



# Deep Learning for Plant Identification and Disease Classification from Leaf Images: Multi-prediction Approaches

JIANPING YAO, School of Information and Communication Technology, University of Tasmania, Australia

SON N. TRAN, School of Information Technology, Deakin University, Australia

SAURABH GARG, School of Information and Communications Technology, University of Tasmania, Australia

SAMANTHA SAWYER, Tasmania Institute of Agriculture, University of Tasmania, Australia

---

Deep learning (DL) plays an important role in modern agriculture, especially in plant pathology using leaf images where convolutional neural networks (CNN) are attracting a lot of attention. While numerous reviews have explored the applications of DL within this research domain, there remains a notable absence of an empirical study to offer insightful comparisons due to the employment of varied datasets in the evaluation. Furthermore, a majority of these approaches tend to address the problem as a singular prediction task, overlooking the multifaceted nature of predicting various aspects of plant species and disease types. Lastly, there is an evident need for a more profound consideration of the semantic relationships that underlie plant species and disease types. In this article, we start our study by surveying current DL approaches for plant identification and disease classification. We categorise the approaches into multi-model, multi-label, multi-output, and multi-task, in which different backbone CNNs can be employed. Furthermore, based on the survey of existing approaches in plant pathology and the study of available approaches in machine learning, we propose a new model named Generalised Stacking Multi-output CNN (GSMo-CNN). To investigate the effectiveness of different backbone CNNs and learning approaches, we conduct an intensive experiment on three benchmark datasets Plant Village, Plant Leaves, and PlantDoc. The experimental results demonstrate that InceptionV3 can be a good choice for a backbone CNN as its performance is better than AlexNet, VGG16, ResNet101, EfficientNet, MobileNet, and a custom CNN developed by us. Interestingly, there is empirical evidence to support the hypothesis that using a single model for both tasks can be comparable or better than using two models, one for each task. Finally, we show that the proposed GSMo-CNN achieves state-of-the-art performance on three benchmark datasets.

CCS Concepts: • Computing methodologies → Computer vision tasks; • Applied computing → Agriculture;

Additional Key Words and Phrases: Deep learning, convolutional neural networks, multi-prediction, plant identification, leaf disease classification, plant pathology

---

Jianping and Son are joint first authors (equal contribution).

Authors' addresses: J. Yao and S. Garg, School of Information and Communication Technology, University of Tasmania, TAS 7248, Tasmania, Australia; e-mail: jianping.yao@utas.edu.au; S. N. Tran (Corresponding author), School of Information Technology, Deakin University, VIC 3125, Victoria, Australia; e-mail: son.tran@deakin.edu.au; S. Sawyer, Tasmania Institute of Agriculture, University of Tasmania, Tasmania, Australia.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 0360-0300/2024/02-ART153

<https://doi.org/10.1145/3639816>

**ACM Reference Format:**

Jianping Yao, Son N. Tran, Saurabh Garg, and Samantha Sawyer. 2024. Deep Learning for Plant Identification and Disease Classification from Leaf Images: Multi-prediction Approaches. *ACM Comput. Surv.* 56, 6, Article 153 (February 2024), 37 pages. <https://doi.org/10.1145/3639816>

---

## 1 INTRODUCTION

**Deep learning (DL)** has been a major disruptor in a wide range of real-life applications from autonomous vehicles to clinical decision support. In agriculture, DL approaches have been emerging as a revolutionising tool for sustainable production. In particular, they play an important role in **precision agriculture (PA)/smart agriculture (SA)** [15, 16, 27, 61, 98], including, but not limited to, pest, weeds or irrigation control, automatic harvesting, yield estimation, and plant disease/fruit detection, and so on. DL within the field of plant pathology has garnered significant attention from both the research and industrial sectors. In these domains, the identification of plants and the classification of diseases based on leaf images have witnessed extensive study and practical application [1, 2, 5, 6, 25, 40, 51, 64, 77, 81, 87, 102]. The reason why leaf images are commonly used is leaves are an important part of plants where they participate in the important photosynthesis process and are the most visible part of most plants throughout their growth [15]. Leaf colour, texture, and shape can characterise plant species and, therefore, are useful in plant identification for large-scale plant and crop management [102]. Besides, many plant diseases can be visible from leaves. Leaf disease is one of the important factors that disrupt the health of plants as a whole and are one of the main causes of reduced crop yield. Therefore, it is critical for farmers to detect the occurrence of leaf disease as early as possible and minimise its negative impact or, at least, keep it under control. As a result, much attention has been paid to leaf disease classification using machine learning, as discussed in [105]. However, there is a rising demand from the industry for effective methods to not only classify leaf diseases but also identify plant species, which would be much helpful for large scale management. Therefore, different from [105], this article aims to address the multi-prediction issue.

DL is revolutionising traditional methods for SA/PA, especially in plant and disease classification. Although being popular in the past, traditional approaches have several obvious limitations, mostly caused by the manual costs, such as labours training, requirement of human involvement in many stages of the prediction process, and experts' knowledge, and so on. Manual methods are difficult to detect timely, and the diagnosis may be based on subjective judgment. Nowadays, with the assistance of computer vision, **Internet of Things (IoT)**, and machine learning, we could detect leaf diseases in real-time through various devices, that is, mobile phone applications [67], websites [100], IoT application [12], and smart glasses [68]. For example, [12] introduced a combined approach of IoT and AI models to detect rice blast disease. Compared to our work, this approach uses a custom **convolutional neural network (CNN)**, similar to the first model we implement in 3.3.1, but it work on different input data. In particular, in [12] an IoT platform for soil cultivation was utilised to extract non-image data while our study focuses on image data.

With these advanced technologies, the difficulty of the production operation can be greatly reduced, and we can achieve improvements in accuracy and efficiency. Among them, machine learning has been emerging as a key player, leading the innovation pathway for more effective prediction solutions in plant identification and disease classification. During the earlier stage, researchers employed traditional machine learning approaches, combining feature extraction and classification, with limited successes [7, 8, 19, 26, 39, 43, 79, 84]. The key issue here is that such approaches rely heavily on an independent step to craft features from the images to classify plant species and disease types. The feature engineering process is normally defined by domain

experts or generated by general image processing techniques such as **Scale-invariant Feature Transform (SIFT)** [39], **grey-level co-occurrence matrix (GLCM)** [8, 18, 46, 79, 94], and so on.

As being independent of the later step of learning a classifier, these handcrafted features may not be optimal for prediction. DL has been emerging recently as an effective solution. For example, CNNs can provide an end-to-end classification pipeline to identify plants and classify diseases directly from leaf images. An advantage of CNNs is their ability to learn distinctive features tailored to specific tasks. Furthermore, their adaptability allows for fine-tuning models to maximise effectiveness for unique datasets. Finally, their distributed computation capability makes them an ideal choice for large-scale solutions, enabling efficient processing and analysis of substantial datasets.

Although the application of DL for plant identification and disease classification is not new, we found that most current studies develop CNN models as single prediction classifier. It would be more convenient if there exists a multi-prediction approach for plant species and disease types as they have different indicative features from leaf images. More importantly, it is apparent that besides common types of disease different plant species will be prone to different diseases of their own. Therefore, we hypothesise that by incorporating the two tasks together in one single model, we can improve the prediction performance for each task. To investigate the proposed hypothesis, we employ multi-label/multi-output/multi-task approaches with DL as the core. The main constraint of the task is a requirement for multiple labels of each image to enable the training of deep models. Let us formally define the problem as follows.

**Problem Statement:** Given a data set  $X = \{(x^{(n)}, p^{(n)}, d^{(n)})|n = 1, \dots, N\}$ , where  $x^{(n)} \in \mathbb{R}^{W \times H \times C}$  is an image with a width  $W$ , height  $H$ , and  $C$  channels;  $p^{(n)} \in \mathcal{P}$  is a plant species; and  $d^{(n)} \in \mathcal{D}$  is a plant disease, how to train a DL model  $N$  to accurately identify the plant species and the disease type from an unseen image  $x^*$ ?

A plethora of DL models, mostly CNNs, have been employed for plant and leaf classification separately, including AlexNet [1, 3, 5, 25], GoogLeNet [98, 109], VGG [1–3, 10, 34, 92], Inception [2, 30, 44, 75], ResNet [3, 29, 47, 98], MobileNet [2, 34, 62, 88], and so on. However, there are several questions which are not been studied properly, including (i) which back-bone CNNs can be most useful for plant identification and disease classification; (ii) what other DL approaches can be employed for this task; (iii) whether separate models for plant identification and disease classification perform better than a single model for both tasks; and (iv) whether their performance comparison is consistent across different datasets.

In this article, we aim to answer the above questions, and finally verify our hypothesis, by surveying, developing, and comparing a wide range of CNN architectures. To this end, we solve the problem of plant identification and disease classification by employing and evaluating a variety of DL approaches. We conduct an empirical study to analyse the usefulness of current DL models for plant species identification and leaf disease detection from leaves. We categorise the DL approaches into:

- Multi-model: This is an ensemble of two CNNs models, one is tasked to predict plant species from leaf images while the other is tasked to predict diseases. For completeness, we will use and compare different backbone CNNs for these models, including our custom CNNs, AlexNet, VGG16, ResNet101, EfficientNet, InceptionV3, and MobileNetV2.
- Multi-label: This is a single CNN model with an output of multi labels. Different from the standard multi-label learning [55], in this case, we combine the labels to make a power-set label for the prediction. In other words, we use a combined label to present multiple labels. We also employ different backbone CNNs for this approach.
- Multi-output: This is a single CNN model with multiple output layers, each for a prediction target. Similarly, different backbone CNNs are used for this model as well.

- Multi-task. We can also adapt multi-task learning to this problem. In this case, the target tasks are different but the input data for the tasks will be from the same distribution. Multi-task learning has been employed effectively in many computer vision problems [107, 108] and DL can bolster this class of approaches because one task's features may benefit other tasks' learning [110, 111].

Furthermore, based on the theoretical analysis of the above approaches, we propose a new method to improve the performance of plant identification and disease classification. After that, we review and select suitable benchmark datasets for an extensive experiment. For the empirical analysis, we use three different datasets including Plant Village [36], Plant Leaves [14], and Plant-Doc [83]. The experimental results show that InceptionV3 is the best CNN backbone for both plant identification and disease classification. Interestingly, there is empirical evidence to support the hypothesis stated earlier, as single models for both tasks can achieve better performance than an ensemble of independent models. Finally, our proposed model can achieve state-of-the-art results on the benchmark datasets studied in this article. The contribution of this article is threefold:

- We conduct a detailed literature review through appropriate selection criteria, including recent common machine learning methods and available public datasets in the field of plant identification and disease classification. We generalise the methods to two classes of multi-prediction paradigm (multi-model CNNs, multi-label CNNs) and extend the study by re-introduce several methods available in the machine learning literature (multi-output CNNs, multi-task learning) but have not been studied deeply in the research field of plant pathology. As far as we know, this is the first study to survey and compare different deep multi-prediction techniques.
- We proposed a novel model, namely Generalised Stacking Multi-output CNNs (GSMo-CNN), to improve the performance of plant identification and disease classification. Our model creates multiple output layers and stacks them one on top another to form a chain of classification in a hierarchical manner. By doing this, we aim to represent the relationship between plant species and disease types during the inference process. The experimental results show that GSMo-CNN achieves state-of-the-art performance in all datasets studied in this article.
- Our study reveals several important empirical findings. First, we show that selection of backbone CNNs is critical for achieving good performance and InceptionV3 has the best performance in our study. Second, we demonstrate the advantage of single models for multi-prediction over multiple models. This is an interesting finding as single models are more compact and easier to train. Third, we showcase the effectiveness of combining labels in a hierarchical manner, together with transfer learning we can significantly improve the prediction performance for both plant species and leaf diseases. These findings offer valuable insights for researchers and practitioners in agriculture. It would save their time to search for suitable approaches for more accurate and efficient plant species identification and disease classification, ultimately contributing to improved crop management and disease control. For the sake of reproducibility, we share the source code and datasets in this study at: [https://github.com/funzi-son/plant\\_pathology\\_dl](https://github.com/funzi-son/plant_pathology_dl).

## 2 MACHINE LEARNING FOR PLANT IDENTIFICATION AND/OR DISEASE CLASSIFICATION

### 2.1 Article Selection Criteria

The academic papers selected for this study primarily focus on three key aspects. Firstly, the publication timeline, as illustrated in Table 1, reveals that over 80% of the research articles in this study were published between 2020 and 2023. This timeframe was chosen to ensure the effectiveness and

Table 1. The Publication Years of Referenced Academic Articles (\* Denotes Review Papers)

Year	Paper	Tech	Review	Total
2021–2023	[56], [86], [98], [79], [18], [70]*, [67], [29], [44], [17], [77], [37], [59], [52]*, [16], [27], [33], [39], [30], [23]*, [75], [41], [57]*, [4], [69], [96], [31], [51], [106], [66], [91]	27	4	31
2020	[103], [83], [43], [8], [7], [34], [92], [84], [9], [5], [47], [48], [42], [46], [88], [100], [71], [87], [11], [10], [19], [68], [61]*, [26], [80], [2], [3], [15]*, [24], [12]	28	2	30
2019–2015	[25], [1], [94], [62], [109], [78], [65], [35], [36], [21], [38], [58], [82], [101]	14	0	14

timeliness of this research. Secondly, the degree of citation and the impact factor of the journals (such as Q1 for journals and CORE A/A\* for conferences) were considered. Lastly, the relevance to this research was determined through keyword searches, including “leaf disease”, “plant disease”, “machine learning”, “deep learning”, “classification”, and “detection”. These keywords were used to search in reputable databases such as EBSCO host, Scopus, and Google Scholar.

## 2.2 Traditional Machine Learning Versus Deep Learning

In the earlier years, traditional (shallow) machine learning was used for plant identification and leaf disease classification [2, 4, 9, 48, 78, 92, 103]. This machine learning paradigm in plant pathology consists of two different steps: feature extraction [52, 56, 57, 70] and classifier training [7, 8, 43, 94]. In some cases, researchers also consider including data segmentation after collecting and pre-processing data before applying feature extraction [56, 57]. Among many feature extraction techniques, K-means clustering [11, 43, 46, 65] and GLCM [8, 18, 46, 79, 94] are the most common feature extraction methods. In terms of shallow learning classifiers, **Support Vector Machines (SVMs)** [7, 8, 11, 18, 19, 26, 43, 46, 59, 65, 79, 84] was the most popular, followed by **K-Nearest Neighbor (KNN)** [8, 94], **Random Forest (RF)** [79], **Multilayer Perceptron (MLP)** [39], and Decision Tree [71]. They are all popular in machine learning applications and achieve good performance in leaf disease detection and/or classification. In the case, where data segmentation is used, we will need to detect the region of interest as a part of the data before feeding it to feature extractors and then classifiers. As we can see, the whole process is complex, involving several consecutive steps such as data acquisition, data processing, feature extraction, and prediction [52, 56, 57, 70].

Substantial changes in technology adoption can be seen under the rise of DL. From recent studies, as Table 2 shows, researchers have affirmed that DL can achieve better performance than traditional (non-deep) learning approaches [1, 25, 47, 77, 80, 86, 98]. A class of DL models, known as CNNs, have been widely applied for plant identification and/or leaf disease classification. In [21] the authors employed a popular CNN model named AlexNet for plant classification. In this work, AlexNet was shown to successfully classify grass and broadleaf with average accuracy up to 99%. For leaf disease classification, AlexNet achieved 91.19% accuracy on Apple leaf images [98], 86.5% on Grape leaves [1], and 95.75% on tomato leaf images [5]. Note that, they are separate models, each for a task. A deeper architecture, known as very deep CNNs (or VGG, VGGNet) has shown better performance than AlexNet in image classification tasks. When applied to plant identification, VGG achieved 97% accuracy on Leaf1 Dataset (eight species), 96.57% on Flavia Dataset (32 species), and 85.37% on D-Leaf Dataset (43 species) [38]. VGG for leaf disease classification also received promising results. For example, in grape leaf disease VGG-16 (16 hidden

Table 2. Comparison Between Deep Learning (DL) and Non-deep Learning (Non-DL) Approaches for Plant Pathology Using Leaf Images

Paper	Year	Dataset	Non-DL	DL
[80]	2020	Plant Village (Part)	66.4%	<b>98%</b>
[77]	2021	Plant Village (Part)	90%	<b>96%</b>
[1]	2019	Plant Village (Grape)	97.5%	<b>99%</b>
[25]	2019	Plant Village (Whole)	87.87%	<b>97.87%</b>
[47]	2020	Plant Village (Modified)	88.06%	<b>96.51%</b>
[98]	2021	Plant Village (Apple)	68.73%	<b>97.62%</b>
[5]	2020	Tomato Leaves	92.94%	<b>98.12%</b>
[86]	2021	Citrus Leaves [72]	87%	<b>89.5%</b>
[106]	2023	Citrus Leaves [72]	86%	<b>99.98%</b>

layers) has been applied to many datasets [1, 34, 92]. Notably, in [92] VGG-16 with Average Pooling (GAP) layer achieved 98.4% accuracy. Another common architecture of VGG is VGG with 19 hidden layers, known as VGG-19, which had achieved 96.86% accuracy in tomato leaf disease classification [10].

Another CNN architecture, known as Residual Networks or ResNets, has shown remarkable results in image classification tasks in recent years (it won the 2015 ImageNet competition). In [69], a set of ResNet variants has been employed for plant identification. An evaluation on a real leaf dataset that the authors collected with 15,207 images (of 201 species) is reported as follows, 91.83% (ResNet-50); 92.71% (Res2Net-50); and 92.95% (Res2Net-101). In the case of leaf disease classification, ResNet-50 achieved 98.40% accuracy for tomato leaves [3], a customised ResNet model had 82.78% accuracy on a modified Plant Village dataset [29], ResNet-34 achieved 99.40% accuracy, and 0.9651 F1-score in betel vine leaf disease [47]. In [98] ResNet-20 was tested on apple leaves images and achieved 92.76% accuracy. Last but not least, InceptionV3 is recently emerging as a good model for classification tasks with leaf images. InceptionV3 is the third version of Google’s Inception CNN. In [2], it achieved 63.4% accuracy on tomato leaf diseases and in [44], it achieved 95.41% in rice leaf disease classification. InceptionV3 has been applied to the famous Plant Village dataset and achieved very promising results, as shown in [30] (98.42% accuracy) and in [75] (99.74% accuracy).

Besides the very deep and complex models discussed above, several studies also showed the advantages of light-weight CNNs in plant identification and leaf disease classification. One of the advantages of light-weight CNNs is they can run on low-resource devices, enabling a wider range of applications in SA and PA. For example, MobileNet has been deployed for smartphones and IoT devices. In [2], it achieved 63.75% accuracy in tomato leaf disease classification and in [34] it achieved 86% accuracy in grape leaf disease classification. Another light-weight CNN model is EfficientNet whose different variants (B0, B4, B7) have been used to classify tomato leaf diseases (Plant Village) [17]. To enable high performance for EfficientNet the authors have relabeled the dataset in three subtasks: task 1: healthy and unhealthy; task 2: five classes of leaf state, including bacterial and fungal; task 3: one healthy and nine diseases. The results show that B7 got the best in task 1 (99.95%) and task 2 (99.12%) and B4 got the best in task3 (99.89%).

The above related work applied CNNs separately for plant identification and disease classification from leaf images. Each paper evaluates the CNNs on a different dataset, and sometimes on a modified dataset, making it difficult to benchmark their performance. Different from them, this paper sets up a comprehensive evaluation to provide a comparative view of the CNNs, using multiple datasets.

Several attempts in recent years have shown promising approaches of using a single CNN for multiple tasks [24, 31, 58, 82, 96, 101]. In the case of plant pathology, a CNN can learn to predict both plant species and diseases at the same time with leaf images as input. For example, in [51] the authors showed the effectiveness of conditional multi-task learning for the simultaneous identification of plant species and classification of diseases. Unlike other studies on large-scale multi-task learning, the approach adapts the multi-task learning idea for interrelated labels, where input data from different tasks are from the same distribution. In Plant Village dataset and PlantDoc dataset, the labels are the combination of both species and diseases, that is, Apple Black Rot in Plant Village dataset and Tomato Leaf Late Blight in PlantDoc dataset. This combination deals with the multi-prediction problem by creating a multi-label known as power-set. This can transform a multi-task or multi-label problem to a large-scale multi-class task. A trained model can predict the species and diseases simultaneously, and the predicted results are joint species-disease labels. Several examples employed this idea are shown in Table 3. In [77, 80, 87] the authors applied power-set CNNs on a subset of Plant Village dataset, and in [25, 29, 42, 47] the author worked on the whole or modified Plant Village dataset. All these studied models achieved more than 90% accuracy. For example, according to a study on PlantDoc dataset, with the support of image segmentation and power-set labelling VGG-16 achieved 60.41% accuracy, InceptionV3 achieved 62.06% accuracy, and InceptionResNet V2 achieved 70.53% accuracy [83].

The above studies are all based on multi-class classification tasks, either directly or indirectly through the use of power-set labelling. Recently, researchers have also begun to adopt multi-task learning methods (different from power-set) to directly address the classification of both species and diseases. For example, in [51] the authors proposed conditional multi-task learning approach (CMTL) with InceptionV3 as a backbone CNN to predict both leaf species and diseases. In addition, the article showed that it is possible to use the predicted results of plant species to help improve the prediction of disease. The dataset for the evaluation of CMTL consists of Plant Village, DigiPathos [28], and Web images which made up to 12,290 leaf images with 1,146 joint species-disease labels (311 species and 289 diseases). The experiment showed that, the total Top-1 accuracy for joint prediction of species-disease labels is 69.43%, the total average accuracy is 64.79%, the disease's Top-1 accuracy is 82.96%, and the species' Top-1 accuracy is 78.64%. Although power-set multi-label CNNs and CMTL are promising, there are many other approaches for multi-prediction that have not been deeply explored, which can be beneficial for plant identification and disease classification.

In this article, we survey, implement, and evaluate a wide range of multi-prediction approaches with different CNN backbones that can be employed for predicting both plant species and diseases. We also proposed a new DL architecture with a learning strategy to improve prediction performance.

