CILEA-NET: Curriculum-Based Incremental Learning Framework for Remote Sensing Image Classification

S. Divakar Bhat, Biplab Banerjee, *Member, IEEE*, Subhasis Chaudhuri, *Fellow, IEEE*, and Avik Bhattacharya, *Senior Member, IEEE*

***Abstract*—In this article, we address class incremental learning (IL) in remote sensing image analysis. Since remote sensing images are acquired continuously over time by Earth's observation sensors, the land-cover/land-use classes on the ground are likely to be found in a gradational manner. This process restricts the deployment of stand-alone classification approaches, which are trained for all the classes together in one iteration. Therefore, for every new set of categories discovered, the entire network consisting of old and new classes requires retraining. This procedure is often impractical, considering vast volumes of data, limited resources, and the complexity of learning models. In this respect, we propose a convolutional-neural-network-based framework (called CILEA-NET, curriculum-based incremental learning framework for remote sensing image classification) to efficiently resolve the difficulties associated with incremental learning paradigm. The framework includes new classes in the already trained model to avoid catastrophic forgetting for the old while ensuring improved generalization for the newly added classes. To manage the IL's stability-plasticity dilemma, we introduce a novel curriculum learning-based approach where the order of the new classes is devised based on their similarity to the already trained classes. We then perform the training in that given order. We notice that the curriculum learning setup distinctly enhances the training time for the new classes. Experimental results on several optical datasets: PatternNet and NWPU-RESISC45, and a hyperspectral dataset: Indian Pines, validate the robustness of our technique.**

***Index Terms*—Classification, curriculum, incremental learning (IL), remote sensing.**

I. Introduction

SIGNIFICANT research in remote sensing image analysis has rendered the field with considerable momentum resulting in the rapid development of innovative solutions to many advanced problems. This drive is majorly attributable to more enhanced satellites being deployed and remote sensing imaging technologies being advanced to acquire abundant data [1]. In the recent past, the deep convolutional neural network (CNN) has proven to achieve remarkable success in this regard, facilitating the notion of data-driven feature representation learning.

The Earth observation sensors obtain images continuously over time. The land cover/ land use classes are discovered only sequentially. Whereas in the general setting, the entire dataset, including all the class information, is present during training. Therefore, many remote sensing data are subjected to temporal, spatial, and spectral resolution limitations. Besides, taking advantage of these rich data reservoirs for supervised learning algorithms, the samples must be meticulously annotated. Annotation becomes demanding for vast multitemporal datasets. The steady accumulation of such an enormous amount of dynamic data necessitates a framework capable of continually learning as and when the sequentially annotated data are made available.

However, selecting model parameters for a deep learning framework to dynamically varying incoming data streams is particularly difficult. On the one hand, even though quick updates will ensure adaptation to new data streams, it will also rapidly forget old information. On the other hand, the network's reactivity drastically decreases if the updates are made gradually for retaining the learned information [2]. This phenomenon is well known in the literature as the stability-plasticity dilemma [3], which is a significant constraint for artificial learning systems.

Therefore, continual learning of dynamic distributions of data will thus lead to catastrophic forgetting [4], i.e., the dramatic decrease in the model's performance when training with data corresponding to new classes added incrementally. At the outset, it may appear that training the network from scratch, every time new data are encountered, which can potentially solve this problem. However, as in remote sensing satellites where earth observation sensors acquire new images from all around the globe every day, storing and retraining over an enormous volume of data becomes an unsustainable task. Thus, new algorithms have to be developed to mitigate forgetting while facilitating continual learning by relying on previously acquired knowledge.

Early attempts in alleviating the effect of catastrophic forgetting consisted of storing old data and replaying them repeatedly in an interleaved fashion along with the new samples [5]. However, this demanded a large amount of specific data storage

Manuscript received January 22, 2021; revised March 30, 2021 and May 3, 2021; accepted May 18, 2021. Date of publication May 27, 2021; date of current version June 16, 2021. This work was supported by the India-Trento (University of Trento, Italy), Program for Advanced Research (ITPAR), funded through the DST, New Delhi, India. *(Corresponding author: Biplab Banerjee.)*

S. Divakar Bhat and Subhasis Chaudhuri are with the Department of Electrical Engineering, Indian Institute of Technology Bombay, Mumbai 400076, India (e-mail: sdivakarbhat@gmail.com; sc@ee.iitb.ac.in).

Biplab Banerjee and Avik Bhattacharya are with the Centre of Studies in Resources Engineering, Indian Institute of Technology Bombay, Mumbai 400076, India (e-mail: getbiplab@gmail.com; avikb@csre.iitb.ac.in).

Digital Object Identifier 10.1109/JSTARS.2021.3084408

requirements. Adding more neurons as suggested in [6] can also primarily alleviate the issue. But, the growing model size would demand more resource allocation and gradually will become infeasible. Enforcing the similarity of previously learned tasks with the current task [7] using the knowledge distillation technique is one of the first methods suggested to alleviate the interference issue in deep networks. Still, this approach is highly dependent on the nature of the task, and the training time linearly increases with the number of tasks. In [8], a proposal is presented to assign importance value to individual neurons for a task and penalize changes in neurons. While [9] recommends modifying the network's architectural properties dynamically in response to incoming new data. Deep generative models-based approaches like the one proposed in [10] utilizes replaying previously encoded information by training the model by interleaved samples with new information with pseudodata generation.

Despite the advancements made in incremental learning, the methods mentioned previously only qualify with minor clarifications. They can be very poor when scaled up to domains with higher complexities. These could translate to high computational and memory resource requirements for remote sensing image analysis to process, manage, and store the continuous inflow of rich data. This perpetual flow of data from satellites would also mean that annotating these extensive image data collections would be tedious but unavoidable. It is practically impossible to fathom the possibility of having all the class information beforehand. Furthermore, unlike expected from an efficient learning system to learn by connecting new information to related knowledge gained earlier, none of these approaches employ similarity between the incoming data and previously acquired knowledge to devise a better training strategy.

Curriculum learning is a training strategy introduced in [11], which aims at learning efficiently by presenting the data in a more meaningful order in terms of constituent concepts or complexity. Moreover, it is apparent from [12] and [13] that animals can learn much faster when a task is decomposed into subtasks from easy to complex based on a particular curriculum. This curriculum-based learning and replay-based revision technique are typically used in a learning ecosystem by humans, allowing us to learn tasks continually in an efficient manner. The curriculum learning approach results in faster training in the incremental setting as the model does not spend time on challenging samples for which it is not equipped at the moment. Instead, the model is trained with a similarity-based curriculum that decides how the data are fed into the learning system. It is observed that curriculum learning makes the approach more generalized by guiding the training toward a better optimum.

Inspired by this, we aspire to combine the efficient learning strategy provided by the curriculum technique with the replay-based incremental learning to counter forgetting, thus resulting in an efficient incremental learning framework. We hypothesize that integrating curriculum learning techniques into the continual learning framework will result in faster training and better overall performance. Fig. 1 provides a concise overview of the existing traditional classification approach and the proposed framework. In this article, we propose a new curriculum-based incremental learning approach to classify remote sensing images. The proposed work consists of the following contributions.

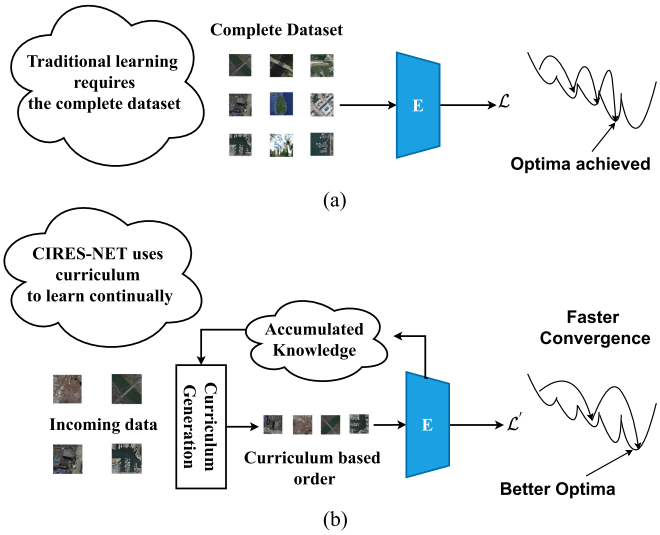

Fig. 1. Illustration comparing the traditional learning approach and the proposed scheme of incremental learning. (a) Traditional deep learning scenario where the complete dataset is expected to be present while training. (b) Proposed approach to learn incrementally by integrating curriculum learning strategy with continual learning.

