



Automated diagnosis of diverse coffee leaf images through a stage-wise aggregated triple deep convolutional neural network

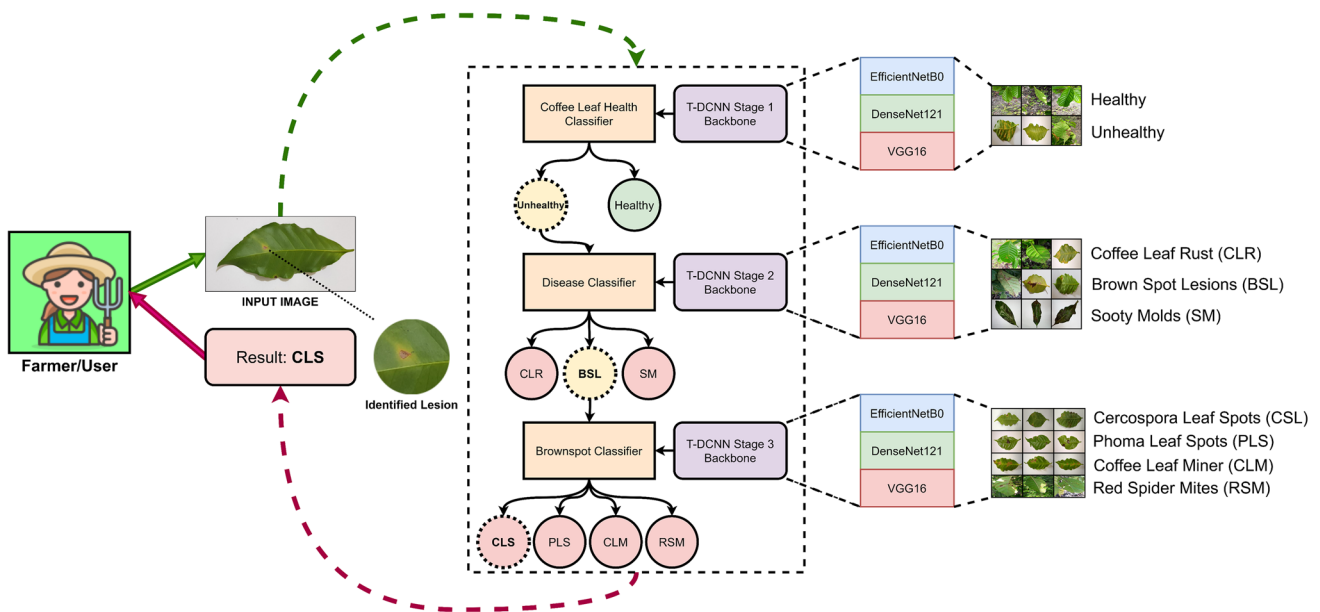
Francis Jesmar P. Montalbo¹

Received: 2 January 2021 / Revised: 20 September 2021 / Accepted: 17 December 2021 / Published online: 21 January 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

Due to the struggles of developing countries in coping with widespread coffee leaf diseases and infestations, the quality and quantity of coffee-based commodities have reduced significantly. This paper proposes a solution to this problem using Deep Convolutional Neural Networks (DCNN) that classifies seven coffee leaf conditions. Unlike other studies, this work proposed a novel Triple-DCNN (T-DCNN) composed of three aggregated DCNN models formed in an ensemble to produce lesser bias and better accuracy than standard models. Added to the proposed T-DCNN, an employed stage-wise approach narrowed down the classification options through a multi-staged structure and diversified the entire feature pool. Upon evaluation, the proposed Stage-Wise Aggregated T-DCNN (SWAT-DCNN) yielded successful diagnoses of diverse coffee leaf conditions in various environmental settings. Furthermore, with an overall accuracy of 95.98%, the SWAT-DCNN outperformed most state-of-the-art DCNNs that performed the same task.

Graphic abstract



Keywords Deep convolutional neural networks · Coffee leaf diseases · Stage-wise model · Image classification · Deep learning

Abbreviations

AUC Area Under the Curve

✉ Francis Jesmar P. Montalbo
francismontalbo@ieee.org

BCE	Binary Cross-Entropy
BrACoL	Brazilian Arabica Coffee Leaves
BSL	Brown Spot Lesions
CCE	Categorical Cross-Entropy
CE	Cross-Entropy
CLM	Coffee Leaf Miner
CLR	Coffee Leaf Rust
CLS	Cercospora Leaf Spots
CNN	Convolutional Neural Networks
DCNN	Deep Convolutional Neural Networks
DL	Deep Learning
FLOPS	Floating-Point Operations Per Second
FN	False Negatives
FP	False Positives
GAP	Global Average Pooling
Grad-CAM	Gradient-Weighted Class Activation Map
LiCoLe	Liberica Coffee Leaves
LR	Learning Rate
PLS	Phoma Leaf Spots
P-R	Precision-Recall
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristic
RoCoLe	Robust Coffee Leaves
RSM	Red Spider Mite
SGD	Stochastic Gradient Descent
SM	Sooty Molds
SWAT-DCNN	Stage-Wise Aggregated Triple-Deep Convolutional Neural Network
T-DCNN	Triple Deep Convolutional Neural Network
TN	True Negatives
TP	True Positives

1 Introduction

Globally, coffee production radiated a broad range of job and business opportunities that contributed to our socioeconomic development [1]. However, supplying coffee products in the market can become challenging as farmers struggle to cope with destructive plant diseases affecting their farmlands [2]. In addition, coffee leaf disease management and diagnosis tend to become tedious for most farmers living in developing countries due to their lack of specialized equipment and expertise [3].

Recently, new methods emerged to alleviate these problems with computer vision and Deep Learning (DL). Since the rise of Convolutional Neural Networks (CNN), computers have become more intelligent as they can now recognize intricate image patterns and produce human-like decisions

[4]. CNN is a trainable multilayered architecture composed of subsequent operations that extract a high to low-level hierarchy of features from a 2D array image using a $k \times k$ striding convolutional filter. The extracted features then enter a pooling operation that reduces the feature's values to prevent computational exhaustion while maintaining and increasing its depth during the entire learning process. These processed feature samples then line up in a single-layered vector that enters a succeeding Multilayered Perceptron (MLP). With MLP, the features receive a respective weight and bias to distinguish their importance apart. Then, a logistic nonlinear classifier takes in these values from each neuron of the MLP, representing a specific learned feature. Also, while a CNN trains from a specific data domain, each weight on the MLP neurons updates through a forward and backward propagation that allows the model to learn progressively from a domain of labeled inputs to generate predictions of future unseen samples [5].

One of the notable breakthroughs in CNNs began with AlexNet, which proposed additional depth to the CNN structure that improved its overall classification performance [6]. Eventually, AlexNet as a deep CNN (DCNN) became a success, further inspiring other models to achieve better performance and cost-efficiency. In other research fields, DCNNs have shown exponential improvements with less bias through aggregation. As a result, feature generation became robust without the additional expense of reconstruction or lengthy training. The study of Minetto et al. showcased these improvements with their work where they aggregated families of ResNet and DenseNet models applied in the classification of geospatial land areas. With their experiments, they found that their DCNNs aggregated into a single classification pipeline called "Hydra" outmatched most state-of-the-art methods [7]. Therefore, this study had the inspiration to employ such a robust method to predict various coffee leaf conditions and attain significant results. For further improvement, this study also proposes a stage-wise approach, a diagnostic process that reduces complexity and increases the likelihood of getting genuine classifications than a conventionally trained DCNN [8]. In addition, upon rigorous review of existing related works, none used or investigated this combined approach of having an aggregated DCNN to perform classifications of coffee leaf diseases in a stage-wise fashion. With that said, this study presents the Stage-Wise Aggregated Triple Deep Convolutional Neural Networks (SWAT-DCNN).

Below presents the significant contributions of this study:

- Unlike most existing works, this study had a curated coffee leaf image dataset with various species and conditions that diversified the feature pool. In addition, the image samples curated had different perspectives, captured from either a controlled or uncontrolled environment, which most did

¹ College of Informatics and Computing Science, Batangas State University, Rizal Ave, Extension, Batangas, 4200 Batangas City, Philippines

- not consider. Therefore, giving the proposed model better opportunities to scale and not only learn from one aspect.
- With a higher possibility of bias predictions from a single trained model with limited data, this work aggregated three well-known DCNNs into a Triple-DCNN (T-DCNN). The three components of the T-DCNN came from a well-defined selection to guarantee cost-efficiency and performance, which further improved through transfer learning and fine-tuning. With that said, even with the aggregation method, T-DCNN's overall composition maintained a reasonable cost than most conventionally trained DCNNs based on the performance it yielded.
 - The proposed model also employed a stage-wise approach that lessened the prediction complexity and added robustness to its overall proficiency. Unlike a conventionally trained DCNN model that predicts all seven coffee leaf conditions in a single run, the proposed SWAT-DCNN does not require going over every class if it already satisfied the initial stages of classification, giving it extra leeway to reduce computational footprints when classifying massive test samples.

The following sections contain in-depth information about the proposed study. Section two discusses the literature review, section three tackles the materials and methods used to develop the proposed model, section four focuses on the evaluations and discussions of the experimental results, and the last section entails the conclusion.