### 3 METHODOLOGY

#### 3.1 Datasets

Data has a central role in modern AI technologies, including machine learning, DL, and computer vision. In this study, data is also necessary for the comparison of different methods. This section aims to survey and then select suitable datasets for the benchmarking in the next step of experiment and testing. Different from previous studies where, in most cases, only one dataset is used, in this article, we select three data sources to make four evaluation sets to provide a comprehensive comparison of different approaches. The role of image datasets for computer vision in plant pathology is clearly important. In [15], the authors showed that the foremost problem most researchers in this field have been facing is the lack of available data sets. This would greatly affect and restrict the research of machine learning for plant identification and

Table 3. Existing Deep Learning Approaches for Plant Identification and/or Leaf Disease Classification

Paper	Year	Dataset	Categories	Size/Ratio (training/test)	Methods & Accuracy
<b>Plant Identification</b>					
[64]	2022	Coriander & Parsley (Private)	2	100 (70% / 30%)	CNN (90%)
[81]	2022	Private Dataset	79	2591 (80%/20%)	DenseNet-161 (97.3%)
		Middle European Woody Plants [63]	119	(80%/20%)	LR(98.72%)
		Flavia Dataset [102]	32	1703 (80%/20%)	LR(99.58%)
		MalayaKew (MK) Leaf Dataset [50]	44	2816 (80%/20%)	LR(89.35%)
[40]	2021	MK Leaf [50] + Synthetic Dataset	44	(80%/20%)	LR(93.33%)
		Folio Dataset [60]	32	(80%/20%)	LR(98.75%)
		Amazon Forest [99]	9	59,441 (80%/20%)	LR(98.87%)
		LeafSnap Dataset [45]	185	23,147 (80%/20%)	LR(89.27%)
		Swedish Dataset [85]	15	1125 (80%/20%)	LR(100%)
[4]	2021	Swedish Dataset	15	1125 (70%/15%/15% <sup>a</sup> )	ANN (98.99%), KNN (96.68%) & RF (97.12%)
		Flavia Dataset	32	1907 (70%/15%/15% <sup>a</sup> )	ANN (96.29%), KNN (93.79%) & RF (95.24%)
		D-Leaf Dataset [90]	43	1290 (70%/15%/15% <sup>a</sup> )	ANN (95.31%), KNN (86.3%) & RF (91.5%)
[38]	2019	Leaf1 Dataset	8	75 (80%/20%)	GoogLeNet (98%) & VGG-16 (97%)
		Flavia Dataset	32	1879 (80%/20%)	GoogLeNet (94%) & VGG-16 (96.57%)
		D-Leaf Dataset [90]	43	1290 (80%/20%)	GoogLeNet (88.74%) & VGG-16 (85.37%)
<b>Disease Classification</b>					
[1]	2019	Plant Village (Grape)	4	3800/200	CNN (99%), Alexnet (86.5%), VGG16 (97.5%)
[98]	2021	Plant Village (Apple)	4	10888/2801	CNN (97.62%), AlexNet (91.19%), GoogLeNet (95.69%), ResNet-20 (92.76%) & VGG-16 (96.32%)
[5]	2020	Tomato Leaves (Self)	4	N/A	CNN (98.12%), AlexNet (95.75%) & ANN (92.94%)
[2]	2020	Plant Village (Tomato)	10	10,000/7,000/500 <sup>a</sup>	CNN (91.2%), Mobilenet (63.75%), VGG-16 (77.2%) & InceptionV3 (63.4%)
[78]	2018	Plant Village (Tomato)	5	500(80%/20%)	CNN (86%)
[17]	2021	Plant Village (Tomato)	2, 6, 10	5-fold	EfficientNet B0, B4, B7 (97% - 99%)
[3]	2020	Plant Village (Tomato)	10	80%/20%	Xception V4 (99.45%), AlexNet (90.1%), Lenet (88.3%), Resnet (98.40%) & VGG-16 (90.1%)
[106]	2023	Citrus Leaves [72]	5	609	C-GAN(99.6% & 97%), CNN(99.97%&99.98%), SGD(85%&86%) & ACO-CNN (99.98%&99.99%) <sup>b</sup>
[66]	2023	Cotton Leaf Disease Dataset	4	1661	SVM(98.7%&98.7%), CNN(98.8%&98.8%) & Hybrid(98.9%&98.9%) <sup>b</sup>
[91]	2023	Plant Village (Tomato)	10	80%/20%	InceptionV3(94.58%), MobileNetV1(82.7%), MobileNetV2(92.1%) & MX-MLF2(99.61%)
<b>Single model for Plant Identification &amp; Disease Classification</b>					
[80]	2020	Plant Village (Part)	19	N/A	CNN (98%)
[77]	2021	Plant Village (Part)	15	80%/20%	CNN (96%)
[25]	2019	Plant Village (Whole)	38	55,636/1950	CNN (97.87%), AlexNet (87.34%), ResNet (92.56%), VGG16 (92.87%) & InceptionV3 (94.32%)
[47]	2020	Plant Village (Modified)	38	15,200 (80%/20%)	ResNet34 (99.40% & 96.51%) <sup>b</sup>
[83]	2020	PlantDoc (Cropped)	28	80%/20%	VGG-16 (60.41%), InceptionV3 (62.06%) & InceptionResNet V2 (70.53%)
[29]	2021	Plant Village (Modified)	61	31718/4540	Stacking Model (87%), ResNet (82.78%), InceptionNet (82.22%), DenseNet (83.44%) & InceptionResNet (84.07%)
[42]	2020	Plant Village (Whole)	38	N/A	Hybrid (AlexNet + Linear SVM) Model (99.98%), Basic AlexNet (96.34%) & AlexNet with GAP Layer (97.29%)
[87]	2020	Plant Village (Peach, Pepper & Strawberry)	6	70%/30%	Multi Convolutional Layered-based CNN (87.47% - 99.25%) with different epochs (50, 75, 100 & 125)
[51]	2021	Plant Village + Digipathos [28] + Web images	1146 <sup>c</sup>	10324 (80%/20%)	InceptionV3 & Conditional Multi-task Learning (CMTL), Total Top-1 Accuracy: 69.43%, Total Avg Accuracy: 64.79, Disease Top_1: 82.96%, Species Top-1: 78.64%

Training/Validation/Test Amount

Accuracy &amp; F1-score

311 species, 289 diseases

Table 4. Public Leaf Disease Datasets

ID	Dataset	Year	Species	Disease	Link
1	Plant Village	2016	14	22	<a href="https://data.mendeley.com/datasets/tywbtsjrv/1">https://data.mendeley.com/datasets/tywbtsjrv/1</a>
2	Plant Leaves	2019	12	22	<a href="https://data.mendeley.com/datasets/hb74ynkjc/1">https://data.mendeley.com/datasets/hb74ynkjc/1</a>
3	Plantae_k	2019	8	9	<a href="https://data.mendeley.com/datasets/t6j2h22jpx/1">https://data.mendeley.com/datasets/t6j2h22jpx/1</a>
4	PlantDoc	2020	13	17	<a href="https://github.com/pratikkayal/PlantDoc-Dataset">https://github.com/pratikkayal/PlantDoc-Dataset</a>
5	Plant Pathology 2021 - FGVC8	2021	1	6	<a href="https://www.kaggle.com/c/plant-pathology-2021-fgvc8/overview">https://www.kaggle.com/c/plant-pathology-2021-fgvc8/overview</a>
6	Maize Leaf (NLB)	2018	1	2	<a href="https://osf.io/p67rz/">https://osf.io/p67rz/</a>
7	Citrus Leaves	2019	1	5	<a href="https://data.mendeley.com/datasets/3f83gxmv5/2">https://data.mendeley.com/datasets/3f83gxmv5/2</a>
8	Rice Diseases Image Dataset	2019	1	4	<a href="https://www.kaggle.com/minhhuy2810/rice-diseases-image-dataset">https://www.kaggle.com/minhhuy2810/rice-diseases-image-dataset</a>
5	JMuBEN (Arabica Coffee Leaf Images)	2021	1	3	<a href="https://data.mendeley.com/datasets/t2r6rszp5c/1">https://data.mendeley.com/datasets/t2r6rszp5c/1</a>
6	JMuBEN2	2021	1	2	<a href="https://data.mendeley.com/datasets/tgv3zb82nd/1">https://data.mendeley.com/datasets/tgv3zb82nd/1</a>
7	Cassava Diseases	2019	1	5	<a href="https://www.kaggle.com/c/cassava-disease/data">https://www.kaggle.com/c/cassava-disease/data</a>
8	UCI Rice Leaf Diseases	2017	1	3	<a href="https://archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases">https://archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases</a>

disease classification from leaf images. Fortunately, in recent years, several attempts have been made successfully and researchers have devoted themselves to the collection of plant disease data, filling the data availability gap in this area. Table 4 shows recent available public datasets about plant leaf diseases for computer vision research. In the table, “Year” denotes the published year; “Species” denotes the number of plant species in the data; “Disease” denotes the number of diseases available in a dataset, because different plants may have different sets of diseases. In the experiment, we select three datasets and we resize the images in those datasets to an appropriate input shape for a model, for example, the input size for CNN and AlexNet will be  $256 \times 256$ .

In Table 4, the datasets can be divided into two groups, multi-prediction datasets (Plant Village, Plant Leaves, PlantDoc, and Plant Pathology 2021), single-prediction datasets (Maize Leaf, Rice Diseases Image, JMuBEN, Cassava Diseases, and UCI Rice Leaf). A multi-prediction dataset has different plant species (as shown in the Species column in Table 4) and different types of diseases (as shown in the “Disease” column in Table 4). These datasets can be useful for both species and disease classification which will be employed this study to explore the usefulness of multi-prediction approaches and to verify our hypothesis. A single-prediction dataset normally only has a single plant species and a set of disease types for that plant. Therefore, the selected benchmark datasets for our study are detailed as follows:

**3.1.1 Plant Village Dataset.** Plant Village dataset is currently one of the most widely used public datasets for research on leaf disease identification and classification. It has different versions, including an original version and a data augmentation version. The original dataset was published in 2016 [36], it consists of 54,305 images of diseased leaves and healthy leaves from 14 plant species (Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Bell Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, and Tomato). Each species has 1 to 10 classes of related diseases, resulting in 22 unique disease categories totally with some species sharing several diseases. In this dataset, there is a total of 38 unique combinations of species and diseases (e.g., Apple Black Rot), and one additional category about images without leaf (1,143 background images). The data augmentation version was released in 2019 [25]. In this version, the creators have applied six augmentation methods to enrich the data, including image flipping, Gamma correction, noise injection, **principle component analysis (PCA)** colour augmentation, rotation, and scaling, to improve the quality and quantity of the data. As the result, the number of samples in this dataset is 61,486, which increased from 54,305 in the original version. In our study, we carry out the experiment on the original version. We split the data into 70%-10%-20% for training, validation, and test sets, respectively. From Figures 1(a) and 2(a), we can see the class distribution of Plant Village clearly, because Tomato has nine groups of diseases and one group of healthy, it has the

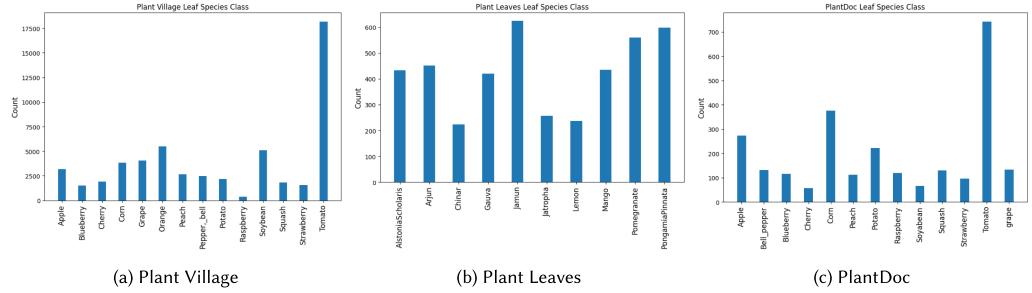


Fig. 1. class distribution for plant species.

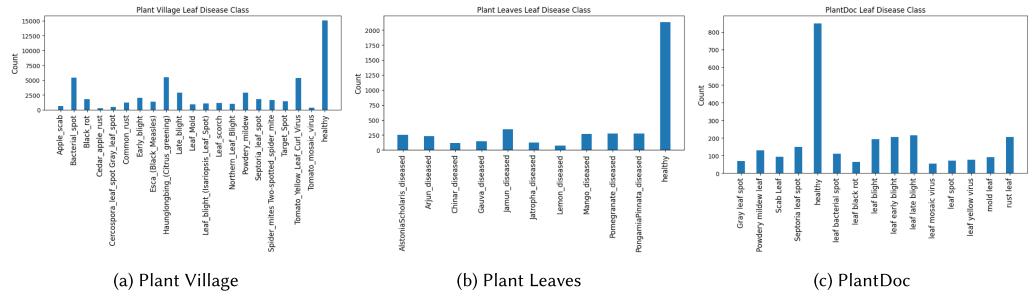


Fig. 2. class distribution for disease types.

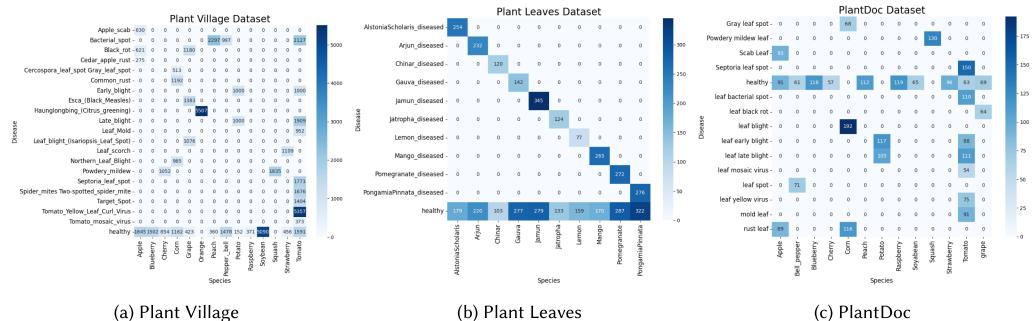


Fig. 3. relationships between species and diseases.

most number of pictures. Figure 3(a) shows the relationships between species and disease of Plant Village.

**3.1.2 Plant Leaves Dataset.** Plant Leaves dataset consists of 4,502 images of healthy and unhealthy leaves divided into 22 categories by species and their health condition. The images are in high-resolution JPG format. The dataset has 12 plant species: AlstoniaScholaris, Arjun, Bael, Basil, Chinar, Gauva, Jamun, Jatropha, Lemon, Mango, Pomegranate, and PongamiaPinnata. We partition the data samples into three different sets with 70% for training, 10% for validation, and 20% for testing. Figures 1(b) and 2(b) show the class distribution of Plant Leaves, because each species has one group of disease and one group of healthy, the healthy category has the most number of pictures. The relationships between species and disease of Plant Leaves have been shown in Figure 3(b).

**3.1.3 PlantDoc Dataset.** Compared to Plant Village Dataset, PlantDoc dataset aims to establish a challenging benchmark with real-field images. The images in Plant Village were taken in a laboratory setup and not in real conditions of cultivation fields that impact the trained model's efficacy in practice [83]. The usefulness of Plant Village may be not fully potential for the development of applications to identify real-world leaf diseases. PlantDoc is a large-scale non-lab data set for leaf disease detection. The images of PlantDoc have cluttered and diverse backgrounds. It has similar categories of plant species and disease types as Plant Village with 2,598 leaf images, 13 plant species, and 17 unique diseases. There are 38 classes for a combination of species and diseases (e.g., Apple Scab Leaf). Originally, the data was partitioned into a training set of 2,360 samples and a small test set of 238 samples. We refer to this set as PlantDoc-1.0. To facilitate deeper comparison (with other works and for future study) we re-partition the data to create another dataset from PlantDoc. We mixed and shuffled the whole dataset and split it into 70%-10%-20% for training, validation, and testing, respectively. This data is referred to as PlantDoc-0.2. We use these two versions of PlantDoc in this study. Figures 1(c) and 2(c) show PlantDoc's class distribution, because Tomato has seven groups of diseases and one group of healthy, it has the most number of leaf images. Figure 3(c) shows the relationships between species and disease of PlantDoc.

## 3.2 Models

In this section, we will survey different DL approaches which have been or can be applied for plant identification and disease classification. The current DL models employed for plant identification or disease classification are CNN models we will describe in Section 3.2.1. However, most of the other approaches we present below have not been applied largely to plant pathology, although they are really relevant and already exist in machine learning literature.

**3.2.1 Backbone CNNs.** In recent years, many different architectures were designed based on CNNs to deal with spatial data such as images and videos, especially in computer vision tasks. With their flexible and computationally efficient architectures, CNNs can be adapted to different scenarios and tasks. In what follows, we re-introduce several CNN models, which are popular and have been proven with excellent performance on image classification tasks. They were tested on benchmark datasets such as CIFAR-100, ImageNet, and also are the state-of-the-art approaches for plant identification and for disease classification in recent research, as shown in Table 3.

**Convolutional Neural Networks (CNN).** CNNs refer to a class of neural networks that employ convolutional operators for information propagation from layers to layers. The convolutional operation is useful for image analysis as it helps neural networks learn local features, which makes CNNs popular for image data [49]. This is also the foundation for a series of deep neural network structures lately. A CNN has an input layer and an output layer, and between these two layers, there are several hidden layers where connections between a lower layer and an upper layer are formed by convolutional operators. The number of hidden layers is chosen depending on the complexity of a task. Recent advanced techniques in CNNs can improve the performance and allow CNNs to be scalable for learning from larger datasets. These include **Rectified Linear units (ReLU)** and other activation functions, pooling (average and max pooling), normalisation (Batch Norm and Layer Norm), and so on. Normally, in a CNN architecture after a series of convolutional layers, there will be several fully-connected layers before the outputs. A fully-connected layer is a normal layer with dense connections (instead of convolutional connections). These layers will process and weave the features from the preceding convolutional layers to make accurate prediction. One of the advantages of CNN is that it can process the raw pixel values from images and learn discriminative features in an end-to-end fashion.

Based on our study on neural networks, we design a custom CNN which can be applied to plant identification or leaf disease classification. The structure of our CNN is similar to [12]. However, different from it, we apply our structure to image data. The input size of the our CNN is set as  $256 \times 256 \times 3$  and it has four convolutional layers in total. There is a batch normalization layer that can normalize the inputs after each convolutional layer. After two convolutional layers and two batch normalization layers, we place a max-pooling layer (this combination is repeated twice). The activation functions for all units are ReLU. On top of these layers, depending on various tasks, we add output layers to perform prediction. The units in these layers are constrained together as a softmax group. Besides the custom CNN, in what follows, we will present the most common off-the-self CNNs, which have been used for image classification in general and plant pathology in specific.

**AlexNet.** This CNN architecture starts with five convolutional layers, and there are two max-pooling layers between the first three convolution layers. In the later stage, AlexNet has three fully connected layers. An interesting feature of AlexNet is its activation functions are designed as non-saturating ReLU. AlexNet was one of the early CNN models that made a breakthrough in image classification, notably being the first CNN to win the ImageNet challenge in 2012.

**VGG16.** This CNN architecture has 16 layers with multiple  $3 \times 3$  kernel-size filters for the convolution. This is different from the first and second large kernel-sized filters in AlexNet. VGG was designed to increase the depth of CNNs where it has several max-pooling layers. In VGG16, there are three large fully connected layers, one with 4,096 units and another with 1,000 units in the later stage of its architecture. In image classification, VGG16 achieved 92.7% top-5 test accuracy in the 2014 ImageNet challenge.

**ResNet101.** This is a powerful structure where we can design and train the model with a lot of layers to gain performance superiority. The key component of ResNet is its “skip connections”, which will skip one or several layers before rejoining to connect to the following layer. This idea can help mitigate the vanishing gradient issue or to deal with the degradation issue. ResNet can help reduce the training error when adding more layers to the CNN, hence providing a good structure for scalable learning [32]. ResNet was the winner of ILSVRC 2015 challenge (a subset of ImageNet).

**InceptionV3.** Inception is a class of CNNs that utilises Inception modules for deeper structure with more efficient computation. The motivation of Inception is to prevent the number of parameters from being too large while building deeper neural networks [89]. Inception consists of asymmetric and symmetric construction blocks. Different layers are employed, including convolution layers, average and max-pooling layers, concatenate layers, and dropout layers and fully connected layers. Each Inception module in this architecture consists of four operations in parallel. The modules will be linked by concatenate layers. The batch Normalization method has been applied to the output of convolutional layers and is widely used in the whole model. In this study, we use the most popular version of Inception, that is, InceptionV3.

**MobileNetV2.** MobileNet is one of the most popular light-weight CNN architectures. It aims to significantly reduce the size of the parameters and to increase the computational speed while maintaining accuracy. It was designed based on the inverted residual structure. However, different from other residual models, its residual block’s input and output are thin bottleneck layers. Also, the light-weight depthwise convolution operator is used in its intermediate expansion layer to reduce the number of parameters [76]. Interestingly enough, the lightweight depthwise convolution can not only reduce the complexity of the model, that is, size of the model, but also greatly reduce computational cost. MobileNet, therefore, is popular for low-resource devices, especially for mobile devices. In this article, we use MobileNetV2.

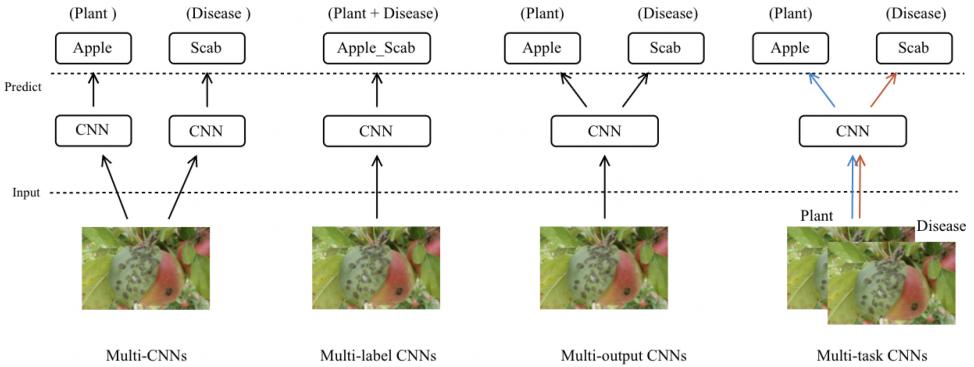


Fig. 4. CNNs approaches for plant identification and disease classification.

