The power of transfer learning in agricultural applications: AgriNet

Zahraa Al Sahili* and Mariette Awad

Department of Electrical and Computer Engineering, Maroun Semaan Faculty of Engineering,
American University of Beirut, Beirut, Lebanon

Advances in deep learning and transfer learning have paved the way for various automation classification tasks in agriculture, including plant diseases, pests, weeds, and plant species detection. However, agriculture automation still faces various challenges, such as the limited size of datasets and the absence of plant-domain-specific pretrained models. Domain specific pretrained models have shown state of art performance in various computer vision tasks including face recognition and medical imaging diagnosis. In this paper, we propose AgriNet dataset, a collection of 160k agricultural images from more than 19 geographical locations, several images captioning devices, and more than 423 classes of plant species and diseases. We also introduce AgriNet models, a set of pretrained models on five ImageNet architectures: VGG16, VGG19, Inception-v3, InceptionResNet-v2, and Xception. AgriNet-VGG19 achieved the highest classification accuracy of 94% and the highest F1-score of 92%. Additionally, all proposed models were found to accurately classify the 423 classes of plant species, diseases, pests, and weeds with a minimum accuracy of 87% for the Inception-v3 model. Finally, experiments to evaluate of superiority of AgriNet models compared to ImageNet models were conducted on two external datasets: pest and plant diseases dataset from Bangladesh and a plant diseases dataset from Kashmir.

KEYWORDS

transfer learning, convolutional neural network, agriculture, pretrained models, plant
disease, pest, weed, plant species

Introduction

The world population is expected to reach over 9 billion by 2050, which will require an increase in food production by 70% (Silva and M. S. U. E, 2021). Considering scarcity of resources and climate change, intervention of artificial intelligence (AI) in agriculture is needed to overcome this challenge (Talaviya et al., 2020). AI advantages can span from plant diseases detection, robotic weeds and pests control, to herbal discovery. Plant diseases are not only a risk for food security only, but they also have disastrous effects on smallholder farmers where pests and weeds can lead to the destruction of around 50% of

the farm's plants (Rachman et al., 2017). Automated recognition of weeds, pests, and plant diseases can support smallholder farmers through free diagnosis services using mobile applications. Additionally, weed control robotics and sensor monitoring are another form of automation applied in regions with a limited number of agricultural expertise. Another important detection task is automated plant species recognition which is used in medical herbal research and in preventing extinction of non-discovered plant species (Tan et al.,).

Historically, the recognition task was relying on algorithms that needs handcrafted features, which were processed using relatively simple discriminative models such as linear classifiers or support vector machines (SVM) (Halevy et al., 2009; Rumpf et al., 2012; Wäldchen and Mäder, 2018; Madsen et al., 2020). After being the leading algorithm in all computer vision tasks, deep learning (DL) has been widely used in agriculture research for plant classification tasks (Madsen et al., 2020). To achieve good accuracy, DL models need very large datasets for their requirements as data-hungry neural networks (Halevy et al., 2009; Madsen et al., 2020). In the agricultural domain, datasets size is limited, so transfer learning would allow models reach higher accuracy without the need for more field data (Gandorfer et al., 2022). However, pretrained models used for transfer learning are not agriculture domain specific and were trained general computer vision datasets such as ImageNet. This creates a big challenge since convolutional models moves from low level features to higher level features and can lead to negative transfer (Gandorfer et al., 2022). For example, Yan et al. proposed transfer learning framework based on synthetic images to improve *in-vitro* soybean segmentation (Yang et al., 2022). The proposed framework resulted in a precision improvement of 8% considering the data abundancy in soybean applications (Yang et al., 2022).

Another challenge is models' robustness which is affected by the type of agricultural data use. Mohanty et al. compared usage of AlexNet and Google LeNet pretrained models for 26 diseases and 14 crops species through PlantVillage dataset which constitutes of 54,306 lab images. The Google LeNet achieved the highest accuracy of 99.3% (Mohanty et al., 2016). However, upon testing on trusted online sources, the accuracy dropped drastically to 31.4% (Mohanty et al., 2016). After that Singh et al. introduced PlantDoc, a 2,598 field images dataset of 13 crops and 27 classes (Singh et al., 2020a). To classify the dataset's images, multiple experiments were done on both uncropped and cropped images. For the non-cropped images, using ImageNet architectures with PlantVillage weights (Mohanty et al., 2016) resulted in twice accuracy compared to using the same architectures but with ImageNet weights (Singh et al., 2020a). In the cropped dataset experiment, transfer learning on VGG16 architecture with ImageNet weights resulted in an accuracy of 44.52% compared to 60.42% accuracy when VGG16 was used with plant village weights (Singh et al., 2020a).

Class imbalance degrades the performance of deep learning models on small classes including agricultural applications. For example, transfer learning was applied through ResNet50 architecture by Thapa et al. to detect two common apple diseases: apple scab and apple rust (Thapa et al., 2020). The accuracy obtained was 97% with an accuracy of only 51% for mixed diseases, which was caused by the small number of apples that have both diseases (Thapa et al., 2020). Additionally, Teimouri et al. used deep learning in the estimation of the weed growth stage (Teimouri et al., 2018). The dataset of 9649 images for various weed species was classified from 1 to 9 growth stages (Teimouri et al., 2018). Inception-v3 model was selected due to its good performance and low computational cost, and transfer learning was applied resulting in a 70% accuracy with a minimum accuracy of 46% for black-grass species that had the smallest set of images in the dataset (Teimouri et al., 2018).

Motivated to provide the agritech field with domain specific pretrained models that are robust and generalizable in various agricultural applications, the contributions of this work can be summarized as follows:

1. AgriNet dataset: a collection of 160k agricultural images from more than 19 geographical locations, several images captioning devices, and more than 423 classes of plant species and diseases.
2. AgriNet models: a set of pretrained models on five ImageNet architectures: VGG16, VGG19, Inception-v3, InceptionResNet-v2, and Xception and using AgriNet dataset. The proposed models are introduced to robustly classify the 423 classes of plant species, diseases, pests, and weeds with a minimum accuracy 94%, 92%, 89%,90%, and 88% for each architecture respectively.
3. Pretraining using AgriNet models: transfer learning using AgriNet models compared to ImageNet models was evaluated using experiments on two agricultural datasets: pest and plant diseases dataset from Bangladesh and a plant diseases dataset from Kashmir.