2 Literature review

With the impact of DCNN models in the agricultural sector, this section discusses the previous studies and solutions conducted in various crops and coffee leaves.

2.1 Leaf disease from various crops

In a recent study, Amara et al. used a classic LeNet model that classified banana leaf diseases. Their method involved images of banana leaves captured in an uncontrolled outdoor setting categorized into three conditions that they resized into $60 \times 60 \times 1$ grayscale images to minimize the computational cost needed. Upon evaluation, they found that LeNet trained through random weight initialization could classify their three banana leaf conditions apart with an accuracy rate of 99.72% [9].

Another work by K. Zhang et al. employed a recent set of DCNN models that identified eight leaf diseases from tomatoes. Their work trained the AlexNet, GoogleNet, and ResNet models that performed feature extraction and predictions from their tomato leaf dataset captured in a controlled and uniformed fashion. However, during training, they found that

the given models consumed massive computing resources. With that in mind, they performed the method of transfer learning and fine-tuning the off-the-shelf models. They performed this process by taking each model's respective pre-trained weights from ImageNet and injecting them to the upper layers of each model accordingly. Also, to make their approach work, they replaced each model's current ending layers to fit their target number of classes. Through their evaluation, the results of their work achieved the highest classification accuracy of 96.51% from a pre-trained and fine-tuned ResNet model [10].

For another similar work, X. Zhang et al. improved the GoogleNet and Cifar10 models. Their task involved nine maize leaf conditions collected from the Plant Village dataset and Google web search that produced data diversity. Their proposed model aimed to increase the image recognition of such models where they added pooling operations, a Rectified Linear Unit (ReLU), and a dropout regularizer. With that said, their infused ReLU operations made their modified network learn sparse feature transformations apart from their dataset that generated other viewpoints and produced additional learnable feature sets compared to a conventional GoogleNet and Cifar10 model. Their dropout also controlled overfitting from the overwhelming features passing through their network, as dropout can remove random neurons in the network. As a result, their accuracy reached 98.9% with GoogleNet and 98.8% with Cifar10 [11].

2.2 Coffee leaf disease classification

With only a handful of papers published about coffee leaf diagnosis with DL, Esgario et al. proposed a study that classified leaf diseases of *Coffea arabica*. Their work involved a dataset with 1747 images divided into three classes, the Coffee Leaf Rust (CLR), Phoma Leaf Spots (PLS), *Cercospora* Leaf Spots (CLS), and Coffee Leaf Miner (CLM). With the shortage of collected images for their task, they performed data augmentation methods that created synthetic transformed images, which increased their feature pool. Their experiments found that one of their pre-trained models, the ResNet50 model, was the best option for classifying these diseases compared to AlexNet, GoogleNet, and VGG16 as it achieved the highest disease classification accuracy of 97.07% [12].

Kumar et al. also used the same dataset from Esgario et al. in their work but with a different state-of-the-art model, the InceptionV3. As a well-known practice, their work employed transfer learning, fine-tuning, and data augmentation that effectively increased their feature sets and improved their model's recognition ability toward the given dataset. As a result, they achieved 97.61% accuracy, 97.4% sensitivity, and 99.2% specificity. With such results, they concluded that a pre-trained DCNN, specifically InceptionV3, fine-tuned, and

Table 1 Summary of recent works that classified leaf diseases using deep convolutional neural networks

Method	Crop	Image environment	Classes	Accuracy (%)	Years
LeNet [9]	Banana	Outdoor/Field	3	99.72	2017
ResNet [10]	Tomato	Controlled	9	96.51	2018
Modified GoogleNet [11]	Maize	Both	9	98.90	2018
ResNet50 [12]	<i>Coffea arabica</i>	Controlled	4	97.07	2020
InceptionV3 [13]	<i>Coffea arabica</i>	Controlled	5	97.61	2020
VGG16 [14]	<i>Coffea liberica</i>	Controlled	3	97.20	2020

given sufficient data through data augmentation, could outperform most classical machine learning and conventional CNNs and even other DCNN models [13].

Due to the growing demand for state-of-the-art DCNNs, newer models came out. Montalbo and Hernandez's study trained recent DCNNs like Xception, ResNetV2, and the previous VGG16, which classified three *Coffea liberica* leaf conditions CLR, CLS, Sooty Molds (SM), and a healthy leaf. However, based on their observation, overfitting and underfitting cases occurred due to the lack of features learned by their singularly trained models. Nonetheless, their results achieved a remarkable accuracy of 97.20% with the VGG16 model and outperformed the other two recent DCNNs. Their results also indicated that even with a later DCNN model, other architectures like the VGG16 can still perform better with its simpler and more straightforward approach than a deeper and more sophisticated Xception and ResNetV2 [14].

Table 1 presents a summary of the discussed works. As shown, DCNNs can generate exceptional accuracies in identifying and classifying various leaf diseases from a wide variety of crops. However, most existing works in leaf disease classification primarily relied on a singularly trained DCNN model to classify either from an image captured in a controlled or uncontrolled (outdoor or field) setting. Due to those limitations, their trained models may have difficulty understanding both situations due to the lack of features learned.

3 Materials and methods

3.1 Coffee leaf dataset specification

Table 2 presents the diseases from the curated coffee leaf dataset used during the experiments, including a healthy coffee leaf (a). One of the well-known diseases, the CLR, emanates from a highly infectious fungus called the *Hemileia vastatrix* that produces rust-like pustules on the leaf, as shown in (b) [15, 16]. Another disease, the CLS (c), from a fungus *Cercospora coffeicola* [17] and PLS (d) from the *Phoma costaricensis* [18, 19], also show signs of dramatic change in the leaf's physical characteristics with brownish halo-like lesions [20]. Although most coffee variants today possess better resistance against these diseases, another problem of








insect infestation deprives the plant's nutrients, causing it to experience a similar demise [21]. Unlike diseases, the presence of leaf-sapping insects like the *Tetranychus urticae* or Red Spider Mites (RSM) and the *Leucoptera coffeina* or Coffee Leaf Miners (CLM) can leave behind injuries to the leaves after extraction, as shown in (e) and (f) [22, 23]. In addition, other insects like mealy bugs, scale, and aphids leave traces of SM, shown in (g). Though not infectious and as destructive, SM, if not attended immediately, can cover the entire surface of the leaf, preventing it from absorbing adequate sunlight [24]. These infections and infestations can limit the plant's capability to prosper and currently has no immediate solution or cure but are controllable through proper diagnosis, treatment, and management [25]. However, due to the difficulty of assessing these diseases and infestations, farmers who lack proper training and experience tend to suffer from a massive and untimely loss of yield [26, 27]. Moreover, even for an expert, identification, and classification of these leaf conditions can still become difficult due to the wide variety of pathogens and insect species [28]. With those said, improper diagnosis and treatment can occur, causing further injuries to the plant, adding more stress and vulnerabilities to other diseases [29].

Based on Table 2, the *Coffea canephora* or Robusta Coffee Leaf (RoCoLe) samples came from the published dataset of Parraga-Alava et al. [30], which included three classes, a healthy leaf, CLR, and RSM captured in an outdoor setting. Another dataset named the Brazilian Arabica Coffee Leaf (BrACoL) by Esgario et al. [12] had images of *Coffea arabica* from a controlled environment classified into a healthy leaf, CLR, CLS, PLS, and CLM. Lastly, a set of *Coffea liberica* or Liberica Coffee Leaves (LiCoLe) dataset served as an additional set of healthy leaves, CLR, and SM from Montalbo and Hernandez [14]. In total, the curated dataset in this study reached 4675 images classified into the seven discussed conditions.

3.1.1 Balancing of data with augmentation

Due to the limited samples available, this study employed data augmentation techniques that increased the sample size of each class with affine transformed images and gave the models additional learnable features. As presented in Table

Table 2 Specification of the curated coffee leaf dataset

Source	Samples						
	 Healthy (a)	 CLR (b)	 CLS (c)	 PLS (d)	 CLM (e)	 RSM (f)	 SM (g)
RoCoLe [30]	632	480	–	–	–	133	–
BrACoL [12]	272	337	88	342	321	–	–
LiCoLe [14]	645	515	–	–	–	–	910
Total	1549	1332	88	342	321	133	910

3, the values selected for augmentation produced new variations from the original images that did not affect their essential features [31]. However, it is worth mentioning that further increasing the given values can cause heavy distortions, making each image unrecognizable or indistinguishable. Therefore, this study made sure only to use subtle transformations that prevented such a problem from happening.

Moreover, due to the stage-wise nature of the proposed model, each stage had different data distributions. Fortunately, this approach balanced each model's dataset for each stage during training with the augmented filler images and prevented any class superiority that could have caused instability and bias [32]. In addition, this study guaranteed that the validation and test samples did not receive any augmentation and undergone a stochastic selection beforehand to prevent data leakage from the train samples, preventing unwanted pre-defined outcomes during experiments [33].

3.2 Triple deep convolutional neural networks

With such a complex task of classifying various coffee leaf diseases from various environmental conditions, this study proposed a stage-wise model based on DCNNs. However, using only a single model for feature learning and classification can result in a less robust and biased diagnosis. Therefore, this study employed an ensembled structure called the T-DCNN composed of carefully selected models from a preliminary benchmark analysis. The T-DCNN model with three different DCNNs aggregated as a single unit can conduct a diverse feature extraction of learnable patterns from a specific leaf condition due to its ensemble nature.

