. We call the network trained on D_0 the old network; otherwise, it is the new network. D_0 is used to train feature extraction and the classifier module to give them good feature-extraction capabilities. In the incremental phase, the soft labels of the D_0 are transferred to the new classification network through knowledge distillation. Then, the true labels of old class exemplars $D_l^0 (1 \leq l \leq L)$ and new class samples $D_l^1 (1 \leq l \leq L)$ are combined with soft labels to jointly train the new network. The linear correction unit is used to correct the bias of the FC layer. We will introduce the specific functions of each unit below.

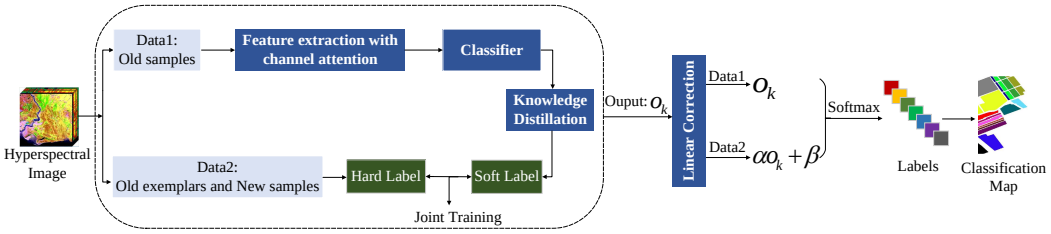


Figure 2. The framework of class-incremental learning for hyperspectral image classification, which contains four basic structures: feature extraction module, classifier module, knowledge distillation, and linear correction unit.

3.2. Channel Attention

HSI usually contains dozens to hundreds of spectral channels, which are rich in spectral information. The traditional deep network uses the spectral value of each channel directly for classification, which means that each channel has an equal contribution to the classification. However, the spectral value of each channel of HSI is different, and the contribution to the classification is also different. Inspired by [56], we introduce the channel attention mechanism into the classification network, which pays attention to the relationship between channels, and is able to automatically learn the importance of different channel features.

The structure diagram of the channel attention module is shown in Figure 3, which contains two basic operations, i.e., squeeze and excitation. Suppose that we now have extracted features with the number of channels C of HSI through the deep network. First, we compress the spatial dimension through global average pooling. Each two-dimensional channel feature will become a scalar, with a total of C. Next, the feature of each channel is recalibrated through excitation, which is achieved through the RELU function. In this way, we get C scalars between 0 and 1 as the weight of each channel, which are used to multiply the corresponding channel to obtain the weighted feature, i.e., feature recalibration. In this paper, the channel attention module is embedded to extract features.

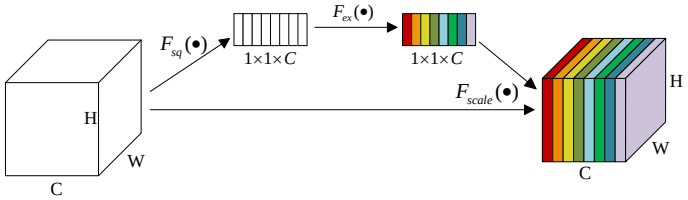


Figure 3. The structure diagram of channel attention module.

3.3. Exemplars Management

Our model has strict control over the memory budget of the old classes, which adjusts the set of old exemplars when adding new classes. No matter how many samples each old class contains, they are treated equally, i.e., P is the total number of old class exemplars stored, and the model encountered M old classes up to now, so each old class has P/M exemplars. In this way, our model can ensure that the memory budget can be fully utilized but not exceeded.

When new classes are added as incremental learning progresses, two processes are required to assist in completing the management of the old class exemplars. One is to reduce the number of exemplars of the original old class set, and the other is to select exemplars for the new arrivals of old classes. We refer to the example management method in [48] and select the old class exemplar which is closest to the class average vector, i.e., when an old class exemplar is added, the average feature vector over all exemplars should be the best approximate of the average feature vector over all training examples, until the number P/M is satisfied. For the original old classes, we delete the exemplars which are far away from the average feature vector until there are only P/M exemplars in each class.

3.4. Knowledge Distillation

Knowledge distillation is able to maintain the classification performance of old classes by training the network with old knowledge, which introduces soft targets related to the old network as part of the total loss, and saves the network features of the old classes to avoid catastrophic forgetting.

Suppose the output of the old and new classifiers are $\hat{o}(x)$ and $o(x)$ respectively. In the l-th learning phase, the distilling loss, which is illustrated in Figure 4, is as follows:

$$L_d = -\sum_{(x,y)\in D_l^0}\sum_{k=1}^{M} q_k(x)\log p_k(x),$$

$$q_k(x) = \frac{e^{\hat{o}_k(x)/T}}{\sum_{m=1}^{M} e^{\hat{o}_m(x)/T}}, p_k(x) = \frac{e^{o_k(x)/T}}{\sum_{m=1}^{M} e^{o_m(x)/T}}, \tag{1}$$

where q_k is the soft label of the old model, which functions as a pseudo label, and p_k is the output probability of the new model. T is the temperature scalar, when $T = 1$, it is the ordinary softmax transform. If $T > 1$, a softened softmax is obtained. The distilling loss is computed by the exemplars from the old classes and its purpose is to transfer the features of the old network to the new.

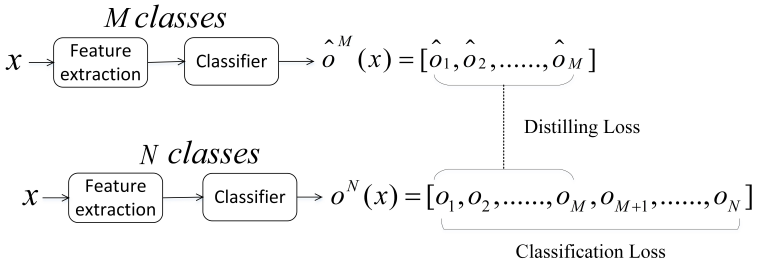


Figure 4. The diagram of distillation loss and classification loss. The distillation loss is used on old classes, while the classification loss is used on all old and new classes.

The classification loss is calculated using the cross-entropy loss function, which is expressed as follows:

$$L_c = -\sum_{(x,y)\in D_l}\sum_{k=1}^{N} \delta_{y=k}\log p_k(x), \tag{2}$$

where $\delta_{y=k}$ is the indicator variable (0 or 1) and $p_k(x)$ is the predicted label of the input x. L_c is computed by all old and new data.

We combine L_d and L_c and assign a weight coefficient to them as the total loss(L), which is illustrated as follows:

$$L = \eta L_d + (1-\eta)L_c. \tag{3}$$

where the η is a hyperparameter we need to train. We will discuss its value in the accompanying experiments.

3.5. Linear Correction

Using knowledge distillation, the information of the old classes can be transferred to the new network. However, we find that the classification results are still not ideal, especially when the spectral curves of the new class and the old class are similar. Inspired from [21], we find that the suboptimal classification results were caused by the bias of the classifier module towards new classes, which is because the number of training samples of the new class is much larger than the exemplars of the old class, i.e., $(P/M << Q/(N-M))$, where the old classes are overshadowed by the new classes.