Materials and methods

Dataset

The AgriNet dataset is a collection of 160142 images belonging to 423 plant classes. The dataset was collected from 19 public datasets (The TensorFlow Team, Flowers (2019); Kumar et al., 2012a; Nilsback and Zisserman; Cassava disease classification (Kaggle); Olsen et al., 2019; Söderkvist, 2016; U. C. I. M. Learning, 2016; Giselsson et al., 2017; Peccia, 2018; Chouhan et al., 2019; J and Gopal, 2019; Krohling et al.,

2019; Rauf et al., 2019; D3v, 2020; Huang and Chuang, 2020; Huang and Chang, 2020; Makerere AI Lab, 2020; Marsh, 2020; Singh et al., 2020b) geographically distributed between United States, Denmark, Australia, United Kingdom, Uganda, India, Brazil, Pakistan, and Taiwan. It includes field and lab images from different cameras and mobile devices, and it can perform multiple agricultural classification tasks, such as species, weed, pest, and plant diseases detection. Sample dataset images is displayed in Figure 1.

The dataset classes were constructed by merging the same classes from multiple datasets in one class. This provides better classification performance through training the neural network to classify images regardless of the image location, quality, and

device, which was a common challenge reported in the literature. For example, for the tomato plant diseases, images were combined from datasets (Rauf et al., 2019; D3v, 2020; Makerere AI Lab, 2020) that included lab and field images from the United States, India, and Taiwan.

The collected dataset is highly imbalanced. As listed in Table 1, the average number of images per class is 378 images. Moreover, the number of classes with images less than 100 is 102 classes and the number of classes with images greater than 1000 is 44. In addition to class imbalance, a categorical imbalance between the three major tasks also exists. While training the models, the class weight mechanism was introduced to mitigate the class imbalance.
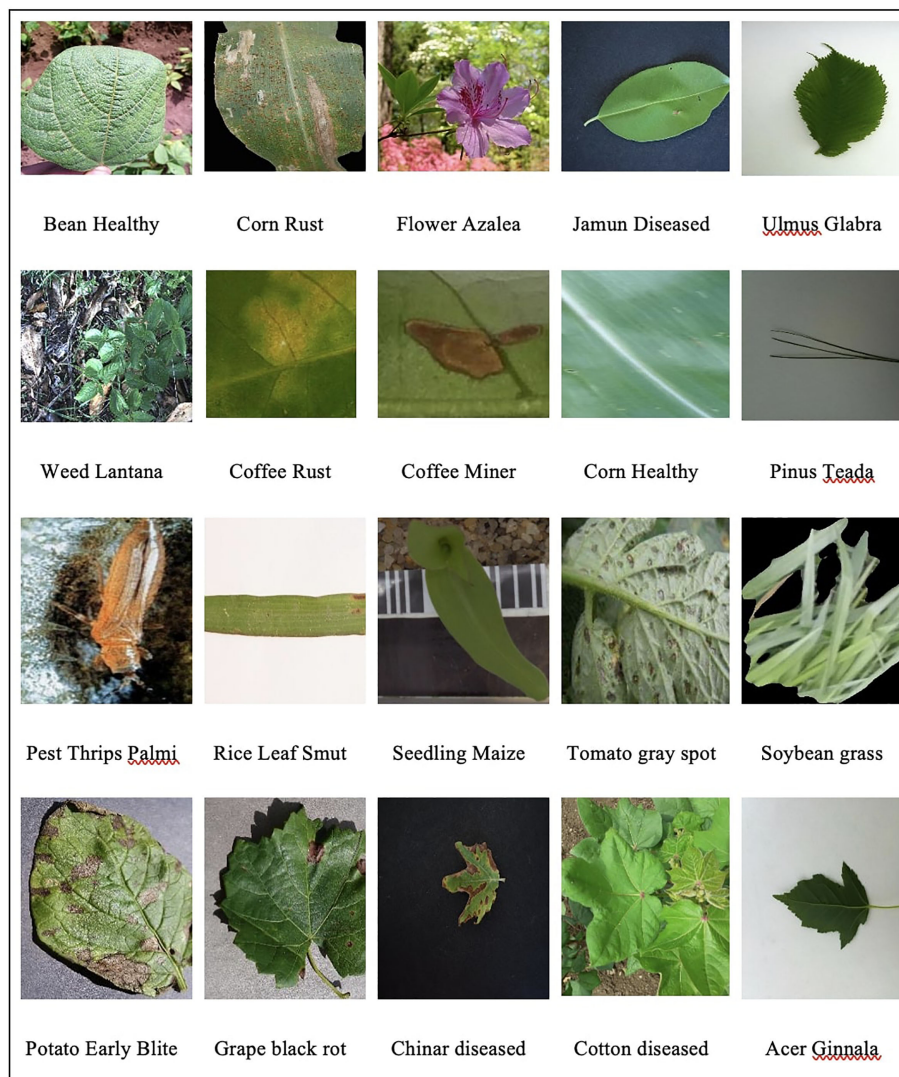

FIGURE 1
Sample images from agrinet dataset.

TABLE 1 Summary of AgriNet dataset per category.

Category	#Images	#Classes	Average	Median	Description	Reference
Species	52150	309	169	144	12 mushroom, 103 flowers, 194 leaves	(Kumar et al., 2012a; Nilsback and Zisserman, ; Olsen et al., 2019; Söderkvist, 2016; U. C. I. M. Learning, 2016; Huang and Chuang, 2020)
Pests & Weeds	38305	33	1161	672	8 weeds,5 soybean weeds,8 pests,12 seedling	(Giselsson et al., 2017; Peccia, 2018; J and Gopal, 2019; Singh et al., 2020b)
Diseases	69687	81	1700	491	30 species	(J and Gopal, 2019; Rauf et al., 2019; Krohling et al., 2019; Chouhan et al., 2019; Singh et al., 2020b; D3v, 2020; Huang and Chang, 2020; Makerere AI Lab, 2020)

Methods

Data preprocessing

The dataset is a collection of images from multiple sources. All images were converted to JPEG format and resized to 224x224 pixels which is the size recommended for the deep learning architectures used. The dataset was then split into 70% train,10% validation, and 20% test. To increase the dataset size and ensure that the model is more robust in classifying images when visual effects are modified, image augmentation was applied to the training set. The augmented images were generated through varying brightness, rotation, width shift, height shift, vertical flip, zoom, and shear.

Convolution neural network

A ConvNet is a sequence of layers where in every layer of a ConvNet one volume of activations is transformed to another volume through a differentiable function (Stanford, a). Three main types of layers are stacked to build a ConvNet architecture: Convolutional Layer, Pooling Layer, and Fully-Connected Layer (Stanford, a). First, a Convolution layer computes the output of neurons that are connected to local regions in the input. Then, an activation function is applied, such as ReLU, which is max (0, x) thresholding at zero (Stanford, a). After that, a pooling layer performs down sampling operation along the spatial dimensions (width, height). Finally, the Fully-Connected layer is a classical neural network layer that computes the class scores (Stanford, a).