1) A novel incremental learning-based approach for remote sensing image classification using a curriculum learning technique (CILEA-NET).
2) We present a pseudo-teacher–student-based approach for incremental curriculum learning. It is shown that faster convergence with more generalized learning is perceived using a curriculum, which yields better results.
3) We perform thorough experiments with both optical and hyperspectral datasets to demonstrate how curriculum learning improves the performance of an incremental learning network for satellite image classification.

II. Related Work

A. Incremental Learning

Early attempts in mitigating catastrophic forgetting involved a continuous replay of previous knowledge interleaved with the newly acquired set of data as proposed in [5]. In contrast, in [14] and [15], it was suggested to use SVM- and RBF-based networks, respectively, for training a model in the incremental setting. In [16], a random-forest-based approach was introduced, which grows hierarchically when a new set of data is encountered. Whereas in [17] and [18], they try to control the extent of catastrophic forgetting by learning masks corresponding to important neurons in each task. While hard attention to the task (HAT) [19] learns masks for activations rather than for parameters.

The approach used in [20] and [21] employ an elastic weight consolidation (EWC) term, which denotes the importance of neurons corresponding to the old tasks and imposes a quadratic

penalty on the difference between previous and updated network. In [8], the importance of weight computation is performed online by tracking the change in loss and updating the parameters accordingly. While [7] and [22] aligns the predictions based on the current tasks at hand.

In [23], it is recommended to use a small fraction of data consisting of the most representative samples of the previous classes and the new training data. In comparison, in [10], the pseudosamples corresponding to the old classes are generated by generative adversarial networks (GANs). Knowledge distillation [24] based training is used in [7] to transfer knowledge from the previous network to the updated one.

While [25] and [26] use most representative samples from the old tasks interleaved with new samples combined with distillation-based learning to propagate information from the previous model, in [27], the authors propose to use an intermediate expert to train the model to adapt to the new task using distillation and sample caching. The approach in [28] tries to retrieve only the samples that are most conflicted. At the same time, [29] proposes to improve the performance by greedily storing samples in memory and retraining on these stored samples while testing. The authors in [30] propose an expansion-based approach for task-free continual learning built upon the Bayesian nonparametric.

Although there are few incremental learning methods in remote sensing, the few existing techniques are mentioned here. A recent work [31] requires an auxiliary network for selecting the task in the absence of which the approach fails drastically. Unlike this, our approach does not utilize any added network for task selection. In [32], the authors have tried to introduce a large-scale remote sensing scene classification benchmark to help develop incremental learning algorithms in the field of remote sensing image scene classification. However, the NWPU-RESISC45 dataset used in this study is much larger than the proposed benchmark dataset regarding the total number of images, classes, and spatial resolution. In [26], the authors use incremental learning to perform semantic segmentation; however, the overall number of classes considered for this purpose is small. In [33], the authors explore an end-to-end incremental semantic segmentation for global mapping of buildings from VHR satellite images. One can note that both these works explore continual learning on an entirely diverse task that differs from the one proposed in this work.

B. Curriculum Learning

The concept of training neural networks with a curriculum was introduced in [34]. The idea is to learn tasks from easy to complex gradually. A well-chosen curriculum can act as a continuation method [35], potentially pushing toward a more general and optimal solution. In [11], it was confirmed that training with a curriculum strategy could result in faster training, and lead toward better regions in the parameter space. Approaches like active learning are similar to the curriculum learning paradigm but essentially differ from it due to the dynamic sampling of training points based on the current hypothesis of the model [36]. While active learning works at the sample level, curriculum learning can happen at the sample and class levels. Moreover, the idea of learning concepts from easy to hard is unique to curriculum learning. Remote sensing witnessed the use of curriculum learning through the work [37], which proposed an interesting approach to improve the weakly supervised object detection performance in high spatial resolution images.

Despite several in-depth studies on incremental learning approaches in the literature, none of these works explores the effectiveness of curriculum learning as a technique for faster convergence and better generalization of the model in the incremental learning perspective. In this regard, our approach is generic as the proposed algorithm can be utilized across any satellite image data to achieve improved performance in continual learning.

III. Problem Statement

Consider an incoming stream s of data, in which the available classes are $\mathcal{D}^s_{\text{train}} = \bigcup_{i=1}^{k} \{(\mathcal{X}^j_{(s-1)k+i}, \mathcal{Y}^j_{(s-1)k+i})\}_{j=1}^{N}$, where k is the number of incremental classes per stream (step size) and N is the number of samples per class of the incoming stream, ordered pair $(\mathcal{X}^j_{(s-1)k+i}, \mathcal{Y}^j_{(s-1)k+i})$ denotes the jth sample and ground truth labels corresponding to the ith class in the current data stream s. Let $\mathcal{D}^s_{\text{mem}}$ be the set denoting the accumulated representative samples up to and including the $(s-1)$th stream of data. At the given stream s, our objective is to learn a representation for the new stream of data $\mathcal{D}^s_{\text{train}}$ with minimum interference to the knowledge acquired till the previous stream.

IV. Methodology

A. Method Overview and Architecture

The CILEA-NET framework uses curriculum learning to boost the performance and decrease convergence time at every incremental step. We utilize the data prepared based on the curriculum generated for the incoming stream s to train the network. Through the process, the classification layers should adapt the parameters to the changes in the features learned due to the unseen classes of data from the newly acquired stream.

While knowledge distillation ensures that the model retains previously learned information without succumbing to catastrophic forgetting. The curriculum-based training facilitates the network to learn by sequentially ordering the new classes in $\mathcal{D}^s_{\text{train}}$ based on their similarity to the information learned from $\mathcal{D}^{s-1}_{\text{train}}$. We utilize a small fraction of $\mathcal{D}^{s-1}_{\text{train}}$ as memory accumulated into $\mathcal{D}^{s-1}_{\text{mem}}$ to form $\mathcal{D}^s_{\text{mem}}$ for replay at the stream s, over which the distillation is performed. The subsequent sections explain this learning process incrementally over a new stream of data with lesser time and better accuracy utilizing the proposed novel pseudo-teacher–student framework.

The CILEA-NET utilizes a CNN-based encoder, denoted by Θ as its architecture's backbone. We introduce an additional d-dimensional fully connected layer as the penultimate classification layer. We extract the features corresponding to the image samples from this d-dimensional layer for curriculum generation. A new set of classification layer nodes are also integrated into the last layer to accommodate the incremental addition of