3.2.1 Model benchmark and selection

In constructing a compelling T-DCNN, this work had a preliminary benchmark performed that included the commonly

Table 3 Augmentation settings

Augmentation	Values
Horizontal flip	True
Vertical flip	True
Shear range	0.15
Rotation range	20
Zoom range	0.40
Fill mode	Constant

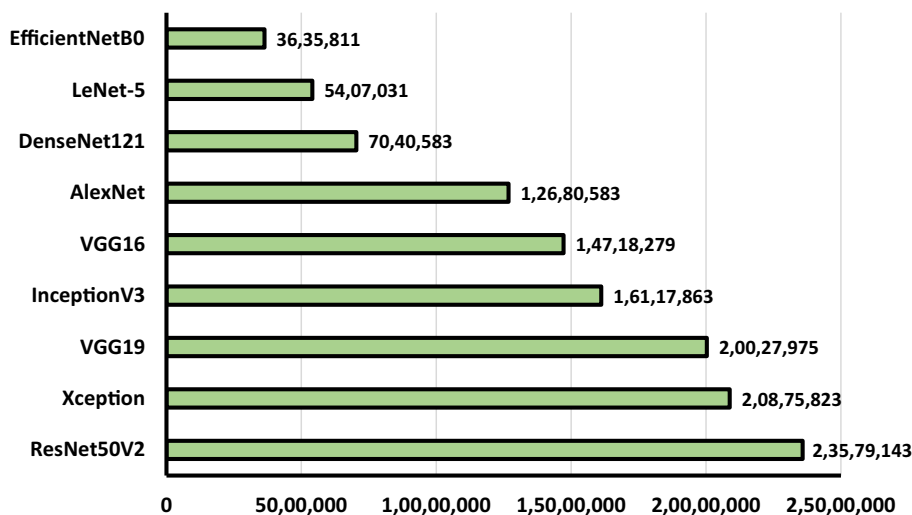
used and recent state-of-the-art DCNN classification models. The models chosen for the benchmark consisted of the AlexNet [6], VGG16/19 [34], InceptionV3 [35], EfficientNetB0 [36], DenseNet121 [37], Xception [38], ResNet50V2 [39], and LeNet-5 [40] where each trained using the curated coffee leaf dataset with seven classes. Subsequently, all models trained had their results analyzed and compared. It is also worth mentioning that these models had their ending layers replaced through fine-tuning to accommodate the said dataset. Without fine-tuning, the models would not have the capability to perform the task. The said fine-tuning process has an in-depth explanation in the later sections of the article.

Table 4 presents the results from the conducted benchmark. Based on calculations, the DenseNet121 and VGG16 had the highest validated accuracies among the rest with 94.21% and 93.46%, respectively, followed by the InceptionV3 model with 93.35%, making these three models the best possible candidates. However, considering models based only on their performance validated on a local dataset can take a toll on the reproducibility and scalability of the proposed T-DCNN. Therefore, as part of the selection process, this work also chose models based on their parameter sizes. DCNNs with fewer parameters entail better cost-efficiency, making them easier to reproduce and deploy in low to mid-end devices [41].

Table 4 State-of-the-art deep convolutional neural networks trained on the diverse coffee leaf dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
DenseNet121 [37]	94.21	93.95	94.21	94.02
VGG16 [34]	93.46	93.30	93.46	93.30
InceptionV3 [35]	93.35	93.43	93.35	92.57
Xception [38]	92.82	92.47	92.82	92.42
ResNet50V2 [39]	92.82	92.51	92.82	92.50
EfficientNetB0 [36]	92.07	91.42	92.07	91.39
VGG19 [34]	91.96	92.19	91.96	92.04
AlexNet [6]	81.35	80.06	81.35	80.08
LeNet-5 [40]	75.67	74.38	75.67	74.72

Entries in bold indicate the highest scores obtained

Fig. 1 Parameter comparison of the candidates

To better identify which should become part of the proposed solution, this work compared the following models in terms of their parameter size to performance ratio.

Illustrated in Fig. 1, the EfficientNetB0 model had the lowest parameters of 3.6M, followed by LeNet-5 with 5.4M and DenseNet121 with 7M. However, even though LeNet-5 had fewer parameters than most models, this study did not consider it for the task due to its inferior performance of 75.67% accuracy. The following potential candidate, the DenseNet121, had around 1.6M more parameters than LeNet-5 but a far better accuracy of 94.21%. Based on Table 4, comparing the DenseNet121 against the other models had shown its superiority, giving it no questions about why it should become part of the proposed solution. Although VGG16 required a larger parameter size of 14.7M than LeNet-5 with 5.4M and AlexNet with 12.6M, VGG16's performance to parameter size ratio still has a better balance, as VGG16 had a significant 93.46% accuracy, unlike LeNet-5 with only 75.67% and AlexNet with 81.35%.

Basing solely on Table 4, the possible candidates may eventually become the top three models with the highest accuracy. However, considering the selection criteria based

on performance and cost, even with InceptionV3's accuracy of 93.35%, its 16.1M parameters can bloat the T-DCNN to become computationally expensive. Its following model, the Xception model with 92.82%, had lesser performance yet higher parameters of 20.8M. The ResNet50V2 had a similar result as Xception but also higher parameters of 23.5M. The EfficientNetB0, on the other hand, had shown that even with its 92.07% accuracy, it only required 3.6M parameters, giving it only 1.28% less performance and having 13M fewer parameters than InceptionV3. Upon analysis of the following, it had shown that DenseNet121, VGG16, and EfficientNetB0 have the best potentials among the given selections to structure the proposed T-DCNN based not only on their performance but also on parameter sizes that may have a significant impact in the future when the dataset increases.

3.2.2 EfficientNetB0

With the aim for accurate classifications and cost-efficiency, this study selected the EfficientNetB0 illustrated in Fig. 2. The said model consists of a compressed architecture that offers better accuracy, scalability, and faster executability

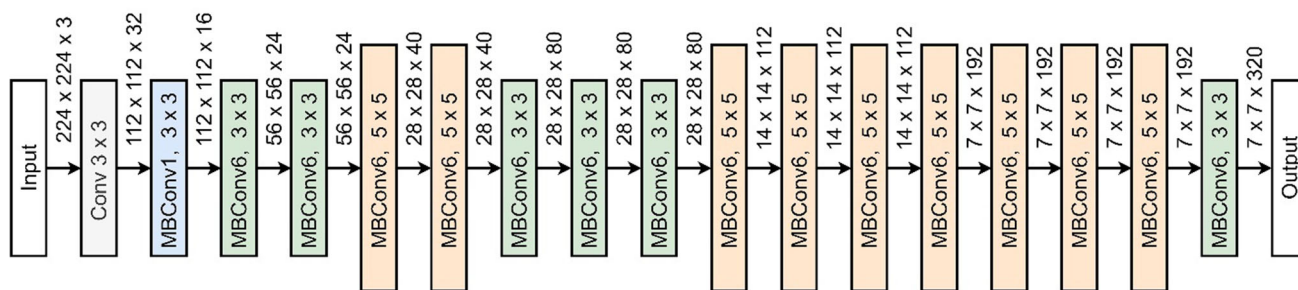


Fig. 2 EfficientNetB0 architecture [36]

than most state-of-the-art DCNN architectures. Its overall structure makes use of a 224×224 input dimension with sixteen succeeding Mobile-inverted Bottlenecks Convolutions (MBConv) with varying kernel sizes of 3×3 and 5×5 , each containing a Squeeze and Excitation (SE) block, Batch Normalization (BN), Depth-Wise Convolution (DWConv) and the recent Swish activation function. Furthermore, from the original benchmark made by EfficientNetB0 from the ImageNet and custom datasets, the model outperformed previous DCNNs in terms of image classifications with a smaller parameter size and computation cost (FLOPS) [36, 42]. With that said, EfficientNetB0 became a suitable choice for the feature generation and classification in this study.

3.2.3 DenseNet121

Compared to the standard CNN and ResNet models, DenseNet captures additional feature sets from its previous layers by concatenating every node directly to reuse features across the entire architecture, as shown in Fig. 3. The profound method yielded fewer parameters that made DenseNet easier to train without having severe performance saturation even with deeper layers. The model's primary concept consists of multiple dense blocks with a $\text{BN} \rightarrow \text{ReLU} \rightarrow 3 \times 3 \text{ Conv} \rightarrow \text{Dropout}$ connectivity pattern. As an entire model, before each dense block, a bottleneck transition performs down-sampling operations using a $\text{BN} \rightarrow \text{ReLU} \rightarrow 1 \times 1 \text{ Conv}$ followed by a 2×2 Average Pooling (AP). Through this approach of handling features, DenseNet became less computational heavy with better gradient handling and compensated the vanishing-gradient problem better than most DCNN models [37].

Due to these traits, DenseNet became a valuable feature extractor for the proposed T-DCNN with its efficient gradient handling and low-end computational requirement when producing learnable patterns from the limited coffee leaf dataset. This study primarily employed the 121-layer DenseNet with the smallest parameter size among the DenseNet family, which achieved the best results during the benchmark study.