Deep learning architectures

Deep learning architectures that were frequently used in agricultural research were selected to train the AgriNet dataset.

VGG16 and VGG19

VGG is named for the Visual Geometry Group at Oxford and was introduced by Karen Simonyan and Andrew Zisserman in 2014 (Simonyan and Zisserman, 2015). The main contribution of this model was the usage of small-sized 3x3 convolutional filters. Pooling was done using Max-pooling over a 2 x 2-pixel window, with a stride of 2. VGG16 is the winning architecture of the ICLRLSVRC-2014 competition, having a top accuracy of 71.3% and a top-5 accuracy of 90.1% (Simonyan and Zisserman, 2015). The model has a depth of 16 and 143 million parameters. The main difference between VGG16 and VGG19, which was ranked second in the competition, is the model depth which is 19 in VGG19 (Simonyan and Zisserman, 2015). VGG19 achieved top accuracy of 71.3% and top 5 accuracies of 90% while having 138 million parameters (Simonyan and Zisserman, 2015).

Inception-v3

The inception model was introduced in 2012 by Szegedy et al. where the main contribution was "going deeper". The model proposed was 27 layers deep, including inception layers. The inception layer is a combination of a (1×1 Convolutional layer, 3×3 Convolutional layer, 5×5 Convolutional layer) with their output filter banks concatenated into a single output vector forming the input of the next stage (Szegedy et al., 2015). Inception-v3 was introduced in 2016 as a convolutional neural network architecture from the Inception family with several improvements including usage of factorized 7 x 7 convolutions, label smoothing, and the use of an auxiliary classifier to propagate label information lower down the network (Szegedy et al., 2015). Those improvements resulted in a top accuracy of 77.9% and a top5 accuracy of 93.7%. It is 159 layers deep and has 23 million parameters (Szegedy et al., 2015).

Xception

Xception model was proposed by Chollet et al. in 2017. It stands for "extreme inception" taking the principle of inception to an extreme. It is a convolutional neural network architecture that relies solely on depth-wise separable convolution layers (Chollet, 2017; Akhtar, 2021). The main difference between inception and Xception is that in inception, 1x1 convolutions were used to compress the original input, and from each of those input spaces different type of filters was used on each of the depth space. On the other hand, Xception reverses this step where filters are applied followed by compression. The second difference is the absence of non-linearities in Xception compared to the usage of ReLU in inception (Chollet, 2017; Akhtar, 2021). The Xception achieved a top accuracy 79% of and a top5 accuracy 94.5% of while having 22.9M parameters and a depth of 126.

InceptionResNetv2

InceptionResNetv2 was proposed by Szegedy et al. in 2016 and builds on the Inception family of architectures but incorporates residual connections by replacing the filter concatenation stage of the Inception architecture (Elhamraoui et al., 2020; Szegedy et al., 2017). Residual connections allow shortcuts in the model leading to better performance while simplifying the Inception blocks (Elhamraoui et al., 2020; Szegedy et al., 2017). The model achieved top accuracy of 80.3% and a top 5 accuracy of 95.4% while having 55.9M parameters and a depth of 572.

Thus, each of the used architecture has its benefits depending on the targeted applications. A detailed comparison of the models is presented in Table 2. Note that Xception model has the smallest size of 88 MB while InceptionResNet-v2 achieved the highest top1-accuracy and top5-acuuracy of 0.803 and 0.953 respectively. In terms of depth and parameters, Xception model has the smallest number of parameters of 2291480 though VGG16 has the shortest depth of 23 layers.

Transfer learning

Transfer learning is the state-of-the-art approach with scarce data applications. The common approach for vision-based application is to train a ConvNet on a very large dataset (for example, ImageNet, which contains 1.2 million images with 1000 categories), and then use the ConvNet either as an initialization or a fixed feature extractor for the task of interest Singh, (2021). Three Transfer Learning methods exist:

ConvNet as fixed feature extractor

This is done by removing the fully connected layer from the ConvNet pretrained on a generic dataset (ex. ImageNet), then treating the rest of the ConvNet as a fixed feature extractor for the new dataset.

Fine-tuning the ConvNet

The second approach adds to the first approach by fine-tuning the weights of the pretrained network by continuing the backpropagation (Singh, 2021). Although retraining the whole model is possible, usually, some of the earlier layers are kept and we only fine-tune some higher-level portion of the network. This is because features of a ConvNet contain more generic features

like edges in the first layers, but later layers of the ConvNet are more detail-oriented toward the pretrained model's classes (Singh, 2021).

Pretrained models

Final ConvNet checkpoints are frequently released to assist in fine-tuning tasks since modern ConvNets are time-consuming. For example, it takes 2-3 weeks to train a ConvNet across multiple GPUs on ImageNet (Singh, 2021).

First, transfer learning was applied on ImageNet pretrained models, where ImageNet was the generic dataset and AgriNet was the target dataset (Figure 2). After training the AgriNet models, the ¢architectures with their weights were saved and proposed as pretrained models for any other agricultural classification task.

Improving the models' performance

To tackle the bias in the DL models, the severe class imbalance in the dataset, and to improve the models' convergence, three main methods were applied to the five AgriNet architectures:

Class weights

Class imbalance can affect the classification accuracy of small classes compared to large classes. To improve the performance of classification in small datasets, multiple solutions exist including oversampling, under-sampling, and class weight. In AgriNet, class weights were added to all the trained neural networks so that a balance is created between classes during the training process (You et al., 2019).

Decaying learning rate

Learning rate decay is a technique for training neural networks, by starting with a large learning rate and then decaying it multiple times (Srivastava et al., 2014). It aims to improve optimization and generalization (Srivastava et al., 2014). This improvement is an outcome of the fact that an initially large learning rate accelerates training or helps the network escape spurious local minima, and then decaying the learning rate helps the network converge to a local minimum and avoid oscillation (Srivastava et al., 2014).

TABLE 2 Comparison of the imagenet architectures used in agrinet.

Model	Size	Top-1 accuracy	Top-5 accuracy	Parameters	Depth
Xception	**88 MB**	0.790	0.945	**22,910,480**	126
VGG19	549 MB	0.713	0.900	143,667,240	26
InceptionResNe-v2	215 MB	**0.803**	**0.953**	55,873,736	572
Inception-v3	92 MB	0.779	0.937	23,851,784	159
VGG16	528 MB	0.713	0.901	138,357,544	**23**

Bold values, for size bold is smallest, for accuracy bold is largest value.