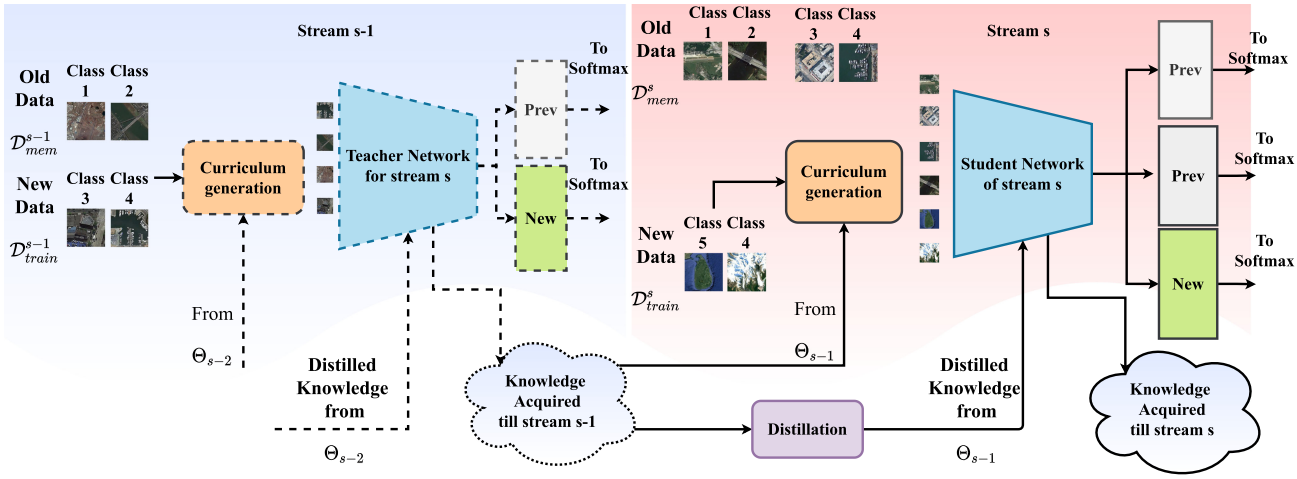

Fig. 2. Complete pipeline of the proposed CILEA-NET framework. In the stream s, the incoming novel classes are ordered into a curriculum. The student network learns the features from a new stream of data based on the curriculum generated. We use the teacher network as a proxy for previously learned information. The teacher Θ_{s-1} transfers the knowledge gained until stream $s-1$ to the student Θ_s. We employ the knowledge distillation technique to transfer this information from the teacher network to the student. With the incremental addition of novel training data in every new stream, a new set of classification layer nodes are integrated into the last layer, as shown in the figure. Note that $\mathcal{D}^s_{\text{train}}$ denote the new stream of data, while $\mathcal{D}^s_{\text{mem}}$ denote the fraction of data retained from previous streams.

novel training data with unseen classes in every new stream. Thus, at the ith stream with each stream of step size k, we have $(i-1)\times k$ node classification layer corresponding to previous classes, and one k node classification layer corresponding to the incoming unseen classes. We determine the dimension of this new set of classification layer nodes based on the number of unseen classes present in the incoming stream. As indicated before, this work utilizes a pseudo-teacher–student-based incremental learning algorithm. Both the teacher and student networks follow the same architecture except for the increment in the number of total classes handled by the student network.

As shown in Fig. 2, the pipeline initially starts with a traditional deep convolutional network framework to learn features corresponding to $\mathcal{D}^s_{\text{train}}$ the data of sth stream at the given instant. A curriculum is generated for the new data $\mathcal{D}^s_{\text{train}}$ by comparing it with the already learned representation corresponding to $\mathcal{D}^{s-1}_{\text{train}}$ from the teacher network at the stream s. This approach allows us to use any convenient architecture with only minor modifications. CILEA-NET being a curriculum-based method, the curriculum generation part required is separately handled by extracting the features from the teacher network in between the streams of data, followed by the steps specified in Section IV-C.

B. CILEA-NET Training Algorithm

For the discussions henceforth, we will use the variable s to denote the current stream of data, $\mathcal{D}^s_{\text{train}}$ will denote the new stream of data, while $\mathcal{D}^s_{\text{mem}}$ will denote the fraction of data retained from the previous streams. It consists of a fraction of image samples retained from each of the classes encountered in the previous streams, i.e., $\mathcal{D}^s_{\text{mem}} = \bigcup_{l=1}^{k(s-1)} \{\mathcal{X}^j_{l_{\text{mem}}}\}_{j=1}^{m.N}$, where l indexes the previously learned classes, and m is the fraction of data retained. We assume that at a given stream s and for incremental step size k, $\bigcap_{i=0}^{s.k}\{\mathcal{Y}_i\} = \phi$. For ease of representation, the sets of sample and ground truth belonging to $\mathcal{D}^s_{\text{train}}$ and $\mathcal{D}^s_{\text{mem}}$ may also be denoted by dropping the indices as $\{\mathcal{X}^s_{\text{train}}, \mathcal{Y}^s_{\text{train}}\}$ and $\{\mathcal{X}^s_{\text{mem}}, \mathcal{Y}^s_{\text{mem}}\}$, respectively. C_s will signify the curriculum generated for the new stream of data $\mathcal{D}^s_{\text{train}}$, and k will indicate the number of classes incrementally added per stream.

In the initial training phase, the model needs to be trained on the first incoming stream of data, i.e., $s=1$. For this, we consider $\mathcal{D}^1_{\text{train}} = \bigcup_{i=1}^{k}\{(\mathcal{X}^j_i, \mathcal{Y}^j_i)\}_{j=1}^{N}$, i.e., the pair of data samples and ground truth for the first k classes. We train the model for this data using the traditional multiclass cross-entropy loss function as in the following equation to perform a k class classification.

$$\mathcal{L}_C = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k} p_{ij}\log q_{ij} \tag{1}$$

where N denotes the total number of samples, and k denotes the total number of classes in the stream $s=1$. p_{ij} is the ground truth and q_{ij} denotes the softmax output for the sample corresponding to the ith sample of the jth class.

After the convergence of the model on the first stream of data, a small fraction of the data $\mathcal{D}^1_{\text{train}}$ is sampled by performing random selection per class; pruning the remaining samples. We store the retained samples as $\mathcal{D}^2_{\text{mem}}$ for replay during the training in subsequent data streams.

For training over streams $s>1$, let us consider $\mathcal{D} = \mathcal{D}^s_{\text{train}} \cup \mathcal{D}^s_{\text{mem}}$ as denoting the set of data for the given stream consisting of both the novel incoming stream and also the retained samples from the previous streams, $\mathcal{Y} = \mathcal{Y}^s_{\text{train}} \cup \mathcal{Y}^s_{\text{mem}}$ be their respective target labels. We assume that $\mathcal{Y}^s_{\text{train}} \cap \mathcal{Y}^s_{\text{mem}} = \phi$, that is, the incoming stream of data will contain no samples from any of the previously seen classes. We generate a curriculum for the incoming data stream based on the previous training phase's knowledge in the proposed work. We use the data prepared based on this curriculum produced for the incoming stream s along with the retained memory to train the network. The following

section discusses in detail the steps involved in the curriculum generation component.

C. Curriculum Generation

We introduce a training approach based on the curriculum designed for the stream s where $s > 1$, depending on the similarity of incoming classes to that of already learned classes. For a given stream s of data, we rely on the features extracted by the CNN, and then, utilize the feature from the penultimate d-dimensional fully connected layer corresponding to the samples from both $\mathcal{X}_s$ and $\mathcal{X}_{s-1}$. The per-class mean vector μ_i for each of these are calculated as

$$\mu_p^{s-1} = \frac{1}{N}\sum_{j=1}^{N} f_{s-1}(x_p^j) \tag{2}$$

$$\mu_q^{s} = \frac{1}{N}\sum_{j=1}^{N} f_{s}(x_q^j) \tag{3}$$

where p denotes the class index for the classes from previous data stream, while q denotes that for the current stream of data, that is, $p \in \{1 \ldots |\{\mathcal{Y}_{\text{train}}^{s-1}\}|\}$ and $q \in \{1 \ldots |\{\mathcal{Y}_{\text{train}}^{s}\}|\}$. x_i^j denotes the jth sample of the ith class and N denotes the number of samples in the class. $f_s(x_i^j)$ will denote the extracted feature of the corresponding sample of the stream s.

Now, we utilize these per class mean values corresponding to both current stream of data $\mathcal{D}_{\text{train}}^{s}$ and previous stream of data $\mathcal{D}_{\text{train}}^{s-1}$ to generate a $|\mathcal{Y}_{\text{train}}^{s-1}| \times |\mathcal{Y}_{\text{train}}^{s}|$ dimension relation matrix R. Each element of the matrix R that is r_{pq} will represent the cosine distance between the class mean vectors corresponding to the pth class of the previous stream and qth class of the current data stream.

$$r_{pq} = \frac{\mu_p^T \mu_q}{\|\mu_p\| \, \|\mu_q\|}. \tag{4}$$

The curriculum C_s corresponding to the current data stream, $\mathcal{D}_{\text{train}}^{s}$ is generated from the relation matrix R, which gives the similarity measure between the classes of the previous stream $s-1$ and current stream s as

$$C_s = \arg\max_p R \times e_q^T \tag{5}$$

where $p \in \{1, \ldots, |\mathcal{Y}_{\text{train}}^{s-1}|\}$ and $q \in \{1, \ldots, |\mathcal{Y}_{\text{train}}^{s}|\}$.