**EfficientNet.** Similar to MobileNet, this is one of the light-weight CNN architectures. EfficientNet is based on a scaling approach, which employs fixed and compound coefficients to scale all depth/ width/ resolution dimensions. In this study, we employ EfficientNet with two core parts, one is inverted bottleneck residual blocks (adopted from MobileNetV2) and the other is Squeeze-and-Excitation blocks (SENet). EfficientNet achieved 77.3% top-1 accuracy in the ImageNet dataset.

**Vision Transformer (ViT).** Vision Transformer (ViT) model [22] was released in 2021, based on the idea of Transformers [97] developed from the natural language processing. ViTs can handle a wide range of image sizes without requiring architectural changes. ViTs have a mechanism to capture global context information. They can attend to all image patches simultaneously, allowing them to understand the relationships between distant parts of an image. ViTs leverage the self-attention mechanism, which can capture complex relationships between image patches. They have shown advantages over CNNs in several tasks, for example ViTs perform better than ResNet in image classification on ImageNet and CIFAR-10 [22].

We have presented popular CNN models used in this study. Those models can work alone to predict plant species or disease types, or they can be the backbone in a multi-prediction model to predict these two labels simultaneously, as shown in what follows.

**3.2.2 Multi-Model CNNs.** This is the most straightforward application of CNNs for multi-prediction. A DL model can consist of two independent CNNs, each for a task. In this study, we use two independent CNNs, one for predicting plant species and the other for predicting disease types, as shown in Figure 4. The two CNNs have the same architecture but each has a different set of parameters.

**3.2.3 Multi-Label (power-set) CNNs.** The second approach for multi-prediction is multi-label where the two tasks (plant prediction and disease prediction) are encoded in a single output layer. The most feasible way for it is to join the labels, as known as power-set labelling. This would help transfer a multi-label task to a multi-class task where we can directly apply the backbone CNNs above. In particular, the plant species label and the disease type label will be combined, making a joint label representing both plants and diseases. For example, a power-set label “apple\_scab” can be created from the plant label “apple” and the disease label “scab”. This is the most common method for plant identification and disease classification in the literature. However, our paper is the first to apply and compare different backbones CNNs. Despite being simple, this multi-label approach has a scalable issue when facing a large number of classes for each label. In the worst case, the power-set label will consist of  $|\mathcal{P}| \times |\mathcal{D}|$  classes, where  $|\mathcal{P}|$  is the number of plants and  $|\mathcal{D}|$  is the number of diseases. It may lead to the growth in computational complexity.

**3.2.4 Multi-Output CNNs.** This is a class of CNN models in which we have an input layer and multiple output layers, each for a task as shown in Figure 4 (third model). Theoretically, compared to the multi-label CNNs, multi-output CNNs have fewer parameters because the latter have fewer connections to the output layer(s). The number of total output units in multi-output CNNs for plant & disease prediction is  $|\mathcal{P}| + |\mathcal{D}|$ . The learning in these CNNs is done by optimising the model for all tasks simultaneously. This is also different from the multi-task learning we will discuss below.

**3.2.5 Multi-Task Deep Learning.** Finally, we can adapt multi-task DL architectures for leaf disease and plant type classification. Originally, a multi-task learning problem is to learn a model from different (related) domains, where each task  $i$  is associated with a dataset  $\mathcal{D}_i = \{(x_i^{(n)}, y_i^{(n)}), n = 1, \dots, N_i\}$ . Multi-task learning is very suitable for DL models, the features learned from one task may benefit another task learning [110]. We can utilise those structures for plant identification (task 1) and disease classification (task 2) by sharing input data among different tasks, that is,  $x_1^{(n)} = x_2^{(n)} = x^{(n)}$  and  $y_1^{(n)} = p^{(n)}, y_2^{(n)} = d^{(n)}$ , where  $\{(x^{(n)}, p^{(n)}, d^{(n)})|n = 1, \dots, N\}$  is a plant leaf dataset in our problem statement. Although having the same outputs as multi-output CNNs, the learning in multi-task models is different in which for each data point (an image), they only optimise for a task. In this study, instead of directly using the backbone CNNs (as we will need to implement the learning strategies), we employ the state-of-the-art multi-task models, as shown below.

**Cross-stitch Network.** Cross-stitch is a multi-task approach to learning shared and task-specific representations [58]. To this end, cross-stitch units are designed with a soft-parameter sharing mechanism. These units integrate the features from outputs of multiple networks, each can represent different patterns of the tasks. In other words, they provide soft feature fusions among multiple single-task networks through a linear combination of every layer's activations. As the result, cross-stitch networks would be able to fuel discriminative features across multiple tasks to improve performance, even with a small number of training examples. In [95], the authors found that we should pre-train each single-task network first before stitching these networks for better performance. In our study, the cross-stitch units have been deployed in the middle and the end of two CNNs, one for plant prediction and the other for disease prediction.

**Multi-Task Attention Network (MTAN).** Different from the parallel learning of shared and task-specific features in cross-stitch networks, MTAN will learn shared (global) features first from the images, and then, allow task-specific features to be learned from those global features via soft-attention modules [54]. MTAN is built upon a single shared network with a global feature pool and associate each task with a task-specific soft-attention module. Compared to cross-stitch networks, as MTAN aims to share a general feature pool among different single-task networks it will not be affected by the scalability issue. However, its limitation would be the lack of diversity in task-specific features as they are all produced from the shared pool [95].

**Task Switching Network (TSN).** TSN is based on a task-conditional single-encoder-single-decoder architecture which works with one task at a time while switching between the tasks [74]. TSN's decoder is based on a U-Net architecture and its encoder is ResNet-based backbone (ResNet-18). In TSN, different tasks will be switched by a small task embedding network, one task after another. Meanwhile, the single encoder-decoder would pair all parameters for sharing among the tasks.

**Model-Contrastive Learning (MOON).** MOON is a recent model from federated learning paradigm that aims to leverage the similarity of different tasks' representations to enhance local training of each task [53]. In [53], it is shown that the global features are more useful than the local features learned from each task's dataset. Therefore, MOON proposes a contrastive learning strategy to fine-tune the local representations at model level by maximizing the agreement between

representations of the local model and the global model. The advantages of MOON are its simplicity, effectiveness, and ability to deal with the non-iid data issue. Compared to state-of-the-art approaches, MOON achieves a significant improvement on various image classification tasks.

**3.2.6 Our Model: Generalised Stacking Multi-Output CNNs (GSMo-CNNs).** For completeness, we propose a new model for plant identification and disease classification. The CNN architecture of our model is inspired by the relationship between plant species and disease types. It is commonly known that some diseases may only appear in some particular plants and, therefore, the information about diseases can be useful for the prediction of plants. This reasoning can be applied contrariwise where plant information can be used to predict diseases. We realise this idea by stacking the output (softmax) layers for plant identification and disease classification one on top of another. We generalise the effect of the relationships between plant species and disease types by adding weights on different loss functions at each level of the stack for each output. In what follows, we will show the details of our model.

**Architecture.** The structure of our model is shown in Figure 5. The model is based on the multi-output approaches with all convolutional layers that can be reused from the backbone CNNs we presented earlier. The changes here, as we can see, are (1) the split of dense layers for different tasks; and (b) the stacking of prediction layers. The motivation behind the split of layers is we can use the convolutional layers to learn global features while the dense layers will learn task-specific features. For the stacking strategy of prediction layers, GSMo-CNNs will temporarily infer the probability of plant species and the probability of diseases in the first prediction level. After that, we apply cross connection, that is, we concatenate the probability of predicted plants and the CNN features to make the final prediction of diseases, and similarly, we concatenate the probability of predicted diseases with the CNN features to make the final prediction of plants.

In particular, the proposed model has two sets of fully connected layers (called here as “branches”) after the flatten layer, each branch will connect to a concatenate layer in the later stage. These two branches will produce a pair of predicted plant and disease results first, named here as  $p^{temp}$  (plant temporary) and  $d^{temp}$  (disease temporary). We did not make an actual prediction at this level, instead, we extract the prediction probability from the two softmax layers for the next step. At the concatenate layer, the image features from the flatten layer of the CNN will be combined with the features from softmax layers (probability) in the temporary prediction level ( $p^{temp}$  or  $d^{temp}$ , depending on the branch used). These two soft-max layers will connect to two different concatenate layers, where they will join with the shared CNN’s flatten layer. On top of each concatenate layer, we have another softmax layer for cross prediction. In other words, our GSMo-CNN uses the prediction probability of plant species (from  $p^{temp}$ ) to predict leaf diseases ( $d$ ) and uses the predicted disease ( $d^{temp}$ ) to predict the plant species ( $p$ ) from leaf images. Note that we can use any of the four output layers ( $p$ ,  $d$ ,  $p^{temp}$  and  $d^{temp}$ ) for prediction but we will show in the experiments that by stacking the softmax layers the final output layers would give better performance. The use of probability in the temporary prediction layer (with the *softmax* function), instead of the predicted values (with the *argmax* function), will help smooth

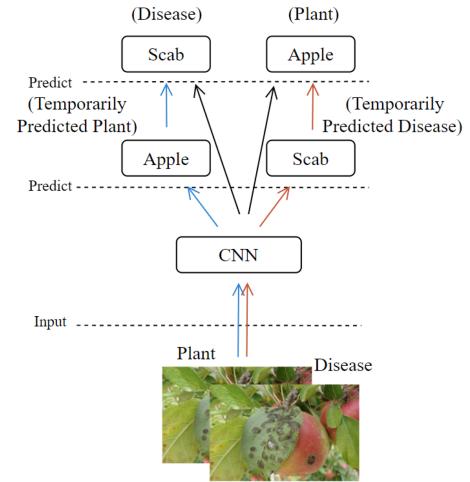


Fig. 5. our new model structure.

the propagation of gradients in the learning. Our model is inspired by classifier chain [73], but the difference here is that the “chain” in GSMo-CNNs is implemented in a stacking fashion with parallel inference instead of sequential inference.

**Training.** For training, as we have four outputs, we aim to optimise the prediction at every output layer. This is because a good estimation of plant species in the temporary prediction layer will help improve the prediction of diseases in the final prediction (top) layer. Similarly, a good estimation of diseases in the temporary prediction layer will help the prediction of plants in the final layer. As the result, we will need four loss functions to minimise. Normally, we can train our model to minimise the losses simultaneously to take advantage of the underlying relationships between plant species and leaf disease. This would also allow us to impose the logical negation constraint that some diseases will not appear on specific plants and vice versa. Furthermore, we are interested in how much relationships and constraints affect the learning. However, there will be an issue associated with such relationships and constraints in the case where the temporary layers do not learn well and make the wrong prediction for the top layers. This will definitely happen during the beginning of the training where, after being initialised, the model will have very low performance (and many mistakes will be made) as it just starts to learn. Therefore, in order to control this impact, we associate each loss with a **balance weight (BW)**. In particular, we train our model by minimising the following total loss function:

$$C = \beta_1 \mathcal{L}(P_1, \hat{P}_1) + \beta_2 \mathcal{L}(P_2, \hat{P}_2) + \delta_1 \mathcal{L}(D_1, \hat{D}_1) + \delta_2 \mathcal{L}(D_2, \hat{D}_2), \quad (1)$$

where  $\mathcal{L}$  is a cross-entropy loss function;  $P_1, P_2$  are the ground truth for the plant species at level 1 and level 2 in the stack, respectively;  $\hat{P}_1, \hat{P}_2$  are the corresponding predicted plant species;  $D_1, D_2$  are the ground truth for the disease types at level 1 and level 2 in the stack, respectively;  $\hat{D}_1, \hat{D}_2$  are the corresponding predicted diseases.  $\beta_1, \beta_2, \delta_1, \delta_2$  are the BW. In the experiment  $\beta_1, \beta_2, \delta_1$ , and  $\delta_2$  are treated as hyper-parameters. The reason for using BW for the losses in each level is that we want to fine-tune the learning in the upper level based on what we have in the lower level. We also apply different BW to the losses on different outputs to see how they influence each other. We would anticipate that these BW would be similar as the prediction of plants would help the prediction of diseases, and vice versa.

**Deduction.** Our model performs two stages of inference for plant species and disease types but only the outcomes of the 2<sup>nd</sup> stage will be used for the final prediction. However, by using the BW in the training, we generalise the multi-output architecture, allowing control of interactions between different outputs and between outputs at different levels. As the result, the multi-model and multi-output approaches are just special cases of our model, as follows: 1) If  $\beta_1 = 1$  and  $\beta_2 = \delta_1 = \delta_2 = 0$  then our model becomes a single CNN for plant identification. In this case, the layer for plant identification at the first level will be used as the output. 2) If  $\delta_1 = 1$  and  $\beta_1 = \beta_2 = \delta_2 = 0$  then our model becomes a single CNN for disease classification. In this case, the layer for disease classification at the first level will be used as the output. 3) If  $\beta_1 = \delta_1 = 1$  and  $\beta_2 = \delta_2 = 0$  our model becomes a regular multi-output CNNs, as shown earlier. The outputs are the two layers in the first prediction level.

### 3.3 Evaluation Metrics

Based on the study of related work and classification tasks, we found that Accuracy and F1-score are the most common evaluation metrics for plant identification and disease classification. Both metrics can be calculated from a confusion matrix [13]. The accuracy metric indicates how accurate a machine learning model is. A prediction is correct if its output is the same as the actual label (ground truth) and accuracy is the ratio of the number of correct predictions to the total number of samples. Generally, accuracy is a sensible metric for classification tasks, however, when

facing the data imbalance issue accuracy will become more biased towards the class with the most number of samples. For example, if 95 out of 100 images are apple leaves, then we can make a simple guess to achieve 95% accuracy. Although not all datasets are imbalanced and sometimes the imbalance issue is not severe, for completeness, we also employ F1-score as another evaluation metric. Accuracy and F1-score are calculated as follows:  $\text{Recall} = \frac{TP}{(TP+FN)}$ ,  $\text{Precision} = \frac{TP}{(TP+FP)}$ ,  $\text{Accuracy} = \frac{(TP+TN)}{(TP+FN+FP+TN)}$ ,  $\text{F1\_score} = \frac{(2*\text{Precision}*\text{Recall})}{(\text{Precision}+\text{Recall})}$ , and  $\text{FPR} = \frac{FP}{(FP+TN)}$ .

Here, TP, TN, FP, FN, and FPR denote True Positives, True Negatives, False Positives, False Negatives, and **False Positive Rate (FPR)**, respectively. As we can see, F1-score is based on both Recall and Precision. It balances the weights of precision and recall, making it a more reliable metric. In this study, F1-score will be the key evaluation metric, for example, used for model selection. Together with Accuracy, they will provide a comprehensive view of the performance of a model. We will evaluate a model based on its performance (Accuracy and F1-score) on plant prediction, disease prediction, and (combined) plant-disease prediction. FPR will be used to assist in evaluating the optimal models for all approaches. FPR represents the number of false positive predictions, and the sum of FP and TN constitutes the total number of true negatives.

### 3.4 Setup

With the models are ready, we are now in a position to prepare for the empirical study. A collection of common and recent CNN models for plant species and diseases with various learning approaches (multi-model, multi-label, multi-output, multi-task, and GSMo-CNNs) will be tested. Backbone models (i.e., CNN, AlexNet, VGG16, ResNet101, EfficientNet, InceptionV3, and MobileNetV2) are implemented in Tensorflow 2.6, and multi-task models are in Pytorch (downloaded and cited from related authors' papers and GitHub). We perform the experiments on three public datasets, including Plant Village, Plant Leaves and PlantDoc, as detailed in Section 3.1. The ratio of training and test sets is 80: 20 and 10% of the training set will be used as a validation set. All hyper-parameters in the experiments were set uniformly across the models and approaches. During the training, we adopt an early stopping method to avoid the problem of over-fitting. In particular, if a model's validation loss has not been decreasing in 50 epochs we stop the training. The input size of leaf images is set to  $256 \times 256 \times 3$  pixels and all samples are in RGB format. Our model selection is done by measuring F1-score on the validation set. For GSMo-CNNs, we search the BW using a coarse grid [0.2, 0.4, 0.6, 0.8] and then narrow down the search to find the best combination for all datasets. We find that  $\beta_1 = 0.1$ ,  $\beta_2 = 0.4$ ,  $\delta_1 = 0.1$ ,  $\delta_2 = 0.5$  are generally good for all three datasets and we report the results of GSMo-CNNs under this configuration. We run each experiment 10 times and report the average accuracy and F1-score with standard deviation.

## 4 EXPERIMENTAL RESULTS

In this section, we report the results of different models on the benchmark datasets and analyse their performance. We divide the results into "Plant Prediction", "Disease Classification", and both (Plant Identification and Disease Classification). In multi-model, multi-label (power-set), multi-output, and our GSMo-CNNs, we evaluate different CNNs backbones, including AlexNet, VGG16, ResNet101, EfficientNet, InceptionV3, MobileNetV2, and our custom CNN. We test two versions of our GSMo-CNNs, one without BW ( $\beta_1 = \beta_2 = \delta_1 = \delta_2 = 1$ ), and the other with BW. To save time for model selection, we choose the best backbone CNN found in the former version (GSMo-CNNs without BW) for the latter. For completeness, we apply transfer learning to our GSMo-CNNs to see if improvement can be achieved. We implement the transfer learning idea by using pre-trained backbone CNNs on the ImageNet dataset to initialise our model and fine-tune it on a leaf image dataset.

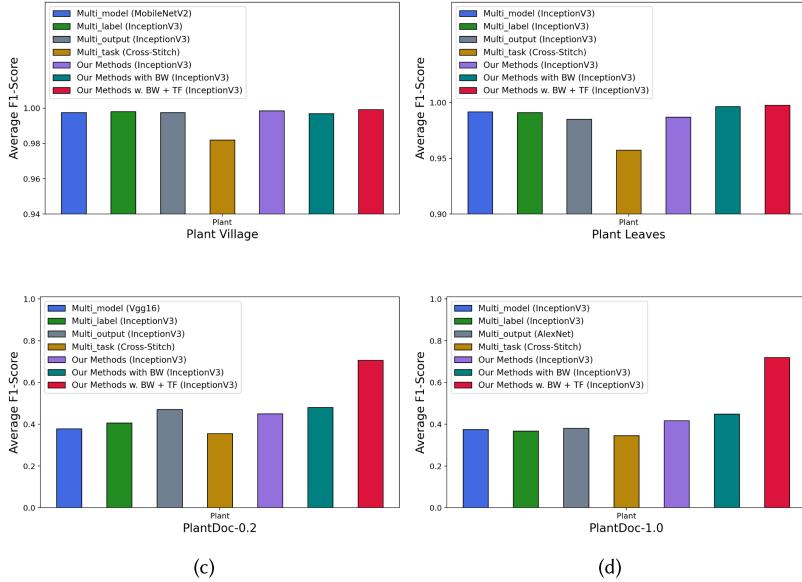


Fig. 6. comparison of the approaches on plant identification.

#### 4.1 Plant Identification

Let us discuss the plant identification task first to evaluate the models and approaches on their ability to predict plant species from leaf images. This is a classification task where we identify plant types among a fixed set of categories (species). Different from previous work on identifying plants using healthy leaves only, the challenge of this task in our study is we have both healthy and diseased leaves images. It is worth noting that identifying plant species from corrupted or damaged leaves is non-trivial. Nevertheless, we expect that an effective classification model should distinguish them accurately. Table 5 shows all the plant type results from the experiment. The results contain the accuracy and F-1 scores of multi-model, multi-label (power-set), multi-output, multi-task models, and GSMo-CNNs on the benchmark datasets with different CNN backbones.

Table 5 and Figure 6 shows that all approaches achieve good performance (more than 90% accuracy and 90% F1-score) on Plant Village and Plant Leaves. These results confirm the effectiveness of DL approaches for classification tasks with image inputs. For PlantDoc (both PlantDoc-0.2 and PlantDoc-1.0), the performance is lower. We anticipate this outcome because the images in Plant Doc are more complex with noisy backgrounds and there can be multiple leaves in one image. Note that, for the sake of the fair evaluation, we do not apply any data processing and augmentation techniques, even though previous research shows that they can improve the performance greatly [1, 25, 77].

In terms of backbone CNNs, as we can see, in multi-model approach, there is an inconsistency of which backbone CNNs has the best performance in all datasets. In particular, MobileNetV2 has the highest accuracy and F1-score in Plant Village, VGG16 is the best in PlantDoc-0.2, while InceptionV3 has the best results in both Plant leaves and PlantDoc-1.0. Different from the multi-model approach, in multi-label and multi-output approaches InceptionV3 clearly shows its advantages with higher performance than other backbones. In PlantDoc-0.2 and PlantDoc-1.0, InceptionV3 achieves the best performance and its advantages are overwhelming. In particular, its accuracy/F1-score on PlantDoc-0.2 are 51.437%/0.47071 and on PlantDoc-1.0 are 43.093%/0.338037,

respectively. In the case of GSMo-CNN, InceptionV3 is also better than other backbone CNNs in Plant Village, Plant Leaves, PlantDoc-2.0, and PlantDoc-1.0.

In terms of common approaches (multi-model, multi-label, multi-output, and multi-task), multi-label (power-set) achieves the best results in Plant Village (Accuracy: 99.469% and F1-score: 0.99469) and multi-model has the highest average accuracy and F1-score (99.175% and 0.99175) in Plant Leaves dataset. Multi-output outperforms multi-model, multi-label, and multi-task in PlantDoc-0.2 (Accuracy: 51.437%, F1-score: 0.4707) and PlantDoc-1.0 (Accuracy: 43.093%, F1-score: 0.38037). Multi-task is inferior to other approaches. This is not surprised as both tasks (plant identification and disease classification) shared the same input data, which is different from original multi-task learning paradigm where different tasks have different sets of data.

For our proposed model GSMo-CNN (denoted as “Our methods (Plant)” in Table 5, we applied 7 backbone CNNs for our new model structure without the BW of each loss (i.e., set  $\beta_1:\beta_2:\delta_1:\delta_2$  as 1:1:1:1). InceptionV3 is also the best backbone in this case. This is the backbone we use for GSMo-CNN with BW. Without the BW, the performance of the GSMo-CNNs is already promising as it achieves better results than multi-model, multi-label, multi-output, and multi-task on Plant Village (Accuracy: 99.850%, F1-score: 0.99850) and on PlantDoc-1.0 (Accuracy: 44.492%, F1-score: 0.41757). With the BW, GSMo-CNNs achieves the best performance in three cases: Plant Leaves, PlantDoc-0.2, and PlantDoc-1.0. The accuracy and F1-score are as follows, Plant Village: 99.687% and 0.99688); Plant Leaves: 99.646% and 0.99647; PlantDoc-0.2: 49.068%, 0.47960; and PlantDoc-1.0: 46.864% and 0.44804.