FIGURE 2
Transfer learning.
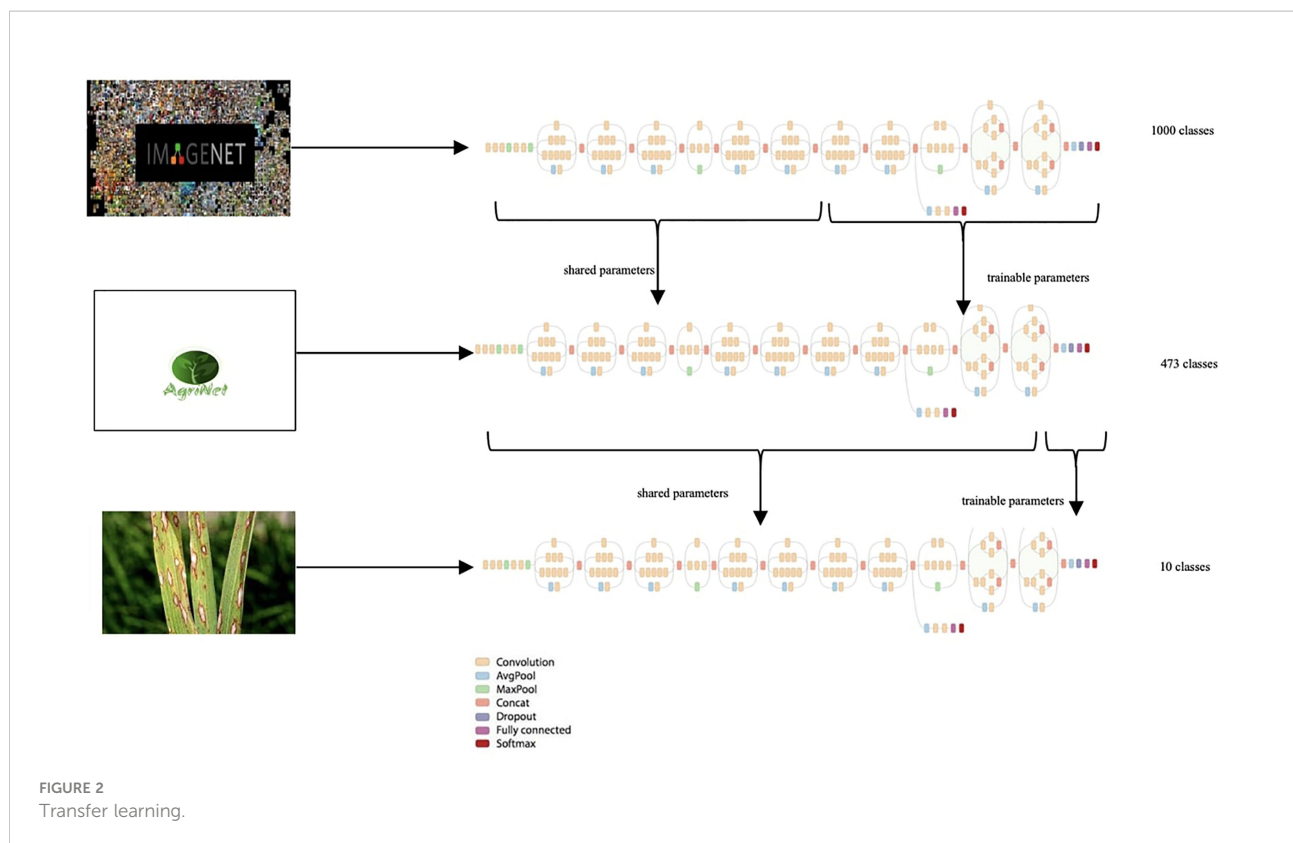
Dropout

Dropout is a regularization method that approximates training a large number of neural networks with different architectures in parallel (Battini, 2018). It was proposed by Srivastava et al. to resolve the overfitting problem in large DL models. Moreover, the term refers to dropping out units, which means temporarily removing units from the network, along with all its incoming and outgoing connections (Battini, 2018). In the simplest case, each unit is retained with a fixed probability p independent of other units, where p can be chosen using a validation set or can simply be set at 0.5, which seems to be close to optimal for a wide range of networks and tasks. For the input units, however, the optimal probability of retention is usually closer to 1 than to 0.5 (Battini, 2018).

Evaluation metrics

Evaluation of the proposed models was based on two metrics.

Accuracy

Accuracy represents the number of correctly classified data instances over the total number of data instances.

F1-score

The F1-score is the harmonic mean of precision and recall. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations while recall is the ratio of correctly predicted positive observations to all observations in the actual class.

Accuracy is the most widely used metric to evaluate the performance of classification models. However, F1-score accompanies accuracy in classification tasks where the dataset is unbalanced.

Results and discussion

Fine tuning AgriNet models

Transfer learning was applied to all AgriNet architectures. For each architecture, multiple experiments were done to propose the most accurate model by changing the number of frozen and trainable layers. The optimizer selected in VGG16 and VGG19 was SGD while in Inception-v3, Xception, and InceptionResNet-v2 Adam optimizer was used. All models were trained on a batch size of 32 (Table 3).

Inception-v3 experiments

The Inception-v3 model constitutes 311 layers. We tested freezing the first 133, 165, 197, 228, 249, and 280 layers. The layers are respectively named mixed4, mixed5, mixed6, mixed7,

TABLE 3 Fine-tuning in agrinet models.

AgriNet architecture	Frozen layers	Trainable layers
Inception-v3	165	146
Xception	116	16
VGG19	17	5
InceptionResNet-v2	400	380
VGG16	15	4

mixed8, and mixed 9. We found that freezing the first 165 layers (mixed5) achieved the highest performance.

Xception experiments: The Xception model has 132 layers. The model was trained while fixing the weights of the first 66, 76, 86, 96, 106, 116, and 126 layers. The layers are respectively named add_5, add_6, add_7, add_8, add_9, add_10, and add_11. We found that fixing the first 116 layers (add_10), achieved the optimum performance.

InceptionResNet-v2 experiments

The InceptionResNet-v2 model constitutes of 780 layers. We tested fixing the first 400, 480, 560, 631, and 711 layers. The layers are respectively named block17_8_mixed, block17_13_mixed, block17_18_mixed, block8_1_mixed, and block8_6_mixed. We found that the first 400 layers (block17_8_mixed) were frozen to achieve the most accurate classification.

VGG16

The VGG16 model has 19 layers. The model was trained while freezing the weights of the first 7, 11, 15, and 19 layers, which are respectively named layer to block2_pool, block3_pool, block4_pool, and block5_pool. We found that fixing the first 15 layers (block4_pool) achieved the highest performance.