We select each column q (where e_q^T denotes the qth column of the identity matrix) corresponding to the previous classes. The maximum row index p gives the closest category in the new stream. Therefore, we get an array of indices corresponding to the curriculum order for the new data stream. The curriculum C_s, thus, obtained is then used to train the novel stream of data together with the retained samples through the pseudo-teacher–student approach. The following section discusses in detail the training algorithm for streams $s > 1$.

D. Curriculum-Based Incremental Learning

After generating the curriculum C_s for the new data stream s, we train the student network on the new set of data in the order specified by the curriculum. Simultaneously, we retain a small fraction of samples from the previous streams as $\mathcal{D}_{\text{mem}}^{s}$ for repetitive replay. We also use the knowledge distillation technique as proposed by [24] to transfer the knowledge gained until the $(s-1)^{th}$ stream from the teacher network to the student. Algorithm 1 describes the complete training process for streams $s > 1$.

Algorithm 1: The Training Procedure for a Stream of Data Incrementally ($s > 1$) Using the CILEA-NET Framework.

Input: $D = \{\mathcal{D}_{train}^{s}, \mathcal{D}_{mem}^{s}\}$
Output: Incrementally trained model $\Theta_s(\theta)$

```
Load the Teacher network Θ_{s-1}
f_s = Θ_{s-1}(D^s_train)
f_{s-1}= Θ_{s-1}(D^{s-1}_train) // Obtained before pruning
Generate Curriculum C_s // refer Section IV-C
while epoch < max epoch do
    Sample B ⊂ D = {D^s_train, D^s_mem}
    q = Θ_s(B)
    L_T(p, q) = L_C(p, q) + L_Dist(p, q)
    Update: θ = θ − α∇_θ L_T
end
Sample m% of the samples from D^s_train to create
    D^{s+1}_mem and prune the remaining
Finetune Θ using α_ft = 0.1 × α on D^{s+1}_mem
```

Unlike the traditional teacher–student approach, which uses a fixed teacher network to transfer information, we propose a unique pseudo-teacher–student approach. For every stream, $s > 1$, the student network from the previous stream serves as the new teacher, distilling the prior knowledge to the current student network while the network learns novel tasks.

We achieve knowledge transfer from the teacher to student network employing the method proposed in [24] using

$$p_i = \frac{\exp z_i/T}{\sum_j \exp z_j/T} \tag{6}$$

where p_i is the soft probability obtained by performing distillation over the logit z_i by comparing it with the other logits. T is called the temperature parameter. When $T = 1$, it acts like a normal softmax where the class with the highest score significantly influences the loss. When $T > 1$, the classes with comparatively lower scores also influence the loss and result in a more fine-grained representation. The value of T is kept equal to 2 as empirically obtained in [24] for optimal performance.

We utilize the incoming data arranged based on the curriculum and the retained data samples from memory to train the network. We train the network incrementally with the cost function as a combination of classification and knowledge distillation loss. We use the classical multiclass cross-entropy loss as the classification loss and the Kullback–Liebler divergence loss as the knowledge distillation loss function acting as the regularizer over distilled information from the teacher network to the student to ensure minimum forgetting. The cross-distilled loss function $\mathcal{L}_T$ is defined as

$$\mathcal{L}_T = \mathcal{L}_C + \mathcal{L}_{\text{Dist}}. \tag{7}$$

The distillation loss is applied to the old classes' classification layers, while the multiclass cross entropy is employed upon all classification layers.

$\mathcal{L}_C$ is the multiclass cross-entropy loss applied to the samples from both the new stream of data and also the previous data. This is the same as the loss function defined in (1).

While $\mathcal{L}_{\text{Dist}}$ is the knowledge distillation loss applied to the samples from the previous data streams and is defined as

$$\mathcal{L}_{\text{Dist}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} p_{ij}^{\text{dist}} \log q_{ij}^{\text{dist}}. \tag{8}$$

Here, again, N is the number of samples, and C denotes the total number of old classes. p_{ij}^{dist} is the ground truth and q_{ij}^{dist} denotes the softmax output obtained both raised to the power $\frac{1}{T}$ for the ith sample of the jth class. The learning phase is then followed by the fine tuning and memory update steps.

We obtain the image samples used for training at a specific stream from an unbalanced mix of both the newly acquired stream and the previously retained data. As the new stream's data dominate in number, the model tends to be biased toward the features corresponding to these classes. In this approach, we mitigate this undesirable effect of class imbalance by fine tuning the model on a balanced subset of samples from each class with a small learning rate. After fine tuning the model over a balanced subset of the samples, we retain only a fraction of the current stream's data by performing random selection and pruning the current stream's samples. $\mathcal{D}_{\text{mem}}^{s}$ denotes this memory and is used to remember the previously learned information via knowledge distillation.

We evaluate the model's performance over the data samples from the classes encountered until the current stream, after every training epoch. For instance, during the training phase over the k novel classes of the stream s along with the previous $(s-1)k$ classes from memory, the model performance is tested over the complete set of classes $1, 2, \ldots, sk$. This test data can be denoted as $\mathcal{D}_{\text{test}}^{s} = \bigcup_{i=0}^{s.k}\{\mathcal{X}_i^j\}_{j=1}^{T}$, where k indicates the incremental step size and T is the total number of samples per class used for inference. Also note that $\bigcap_{i=0}^{s.k}\{\mathcal{X}_i^j\}_{j=1}^{T} = \phi$.

V. Experiments

We present our results using three datasets: NWPU-RESISC45, Indian Pines, and Patternnet. Kindly note that the NWPU-RESISC45 dataset is used for ablation studies discussed in this work. We first compare the accuracy of the proposed approach with existing methods with incremental learning on these datasets. Then, we compare the performance of the curriculum-based learning approach to that of the noncurriculum-based learning approach by examining the model accuracy, the extent of forgetting, and the cumulative time of convergence for the entire training process. This comparison provides us with an insight into how curriculum learning helps us attain better results and faster convergence time. Finally, we show the effect of the memory size on the incremental learning framework by varying it from 5% to 30% in the steps of 5.

The following section will detail the datasets used and the preprocessing steps taken into consideration. Section V-B presents the implementation details and will address the hyperparameters used along with the evaluation metrics employed in this article. Section V-D talks about the existing works in incremental learning and the protocols adopted in training these frameworks for contrast. Finally, in Section V-E, we perform the ablation study to analyze the influence of specific components on the model performance.

A. Datasets

We show our results on two challenging large-scale optical remote sensing image datasets, namely NWPU-RESISC45, PatternNet, and one hyperspectral image dataset, Indian Pines, to establish the robustness of our proposed approach. We chose the optical image datasets as they are two very complex datasets with highly varying spatial resolutions and consist of a significant amount of background clutter. Simultaneously, the Indian Pines hyperspectral image dataset's evaluation ensures that the proposed model can also handle variations in the spectral content and perform well on land cover classification.

For each of the following datasets, the training/testing data were prepared by randomly forming a group of k classes from the total number of classes in the dataset. Each group corresponds to a specific stream of data. One of these groups will form the base set for training at stream $s = 1$ while the remaining streams will be fed incrementally for $s > 1$. This section also covers the preprocessing steps followed and the training-inference split for these datasets.

1) NWPU-RESISC45 [38]: This dataset is a publicly available benchmark for Remote Sensing Image Scene Classification (RESISC), created by Northwestern Polytechnical University (NWPU). This dataset consists of 31 500 images, covering 45 classes with 700 images in each category. We use an 80:20 ratio training–testing split with 560 images per class in training data and the remaining images for testing.

