Remote Sensing Based Crop Type Classification Via Deep Transfer Learning

Krishna Karthik Gadiraju and Ranga Raju Vatsavai

***Abstract*—Machine learning methods using aerial imagery (satellite and unmanned-aerial-vehicles-based imagery) have been extensively used for crop classification. Traditionally, per-pixel-based, object-based, and patch-based methods have been used for classifying crops worldwide. Recently, aided by the increased availability of powerful computing architectures such as graphical processing units, deep learning-based systems have become popular in other domains such as natural images. However, building complex deep neural networks for aerial imagery from scratch is a challenging affair, owing to the limited labeled data in the remote sensing domain and the multitemporal (phenology) and geographic variability associated with agricultural data. In this article, we discuss these challenges in detail. We then discuss various transfer learning methodologies that help overcome these challenges. Finally, we evaluate whether a transfer learning strategy of using pretrained networks from a different domain helps improve remote sensing image classification performance on a benchmark dataset. Our findings indicate that deep neural networks pretrained on a different domain dataset cannot be used as off-the-shelf feature extractors. However, using the pretrained network weights as initial weights for training on the remote sensing dataset or freezing the early layers of the pretrained network improves the performance compared to training the network from scratch.**

***Index Terms*—Agriculture, crop classification, deep learning, remote sensing, transfer learning.**

I. Introduction

REMOTE sensing image classification is an active area of research for the past two decades in areas such as agriculture, national security, poverty mapping, and disaster management. More specifically, research in the agricultural domain has focused on areas such as crop classification, crop health monitoring, and crop yield prediction. In this article, our focus is on crop classification. Crop classification serves as an essential step in estimating the crop area coverage as well as an initial step for crop yield prediction. As described in [1], accurate mapping of crops is important to several stakeholders such as farmers and key policy makers in the government. As described in [2], accurate agricultural estimates have important economic impacts. In addition, accurate mapping of crops is essential from a food security perspective, since it allows governments to make strategic plans to sustain the growing world population.

Several works have addressed the task of crop mapping. These methods have evolved over time, depending upon the available computing power, advances in machine learning methods and the spatial and temporal resolutions of the aerial imagery. For example, improving spatial (such as the data provided by National Agricultural Imagery Program (NAIP, 1-m spatial resolution) [3] and temporal resolutions and availability of graphical process units (GPU)-based computing has allowed for the usage of deep learning methods in the domain of agriculture. However, building deep learning solutions for the domain of remote sensing has its own challenges. In the following section, we first highlight the common challenges existing in building accurate crop maps using satellite imagery. Then, we discuss how while some of the challenges can be overcome using deep learning, other challenges may still remain.

A. Crop Classification With Remote Sensing Imagery: Challenges

The challenges associated with performing crop classification using remote sensing imagery can be broadly categorized into the following three sections: domain, data, and methodology-based challenges. We describe each of these sections in detail as follows.

1) Domain Challenges: Classification of crops has several domain specific challenges. For instance, high variability can exist in agricultural data due to differences in terrain, topology, weather, soil properties, crop health, noise due to the presence of other classes such as cloud cover or built up area, and time of acquisition of the image. Fig. 1 shows some examples of variations caused due to crop health, date of acquisition, and the presence of other classes in the images. Depending upon the date of acquisition of a satellite image with respect to the growth cycle of a crop, images belonging to the same crop may look vastly different, while images belonging to different crops may look vastly similar. This results in *high interclass similarity* and *low intraclass similarity*.

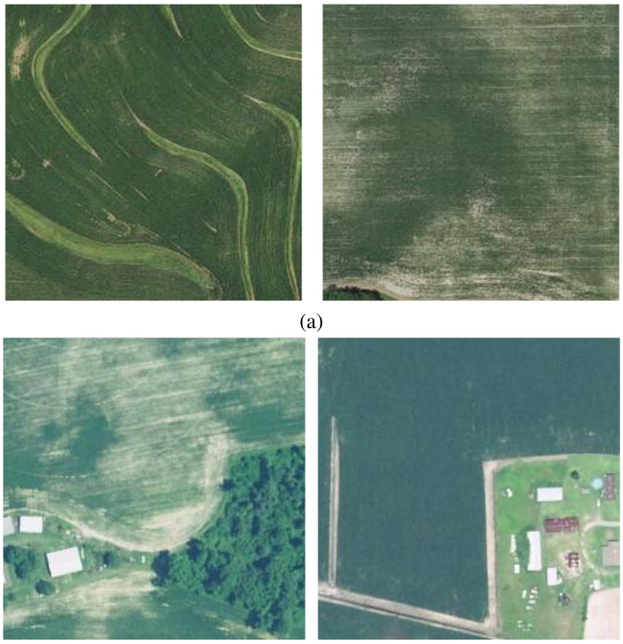
2) Data Challenges: Data challenges are twofold described as follows.

1) *Lack of large labeled data:* While other domains such as natural images have large labeled datasets such as ImageNet [4], despite petabytes of data being collected regularly using a variety of sensors, the remote sensing domain has limited labeled data. As described in [5], acquisition of labeled data is a challenging operation.

Manuscript received 1 June 2022; revised 14 October 2022 and 10 February 2023; accepted 7 April 2023. Date of publication 25 April 2023; date of current version 29 May 2023. (*Corresponding author: Krishna Karthik Gadiraju.*)

The authors are with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695-7001 USA (e-mail: kkg.research23@gmail.com; rrvatsav@ncsu.edu).

Digital Object Identifier 10.1109/JSTARS.2023.3270141



(a)

(b)

Fig. 1. Variability introduced in data. (a) Randomly selected imagery for the Corn class from Iowa and Illinois. (b) Randomly selected imagery for the Corn class from North Carolina for 2016.

2) *Dependence on data due to high variability in data sources:* This high variability can occur due to a variety of conditions such as follows:
   a) depending upon when the images are acquired, one can observe variability due to changes in weather creating noisy conditions such as cloud cover;
   b) depending upon the properties/features of the collection medium such as the spatial and temporal resolutions of the satellites/unmanned aerial vehicle (UAV) devices used to collect the data, and the difference in the type of on-board sensors.

Traditionally, satellites with lower spatial resolution such as Moderate Resolution Imaging Spectrometer (MODIS, 250-m spatial resolution) [6] and LANDSAT (30-m spatial resolution) have been used to map the crops around the world [7], [8]. The challenge with classification of data using coarse resolution imagery is that it is purely dependent on the spectral information, since the size of an object will be significantly lower than the size of a pixel. As a result, most historical efforts for mapping crops across the world [9], [10], [11], [12] have focused on per-pixel classification efforts. In contrast, very high spatial resolution (VHR) multispectral imagery such as the data provided by National Agricultural Imagery Program (NAIP, 1-m spatial resolution) [3], allows for higher spatial information. The high spatial resolution allows individual objects within a region to be distinguished, and the size of a pixel is much lesser than the size of an object. More specifically, in the area of crop classification, high spatial resolution allows for the development of fine grained crop maps. In particular, in Asian countries where farm sizes can be small in area, high-resolution images are critical for identifying crop types. We discuss more about the various methodologies used for classifying remote sensing imagery and their challenges in the next section.

3) Methodology Challenges: Image classification efforts using aerial imagery for crop classification are either pixel-based [13], [14], [15], [16], object-based [17], [18], [19], neighborhood-based [20], [21] or patch-based [22], [23] approaches. The main drawback of pixel-based approaches is their inability to capture the spatial autocorrelation of neighboring pixels and often require postprocessing steps to improve classification. In addition, they are computationally inefficient when dealing with VHR imagery. In contrast, neighborhood-based and object-based approaches take into consideration the spatial relationship between neighboring pixels. Neighborhood-based approaches typically use methods such as Markov random fields (MRFs) and are computationally expensive when using VHR imagery. Object-based approaches typically identify individual objects from images using image segmentation methods followed by classifying these individual objects based on their properties. Object-based approaches are dependent on the efficiency of the segmentation and are more popular for detecting objects such as buildings and roads. Next, we have the patch-based approaches such as Bag of Visual Words (BOW)-based approaches. However, certain BOW-representations also ignore the relationships between the words in the bag, which may prove crucial in improving the predictive performance. Finally, a majority of the aforementioned approaches require the generation of additional features [24], [25] to achieve optimal predictive performance. The authors in [26] and [27] provide a detailed overview over the various hand-crafted features used for remote sensing image classification.