VGG19

The VGG19 model constitutes 21 layers. We tested freezing the first 7,12,17, and 21 layers, which are respectively named block2_pool, block3_pool, block4_pool, and block5_pool. We found that the first 17 layers (block4_pool) were frozen to achieve the most accurate classification.

Thus, each of the architectures had its optimum freezing percentage. For Inception-v3 and InceptionResNet-v2, 53% and

51.3% freezing of weights achieved the highest accuracy respectively. On the other hand, for VGG16, VGG19, and Xception, the highest accuracies were achieved when freezing percentages of 78.9,77.3, and 87.9 respectively.

Evaluation of AgriNet models as classification models

Overall networks performance

After training the five architectures, the overall test and per class accuracies and F1-score were reported. Supplementary materials include the total number of images for each class and the per-class test accuracy for each of the five models.

VGG19 surpassed all other models with a test accuracy of 94% and an F1 score of 92%. VGG16 was ranked second, followed by InceptionResNet-v2. However, the Inception-v3 model was the least performing with an average accuracy of 88% and an F1-score of 84% (Table 4).

Another comparison was done for each of the models' sizes, the number of parameters, and Floating-Point Operations (FLOPs). InceptionResNet-v2 had the smallest FLOPs of 375,982,836 operations, followed by Xception with 623,900,414 operations and then VGG19 that reported 718,281,877 operations. Similarly, InceptionResnet-v2 had the smallest number of parameters which is 12,983,584 parameters, followed by VGG16 with 41,888,999 parameters and then by VGG19 with 94,092,935 parameters. For the smallest model size, VGG19 is ranked first with a model size of 159.9MB, followed by VGG16 (180.1MB), and then by InceptionResNet-v2 (980.3MB). Results are displayed in Table 5.

InceptionResNetv2 achieved the best compromise between accuracy, F1-score, FLOPs, number of parameters, and model size. Additionally,

TABLE 4 AgriNet models evaluation.

	Train accuracy	Val accuracy	Test accuracy	F1-score (macro average)
Inception-v3	93.69	88.36	88	84
Xception	94.1	87.73	89	85
VGG19	**95.76**	**93.84**	**94**	**92**
InceptionResNet- v2	91.03	89.82	90	87
VGG16	91.11	91.55	92	90

Bold values, highest accuracy/f1-score.

TABLE 5 Comparing imagenet architectures used.

	FLOPs	Total number of parameters-agrinet	Size-agrinet
VGG16	802,046,505	41,888,999	**159.9 MB**
VGG19	718,281,877	94,092,935	180.1 MB
Xception	623,900,414	124,056,527	1.28 GB
Inception-v3	951,318,693	74,666,183	816.2 MB
InceptionResNet-v2	**375,982,836**	**12,983,584**	980.3 MB

Bold values, highest accuracy/f1-score.

the model has a size of 980.3MB, the lowest FLOPs and number of parameters, it achieved a 90% accuracy and an 87% F1 score.

Categorical evaluation of the models

The AgriNet Dataset consists of three main categories: species, weeds and pests, and diseases. Evaluation per category analysis was performed on each of the architectures proposed. VGG19 outperformed other architectures in species recognition and pests, weeds, and diseases detection tasks (Table 6).

Species classification task

VGG19 achieved the highest accuracies in flowers, leaves, and mushrooms detection. Flowers classes are a combination of the VGG flowers dataset (103 classes) combined with the TensorFlow flowers dataset (5 classes merged with classes of the VGG flowers dataset). The VGG flowers dataset achieved a 70.4% accuracy (Nilsback and Zisserman,). As shown in Table 7, all AgriNet models outperformed the baseline model in the flower classification task. Similarly, leaves are majorly composed of the Leafsnap dataset which achieved an accuracy of 70.8 in (Kumar et al., 2012b). In the mushrooms classification task, the highest accuracy of 75.77% was achieved by VGG19. The low accuracy of mushrooms classification compared to other classification tasks in AgriNet is mainly caused by the different

image patterns of mushrooms compared to leaves and flowers constituting all other classes.

Pests and weeds classification task

The five pretrained models were able to achieve high accuracies in classifying pests and weeds as shown in Table 8. For weed images retrieved from the deep weeds dataset, the highest macro average accuracy of 89.9% was achieved by VGG19 while (Rahman et al., 2018) achieved the highest macro average accuracy of 74.93% using ResNet50. Same for weed seedlings, VGG19 achieved the top performance while reaching an accuracy of 98.62%. Moreover, for soybean weeds, all models achieved around 99% accuracy. Finally, in pests classification, Xception had the minimum accuracy of 94.87% and VGG19 had the highest accuracy of 98.62%.

Plant diseases classification task

Plant diseases classes were merged from different datasets. VGG19 was able to classify the largest number of plant diseases most accurately with a macro-average accuracy of 92.51%. VGG16 achieved top-class accuracy in a smaller number of classes than VGG19 and was ranked second with a macro-average accuracy of 90.84%. Sample macro average accuracies are presented in Table 9.

Evaluation of AgriNet models as pretrained models

Evaluation on rice pests and diseases dataset

To evaluate the superiority of the proposed models, transfer learning was applied using ImageNet and AgriNet weights for the five ImageNet architectures on the rice pest and plant diseases dataset (Kour and Arora, 2019). The dataset was split into 70% training, 10% validation, and 20% test sets. This dataset is a collection of 1426 field images of rice pests and diseases

TABLE 6 Categorical macro-average test accuracy.

Category	Species	Pests and weeds	Plant diseases
VGG16	87.38	91.56	90.84
VGG19	**91.96**	**94.71**	**92.51**
Xception	84.94	85.55	85.6
Inception-v3	82.15	86.99	87.93
InceptionResNet-v2	84.28	90.23	89.5

Bold values, highest accuracy/f1-score.

TABLE 7 Macro average test accuracies for species category on test set.

Category	#classes	Inception-v3	Xception	VGG16	VGG19	InceptionResNet-v2
Flowers	103	80.63	86.92	86.88	**93.47**	80.63
Leaves	194	86.42	82.65	88.67	**91.02**	86.84
Mushrooms	12	69.83	72.44	70.71	**75.77**	73.7

Bold values, highest accuracy/f1-score.

TABLE 8 Macro average test accuracies for pests and weeds category on test set.

Category	#classes	Inception-v3	Xception	VGG16	VGG19	InceptionResNet-v2
Weeds	8	79.92	73.2	86.85	**89.99**	85.61
Seedling	12	76.72	81.88	88.66	**92.72**	86.35
Pests	8	94.95	94.87	95.74	**98.62**	94.9
Soybean	5	99.2	98.82	99.34	**99.43**	**99.43**

Bold values, highest accuracy/f1-score.