Finally, when transfer learning is applied we can see improvement in all cases, especially for PlantDoc-0.2 and PlantDoc-1.0. In Plant Village, the best results without transfer learning are 99.850% accuracy and 0.99850 F1-score while transfer learning achieves 99.928% accuracy and 0.99927 F1-score. Huge improvement can be seen in PlantDoc sets. In particular, our model with transfer learning increase the performance from 51.437% accuracy and 0.48051 F1-score to 71.262% and 0.70602 on PlantDoc-0.2, and from 46.992% accuracy and 0.44934 F1-score to 73.644% and 0.71980 on PlantDoc-1.0.

## 4.2 Disease Classification

The disease classification task aims to predict the diseases of plants from leaf images. Note that, “healthy” is also treated as a category of plant diseases. The task will not only identify whether a plant has a disease or not but also need to accurately categorise the disease types from different plants. A model should be able to focus on the diseases and not be confused by the common patterns from leaves of the same type. Table 6 shows the results of different models and approaches, including all the disease classification results from multi-model CNNs, multi-label (power-set) CNNs, multi-output CNNs, multi-task learning approaches and our new model.

As we can see in Table 6 and Figure 7, all approaches achieve good performance on Plant Village and Plant Leaves datasets (most accuracy and F1-score over 80% and 0.80). In terms of backbone CNNs, we observe a similar trend as in Plant identification, where InceptionV3 performs better than other CNNs in most of the approaches and datasets. The exception can only be seen in the multi-model approach where MobileNet got the best results in Plant Village (99.275% accuracy and 0.99274 F1-score) and VGG16 achieved 40.291% accuracy and 0.33305 F1-score in PlantDoc-0.2. In terms of common approaches (multi-model, multi-label, multi-output, multi-task), multi-model (with InceptionV3) achieves the best accuracy (51.992%) and F1-score (0.48344) on PlantDoc-1.0; multi-output (with InceptionV3) is better than the others on Plant Village (99.414% accuracy and 0.99413 F1-score) and on PlantDoc-0.2 (43.165% accuracy and 0.41892 F1-score); multi-label (with InceptionV3) achieves the highest accuracy (97.547%) and F1-score (0.97468) on Plant Leaves. Our proposed model also performs well in the case of disease classification. Without the balance

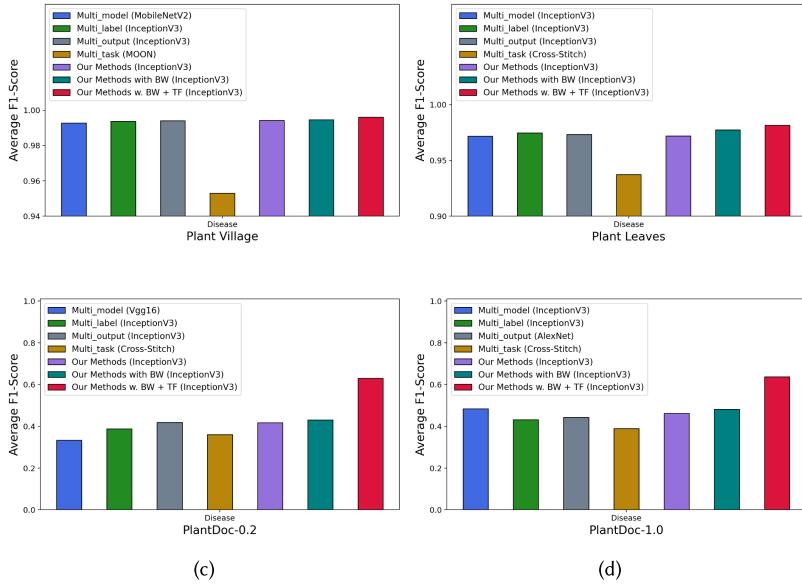


Fig. 7. comparison of the approaches on disease classification.

weights, GSMo-CNN achieves better results than multi-model, multi-label, multi-output, and multi-task approaches in PlantVillage and PlantDoc-0.2. The accuracy and F1-score of our new model without BW and with InceptionV3 as the backbone are as follows: Plant Village (99.418% and 0.99417), Plant Leaves (97.292% and 0.97201), PlantDoc-0.2 (45.029%, 0.41716) and PlantDoc-1.0 (47.542% and 0.46156). When BW are used, with InceptionV3 as the best backbone GSMo-CNNs achieves the best results in 3 out of 4 cases. In particular, it has 99.466% accuracy and 0.994466 F1-score on Plant Village dataset; 97.759% accuracy and 0.97738 F1-score on Plant Leaves; and 45.301% accuracy and 0.43095 F1-score on PlantDoc-0.2. Only in PlantDoc-1.0 where GSMo-CNN ranks second, here, multi-model wins the best accuracy (51.992%) and F1-score (0.48344). If we use InceptionV3 pre-trained weights from ImageNet as the backbone for GSMo-CNN, we can even achieve much higher performance. As we can see, the accuracy and F1-score of our model in all datasets are improved significantly as follows, Plant Village (99.615% accuracy and 0.99615 F1-score), Plant Leaves (98.149% accuracy and 0.98146 F1-score), PlantDoc-0.2 (64.544% accuracy and 0.62968 F1-score) and PlantDoc-1.0 (63.305% accuracy and 0.63757 F1-score).

### 4.3 Plant Identification and Disease Classification

For completeness, we will evaluate the models and approaches in predicting plant species and disease types altogether. This is the combination of the plant identification task and the disease classification task, as discussed earlier. However, instead of evaluating each task separately, we are interested in studying how DL models can perform accurate predictions for both tasks. A prediction is accurate if and only if both plant species and disease type are inferred correctly. This would be the key criteria for users to select a model for their applications, as usually we want to have information about both plants and diseases to find a suitable treatment. For evaluation, plant species and disease type will be combined to the joint species-disease labels for the calculation of average accuracy and F1-score.

Table 7 and Figure 8 shows the results of all models and approaches. Although the combined prediction of plant species and disease types is more complex than the previous two tasks, all

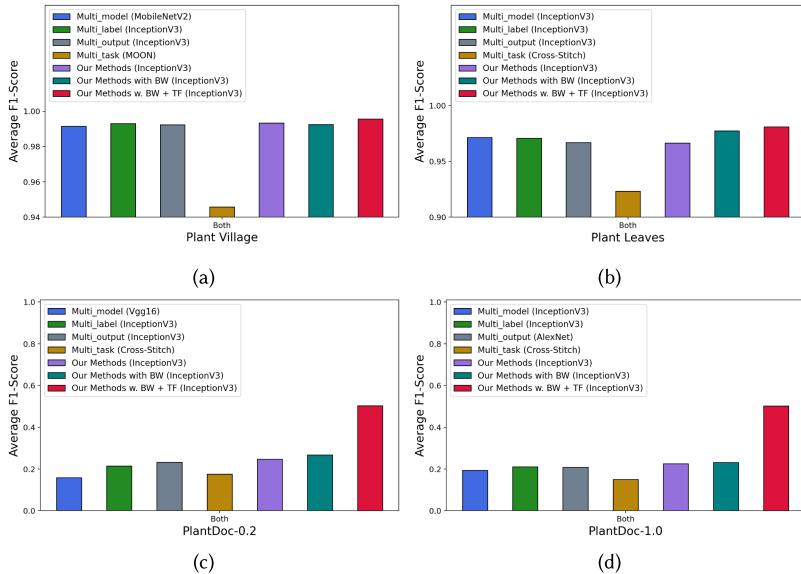


Fig. 8. comparison of the tpproaches on plant identification and disease classification.

approaches achieved promising results with more than 85% average accuracy and 0.85 F1-score on Plant Village and Plant Leaves. In terms of backbone CNNs, the multi-model approach demonstrates that MobileNet achieves the best performance (99.05% accuracy and 0.99148 F1-score) on Plant Village and VGG16 achieves the highest results on PlantDoc-0.2 (16.252% accuracy and 0.15821 F1-score). The multi-output approach shows that AlexNet has the highest accuracy of 22.458% and F1-score of 0.20815 on PlantDoc-1.0. Except for such three cases, InceptionV3 achieves higher performance than other backbone CNNs in the other cases.

Among the common multi-prediction approaches (multi-model, multi-label, multi-output, and multi-task), multi-label (power-set) performs the best on Plant Village, Plant Leaves, and PlantDoc-1.0 while multi-output has the best results on PlantDoc-0.2. It makes sense as the power-set labelling in the multi-task approach combines the two labels together, therefore, the learning would directly optimise the prediction of the combined label (plant and disease). Interestingly, multi-output performs well in PlantDoc-0.2 even though there is no communication between the two labels during the learning. Having said that, it can be the back-propagation of gradients from the two branches to the shared CNNs that plays a role in regularisation for the shared features in the learning. Such regularisation would help the model to learn more generalised features rather than task-specific features. The multi-task learning models (TSNs, MOON, Cross Stitch, and MTAN) show good results in Plant Village and Plant Leaves, however, they are not comparable to other approaches.

Our proposed model GSMo-CNN achieves the best performance on all datasets, except accuracy on PlantDoc-1.0 where multi-label with InceptionV3 backbone tops the list with 25.466%. In the case where BW are not applied ( $\beta_1 : \delta_1 : \beta_2 : \delta_2 = 1 : 1 : 1 : 1$ ), GSMo-CNN is better than multi-model, multi-label, multi-output, and multi-task. Its best performance is: 99.315% accuracy and 0.99333 F1-score on Plant Village; 96.593% accuracy and 0.96641 F1-score on Plant Leaves; 25.359% accuracy and 0.24692 F1-score on PlantDoc-0.2; and 23.814% accuracy and 0.22487 F1-score on PlantDoc-1.0. Again, InceptionV3 is the best backbone CNN for GSMo-CNN. When BW are employed, the improvement can be seen on Plant Leaves, PlantDoc-0.2, and PlantDoc-1.0. As

mentioned earlier, we apply a grid search and narrow down the grid to find a combination of the weights that perform well on all datasets. The reported results in Tables 5, 6, and 7 are from the BW  $\beta_1 : \delta_1 : \beta_2 : \delta_2 = 0.1 : 0.1 : 0.4 : 0.5$ . In particular, GSMo-CNN with BW achieves 99.208% accuracy and 0.99245 F1-score on Plant Village; 97.524% accuracy and 0.97746 F1-score on Plant Leaves; 26.175% accuracy and 0.26760 F1-score on PlantDoc-0.2; and 24.153% accuracy and 0.23191 F1-score on PlantDoc-1.0.

Similar to the independent prediction of plant and disease as we shown in 4.1 and 4.2, when we initialise InceptionV3 from the weights pre-trained on ImageNet, GSMo-CNN can improve the performance in all datasets, producing the highest accuracy and F1-scores in this study. The accuracy and F1-scores are as follows, Plant Village: 99.558% and 0.99565; Plant Leaves: 98.042% and 0.98116; PlantDoc-0.2: 50.291% and 0.50396; and PlantDoc-1.0: 50.000% and 0.50191.

It is worth noting that our proposed method does not have the problem of computational overhead because (1) we do not need to combine the labels during the learning; and (2) stacking the labels is efficient and its structure only needs a single CNN as the backbone. The only complexity users might find from our approach is the search for BW. Fortunately, as we showed in our intensive experiment that it can be optional as GSMo-CNN without BW can already achieve better performance than other approaches. Together with the transfer learning results the BW even bolster the effectiveness of GSMo-CNN. This shows the flexibility of our approach where users are provided with different options, including the choice of backbone CNNs, the choice of the BW for the losses, and/or the choice of pre-training weights. All these options have proved to be effective in improving the performance of plant identification and disease classification. We have shown in the experiment that the idea of the proposed DL structure, where two labels (Plant and Disease) can help each other to eliminate some wrong options to improve the prediction accuracy. In the next section, let us summarise the findings and provide further analysis of the results.

#### 4.4 Accuracy, F1-Score, and False Positive Rates of the Optimums

Table 8 offers a comprehensive performance evaluation of various approaches' optimal models across diverse datasets. All models leverage Inception V3 backbones, recognised as the top-performing backbone in Table 5, 6, and 7. Within the table, we present accuracy (Acc), F1-score (F1), and FPR for each model. In the context of accuracy and F1-score, higher values are indicative of better performance, while for FPR, closer values to 0 will be better performance. Notably, our methods, specifically the classifier chain with optimised BW and the transfer learning approach (the last row), demonstrate their superiority in this table. We can see that it achieves accuracy and F1-scores close to 1.0 in both the Plant Village and Plant Leaves datasets. Notably, compared to the results in the existing literature [105], this model surpasses other methods combined with transfer learning on the Plant Village dataset. It also performs exceptionally well in the two datasets of PlantDoc, achieving over 0.7 for plant species classification (Plant) and over 0.6 for leaf disease classification (Dis). And the overall classification (Both) exceeds 0.5 in both Accuracy and F1-score. In terms of FPR, each column's values are significantly close to 0 compared to all other rows, highlighting the exceptional robustness of this model.

In summary, this supremacy underscores the effectiveness of incorporating the classifier chain structure, optimising weights through grid-search, utilising transfer learning from ImageNet weights, and adopting Inception V3 backbones. These techniques collectively result in a more robust and precise classification system. Furthermore, these methods consistently maintain relatively low FPR, underscoring their capability to minimise misclassifications. As we focus on the rows related to the classifier chain in the table (the last three rows), the effectiveness of various methods becomes evident. Their accuracy and F1-scores exhibit a progressive increase, while the FPR consistently decreases. Ultimately, this amalgamation leads to a heightened level of model

Table 5. Plant Type Results

		Plant Village			Plant leaves			PlantDoc-0.2			PlantDoc-1.0		
		F1	Acc	F1	F1	Acc	F1	F1	Acc	F1	F1	Acc	F1
Multi-model (Plant)	CNN	97.448% ± 0.00000%	97.441% ± 0.00152	94.451% ± 0.00000%	0.94416 ± 0.00872	35.515% ± 0.00000%	0.984%	37.179% ± 0.00757	37.712% ± 0.00000%	0.33216% ± 0.00000%	37.712% ± 0.00000%	0.33216% ± 0.00000%	37.712% ± 0.00000%
	AlexNet	98.655% ± 0.00000%	98.654% ± 0.00322	96.885% ± 0.00000%	0.96089 ± 0.00852	40.19% ± 0.00000%	0.968%	36.353% ± 0.00000%	36.398% ± 0.00000%	0.34042% ± 0.00000%	36.398% ± 0.00000%	0.34042% ± 0.00000%	36.398% ± 0.00000%
	VGG16	99.465% ± 0.00000%	99.465% ± 0.00320	96.792% ± 0.00000%	0.96789 ± 0.00758	45.553% ± 0.00000%	0.971%	37.287% ± 0.01285	42.119% ± 0.02285	0.36066% ± 0.00000%	42.119% ± 0.02285	0.36066% ± 0.00000%	42.119% ± 0.02285
	ResNet101	99.327% ± 0.00000%	99.328% ± 0.00005%	98.467% ± 0.00000%	0.98465 ± 0.00000%	36.369% ± 0.00000%	0.934%	0.18736 ± 0.01701	29.958% ± 0.01444%	0.21007% ± 0.02095	29.958% ± 0.01444%	0.21007% ± 0.02095	29.958% ± 0.01444%
	EfficientNet	99.389% ± 0.00000%	99.388% ± 0.00075	97.417% ± 0.00000%	0.94707 ± 0.00674	34.369% ± 0.00000%	0.94%	0.28476 ± 0.00829	31.780% ± 0.01705%	0.23555% ± 0.01912	31.780% ± 0.01705%	0.23555% ± 0.01912	31.780% ± 0.01705%
	InceptionV3	99.389% ± 0.00000%	99.388% ± 0.00075	99.175% ± 0.00000%	0.99175 ± 0.00000	45.068% ± 0.00000%	0.945%	0.36668 ± 0.02233	40.297% ± 0.0301%	0.37522% ± 0.03833	40.297% ± 0.0301%	0.37522% ± 0.03833	40.297% ± 0.0301%
	MobileNetV2	99.750% ± 0.00000%	99.750% ± 0.00000%	96.260% ± 0.00000%	0.96065 ± 0.00023	33.592% ± 0.00000%	0.905%	0.20253 ± 0.00000%	29.593% ± 0.00728	0.11020% ± 0.00417	29.593% ± 0.00728	0.11020% ± 0.00417	29.593% ± 0.00728
	ViT	92.045% ± 0.00000%	92.045% ± 0.00000%	90.875% ± 0.00000%	0.90875 ± 0.00080	94.617% ± 0.00000%	0.94635 ± 0.00153	30.874% ± 0.00000%	0.7973%	0.15459 ± 0.01404	29.661% ± 0.00000%	0.14268% ± 0.00854	29.661% ± 0.00000%
	CNN	98.645% ± 0.00000%	98.643% ± 0.00159	94.384% ± 0.00000%	0.94040 ± 0.01318	38.250% ± 0.00000%	0.984%	0.35807 ± 0.00239	34.746% ± 0.00000%	0.32100% ± 0.00000%	34.746% ± 0.00000%	0.32100% ± 0.00000%	34.746% ± 0.00000%
	AlexNet	98.578% ± 0.00000%	98.581% ± 0.00089	96.404% ± 0.00000%	0.96405 ± 0.00501	33.981% ± 0.00000%	0.944%	0.35207 ± 0.02660	32.500% ± 0.00000%	0.31231% ± 0.00649	32.500% ± 0.00000%	0.31231% ± 0.00649	32.500% ± 0.00000%
Multi-label (Plant)	VGG16	97.114% ± 0.00000%	97.110% ± 0.00187	92.175% ± 0.00055%	0.92139 ± 0.00031	30.252% ± 0.01734%	0.78%	0.23354 ± 0.01460	23.771% ± 0.301%	0.22337% ± 0.02978	23.771% ± 0.301%	0.22337% ± 0.02978	23.771% ± 0.301%
	ResNet101	99.701% ± 0.00000%	99.704% ± 0.00056	98.257% ± 0.00051%	0.98261 ± 0.00048	27.340% ± 0.02315%	0.75%	0.24259 ± 0.02912	23.390% ± 0.187%	0.21037% ± 0.01491	23.390% ± 0.187%	0.21037% ± 0.01491	23.390% ± 0.187%
	EfficientNet	99.314% ± 0.00000%	99.315% ± 0.00089	95.538% ± 0.00089	0.95510 ± 0.00092	23.903% ± 0.01097%	0.90%	0.23702 ± 0.01132	21.102% ± 0.0559%	0.17609% ± 0.0528	21.102% ± 0.0559%	0.17609% ± 0.0528	21.102% ± 0.0559%
	InceptionV3	99.810% ± 0.00000%	99.810% ± 0.00118	99.112% ± 0.00000%	0.99111 ± 0.00000	41.981% ± 0.00088%	0.608%	0.40630 ± 0.01026	38.771% ± 0.00000%	0.36693% ± 0.00783	38.771% ± 0.00000%	0.36693% ± 0.00783	38.771% ± 0.00000%
	MobileNetV2	99.469% ± 0.00000%	99.469% ± 0.00019	95.450% ± 0.00000%	0.95464 ± 0.00078	13.592% ± 0.00000%	0.000%	0.02323 ± 0.00000%	0.02187% ± 0.00000%	0.02187% ± 0.00000%	0.02187% ± 0.00000%	0.02187% ± 0.00000%	0.02187% ± 0.00000%
	ViT	92.328% ± 0.00000%	92.328% ± 0.00168	95.566% ± 0.00000%	0.95563 ± 0.00350	22.718% ± 0.00000%	0.583%	0.20821 ± 0.00099	17.797% ± 0.01495%	0.16317% ± 0.02967	17.797% ± 0.01495%	0.16317% ± 0.02967	17.797% ± 0.01495%
	CNN	97.943% ± 0.00000%	97.938% ± 0.00150	92.965% ± 0.00000%	0.92930 ± 0.01038	39.398% ± 0.00000%	0.940%	0.37328 ± 0.02085	35.805% ± 0.00000%	0.33279% ± 0.00302	35.805% ± 0.00000%	0.33279% ± 0.00302	35.805% ± 0.00000%
	AlexNet	99.356% ± 0.00000%	99.357% ± 0.00056	95.871% ± 0.00000%	0.95862 ± 0.00473	39.456% ± 0.00000%	0.94%	0.37112 ± 0.01990	40.254% ± 0.00000%	0.36676% ± 0.00000%	40.254% ± 0.00000%	0.36676% ± 0.00000%	40.254% ± 0.00000%
	VGG16	99.407% ± 0.00000%	99.406% ± 0.00034	96.658% ± 0.00000%	0.96652 ± 0.00803	41.592% ± 0.00000%	0.91%	0.33072 ± 0.00733	42.585% ± 0.0178%	0.35115% ± 0.01855	42.585% ± 0.0178%	0.35115% ± 0.01855	42.585% ± 0.0178%
Multi-output (Plant)	ResNet101	99.683% ± 0.00000%	99.683% ± 0.00019	97.658% ± 0.00033%	0.97651 ± 0.00031	34.951% ± 0.00000%	0.90%	0.25029 ± 0.01327	31.695% ± 0.01525%	0.21581% ± 0.01123	31.695% ± 0.01525%	0.21581% ± 0.01123	31.695% ± 0.01525%
	EfficientNet	99.752% ± 0.00000%	99.752% ± 0.00016%	95.183% ± 0.00000%	0.95161 ± 0.01112	33.126% ± 0.00228%	0.51%	0.47071 ± 0.02533	43.095% ± 0.03027	0.38037% ± 0.04974	43.095% ± 0.03027	0.38037% ± 0.04974	43.095% ± 0.03027
	InceptionV3	99.334% ± 0.00000%	99.334% ± 0.00104	94.750% ± 0.00000%	0.94731 ± 0.01103	25.010% ± 0.07474%	0.40818%	0.10613 ± 0.04818	20.127% ± 0.09110%	0.07708% ± 0.05521	20.127% ± 0.09110%	0.07708% ± 0.05521	20.127% ± 0.09110%
	MobileNetV2	95.352% ± 0.00000%	95.352% ± 0.00026	90.035% ± 0.00026	0.89069 ± 0.00047	29.900% ± 0.00000%	0.000%	0.15770 ± 0.00000%	29.237% ± 0.00000%	0.13229% ± 0.00000%	0.13229% ± 0.00000%	0.13229% ± 0.00000%	0.13229% ± 0.00000%
	Cross-Split	98.197% ± 0.00000%	98.191% ± 0.00153	95.745% ± 0.00000%	0.95737 ± 0.00324	39.845% ± 0.00000%	0.720%	0.35556 ± 0.01282	38.814% ± 0.0174%	0.34577% ± 0.013165	38.814% ± 0.0174%	0.34577% ± 0.013165	38.814% ± 0.0174%
	MTAN	95.461% ± 0.00000%	95.464% ± 0.00395	90.377% ± 0.00000%	0.90345 ± 0.02651	27.825% ± 0.00000%	0.394%	0.19536 ± 0.02203	30.169% ± 0.0180%	0.18824% ± 0.01435	30.169% ± 0.0180%	0.18824% ± 0.01435	30.169% ± 0.0180%
	TsNs	91.704% ± 0.01452%	91.968% ± 0.01468	85.749% ± 0.00000%	0.85880 ± 0.03514	27.786% ± 0.02775%	0.18%	0.18716 ± 0.02020	32.881% ± 0.01898%	0.21219% ± 0.03381	32.881% ± 0.01898%	0.21219% ± 0.03381	32.881% ± 0.01898%
	MOON	97.589% ± 0.00000%	97.586% ± 0.00411	95.627% ± 0.01016	0.95646 ± 0.01016	36.350% ± 0.01437%	0.61%	0.25105 ± 0.01992	35.339% ± 0.01542%	0.25096% ± 0.01692	35.339% ± 0.01542%	0.25096% ± 0.01692	35.339% ± 0.01542%
Multi-task (Plant)	CNN	98.550% ± 0.00000%	98.558% ± 0.00000%	97.780% ± 0.00000%	0.97781 ± 0.00000%	42.252% ± 0.00000%	0.621%	0.36268 ± 0.010666	38.390% ± 0.0230%	0.35370% ± 0.02055	38.390% ± 0.0230%	0.35370% ± 0.02055	38.390% ± 0.0230%
	AlexNet	98.625% ± 0.00000%	98.625% ± 0.00085	95.716% ± 0.00000%	0.95688 ± 0.00816	37.942% ± 0.00000%	0.61%	0.34341 ± 0.01388	38.856% ± 0.0144%	0.35983% ± 0.03556	38.856% ± 0.0144%	0.35983% ± 0.03556	38.856% ± 0.0144%
	VGG16	99.511% ± 0.00000%	99.512% ± 0.00085	96.249% ± 0.00000%	0.96533% ± 0.00816	44.621% ± 0.00000%	0.66%	0.36913 ± 0.00923	43.898% ± 0.01495%	0.35444% ± 0.02802	43.898% ± 0.01495%	0.35444% ± 0.02802	43.898% ± 0.01495%
	ResNet101	99.714% ± 0.00000%	99.714% ± 0.00052	98.069% ± 0.00000%	0.98066 ± 0.00217	36.039% ± 0.00000%	0.150%	0.29830 ± 0.03925	36.610% ± 0.03536%	0.27054% ± 0.01914	36.610% ± 0.03536%	0.27054% ± 0.01914	36.610% ± 0.03536%
	EfficientNet	99.625% ± 0.00000%	99.625% ± 0.00038	95.816% ± 0.00000%	0.95798 ± 0.00596	31.631% ± 0.00000%	0.592%	0.23771 ± 0.03248	33.898% ± 0.0271%	0.24964% ± 0.01032	33.898% ± 0.0271%	0.24964% ± 0.01032	33.898% ± 0.0271%
	InceptionV3	99.850% ± 0.00000%	99.850% ± 0.00019	98.690% ± 0.00000%	0.98692 ± 0.00098	46.757% ± 0.00000%	0.97%	0.45049 ± 0.03376	44.492% ± 0.0196%	0.41757% ± 0.05976	44.492% ± 0.0196%	0.41757% ± 0.05976	44.492% ± 0.0196%
	MobileNetV2	95.595% ± 0.00000%	95.595% ± 0.00120	95.627% ± 0.00000%	0.95682 ± 0.00727	13.592% ± 0.00000%	0.761%	0.02353 ± 0.20127	0.00000% ± 0.110%	0.07708% ± 0.05521	0.00000% ± 0.110%	0.07708% ± 0.05521	0.00000% ± 0.110%
	ViT	90.282% ± 0.00000%	90.282% ± 0.00281	94.458% ± 0.00000%	0.94426 ± 0.00472%	32.136% ± 0.01650%	0.472%	0.23778 ± 0.02330	30.085% ± 0.01271%	0.24128% ± 0.00673	30.085% ± 0.01271%	0.24128% ± 0.00673	30.085% ± 0.01271%
	Our Methods with BW (Plant)	InceptionV3	99.687% ± 0.00000%	99.688% ± 0.00000%	99.646% ± 0.00000%	0.99647 ± 0.00000%	49.262% ± 0.02442%	0.48051 ± 0.01716	46.992% ± 0.03637%	0.44934% ± 0.03012	46.992% ± 0.03637%	0.44934% ± 0.03012	46.992% ± 0.03637%
	Our Methods w. BW + TF (Plant)	InceptionV3	99.928% ± 0.0012%	99.927% ± 0.00012	99.776% ± 0.00000%	0.99777 ± 0.000226	71.262% ± 0.39.62%	0.70602 ± 0.04275	73.644% ± 0.02548%	0.71980% ± 0.02900	73.644% ± 0.02548%	0.71980% ± 0.02900	73.644% ± 0.02548%