Classification of imagery using deep neural networks such as convolutional neural networks (CNNs) also falls into the category of patch-based approaches. As described in [28], CNNs are deep neural networks that are designed to work on data that has a grid-like structure such as time series, images, and videos. When considering images (2-D), in contrast to traditional neural networks, which acts on each pixel of the image, CNNs act on the neighborhood of each pixel. This is done using linear operations (convolutions). The outputs of the convolution operations are known as feature maps. As a result, CNNs are better able to capture neighborhood relationships in contrast to traditional per-pixel, object-based and BOW-based classifiers. In addition, CNNs simulate the features perceived by the human brain by detecting high-level task-relevant features as a function of low-level feature representations. In other words, they can perform representation learning. This ability allows CNNs to extract hierarchical semantic features relevant to a particular task [29]. As a result, CNNs have outperformed traditional image classification methods in domains such as natural images. Deep CNNs such as the VGG network [30] from the Visual Geometry Group (VGG) from the University of Oxford, GoogleNet [31], Residual Network (ResNet) [32], and DenseNet [33] have demonstrated significant improvements in performance in the domain of natural images. While deep CNNs have since evolved to solutions in the temporal (1-D-CNNs) and spatiotemporal

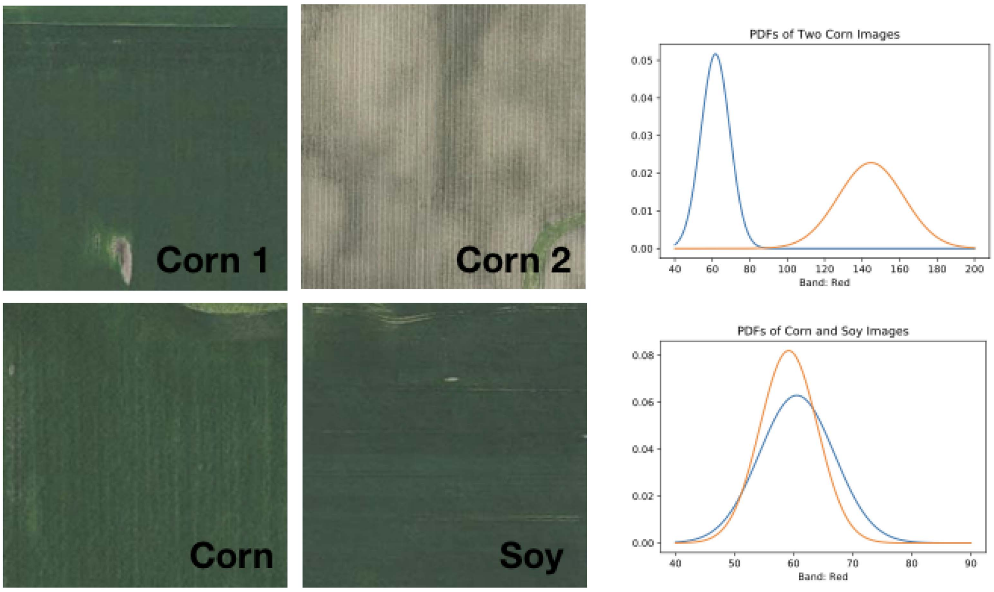


Fig. 2. Low interclass similarity observed in the probability distributions (PDFs) between two different corn images and high intraclass similarity between corn and soy images.

domains (3-D-CNNs), in this article, we specifically study CNNs from the 2-D perspective.

However, even while using deep neural networks such as CNNs, the following challenges exist.

1) Deep CNNs require high spatial resolution imagery. Popular remote sensing sources used for crop classification such as MODIS (250 m) have coarse spatial resolution. Coarse spatial resolution leads to the aggregation of visual information such as combination of spectral information of built-up areas and vegetation. This hinders the ability of CNNs to find low-level features that are relevant to the task.
2) Supervised deep neural networks need to be trained on large datasets for optimal performance, which is a challenge due to the limited labeled data available in the remote sensing domain. While labeled data exist at high resolution scale in developed countries, such data are not available in developing countries. In addition, most existing datasets are typically small or medium scale in size.
3) Deep neural networks also suffer from the variability challenge arising due to high interclass and low intraclass similarity. This, when combined with the limited data, becomes a bigger challenge.
4) Finally, deep learning has a high computational requirement.

The limitation of coarse resolution imagery has been overcome in the recent past due to the availability of VHR imagery. Computational challenges can be overcome using high performance computing (HPC) systems and GPUs. As described in [29], the variability due to a variety of factors (such as the ones described in the aforementioned domain and data challenges) can cause a shift in the class distributions between training and test data. This is contrary to the assumption of traditional machine learning approaches that training and test data have similar distributions. For instance, depending upon the geographical location, climate and weather conditions, and the acquisition time of the remote sensing imagery, images of the same crop can appear to be vastly different. Similarly, images of different crops can appear vastly similar. This is clearly seen in Fig. 2, where there is a clear distinction in the marginal distributions of the two corn images, while the corn and soy images have a very similar distribution. As a result, traditional machine learning approaches demonstrate a poor performance when there is a shift in domain. Donahue et al. [34] demonstrated that deep neural networks are impacted when such a domain shift occurs. While it can be argued that with sufficiently large training data, one can build models that can sufficiently capture the variabilities, as discussed earlier, there is lack of large labeled data in the remote sensing domain. Another solution is to build temporal methods that can capture different stages of the crop growth period (crop phenology). In this case, classification will be based on differentiating the growth patterns of different crops. However, as described in [35], these critical patterns may change with changing seasons, geographical locations, and a variety of other conditions and choosing the relevant time periods is a challenge. Purely temporal solutions also tend to be per-pixel based and as a result cannot capture the spatial autocorrelation between neighboring pixels remote sensing data. Spatiotemporal solutions using methods such as 3-D CNNs [36] that can capture both the spatial and temporal relationships, which can perform better. However, not all remote sensing sources have both high spatial and temporal resolutions, or sufficiently large amount of training data to achieve high classification accuracy. Finally, another solution is to build local models to capture the specific properties in a geospatial location. However, even this approach can quickly become computationally expensive when operating at a national or a global scale.

In this article, we focus on transfer learning as a means to overcome the challenge of limited labeled data. We study transfer learning in detail in Section III. We describe our contributions in the next section.

II. Our Contributions

In this article, we offer the following contributions.

1) We provide a detailed overview of the various transfer learning strategies to overcome the challenge of limited labeled data in the remote sensing community in Section III.
2) We evaluate several finetuning strategies in the context of deep neural networks and identify optimal strategies. A detailed overview of these methods can be seen in Section V.
3) We created a large scale crop imagery benchmark dataset to evaluate various aspects of transfer learning. A detailed description of the dataset is included in Section V-A.

The rest of this article is organized as follows. In Section III, we provide a brief background of transfer learning and discuss the various transfer learning methods described in the literature. This is followed by a description of the methodology used in this article in Section IV, and the experiments and results are presented in Section V. Finally Section VI conclude this article.

III. Related Work

Machine learning methods for remote sensing image classification have evolved with the improving spatial and temporal resolutions of available imagery. We discussed the limitations of traditional approaches such as pixel-based, neighborhood-based, and object-based approaches in the previous section. We also discussed the limitations of certain patch-based approaches such as the BOW-based methods.

As described earlier, deep neural networks, such as deep CNNs on the other hand, can perform representation learning by detecting hierarchical task-relevant features as a function of low-level features such as edges. We described in detail the challenges associated with using deep neural networks for performing remote sensing image classification in Section I-A3.

Previously, methods such as greedy layer-wise unsupervised pretraining were popular to tackle the challenge of limited labeled data. Greedy layer-wise unsupervised pretraining trains each layer (such as autoencoders or restricted Boltzmann machines, which can learn latent representations [28]) of a network on unlabeled data before adding the next layer. After the unsupervised training, finetuning is performed by training all the layers using a supervised learning method on the limited labeled data. As described in [27], the pretraining can serve as a regularizer, and the learned parameters from pretraining can be good initial parameters for the supervised learning step. In [37], the authors learn sparse representations of satellite images by using greedy layer-wise unsupervised pretraining together with the Enforcing Population and Lifetime Sparsity algorithm. The trained network is used as a feature extractor and combined with a simple classifier. Liang et al. [38] use greedy layer-wise unsupervised training to train the nonpenultimate layers of a stacked denoising autoencoder, followed by supervised training on labeled data. However, as described in [28], solutions such as transfer learning, Bayesian learning, and deep CNNs have overshadowed the greedy layer-wise unsupervised pretraining effort. As discussed earlier, in this article, we focus on transfer learning approaches.

Transfer learning is one possible solution to the challenges of limited labeled data and variability. As described in [39], given a source domain S and its learning task L_S, a target domain T and its learning task L_T, transfer learning methods allow for the improvement of performing learning tasks on T using the knowledge gained from S. Pan and Yang [39] categorize transfer learning as inductive, transductive, or unsupervised depending upon the similarities/dissimilarities between the two domains, the two tasks and the availability of labeled data in the source and target tasks. Some examples of domain dissimilarity include differences in the marginal distributions or the feature space. Some examples of task dissimilarities include differences in the classification labels. Traditional methods such as Random Forests were also used previously for pixel-based classification of crops [40], where a Random Forest model trained on multi-temporal NDVI data from one location is used to evaluate the performance at a different location. However, in [40], it should be noted that both the source and target domains, as well as source and target tasks are the same. Our focus in this work is based on transfer learning for patch-based classification using deep neural networks.