2) PatternNet [39]: PatternNet is a high-resolution remote sensing image dataset collected via Google Earth imagery or the Google Map API for remote sensing image retrieval. It consists of 38 classes with 800 images per class of dimension 256×256. The dataset was split into 560 images per class for training, retaining the remaining 240 images per class for the testing. Two classes, namely, airplane and baseball field, were omitted from the dataset to maintain a consistent number of classes per stream throughout the process.

3) Indian Pines [40]: Gathered by the AVIRIS sensor over the Indian Pines test site in North-Western Indiana, this dataset consists of 145×145 pixels and 224 spectral reflectance bands in the wavelength range of $0.4 - 2.5 \times 10^{-6}$ m. The Indian Pines scene contains 16 classes with varying numbers of data points in each class altogether. This dataset was made available by reducing the number of bands to 200.

To fit our requirement for learning to classify images incrementally using a curriculum and match the designed network, we further choose to keep only the three most representative channels out of 200 by performing principal-component analysis

over the reflectance bands. Followed by this, we extract 11×11 patches from the image and label every such patch with the ground truth label corresponding to its central pixel. Also, to balance the number of such images, we restrict the number of training patches to 500 samples per class and the number of testing patches to 100 samples per class. This derived data consists of 9600 image patches covering 16 classes with 600 image patches belonging to each class.

B. Implementation Details

We use the pretrained VGG-16 network as the backbone of the pipeline. We follow [41] for initializing the dynamic fully connected layers. We also use batch-norm [42] after every convolutional layer to ensure a minimal covariance shift in the model. The kernel size we use for the network is three, along with a stride of 1. Max-pooling layers utilize a 2×2 window with a stride of 2. The activation function used is ReLU, which is present following every convolution layer except for the last one. For optimization, we use Adam optimizer with a learning rate of 1×10^{-6} and weight decay of 1×10^{-4}. We resize each input image to 224×224 pixels and maintain the number of epochs for training as 40 per stream. The dimension d for the extracted feature is 128, and accordingly, we add a 128-dimensional fully connected layer to the architecture. The number of samples retained is fixed to 30% of the samples per class present in the training data. We use the Adam optimizer for the fine-tuning phase for optimization with a learning rate of 1×10^{-7} and is fine tuned for 30 epochs after training the model over each stream of data starting from $s = 2$. We used a 12-GB Nvidia GeForce GTX 1080 Ti graphics card to run the experiments.

C. Performance Evaluation Metrics

1) Average Accuracy: The average accuracy is calculated as the mean of accuracy values at each incremental step. At stream s, it is defined as, $A_s = \frac{1}{s-1} \sum_{j=2}^{s} a_j$. The higher the value of A_s better the performance. Note that we do not consider the accuracy of the first stream of data while computing the average accuracy as it does not represent incremental learning.

2) Forgetting measure [43]: It is defined as the difference between a task's maximum accuracy in the past and the current accuracy. This helps us estimate the forgetting happened in the model. We can quantify it for a task j after training for the task/stream s as, $f_j^s = \max_{l \in \{1,\ldots,s-1\}} a_{l,j} - a_{s,j} \; \forall j < s$. The average forgetting at the stream s is written as, $F_s = \frac{1}{s-1} \sum_{j=1}^{s-1} f_j^s$.

3) Time: We also analyze and compare the performance of our approach by considering the time factor involved in training the model. This is calculated as the sum of time taken for each incremental step to complete the training process.

D. Comparison With the Existing Literature

We compare the performance of our approach with multiple existing frameworks. First, we consider the learning without forgetting (LwF) approach, as implemented in [7]. Here, they make the CNN network classify previous classes similar to the new classes using knowledge distillation as regularization. Then, we consider the elastic weight consolidation-based approach proposed in [20]. This approach employs a quadratic penalty over the difference between learned parameters for the old and new classes. They utilized the Fisher information metric to obtain the diagonal weighting over the parameters for learned classes. We then compare ours with [25], which utilizes a replay and fine-tuning-based approach to reduce class imbalance and knowledge distillation for regularization. Similarly, we compare with [31], which is a replay-based approach in remote sensing. We also compare with [23], which uses the nearest mean classifier and a small fraction of data from the previous tasks. Comparison is also performed with [27], which proposes an intermediate expert to adapt the target model to the new task and [28], which retrieves the samples that are frequently conflicted. We also include our comparison with [29], which greedily stores samples in memory and trains a model from scratch and uses these samples during testing. Finally, we compare with [30], which increases the number of neural network experts under the Bayesian nonparametric framework.

We carried out experiments for comparing the results obtained from the proposed approach with different algorithms, which are as follows:

1) LwF refers to the implementation as performed in [7];
2) EWC refers to the work utilizing elastic weight consolidation carried out in [20];
3) E2E refers to the end-to-end incremental learning work proposed in [25];
4) iCaRL refers to the work proposed in [23];
5) RS refers to the replay-based work proposed in [31].

Please note that DR, MIR, GDUMB, and CN-DPM are also used for comparison, which refer to the works proposed in [27]–[30], respectively. We bestow on the extensive work presented in [44] and the code provided here[1] to replicate these results. We obtain the results for EWC by using the code found here.[2] E2E and LwF are obtained using the code from this link[3] by suitably changing the model parameters detailed as follows. We now discuss the implementation aspects of the algorithms mentioned previously and the results obtained from them to compare with our approach, as shown in Table I on the NWPU-RESISC45, PatternNet, and Indian Pines image datasets. We follow the same architectural and training protocols used in our method. The backbone model architecture is fixed as VGG-16 with batch normalization and dropout for all the algorithms. We also use identical hyperparameters to train the incremental learning model using our algorithm with the exemplar sample memory size fixed as 2000. We trained the model for 40 epochs using the Adam optimizer with a learning rate of 1×10^{-6} and weight decay of 1×10^{-4}. This training phase was followed by a balanced fine-tuning stage for 30 epochs, only if the algorithm in consideration requires a fine-tuning stage. Table I displays the results for a fixed step size of five classes, four classes, and six

[1][Online]. Available: https://github.com/RaptorMai/online-continual-learning

[2][Online]. Available: https://github.com/xialeiliu/RotateNetworks

[3][Online]. Available: https://github.com/kibok90/iccv2019-inc

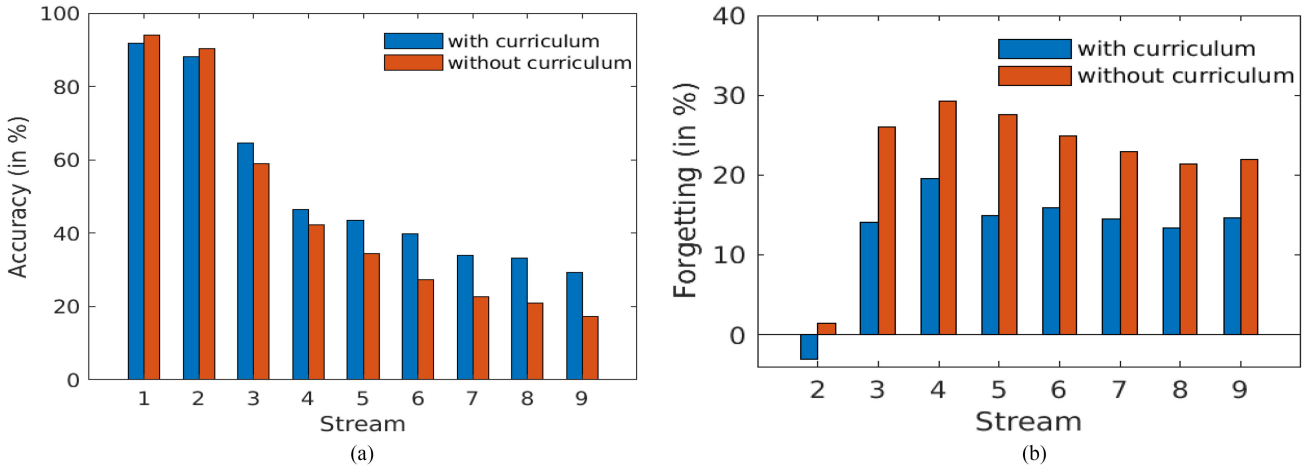

Fig. 3. Results for accuracy and forgetting at each stream for the NWPU-RESISC45 dataset. The results are shown for both with and without curriculum approach. (a) Accuracy per stream for the NWPU-RESISC45 dataset with a step size of 5. (b) Average forgetting per stream for NWPU-RESISC45 data with a step size of 5.