Table 6. Plant Disease Results

Plant Disease Results										PlantDoc-1.0			PlantDoc-0.2		
Plant Village				Plant leaves				Plant leaves				PlantDoc-1.0		PlantDoc-0.2	
		F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
Multi-label (Disease)	CNN	95.374% ± 0.323%	0.95364 ± 0.00312	89.343% ± 0.2442%	0.89046 ± 0.02561	33.146% ± 0.573%	0.30879 ± 0.01138	42.797% ± 0.000%	0.36857 ± 0.0000%	42.797% ± 0.000%	0.36857 ± 0.0000%	41.059% ± 0.01534	0.35199 ± 0.02495	41.059% ± 0.01534	0.35199 ± 0.02495
	AlexNet	97.399% ± 0.322%	0.97400 ± 0.00318	90.165% ± 0.543%	0.89938 ± 0.00537	33.864% ± 0.303%	0.30133 ± 0.01534	44.237% ± 0.01226%	0.35205 ± 0.02199	44.237% ± 0.01226%	0.35205 ± 0.02199	41.314% ± 0.01724	0.34279 ± 0.01651	41.314% ± 0.01724	0.34279 ± 0.01651
	VGG16	99.105% ± 0.715%	0.99104 ± 0.00716	93.045% ± 0.0042%	0.92978 ± 0.0042%	40.291% ± 0.1226%	0.24170 ± 0.01113	31.204% ± 0.139%	0.24170 ± 0.01113	31.204% ± 0.139%	0.24170 ± 0.01113	39.873% ± 0.0134%	0.29068 ± 0.00875	39.873% ± 0.0134%	0.29068 ± 0.00875
	ResNet101	98.828% ± 0.058%	0.98825 ± 0.00056	92.642% ± 0.0094%	0.92414 ± 0.00113	31.204% ± 0.139%	0.24170 ± 0.01113	31.320% ± 0.990%	0.25177 ± 0.00279	31.320% ± 0.990%	0.25177 ± 0.00279	39.873% ± 0.481%	0.31227 ± 0.00996	39.873% ± 0.481%	0.31227 ± 0.00996
	EfficientNet	98.361% ± 0.074%	0.98558 ± 0.00076	88.975% ± 0.921%	0.88555 ± 0.01084	31.204% ± 0.139%	0.24170 ± 0.01113	31.320% ± 0.990%	0.25177 ± 0.00279	31.320% ± 0.990%	0.25177 ± 0.00279	39.873% ± 0.481%	0.31227 ± 0.00996	39.873% ± 0.481%	0.31227 ± 0.00996
	InceptionV3	98.261% ± 0.074%	0.98558 ± 0.00076	87.170% ± 0.000%	0.97164 ± 0.00000	35.786% ± 0.2346%	0.30539 ± 0.02399	51.992% ± 0.0237%	0.48344 ± 0.05286	51.992% ± 0.0237%	0.48344 ± 0.05286	48.75% ± 0.00000	0.3124 ± 0.00000	48.75% ± 0.00000	0.3124 ± 0.00000
Multi-label (Disease)	MobileNetV2	99.275% ± 0.075%	0.99274 ± 0.00000	87.807% ± 0.1686%	0.89693 ± 0.01975	30.029% ± 0.078%	0.0923 ± 0.00020	38.701% ± 0.220%	0.18810 ± 0.03184	38.701% ± 0.220%	0.18810 ± 0.03184	39.237% ± 0.000%	0.25129 ± 0.03408	39.237% ± 0.000%	0.25129 ± 0.03408
	ViT	90.314% ± 0.0131%	0.89702 ± 0.01010	89.783% ± 0.0088%	0.89693 ± 0.00833	30.680% ± 0.220%	0.18810 ± 0.03184	38.701% ± 0.000%	0.1324 ± 0.00000	38.701% ± 0.000%	0.1324 ± 0.00000	39.237% ± 0.000%	0.25129 ± 0.03408	39.237% ± 0.000%	0.25129 ± 0.03408
	CNN	96.748% ± 0.023%	0.96752 ± 0.00025	91.632% ± 0.9977%	0.91562 ± 0.01059	35.184% ± 0.534%	0.33953 ± 0.01026	46.610% ± 0.000%	0.42350 ± 0.00000	46.610% ± 0.000%	0.42350 ± 0.00000	41.864% ± 0.01745	0.38276 ± 0.05842	41.864% ± 0.01745	0.38276 ± 0.05842
	AlexNet	96.438% ± 0.115%	0.96441 ± 0.00121	92.431% ± 0.381%	0.92444 ± 0.00312	32.000% ± 0.120%	0.30747 ± 0.01745	39.484% ± 0.01745	0.38276 ± 0.05842	39.484% ± 0.01745	0.38276 ± 0.05842	39.484% ± 0.01745	0.38276 ± 0.05842	39.484% ± 0.01745	0.38276 ± 0.05842
	VGG16	94.680% ± 0.444%	0.94659 ± 0.00436	88.380% ± 0.736%	0.88040 ± 0.00741	32.757% ± 0.815%	0.29959 ± 0.00604	39.958% ± 0.137%	0.35054 ± 0.01112	39.958% ± 0.137%	0.35054 ± 0.01112	39.958% ± 0.137%	0.35054 ± 0.01112	39.958% ± 0.137%	0.35054 ± 0.01112
	ResNet101	98.326% ± 0.0833%	0.98927 ± 0.00083	95.572% ± 0.402%	0.95467 ± 0.00394	30.158% ± 0.344%	0.28404 ± 0.02025	38.220% ± 0.344%	0.34683 ± 0.01162	38.220% ± 0.344%	0.34683 ± 0.01162	38.220% ± 0.344%	0.34683 ± 0.01162	38.220% ± 0.344%	0.34683 ± 0.01162
Multi-task (Disease)	EfficientNet	98.087% ± 0.183%	0.98083 ± 0.00185	91.188% ± 0.089%	0.90920 ± 0.00027	26.777% ± 0.180%	0.24184 ± 0.01083	32.203% ± 0.932%	0.30440 ± 0.04113	32.203% ± 0.932%	0.30440 ± 0.04113	32.203% ± 0.932%	0.30440 ± 0.04113	32.203% ± 0.932%	0.30440 ± 0.04113
	InceptionV3	99.383% ± 0.099%	0.99384 ± 0.00099	97.547% ± 0.033%	0.97468 ± 0.00029	39.753% ± 0.218%	0.33842 ± 0.00837	46.483% ± 0.00194	0.43180 ± 0.00194	46.483% ± 0.00194	0.43180 ± 0.00194	46.483% ± 0.00194	0.43180 ± 0.00194	46.483% ± 0.00194	0.43180 ± 0.00194
	MobileNetV2	98.571% ± 0.064%	0.98569 ± 0.00064	91.783% ± 0.372%	0.91625 ± 0.01533	71.446% ± 0.078%	0.04953 ± 0.00020	47.624% ± 0.00250	0.05075 ± 0.000250	47.624% ± 0.00250	0.05075 ± 0.000250	47.624% ± 0.00250	0.05075 ± 0.000250	47.624% ± 0.00250	0.05075 ± 0.000250
	ViT	90.052% ± 0.068%	0.89733 ± 0.00664	92.675% ± 0.0222%	0.92528 ± 0.00238	26.796% ± 0.000%	0.24090 ± 0.00985	38.983% ± 0.000%	0.32884 ± 0.01774	38.983% ± 0.000%	0.32884 ± 0.01774	38.983% ± 0.000%	0.32884 ± 0.01774	38.983% ± 0.000%	0.32884 ± 0.01774
	CNN	94.898% ± 0.493%	0.94882 ± 0.00487	89.467% ± 0.948%	0.89130 ± 0.01004	39.592% ± 0.452%	0.33808 ± 0.01346	41.314% ± 0.000%	0.39240 ± 0.00270	41.314% ± 0.000%	0.39240 ± 0.00270	41.314% ± 0.000%	0.39240 ± 0.00270	41.314% ± 0.000%	0.39240 ± 0.00270
	AlexNet	98.558% ± 0.127%	0.98056 ± 0.00127	92.508% ± 0.114%	0.92443 ± 0.00163	36.621% ± 0.807%	0.33630 ± 0.02018	47.034% ± 0.000%	0.44287 ± 0.00000	47.034% ± 0.000%	0.44287 ± 0.00000	47.034% ± 0.000%	0.44287 ± 0.00000	47.034% ± 0.000%	0.44287 ± 0.00000
Multi-output (Disease)	VGG16	98.645% ± 0.0422%	0.98642 ± 0.00044	94.833% ± 0.049%	0.94706 ± 0.01153	42.252% ± 0.816%	0.33853 ± 0.01492	48.517% ± 0.1	0.39686 ± 0.03644	48.517% ± 0.1	0.39686 ± 0.03644	48.517% ± 0.1	0.39686 ± 0.03644	48.517% ± 0.1	0.39686 ± 0.03644
	ResNet101	99.021% ± 0.0049%	0.99020 ± 0.00049	95.627% ± 0.466%	0.95588 ± 0.00249	35.730% ± 0.000%	0.24951 ± 0.00000	41.949% ± 0.000%	0.41892 ± 0.00977	41.949% ± 0.000%	0.41892 ± 0.00977	41.949% ± 0.000%	0.41892 ± 0.00977	41.949% ± 0.000%	0.41892 ± 0.00977
	EfficientNet	98.339% ± 0.103%	0.98539 ± 0.00103	91.833% ± 0.184%	0.91649 ± 0.01100	30.583% ± 0.558%	0.24967 ± 0.00594	38.644% ± 0.01	0.40202 ± 0.03922	38.644% ± 0.01	0.40202 ± 0.03922	38.644% ± 0.01	0.40202 ± 0.03922	38.644% ± 0.01	0.40202 ± 0.03922
	InceptionV3	99.414% ± 0.056%	0.99413 ± 0.00056	97.414% ± 0.494%	0.97330 ± 0.00540	43.165% ± 0.089%	0.41892 ± 0.00977	46.441% ± 0.03	0.40202 ± 0.03922	46.441% ± 0.03	0.40202 ± 0.03922	46.441% ± 0.03	0.40202 ± 0.03922	46.441% ± 0.03	0.40202 ± 0.03922
	MobileNetV2	98.583% ± 0.207%	0.98587 ± 0.00207	90.952% ± 0.740%	0.91873 ± 0.00712	36.733% ± 0.1946%	0.24552 ± 0.00000	34.575% ± 0.000%	0.15054 ± 0.00000	34.575% ± 0.000%	0.15054 ± 0.00000	34.575% ± 0.000%	0.15054 ± 0.00000	34.575% ± 0.000%	0.15054 ± 0.00000
	ViT	90.553% ± 0.098%	0.90458 ± 0.000712	86.733% ± 0.01946%	0.863373 ± 0.02181	31.456% ± 0.000%	0.15054 ± 0.00000	30.393% ± 0.000%	0.15054 ± 0.00000	30.393% ± 0.00000	0.15054 ± 0.00000	30.393% ± 0.00000	0.15054 ± 0.00000	30.393% ± 0.00000	0.15054 ± 0.00000
Multi-task (Disease)	Cross-Stitch	95.5241% ± 0.169%	0.95237 ± 0.00179	93.783% ± 0.248%	0.93739 ± 0.00276	40.466% ± 0.995%	0.33972 ± 0.00859	44.492% ± 0.01102	0.38920 ± 0.01102	44.492% ± 0.01102	0.38920 ± 0.01102	44.492% ± 0.01102	0.38920 ± 0.01102	44.492% ± 0.01102	0.38920 ± 0.01102
	MTAN	91.410% ± 0.710%	0.91411 ± 0.00702	87.148% ± 0.273%	0.86893 ± 0.02296	30.971% ± 0.175%	0.26367 ± 0.01803	34.229% ± 0.0175%	0.30392 ± 0.02078	34.229% ± 0.0175%	0.30392 ± 0.02078	34.229% ± 0.0175%	0.30392 ± 0.02078	34.229% ± 0.0175%	0.30392 ± 0.02078
	T-NSs	79.520% ± 0.022%	0.79681%	87.0990% ± 0.02653	0.81958% ± 0.01060%	0.92804 ± 0.011191	0.24798 ± 0.02701	34.575% ± 0.01745	0.31160 ± 0.02727	34.575% ± 0.01745	0.31160 ± 0.02727	34.575% ± 0.01745	0.31160 ± 0.02727	34.575% ± 0.01745	0.31160 ± 0.02727
	MOON	95.307% ± 0.0889%	0.95296 ± 0.00886	92.989% ± 0.1	0.90458 ± 0.00060	42.058% ± 0.992%	0.38199 ± 0.01462	46.229% ± 0.01399	0.41789 ± 0.01399	46.229% ± 0.01399	0.41789 ± 0.01399	46.229% ± 0.01399	0.41789 ± 0.01399	46.229% ± 0.01399	0.41789 ± 0.01399
	CNN	96.363% ± 0.000%	0.96352 ± 0.00000	95.005% ± 0.000%	0.92376 ± 0.00149	32.923% ± 0.9928%	0.302387 ± 0.00563	37.631% ± 0.00563	0.30592 ± 0.03019	37.631% ± 0.00563	0.30592 ± 0.03019	37.631% ± 0.00563	0.30592 ± 0.03019	37.631% ± 0.00563	0.30592 ± 0.03019
	AlexNet	97.380% ± 0.496%	0.97376 ± 0.00149	95.061% ± 0.259%	0.94922 ± 0.02299	42.718% ± 0.951%	0.34025 ± 0.01214	45.636% ± 0.01234	0.37864 ± 0.01234	45.636% ± 0.01234	0.37864 ± 0.01234	45.636% ± 0.01234	0.37864 ± 0.01234	45.636% ± 0.01234	0.37864 ± 0.01234
Our Methods v/o BW	VGG16	98.716% ± 0.139%	0.98713 ± 0.01142	95.061% ± 0.259%	0.94922 ± 0.02299	42.718% ± 0.951%	0.34025 ± 0.01214	45.636% ± 0.01234	0.37864 ± 0.01234	45.636% ± 0.01234	0.37864 ± 0.01234	45.636% ± 0.01234	0.37864 ± 0.01234	45.636% ± 0.01234	0.37864 ± 0.01234
	ResNet101	98.943% ± 0.056%	0.98943 ± 0.00056	95.461% ± 0.674%	0.95442 ± 0.00664	32.893% ± 0.104%	0.23341 ± 0.00812	38.729% ± 0.02078	0.39379 ± 0.02078	38.729% ± 0.02078	0.39379 ± 0.02078	38.729% ± 0.02078	0.39379 ± 0.02078	38.729% ± 0.02078	0.39379 ± 0.02078
	EfficientNet	98.639% ± 0.034%	0.98642 ± 0.00035	92.786% ± 0.880%	0.92687 ± 0.00850	30.194% ± 0.212%	0.24793 ± 0.02109	40.466% ± 0.0212%	0.33290 ± 0.0212%	40.466% ± 0.0212%	0.33290 ± 0.0212%	40.466% ± 0.0212%	0.33290 ± 0.0212%	40.466% ± 0.0212%	0.33290 ± 0.0212%
	InceptionV3	99.413% ± 0.029%	0.99417 ± 0.00030	97.292% ± 0.546%	0.97201 ± 0.00608	45.029% ± 0.134%	0.41716 ± 0.03430	47.542% ± 0.0644%	0.46156 ± 0.03985	47.542% ± 0.0644%	0.46156 ± 0.03985	47.542% ± 0.0644%	0.46156 ± 0.03985	47.542% ± 0.0644%	0.46156 ± 0.03985
	MobileNetV2	98.645% ± 0.335%	0.98648 ± 0.00329	91.726% ± 0.414%	0.91710 ± 0.01342	37.010% ± 0.058%	0.09918 ± 0.00015	36.356% ± 0.211%	0.0787 ± 0.00294	36.356% ± 0.211%	0.0787 ± 0.00294	36.356% ± 0.211%	0.0787 ± 0.00294	36.356% ± 0.211%	0.0787 ± 0.00294
	ViT	80.728% ± 0.087%	0.86416 ± 0.00183	90.330% ± 0.354%	0.86416 ± 0.00410	32.913% ± 0.291%	0.24160 ± 0.01754	38.983% ± 0.484%	0.30192 ± 0.00490	38.983% ± 0.484%	0.30192 ± 0.00490	38.983% ± 0.484%	0.30192 ± 0.00490	38.983% ± 0.484%	0.30192 ± 0.00490
Our Methods with BW (Disease)	InceptionV3	99.466% ± 0.000%	0.99466 ± 0.00000	97.759% ± 0.00000%	0.97738 ± 0.00000	45.301% ± 0.939%	0.43104 ± 0.03197	50.212% ± 0.584%	0.48154 ± 0.03108	50.212% ± 0.584%	0.48154 ± 0.03108	50.212% ± 0.584%	0.48154 ± 0.03108	50.212% ± 0.584%	0.48154 ± 0.03108
Our Methods w. BW + TF (Disease)	InceptionV3	99.615% ± 0.049%	0.99615 ± 0.00049	98.149% ± 0.167%	0.98146 ± 0.0										