While other methods such as self-supervised learning [41], [42] and zero-shot learning have also overcome the limited labeled data challenges, our focus in this article will be on transfer learning methods. We refer the readers to [43] for further reading on these methods. It should be noted here that some methods described in the following may fall under several categories, and they are not necessarily mutually exclusive, and can be used together as well. A comprehensive review of machine learning methods to tackle the limited labeled data and variability challenges is beyond the scope of this article. We will primarily focus on deep transfer learning methodologies. More specifically, we will focus on the scenarios where labeled data exist either in the source or target domains or both.

1) *Inductive transfer learning* [39]: In inductive transfer learning, the target and source learning tasks are different but related, while the source and target domains may or may not remain the same. Methods such as finetuning in deep neural networks and multitask learning fall under this category.

   The concept of supervised finetuning can be observed in Fig. 3. In *finetuning* deep neural networks such as CNNs, the assumption is that a deep neural network trained on a sufficiently large source dataset would learn features that are transferable to analyzing another target dataset when the source and target domains are similar to each other. This is because, earlier layers in a CNN learn generalized features [44] (such as edges) and the later layers learn more advanced task-specific features. While using the pretrained model and performing training on the target dataset, the layers extracted from the pretrained model can either be trained again, or we can fix the parameters

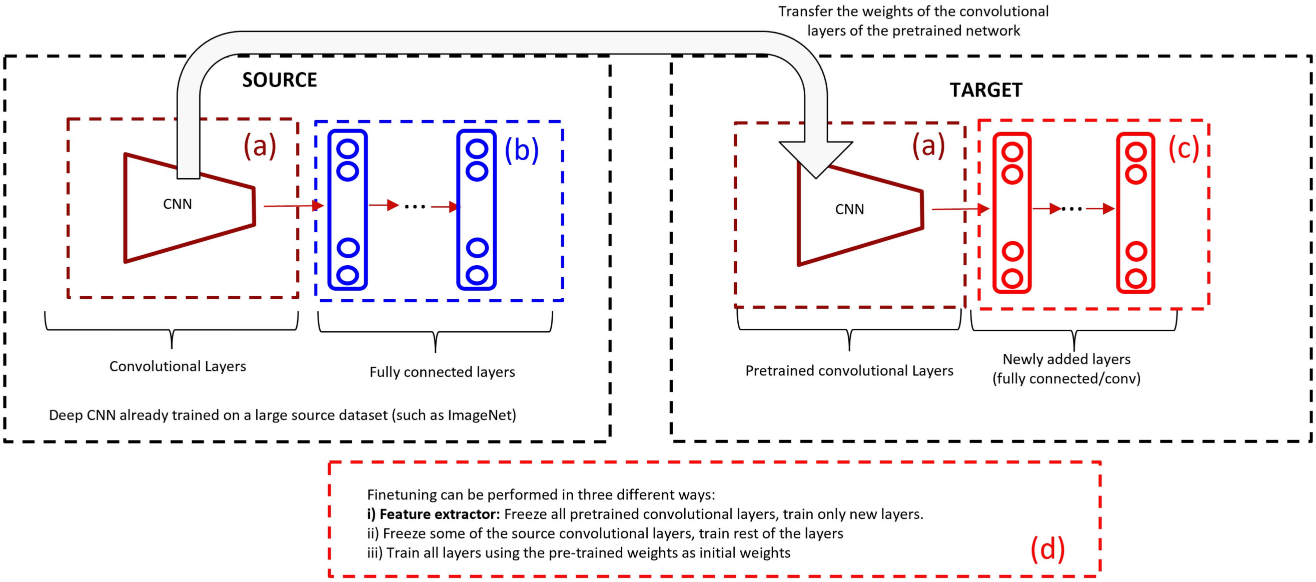

Transfer the weights of the convolutional layers of the pretrained network
SOURCE
(a)
CNN
(b)
Convolutional Layers
Fully connected layers
Deep CNN already trained on a large source dataset (such as ImageNet)
TARGET
(a)
CNN
(c)
Pretrained convolutional Layers
Newly added layers (fully connected/conv)
Finetuning can be performed in three different ways:
i) Feature extractor: Freeze all pretrained convolutional layers, train only new layers.
ii) Freeze some of the source convolutional layers, train rest of the layers
iii) Train all layers using the pre-trained weights as initial weights
(d)

Fig. 3. Overall transfer learning and finetuning architecture. In this framework, the CNN part of the network denoted by (a) remains same in the SOURCE and TARGET networks. On the source side, the full network [(a) convolutional layers and (b) fully connected layers] is trained on a large source dataset (e.g., ImageNet). For transfer learning, the weights of the CNN (a) from SOURCE are used to initialize the CNN (a) on the TARGET side. New fully connected/convolutional layers (c) on TARGET side are appropriately designed for the target domain. This new TARGET network is finetuned on the smaller target dataset using the three methods described in (d).

of these layers (referred to as freezing the layers) and train only the new layers. It should be noted here that the pretrained network becomes purely as a feature extractor [45] if all the layers of the network are frozen. Several examples exist in literature where finetuning deep neural networks using large source datasets such as the ImageNet [4] has improved classification performance on a similar target domain (such as natural images). However, the domain of remote sensing imagery is different in comparison to natural images [46]. First, the size, shape, and orientation of objects typically found in remote sensing imagery (which is typically overhead imagery) is different to those found in natural images. Second, the type of sensors (LIDAR, thermal, radar, and multispectral [46]) used in remote sensing imagery are typically not available in the domain of natural images. The important question that we aim to answer in this work is to verify if the principle of finetuning using well-known models that were trained on large scale dataset of natural images such as ImageNet [4] can help improve the classification performance of a different domain such as remote sensing imagery.

Some recent research efforts that use finetuning in the domain of remote sensing include the following: Marmanis et al. [47] train a smaller CNN on the features extracted from a much larger CNN that was pretrained on the ImageNet dataset to show improved predictive performance on the University of California (UC), Merced Land Use dataset [48]. Nogueira et al. [49] prove that using several well-known networks that extracting features from fintetuned networks and classifying them using an SVM classifier improved performance on Merced [48], Brazilian Coffee Scenes [50], and the RS-19 [51] datasets in contrast to training the networks from scratch or using hand-crafted features. Yang et al. [52] combine the outputs of a CNN capturing spectral information and another CNN capturing spatial information. In their solution, earlier layers of a network pretrained on different hyperspectral imagery collected from the same sensor are combined with new randomly initialized layers. In [49] and [52], the finetuning strategies consider both the source and target datasets being from the remote sensing domain. In contrast, this work aims at answering the question whether finetuning strategy can improve classification performance when the source and target domains are different. Cheng et al. [53] develop a two-phase approach, where they use the CNNs that were pretrained on the ImageNet dataset (natural images) as off-the-shelf feature extractors to extract spatial features from hyperspectral imagery, followed by a metric learning-based feature fusion with spectral features approach to improve the classification performance. In contrast, we evaluate a broader spectrum of finetuning approaches in addition to treating the pretrained CNNs as feature extractors. In addition, the dataset used in this work is based on NAIP imagery (upto four spectral bands) in contrast to the hyperspectral imagery used in [52] and [53], which has a significantly larger number of spectral bands. Chew et al. [54] also use ImageNet pretrained network albeit purely as a feature extractor for crop classification using drone (UAV) imagery. The extracted features are then fed to a feed-forward neural network for classification. In contrast, in this work, we evaluate a broader spectrum of finetuning approaches. Although, Gadiraju and Vatsavai [55] evaluate finetuning strategies on NAIP imagery, they do not show the impact of increasing data size. In addition, they do not study the number of layers to be frozen as we presented in this work. While Nowakowski et al. [56] perform similar studies as this work, they do not study the impact on finetuning strategies with increasing dataset size as we presented in this article. Other notable examples of inductive transfer learning include metalearning-based solutions [57], [58]. While in this article, we focus purely on inductive transfer learning and finetuning in deep neural networks, we provide a brief window into research in other types of transfer learning as follows.

Multitask learning focuses on learning multiple-related tasks simultaneously. As described in [39], the tasks can be related or unrelated. As described in [59], the additional information provided by learning multiple related tasks together results in an inductive bias, which allows the models to prefer one hypothesis over the other thereby improving the generalizability of the models. As described in [39], the typical goal of multitask learning is to detect common features between the multiple tasks. More specifically in terms of multitask learning using deep learning, the parameter sharing between the tasks can be *hard parameter sharing*, where the data pertaining to all the tasks initially share a common network before the task-specific layers are incorporated, or *soft parameter sharing*, where each task does not share networks, and the distance between parameters is reduced using methods such as l_2 regularization. Benedetti et al. [60] build a multitask solution for crop classification that uses a two-stream neural network, with the spatial stream using high spatial resolution imagery and the temporal stream using high temporal resolution imagery. They use a custom loss function, which is a weighted sum of losses of the spatial, temporal, and combined outputs. Bischke et al. [61] use a cascading multitask loss to achieve a better semantic segmentation for segmenting buildings from satellite imagery. Liu and Shi [62] use multitask learning with multiple hyperspectral imagery datasets to improve the performance. Dobrescu et al. [63] use multitask learning to learn multiple characteristics of the plant such as leaf count and phenotype counting simultaneously.