To deal with the aforementioned problem, we design a linear correction unit (LC) after the classification layer to relax the bias. The balanced subset is sampled from both the old and the new classes, which are very small, and used to train the LC unit. Our strategy consists of two steps: first, we use the baseline model train the whole network except the LC unit. Second, we freeze the feature extraction module and the classifier module, and use the balanced subset train the LC unit to relax the bias towards the old classes.

The balanced set has no intersection with the training set. Composed of equal amounts of new and old data, it can better represent the unbiased distribution of old and new data. Our data distribution strategy is as follows: The saved old class exemplars are divided into two parts $D_l^0 = D_l^{(0,0)} \cup D_l^{(0,1)}$. Similarly, the new class training samples are also divided into two parts $D_l^1 = D_l^{(1,0)} \cup D_l^{(1,1)}$. $D_l^{(0,0)}$ and $D_l^{(1,0)}$ are used as the basic training set to train the feature extraction module and the classifier module. $D_l^{(0,1)}$ and $D_l^{(1,1)}$ are used as a balanced set to train the LC unit. The data distribution method is shown in Figure 5.

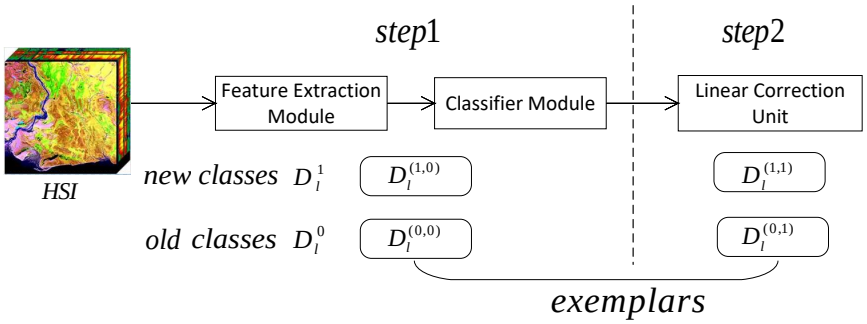


Figure 5. The diagram of data distribution method: The training samples of the new classes and exemplars of the old classes are divided into training and balanced sets. The training set is used for the feature extraction module and classifier module, and the balanced set for linear correction unit.

The LC is a linear module with two parameters, i.e., α and β, and it follows the classifier module and corrects the output o_k of the FC layer. The expression is as follows:

$$q_k = \begin{cases} o_k & 0 \le k \le M \\ \alpha o_k + \beta & N - M + 1 \le k \le N, \end{cases} \tag{4}$$

where α and β are the parameters we need to train and update continuously, o_k is the output of the FC layer, and q_k is the output after the correction of the LC unit. For M old classes, there is no need to set a linear correction, because there are no new data and the FC layer is only biased for the new, so the LC is only added when the $N - M$ new classes enter the network.

Our training process is summarized in Algorithm 1. We have three main stages: feature extraction (line 3), knowledge distillation (line 5), and linear correction (line 6–7).

Algorithm 1 Class-Incremental Learning for HSI

Input:
$D_i \in D_0, D_1, ..., D_L$ // HSI data set
L // learning phase
P // the total number of old exemplars
Output:
OA, AA, κ // classification results of each phase

```
for samples in D_i do
    if D_i = D_0 then
        Update the network parameters by Equation (2);
    else
        Update old exemplars;
        Update the network parameters by Equation (3);
        Train the LC unit by Equation (4);
    end if
end for
```

4. Results

In this part, we conducted extensive experiments to verify the effectiveness of our proposed model, with three widely used HSI data sets. All experiments are implemented with PyTorch on the platform of a desktop computer with Intel Core i7 4.0-GHz CPU, NVIDIA GeForce RTX 2080Ti GPU, and 32 GB memory.

4.1. Data Description

In this section, three hyperspectral data sets with different settings are used to test the effectiveness of the proposed model. They are captured on different sites, i.e., University of Pavia, Salinas, and Houston.

University of Pavia (PaviaU) data set was acquired by the ROSIS sensor during a flight campaign over the University of Pavia, northern Italy. The PaviaU data set has 103 spectral channels covering from 430 to 860 nm after excluding the band affected by noise, and has a spatial resolution of 1.3 m. The spatial size is 610 $\times$ 340 pixels, which contains a total of 42,776 labeled samples in 9 classes. The pseudocolor images with the corresponding ground-truth map are shown in Figure 6.

Salinas data set was taken by the AVIRIS sensor in the Salinas Valley, California, USA. The original image has 224 spectral channels. After discarding 20 water-absorption bands, there are 204 dimensional spectral channels covering from 400 to 2500 nm. This hyperspectral image has a spatial resolution of 3.7 m and a spectral resolution of 10 nm. The spatial size is 512 $\times$ 217 pixels, which contains 54,129 samples in 16 classes related to vegetation that can be used for specific experimental classification. The pseudocolor images with the corresponding ground-truth map are shown in Figure 7.

Houston data set was acquired by the CASI sensor over the University of Houston campus and its neighboring areas. Houston data set has 144 spectral channels covering from 380 to 1050 nm geometric resolution 2.5 m, and the spatial size is 349 $\times$ 1905 pixels. It contains a total of 15,029 labeled samples in 15 classes, and the pseudocolor image with the corresponding ground-truth map are shown in Figure 8.

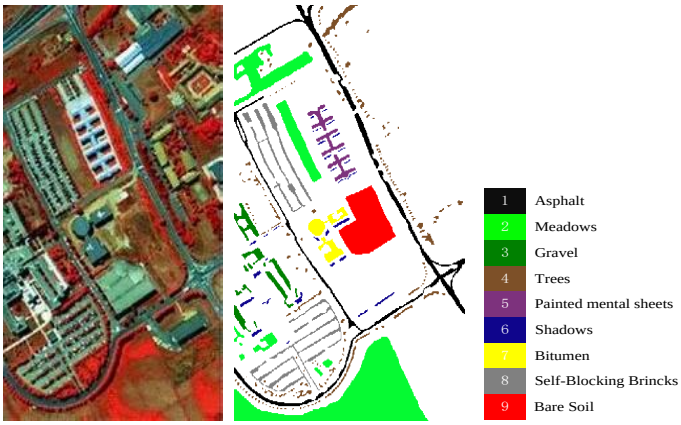


Figure 6. Ground truth map, pseudocolor images, and available samples of University of Pavia scene.

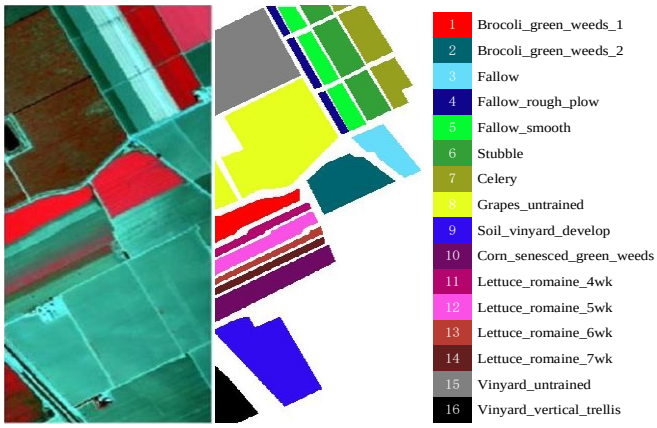


Figure 7. Ground truth map, pseudocolor images, and available samples of Salinas scene.