Table 7. Both (Type and Disease) Results

		Plant Village			Plant leaves			PlantDoc-0.2			PlantDoc-1.0		
		Acc	F1	Acc	Acc	F1	Acc	Acc	F1	Acc	Acc	F1	
Multi-model (Both)	CNN	93.273% ± 0.0.396%	0.94202 ± 0.00310	85.283% ± 03.019%	0.86703 ± 0.02708	11.456% ± 00.583%	0.12928 ± 0.00427	12.712% ± 00.000%	0.11760 ± 0.00000	14.746% ± 02.678%	0.12720 ± 0.02971	14.746% ± 01.678%	
	AlexNet	96.275% ± 00.582%	0.96831 ± 0.00447	88.007% ± 00.971%	0.88788 ± 0.08004	12.117% ± 01.836%	0.12140 ± 0.03146	10.5821 ± 0.02069	0.15821 ± 0.01468	0.15821 ± 0.02069	0.12205 ± 0.01727	0.12205 ± 0.0150%	
	VGG16	98.634% ± 00.978%	0.98852 ± 0.00827	90.775% ± 00.283%	0.92730 ± 0.00409	16.252% ± 01.566%	0.15218 ± 0.00847	0.09337 ± 0.00390	0.09337 ± 0.00390	0.09337 ± 0.00390	0.09337 ± 0.00390	0.09337 ± 0.00390	
	ResNet101	98.205% ± 00.065%	0.98534 ± 0.00052	91.698% ± 00.094%	0.92189 ± 0.00087	05.612% ± 00.566%	0.06758 ± 0.0038%	07.553% ± 00.058%	0.06758 ± 0.0038%	06.805% ± 00.61%	0.06758 ± 0.0038%	06.805% ± 00.61%	
	EfficientNet	98.034% ± 00.000%	0.98301 ± 0.00029	85.814% ± 00.522%	0.87085 ± 0.00535	07.553% ± 00.058%	0.06758 ± 0.0038%	06.675% ± 00.0492	07.839% ± 01.820%	0.06758 ± 0.0038%	0.06758 ± 0.0038%	0.06758 ± 0.0038%	
	InceptionV3	98.034% ± 00.000%	0.98301 ± 0.00029	96.4623% ± 00.000%	0.97142 ± 0.00000	14.194% ± 00.421%	0.13135 ± 0.0039%	20.212% ± 02.924%	0.13135 ± 0.0039%	07.839% ± 01.820%	0.13135 ± 0.0039%	07.839% ± 01.820%	
Multi-label (Both)	MobileNetV2	99.050% ± 00.000%	0.99148 ± 0.00000	85.307% ± 01.726%	0.97846 ± 0.01687	04.078% ± 01.553%	0.03631 ± 0.00361	04.078% ± 01.553%	0.03631 ± 0.00361	0.03631 ± 0.00361	0.03631 ± 0.00361	0.03631 ± 0.00361	
	ViT	84.481% ± 01.364%	0.85548 ± 0.01771	86.238% ± 00.999%	0.87549 ± 0.00578	03.883% ± 01.00578	04.802% ± 00.00369	04.802% ± 00.00369	04.802% ± 00.00369	04.802% ± 00.00369	04.802% ± 00.00369	04.802% ± 00.00369	
	CNN	96.537% ± 00.057%	0.96542 ± 0.00062	89.523% ± 01.655%	0.89448 ± 0.01616	20.680% ± 00.445%	0.18933 ± 0.00913	20.6763% ± 00.000%	20.6763% ± 00.000%	20.6763% ± 00.000%	20.6763% ± 00.000%	20.6763% ± 00.000%	
	AlexNet	96.150% ± 00.172%	0.96156 ± 0.00175	91.099% ± 00.632%	0.90945 ± 0.00583	18.000% ± 00.772%	0.15703 ± 0.01071	19.195% ± 04.383%	0.15703 ± 0.01071	19.195% ± 04.383%	0.15703 ± 0.01071	19.195% ± 04.383%	
	VGG16	94.350% ± 00.433%	0.94316 ± 0.00444	85.527% ± 00.743%	0.85292 ± 0.00795	16.854% ± 00.986%	0.14534 ± 0.00831	13.432% ± 01.748%	0.14534 ± 0.00831	13.432% ± 01.748%	0.14534 ± 0.00831	13.432% ± 01.748%	
	ResNet101	98.841% ± 00.070%	0.98841 ± 0.00071	94.685% ± 00.230%	0.94643 ± 0.00240	16.000% ± 00.956%	0.12609 ± 0.02183	11.864% ± 00.709%	0.12609 ± 0.02183	11.864% ± 00.709%	0.12609 ± 0.02183	11.864% ± 00.709%	
Multi-output (Both)	EfficientNet	99.7923% ± 00.204%	0.97908 ± 0.00207	89.789% ± 00.444%	0.89457 ± 0.00349	12.913% ± 00.806%	0.09489 ± 0.00559	08.983% ± 06.678%	08.983% ± 06.678%	08.983% ± 06.678%	08.983% ± 06.678%	08.983% ± 06.678%	
	InceptionV3	99.310% ± 00.171%	0.99311 ± 0.00173	97.114% ± 00.000%	0.97077 ± 0.00000	22.971% ± 01.129%	0.21467 ± 0.00871	25.466% ± 00.127%	0.21467 ± 0.00871	25.466% ± 00.127%	0.21467 ± 0.00871	25.466% ± 00.127%	
	MobileNetV2	98.483% ± 00.023%	0.98481 ± 0.00021	90.122% ± 01.469%	0.90013 ± 0.01574	06.408% ± 01.553%	05.000% ± 00.254%	05.000% ± 00.254%	05.000% ± 00.254%	05.000% ± 00.254%	05.000% ± 00.254%	05.000% ± 00.254%	
	ViT	87.699% ± 00.571%	0.87899 ± 0.00500	0.90899% ± 00.533%	0.90881 ± 0.00533%	13.592% ± 00.900%	0.10092% ± 0.00978	11.864% ± 00.0795%	11.864% ± 00.0795%	11.864% ± 00.0795%	11.864% ± 00.0795%	11.864% ± 00.0795%	
	CNN	93.208% ± 00.369%	0.93766 ± 0.00483	85.098% ± 01.141%	0.86082 ± 0.01036	14.078% ± 00.701%	0.15213 ± 0.00644	12.288% ± 01.271%	0.15213 ± 0.00644	12.288% ± 01.271%	0.15213 ± 0.00644	12.288% ± 01.271%	
	AlexNet	97.719% ± 00.131%	0.97855 ± 0.00121	90.777% ± 00.578%	0.90887 ± 0.00518	16.272% ± 01.319%	0.160622 ± 0.01750	22.458% ± 00.000%	0.160622 ± 0.01750	22.458% ± 00.000%	0.160622 ± 0.01750	22.458% ± 00.000%	
Multi-task (Both)	VGG16	98.258% ± 00.085%	0.98359 ± 0.00066	92.941% ± 01.441%	0.93384 ± 0.01352	17.767% ± 01.456%	0.15111 ± 0.01682	16.737% ± 02.331%	0.15111 ± 0.01682	16.737% ± 02.331%	0.15111 ± 0.01682	16.737% ± 02.331%	
	ResNet101	98.789% ± 00.071%	0.98843 ± 0.00053	94.295% ± 00.466%	0.94471 ± 0.01399	09.126% ± 00.800%	0.06699 ± 0.00093	06.737% ± 00.0093	06.737% ± 00.0093	06.737% ± 00.0093	06.737% ± 00.0093	06.737% ± 00.0093	
	EfficientNet	98.195% ± 00.162%	0.98295 ± 0.00135	89.73% ± 01.825%	0.89997 ± 0.01753	73.78% ± 01.387%	0.08637 ± 0.00493	06.737% ± 01.127%	06.737% ± 01.127%	06.737% ± 01.127%	06.737% ± 01.127%	06.737% ± 01.127%	
	InceptionV3	99.215% ± 00.017%	0.99239 ± 0.00018	96.492% ± 00.519%	0.96693 ± 0.00336	23.650% ± 00.356%	0.32328% ± 0.01445	18.136% ± 04.308%	0.32328% ± 0.01445	18.136% ± 04.308%	0.32328% ± 0.01445	18.136% ± 04.308%	
	MobileNetV2	98.309% ± 00.284%	0.98401 ± 0.00276	89.234% ± 01.277%	0.89726 ± 01.118	02.155% ± 03.292%	0.02289 ± 0.00441	02.119% ± 02.119%	0.02289 ± 0.00441	02.119% ± 02.119%	0.02289 ± 0.00441	02.119% ± 02.119%	
	ViT	87.276% ± 01.327%	0.87476 ± 01.048	82.845% ± 02.300%	0.87255 ± 0.02119	01.942% ± 00.000%	0.00741 ± 0.00000	03.390% ± 00.000%	03.390% ± 00.000%	03.390% ± 00.000%	03.390% ± 00.000%	03.390% ± 00.000%	
Multi-task (Both)	Cross-Stitch	93.870% ± 00.310%	0.94509 ± 0.00276	91.350% ± 00.106%	0.92334 ± 0.00001	17.068% ± 00.629%	0.175732 ± 0.005541	16.610% ± 02.241%	0.175732 ± 0.005541	16.610% ± 02.241%	0.175732 ± 0.005541	16.610% ± 02.241%	
	MTAN	89.509% ± 00.398%	0.90269 ± 0.00836	82.424% ± 03.549%	0.83209 ± 0.03344	04.175% ± 01.210%	0.02986 ± 0.01344	06.102% ± 00.988%	0.02986 ± 0.01344	06.102% ± 00.988%	0.02986 ± 0.01344	06.102% ± 00.988%	
	TsNs	76.808% ± 03.190%	0.77433 ± 0.03101	74.673% ± 05.778%	0.75069 ± 0.05800	05.029% ± 01.179%	0.02603 ± 0.01110	05.975% ± 01.545%	0.02603 ± 0.01110	05.975% ± 01.545%	0.02603 ± 0.01110	05.975% ± 01.545%	
	MOON	94.414% ± 00.997%	0.94586 ± 0.00974	90.744% ± 01.470%	0.90727 ± 0.01542	08.272% ± 01.571%	0.04720 ± 0.01639	09.068% ± 01.056%	0.04720 ± 0.01639	09.068% ± 01.056%	0.04720 ± 0.01639	09.068% ± 01.056%	
	CNN	95.500% ± 00.000%	0.95741 ± 0.00000	93.563% ± 00.000%	0.93847 ± 0.00000	21.748% ± 01.553%	0.152149 ± 0.01257	18.178% ± 02.524%	0.152149 ± 0.01257	18.178% ± 02.524%	0.152149 ± 0.01257	18.178% ± 02.524%	
	AlexNet	96.968% ± 00.588%	0.97061 ± 0.00561	90.765% ± 01.310%	0.90818 ± 0.01255	15.553% ± 02.203%	0.15499% ± 0.02119	17.246% ± 01.229%	0.15499% ± 0.02119	17.246% ± 01.229%	0.15499% ± 0.02119	17.246% ± 01.229%	
Our Methods w/o BW (Both)	VGG16	98.395% ± 00.198%	0.98445 ± 0.00189	93.219% ± 00.644%	0.93439 ± 0.00628	16.000% ± 01.332%	0.14580% ± 0.01274	17.119% ± 02.330%	0.14580% ± 0.01274	17.119% ± 02.330%	0.14580% ± 0.01274	17.119% ± 02.330%	
	ResNet101	98.773% ± 00.053%	0.98796 ± 0.00056	94.550% ± 00.737%	0.94664 ± 0.00664	09.8325% ± 01.492%	0.07677% ± 0.01649	10.254% ± 00.678%	0.07677% ± 0.01649	10.254% ± 00.678%	0.07677% ± 0.01649	10.254% ± 00.678%	
	EfficientNet	98.445% ± 00.052%	0.98482 ± 0.00060	90.877% ± 00.980%	0.90985 ± 0.00847	08.039% ± 01.318%	0.06106% ± 0.01903	0.07318% ± 0.00716	0.06106% ± 0.01903	0.07318% ± 0.00716	0.06106% ± 0.01903	0.07318% ± 0.00716	
	InceptionV3	99.315% ± 00.037%	0.99333 ± 0.00037	96.592% ± 00.447%	0.96641 ± 0.00457	25.559% ± 02.170%	0.24692% ± 0.02342	23.814% ± 02.910%	0.24692% ± 0.02342	23.814% ± 02.910%	0.24692% ± 0.02342	23.814% ± 02.910%	
	MobileNetV2	98.408% ± 00.404%	0.98445 ± 0.00392	89.989% ± 01.468%	0.90320 ± 0.01200	03.689% ± 01.165%	0.02868% ± 0.00226	04.661% ± 00.424%	0.02868% ± 0.00226	04.661% ± 00.424%	0.02868% ± 0.00226	04.661% ± 00.424%	
	ViT	83.0303% ± 00.064%	0.83316 ± 0.00037	87.913% ± 00.295%	0.87931 ± 0.00502	08.447% ± 01.262%	0.07872% ± 0.01946	06.780% ± 00.424%	06.780% ± 00.424%	06.780% ± 00.424%	06.780% ± 00.424%	06.780% ± 00.424%	
Our Methods with BW (Both)	InceptionV3	99.208% ± 00.000%	0.99245 ± 0.00000	97.524% ± 00.000%	0.97746 ± 00.000	26.175% ± 03.008%	0.26760 ± 0.03055	24.153% ± 02.584%	0.23191 ± 0.02879	24.153% ± 02.584%	0.23191 ± 0.02879	24.153% ± 02.584%	
	Our Methods w. BW + TF (Both)	InceptionV3	99.558% ± 00.055%	0.99565 ± 0.00064	98.042% ± 00.270%	0.98116 ± 0.00196	50.291% ± 03.895%	0.50396% ± 0.04018	50.000% ± 02.023%	0.50191% ± 0.02036	50.000% ± 02.023%	0.50191% ± 0.02036	

Table 8. The Optimums' Accuracy, F1-score and False Positive Rates

Approaches	Metric	Plant Village			Plant Leaves			Plant Doc 0.2			Plant Doc Original		
		Plant	Dis	Both	Plant	Dis	Both	Plant	Dis	Both	Plant	Dis	Both
Multi-model	Acc	0.99750	0.99275	0.99050	0.99175	0.97170	0.96462	0.44854	0.36311	0.14563	0.44492	0.50847	0.21186
	F1	0.99750	0.99274	0.99148	0.99175	0.97164	0.97142	0.36914	0.30423	0.13374	0.43324	0.47997	0.21345
	FPR	0.00033	0.00039	0.00022	0.00093	0.00373	0.00118	0.05168	0.04737	0.01723	0.04504	0.03634	0.01221
Multi-label	Acc	0.99834	0.99558	0.99484	0.99112	0.97558	0.97114	0.33786	0.37282	0.21748	0.41949	0.47034	0.24576
	F1	0.99834	0.99559	0.99485	0.99111	0.97477	0.97077	0.32529	0.33752	0.17350	0.41247	0.46042	0.21948
	FPR	0.00013	0.00023	0.00014	0.00080	0.00361	0.00137	0.05330	0.04435	0.02964	0.04725	0.03855	0.02834
Multi-output	Acc	0.99888	0.99388	0.99288	0.98890	0.96670	0.96004	0.52427	0.43107	0.23883	0.44492	0.51271	0.19915
	F1	0.99887	0.99388	0.99319	0.98893	0.96616	0.96127	0.48730	0.42531	0.24174	0.39350	0.45603	0.17277
	FPR	0.00017	0.00032	0.00020	0.00104	0.00437	0.00167	0.04249	0.03991	0.01745	0.04800	0.03861	0.02533
Our Methods	Acc	0.99838	0.99400	0.99300	0.99112	0.97336	0.96559	0.46214	0.44854	0.28155	0.51695	0.50424	0.25424
	F1	0.99837	0.99400	0.99319	0.99119	0.97201	0.96599	0.44974	0.41720	0.25822	0.47412	0.48480	0.24085
	FPR	0.00025	0.00032	0.00020	0.00081	0.00414	0.00149	0.04403	0.03973	0.01491	0.04185	0.03658	0.02038
Our Methods with BW	Acc	0.99725	0.99450	0.99225	0.99646	0.97759	0.97524	0.48932	0.50874	0.31456	0.53814	0.53390	0.31356
	F1	0.99726	0.99449	0.99248	0.99647	0.97738	0.97746	0.51162	0.47363	0.33094	0.51138	0.50913	0.28519
	FPR	0.00037	0.00030	0.00022	0.00039	0.00327	0.00107	0.04121	0.03774	0.01443	0.03818	0.03411	0.01660
Our Methods w. BW + TF	Acc	0.99913	0.99650	0.99600	1.00000	0.98231	0.98231	0.76505	0.65049	0.55340	0.76271	0.62712	0.51271
	F1	0.99912	0.99650	0.99625	1.00000	0.98237	0.98225	0.76336	0.63742	0.54967	0.75084	0.63908	0.50652
	FPR	0.00013	0.00019	0.00011	0.00000	0.00217	0.00094	0.02015	0.02508	0.01113	0.02011	0.02545	0.01304

effectiveness, that may render them an invaluable choice for practical applications in SA and plant health management.

#### 4.5 Summary and Ablation Study

**4.5.1 Backbone CNNs and ViT.** The very first question we would like to answer in this study is which CNN architectures are most useful for plant identification and disease classification. Although we cannot test all CNNs available in the literature, we selected the most popular ones in image classification and plant pathology. AlexNet, VGG16, ResNet101, EfficientNet, InceptionV3, MobileNetV2, ViT, and our custom CNN are implemented for the study. In summary, we plot four groups of histograms in Figure 9, where each column representing the average F1-score of a backbone CNN in all approaches of this study (multi-model, multi-label, multi-output, and GSMo-CNN). These related results are elaborated in Tables 5, 6, and 7. As highlighted in Section 3.3, the F1 score proves more appropriate for evaluating prediction results which have imbalanced data, as discussed earlier. This graph aims to compare the performance of all backbone CNNs statistically and provide a view of how different backbone CNNs perform on different datasets. Each plot in Figure 9 shows average performance in plant identification, disease classification; and the combination of plant and disease prediction (Both) over different approaches. As we can see, InceptionV3 has a better average F1-score than AlexNet, VGG16, ResNet101, EfficientNet, MobileNet, ViT, and our custom CNN in all datasets. In PlantVillage, although being better, the difference between InceptionV3 and the other backbone CNNs does not stand out, as all backbone CNNs achieve more than 0.95 F1-score. However, in Plant Leaves, PlantDoc-0.2, and PlantDoc-1.0, InceptionV3 clearly demonstrates its advantage over the other CNNs. The second-best backbone CNN in Plant Village and Plant Leaves is ResNet101. In PlantDoc-0.2 and PlantDoc-1.0, our custom CNN and AlexNet are better than other backbones, just behind InceptionV3. This is different from Plant Village and Plant Leaves where our custom CNN has the lowest performance. Besides, ViT showcased somewhat satisfactory outcomes in testing across all datasets, but it fell short of attaining optimal results. We can also notice that, the light-weight CNNs (MobileNetV2

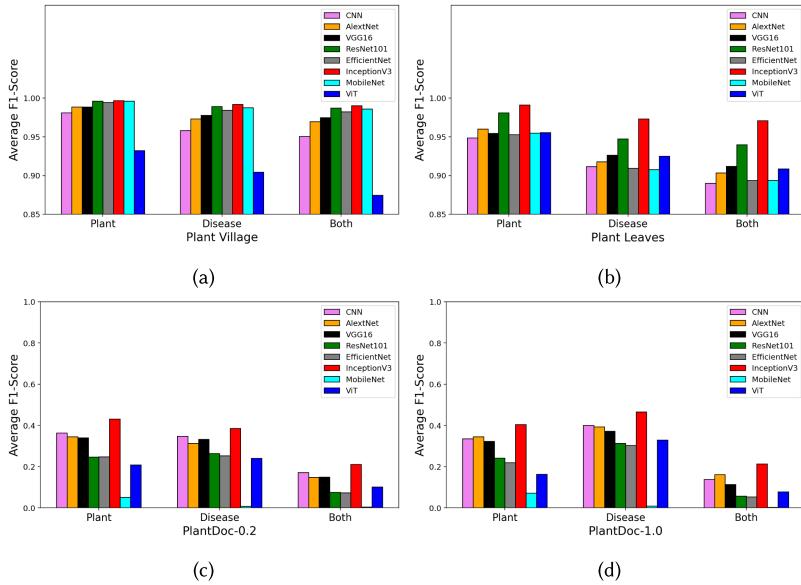


Fig. 9. comparison of backbone CNNs (Including ViT).

and EfficientNet) perform reasonably well on Plant Village and Plant Leaves and they have similar results. However, for PlantDoc-0.2 and PlantDoc-1.0 both light-weight CNNs have the lowest F1-score with MobileNetV2 trailing behind. It would suggest that light-weights CNNs should only be used for close-up images of a leaf, and not for images with multiple leaves and complex backgrounds.

The advantage of InceptionV3 can be justified by three factors. First, InceptionV3 can be deeper than VGG with a similar number of parameters. Second, InceptionV3 incorporates techniques like batch normalization and dropout to help mitigate overfitting, making it more robust on smaller datasets compared to VGG. Finally and more importantly, InceptionV3 uses a combination of convolutional layers with different kernel sizes and max-pooling layers to capture multi-scale features efficiently. This allows it to learn features representing disease types from small areas of leaves, thus, achieving competitive performance compared to other backbone CNNs.

**4.5.2 Single Models Versus Dual Models.** In the past, it was common and effective to use a model for a task, such as a plant species identification [40, 64, 81] or leaf disease classification [1, 5, 98]. Training an individual model for a particular task would be reasonable and straightforward as ones can optimise the model for that specific task. Multi-task learning has been shown useful, however, it is designed for tasks with different data where the combination of data can be useful for all task. This is not the case in this study where the data for all tasks are the same. That is the reason why multi-task approach does not show its advantage in our experiment. Despite that, we are interested in investigating whether using a single model for the two tasks will be better than using an ensemble of CNNs, each for a task. Also, it would be more efficient, in terms of both computation and memory storage, if a single model can effectively predict both plant species and disease types.