3.2.4 VGG16

Unlike the selected EfficientNetB0 and DenseNet121, VGG16 had a much simpler feature extraction process that made it a go-to backbone model for most image classifications. Illustrated in Fig. 4, VGG16 uses a 3×3 kernel with a composition of succeeding Conv blocks containing two Conv layers activated by a ReLU function and down-sized by a following 2×2 max-pooling operation. In addition, VGG16 had an increased number of Conv layers from the third to the fifth Conv block with a similar pattern from the first and second. Unfortunately, due to its large neuron size of two 4096 FC neurons, the VGG16 became inflated, making it slow and costly to train [34]. Nonetheless, this study solved this problem through fine-tuning that reduced the network's FC neurons size yet maintained its extraction prowess for the task.

3.2.5 The structured T-DCNN classifiers

Figure 5 presents the proposed T-DCNN composed of the mentioned aggregated DCNN models. As illustrated, the selected DCNNs became aggregated as a single feature extraction unit connected directly to their corresponding averaging layer. Through this design, the T-DCNN managed to generate relevant predictions from their respective datasets, where these T-DCNNs performed specific classifications in a particular stage based on a broader spectrum of features. Thus, compared to a conventionally trained single model, due to these improvements made, the prediction probability by the T-DCNNs can become more dependable. Furthermore, such an aggregation method can reduce errors and bias that can alleviate issues regarding future input data [43].

3.3 Proposed stage-wise classification approach

This study's primary intuition is to have three distinct expert-level classification models that work together as a single unit to perform less biased classifications simultaneously from fewer options.

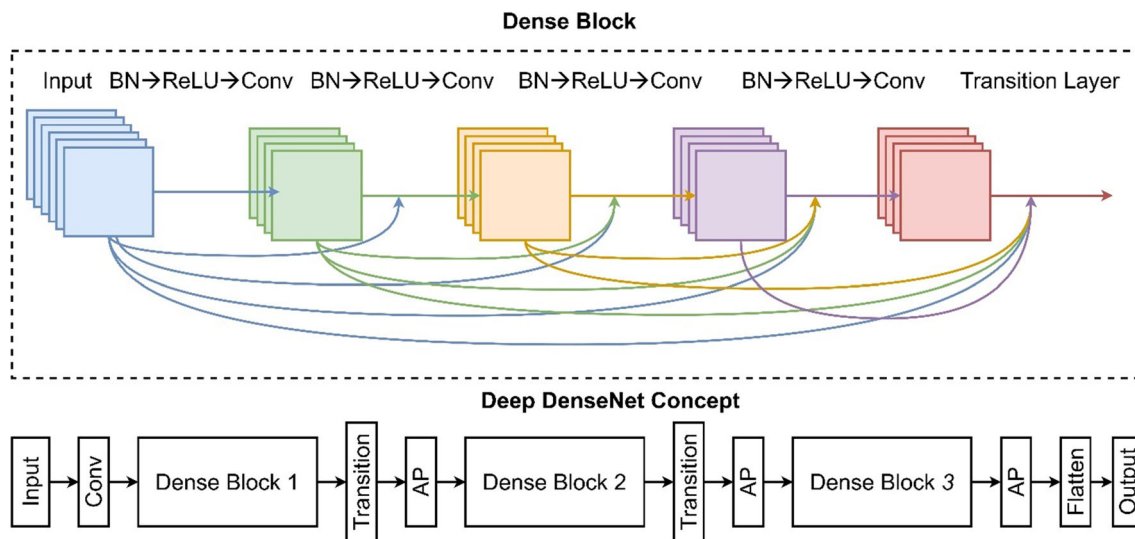


Fig. 3 DenseNet concept [37]

Fig. 4 VGG16 model [34]

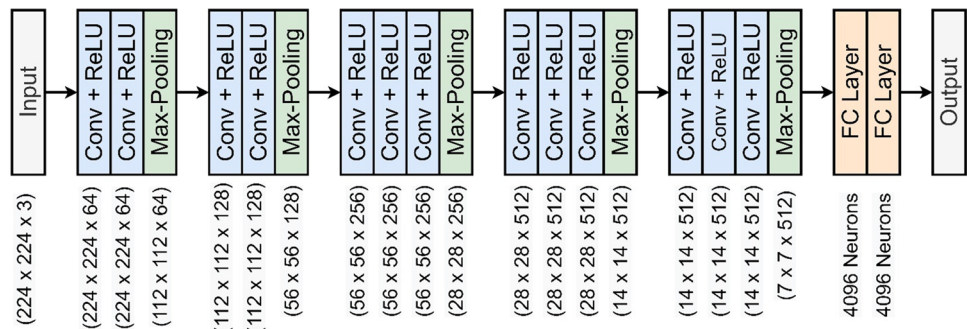


Figure 6 presents the proposed stage-wise design. The SWAT-DCNN begins with a coffee leaf health classifier supported by the T-DCNN stage 1 backbone identifying whether the leaf contains any infections, lesions, or molds. If none of the mentioned exists, the classifier immediately predicts and outputs that the coffee leaf is healthy and will no longer progress through the succeeding stages. Otherwise, if the model finds any of the mentioned anomalies, the model will pass the image to the second stage to identify whether the leaf has CLR, Brown Spot Lesions (BSL), or SM. Again, suppose the leaf had any features that resemble either CLR or SM, the model will eventually output its final prediction based only on the two options. In that case, the model will immediately set the entire model to a halt to prevent further consumption of resources. For the BSL case, this study did not consider including the CLS, PLS, CLM, and RSM together with CLR and SM at the second stage as it dramatically affected the overall performance due to their visual similarities, analyzed in the results section. Instead, this study added a third classification stage with another T-DCNN backbone that focuses only on the BSL, making the specific

identification of these lesions less confusing and robust for the entire SWAT-DCNN.

3.4 Transfer learning and fine-tuning the individual models

Before producing the T-DCNN, the three DCNNs underwent transfer learning and fine-tuning to adjust their functionality to classify coffee leaf conditions. With transfer learning, pre-trained weights from the ImageNet dataset transported readily available image recognition features that added leverage for the models to train faster and achieve better performance [44]. However, such a method also had the models inherit the pre-trained neurons of one thousand unnecessary classes, making them unsuitable for the task. Therefore, this study fine-tuned the said models that replaced their ending layers based on the classes per stage.

Due to the primary intent of DCNNs not being for coffee leaf classification, this study deducted unnecessary layers from each model accordingly. Fine-tuning helps reduce excessive parameters while preserving the most substantial number of features during the feature extraction process [45].

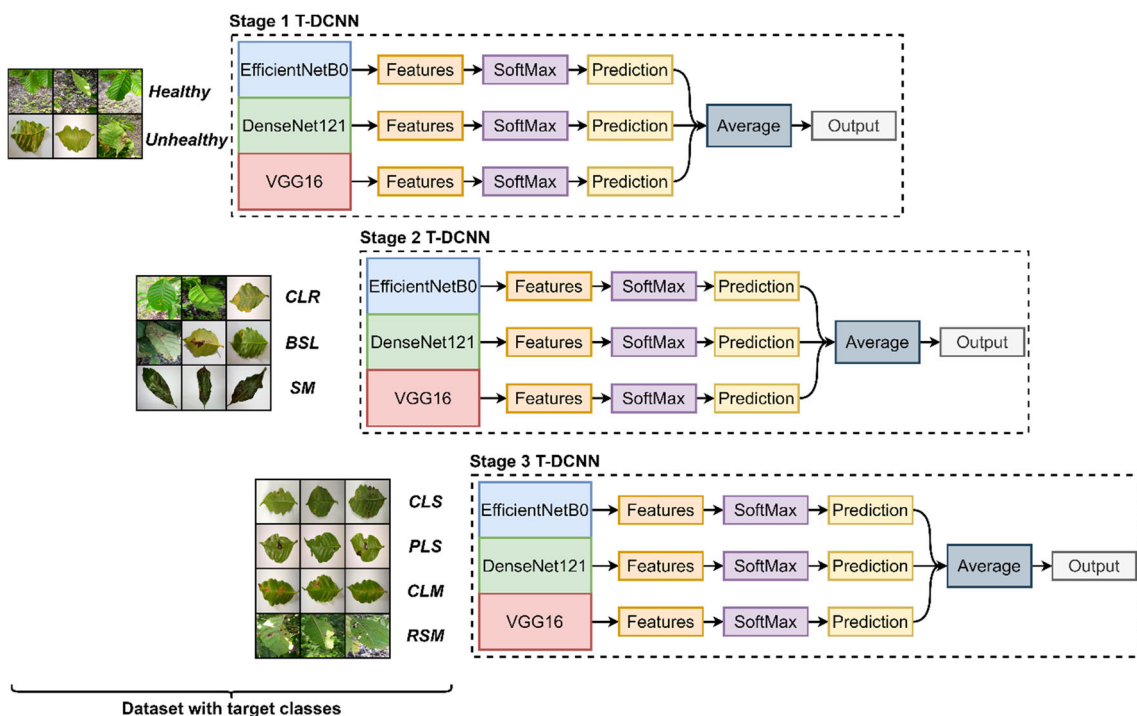
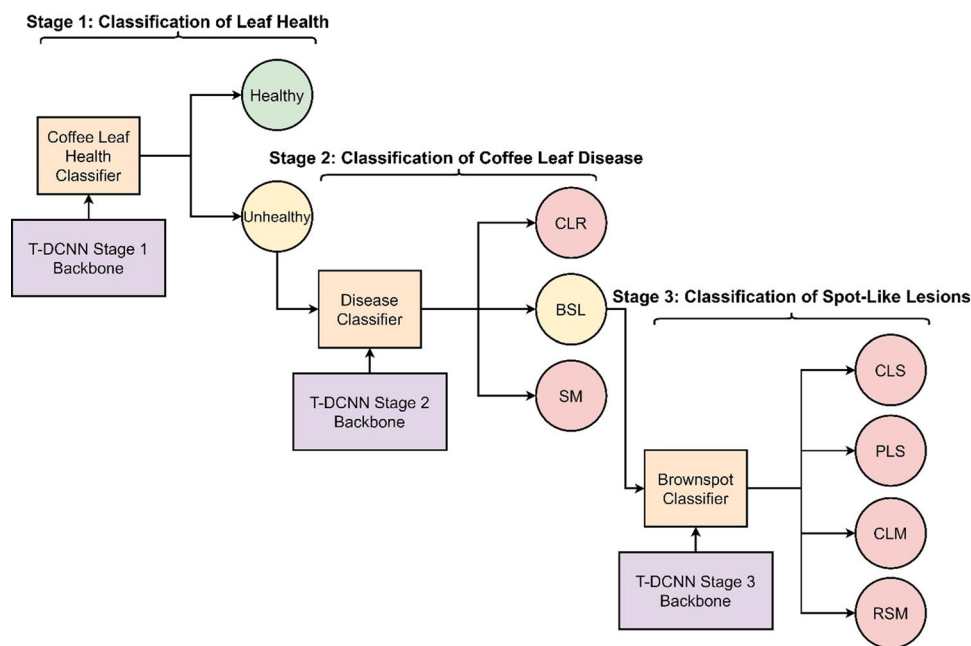