TABLE 9 Macro average test accuracies for some species used in plant diseases category on test set.

Category	#classes	Inception-v3	Xception	VGG16	VGG19	InceptionResNet-v2
Apple	3	83.06	86.21	91.27	**93.23**	88.98
Bean	3	90.8	88.5	90.03	**93.89**	95.05
Cassava	5	67.77	64.1	71.3	**72.75**	69.29
Cherry	2	95.2	93.53	97.88	**98.07**	97.6
Coffee	5	94.14	92.82	**97.52**	97.26	95.67
Corn	4	93.07	92.23	93	**95.51**	90.68
Cotton	4	95.62	91.38	**97.6**	96.5	95.63
Guava	2	94.13	90.66	95.42	**95.62**	94.33
Grape	4	93.69	93.83	**98.9**	98.38	97.83
Citrus	2	90.3	98.09	93.87	**99.59**	89.57
Mango	2	86.8	91.5	**98.11**	96.64	93.34
Potato	2	87.1	88.28	94.73	**95.20**	85.17
Rice	3	83.33	75	83.33	**87.5**	75
Strawberry	2	95.05	97.97	97.75	**99.54**	99.54
Tomato	12	78.5	73.3	84.95	**87.04**	81.09

Bold values, highest accuracy/f1-score.

collected from paddy fields of Bangladesh Rice Research Institute (BRRI) for 7 months (Table 10).

AgriNet models achieved higher accuracies than ImageNet models. In VGG16, the model achieved a 90% accuracy using AgriNet weights, compared to 83% for ImageNet weights (Figure 3). On the VGG19 side, using AgriNet weights resulted in an 88% accuracy compared to 83% accuracy using ImageNet weights. Although VGG19 achieved the highest accuracy on the AgriNet dataset, VGG16 performed better on the rice pest and plant diseases dataset. This can be caused by specific image features that vary between a dataset and another. Thus, it is recommended to evaluate any

TABLE 10 Rice pest and diseases dataset description.

Class	Number of images
False Smut	93
Brown Plant Hopper	71
Bacterial Leaf Blight	138
Neck Blast	286
Stemborer	201
Hispa	73
Sheath Blight and/or Sheath Rot	219
Brown Spot	111

agricultural dataset on multiple AgriNet architectures to achieve the best performance possible. It should be noted that the above accuracies on the dataset resulted after freezing the models and only training the dense layers. Further experiments can result in higher accuracies.

Evaluation on plant diseases of Kashmir dataset

The plant diseases dataset of Kashmir contains 2136 images for eight plant species: Apple, Apricot, Cherry, Cranberry, Grapes, Peach, Pear, and Walnut with a total of 1201 healthy images and 935 diseased images (Table 11) (52). The dataset was split into 70% training, 10% validation, and 20% test set. Similar to the case of the pest and plant diseases dataset, VGG16 achieved the highest accuracy and F1-score for both ImageNet and AgriNet models. All five AgriNet models achieved higher accuracies than ImageNet models. Moreover, the highest AgriNet accuracy reported was 83% compared to 70% in ImageNet on VGG16 as mentioned above (Figure 4).

Conclusion

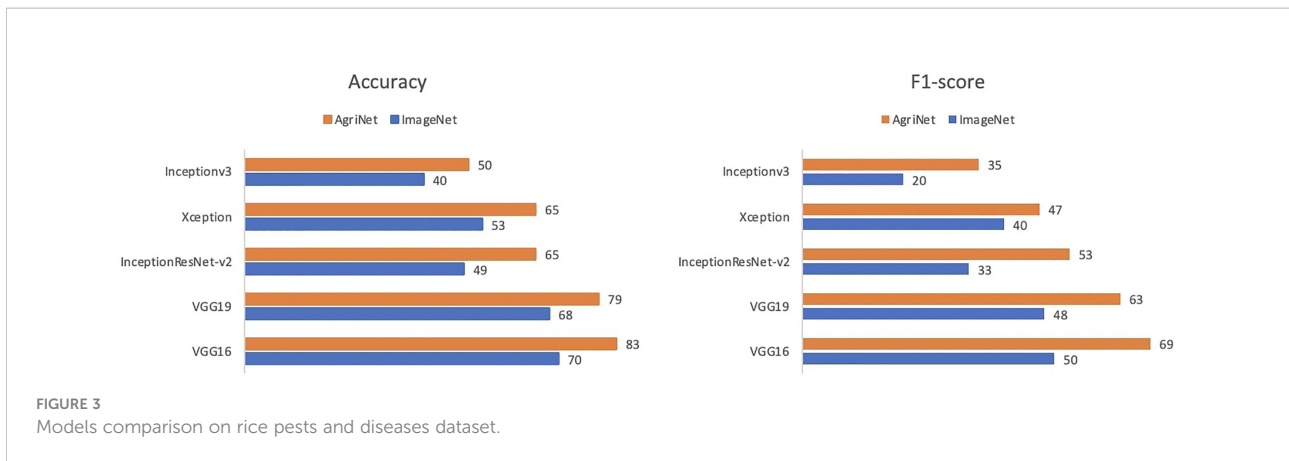In this paper, we present AgriNet dataset and AgriNet models, a collection 160k agriculture images dataset and a set

FIGURE 3
Models comparison on rice pests and diseases dataset.

TABLE 11 Plant diseases of kashmir dataset description.

Class	Number of images
Apple Healthy	93
Apple Diseased	100
Apricot Healthy	86
Apricot Diseased	100
Cherry Healthy	82
Cherry Diseased	95
Cranberry Healthy	100
Cranberry Diseased	94
Grapes Healthy	100
Grapes Diseased	9
Peach Healthy	100
Peach Diseased	18
Pear Healthy	100
Pear Diseased	58
Walnut Healthy	93
Walnut Diseased	100