In Figure 10, we show the comparison of all approaches in this study, including multi-model, multi-label, multi-output, multi-task, and our methods (GSMo-CNNs with and without BW). Note that, the multi-model approach applies dual models, one for each task. For the sake of comparison, each column bar in the graph represents the highest F1-score of an approach in plant

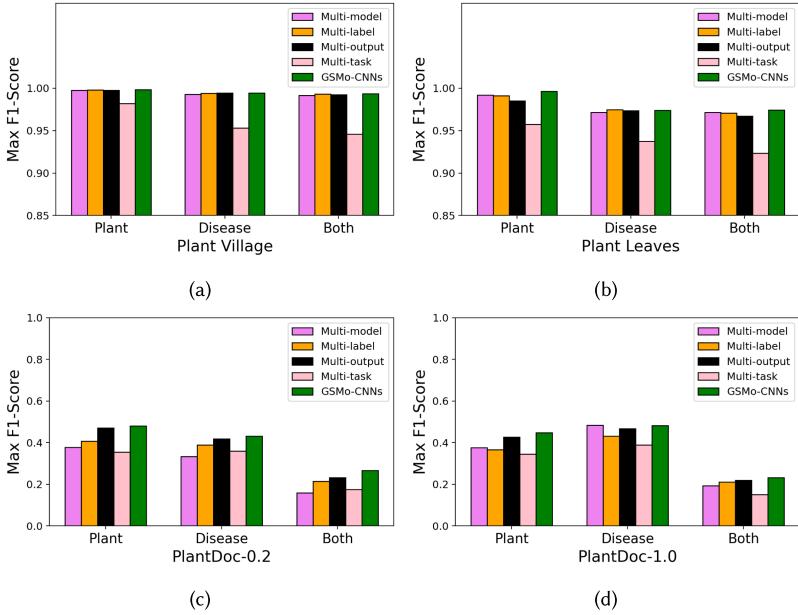


Fig. 10. comparison of different approaches: multi-model, multi-label, multi-output, multi-task, and GS Mo-CNNs (Our methods).

identification (Plant), disease classification (Disease), or combined plant and disease prediction (Both). The related results can be checked in Tables 5, 6, and 7. As mentioned in Section 3.3, the evaluation indicator in this part is F1-score, which is more suitable for evaluating forecasts with imbalanced data. We can see that using dual models, one for each task, is not really advantageous as it has lower performance than approaches using single CNNs for both tasks in most cases. Compared to multi-label, multi-model is only better in Plant Leaves for plant identification, and in PlantDoc-1.0 for plant identification and combined prediction. Multi-model is shown to beat multi-output in Plant Leaves but is inferior in Plant Village, PlantDoc-0.2, and PlantDoc-1.0. Compared to our methods, multi-model achieves lower performance in all datasets. The above results demonstrate the potential and superiority of the single-model approaches, where using shared backbone CNNs for multi-predictions can be more effective than training individual CNNs, each for prediction of a label. We attribute the advantages of single models over a dual models are they can learn the common features of plant species classification and disease classification, which can influence each other efficiently. A single model have an end-to-end structure where the backbone can help to learn common features while the prediction layers will be optimised to learn task-specific features. By doing this, a single model can achieve better generalisation.

**4.5.3 Learning Approaches.** Figure 10 also shows F1-score of different approaches (the related results have been detailed in Tables 5, 6, and 7). As mentioned in Section 3.3, the F1 score is better suited for evaluating prediction results on imbalanced data compared to Accuracy. Overall, the proposed GS Mo-CNNs achieve the best results in all datasets, followed by multi-output and multi-label. Multi-output is comparable to multi-label in Plant Village and slightly better in Plant Leaves, however, multi-label is better in PlantDoc-0.2 and PlantDoc-1.0. Compared to existing methods, as summarised in Table 3 we achieve state-of-the-art results. The only approach achieves 99.98% validation accuracy on Plant Village is AlexNet+SVM in [42]. However, this is a hybrid approach and the result is for a validation set, and it is not clear how the data is partitioned for evaluation.

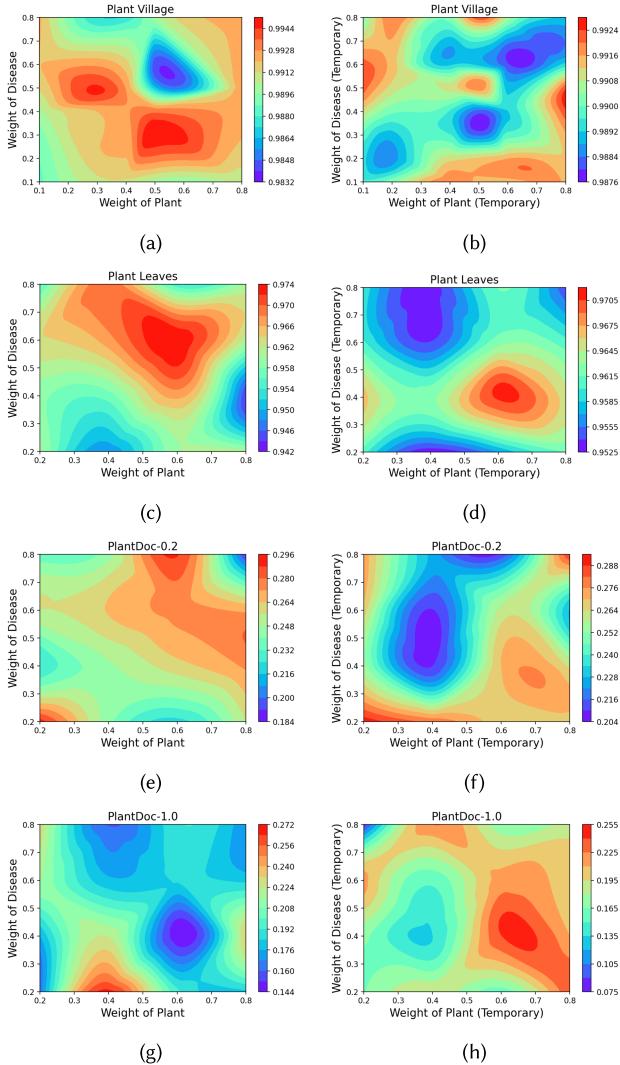


Fig. 11. combinations of balance weights for plant prediction and disease prediction.

The remarkable performance of GSMo-CNNs can be attributed to several factors. First, it uses a chain of prediction to jointly predict species and diseases. This would enhance feature learning and relationships of plant species and disease types, and facilitates feature sharing, enabling the model to extract general image characteristics effectively; Second, the BW were employed to address class imbalances, ensuring accurate predictions for all classes. These theoretical and technical advantages collectively make GSMo-CNNs an advantageous approach for plant species identification and disease classification.

**4.5.4 Balance Weights.** As presented in Section 3.2, GSMo-CNN can use BW for the loss functions. We show the heat maps for the BW of plant and disease on the final prediction layer and the BW of plant and disease on the temporary layer in Figure 11. The hotter colour represents the higher F1-score. As we can see, in general, the BW for plant and disease need to be analysed

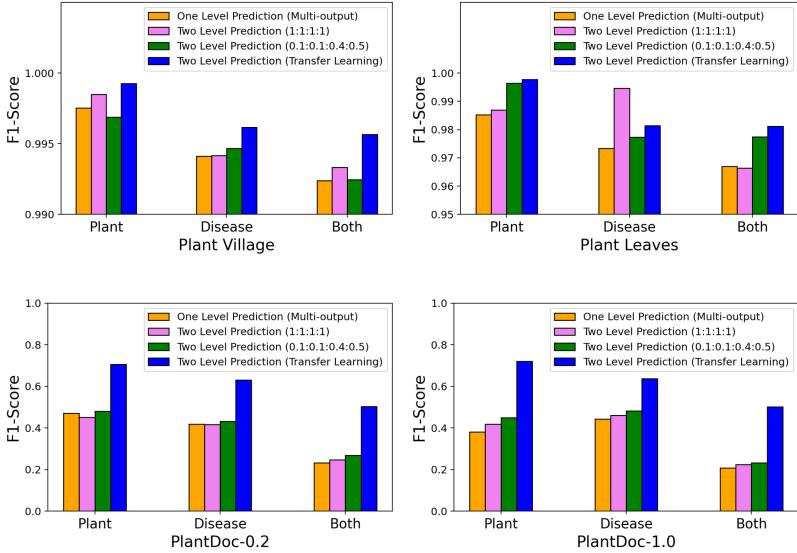


Fig. 12. comparison of 1st and 2nd prediction layer.

specifically, as different tasks have varying weight values. For example, in Figure 11(a), the highest F1-scores of GSMo-CNN in the Plant Village dataset appear in 0.3 **Weights of Plant (WP)** and 0.5 **Weights of Disease (WD)** and 0.5 - 0.6 WP and 0.2 - 0.4 WD. But in Figure 11(c), the highest F1-scores of GSMo-CNN in the Plant Leaves dataset appear in 0.6 WP and 0.6 WD, which trend to similar.

**4.5.5 Prediction Layer.** In Figure 12, we show that by stacking prediction layers we can achieve the performance and the BW can help increase F1-score. Finally, we show that transfer learning would be useful in this study.

As we can see, with one prediction level we have the multi-output approach which achieves good results on Plant Village and Plant Leaves, although the performance on PlantDoc-0.2 and PlantDoc-1.0 is not high it is still comparable or better than multi-model, multi-label, and multi-task approaches. When adding a stack of another prediction layer on top of the first (temporary) prediction layer with cross connections: disease (temporary) → plant; plant (temporary) → disease, we can achieve the improvement in all cases, except plant identification and disease classification in PlantDoc-0.2. With BW ( $\beta_1 : \delta_1 : \beta_2 : \delta_2 = 0.1 : 0.1 : 0.4 : 0.5$ ) for the training loss functions, further improvement can be seen in 9 out of 12 cases. The other three cases are plant identification (Plant Village), disease classification (Plant Leaves), and combined prediction (Plant Village) where the BW do not show their advantage. Finally, the usefulness of transfer learning is apparent as it only fails to increase the performance of disease classification on Plant Leaves. Especially on PlantDoc-0.2 and PlantDoc-1.0, we can achieve significant improvement with large margins.

## 5 CONCLUSION AND FUTURE WORK

We presented a comprehensive survey and empirical study on DL for plant identification and disease classification from leaf images. The article aims to address the gaps in modern plant pathology where cutting-edge technologies such as DL have been adopted largely but there is still a lack of benchmarking studies. Besides reviewing currently used methods, we also show the available methods from machine learning literature which have not been employed for plant identification

and/or disease classification. Furthermore, we investigate the hypothesis that a single model for multi-prediction would be more useful than multiple models, each for a task, in terms of implementation, computation, memory saving, and effectiveness. To this end, we categorise different approaches into multi-model, multi-label, multi-output, and multi-task where different CNN structures can be the backbone, if applicable. For completeness, we also propose a new model, based on the multi-output approach and the idea of classifier chain, by stacking and cross-connecting prediction layers. We run intensive experiments to evaluate and compare these backbone models and approaches in uniform settings. We found that:

- InceptionV3 is a best choice for a backbone CNN in this study as it performs better than AlexNet, VGG16, ResNet101, MobileNetV2, EfficientNet, ViT, and our custom CNN.
- Using a single model for both tasks is more useful than using separate models for each of them. Single models can be more convenient, that is, easier for model selection, more memory saving, more efficient, and we also achieve better results.
- Stacking and cross-connecting prediction layers can improve the accuracy and F1-score for plant identification and disease classification. This approach is also flexible where BW can be applied to search for a better combination of the loss functions at the prediction layers.
- Transfer learning is promising. We showed that by transferring InceptionV3's weights trained on ImageNet we can improve the performance of our new model significantly.

The future directions of this research will be extensive and have the potential to significantly benefit the agricultural industry. First, we can expand the idea of a hierarchical combination of labels for further improvement. One idea is to pioneer the development of an algorithm capable of autonomously determining optimal BW. This innovation holds immense potential, particularly in scenarios characterized by a multitude of publicly available datasets, each associated with diverse plant species and diseases. Additionally, we can explore the integration of deep supervision to smooth the gradients during the training at different prediction levels. Here, at each output layer, we compute a prediction loss and add it to the final loss function. A preliminary result of deep supervision for training a classifier chain can be found in [104], showcasing a potential for future improvements.

Second, we found that each dataset has different sets of plant species and disease types. While we can build a model on each dataset for a narrow task, it is unable to deploy it to another task for new plants and diseases. We intend to investigate deeper into the idea of life-long learning, seeking to uncover the full extent of its advantages and applications. A foundation model for plant pathology can be developed so that it can continually acquire and adapt knowledge over time based on neurosymbolic computing [20, 93]. In particular, the model can learn from newly available datasets without re-training from the previously learned datasets. At the same time, one can distill knowledge from the model for different downstream tasks.

We believe that these efforts and endeavours will propel our research further, leading to new breakthroughs that will benefit the field of PA and SA.

## REFERENCES

- [1] Mohit Agarwal, Suneet K. r. Gupta, and K. K. Biswas. 2019. Grape disease identification using convolution neural network. In *Proceedings of the 2019 23rd International Computer Science and Engineering Conference*. 224–229. DOI: <https://doi.org/10.1109/ICSEC47112.2019.8974752>
- [2] Mohit Agarwal, Abhishek Singh, Siddhartha Arjaria, Amit Sinha, and Suneet Gupta. 2020. ToLeD: Tomato leaf disease detection using convolution neural network. *Procedia Computer Science* 167, 2 (2020), 293–301. Retrieved from <https://login.ezproxy.utas.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edselp&AN=S1877050920306906&site=eds-live>
- [3] T ANANDHAKRISHNAN and SM JAISAKTHI. 2020. Identification of tomato leaf disease detection using pre-trained deep convolutional neural network models. *Scalable Computing: Practice and Experience* 21, 4 (2020), 625–635.

Retrieved from <https://login.ezproxy.utas.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=aps&AN=147704496&ssite=eds-live>

- [4] Sujith Ariyapadath. 2021. Plant leaf classification and comparative analysis of combined feature set using machine learning techniques. *Traitemen du Signal* 38, 6 (01 Dec 2021), 1587–1598.
- [5] Surampalli Ashok, Gemini Kishore, Velpula Rajesh, S. Suchitra, S. G. Gino Sophia, and B. Pavithra. 2020. Tomato leaf disease detection using deep learning techniques. In *Proceedings of the 2020 5th International Conference on Communication and Electronics Systems*. 979–983. DOI : <https://doi.org/10.1109/ICCES48766.2020.9137986>
- [6] Muhammad Azfar Firdaus Azlah, Lee Suan Chua, Fakhrul Razan Rahmad, Farah Izana Abdullah, and Sharifah Rafidah Wan Alwi. 2019. Review on techniques for plant leaf classification and recognition. *Computers* 8, 4 (2019). DOI : <https://doi.org/10.3390/computers8040077>
- [7] Stefania Barburiceanu, Romulus Terebes, and Serban Meza. 2020. Grape leaf disease classification using LBP-derived texture operators and colour. In *Proceedings of the 2020 IEEE International Conference on Automation, Quality and Testing, Robotics*. 1–6. DOI : <https://doi.org/10.1109/AQTR49680.2020.9130019>
- [8] Anil A. Bharate and M. S. Shirdhonkar. 2020. Classification of grape leaves using KNN and SVM classifiers. In *Proceedings of the 2020 4th International Conference on Computing Methodologies and Communication*. 745–749. DOI : <https://doi.org/10.1109/ICCMC48092.2020.ICCMC-000139>
- [9] Shyamtanu Bhowmik, Anjan Kumar Talukdar, and Kandarpa Kumar Sarma. 2020. Detection of disease in tea leaves using convolution neural network. In *Proceedings of the 2020 Advanced Communication Technologies and Signal Processing*. 1–6. DOI : <https://doi.org/10.1109/ACTS49415.2020.9350413>
- [10] Paarth Bir, Rajesh Kumar, and Ghanshyam Singh. 2020. Transfer learning based tomato leaf disease detection for mobile applications. In *Proceedings of the 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*. 34–39. DOI : <https://doi.org/10.1109/GUCON48875.2020.9231174>
- [11] Vandana Chaudhari and Manoj Patil. 2020. Banana leaf disease detection using K-means clustering and Feature extraction techniques. In *Proceedings of the 2020 International Conference on Advances in Computing, Communication Materials*. 126–130. DOI : <https://doi.org/10.1109/ICACCM50413.2020.9212816>
- [12] Wen-Liang Chen, Yi-Bing Lin, Fung-Ling Ng, Chun-You Liu, and Yun-Wei Lin. 2020. RiceTalk: Rice blast detection using internet of things and artificial intelligence technologies. *IEEE Internet of Things Journal* 7, 2 (2020), 1001–1010. DOI : <https://doi.org/10.1109/JIOT.2019.2947624>
- [13] Davide Chicco and Giuseppe Jurman. 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* 21, 1 (2020), 1–13.
- [14] Siddharth Singh Chouhan, Ajay Kaul, Uday Pratap Singh, and Sanjeev Jain. 2019. A Database of Leaf Images: Practice Towards Plant Conservation with Plant Pathology. In *Proceedings of the IEEE 4th International Conference on Information Systems and Computer Networks*. DOI : <https://doi.org/10.17632/hb74ynkjen.1>
- [15] Siddharth Singh Chouhan, Uday Pratap Singh, and Sanjeev Jain. 2020. Applications of computer vision in plant pathology: A survey. *Archives of Computational Methods in Engineering: State of the Art Reviews* 27, 2 (2020), 611. Retrieved from <https://login.ezproxy.utas.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edssjs&AN=edssjs.15D0A2D2&suite=eds-live>
- [16] Siddharth Singh Chouhan, Uday Pratap Singh, and Sanjeev Jain. 2021. Automated plant leaf disease detection and classification using fuzzy based function network. *Wireless Personal Communications: An International Journal* 121, 3 (2021), 1757–1779. Retrieved from <https://login.ezproxy.utas.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edssjs&AN=edssjs.8410D5A3&suite=eds-live>
- [17] Muhammad E. H. Chowdhury, Tawsifur Rahman, Amith Khandakar, Mohamed Arselene Ayari, Aftab Ullah Khan, Muhammad Salman Khan, Nasser Al-Emadi, Mamun Bin Ibne Reaz, Mohammad Tariqul Islam, and Sawal Hamid Md Ali. 2021. Automatic and reliable leaf disease detection using deep learning techniques. *AgriEngineering* 3, 2 (2021), 294–312. Retrieved from <https://login.ezproxy.utas.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=asn&AN=151077308&suite=eds-live>
- [18] Hanh Dang-Ngoc, Trang N. M. Cao, and Chau Dang-Nguyen. 2021. Citrus leaf disease detection and classification using hierarchical support vector machine. In *Proceedings of the 2021 International Symposium on Electrical and Electronics Engineering*. 69–74. DOI : <https://doi.org/10.1109/ISEE51682.2021.9418680>
- [19] Debasish Das, Mahinderpal Singh, Sarthak Swaroop Mohanty, and S. Chakravarty. 2020. Leaf disease detection using support vector machine. In *Proceedings of the 2020 International Conference on Communication and Signal Processing*. 1036–1040. DOI : <https://doi.org/10.1109/ICCSPI48568.2020.9182128>
- [20] Artur S. d'Avila Garcez, Marco Gori, L. Lamb, Luciano Serafini, Michael Spranger, and S. Tran. 2019. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *FLAP* 6, 4 (2019), 611–632. Retrieved from <https://api.semanticscholar.org/CorpusID:155092677>
- [21] Alessandro dos Santos Ferreira, Daniel Matte Freitas, Gercina Gonçalves da Silva, Hemerson Pistori, and Marcelo Theophilo Folhes. 2017. Weed detection in soybean crops using ConvNets. *Computers and Electronics in Agriculture* 143, 12 (2017), 314–324. DOI : <https://doi.org/10.1016/j.compag.2017.10.027>

- [22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*. Retrieved from <https://openreview.net/forum?id=YicbFdNTTy>
- [23] EMTYK Ekanayake and RD Nawarathna. 2021. Novel deep learning approaches for crop leaf disease classification: A review. In *Proceedings of the 2021 International Research Conference on Smart Computing and Systems Engineering*, Vol. 4. 49–52. DOI :<https://doi.org/10.1109/SCSE53661.2021.9568324>
- [24] Yan Fu, Xutao Li, and Yunming Ye. 2020. A multi-task learning model with adversarial data augmentation for classification of fine-grained images. *Neurocomputing* 377, 7 (2020), 122–129. DOI :<https://doi.org/10.1016/j.neucom.2019.10.002>
- [25] Geetharamani G. and Arun Pandian J. 2019. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Computers and Electrical Engineering* 76, 4 (2019), 323–338. Retrieved from <https://login.ezproxy.utas.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edselp&AN=S0045790619300023&site=eds-live>
- [26] Haridas D. Gadade and D. K. Kirange. 2020. Tomato leaf disease diagnosis and severity measurement. In *Proceedings of the 2020 4th World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. 318–323. DOI :<https://doi.org/10.1109/WorldS450073.2020.9210294>
- [27] Ruchi Gajjar, Nagendra Gajjar, Vaibhavkumar Jigneshkumar Thakor, Nikhil Kumar Pareshbhai Patel, and Stavan Ruparelia. 2021. Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform. *The Visual Computer: International Journal of Computer Graphics* 38, 8 (2021), 2923–2938. Retrieved from <https://login.ezproxy.utas.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edssjs&AN=edssjs.4D1044AE&site=eds-live>
- [28] Jayme Garcia Arnal Barbedo, Luciano Vieira Koenigkan, Bernardo Almeida Halfeld-Vieira, Rodrigo Veras Costa, Katia Lima Nechet, Claudia Vieira Godoy, Murillo Lobo Junior, Flavia Rodrigues Alves Patrício, Viviane Talamini, Luiz Gonzaga Chitarra, Saulo Alves Santos Oliveira, Alessandra Keiko Nakasone Ishida, Jose Mauricio Cunha Fernandes, Thiago Teixeira Santos, Fabio Rossi Cavalcanti, Daniel Terao, and Francislene Angelotti. 2018. Annotated Plant Pathology Databases for Image-Based Detection and Recognition of Diseases. *IEEE Latin America Transactions* 16, 6 (2018), 1749–1757. DOI :<https://doi.org/10.1109/TLA.2018.8444395>
- [29] Xulang Guan. 2021. A novel method of plant leaf disease detection based on deep learning and convolutional neural network. In *Proceedings of the 2021 6th International Conference on Intelligent Computing and Signal Processing*. 816–819. DOI :<https://doi.org/10.1109/ICSP51882.2021.9408806>
- [30] Sk Mahmudul Hassan, Arnab Kumar Maji, Michal Jasiński, Zbigniew Leonowicz, and Elżbieta Jasińska. 2021. Identification of plant-leaf diseases using CNN and transfer-learning approach. *Electronics* 10, 12 (2021), 1388. DOI :<https://doi.org/10.3390/electronics10121388>
- [31] Mohammed Hassanin, Ibrahim Radwan, Salman Khan, and Murat Tahtali. 2021. Learning Discriminative Representations for Multi-Label Image Recognition. arXiv:[2107.11159](https://arxiv.org/abs/2107.11159). Retrieved from <https://arxiv.org/abs/2107.11159>
- [32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*. 770–778. DOI :<https://doi.org/10.1109/CVPR.2016.90>
- [33] Gensheng Hu, Huaiyu Wang, Yan Zhang, and Mingzhu Wan. 2021. Detection and severity analysis of tea leaf blight based on deep learning. *Computers and Electrical Engineering* 90 (2021). Retrieved from <https://login.ezproxy.utas.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edselp&AN=S0045790621000471&site=eds-live>
- [34] Zhaohua Huang, Ally Qin, Jingshu Lu, Aparna Menon, and Jerry Gao. 2020. Grape leaf disease detection and classification using machine learning. In *Proceedings of the 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybernetics (Cybernetics)*. 870–877. DOI :<https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybernetics50389.2020.00150>
- [35] David P. Hughes and Marcel Salathé. 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing. arXiv:[1511.08060](https://arxiv.org/abs/1511.08060). Retrieved from [http://arxiv.org/abs/1511.08060](https://arxiv.org/abs/1511.08060)
- [36] David. P. Hughes and Marcel Salathe. 2016. An Open Access Repository of Images on Plant Health to Enable the Development of Mobile Disease Diagnostics. arXiv:[1511.08060](https://arxiv.org/abs/1511.08060). Retrieved from [http://arxiv.org/abs/1511.08060](https://arxiv.org/abs/1511.08060)
- [37] S. JANA, A. RIJUVANA BEGUM, S. SELVAGANESAN, and P. SURESH. 2021. Deep belief network based disease detection in pepper leaf for farming sector. *Turkish Journal of Physiotherapy Rehabilitation* 32, 2 (2021), 994–1002. Retrieved from <https://login.ezproxy.utas.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=ccm&AN=151006060&site=eds-live>