Fig. 5 Triple deep convolutional neural network backbone setup

Fig. 6 The stage-wise classification of coffee leaves with the trained backbones



The EfficientNetB0 had its last five layers removed in this study, making the “block7a_project_conv” with 320 depth features as its ending layer. Similarly, DenseNet121 had three layers deducted, ending with the “conv5_block16_concat” with more extensive features of 1024. Also, VGG16 only had two layers removed, significantly decreasing its parameter size compared to the other two models, leaving its ending layer with the “block5_conv3” with 512 features.

Subsequently, a set of proposed layers replaced the previously deducted layers so that the model could correctly classify the specific coffee leaf conditions. Instead of a typical FC dense layer, this study used the Global Average Pooling (GAP) that averaged the entire feature space and summed up the spatial feature information to produce a flattened vector passed to the following layer. The GAP layer also does not require complex optimization methods as it does not include

parameters compared to conventional FC dense layers, making it an ideal option to counteract overfitting [46].

This study also included a dropout layer for added assurance to provide better regularization and gradient flow from the previous layer. The dropout layer regularized the gradient flow by eliminating a random set of values from the flattened layer at a specific rate that relieved the network from potential instability during training due to the overwhelming flow of features [47]. For the rest of the network, the models had a respective number of dense neurons attached to a softmax classifier.

3.5 Hyper-parameter selection and model compilation

Due to the unfamiliarity of the models with the curated dataset, an appropriate selection of hyper-parameters is imperative to achieve the best possible results. Hyper-parameters are the “bells-and-whistles” of ML models that play a vital impact in their learning process. A well-tuned set of hyper-parameters can help the model achieve the lowest possible errors and potential highest performance toward a specific set of data [48].

Table 5 presents the empirically tuned hyper-parameter values based on current computing resources at the experiments’ time. The specification used for the experiments only had an 8GB GTX 1070 non-specialized GPU and an i5 fourth-generation Intel processor coupled with 16GB of RAM. The proposed values produced well-converged models as they prevented overfitting and underfitting issues along the way without the depletion of resources. The hyper-parameter values had intricate adjustments in the Learning Rate (LR) and epoch when such cases occur. A constant value of 16 provided sufficient transfer speed for the batch size without sacrificing too much memory. The selected epochs stayed at 25 to 30 as the models tend to provide less to no improvements beyond the given. For the optimizer, a go-to algorithm, Adam, an easily tune-able optimizer, provided a fast and reliable stochastic descent during weight training. Adam also became the choice for this study as it consumes less memory than a standard Stochastic Gradient Descent (SGD) [49] and RMSProp [50]. It is worth noting that the presented configurations yielded the most success in this study. However, such settings may still vary according to the present machine specifications if reproduced.

Furthermore, this study did not employ hyper-parameter optimization methods like random or grid search due to the mentioned limitation as it can become too costly, specifically for convoluted DCNNs [51]. Instead, all values came from an empirical trial and error estimation approach until an adequate convergence or result turned out. Nonetheless, though not considered entirely optimal, the results from the given settings still attained exceptional outcomes.

Table 5 The selected hyper-parameter configurations for each model

T-DCNN	Models	Softmax neurons	Epochs	LR
Stage 1	EfficientNetB0	2	25	$0.7e^{-5}$
	DenseNet121		25	$0.9e^{-6}$
	VGG16		30	$0.88e^{-7}$
Stage 2	EfficientNetB0	3	25	$0.3925e^{-5}$
	DenseNet121		25	$0.925e^{-6}$
	VGG16		25	$0.25e^{-6}$
Stage 3	EfficientNetB0	4	25	$0.224e^{-5}$
	DenseNet121		25	$0.925e^{-6}$
	VGG16		30	$0.175e^{-6}$

3.6 Loss function

Training an efficient DL model does not solely depend on high accuracies but also low error rates or losses. This study employed different loss functions to measure the number of errors produced during the training and validation processes. Due to the different class numbers in each T-DCNN stage, the use of a proper loss function like the Cross-Entropy (CE) loss measured each model’s loss appropriately. At the first stage, the models trained with a Binary CE loss (BCE_{loss}), which measured the losses between only two classes. However, the succeeding T-DCNN models had more than two classes, which indicated a multi-class classification. Therefore, instead of a BCE_{loss} , the following stages had a Categorical CE loss (CCE_{loss}). The following equations below denote the given loss functions. In BCE_{loss} , y is a binary indicator 0 or 1 based on a given class c from the observation made o and p as the prediction that justifies if the o belongs correctly to c . On the other hand, the CCE_{loss} represents M that signifies the multiple instances of classes for an appropriate loss measurement of a multi-class model [52].

$$BCE_{loss} = -(y \log(p) + (1 - y) \log(1 - p)) \quad (1)$$

$$CCE_{loss} = - \sum_{c=1}^M y_{o,c} \log(p_{o,c}). \quad (2)$$

3.7 Evaluation metrics

Like most DCNN models, this study employed the standard evaluation metrics used by most DL classification models. In addition, this study considered metrics like accuracy, precision, recall, and f1-score as the primary comparative measures. For the calculation of the following, this study relied on the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Each

Table 6 Distribution of the curated dataset into their respective splits

Split	Data	Healthy	CLR	CLS	PLS	CLM	RSM	SM	Total
Train (70%)	RoCoLe	443	336	–	–	–	93	–	872
	BrACoLe	191	236	62	240	225	–	–	954
	LiCoLe	452	361	–	–	–	–	637	1450
	Total	1086	933	62	240	225	93	637	3276
Validation (20%)	RoCoLe	126	96	–	–	–	27	–	249
	BrACoLe	54	67	18	68	64	–	–	271
	LiCoLe	128	103	–	–	–	–	182	413
	Total	308	266	18	68	64	27	182	933
Test (10%)	RoCoLe	63	48	–	–	–	13	–	124
	BrACoLe	27	34	8	34	32	–	–	135
	LiCoLe	65	51	–	–	–	–	91	207
	Total	155	133	8	34	32	13	91	466

of the given came from the classification instances performed by the model.

TP represents positive images classified correctly as the actual ground truth class, whereas the TN represents a correct classification of the non-positive or other classes. Either way, the vital aspect of these values in this study lies in the number of correct TPs and TNs produced by a specific model. For FP, this indicates that the model identified a negative class incorrectly with the wrong label, while FN does the same with a positive class. For the computation of the overall performance, this study considered the following equations below [53].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (3)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5)$$

$$\text{F1 - Score} = \frac{2 * \text{precision} * \text{recall}}{\text{Precision} + \text{recall}} \quad (6)$$

4 Experimental results and discussion

It is worth mentioning that this section focused on the previously identified evaluation metrics and other measurement approaches commonly used in vision-based DL. In addition, this study also presents other points of view to fully justify the significance of the SWAT-DCNN compared to a single-staged T-DCNN model and other state-of-the-art DCNNs. Through these approaches, the developed SWAT-DCNN effectively presented its contribution in terms of coffee leaf diagnosis.

4.1 Experimental setup and data handling

During experiments, as mentioned, this study used a GTX 1070 8GB graphics card to train the SWAT-DCNN and other DCNNs using the specified dataset in Table 2. The said dataset consisted of 4675 samples collectively. However, to train the models, this study needed to divide the dataset into 70% train, 20% validation, and 10% test set, as shown in Table 6. The division occurred for each class rather than the entire dataset of 4675 due to the imbalanced and limited quantities. Through this approach, the healthy leaf samples of 1549 became divided into 1086 train samples, 308 validation samples, and 155 test samples. Due to the uneven numbers after the split, the partial numbers went into the training samples as this work prioritized more on the learning process of the models. For the rest of the classes, this study also performed the same procedure of distribution.