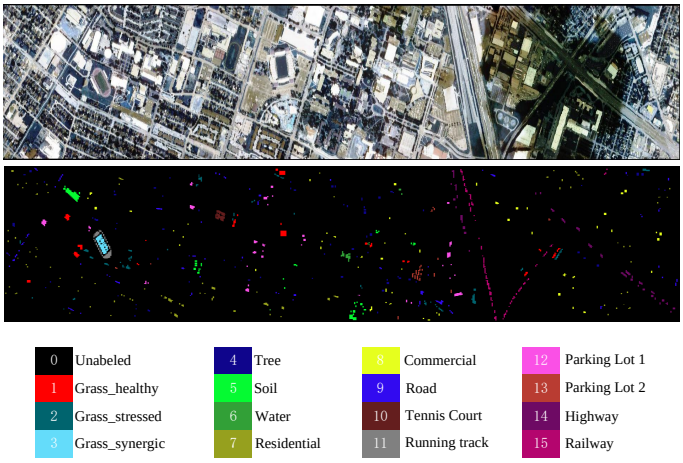


Figure 8. Ground truth map, pseudocolor images, and available samples of Houston scene.

4.2. Experimental Setup

For all experiments in this study, we split the labeled samples of the HSI data sets into three subsets, i.e., a training set, a balanced set, and a testing set. The training set is used to train feature extraction and the classifier module, while the balanced set is used to train the linear correction unit. In the initial phase, there are no new classes, and 10% of old class samples are used as training set D_0 to train the base model. In the later incremental phases, the set D_l, including the training and balanced set, is composed of 10% new class samples

of D_l^1 and several old class exemplars of D_l^0. In detail, we saved 100 old exemplars for each data set. The number of stored old exemplars is about 0.2% of the total number for Salinas and PaviaU, and about 0.6% for Houston, and we split them into two subsets, i.e., $D_l^{(0,0)}$ and $D_l^{(0,1)}$. Since the memory budget is constant, the saved old exemplars of each class will be reduced as the incremental learning process goes on. The number of samples in $D_l^{(1,1)}$ is the same as in $D_l^{(0,1)}$, and they compose the balanced set. The remaining samples of D_l constitute the training set (shown in Figure 5). The rest of the samples are used as the testing set to verify the effectiveness of the model.

For all HSIs, we preprocess them before experiments, including Gaussian filtering and normalization. Gaussian filtering is a process of weighted averaging of images. The value of each pixel is obtained by weighted averaging of itself and other pixel values in its neighborhood. Since the adjacent pixels of the hyperspectral image belong to the same category with a high probability, in order to further eliminate the influence of noise and make full use of the spatial information of the hyperspectral image, we perform a spatial filtering operation. The filter kernel is usually a Gaussian kernel, which performs a weighted average on the image to achieve the purpose of spatial smoothing. The normalization operation is carried out to avoid the feature dimensions with larger eigenvalues drowning the feature dimensions with smaller eigenvalues, so as to balance the contributions of various features and prevent the disappearance of features.

The model is trained on about half of the classes in the 0-th phase, i.e., 5 classes for PaviaU, 8 classes for Salinas, and 9 classes for Houston. Then, it learns the remaining classes on average in the subsequent L phases, 2 phases for PaviaU, 4 phases for Salinas, and 3 phases for Houston. We train the model for 100 epochs in each phase. The learning rate is set to 0.002 for the parameters optimization of the feature extraction module, which decays to 1/10 of the original value after 50 epochs. The learning rate is set to 0.5 for parameters optimization of the LC unit. We use an SGD optimizer with batch size 128 to train the models in all settings.

The performance of the proposed classification method in this paper is evaluated using three measurements, which are the most commonly used metrics in HSI classification: overall accuracy (OA), average accuracy (AA), and Kappa coefficient (κ). The OA is obtained by dividing all correctly classified samples by the total test samples, AA is the average of the accuracy of all classes, and κ is a measurement of robustness.

4.3. Experimental Results

In this subsection, we first execute an ablation study on the three hyperspectral data sets to verify the performance of each structure, and secondly, we compare the proposed model with other state-of-the-art incremental models to verify the superiority of our proposed model.

4.3.1. Ablation Experiments

We proposed three modules for avoiding catastrophic forgetting in this paper, i.e., channel attention, knowledge distillation, and linear correction. In order to explore the role of each module, we set up different models to conduct ablation experiments. The different model setups are explained as follows, and also in Table 1:

Table 1. Comparison of different models.

Model	L_c	L_d	LC	Attention
1	✓			
2	✓	✓		
3	✓	✓	✓	
Ours	✓	✓	✓	✓

Model-1: Only the classification loss is used alone for training, i.e., the model does not use knowledge distillation.

Model-2: Both classification loss and distillation loss are used for training, i.e., the model is trained using knowledge distillation.

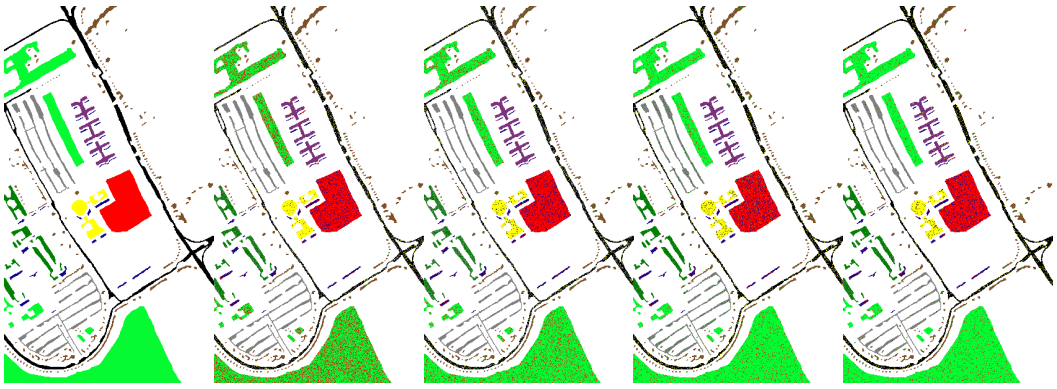
Model-3: The model is trained using knowledge distillation and linear correction unit.

Ours: The model is trained using the whole model we proposed, including channel attention, knowledge distillation, and linear correction unit.

Table 2 shows the ablation classification results of HSI class-incremental learning with different learning phases. Figure 9–11 are the classification maps of the last incremental phase of different hyperspectral data sets.

Table 2. Classification results of different HSIs and different models.