2) *Transductive transfer learning* [39]: In transductive transfer learning, the source and target tasks are the same, while the source and target domains are different. The specific case where the probability distributions of the source and target domains are different involves methods such as domain adaptation (DA), covariate shift, and selection bias.

DA is a well-known strategy that aims at finding features invariant to the differences in the source and target domains [43]. This helps alleviate the high variability challenge. Elshamli et al. [64] provide a detailed overview over the various DA methodologies adapted by the remote sensing community to overcome the variability challenge. Othman et al. [65] apply DA by adding additional fully connected layers on top of pretrained deep CNNs and reduces the shift in domains by incorporating regularization in the form of maximum mean discrepancy (MMD) and graph Laplacian. Li et al. [66] adopt a two-stage deep DA—in the first stage, MMD is used to reduce the domain shift between the labeled source and unlabeled target domains. In the second stage, the network learned in the previous stage is used with the labeled source data and labeled target data and pairwise loss is used to reduce the distance between the domains. Recently, generative adversarial networks (GANs) have become popular for DA. Jia et al. [35] develop a purely temporal LSTM-based solution that uses attention to capture the critical time periods in the crop phenology contributing to the classification (also known as discriminative period [35]). They use unsupervised DA in the form of a cyclic GAN that learns a transformation function that maps the data in the target domain to a similar distribution in the source domain. Yu et al. [67] also perform unsupervised DA using adversarial domain learning for alignment of features between the source and target domains.

As described earlier, we will focus more on the inductive transfer learning approaches. More specifically, we evaluate several finetuning strategies to overcome the challenge of limited labeled data and high variability. In the next section, we discuss the methodology of these finetuning strategies.

IV. Methodology

As described in the previous section, in this article, we focus on supervised finetuning methods for crop classification. As shown in Fig. 3, in supervised finetuning, the focus is on transferring relevant knowledge learnt on a large dataset to train on a smaller dataset. Typically, the domains of the source and target domains remain similar. However, in this work, we focus on evaluating whether information can be transferred even when the source and target domains are dissimilar. The primary focus is on identifying whether information can be transferred, and if yes, how many layers of information can be transferred. In this section, we describe the following four training strategies we use in this article.

1) *Random weight initialization:* This strategy is the traditional method of training deep neural networks from scratch. We train the network from scratch only using the available training data where the weights of the deep neural network under consideration are initialized randomly. The main goal of this experiment is to evaluate performance of deep neural networks when trained from scratch using limited labeled data. In the rest of this article, we will refer to this strategy as s_1. In short, no information is transferred in this strategy from source to target domains.

2) *Finetuning strategies:* As described in [44], the earlier layers of a deep CNN represent/identify simple generic features such as edges, while more task specific features are identified by the deeper layers. As a result, a deep neural network trained on a large dataset can identify certain features much easier than when training the network

from scratch using random initial weights. The number of identifiable features depends upon the similarity of the source and target domains and tasks. Some common finetuning practices are as follows.

a) Using the pretrained weights of the CNN and not training them (also known as freezing the layers) on the target dataset, while only training the final classification layer(s). In other words, this strategy treats the deep neural network purely as a feature extractor. In the rest of this article, we refer to this strategy as (s_2).

b) Freezing specific layers of the CNN such as the earlier layers and training only the later layers to detect task specific features. In the rest of this article, we refer to this strategy as (s_3). To evaluate this strategy, we perform multiple experiments where we progressively increase the number of frozen layers in the network, starting from the first three layers to the penultimate layer. These experiments help us answer the question of how many layers of information can be transferred when the source and target domains are different. It should be noted here that when all the convolutional layers of the network are frozen, this experiment reduces to scenario s_2. It should also be noted that the layers being trained use the pretrained weights for initialization.

c) Freezing none of the layers of the CNN and training the network from scratch using the target dataset and using the pretrained weights purely for initialization. In the rest of this article, we refer to this strategy as (s_4).

In the next section, we describe the experimental setup and discuss the outcomes of the experiments.

V. Experiments and Results

In this section, we first describe the dataset used in this article. This is then followed by the experiments designed to evaluate the various training strategies.

A. Dataset

The dataset used in this article extends the spatial component of the dataset proposed in [68]. The original dataset [68] consisted of image patches collected during the crop growing season from the states of Iowa, Illinois, Georgia, North Dakota, Oklahoma, Alabama, Colorado, and Montana from the year 2017. Over 7000 additional image patches for corn, soy, and cotton were collected from North Carolina for the year 2016, and were randomly added to the train and test sets of the dataset. The new image patches introduce challenges discussed in Section I-A such as geographical and annual variations and noise due to the presence of other classes such as built up area. Table I gives the train, validation, and test split for updated dataset. To collect this additional data from North Carolina, we use the Cropland Data Layer (CDL) [12] [produced by the United States Department of Agriculture (USDA), National Agricultural Statistics Service (NASS)] as an alternate for ground truth, since manual surveying and collection of ground truth over such large coverage area is a challenging operation.

TABLE I
Number of Training, Validation, and Testing Image Patches Per Class for the Dataset

Crop	#Train	#Validation	#Test
Corn	8 049	2 245	3 560
Soy	8 053	2 245	3 562
Cotton	8 051	2 245	3 565
Spring wheat	6 737	2 245	2 247
Winter wheat	5 839	1 946	1 947
Barley	3 903	1 301	1 302

Similar to the original dataset, the CDL imagery for the state of North Carolina is parsed to find 8 × 8 patches of corn, soy, and cotton classes. Care is taken to ensure that over 80% of the collected patch belongs to the specified class. Then, the corresponding NAIP image patches (240 × 240) for the same geographic location are extracted. Since CDL data are at 30-m resolution and the NAIP imagery is at 1-m resolution, the 8 × 8 CDL image patches cover the same area as the 240 × 240 NAIP image patches.

B. Hardware and Software Configuration

We perform our experiments on NVIDIA RTX GPUs on the ARC cluster [69] and a ThinkStation P920 workstation with NVIDIA GV100. For building, training, and evaluating the deep learning model, we used the Tensorflow [70] deep learning library.

C. Experiments

In this section, we describe various experiments to evaluate the training strategies $s_1 - s_4$ described in Section IV. In order to study how the performance of each of the strategies $s_1 - s_4$ changes with changes in dataset size, we create four different subsets of the training data: D_1, D_2, D_3, and D_4 containing roughly 25%, 50%, 75%, and 100% of the original training data. Fig. 5 shows the sizes of each of these subsets of training data. Care is taken to ensure that sampling is performed in a stratified manner. It should be noted here that the size of training data increases from D_1 to D_4, but the test data remains the same. All the experiments are evaluated on a test set consisting of 16 183 images.

For each of the four strategies discussed previously (s_1 to s_4), we perform training using each of the D_1–D_4 datasets and we perform prediction on the original test set consisting of 16 183 images. In each experiment, we preprocess the data, followed by training and inference, and finally, we evaluate our predictions. While we follow the same preprocessing procedure for all the experiments, specific parts of the training procedure differ depending upon the training strategy. We describe each step in detail as follows.

1) Preprocessing: In all the experiments, for both training and inference, the collected images, which are originally 240 × 240 are resized to 224 × 224 to match the input shape of the deep neural network. Then, the image features are normalized to

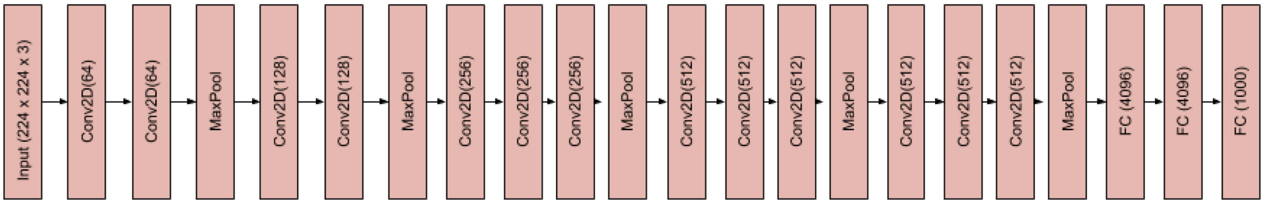


Fig. 4. Architecture of the VGG16 Network [30].

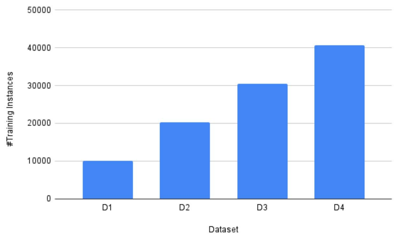


Fig. 5. Dataset size versus number of training instances.

prevent numerical overflow. We describe the training procedure as follows.