With the concept of having multiple stages to perform the given tasks, the prepared dataset in Table 6 had its distribution designed for the stage-wise approach. However, upon distribution, the dataset had shown imbalances of train samples in all classes. Therefore, in Table 7, this study employed data augmentation to appropriately re-distribute and balance the train data for each stage.

For the first stage dataset, the health classifier had two classes, healthy and unhealthy leaves. The healthy class contained all the healthy leaf samples from the entire dataset, while the unhealthy class contained all the other classes. On the other hand, the following stage 2 dataset had CLR, BSL, and SM classes. The intuition behind the consolidated BSL is to reduce confusion among the highly similar characteristics of CLS, PLS, CLM, and RSM classes. However, the model still needed to classify each BSL specifically. Therefore, this study also established another dataset distribution

Table 7 Stage-wise distribution of the augmented dataset for a respective classification stage

Stage	Class	Train + Augmentation	Validation	Test
1	Healthy	1086 + 1914 = 3000	308	155
	Unhealthy	2190 + 810 = 3000	625	311
	Total	3276 + 2724 = 6000	933	466
2	CLR	933 + 1067 = 2000	266	133
	BSL	620 + 1380 = 2000	177	87
	SM	637 + 1363 = 2000	182	91
	Total	2190 + 3810 = 6000	625	311
3	CLS	62 + 1438 = 1500	18	8
	PLS	240 + 1260 = 1500	68	34
	CLM	225 + 1275 = 1500	64	32
	RSM	93 + 1407 = 1500	27	13
	Total	620 + 5380 = 6000	177	87

for the third stage of classification that focused only on the specific BSL classes.

4.2 Progress of training and validation

In DL, during the progress of training and validation, it is crucial to prevent overfitting and underfitting as it can impact the overall classification prowess of a model toward future unseen data. Based on the learning curves, this study monitored the changes of accuracy and losses over time [54]. In Fig. 7, all models successfully trained and validated from their respective datasets, illustrated by the converged train and validation graphs. In addition, even without an optimal hyper-parameter tuning approach, the selected values worked well with the prepared dataset combined with the proposed pre-training and fine-tuning methods. Though not all achieved full convergence, the results showed that all models had learned progressively in a stable manner within a brief period and avoided immense overfitting or underfitting.

4.3 Overall performance of the individual models using the validation dataset

Table 8 presents the classification results of each model trained on a specific stage using their respective validation data. For the first stage, the models trained to perform classifications between a healthy and an unhealthy leaf. On the other hand, the second stage focused on the three conditions, the CLR, BSL, and SM. Finally, the third stage had the CLS, PLS, CLM, and RSM. As observed, all models trained and validated well with their respective validation datasets. However, a slight decrease in accuracy occurred with EfficientNetB0 and DenseNet121 upon additional classes at the succeeding stages. Compared to the two, VGG16's accuracy slightly increased at the second stage of classifications but eventually

decreased again at the last stage. Nonetheless, even with the models' shifting performance, all still performed as a single unit in the form of a T-DCNN.

4.4 T-DCNN classification results from the test set with a confusion matrix

Figure 8 presents the classification results of the individual T-DCNN stages with their respective test datasets visualized using a normalized confusion matrix [55]. Also, an added Base T-DCNN model performed a similar task to highlight certain deficiencies when the stage-wise approach is not employed. The term "Base" indicates a T-DCNN model trained with all seven classes and did not perform a stage-wise approach. As evaluated, the Base T-DCNN had a slight classification advantage in classifying healthy leaves with 99.35% and CLS with 87.50% earning a 1.93% and 12.5% higher recall than the first and third T-DCNN stages, respectively. However, such a result does not immediately indicate that the Base T-DCNN entirely outperformed the T-DCNNs trained in stages. In a holistic view, the stage-wise T-DCNNs still had a significantly better classification among the other classes than the Base T-DCNN and that the Base T-DCNN also had the worst performance with RSM.

4.5 Receiver operating characteristic and area under the curve

In this section, the Receiver Operating Characteristic (ROC) curve estimated each specific model's classification ability in various thresholds [56]. As defined, ROC curves can generate a respective Area Under the Curve (AUC) by summarizing the trade-off between the sensitivity and specificity for each class to identify whether the model genuinely distinguished a specific class. A higher AUC for a specific class indicates better performance, and a lower one means poor. Having an $AUC \leq 0.50$ also entails that the model merely depends on guessing instead of actual classifications.

As depicted in Fig. 9, compared to the first stage, stages 2, 3, and the base T-DCNN had more evident fluctuations at the lower thresholds of their AUCs due to additional classes and complexity involved. With that said, the Base T-DCNN with the most considerable number of classes had the most noticeable noise than the rest of the T-DCNNs, specifically with the RSM class, indicating that the stage-wise approach does have an impact in terms of performance.

4.6 Precision–recall curve

Due to the unbalanced test samples used for each class, the Precision–recall (P–R) curve [57] became a more valuable evaluation tool that identified the FP and FN rate of each T-DCNN in different thresholds. Similarly, a P–R curve with a

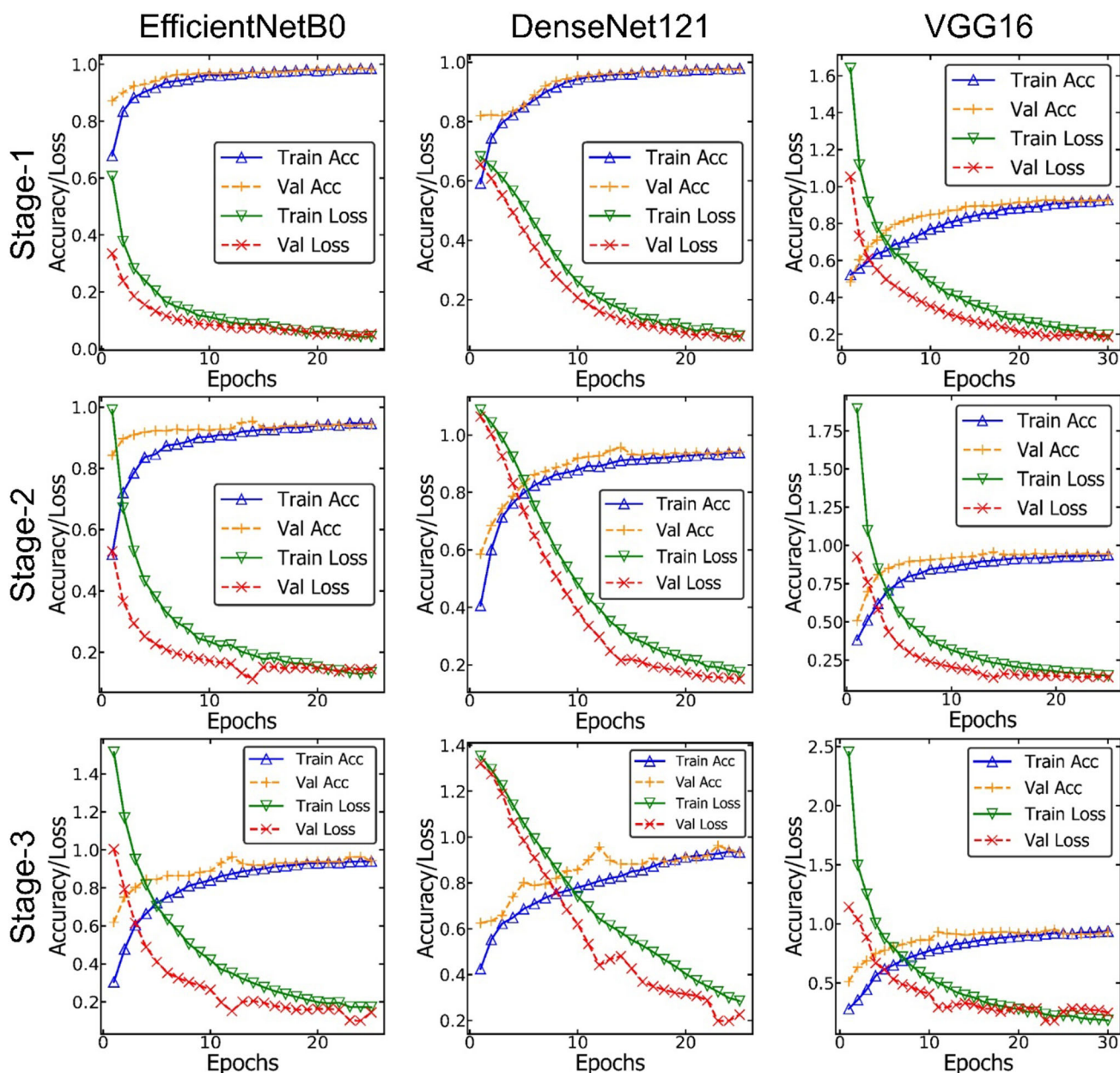


Fig. 7 The learning progress of the selected models during training and validation

higher AUC value indicates better performance in producing relevant results like the ROC curve.