Model	Meas.	PaviaU			Salinas					Houston			
		1–5	1–7	1–9	1–8	1–10	1–12	1–14	1–16	1–9	1–11	1–13	1–15
1	OA(%)	99.90	92.05	75.30	99.89	96.52	95.93	95.47	80.81	99.23	86.85	80.31	75.64
	AA(%)	99.87	89.24	79.08	99.77	95.95	95.64	95.31	89.67	99.33	87.54	80.18	77.53
	$\kappa(\times 100)$	99.84	88.20	69.24	99.87	95.62	94.80	94.63	77.79	99.12	85.02	77.53	73.16
2	OA(%)	99.91	93.63	80.34	99.90	97.13	96.28	96.34	82.01	99.22	88.94	82.65	79.02
	AA(%)	98.89	90.46	81.80	99.50	96.40	95.95	96.37	91.26	99.31	89.41	82.37	80.33
	$\kappa(\times 100)$	99.83	91.59	77.02	99.84	96.38	96.25	95.73	80.86	99.14	88.29	81.05	77.37
3	OA(%)	99.91	96.85	88.18	99.90	97.22	96.43	96.18	86.25	99.22	91.24	86.70	86.27
	AA(%)	99.88	93.86	85.57	99.79	97.06	96.19	96.65	93.08	99.32	90.42	84.29	85.93
	$\kappa(\times 100)$	99.85	95.15	84.28	99.81	96.48	96.36	96.25	85.49	99.10	90.27	84.95	84.31
Ours	OA(%)	99.91	97.29	89.13	99.92	98.04	96.50	96.31	87.24	99.23	91.63	87.50	87.12
	AA(%)	99.87	94.79	85.76	99.82	97.37	96.18	96.49	93.31	99.31	90.51	84.90	86.54
	$\kappa(\times 100)$	99.84	95.81	85.10	99.90	96.90	96.41	96.52	86.53	99.13	90.60	85.29	85.40



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 9. PaviaU. Classification maps of different models. (**a**) Pseudocolor images; (**b**) Model-1; (**c**) Model-2; (**d**) Model-3; (**e**) ours.

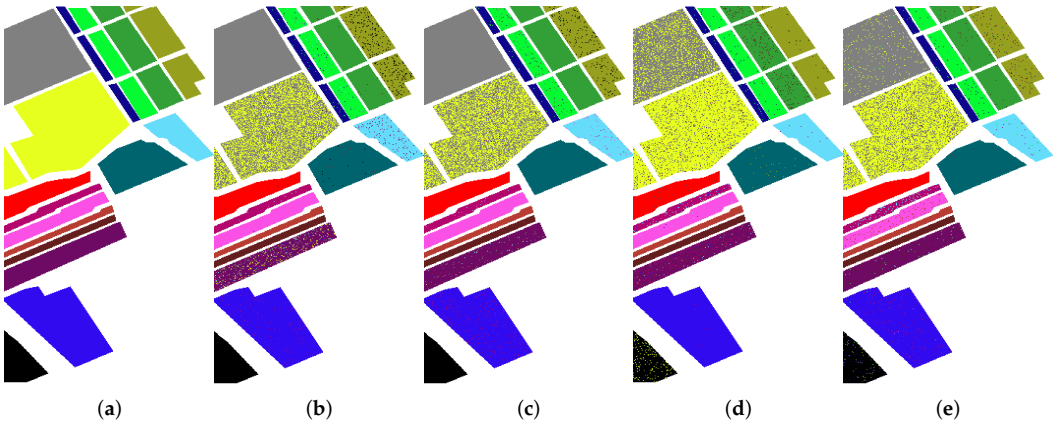


Figure 10. Salinas. Classification maps of different models. (**a**) Pseudocolor images; (**b**) Model-1; (**c**) Model-2; (**d**) Model-3; (**e**) ours.

From Table 2 we can see, in the 0-th phase, i.e., 1–5 classes of PaviaU, 1–8 classes of Salinas, and 1–9 classes of Houston, there are no new classes, so there is no catastrophic forgetting problem; the results of Model-1, Model-2 and Model-3 are basically the same; and the results with channel attention are a little better, which shows that the channel attention module works in all phases, while knowledge distillation and the bias correction unit only work in the incremental phases. The classification results of the first block (Model-1) are worse than others because it only uses classification loss and it has a problem of catastrophic forgetting. We use knowledge distillation in the second block (Model-2), and compare it with the first block; it achieves better results in most of the experiments. For the PaviaU and Houston datasets, knowledge distillation can improve the overall accuracy by more than 3% in the last stage, while for the Salinas dataset, the effect of knowledge distillation on improving the accuracy is not obvious. Because the dataset is easy to classify and not challenging, the classification loss can achieve good results.

If comparing the third block (Model-3) to the second block (Model-2), it is obvious that Model-3 can clearly improve the model performance, especially for the last phase of PaviaU (class 1–9) and Salinas (class 1–16), and the first phase of Houston (class 1–11). In the last incremental phase of PaviaU, the ninth class is the new class, i.e., Bare Soil, whose spectral curve is very similar to the second class of the old class. When there is a huge difference in sample size between these two classes, the classifier will be biased towards the larger number class, and the LC unit can correct the bias of the classifier, especially when the two classes are similar. In other words, when the new class is very similar to the old class, which is a more difficult scenario in incremental learning classification, adding a linear correction unit will be very useful. Similar situations also occur in the last incremental phase of the Salinas and Houston dataset.

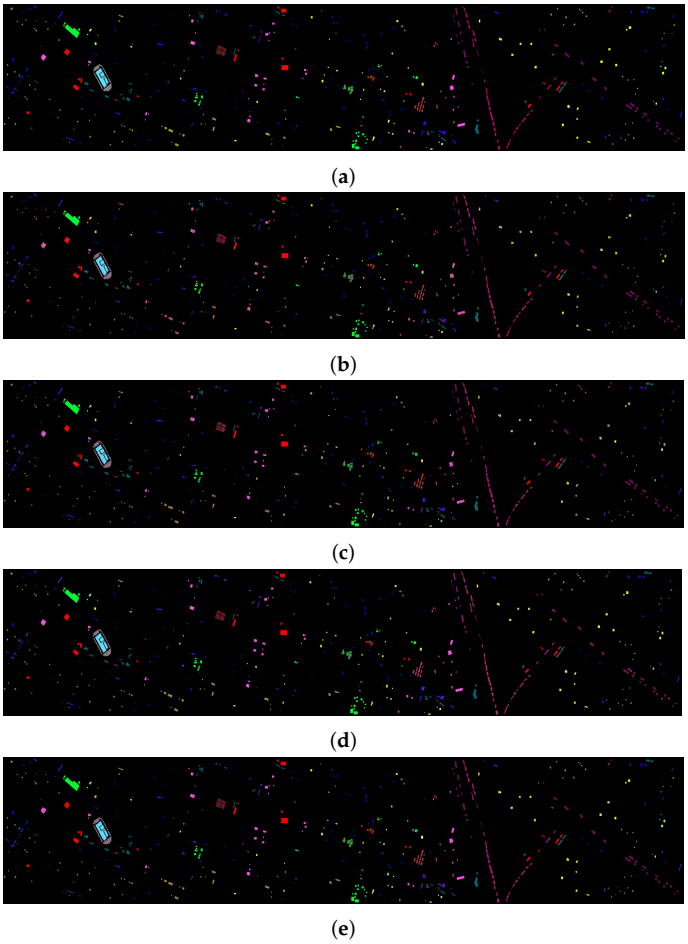


Figure 11. Houston. Classification maps of different models. (**a**) Pseudocolor images; (**b**) Model-1; (**c**) Model-2; (**d**) Model-3; (**e**) Ours.