2) Training: In each experiment, we perform training for 32 epochs. Given the limited amount of labeled data, deep neural networks are prone to overfitting. In order to avoid this, we add a dropout layer [71] for regularization. In addition, we also use early stopping [72] as another means of regularization. Early stopping ensures that if the validation loss does not improve for over eight successive epochs, the training procedure is terminated. We use categorical cross entropy loss and the Adam [73] optimizer in the training. In each experiment, we perform a grid search on the initial learning rate for the Adam optimizer and the dropout rate to identify the optimal hyperparameters. The hyperparameters that achieve the highest validation accuracy are picked.

Based on the findings in the previous work [55], we choose the VGG network as our backbone network. Fig. 4 shows the architecture of the VGG16 [30] network. The VGG16 network has 13 convolutional layers interspersed with five pooling layers for dimensionality reduction and three fully connected layers (classification layers). As described earlier, we use the pretrained network that was trained on 1000 classes of the ImageNet dataset [4]. We remove the three final classification (fully connected) layers and replace them with a Global Average Pooling layer, followed by two fully connected layers FC_1 (512 units) and FC_2 (256 units) with ReLU activation followed by a fully connected layer (six units, equal to the number of classes) with softmax activation to achieve the predicted probabilities. FC_1 and FC_2 have a dropout layer in between for regularization.

Next, we discuss the specific training procedures for each of the four strategies.

1) *Strategy s_1:* As described in Section IV, in this strategy, the network is being trained from scratch, and no information is being carried over from the pretrained model. This is achieved by randomly initializing the weights of the neural network and then training the entire network on the dataset. We perform four experiments using this strategy to help us understand the importance of the dataset size on model performance. Each experiment involves training the model using strategy s_1 on each of the four datasets D_1, D_2, D_3, and D_4.
2) *Strategy s_2:* As described in Section IV, this strategy treats the network purely as a feature extractor. Only the final classification layers (FC_1 and FC_2) are trained and we use the ImageNet pretrained weights for the rest of the layers. Similar to the experiments performed on s_1, four experiments are performed using strategy s_2 on datasets D_1 to D_4.
3) *Strategy s_3:* As described in Section IV, this strategy evaluates how much information is transferable between the two domains. For each dataset, D_1–D_4, we perform multiple experiments where we freeze specific layers of the network and train the rest of the layers. In other words, in this strategy, we treat the number of frozen layers also as a hyperparameter.
4) *Strategy s_4:* As described in Section IV, this strategy focuses on training the model from scratch. The difference between this strategy and s_1 is that the weight initialization is using the ImageNet pretrained weights. This strategy helps us understand the importance of weight initialization.

The outcomes of these experiments are discussed in Section V-D. Once training is completed and optimal hyperparameters are identified, we evaluate the outcomes described as follows.

3) Evaluation: We evaluate the methods by comparing their error rates. In addition, we also extract the features from the global average pooling layer for the best performing run of each experiment and use the t-SNE (t-distributed stochastic neighbor embedding) [74] for visualizing the separation ability of the trained network.

We discuss the outcomes of these experiments in the following section.

We describe how each of the aforementioned experiments were evaluated, and discuss the outcomes of these experiments in the following section.

D. Evaluation and Discussion

Once the optimal hyperparameters have been identified for each strategy, we perform training and inference using the optimal hyperparameters five times and calculate the mean and

TABLE II
ERROR RATES REPORTED FOR EACH OF THE TRANSFER LEARNING STRATEGIES

Experiment	D_1 Error rate	D_2 Error rate	D_3 Error rate	D_4 Error rate
s_1 Training the CNN from scratch using random weight initialization	23.11±0.68	18.88±0.76	16.57±0.65	14.73±0.77
s_2 Using the CNN purely as a feature extractor	31.63±1.12	28.32±0.45	27.30±0.22	25.41±0.47
s_3 Freezing early layers in the CNN, training the remaining layers	14±0.52	11.87±0.46	10.89±0.3	9.82±0.32
s_4 Training the CNN from scratch with pretrained weights as initialization	14.51±1.16	11.69±0.48	10.42±0.42	**9.56±0.34**

Each experiment is run five times and the average and the standard deviation are noted.
The bold values indicate lowest error rate.

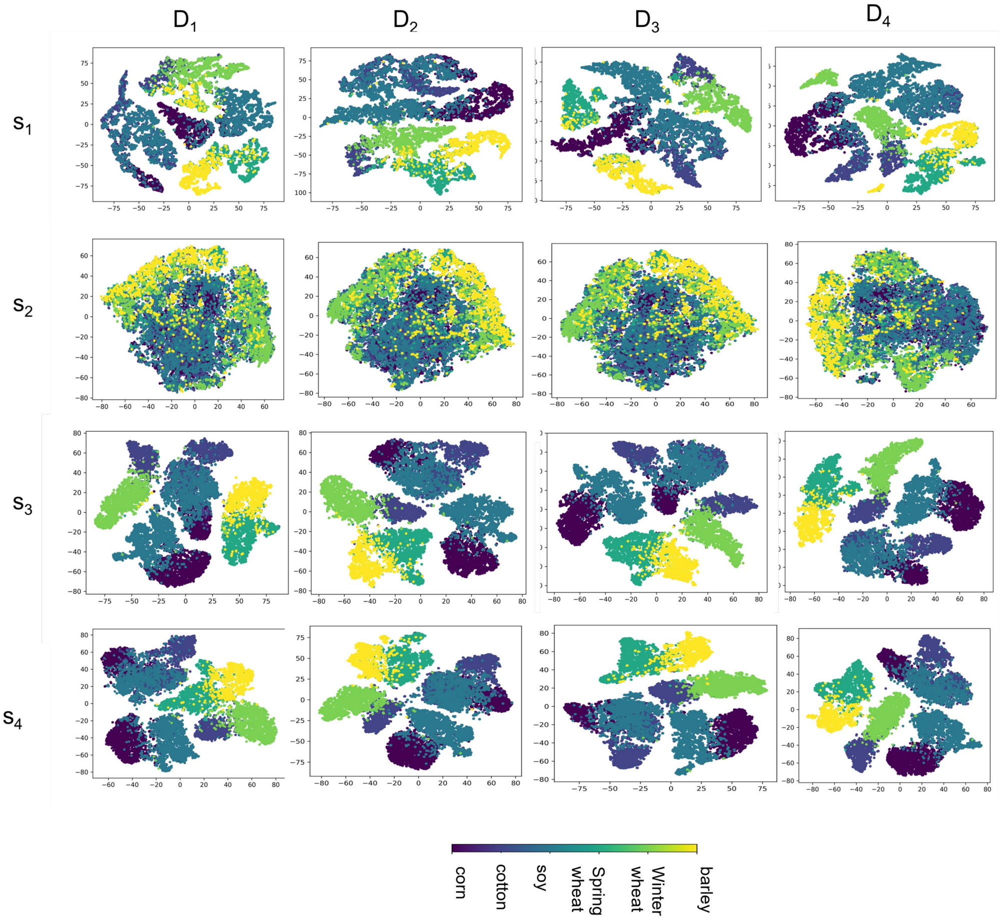


Fig. 6. t-SNE plots for all the experiments for all versions of the dataset. Each row corresponds to an experiment (as indicated by the first column on the left), and each column refers to one of the dataset versions (as indicated by the first row on the top). It can be clearly seen that training using the *s2* strategy does not produce well-separable features. In the strategy s_1, the separability improves as the size the dataset increases from D_1 to D_4. *s3 and s4* produce well-separable features.

standard deviation of the error rates for each of these runs. Table II depicts the error rates obtained for the four approaches described in the previous section. In addition, Fig. 6 shows the t-SNE plots for all the outcomes of experiments from Table II. Each row in the table corresponds to one of the strategies $(s_1 - s_4)$, while each column corresponds to dataset $(D_1 - D_4)$ used for that approach. In the rest of this section, we will refer to the combination of training strategy and the corresponding data subset using the $(s_i - D_i)$ notation, referring to applying the training strategy s_i on data subset D_i.