As illustrated in Fig. 10, the Base T-DCNN also had the most noticeable distortion with RSM. Even with the slight AUC movements observed from the other stage-wise models, the lowest AUC attained was only 0.983 AUC from BSL in stage 2. The unstable distortion seen on the AUC of the Base T-DCNN from the RSM class landed as the lowest recorded AUC of 0.664. Unlike the rest of the T-DCNNs with fewer classes, the Base T-DCNN struggled the most in producing relevant results due to its higher number of classes. With results evaluated by the P-R and ROC curves,

the Base T-DCNN that classified all seven classes simultaneously became less dependable due to its complexity in classifying coffee leaf conditions than the proposed stage-wise approach.

4.7 Comparison of performance with a stage-wise approach

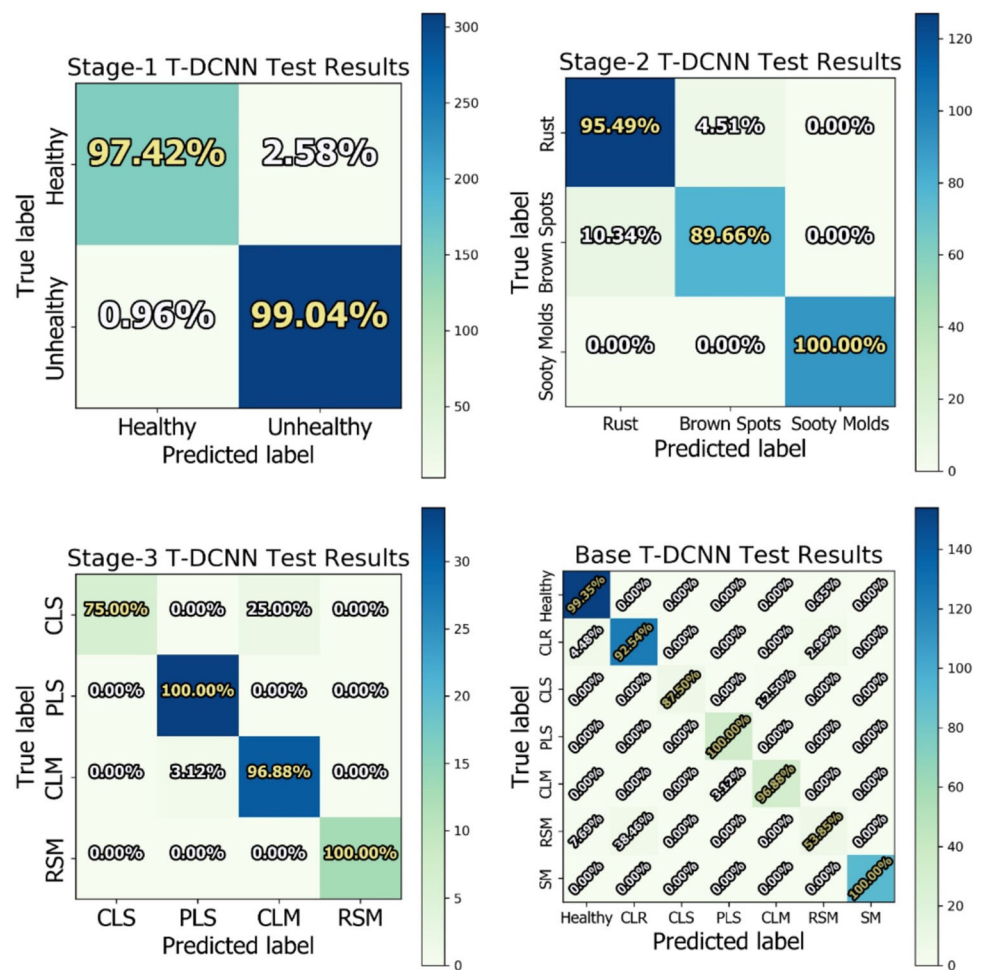
Figure 11 presents a visual comparison of the averaged overall performance of the SWAT-DCNN classification pipeline against the Base T-DCNN model. As expected from the results presented by the confusion matrices, ROC, and P-R

Table 8 Individual validation results of each model per stage

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Classes
<i>Stage 1 T-DCNN</i>					
EfficientNetB0	98.18	98.21	98.18	98.18	Healthy and Unhealthy
DenseNet121	97.32	97.37	97.32	97.33	
VGG16	92.71	92.91	92.71	92.77	
<i>Stage 2 T-DCNN</i>					
EfficientNetB0	95.20	95.21	95.20	95.18	CLR, BSL, SM
DenseNet121	94.72	94.79	94.72	94.68	
VGG16	94.40	94.42	94.40	94.37	
<i>Stage 3 T-DCNN</i>					
EfficientNetB0	94.35	94.75	94.35	94.27	CLS, PLS, CLM, RSM
DenseNet121	93.22	93.33	93.22	91.52	
VGG16	92.09	92.11	92.09	91.93	

The bolded entries indicate the highest scores obtained in a specific T-DCNN stage

Fig. 8 T-DCNN confusion matrix results from the test data for each stage



curves, the SWAT-DCNN outperformed the Base T-DCNN. As illustrated, the SWAT-DCNN had an evident increase of 0.5% accuracy, 0.41% precision, 0.05% recall, and a 0.5% f1-score than the Base T-DCNN, showing the significance of having a stage-wise model.

4.8 Gradient-weighted activation maps

For added transparency and further evaluation of this study, the Gradient-Weighted Activation Map (Grad-CAM) algorithm by Selvaraju et al. helped visualize how the SWAT-

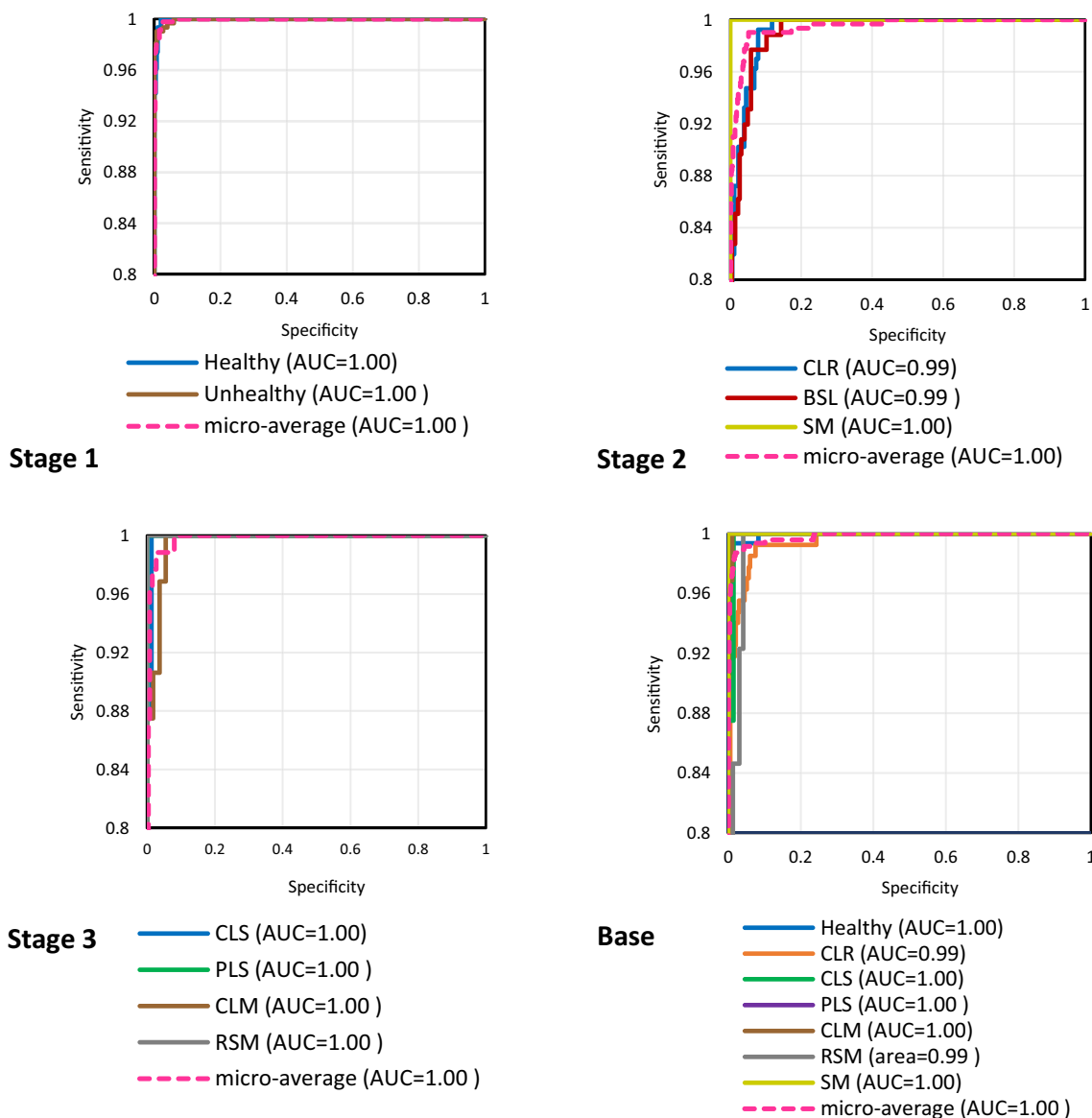


Fig. 9 Sensitivity versus specificity of the T-DCNN stages and Base T-DCNN with the test dataset

DCNN model interpreted different coffee leaf samples. With this algorithm, the SWAT-DCNN generated various saliency maps from the identified feature importance’s learned. This method also entails how the model provided its visual interpretation toward a specific class of interest without the need for model reconstruction or re-training. Furthermore, the Grad-CAM algorithm can also work with most convolutional vision-based models as it only relies on feature values generated from the last Conv layer of a model just before the FC layers [58].