- [38] P. Jasitha, M. R. Dileep, and M Divya. 2019. Venation based plant leaves classification using googlenet and VGG. In *Proceedings of the 2019 4th International Conference on Recent Trends on Electronics, Information, Communication, and Technology*. 715–719. DOI : <https://doi.org/10.1109/RTEICT46194.2019.9016966>
- [39] K. JAYAPRAKASH and S. P. BALAMURUGAN. 2021. An efficient hand-crafted features with machine learning-based plant leaf disease diagnosis and classification model. *Turkish Journal of Physiotherapy Rehabilitation* 32, 2 (2021), 1683–1698. Retrieved from <https://login.ezproxy.utas.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=ccm&AN=151006151&site=eds-live>
- [40] Paul Shekonya Kanda, Kewen Xia, and Olanrewaju Hazzan Sanusi. 2021. A deep learning-based recognition technique for plant leaf classification. *IEEE Access* 9, 1 (2021), 162590–162613. DOI : <https://doi.org/10.1109/ACCESS.2021.3131726>
- [41] Gugan Kathiresan, M. Anirudh, M. Nagharjun, and R. Karthik. 2021. Disease detection in rice leaves using transfer learning techniques. *Journal of Physics: Conference Series* 1911, 1 (May 2021), 012004. DOI : <https://doi.org/10.1088/1742-6596/1911/1/012004>
- [42] Mihir Kawatra, Shreyas Agarwal, and Raghu Kapur. 2020. Leaf disease detection using neural network hybrid models. In *Proceedings of the 2020 IEEE 5th International Conference on Computing Communication and Automation*. 225–230. DOI : <https://doi.org/10.1109/ICCCA49541.2020.9250885>
- [43] Kirti and Navin Rajpal. 2020. Black rot disease detection in grape plant (*vitis vinifera*) using colour based segmentation amp; machine learning. In *Proceedings of the 2020 2nd International Conference on Advances in Computing, Communication Control and Networking*. 976–979. DOI : <https://doi.org/10.1109/ICACCN51052.2020.9362812>
- [44] N. KRISHNAMOORTHY and V. R. LOGA PARAMESWARI. 2021. Rice leaf disease detection via deep neural networks with transfer learning for early identification. *Turkish Journal of Physiotherapy Rehabilitation* 32, 2 (2021), 1087–1097. Retrieved from <https://login.ezproxy.utas.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=ccm&AN=151006069&site=eds-live>
- [45] Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. John Kress, Ida C. Lopez, and João V. B. Soares. 2012. Leafsnap: A computer vision system for automatic plant species identification. In *Proceedings of the ECCV 2012*. Springer Berlin Heidelberg, Berlin, Heidelberg, 502–516.
- [46] Sandeep Kumar, KMVV Prasad, A. Srilekha, T. Suman, B. Pranav Rao, and J. Naga Vamshi Krishna. 2020. Leaf Disease Detection and Classification based on Machine Learning. In *Proceedings of the 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics*. 361–365. DOI : <https://doi.org/10.1109/ICSTCEE49637.2020.9277379>
- [47] Vinod Kumar, Hritik Arora, Harsh, and Jatin Sisodia. 2020. ResNet-based approach for Detection and classification of plant leaf diseases. In *Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems*. 495–502. DOI : <https://doi.org/10.1109/ICESC48915.2020.9155585>
- [48] R. Kavitha lakshmi and S. Nickolas. 2020. Deep learning based betelvine leaf disease detection (*piper betlel*). In *Proceedings of the 2020 IEEE 5th International Conference on Computing Communication and Automation*. 215–219. DOI : <https://doi.org/10.1109/ICCCA49541.2020.9250911>
- [49] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324. DOI : <https://doi.org/10.1109/5.726791>
- [50] Sue Han Lee, Chee Seng Chan, Paul Wilkin, and Paolo Remagnino. 2015. Deep-plant: Plant identification with convolutional neural networks. In *Proceedings of the 2015 IEEE International Conference on Image Processing*. 452–456. DOI : <https://doi.org/10.1109/ICIP.2015.7350839>
- [51] Sue Han Lee, Hervé Goëau, Pierre Bonnet, and Alexis Joly. 2021. Conditional multi-task learning for plant disease identification. In *Proceedings of the 2020 25th International Conference on Pattern Recognition*. 3320–3327. DOI : <https://doi.org/10.1109/ICPR48806.2021.9412643>
- [52] Lili Li, Shujuan Zhang, and Bin Wang. 2021. Plant disease detection and classification by deep learning—A Review. *IEEE Access* 9, 1 (2021), 56683–56698. DOI : <https://doi.org/10.1109/ACCESS.2021.3069646>
- [53] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10713–10722.
- [54] Shikun Liu, Edward Johns, and Andrew J. Davison. 2019. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- [55] Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Sašo Džeroski. 2012. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition* 45, 9 (2012), 3084–3104. DOI : <https://doi.org/10.1016/j.patcog.2012.03.004> Best Papers of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'2011).
- [56] Vishakha A. Metre and Sudhir D. Sawarkar. 2021. Research opportunities for the detection and classification of plant leaf diseases. In *Proceedings of the 2021 2nd Global Conference for Advancement in Technology*. 1–8. DOI : <https://doi.org/10.1109/GCAT52182.2021.9587775>

- [57] Vishakha A. Metre and Sudhir D. Sawarkar. 2022. Reviewing important aspects of plant leaf disease detection and classification. In *Proceedings of the 2022 International Conference for Advancement in Technology*. 1–8. DOI :<https://doi.org/10.1109/ICONAT53423.2022.9725870>
- [58] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- [59] Somnath Mukhopadhyay, Munti Paul, Ramen Pal, and Debasish De. 2021. Tea leaf disease detection using multi-objective image segmentation. *Multimedia Tools and Applications: An International Journal* 80, 1 (2021), 753. Retrieved from <https://login.ezproxy.utas.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edssjs&AN=edssjs.30A9493B&site=eds-live>
- [60] Trishen Munisami, Mahess Ramsurn, Somveer Kishnah, and Sameerchand Pudaruth. 2015. Plant leaf recognition using shape features and colour histogram with k-nearest neighbour classifiers. *Procedia Computer Science* 58, 16 (2015), 740–747. DOI :<https://doi.org/10.1016/j.procs.2015.08.095> Second International Symposium on Computer Vision and the Internet (VisionNet'15).
- [61] H. B. Muregan, A. M. Coroiu, and A. D. Călin. 2020. Detecting leaf plant diseases using deep learning: A review. In *Proceedings of the 2020 International Conference on Software, Telecommunications and Computer Networks*. 1–6. DOI :<https://doi.org/10.23919/SoftCOM50211.2020.9238318>
- [62] Ernest Mwebaze, Timnit Gebru, Andrea Frome, Solomon Nsumba, and Jeremy Tusubira. 2019. iCassava 2019 Fine-Grained Visual Categorization Challenge. arXiv:[1908.02900](https://arxiv.org/abs/1908.02900). Retrieved from <https://arxiv.org/abs/1908.02900>
- [63] Petr Novotný and Tomáš Suk. 2013. Leaf recognition of woody species in Central Europe. *Biosystems Engineering* 115, 4 (2013), 444–452. DOI :<https://doi.org/10.1016/j.biosystemseng.2013.04.007>
- [64] Nor Azlan Othman, Nor Salwa Damanhuri, Nabilah Md Ali, Belinda Chong Chiew Meng, and Ahmad Asri Abd Samat. 2022. Plant leaf classification using convolutional neural network. In *Proceedings of the 2022 8th International Conference on Control, Decision and Information Technologies*, Vol. 1. 1043–1048. DOI :<https://doi.org/10.1109/CoDIT55151.2022.9804121>
- [65] Pranjali B. Padol and Anjali A. Yadav. 2016. SVM classifier based grape leaf disease detection. In *Proceedings of the 2016 Conference on Advances in Signal Processing*. 175–179. DOI :<https://doi.org/10.1109/CASP.2016.7746160>
- [66] Bihari Nandan Pandey, Raghvendra Pratap Singh, Mahima Shanker Pandey, and Sachin Jain. 2023. Cotton leaf disease classification using deep learning based novel approach. In *Proceedings of the 2023 International Conference on Disruptive Technologies*. 559–561.
- [67] Ananda S. Paymode, Shyamsundar P. Magar, and Vandana B. Malode. 2021. Tomato leaf disease detection and classification using convolution neural network. In *Proceedings of the 2021 International Conference on Emerging Smart Computing and Informatics*. 564–570. DOI :<https://doi.org/10.1109/ESCI50559.2021.9397001>
- [68] Vijayakumar Ponnusamy, Amrith Coumaran, Akhash Subramanian Shunmugam, Kritin Rajaram, and Sanoj Senthivelavan. 2020. Smart glass: Real-time leaf disease detection using yolo transfer learning. In *Proceedings of the 2020 International Conference on Communication and Signal Processing*. 1150–1154. DOI :<https://doi.org/10.1109/ICCPSP48568.2020.9182146>
- [69] Xiao Qin, Yu Shi, Xiao Huang, Huiting Li, Jiangtao Huang, Changan Yuan, and Chunxia Liu. 2021. Attention-based deep multi-scale network for plant leaf recognition. In *Proceedings of the Intelligent Computing Theories and Application*, De-Shuang Huang, Kang-Hyun Jo, Jianqiang Li, Valeriya Gribova, and Vitoantonio Bevilacqua (Eds.). Springer International Publishing, Cham, 302–313.
- [70] Sakshi Raina and Abhishek Gupta. 2021. A study on various techniques for plant leaf disease detection using leaf image. In *Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems*. 900–905. DOI :<https://doi.org/10.1109/ICAIS50930.2021.9396023>
- [71] B. Rajesh, M. Vishnu Sai Vardhan, and L. Sujihelen. 2020. Leaf disease detection and classification by decision tree. In *Proceedings of the 2020 4th International Conference on Trends in Electronics and Informatics(48184)*. 705–708. DOI :<https://doi.org/10.1109/ICOEI48184.2020.9142988>
- [72] Hafiz Tayyab Rauf, Basharat Ali Saleem, M. Ikram Ullah Lali, Muhammad Attique Khan, Muhammad Sharif, and Syed Ahmad Chan Bukhari. 2019. A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning. *Data in Brief* 26, 5 (2019), 2352–3409. DOI :<https://doi.org/10.1016/j.dib.2019.104340>
- [73] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2009. Classifier chains for multi-label classification. In *Proceedings of the Machine Learning and Knowledge Discovery in Databases*, Wray Buntine, Marko Grobelnik, Dunja Mladenić, and John Shawe-Taylor (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 254–269.
- [74] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (Eds.). Springer International Publishing, Cham, 234–241.

- [75] Dilli Charan Sai. 2021. A review on crop disease detection using deep learning. *Journal of Network and Information Security* 9, 1 (2021), 18–22. Retrieved from <https://login.ezproxy.utas.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=aps&AN=148445196&site=eds-live>
- [76] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- [77] R. Vijaya Saraswathi, Vaishali Bitla, P. Radhika, and T. Navneeth Kumar. 2021. Leaf disease detection and remedy suggestion using convolutional neural networks. In *Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication*. 788–794. DOI : <https://doi.org/10.1109/ICCMC51019.2021.9418013>
- [78] Melike Sardogan, Adem Tuncer, and Yunus Ozen. 2018. Plant leaf disease detection and classification based on CNN with LVQ algorithm. In *Proceedings of the 2018 3rd International Conference on Computer Science and Engineering (UBMK)*. 382–385. DOI : <https://doi.org/10.1109/UBMK.2018.8556635>
- [79] S. M. Shahidur Harun Rumy, Md. Ishan Arefin Hossain, Forji Jahan, and Tanjina Tanvin. 2021. An IoT based system with edge intelligence for rice leaf disease detection using machine learning. In *Proceedings of the 2021 IEEE International IOT, Electronics and Mechatronics Conference*. 1–6. DOI : <https://doi.org/10.1109/IEMTRONICS52119.2021.9422499>
- [80] Pushkara Sharma, Pankaj Hans, and Subhash Chand Gupta. 2020. Classification of plant leaf diseases using machine learning and image preprocessing techniques. In *Proceedings of the 2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence)*. 480–484. DOI : <https://doi.org/10.1109/Confluence47617.2020.9057889>
- [81] Ankita Shelke and Ninad Mehendale. 2022. A CNN-based android application for plant leaf classification at remote locations. *Neural Computing and Applications* 35, 3 (2023), 2601–2607. DOI : <https://doi.org/10.1007/s00521-022-07740-1>
- [82] Yusuke Shinohara. 2016. Adversarial multi-task learning of deep neural networks for robust speech recognition. In *Proceedings of the Interspeech 2016*. 2369–2372. DOI : <https://doi.org/10.21437/Interspeech.2016-879>
- [83] Davinder Singh, Naman Jain, Pranjali Jain, Pratik Kayal, Sudhakar Kumawat, and Nipun Batra. 2020. PlantDoc: A dataset for visual plant disease detection. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD* (Jan 2020). DOI : <https://doi.org/10.1145/3371158.3371196>
- [84] Ujjwal Singh, Anuj Srivastava, Divyanshu Chauhan, and Astha Singh. 2020. Computer vision technique for detection of grape esca (black measles) disease from grape leaf samples. In *Proceedings of the 2020 International Conference on Contemporary Computing and Applications*. 110–115. DOI : <https://doi.org/10.1109/IC3A48958.2020.233281>
- [85] Oskar Söderkvist. 2001. *Computer Vision Classification of Leaves from Swedish Trees*. 74 pages.
- [86] R. Sujatha, Jyotir Moy Chatterjee, N. Z. Jhanjhi, and Sarfraz Nawaz Brohi. 2021. Performance of deep learning vs machine learning in plant leaf disease detection. *Microprocessors and Microsystems* 80, 1 (2021), 103615. DOI : <https://doi.org/10.1016/j.micpro.2020.103615>
- [87] C. K. Sunil, C. D. Jaidhar, and Nagamma Patil. 2020. Empirical study on multi convolutional layer-based convolutional neural network classifier for plant leaf disease detection. In *Proceedings of the 2020 IEEE 15th International Conference on Industrial and Information Systems*. 460–465. DOI : <https://doi.org/10.1109/ICIIS51140.2020.9342729>
- [88] Rafi Surya and Elliana Gautama. 2020. Cassava leaf disease detection using convolutional neural networks. In *Proceedings of the 2020 6th International Conference on Science in Information Technology*. 97–102. DOI : <https://doi.org/10.1109/ICSI Tech49800.2020.9392051>
- [89] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*. 1–9. DOI : <https://doi.org/10.1109/CVPR.2015.7298594>
- [90] JING WEI TAN and Siow-Wee Chang. 2018. D-leaf dataset. (1 2018). DOI : <https://doi.org/10.6084/m9.figshare.5732955.v1>
- [91] Rajasekaran Thangaraj, P. Pandiyan, S. Anandamurugan, and Sivaramakrishnan Rajendar. 2023. A deep convolution neural network model based on feature concatenation approach for classification of tomato leaf disease. *Multimedia Tools and Applications* (2023), 1. DOI : <https://doi.org/10.1007/s11042-023-16347-0>
- [92] Khaing Zin Thet, Khine Khine Htwe, and Myint Myint Thein. 2020. Grape leaf diseases classification using convolutional neural network. In *Proceedings of the 2020 International Conference on Advanced Information Technologies*. 147–152. DOI : <https://doi.org/10.1109/ICAIT51105.2020.9261801>
- [93] Son N. Tran and Artur d'Avila Garcez. 2023. Neurosymbolic reasoning and learning with restricted boltzmann machines. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 5 (Jun. 2023), 6558–6565. DOI : <https://doi.org/10.1609/aaai.v37i5.25806>
- [94] Amrita S. Tulshan and Nataasha Raul. 2019. Plant leaf disease detection using machine learning. In *Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies*. 1–6. DOI : <https://doi.org/10.1109/ICCCNT45670.2019.8944556>

- [95] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. 2021. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 7 (2021), 3614–3633. DOI : <https://doi.org/10.1109/tpami.2021.3054719>
- [96] Meenakshi Srivastava and Jasraj Meena. 2023. Plant leaf disease detection and classification using modified transfer learning models. *Multimedia Tools and Applications* (2023), 1. DOI : <https://doi.org/10.1007/s11042-023-16929-y>
- [97] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. Retrieved from [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fdb053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fdb053c1c4a845aa-Paper.pdf)
- [98] T. Vijaykanth Reddy and K Sashi Rekha. 2021. Deep leaf disease prediction framework (DLDPF) with transfer learning for automatic leaf disease detection. In *Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication*. 1408–1415. DOI : <https://doi.org/10.1109/ICCMC51019.2021.9418245>
- [99] Gerson Vizcarra, Danitza Bermejo, Antoni Mauricio, Ricardo Zarate Gomez, and Erwin Dianderas. 2021. The peruvian amazon forestry dataset: A leaf image classification corpus. *Ecological Informatics* 62, 2 (2021), 101268. DOI : <https://doi.org/10.1016/j.ecoinf.2021.101268>
- [100] Radhika Wadhawan, Mayyank Garg, and Ashish Kumar Sahani. 2020. Rice plant leaf disease detection and severity estimation. In *Proceedings of the 2020 IEEE 15th International Conference on Industrial and Information Systems*. 455–459. DOI : <https://doi.org/10.1109/ICIIS51140.2020.9342653>
- [101] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. 2016. CNN-RNN: A unified framework for multi-label image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- [102] Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang, and Qiao-Liang Xiang. 2007. A leaf recognition algorithm for plant classification using probabilistic neural network. In *Proceedings of the 2007 IEEE International Symposium on Signal Processing and Information Technology*. 11–16. DOI : <https://doi.org/10.1109/ISSPIT.2007.4458016>
- [103] Xiaoyue Xie, Yuan Ma, Bin Liu, Jinrong He, Shuqin Li, and Hongyan Wang. 2020. A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. *Frontiers in Plant Science* 11, 1 (2020), 751. DOI : <https://doi.org/10.3389/fpls.2020.00751>
- [104] Jianping Yao and Son N. Tran. 2023. Multi-output deep-supervised classifier chains for plant pathology. In *Proceedings of the 2023 International Joint Conference on Neural Networks*. 1–8. DOI : <https://doi.org/10.1109/IJCNN54540.2023.10191302>
- [105] Jianping Yao, Son N. Tran, Samantha Sawyer, and Saurabh Garg. 2023. Machine learning for leaf disease classification: Data, techniques and applications. *Artificial Intelligence Review* 56, suppl 3 (2023), 3571–3616. DOI : <https://doi.org/10.1007/s10462-023-10610-4>
- [106] Algani Yousef Methkal Abd, Caro Orlando Juan Marquez, Bravo Liz Maribel Robladillo, Kaur Chamandeep, Ansari Mohammed Saleh Al, and B. Kiran Bala. 2023. Leaf disease identification and classification using optimized deep learning. 25 (2023). DOI : <https://doi.org/10.1016/j.measen.2022.100643>
- [107] Xiao-Tong Yuan, Xiaobai Liu, and Shuicheng Yan. 2012. Visual classification with multitask joint sparse representation. *IEEE Transactions on Image Processing* 21, 10 (2012), 4349–4360. DOI : <https://doi.org/10.1109/TIP.2012.2205006>
- [108] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. 2013. Robust visual tracking via structured multi-task sparse learning. *International Journal of Computer Vision* 101, 2 (2013), 367–383. Retrieved from <https://login.ezproxy.utas.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edssjs&AN=edssjs.7520745&site=eds-live>
- [109] Xihai Zhang, Yue Qiao, Fanfeng Meng, Chengguo Fan, and Mingming Zhang. 2018. Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access* 6, 1 (2018), 30370–30377. DOI : <https://doi.org/10.1109/ACCESS.2018.2844405>
- [110] Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering* 34, 12 (2021), 5586–5609. DOI : <https://doi.org/10.1109/TKDE.2021.3070203>
- [111] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. 2014. Facial landmark detection by deep multi-task learning. In *Proceedings of the Computer Vision – European Conference on Computer Vision 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 94–108.

Received 4 November 2022; revised 14 December 2023; accepted 21 December 2023