Specifically, compared with Model-2, the OA of Model-3 is improved by about 3% in the first incremental phase of PaviaU and about 8% in the last phase. As for the Salinas and Houston dataset, the OA in the last phase is improved by more than 4%.

As for the fourth block (ours), we add channel attention module to the network, which has improved the classification accuracy at all phases. However, since the classification result maps we give are for the last incremental phase, the difference in the classification maps between the methods may not be significant, as not all similar classes are distributed in the last phase. In addition, due to the large difference in the number of samples in each class of hyperspectral images, the number of samples in some classes is small. There may be a significant improvement in the classification result indicators, but the classification maps are not significantly different.

4.3.2. Comparison

In order to prove the superiority of our proposed model, we compare ours against the other two incremental methods, iCaRL [48] and LUCIR [57]. In the comparative experiments, the proportion of training samples of the new class and the number of saved exemplars of the old class is the same as ours. The experimental results of PaviaU are shown in Figure 12. As we can see, our model is the best performer, and can better alleviate the problem of catastrophic forgetting compared to others.

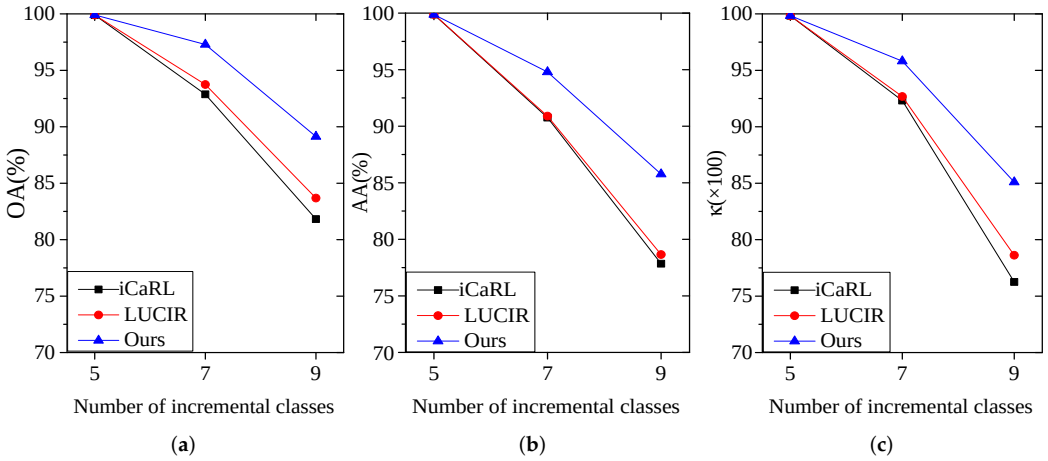


Figure 12. Comparison of classification results of different models, OA (**a**); AA (**b**); and κ (**c**).

We also apply our method to some scenarios in LPILC [55]. In LPILC, four different scenarios and two different tasks are set up. Firstly, our model is similar to the second scenario: the data imbalance situation. We store no more than 0.3% of the old class examples, which is less than 0.5% in LPILC. Secondly, ours is similar to the first type of task: add a new class, the difference is that we have more new classes and more incremental phases.

We take eight classes as the old class and one as the new class of PaviaU, similar to LPILC, we select the Shadows with the smallest number of samples as the new class. We randomly select 5% of the samples as the training data and store 210 (0.5%) exemplars of old data. The classification results of each class are shown in Table 3. In addition, we also set up two incremental phases for PaviaU, and compare ours with LPILC. The results are shown in Figure 13.

Table 3. Classification results of each class in the case of adding a new class.

Class	LPILC	Ours
Asphalt	98.56	97.91
Meadows	99.68	98.98
Gravel	94.25	92.65
Trees	92.4	95.74
Painted Metal Sheets	98.62	99.65
Bare Soil	99.37	98.10
Bitumen	93.96	92.42
Self-Blocking Bricks	95.04	86.57
Shadows (new class)	97.52	97.24
Original OA(%)	98.03	98.90
New OA(%)	97.52	97.24
OA(%)	98.02	96.59
AA(%)	96.60	95.47
$\kappa(\times 100)$	97.38	95.33

It can be seen from Table 3 that our model has achieved classification results comparable to LPILC when there is only one incremental phase and one new class. Because the spectral curve of the Shadows class is very distinguishable from other classes, it is not dominant to choose it as a new class for our model. As to Figure 13, our model is the best performer when there are two new classes. The focus of our proposed model is to consider the more difficult case where the new class is similar to the old classes.

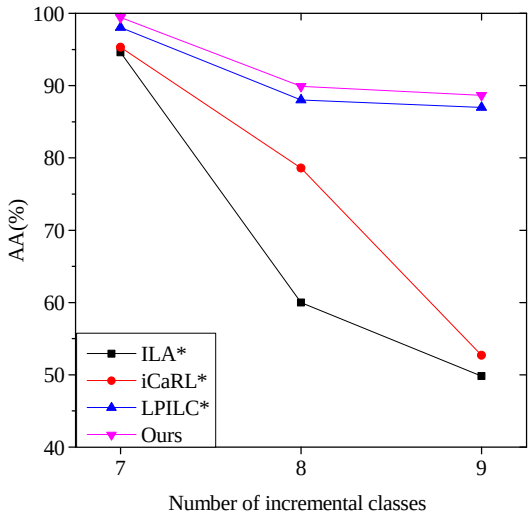


Figure 13. Comparison AA of PaviaU with other methods (ILA [58], iCaRL [48], LPILC [55]) when adding a new class.

5. Discussion

In this section, in order to explore the optimal classification performance and the application of the proposed model in practical remote sensing classification, we conduct further experiments and discussions on the following two aspects. First, we conduct parameter analysis experiments to discuss how to set each parameter to achieve optimal performance. Second, we compare the memory budget and running time of different strategies to explore the efficiency of our model.

5.1. Parameters Analysis

The parameters we analyzed can be divided into three categories. One is the parameters related to the network, i.e., network structure parameters. One is the hyperparameter, i.e., the η of loss in Equation (3). Another is the parameters related to sample allocation, i.e., the split of exemplars from old classes.

5.1.1. Network Parameters

There are many parameters related to the network structure. What we want to analyze here is the number of network layers, and the size of the Gaussian kernel. They work together to determine the structure of the network to obtain the best classification results, so we analyze them together.