of five agriculture-domain specific pretrained models respectively. VGG architectures achieved the highest accuracy with 94% accuracy for VGG19 and 92% accuracy for VGG16. InceptionResNet-v2 had the best compromise between the model's performance, and the computational cost through the number of trainable parameters, FLOPs, added to the model's size. In addition, the superiority of the proposed models was evaluated by comparing the AgriNet models with the original ImageNet models on two external pest and plant diseases datasets.VGG architectures resulted in best performance in both ImageNet and AgriNet models, where AgriNet surpassed the ImageNet models with accuracy increase of 18.6% and 8.4% using VGG16 for Kashmir dataset and rice dataset respectively.Further advancements to the AgriNet project include training the AgriNet dataset on more recent convolutional neural networks architectures, expanding pretraining to vision transformers, and increasing the dataset size through adding extra datasets or through applying advanced image augmentation techniques. However, adding additional datasets is restricted to the limited number of agricultural
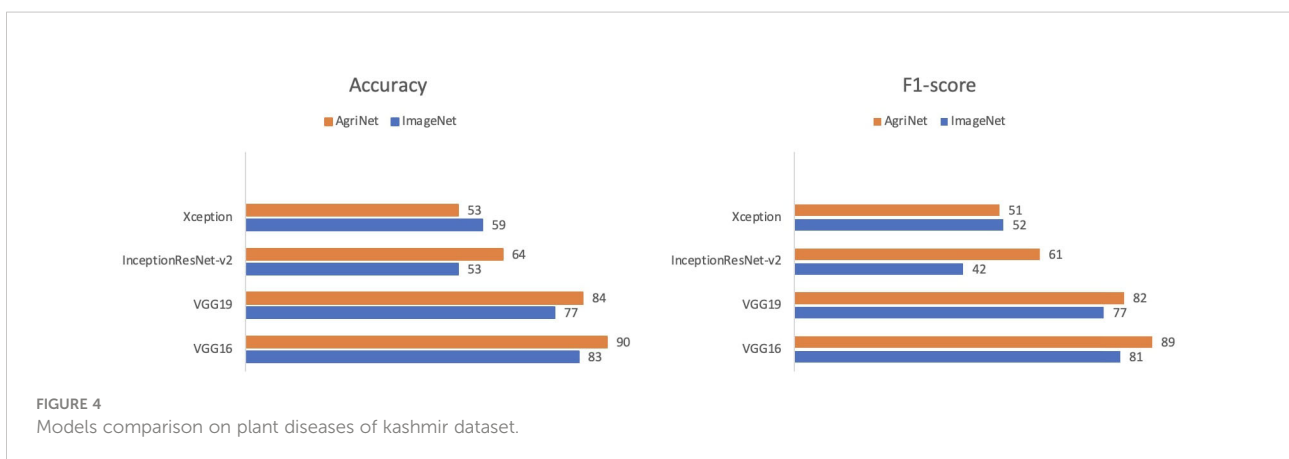

FIGURE 4
Models comparison on plant diseases of kashmir dataset.

public datasets, which urges the research community in retrieving private datasets to public status whenever possible.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material. Further inquiries can be directed to the corresponding author. The models are available here; Dataset will be publicly available on Kaggle. AgriNet Models link: https://drive.google.com/drive/folders/183REzCkgMSI0nlWXb4Y2APdXujSsN_em?usp=sharing.

Author contributions

ZA and MA contributed data collection, methodology, and models evaluation. Both contributed to manuscript revision, read, and approved the submitted version. MA is the advisor of ZA and was following up over all the project tasks. Both authors contributed to the article and approved the submitted version. ZA collected the dataset, proposed hte methodology, trained the models, and evaluated the models, all under MA's supervision. The project is based on MA's idea where she supervised the research including methodology updated and revised the manuscript for several times.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2022.992700/full#supplementary-material

References

Akhtar, Z. (2021) *Xception: Deep learning with depth-wise separable convolutions* (OpenGenus IQ: Computing Expertise & Legacy). Available at: https://iq.opengenus.org/xception-model/ (Accessed 27-Aug-2021).

(2016) *Improving inception and image classification in TensorFlow* (Google AI Blog). Available at: https://ai.googleblog.com/2016/08/improving-inception-and-image.html (Accessed 27-Aug-2021).

Battini, D. (2018) *Implementing drop out regularization in neural networks* (Tech). Available at: https://www.tech-quantum.com/implementing-drop-out-regularization-in-neural-networks/ (Accessed 27-Aug-2021).

Cassava disease classification (Kaggle). Available at: https://www.kaggle.com/c/cassava-disease/overview (Accessed 27-Aug-2021).

Chollet, F. (2017). "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1800–1807.

Chouhan, U. P., Singh, A. Kaul, and Jain, S., "A Data Repository of Leaf Images: Practice towards Plant Conservation with Plant Pathology," 2019 4th International Conference on Information Systems and Computer Networks (ISCON), 2019, pp. 700–707. doi: 10.1109/ISCON47742.2019.9036158

D3v, (2020) *Cotton disease dataset* (Kaggle). Available at: https://www.kaggle.com/janmejaybhoi/cotton-disease-dataset (Accessed 27-Aug-2021).

Elhamraoui, Z. (2020) *Inceptionresnetv2 simple introduction* (Medium). Available at: https://medium.com/@zahraelhamraoui1997/inceptionresnetv2-simple-introduction-9a2000edcdb6 (Accessed 27-Aug-2021).

Gandorfer, M., Christa, H., Nadja El, B., Marianne, C., Thomas, A., Helga, F., et al. (2022). "Künstliche intelligenz in der agrar-und ernährungswirtschaft," in *Lecture Notes in Informatics (LNI)*, Bonn (Gesellschaft für Informatik).

Giselsson, T. M., Jørgensen, R., Jensen, P., Dyrmann, M., and Midtiby, H. (2017). A public image database for benchmark of plant seedling classification algorithms. *ArXiv*.

Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intell. Syst.* doi: 10.1109/MIS.2009.36

Huang, M.-L., and Chang, Y.-H. (2020). Dataset of tomato leaves. *Mendeley Data* 1. doi: 10.17632/ngdgg79rzb.1

Huang, M.-L., and Chuang, T. C. (2020). A database of eight common tomato pest images. *Mendeley Data* 1. doi: 10.17632/s62zm6djd2.1

J, A. P., and Gopal, G. (2019). Data for identification of plant leaf diseases using a 9-layer deep convolutional neural network. (PlantaeK: A leaf database of native plants of Jammu and Kashmir) *Mendeley Data* 1. 76, 323–338. https://doi.org/10.1016/j.compeleceng.2019.04.011.(https://www.sciencedirect.com/science/article/pii/S0045790619300023

Kour, V. P., and Arora, S. (2019). PlantaeK: A leaf database of native plants of jammu and Kashmir. *Mendeley Data* 2. doi: 10.17632/t6j2h22jpx.2

Krohling Renato, A., Esgario Guilherme, J. M., and Ventura José, A. (2019), "BRACOL - A Brazilian Arabica Coffee Leaf images dataset to identification and quantification of coffee diseases and pests", Mendeley Data, V1, doi: 10.17632/yy2k5y8mxg.1

Kumar, N., Belhumeur, P. N., Biswas, A., Jacobs, D. W., Kress, W. J., Lopez, I. C., et al. (2012a) "Leafsnap: A computer vision system for automatic plant species identification," in *European Conference on Computer Vision*. Available at: http://leafsnap.com/dataset/.