1) In all four strategies, we can clearly see the improvement in performance with increasing the size of the training data from D_1 to D_4. In the strategy s_1, where we train the CNN from scratch using random weight initialization (and therefore, not transferring any information from the source domain), this improvement is clearly evident, there by proving the importance of large labeled datasets when training deep neural networks from scratch. In addition, we can observe the improvement in separability of classes from top left to bottom right in Fig. 6. The number of

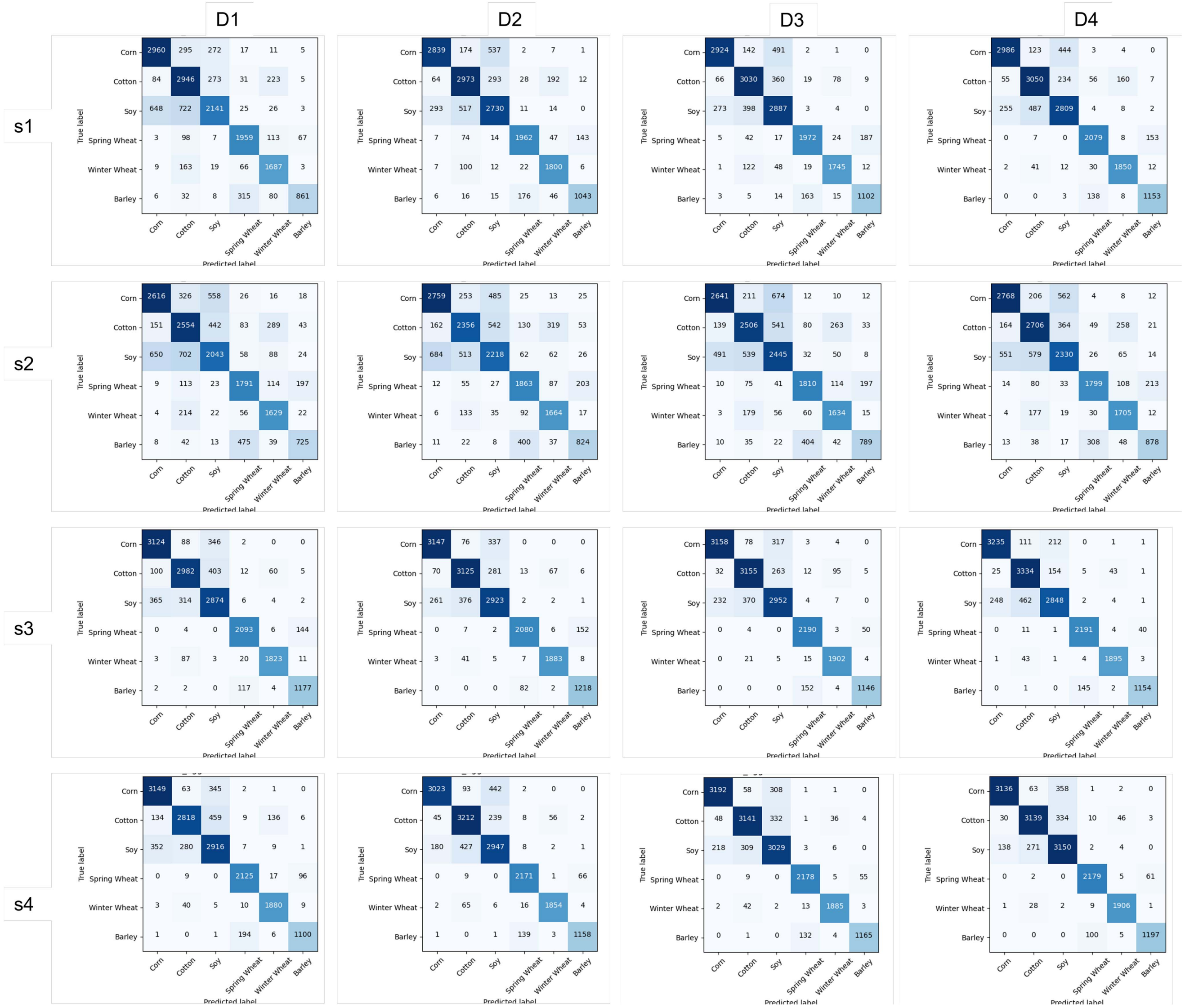


Fig. 7. Confusion matrices of all the experiments for all versions of the dataset. Each row corresponds to an experiment (as indicated by the first column on the left), and each column refers to one of the dataset versions (as indicated by the first row on the top). In general, training using the *s2* strategy produces large number of misclassifications across all classes. In the strategy s_1, the number of misclassifications reduce as the size the dataset increases from D_1 to D_4. *s3 and s4* produce less misclassifications.

misclassifications for each class can also be seen to reduce in Fig. 7.

2) Using the ImageNet pretrained model purely as an off-the-shelf feature extractor in s_2 consistently performs poorly for all the datasets D_1–$D4$. In contrast, training from scratch using random weight initialization with a small dataset ($s_1 - D_1$) still outperforms training on a much larger dataset with s_2 (i.e., $s_2 - D_4$). This clearly indicates that deep CNNs cannot be used as purely off-the-shelf feature extractors when there is a difference between the source and target domains. We can clearly see the poor separability between the features in the corresponding t-SNE plots for the strategy s_2. We can also see comparatively much larger number of misclassified labels for all the classes for the strategy s_2 in Fig. 7.

3) Based on the aforementioned observation, we next study whether a portion of the pretrained network trained on a dataset from a different domain can be used for training in the strategy s_3. As described earlier, in the strategy s_3, we perform multiple experiments where we progressively increase the number of frozen layers in the network. Fig. 8 demonstrates the error rates on the test datasets for the strategy s_3, where the x-axis represents the number of frozen layers and the y-axis represents the error rate on test data. A clear upward trend can be noticed for all four datasets $D_1 - D_4$ as the number of trained layers decreases (in other words, as the number of frozen layers increases). This reinforces the earlier inference that not all layers' features are transferable from deep CNN pretrained on a dataset from a different domain. The best

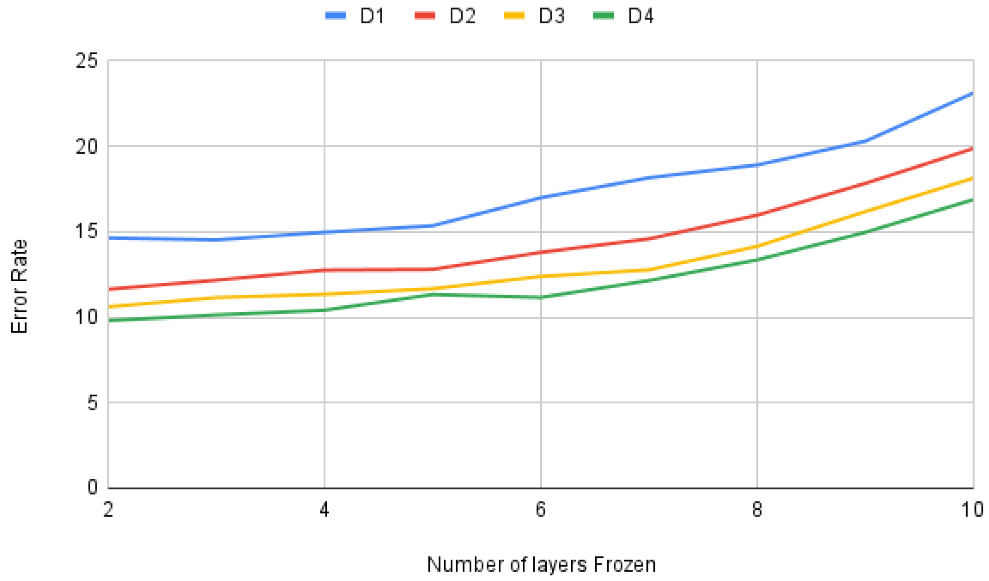


Fig. 8. Comparison of error rates on test data with increasing number of layers frozen.

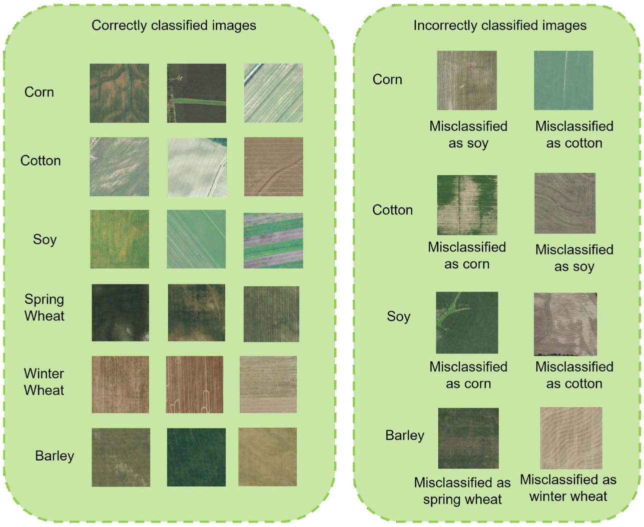


Fig. 9. Examples of crop image classification. Best overall average classification accuracy for $s_4 - D_4 = 90.44\%$, overall average error rate for $s_4 - D_4 = 9.56\%$.

error rate for s_3 for all datasets was observed with the early two convolutional layers of the VGG network being frozen. This proves that early layers in a deep CNN learn more generic features, while the latter layers learn more task specific features. We perform training and prediction on the deep neural network by freezing the early two convolutional layers in addition to the other optimal hyperparameters—initial learning rate and dropout rate. The mean and standard deviation of this approach are listed in Table II.