The results of the network parameter experiments are shown in Table 4. To more intuitively choose the best parameters, we use bold fonts to denote the best classification results under the corresponding settings. The Baseline in the first column refers to the layers of ResNet. Because HSI classification is a relatively small task, the networks we choose are also small. The second column is the memory related to the corresponding network, and the third column is the size of the Gaussian kernel.

From the table, we can draw the following conclusions. Generally speaking, the more network layers, the better the feature extraction ability, until the limitations of this method are reached. However, more layers bring more network parameters, whose optimization may require more epochs, and the time and storage costs will also increase. The parameter amount of the eight-layer ResNet is about twice of the six-layer. In the comparison test of 100 epochs, the eight-layer does not show much better performance than the six-layer. For the Gaussian kernel, it generally becomes larger as the spatial resolution increases, and too-large size also brings more calculations. The spatial resolution of HSI is high, and the probability that adjacent pixels belong to the same category is high, so the Gaussian kernel is relatively large. We finally choose the first row of the table, i.e., the six-layer ResNet as the baseline, and 13×13 as the size of the Gaussian kernel.

Table 4. Classification results of different network parameters.

Baseline	Parameter	Kernel Size	Measurement	Classes of PaviaU		
				1–5	1–7	1–9
ResNet 6	2.41 M	13 × 13	OA(%)	**99.91**	**97.29**	89.13
			AA(%)	99.87	**94.79**	**85.76**
			$\kappa(\times 100)$	**99.84**	**95.81**	85.10
ResNet 8	5.96 M	13 × 13	OA(%)	99.90	96.96	**89.74**
			AA(%)	**99.88**	94.16	84.89
			$\kappa(\times 100)$	99.82	94.76	**85.13**
ResNet 6	2.41 M	11 × 11	OA(%)	99.88	96.13	87.89
			AA(%)	99.85	94.01	84.06
			$\kappa(\times 100)$	99.81	94.38	83.92
ResNet 6	2.41 M	15 × 15	OA(%)	99.87	96.27	87.96
			AA(%)	99.86	94.15	84.28
			$\kappa(\times 100)$	99.80	94.40	84.57

5.1.2. Hyperparameter

Next, we analyze the impact of η in the loss function on classification performance. The loss function of the new class is a combination of classification loss and distillation loss. We will analyze the proportion of each loss to achieve the best classification results; the analysis results are shown in Figure 14. Because the value of distillation loss is large while the value of classification loss is small, the value of the coefficient η of distillation loss is small. Through the classification results of different values, we finally choose the value of η to be 6%.

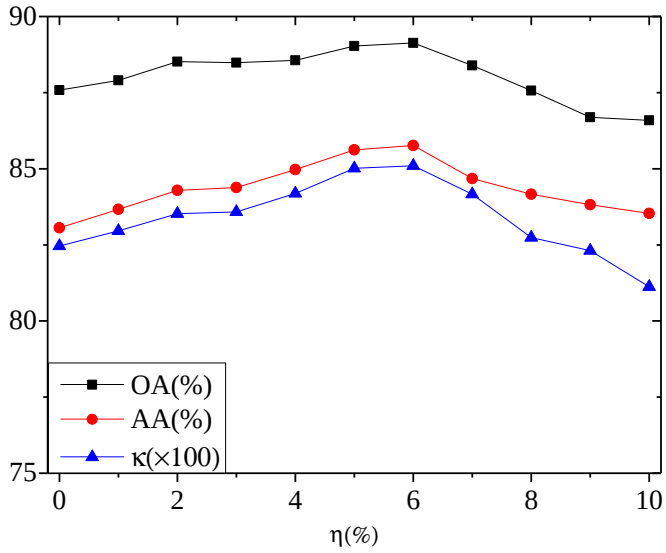


Figure 14. Classification results of different values of η.

5.1.3. Sample Parameters

We store a certain number of old class exemplars in the experiment and split them into $D_l^{(0,0)}$ and $D_l^{(0,1)}$, as illustrated in Figure 5. The $D_l^{(0,0)}$, combined with $D_l^{(1,0)}$, is used as training set to train the feature extraction and classifier module, while the $D_l^{(0,1)}$ and $D_l^{(1,1)}$ is used as a balanced set to train the LC unit. The number of stored exemplars of the old class is certain. It is necessary to find a good split to deal with the trade-off between feature representation and linear correction. In this part, we explore how to allocate these samples to the training set and the balanced set to achieve the best classification results. In order to

simplify the experiment, we use the ratio of the two sets instead of the sample size of each set to illustrate the results of the exploration.

Table 5 shows the different splits of the training set and balanced set. Where four different splits are set, the bold fonts are the best classification results under the corresponding setting. We compared these four splitting methods and found that the best results are obtained in 8:2, especially when there are more incremental phases.

Table 5. Classification results of different split ratio of exemplars.

Split	Measurement	Classes of PaviaU		
		1–5	**1–7**	**1–9**
9:1	OA(%)	99.90	96.87	88.32
	AA(%)	99.87	94.05	84.88
	$\kappa(\times 100)$	99.83	**95.84**	83.96
8:2	OA(%)	99.91	**97.29**	**89.13**
	AA(%)	99.87	**94.79**	**85.76**
	$\kappa(\times 100)$	**99.84**	95.81	**85.10**
7:3	OA(%)	99.90	97.04	88.87
	AA(%)	**99.88**	94.27	84.82
	$\kappa(\times 100)$	99.83	95.43	83.75
6:4	OA(%)	**99.91**	95.28	82.41
	AA(%)	99.87	90.92	80.64
	$\kappa(\times 100)$	99.83	92.49	76.31

In this paper, we use split 8:2 for all three hyperspectral data sets. In other words, 80% of the old examples are used to train the feature extraction and classifier module, and 20% are used to train the LC unit. A small number of samples is good enough to correct the bias of the classifier and estimate the bias parameters (α and β in Equation (4)).

5.2. Memory Budget and Running Time

We tested the data memory and running-time requirements of the proposed model, and compared it with methods that do not perform incremental learning, i.e., every time new data is added, the old class sample and the new class sample are used to retrain the network. The comparison results are shown in Table 6. It can be seen from the table that the memory and time requirements of the two methods are the same in the 0th phase because it does not involve incremental learning. In the subsequent incremental learning phase, our method greatly saves data memory and running time compared with the original method, and as the incremental learning phases increase, this contrast becomes more obvious. Therefore, our model can achieve good classification performance with as little memory budget and running time. However, we need to admit that our model relies on old examples, and if old data are not available, only knowledge distillation can work, and the effectiveness of the model may be affected. However, taking into account both the running time and the memory budget, our model still has a great advantage. In the practice of remote sensing classification application, the amount of data will be larger, and the advantages of our model will be more obvious, so our model has practical application significance.

Table 6. Classification results of different strategies.