TABLE I
COMPARISON OF ACCURACY (IN %) ON NWPU-RESISC45, PATTERNNET, AND INDIAN PINES DATASET AS SHOWN, USING EXISTING APPROACHES

Method	Dataset		
	NWPU	Indian Pines	PatternNet
LwF [7]	30.17	54.76	29.79
EwC [20]	20.50	25.04	30.39
E2E [25]	29.40	55.33	27.08
DR [27]	22.09	-	-
iCaRL [23]	37.20	-	60.97
RS [31]	36.54	-	57.38
MIR [28]	27.95	-	53.82
GDUMB [29]	39.68	-	59.02
CN-DPM [30]	20.58	-	37.40
Proposed	**49.42**	**93.20**	**62.31**
w/o curriculum	39.31	81.98	53.04
w/o fine-tuning	38.40	-	-

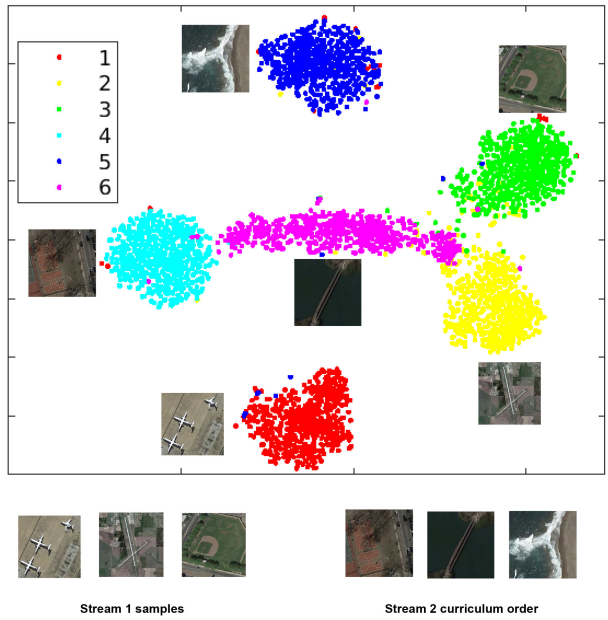

Fig. 4. Sample of the curriculum generated for the NWPU-RESISC45 dataset with step size 3. The tsne plot along with corresponding image samples is provided for reference.

classes per stream of data for NWPU-RESISC45, Indian Pines, and PatternNet datasets, respectively.

From the results presented in Table I, it is apparent that our approach outperforms the existing algorithms by a margin of 19% and about 31% for the NWPU-RESISC45 and PatternNet datasets, respectively, and by 37% for the Indian Pines dataset.

E. Ablation Study

In the proposed work, we explore the impact of adopting curriculum learning to improve the satellite image classification using an incremental learning framework. We expect the curriculum learning approach to improve the time taken to converge to an optimum by facilitating the learning process and reaching a better optimum. In the subsequent sections, we examine the curriculum-based and curriculum-less approach, where the latter is simply the CILEA-NET approach without the curriculum component. Also, to provide more clarity, we have tried to illustrate the curriculum generation for the NWPU-RESISC45 dataset for a step size of 3 as can be seen from Fig. 4.

1) Comparison of Convergence Time: We plot the time taken per stream to converge for both the curriculum-based and curriculum-less approaches to exhibit the improved convergence time. The time taken is estimated for the 40 epochs of training and incorporates the curriculum generation and fine-tuning stages throughout for uniformity. From Fig. 5, its evident that the CILEA-NET approach has the fastest convergence time in comparison with other methods. Cumulatively for the whole process, the CILEA-NET takes around only 44.30 h in contrast to the highest time taken by DR [27] of 182.09 h and 105.96 h when the curriculum is removed from the CILEA-NET framework. Table III presents the comparative results for both approaches

TABLE II
ABLATION RESULTS OF CILEA-NET IN THE PRESENCE AND ABSENCE OF CURRICULUM AND FINE-TUNING AS OBTAINED FOR THE NWPU-RESISC45 DATASET

Stream	Joint*	CILEA-NET		w/o curriculum		w/o fine-tuning	w/o curriculum and fine-tuning
	Accuracy (in %)	Accuracy (in %)	Forgetting (in %)	Accuracy (in %)	Forgetting (in %)	Accuracy (in %)	Accuracy (in %)
1		91.81	–	94.10	–	90.86	90.48
2		88.19	-3.05	90.41	1.43	69.84	62.92
3		64.67	14.08	59.02	26.01	51.05	49.05
4		46.48	19.56	42.30	29.27	41.35	36.34
5	82.54	43.49	14.96	34.35	27.50	37.16	29.59
6		39.80	15.88	27.28	24.89	32.05	25.53
7		33.91	14.46	22.77	22.91	28.76	22.32
8		33.12	13.33	21.02	21.43	25.06	19.30
9		29.35	14.68	17.38	21.87	21.2	16.43

*not a result obtained on incremental learning setting.

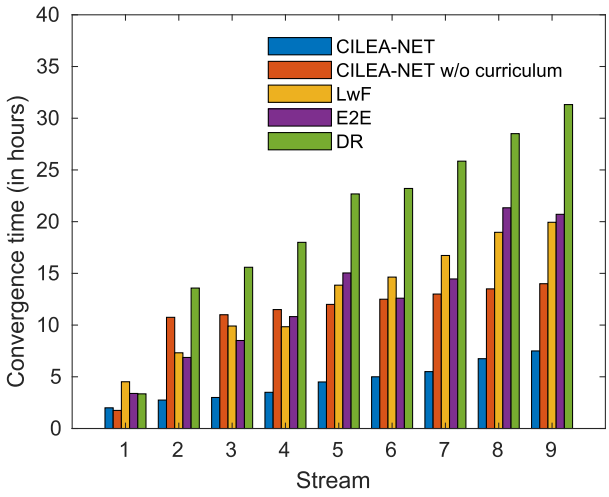

Fig. 5. Convergence time versus stream plot for comparing multiple methods. The result is shown for a step size of 5 on the NWPU dataset.

TABLE III
ACCURACY AND TOTAL TIME CONSUMED FOR DIFFERENT STEP SIZES FOR THE NWPU-RESISC45 DATASET WITH AND WITHOUT A CURRICULUM

Step size	Accuracy (in %)		Time taken (in hrs)	
	with curriculum	w/o curriculum	with curriculum	w/o curriculum
3	48.39	34.86	68.15	133.42
5	49.42	39.32	44.30	105.96
9	49.62	39.57	19.53	101.50

with various incremental step sizes for the NWPU-RESISC45 satellite image dataset. Also, from Fig. 5, it is evident that the CILEA-NET approach has the fastest convergence time in comparison with other methods. This depicts how introducing the curriculum learning technique helps in reducing the time taken to train the model.

2) Accuracy of the Model: To investigate how introducing a curriculum-based approach have an impact on achieving a better optimal solution. We examine the average accuracy obtained for both the curriculum-based and curriculum-less learning approaches for various incremental step sizes, as shown in Table III.

TABLE IV
ACCURACY AND TOTAL TIME CONSUMED FOR DIFFERENT STEP SIZES FOR THE INDIAN PINES DATASET WITH AND WITHOUT A CURRICULUM

Step size	Accuracy (in %)		Time taken (in hrs)	
	with curriculum	w/o curriculum	with curriculum	w/o curriculum
2	37.71	37.31	15.06	17.26
4	93.20	81.98	8.53	9.65
8	70.67	57.13	6.55	7.02

For the NWPU-RESISC45 dataset, we divided the 45 classes in three different ways, yielding incremental step sizes of 3, 5, and 9 classes per stream. In all these cases, we can observe from Table III that the curriculum-based approach depicts better performance with a margin of approximately 10% as opposed to that of the curriculum-less method.