Kumar, N., Belhumeur, P., Biswas, A., Jacobs, D., Kress, W., Lopez, I., et al. (2012b). "Leafsnap: A computer vision system for automatic plant species identification," in *European Conference on Computer Vision - ECCV*, Vol. 7573. 502–516. doi: 10.1007/978-3-642-33709-3_36

Madsen, S. L., Mathiassen, S. K., Dyrmann, M., Laursen, M. S., Paz, L.-C., and Jørgensen, R. N. (2020). Open plant phenotype database of common weeds in Denmark. *Remote Sens.* 128, 1246. doi: 10.3390/rs12081246

Makerere AI Lab (2020) *Bean disease dataset*. Available at: https://github.com/AI-Lab-Makerere/ibean/ (Accessed 27-Aug-2021).

Marsh (2020). *Rice leaf diseases dataset* (Kaggle). Available at: https://www.kaggle.com/vbookshelf/rice-leaf-diseases (Accessed 27-Aug-2021).

Mohanty, S. P., Hughes, D. P., and Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Front. Plant Sci.* 7. doi: 10.3389/fpls.2016.01419

Nilsback, M., and Zisserman, A. *102 category flower dataset* (Visual Geometry Group - University of Oxford). Available at: https://www.robots.ox.ac.uk/~vgg/data/flowers/102/ (Accessed 17-Nov-2020).

Nilsback, M.-E., and Zisserman, A.Automated flower classification over a large number of classes in *(2008), Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing.*

Olsen, A., Konovalov, D. A., Philippa, B., et al. (2019). DeepWeeds: A multiclass weed species image dataset for deep learning. *Sci. Rep.* 9, 2058. doi: 10.1038/s41598-018-38343-3

Olsen, A., Konovalov, D. A., Philippa, B., Ridd, P., Wood, J. C., Johns, J., et al. *"DeepWeedsDataset" scientific reports.* Available at: https://nextcloud.qriscloud.org.au/index.php/s/a3KxPawpqkiorST/download.

Peccia, F. (2018) *Weed detection in soybean crops* (Kaggle). Available at: https://www.kaggle.com/fpeccia/weed-detection-in-soybean-crops (Accessed 27-Aug-2021).

Rachman, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., and Hughes, D. (2017). Deep learning for image-based cassava disease detection. *Front. Plant Sci.* 1852. doi: 10.3389/fpls.2017.01852

Rahman, C. R., Arko, P. S., Ali, M. E., Khan, M. A., Wasif, A., Jani, M. R., et al. (2018).). identification and recognition of rice diseases and pests using deep convolutional neural networks. *ArXiv.*

Rauf, H. T., Saleem, B. A., Lali, M.I. U., Khan, A., Sharif, M., and Bukhari, S. A. C. (2019). A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning. *Mendeley Data* 2. doi: 10.17632/3f83gxmv57.2

Rumpf, T., Römer, C., Weis, M., Sökefeld, M., Gerhards, R., and Plümer, L. (2012). Sequential support vector machine classification for small-grain weed species discrimination with special regard to cirsium arvense and galium aparine. *Comput. Electron. Agric.* 80, 89–96. doi: 10.1016/j.compag.2011.10.018

Silva, G.M. S. U. E (2021) *Feeding the world in 2050 and beyond – part 1: Productivity challenges.* Available at: https://www.canr.msu.edu/news/feeding-the-world-in-2050-and-beyond-part.

Simonyan, K., and Zisserman, A. (2015). Very deep convolutional networks for Large-scale image recognition. *CoRR.*

Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., and Batra, N. (2020a). "PlantDoc: A dataset for visual plant disease detection," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD.*

Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., and Batra, N. (2020b) *PlantDoc-dataset* (GitHub). Available at: https://github.com/pratikkayal/PlantDoc-Dataset (Accessed 27-Aug-2021).

Söderkvist, O. (2016) *Swedish Leaf dataset.* Available at: https://www.cvl.isy.liu.se/en/research/datasets/swedish-leaf/ (Accessed 17-Nov-2020).

Singh, K. (2021). How to dealing with imbalanced classes in machine learning. *Analytics Vidhya.* Available at: https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/ (Accessed 27-Aug-2021).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15 (56), 1929–1958.

Stanford *CS231n convolutional neural networks for visual recognition.* Available at: https://cs231n.github.io/convolutional-networks/ (Accessed 27-Aug-2021).

Stanford *CS231n convolutional neural networks for visual recognition.* Available at: https://cs231n.github.io/transfer-learning/ (Accessed 27-Aug-2021).

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-ResNet and the impact of residual connections on learning. *AAAI.* doi: 10.1609/aaai.v31i1.11231

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., et al. (2015). "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 1–9.

Talaviya, T., Shah, D., Patel, N., Yagnik, H., and Shah, M. (2020). Implementation of artificial intelligence in agriculture for optimization of irrigation and application of pesticides and herbicides. *Artif. Intell. Agric.* 4, 58–73.

Tan, K. C., Liu, Y., Ambrose, B., Tulig, M., and Belongie, S. J. The herbarium challenge 2019 dataset. *ArXiv.*

Teimouri, N., Dyrmann, M., Nielsen, P., Mathiassen, S., Somerville, G., and Jørgensen, R. (2018). Weed growth stage estimator using deep convolutional neural networks. *Sensors* 185, 1580. doi: 10.3390/s18051580

Thapa, R., Zhang, K., Snavely, N., Belongie, S., and Khan, A. (2020). The plant pathology challenge 2020 data set to classify foliar disease of apples. *Appl. Plant Sci.* 89. doi: 10.1002/aps3.11390

The TensorFlow Team,Flowers. (2019) Available at: http://download.tensorflow.org/example_images/flower_photos.tgz (Accessed 17-Nov-2020).

U. C. I. M. Learning (2016) *Mushroom classification* (Kaggle). Available at: https://www.kaggle.com/uciml/mushroom-classification (Accessed 27-Aug-2021).

Wäldchen, J., and Mäder, P. (2018). Plant species identification using computer vision techniques: A systematic literature review. *Arch. Comput. Method E* 25, 507–543. doi: 10.1007/s11831-016-9206-z

Yang, S., Zheng, L., Chen, X., Zabawa, L., Zhang, M., and Wang, M. (2022). "Transfer learning from synthetic *In-vitro* soybean pods dataset for *In-situ* segmentation of on-branch soybean pods," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).* 1665–1674.

You, K., Long, M., Jordan, M. I., and Wang, J. (2019). Learning stages: Phenomenon, root cause, mechanism hypothesis, and implications. *ArXiv.*