Phase	Method	Time	Data Memory
0th	original	1.28 min	2.50 M
	Ours	1.28 min	2.50 M
2nd	original	1.47 min	2.86 M
	Ours	0.86 min	0.44 M
1st	original	1.78 min	3.36 M
	Ours	0.87 min	0.58 M

6. Conclusions

This paper proposes a model to address the class-incremental learning issue for HSI classification. Specifically, three incremental learning architectures, i.e., channel attention module, knowledge distillation, and linear correction unit, are used to keep the model learning new classes. The channel module can make good use of the interdependence between feature channels by assigning different weights to different channels, the knowledge distillation is able to transfer old knowledge to new, and the linear correction unit is proposed to balance old and new classes and correct the bias of FC layer to new classes. Experiments performed on three widely used real hyperspectral data sets demonstrate the outstanding performance and less memory and time requirements of the proposed class-incremental HSI classification method compared with the methods without our proposal.

Since the method of class-incremental learning is closer to the way the ground covers develop and change in nature, and it avoids the use of large-scale old HSI to repeatedly train new networks, the proposed model is very useful in practical applications.

However, we need to acknowledge that the performance of the method is still unsatisfactory in complicated scenarios, which can be attributed to many factors, such as spectral similarity among different classes. Incremental learning has a very important meaning for HSI classification. In the future, we will continue to explore the direction of class-incremental learning for HSI to seek better classification results.

Author Contributions: Conceptualization, M.X.; methodology, Y.Z. and Y.L.; software, Y.Z.; validation, M.X. and Y.Z.; writing—original draft preparation, M.X.; writing—review and editing, M.X. and Y.Z.; supervision, X.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grants 61801078 and Liaoning Province Natural Science Foundation under grant 2020-MS-099.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank all researchers and organizations for providing the hyperspectral images with corresponding ground-truth maps, and related experimental codes used in our experiments.

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

References

1. Li, Z.; Huang, L.; He, J. A Multiscale Deep Middle-Level Feature Fusion Network for Hyperspectral Classification. *Remote Sens.* **2019**, *11*, 695. [CrossRef]
2. Hsieh, T.H.; Kiang, J.F. Comparison of CNN Algorithms on Hyperspectral Image Classification in Agricultural Lands. *Sensors* **2020**, *20*, 1734. [CrossRef] [PubMed]
3. Masoud, M.; Bahram, S.; Mohammad, R.; Fariba, M.; Yun, Z. Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery. *Remote Sens.* **2018**, *10*, 1119.
4. Zhao, C.; Wang, Y.; Qi, B.; Wang, J. Global and Local Real-Time Anomaly Detectors for Hyperspectral Remote Sensing Imagery. *Remote Sens.* **2015**, *7*, 3966–3985. [CrossRef]
5. Zhang, X.; Sun, Y.; Jiang, K.; Li, C.; Jiao, L.; Zhou, H. Spatial Sequential Recurrent Neural Network for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2018**, *11*, 4141–4155. [CrossRef]

6. Liu, S.; Shi, Q.; Zhang, L. Few-Shot Hyperspectral Image Classification With Unknown Classes Using Multitask Deep Learning. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 5085–5102. [CrossRef]
7. Melgani, F.; Bruzzone, L. Classification of Hyperspectral Remote Sensing Images with Support Vector Machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
8. Plaza, J.; Plaza, A.J.; Barra, C. Multi-Channel Morphological Profiles for Classification of Hyperspectral Images Using Support Vector Machines. *Sensors* **2009**, *9*, 196–218. [CrossRef]
9. Ma, W.; Gong, C.; Hu, Y.; Meng, P.; Xu, F.; Zhang, L.; Yang, J. The Hughes Phenomenon in Hyperspectral Classification Based on the Ground Spectrum of Grasslands in the Region Around Qinghai Lake. In *International Symposium on Photoelectronic Detection and Imaging 2013: Imaging Spectrometer Technologies and Applications*; International Society for Optics and Photonics: Washington, DC, USA 2013; p. 89101G.
10. Martinez-UsoMartinez-Uso, A.; Pla, F.; Sotoca, J.M.; Garcia-Sevilla, P. Clustering-Based Hyperspectral Band Selection Using Information Measures. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 4158–4171. [CrossRef]
11. Licciardi, G.; Marpu, P.R.; Chanussot, J.; Benediktsson, J.A. Linear Versus Nonlinear PCA for the Classification of Hyperspectral Data Based on the Extended Morphological Profiles. *IEEE Geosci. Remote Sens. Lett.* **2012**, *9*, 447–451. [CrossRef]
12. Li, H.; Xiao, G.; Xia, T.; Tang, Y.Y.; Li, L. Hyperspectral Image Classification Using Functional Data Analysis. *IEEE Trans. Cybern.* **2014**, *44*, 1544–1555. [PubMed]
13. Bandos, T.V.; Bruzzone, L.; Camps-Valls, G. Classification of Hyperspectral Images With Regularized Linear Discriminant Analysis. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 862–873. [CrossRef]
14. Fauvel, M.; Benediktsson, J.A.; Chanussot, J.; Sveinsson, J.R. Spectral and Spatial Classification of Hyperspectral Data Using SVMs and Morphological Profiles. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 3804–3814. [CrossRef]
15. Patra, S.; Bhardwaj, K.; Bruzzone, L. A Spectral-Spatial Multicriteria Active Learning Technique for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2017**, *10*, 5213–5227. [CrossRef]
16. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep Learning-Based Classification of Hyperspectral Data. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2014**, *7*, 2094–2107. [CrossRef]
17. Cui, B.; Cui, J.; Hao, S.; Guo, N.; Lu, Y. Spectral-spatial hyperspectral image classification based on superpixel and multi-classifier fusion. *Int. J. Remote. Sens.* **2020**, *41*, 6157–6182. [CrossRef]
18. Sk, A.; Kk, B.; Aa, C. A New CNN Training Approach with Application to Hyperspectral Image Classification. *Digit. Signal Prog.* **2021**, *113*, 103016.
19. Li, Z.; Cui, X.; Wang, L.; Zhang, H.; Zhang, Y. Spectral and Spatial Global Context Attention for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 771. [CrossRef]
20. French, R.M. Catastrophic forgetting in connectionist networks. *Trends Cognit. Sci.* **1999**, *3*, 128–135. [CrossRef]
21. Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; Fu, Y. Large Scale Incremental Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; Volume 1, pp. 374–382.
22. Li, S.; Zhu, X.; Bao, J. Hierarchical Multi-Scale Convolutional Neural Networks for Hyperspectral Image Classification. *Sensors* **2019**, *19*, 1714. [CrossRef]
23. Li, C.; Yang, S.X.; Yang, Y.; Gao, H.; Zhao, J.; Qu, X.; Wang, Y.; Yao, D.; Gao, J. Hyperspectral Remote Sensing Image Classification Based on Maximum Overlap Pooling Convolutional Neural Network. *Sensors* **2018**, *18*, 2166–2170. [CrossRef]
24. Tun, N.L.; Gavrilov, A.; Tun, N.M.; Trieu, D.M.; Aung, H. Hyperspectral Remote Sensing Images Classification Using Fully Convolutional Neural Network. In Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), St. Petersburg, Russia, 26–29 January 2021; pp. 2166–2170.
25. Chen, Y.; Zhu, K.; Zhu, L.; He, X.; Ghamisi, P.; Benediktsson, J.A. Automatic Design of Convolutional Neural Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 7048–7066. [CrossRef]
26. Zhang, J.; Wei, F.; Feng, F.; Wang, C.y. Spatial–Spectral Feature Refinement for Hyperspectral Image Classification Based on Attention-Dense 3D-2D-CNN. *Sensors* **2020**, *20*, 5191. [CrossRef] [PubMed]
27. Zhang, X.; Wang, Y.; Zhang, N.; Xu, D.; Luo, H.; Chen, B.; Ben, G. Spectral–Spatial Fractal Residual Convolutional Neural Network With Data Balance Augmentation for Hyperspectral Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 10473–10487. [CrossRef]
28. Ma, X.; Wang, H.; Geng, J. Spectral-Spatial Classification of Hyperspectral Image Based on Deep Auto-Encoder. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2016**, *9*, 4073–4085. [CrossRef]
29. Zhou, S.; Xue, Z.; Du, P. Semisupervised Stacked Autoencoder With Cotraining for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 3813–3826. [CrossRef]
30. Tao, C.; Pan, H.; Li, Y.; Zou, Z. Unsupervised Spectral-Spatial Feature Learning With Stacked Sparse Autoencoder for Hyperspectral Imagery Classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2438–2442.
31. Jinling, Z.; Lei, H.; Yingying, D.; Linsheng, H.; Shizhuang, W. A combination method of stacked autoencoder and 3D deep residual network for hyperspectral image classification. *Int. J. Appl. Earth Obs. Geoinform.* **2021**, *102*, 102459.
32. Chen, Y.; Zhao, X.; Jia, X. Spectral-Spatial Classification of Hyperspectral Data Based on Deep Belief Network. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2015**, *8*, 2381–2392. [CrossRef]
33. Zhong, P.; Gong, Z.; Li, S.; Schonlieb, C.B. Learning to Diversify Deep Belief Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3516–3530. [CrossRef]