Likewise, we train on the Indian Pines dataset using three different incremental step sizes of 2, 4, and 8 classes per stream. For all these individual cases, we can observe from Table IV that the approach integrated with curriculum learning exhibits better performance in classification of the hyperspectral image patches when confronted with that which does not and depicts that the former facilitates faster convergence of the algorithm.

Even though we see from Table II that the accuracy for the joint training is better as expected. Please note that this is only a result of training the network with all the classes together in a traditional fashion. This procedure is seldom relevant in practical scenarios.

3) Extent of Forgetting: We have adopted the forgetting measure to analyze the proposed approach's performance on the NWPU-RESISC45 dataset, as shown in Fig. 3(b). It is apparent from both the average forgetting and the average accuracy per stream for the given step size that the proposed curriculum-based approach can mitigate forgetting more efficiently. Table II depicts the per-stream average accuracy and average forgetting values for the said dataset; we can observe that for our approach, the forgetting is more limited by an average margin of 8.93%.

4) Effect of the Memory Size: Here, we analyze the change in performance of the model with the variation in per-class memory. From Table V, it is evident that with the reduction in the number of samples retained per class, the performance deteriorates. This trend is consistent with the expected behavior

TABLE V
ACCURACY OF DIFFERENT PER CLASS MEMORY SIZES FOR NWPU-RESISC45 AND INDIAN PINES DATASET FOR A STEP SIZE OF NINE CLASSES PER STREAM AND FOUR CLASSES PER STREAM, RESPECTIVELY

Per class memory (in %)	NWPU-RESISC45	Indian Pines
5	28.57	67.06
10	39.01	90.85
15	40.94	91.91
20	50.21	86.92
25	46.74	94.38
30	49.62	93.20

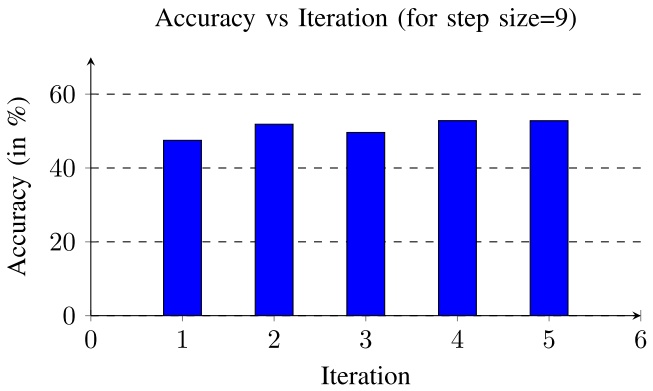

Fig. 6. Performance of the proposed approach across five different iterations with a random set of classes in each stream of every iteration.

as the number of samples used for memorizing the previous classes is reduced.

Nevertheless, the performance decline for the proposed approach with change in the per-class memory remains within 4% for the NWPU dataset. It is within 8% for the Indian Pines dataset during the first 5–10% reduction in the per-class memory. Whereas, when the amount of decrease in samples retained varies from 30 to 10% accounting to a 20% drop in memory utilization, the accuracy declines by a margin of 11% and 4% for the NWPU-RESISC45 and Indian Pines datasets, respectively (we do not consider the intermediate outlier result for 20% memory for this calculation).

5) Order of Class Acquisition: To verify the objectivity of the model performance toward the order of acquisition of the classes, we conduct experiments by training the model using multiple arbitrary orderings of the classes, i.e., for every such investigation, the set of classes present in a given stream i will be unique. Fig. 6 presents the results obtained for five separate experiments on the NWPU-RESISC45 dataset. Therefore, we establish that the performance of the proposed curriculum-based approach is consistent across any order of class acquisition.

6) Fine Tuning: From the result in Table II, the fine-tuning phase integrated into the training framework plays a crucial role in deciding the performance of the model. The absence of fine-tuning will lead to degraded performance due to class imbalance,

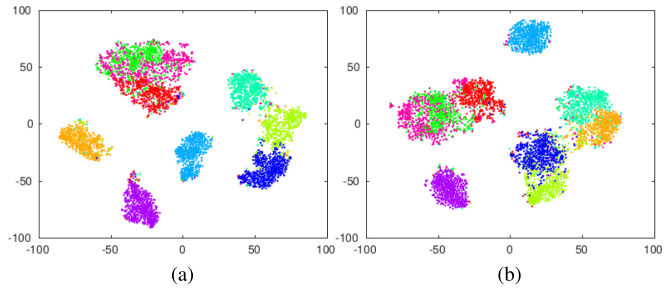

Fig. 7. Tsne plot comparison between the decision boundaries learned for (a) CILEA-NET approach and (b) approach with the curriculum removed. The result was generated for an incremental step size of 3 for the NWPU-RESISC45 dataset. Different colors represent different classes.

accounting for a drop in the performance by about 11% resulting in an inferior overall accuracy of 38.30% as seen in Table I.

7) Comparison of Decision Boundaries: To understand how the curriculum learning concept helps attain a better optimum and learn a finer decision boundary, we visualize both the curriculum-based and curriculum-less approaches using the tsne plot, as seen in Fig. 7. We can observe that the curriculum-based incremental learning approach guarantees better interclass separation than the largely overlapping decision boundaries, as seen in the curriculum-less approach. From this observation, we may conclude that the curriculum learning approach's usage enforces the model to learn faster and better by sequentially encountering samples from simple to complicated and discover a better decision boundary.

VI. CONCLUSION

This article presents CILEA-NET, a novel curriculum learning-driven framework for class incremental learning for remote sensing image classification. Incremental learning is considered essential in remote sensing, given the continuous acquisition of images with novel land-cover classes. In this regard, we tackle two critical issues of the incremental learning setup.

1) How to deal with the stability/plasticity trade-off?
2) How to ensure quick learning for new classes?

To this end, within our proposed framework, we suitably utilize concepts from knowledge distillation and curriculum learning. Thorough experiments confirm the superiority of the proposed model over several recent state-of-the-art approaches for different types of remote sensing data. Thus, in a generic sense, one can utilize the curriculum learning paradigm along with any existing approaches in a deep learning framework to boost the performance of the network. We are presently interested in extending this model for few-shot incremental learning, where we assume that only a few training samples are available for the classes. Given the issues of annotating remote sensing data, such a few-shot paradigm will help the community.

ACKNOWLEDGMENT

B. Banerjee dedicates this to M. M. Banerjee.

REFERENCES

[1] A. S. Belward and J. O. Skøien, "Who launched what, when and why; trends in global land-cover observation capacity from civilian Earth Observation Satellites," ISPRS J. Photogrammetry Remote Sens., vol. 103, pp. 115–128, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924271614000720

[2] A. Gepperth and B. Hammer, "Incremental learning algorithms and applications," in *Proc. Eur. Symp. Artif. Neural Netw.*, 2016.

[3] M. Mermillod, A. Bugaiska, and P. Bonin, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," *Front. Psychol.*, vol. 4, p. 504, 2013.

[4] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends Cogn. Sci.*, vol. 3, no. 4, pp. 128–135, 1999.

[5] A. Robins, "Catastrophic forgetting in neural networks: The role of rehearsal mechanisms," in *Proc. IEEE 1st New Zealand Int. Two-Stream Conf. Artif. Neural Netw. Expert Syst.*, 1993, pp. 65–68.

[6] G. I. Parisi, J. Tani, C. Weber, and S. Wermter, "Lifelong learning of spatiotemporal representations with dual-memory recurrent self-organization," *Front. Neurorobot.*, vol. 12, p. 78, 2018.

[7] Z. Li and D. Hoiem, "Learning Without Forgetting," *IEEE Trans. Pattern Ana. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[8] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. Mach. Learn. Res.*, vol. 70, 2017, Art. no. 3987.

[9] A. A. Rusu *et al.*, "Progressive neural networks," 2016, *arXiv:1606.04671*.

[10] Y. Wu *et al.*, "Incremental classifier learning with generative adversarial networks," 2018, *arXiv:1802.00853*.

[11] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. ACM 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.

[12] B. F. Skinner, "Reinforcement today," *Amer. Psychol.*, vol. 13, no. 3, p. 94, 1958.