4) In strategy s_4, where we train the entire network using the ImageNet weights purely as an initialization demonstrates a similar performance to strategy s_3. In comparison to the strategy s_1, this shows the importance of good weight initialization. It also demonstrates that with good weight initialization or using finetuning, we can achieve equivalent predictive performance using a significantly smaller dataset than when training deep neural networks from scratch using random weight initialization. This is clearly demonstrated when comparing $s_1 - D_4$ and $s_4 - D_1$. This can also be seen in the confusion matrices in Fig. 7. Training from scratch using random weight initialization (s_1) requires significantly larger datasets to achieve good performance and this introduces computational overhead due to the size of the large dataset.

Finally, Fig. 9 shows some examples of correctly and incorrectly classified images for the various classes.

VI. Conclusion

In this article, we discuss various challenges associated with performing crop classification using remote sensing imagery. We highlight how limited labeled data, when combined with the high variability in the remote sensing data limit the performance of traditional machine learning approaches and modern deep learning approaches. We discuss in detail the various deep transfer learning methodologies employed in literature to alleviate these challenges for deep neural networks. We evaluated several finetuning strategies using the VGG16 Network using a benchmark dataset.

Based on our experiments, we make the following observations.

1) When finetuning deep neural networks using a pretrained network from a different domain, the pretrained network is unsuitable to be used purely as an off-the-shelf feature extractor (strategy s_2).
2) Instead, when we evaluated whether a portion of the pretrained network is transferable (by freezing these layers),

we observed an increasing trend in the error rate as the number of frozen layers increases. We observed that freezing the early layers of the deep neural network achieved the best performance.

3) We also observed that the strategy s_4, where we only use the pretrained weights as a good weight initialization performed significantly better in comparison to training from scratch using random weights. This indicates the importance of a good weight initialization for deep neural networks.
4) Finally, strategies s_3 and s_4, when compared with the strategy s_1 clearly demonstrate the advantage of using a pretrained network trained using a different domain dataset. We can achieve the same level of performance using s_3 and s_4 using a much smaller dataset (D_1) in comparison to training from scratch with random weight initialization (s_1) using a significantly larger dataset (D_4). s_1 requires a much larger dataset to achieve comparable performance as s_3 and s_4, which makes s_1 computationally more expensive as well.

Our experiments and study can provide a good starting point for those researchers who are seeking to build effective deep learning solutions for the remote sensing and agricultural domains with limited labeled data. When very limited data are present, either using the pretrained weights as an initialization or freezing early layers of a network trained from a different domain is a good strategy to achieve significantly better performance than when training the model from scratch or using the pretrained models purely as a feature extractor.

While this article has focused on a single data source, with the increasing number of data sources available for the same region at multiple spatial and temporal resolutions and modalities, our next research will focus on building deep learning solutions that can combine the most useful information from each of these sources to improve the overall predictive outcomes.

Acknowledgment

The authors would like to thank Dr. F. Mueller, North Carolina State University, for infrastructure support via the ARC cluster. The authors would also like to thank Lenovo for infrastructure support, and the members of STAC Lab, NCSU, for their help and support.

References

[1] D. M. Howard, B. K. Wylie, and L. L. Tieszen, “Crop classification modelling using remote sensing and environmental data in the greater platte river basin, USA,” *Int. J. Remote Sens.*, vol. 33, no. 19, pp. 6094–6108, 2012.

[2] C. Planque et al., “National crop mapping using sentinel-1 time series: A knowledge-based descriptive algorithm,” *Remote Sens.*, vol. 13, no. 5, 2021, Art. no. 846.

[3] National Agriculture Imagery Program (NAIP), 2018. [Online]. Available: https://www.usgs.gov/centers/eros/science/usgs-eros-archive-aerial-photography-national-agriculture-imagery-program-naip

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[5] M. Chen, L. Ma, W. Wang, and Q. Du, “Augmented associative learning-based domain adaptation for classification of hyperspectral remote sensing images,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 6236–6248, 2020.

[6] A. Huete, C. Justice, and W. Van Leeuwen, “MODIS vegetation index (MOD13),” *Algorithm Theor. Basis Document*, vol. 3, no. 213, pp. 295–309, 1999.

[7] Z. Sun, L. Di, H. Fang, and A. Burgess, “Deep learning classification for crop types in north dakota,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 2200–2213, 2020.

[8] H. Kerner et al., “Resilient in-season crop type classification in multispectral satellite observations using growth stage normalization,” 2020, *arXiv:2009.10189*.

[9] E. Bartholome and A. S. Belward, “GLC2000: A new approach to global land cover mapping from earth observation data,” *Int. J. Remote Sens.*, vol. 26, no. 9, pp. 1959–1977, 2005.

[10] O. Arino et al., “GlobCover: ESA service for global land cover from MERIS,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2007, pp. 2412–2415.

[11] M. A. Friedl et al., “Modis collection 5 global land cover: Algorithm refinements and characterization of new datasets,” *Remote Sens. Environ.*, vol. 114, no. 1, pp. 168–182, 2010.

[12] C. Boryan, Z. Yang, R. Mueller, and M. Craig, “Monitoring US agriculture: The US Department of Agriculture, National Agricultural Statistics Service, Cropland Data Layer Program,” *Geocarto Int.*, vol. 26, no. 5, pp. 341–358, 2011.

[13] A. Mathur and G. M. Foody, “Crop classification by support vector machine with intelligently selected training data for an operational application,” *Int. J. Remote Sens.*, vol. 29, no. 8, pp. 2227–2240, 2008.

[14] J. A. Benediktsson, X. C. Garcia, B. Waske, J. Chanussot, J. R. Sveinsson, and M. Fauvel, “Ensemble methods for classification of hyperspectral data,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2008, pp. I–62–I–65.

[15] G. Camps-Valls, L. Gómez-Chova, J. Calpe-Maravilla, E. Soria-Olivas, J. D. Martín-Guerrero, and J. Moreno, “Support vector machines for crop classification using hyperspectral data,” in *Proc. Iberian Conf. Pattern Recognit. Image Anal.*, 2003, pp. 134–141.

[16] A. O. Ok, O. Akar, and O. Gungor, “Evaluation of random forest method for agricultural crop classification,” *Eur. J. Remote Sens.*, vol. 45, no. 1, pp. 421–432, 2012.

[17] I. L. Castillejo-González et al., “Object-and pixel-based analysis for mapping crops and their agro-environmental associated measures using quickbird imagery,” *Comput. Electron. Agriculture*, vol. 68, no. 2, pp. 207–215, 2009.

[18] J. M. Peña-Barragán, M. K. Ngugi, R. E. Plant, and J. Six, “Object-based crop identification using multiple vegetation indices, textural features and crop phenology,” *Remote Sens. Environ.*, vol. 115, no. 6, pp. 1301–1316, 2011.

[19] M. Immitzer, F. Vuolo, and C. Atzberger, “First experience with sentinel-2 data for crop and tree species classifications in central Europe,” *Remote Sens.*, vol. 8, no. 3, 2016, Art. no. 166.

[20] S. Shekhar, P. R. Schrater, R. R. Vatsavai, W. Wu, and S. Chawla, “Spatial contextual classification and prediction models for mining geospatial data,” *IEEE Trans. Multimedia*, vol. 4, no. 2, pp. 174–188, Jun. 2002.

[21] P. Ghamisi, J. A. Benediktsson, and M. O. Ulfarsson, “Spectral–spatial classification of hyperspectral images based on hidden Markov random fields,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2565–2574, May 2014.

[22] R. R. Vatsavai, “Gaussian multiple instance learning approach for mapping the slums of the world using very high resolution imagery,” in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2013, pp. 1419–1426. [Online]. Available: https://doi.org/10.1145/2487575.2488210

[23] S. R. Blanco, D. B. Heras, and F. Argüello, “Texture extraction techniques for the classification of vegetation species in hyperspectral imagery: Bag of words approach based on superpixels,” *Remote Sens.*, vol. 12, no. 16, Aug. 2020.

[24] R. M. Haralick et al., “Statistical and structural approaches to texture,” *Proc. IEEE*, vol. 67, no. 5, pp. 786–804, May 1979.

[25] 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.

[26] V. Vijayaraj et al., “Overhead image statistics,” in *Proc. 37th IEEE Appl. Imagery Pattern Recognit. Workshop*, Washington, DC, USA, 2008, pp. 1–8.

[27] R. R. Vatsavai, "High-resolution urban image classification using extended features," in *Proc. IEEE 11th Int. Conf. Data Mining Workshops*, Vancouver, BC, Canada, 2011, pp. 869–876.

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learn.* Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[29] D. Lunga, H. L. Yang, A. Reith, J. Weaver, J. Yuan, and B. Bhaduri, "Domain-adapted convolutional networks for satellite image classification: A large-scale interactive learning workflow," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 3, pp. 962–977, Mar. 2018.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[31] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[33] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[34] J. Donahue et al., "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. I-647–I-655.