34. Arsa, D.M.S.; Jati, G.; Mantau, A.J.; Wasito, I. Dimensionality Reduction Using Deep Belief Network in Big Data Case Study: Hyperspectral Image Classification. In Proceedings of the 2016 International Workshop on Big Data and Information Security (IWBIS), Jakarta, Indonesia, 18–19 October 2016; pp. 71–76.
35. Mughees, A.; Tao, L. Multiple Deep-Belief-Network-Based Spectral-Spatial Classification of Hyperspectral Images. *Tsinghua Sci. Tech.* **2019**, *24*, 183–194. [CrossRef]
36. Chen, C.; Ma, Y.; Ren, G. Hyperspectral Classification Using Deep Belief Networks Based on Conjugate Gradient Update and Pixel-Centric Spectral Block Features. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2020**, *13*, 4060–4069. [CrossRef]
37. Liu, Y.; Cao, G.; Sun, Q.; Siegel, M. Hyperspectral Classification via Deep Networks and Superpixel Segmentation. *Int. J. Remote Sens.* **2015**, *36*, 3459–3482. [CrossRef]
38. Chen, X.; Li, M.; Yang, X. Stacked Denoise Autoencoder Based Feature Extraction and Classification for Hyperspectral Images. *J. Sens.* **2015**, *2016*, 1–10.
39. Rahaf, A.; Francesca, B.; Mohamed, E.; Marcus, R.; Tinne, T. Memory Aware Synapses: Learning what (not) to forget. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 139–154.
40. Li, Z.; Hoiem, D. Learning Without Forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2935–2947. [CrossRef]
41. Rannen, A.; Aljundi, R.; Blaschko, M.B.; Tuytelaars, T. Encoder Based Lifelong Learning. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1329–1337.
42. Zhu, F.; Zhang, X.Y.; Wang, C.; Yin, F.; Liu, C.L. Prototype Augmentation and Self-Supervision for Incremental Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 5867–5876.
43. Jaehong, Y.; Eunho, Y.; Jungtae, L.; Hwang, S.J. Lifelong Learning with Dynamically Expandable Networks. *arXiv* **2017**, arXiv:1708.01547.
44. Mallya, A.; Davis, D.; Lazebnik, S. Piggyback: Adapting a Single Network to Multiple Tasks by Learning to Mask Weights. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 72–88.
45. Serra, J.; Suris, D.; Miron, M.; Karatzoglou, A. Overcoming Catastrophic Forgetting with Hard Attention to the Task. *Int. Conf. Mach. Learn.* **2018**, *80*, 4548–4557.
46. Achituve, I.; Navon, A.; Yemini, Y.; Chechik, G.; Fetaya, E. GP-Tree: A Gaussian Process Classifier for Few-Shot Incremental Learning. *arXiv* **2021**, arXiv:2102.07868v4.
47. Chaudhry, A.; Marc'Aurelio, R.; Rohrbach, M.; Elhoseiny, M. Efficient Lifelong Learning with A-GEM. *arXiv* **2019**, arXiv:1812.00420.
48. Rebuffi, S.A.; Kolesnikov, A.; Sperl, G.; Lampert, C.H. iCaRL: Incremental Classifier and Representation Learning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5533–5542.
49. Liu, Y.; Schiele, B.; Sun, Q. Adaptive Aggregation Networks for Class-Incremental Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2544–2553.
50. Yan, S.; Zhou, J.; Xie, J.; Zhang, S.; He, X. An EM Framework for Online Incremental Learning of Semantic Segmentation. *arXiv* **2021**, arXiv:2108.03613.
51. Cermelli, F.; Mancini, M.; Rota Bulo, S.; Ricci, E.; Caputo, B. Modeling the Background for Incremental Learning in Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9230–9239.
52. Guanglei, Y.; Enrico, F.; Dan, X.; Paolo, R.; Mingli, D.; Hao, T.; Xavier, A.P.; Elisa, R. Continual Attentive Fusion for Incremental Learning in Semantic Segmentation. *arXiv* **2022**, arXiv:2202.00432.
53. Fabio, C.; Massimiliano, M.; Samuel, R.B.; Elisa, R.; Barbara, C. Modeling the Background for Incremental and Weakly-Supervised Semantic Segmentation. *arXiv* **2022**, arXiv:2201.13338.
54. Tasar, O.; Tarabalka, Y.; Alliez, P. Incremental Learning for Semantic Segmentation of Large-Scale Remote Sensing Data. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2019**, *12*, 3524–3537. [CrossRef]
55. Bai, J.; Yuan, A.; Xiao, Z.; Zhou, H.; Wang, D.; Jiang, H.; Jiao, L. Class Incremental Learning With Few-Shots Based on Linear Programming for Hyperspectral Image Classification. *IEEE Trans. Cybern.* **2020**, 1–12. [CrossRef]
56. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–19 June 2019; pp. 7132–7141.
57. Hou, S.; Pan, X.; Loy, C.C.; Wang, Z.; Lin, D. Learning a Unified Classifier Incrementally via Rebalancing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 831–839.
58. Mariela, A.; Eduardo, G.; Rendón, E. Implementation of incremental learning in artificial neural networks. In Proceedings of the 3rd Global Conference on Artificial Intelligence (GCAI), Miami, FL, USA, 18–22 October 2017; pp. 221–232.