[13] K. A. Krueger and P. Dayan, "Flexible shaping: How learning in small steps helps," *Cognition*, vol. 110, no. 3, pp. 380–394, 2009.

[14] Y.-M. Wen and B.-L. Lu, "Incremental learning of support vector machines by classifier combining," in *Advances in Knowledge Discovery and Data Mining*, Z.- H. Zhou, H. Li, and Q. Yang, Eds. Berlin, Germany: Springer, 2007, pp. 904–911.

[15] L. Bruzzone and D. F. Prieto, "An incremental-learning neural network for the classification of remote-sensing images," *Pattern Recognit. Lett.*, vol. 20, no. 11–13, pp. 1241–1248, 1999.

[16] B. Lakshminarayanan, D. M. Roy, and Y. W. Teh, "Mondrian forests: Efficient online random forests," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3140–3148.

[17] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 67–82.

[18] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7765–7773.

[19] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," 2018, *arXiv:1801.01423*.

[20] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," in *Proc. Nat. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, 2017.

[21] X. Liu, M. Masana, L. Herranz, J. Van de Weijer, A. M. Lopez, and A. D. Bagdanov, "Rotate your networks: Better weight consolidation and less catastrophic forgetting," in *Proc. IEEE 24th Int. Conf. Pattern Recognit.*, 2018, pp. 2262–2268.

[22] H. Jung, J. Ju, M. Jung, and J. Kim, "Less-forgetful learning for domain expansion in deep neural networks," 2017, *arXiv:1711.05959*.

[23] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2001–2010.

[24] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[25] F. M. Castro, M. J. Marin-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 233–248.

[26] O. Tasar, Y. Tarabalka, and P. Alliez, "Incremental learning for semantic segmentation of large-scale remote sensing data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3524–3537, Sep. 2019.

[27] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Lifelong learning via progressive distillation and retrospection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 437–452.

[28] R. Aljundi *et al.*, "Online continual learning with maximal interfered retrieval," in *Proc. Adv. Neural Inf. Process. Syst.* 32, 2019, pp. 11 849–11 860. [Online]. Available: http://papers.nips.cc/paper/9357-online-continual-learning-with-maximal-interfered-retrieval.pdf

[29] A. Prabhu, P. Torr, and P. Dokania, "Gdumb: A simple approach that questions our progress in continual learning," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020.

[30] S. Lee, J. Ha, D. Zhang, and G. Kim, "A neural Dirichlet process mixture model for task-free continual learning," 2020, *arXiv:2001.00689*.

[31] N. Ammour, Y. Bazi, H. Alhichri, and N. Alajlan, "Continual learning approach for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, to be published, doi: 10.1109/LGRS.2020.3019071.

[32] H. Li *et al.*, "Clrs: Continual learning benchmark for remote sensing image scene classification," *Sensors*, vol. 20, no. 4, 2020, Art. no. 1226.

[33] N. Yang and H. Tang, "Geoboost: An incremental deep learning approach toward global mapping of buildings from VHR remote sensing images," *Remote Sens.*, vol. 12, no. 11, 2020, Art. no. 1794.

[34] J. L. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, vol. 48, no. 1, pp. 71–99, 1993.

[35] E. Allgower and K. Georg, "Simplicial and continuation methods for approximating fixed points and solutions to systems of equations," *SIAM Rev.*, vol. 22, no. 1, pp. 28–85, 1980.

[36] G. Hacohen and D. Weinshall, "On the power of curriculum learning in training deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2535–2544.

[37] X. Yao, X. Feng, J. Han, G. Cheng, and L. Guo, "Automatic weakly supervised object detection from high spatial resolution remote sensing images via dynamic curriculum learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 675–685, Jan. 2021.

[38] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.

[39] W. Zhou, S. Newsam, C. Li, and Z. Shao, "Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval," *ISPRS J. Photogrammetry Remote Sens.*, vol. 145, pp. 197–209, 2018.

[40] M. F. Baumgardner, and L. L. Biehl, and D. A. Landgrebe, "220 band AVIRIS hyperspectral image data set: Jun. 12, 1992 Indian Pine Test Site 3," Sep. 2015. [Online]. Available: https://purr.purdue.edu/publications/1947/1

[41] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.

[43] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 532–547.

[44] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner, "Online continual learning in image classification: An empirical survey," 2021, *arXiv:2101.10423*.

S. Divakar Bhat received the bachelor's degree in electrical engineering from Model Engineering College, Kochi, India, in 2017. He is currently working toward the master's degree with the Department of Electrical Engineering, Indian Institute of Technology Bombay (IIT Bombay), Mumbai, India.

He works as a Project Research Assistant under Prof. S. Chaudhuri with the Vision and Image Processing Lab, IIT Bombay. His main research interests include machine learning, pattern recognition, computer vision, continual learning, and metalearning techniques.

Biplab Banerjee (Member, IEEE) received the M.E. degree in computer science and engineering from Jadavpur University, Kolkata, India, in 2010, and the Ph.D. degree in image analysis from the Indian Institute of Technology Bombay (IIT Bombay), Mumbai, India, in 2015.

He was a Postdoctoral Researcher with the University of Caen Basse-Normandy, France, and the Istituto Italiano di Technologia Genova, Italy. He then worked as an Assistant Professor with the Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, India, between 2016 and 2018. Since June 2018, he has been working as an Assistant Professor (from June, 2018 onward) in machine learning and visual computing with the Centre of Studies in Resources Engineering, and is an Associate Member with the Center of Machine Intelligence & Data Science, IIT Bombay. His research interests include zero-shot learning, metalearning, multitask learning, domain adaptation and transfer learning, multimodal analysis of remote sensing data, deep reinforcement learning, etc.

Subhasis Chaudhuri (Fellow, IEEE) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kharagpur, Kharagpur, India, in 1985, the M.Sc. degree in electrical engineering from the University of Calgary, Calgary, Canada, in 1987, and the Ph.D. degree in electrical engineering from the University of California San Diego, La Jolla, CA, USA, in 1990.

He is currently serving as the Director of the Indian Institute of Technology Bombay, Mumbai, India, where he was a former KN Bajaj Chair Professor with the Department of Electrical Engineering, and was also the Head of the Department from 2005 to 2008 and the Dean (International Relations) from 2009 to 2013. His main research interests include pattern recognition, image processing, computer vision, and haptics.

Dr. Chaudhuri is a Fellow of the science and engineering academies in India. He was the recipient of the Bhatnagar Prize in Engineering Sciences in 2004 and the GD Birla Award in 2011.

Avik Bhattacharya (Senior Member, IEEE) received the integrated M.Sc. degree in mathematics from the Indian Institute of Technology, Kharagpur, India, in 2000, and the Ph.D. degree in remote sensing image processing and analysis from Télécom ParisTech, Paris, France, and the Ariana Research Group, Institut National de Recherche en Informatique et en Automatique (INRIA), Sophia Antipolis, Nice, France, in 2007.

He is currently an Associate Professor with the Centre of Studies in Resources Engineering, Indian Institute of Technology Bombay (IITB), Mumbai, India, where he is also leading the Microwave Remote Sensing Lab. Before joining the IITB, he was a Canadian Government Research Fellow with the Canadian Centre for Remote Sensing, Ottawa, ON, Canada. He received the Natural Sciences and Engineering Research Council of Canada Visiting Scientist Fellowship with the Canadian National Laboratories, from 2008 to 2011. His current research interests include synthetic aperture radar (SAR) polarimetry, statistical analysis of polarimetric SAR images, and applications of radar remote sensing in agriculture, cryosphere, urban, and planetary studies.

Dr. Bhattacharya is the Editor-in-Chief for the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS. He was an Associate Editor for the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS. He has been the Guest Editor of the special issue on Applied Earth Observations and Remote Sensing in India in IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, 2017. He was one of the guest editors of the special stream on Advanced Statistical Techniques in SAR Image Processing and Analysis for the IEEE GEOSCIENCE AND REMOTE SENSING LETTER, 2018. He is the Founding Chairperson of the IEEE Geoscience and Remote Sensing Society Chapter of the Bombay Section.