[35] X. Jia, G. Nayak, A. Khandelwal, A. Karpatne, and V. Kumar, "Classifying heterogeneous sequential data by cyclic domain adaptation: An application in land cover detection," in *Proc. SIAM Int. Conf. Data Mining.*, 2019, pp. 540–548.

[36] S. Ji, C. Zhang, A. Xu, Y. Shi, and Y. Duan, "3D convolutional neural networks for crop classification with multi-temporal remote sensing images," *Remote Sens.*, vol. 10, no. 1, 2018. [Online]. Available: https://www.mdpi.com/2072-4292/10/1/75

[37] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016.

[38] P. Liang, W. Shi, and X. Zhang, "Remote sensing image classification based on stacked denoising autoencoder," *Remote Sens.*, vol. 10, no. 1, 2018, Art. no. 16.

[39] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[40] P. Hao, L. Di, C. Zhang, and L. Guo, "Transfer learning for crop classification with cropland data layer data (CDL) as training samples," *Sci. Total Environ.*, vol. 733, 2020, Art. no. 138869.

[41] C. Tao, J. Qi, W. Lu, H. Wang, and H. Li, "Remote sensing image scene classification with self-supervised paradigm under limited labeled samples," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 8004005.

[42] K. Ayush et al., "Geography-aware self-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10181–10190.

[43] R. Ghosh, X. Jia, and V. Kumar, "Land cover mapping in limited labels scenario: A survey," 2021, *arXiv:2103.02429*.

[44] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks ?," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, Cambridge, MA, USA, 2014, pp. 3320–3328.

[45] B. Q. Huynh, H. Li, and M. L. Giger, "Digital mammographic tumor classification using transfer learning from deep convolutional neural networks," *J. Med. Imag.*, vol. 3, no. 3, 2016, Art. no.034501.

[46] J. Song, S. Gao, Y. Zhu, and C. Ma, "A survey of remote sensing image classification based on CNNs," *Big Earth Data*, vol. 3, no. 3, pp. 232–254, 2019.

[47] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "Deep learning Earth observation classification using ImageNet pretrained networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 105–109, Jan. 2016.

[48] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, New York, NY, USA, 2010, pp. 270–279. [Online]. Available: https://doi.org/10.1145/1869790.1869829

[49] K. Nogueira, O. A. Penatti, and J. A. Dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognit.*, vol. 61, pp. 539–556, 2017.

[50] O. A. Penatti, K. Nogueira, and J. A. Dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 44–51.

[51] G.-S. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun, and H. Maître, "Structural high-resolution satellite image indexing," in *ISPRS TC VII Symp.—100 Years ISPRS*, 2010, vol. 38, pp. 298–303.

[52] J. Yang, Y.-Q. Zhao, and J. C.-W. Chan, "Learning and transferring deep joint spectral–spatial features for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4729–4742, Aug. 2017.

[53] G. Cheng, Z. Li, J. Han, X. Yao, and L. Guo, "Exploring hierarchical convolutional features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6712–6722, Nov. 2018.

[54] R. Chew et al., "Deep neural networks and transfer learning for food crop identification in UAV images," *Drones*, vol. 4, no. 1, 2020, Art. no. 7.

[55] K. K. Gadiraju and R. R. Vatsavai, "Comparative analysis of deep transfer learning performance on crop classification," in *Proc. 9th ACM SIGSPATIAL Int. Workshop Analytics Big Geospatial Data*, 2020, pp. 1–8.

[56] A. Nowakowski et al., "Crop type mapping by using transfer learning," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 98, 2021, Art. no. 102313.

[57] M. Rußwurm, S. Wang, M. Korner, and D. Lobell, "Meta-learning for few-shot land cover classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 200–201.

[58] Y. Li, Z. Shao, X. Huang, B. Cai, and S. Peng, "Meta-fseo: A meta-learning fast adaptation with self-supervised embedding optimization for few-shot remote sensing scene classification," *Remote Sens.*, vol. 13, no. 14, 2021, Art. no. 2776.

[59] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, *arXiv:1706.05098*.

[60] P. Benedetti, D. Ienco, R. Gaetano, K. Ose, R. G. Pensa, and S. Dupuy, "M3Fusion: A deep learning architecture for multiscale multimodal multitemporal satellite data fusion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 12, pp. 4939–4949, Dec. 2018.

[61] B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel, "Multi-task learning for segmentation of building footprints with deep neural networks," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 1480–1484.

[62] S. Liu and Q. Shi, "Multitask deep learning with spectral knowledge for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 12, pp. 2110–2114, Dec. 2020.

[63] A. Dobrescu, M. V. Giuffrida, and S. A. Tsaftaris, "Doing more with less: A multitask deep learning approach in plant phenotyping," *Front. Plant Sci.*, vol. 11, 2020. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpls.2020.00141

[64] A. Elshamli, G. W. Taylor, A. Berg, and S. Areibi, "Domain adaptation using representation learning for the classification of remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 9, pp. 4198–4209, Sep. 2017.

[65] E. Othman, Y. Bazi, F. Melgani, H. Alhichri, N. Alajlan, and M. Zuair, "Domain adaptation network for cross-scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4441–4456, Aug. 2017.

[66] Z. Li, X. Tang, W. Li, C. Wang, C. Liu, and J. He, "A two-stage deep domain adaptation method for hyperspectral image classification," *Remote Sens.*, vol. 12, no. 7, 2020, Art. no. 1054.

[67] C. Yu, C. Liu, H. Yu, M. Song, and C.-I. Chang, "Unsupervised domain adaptation with dense-based compaction for hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 12287–12299, 2021.

[68] K. K. Gadiraju, B. Ramachandra, Z. Chen, and R. R. Vatsavai, "Multimodal deep learning based crop classification using multispectral and multitemporal satellite imagery," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2020, pp. 3234–3242. [Online]. Available: https://doi.org/10.1145/3394486.3403375

[69] F. Mueller, "ARC: A root cluster for research into scalable computer systems." Accessed: May 1, 2023. [Online]. Available: https://arcb.csc.ncsu.edu/~mueller/cluster/arc/

[70] M. Abadi et al., "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Conf. Operating Syst. Des. Implementation*, 2016, pp. 265–283.

[71] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[72] R. Caruana, S. Lawrence, and C. L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 402–408.

[73] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[74] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

Krishna Karthik Gadiraju received the M.S. degree in computer science from the University of Cincinnati, Cincinnati, OH, USA, in 2014. He is currently working toward the Ph.D. degree in computer science with a focus on building machine learning solutions to address challenges in agriculture with the Department of Computer Science, North Carolina State University, Raleigh, NC, USA .

He currently works as a Software Engineer with Juniper Networks, Sunnyvale, CA, USA, building machine learning solutions in the traffic engineering space. He has authored and co-authored peer-reviewed articles in leading conferences such as ACM SIGKDD Conference on Knowledge Discovery and Data Mining, International Conference on Computational Science, and International Conference on Machine Learning and Applications. His research interests include geospatial data analysis, deep learning, time-series analysis, anomaly detection, and Big Data analysis and management.

Mr. Gadiraju co-organized the Workshop on Analytics for Big Geospatial Data 2022 (BigSpatial '22) that is held along with the ACM SIGSPATIAL GIS Conference.

Ranga Raju Vatsavai received the M.S. and Ph.D. degrees in computer science from the University of Minnesota, Minneapolis, MN, USA, in 2003 and 2008, respectively.

He worked with many leading research laboratories including the Center for Development of Advanced Computing (CDAC, Pune University Campus, India) in 1995–1998, AT&T Labs (R&D HQ, Middletown, NJ, USA) in 1998, University of Minnesota (Remote Sensing Laboratory, St. Paul, USA) in 1999–2004, IBM-Research (IRL, IIT-Delhi Campus, India) in 2004–2006, and Oak Ridge National Laboratory in 2006–2014. He is currently a Chancellor's Faculty Excellence Program (CFEP) Cluster Professor in geospatial analytics with the Department of Computer Science, North Carolina State University, Raleigh, NC, USA. As the Associate Director of the Center for Geospatial Analytics (CGA), he plays a leadership role in the center's strategic vision for GeoAI and spatial computing research. He has authored and co-authored more than 100 peer-reviewed articles in leading conferences and journals. His research interests include the intersection of spatial and temporal Big Data management, analytics, and high-performance computing with applications in the national security, geospatial intelligence, natural resources, climate change, location-based services, and human terrain mapping.

Dr. Vatsavai served on program committees of leading international conferences including ACM KDD, AAAI, ACM GIS, ECML/PKDD, ICDM, SDM, CIKM, IEEE BigData, WACV and co-chaired several workshops including ICDM/SSTDM, ICDM/KDCloud, ACM SIGSPATIAL BigSpatial, Supercomputing/BDAC, KDD/LDMTA, KDD/Sensor-KDD, SDM/ACS, and ICDM/WAIN. He has edited two books on "Knowledge Discovery from Sensor Data."