Figure 12 illustrates that the SWAT-DCNN had successful interpretations from the presented randomly selected samples of each class. Though not perfect, the model still isolated the most salient features specifically for the CLR, CLS, PLS,

CLM, and RSM. However, due to the Grad-CAM algorithm’s limitation with multiple instances of targets, SM did not achieve exact isolations of its affected areas due to its sparsely distributed characteristic compared to the rest. Similarly, due to the absence of any salient lesion on a healthy leaf, the SWAT-DCNN seemed only to detect the entire leaf. Nonetheless, even with inexact expectations from the given interpretations, the proposed SWAT-DCNN still performed remarkably in both images captured from a controlled and outdoor environment. Therefore, this study validated that the model did not interpret the given samples randomly, as supported by the Grad-CAM results.

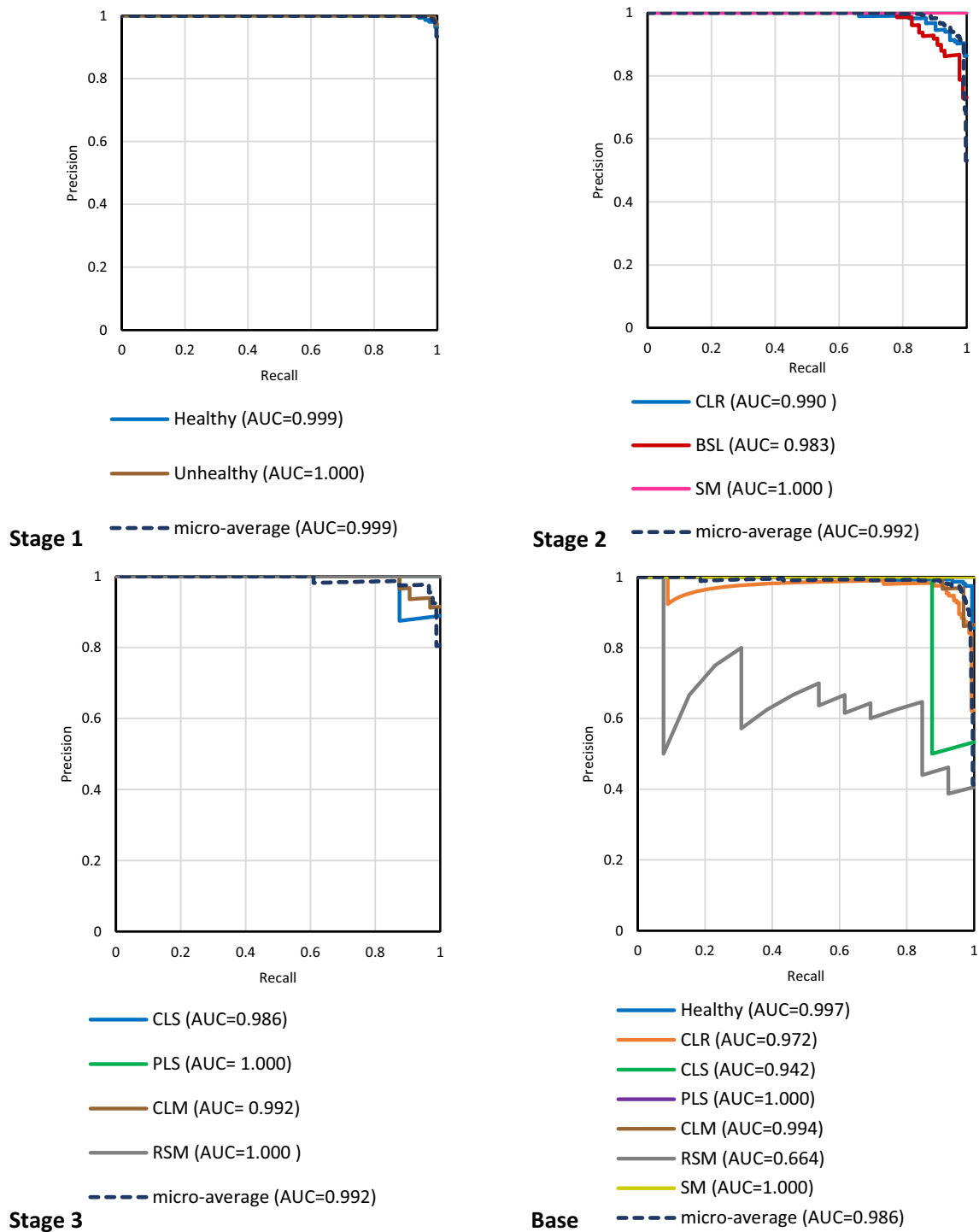


Fig. 10 Precision versus recall of the T-DCNN stages and the Base T-DCNN with the test dataset

4.9 Discussion

Based on the results shown in the confusion matrices, ROC, P-R curves, and other performance metrics, this study proved that the aggregating DCNN models could significantly increase the overall classification of coffee leaf diseases due

to the larger feature spectrum, added with the averaging of prediction results from multiple expert classifiers. Even with the significant improvements already produced through model aggregation, the proposed stage-wise approach further reduced the classification complexity and misclassifications of the Base T-DCNN that yielded additional improvements in

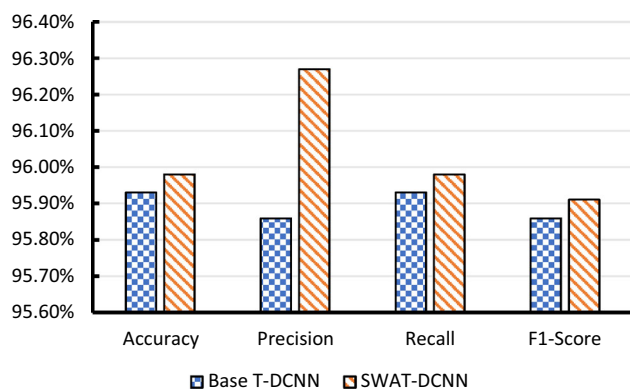


Fig. 11 T-DCNN overall performance compared with the base classifier

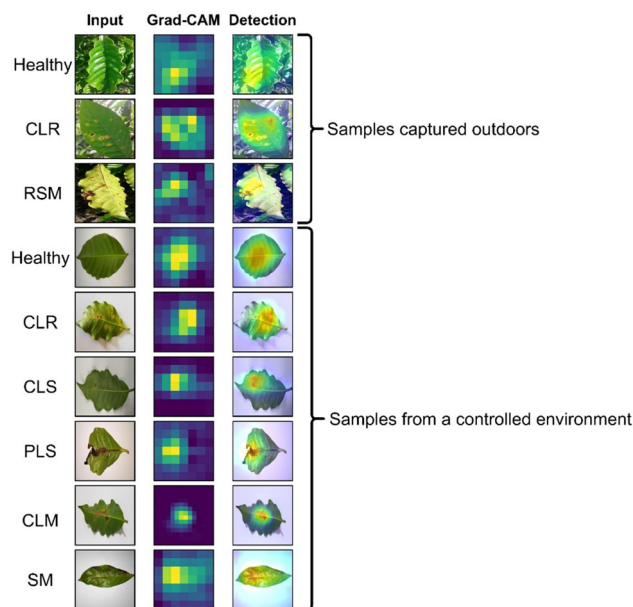


Fig. 12 Generated gradient-weighted class activations by the SWAT-DCNN from test samples

the form of the SWAT-DCNN. This study also demonstrated that the SWAT-DCNN did not perform random interpretations when it diagnosed the various coffee leaf conditions. With the Grad-CAM algorithm, the proposed study had shown excellent isolation of the salient affected areas. With those said, the proposed SWAT-DCNN contributes significantly to solving the challenging task of classifying a diverse set of coffee leaf conditions captured from various environments.

In Table 9, for an overall comparison, this study also compared the SWAT-DCNN's performance against existing state-of-the-art and classic DCNNs that performed the task similarly. Upon evaluation, SWAT-DCNN achieved the best results across all metrics with an overall 95.98% accuracy, followed by the Base T-DCNN (non-staged version) with

95.93%, and even outperformed a wide range of conventionally trained DCNNs.

This study also trained other models and compared their performance against each T-DCNN stage to generate broader findings. Presented in Table 10, the classification results produced from the first stage with two classes, healthy and unhealthy had shown that the T-DCNN dominantly performed across all metrics with an overall accuracy of 98.39%, followed by EfficientNetB0 with 98.18%.

In Table 11, stage 2, the following results had CLR, BSL, and SM. Upon evaluation, the T-DCNN again achieved the highest accuracy of 95.20%, with EfficientNetB0 and InceptionV3 being similar.

Lastly, in Table 12, the third and final stage had CLS, PLS, CLM, and RSM. Surprisingly, the T-DCNN did not attain the best performance. Instead, the VGG19 had the best performance with a 96.05% accuracy, making it 1.7% better.

Even if the T-DCNN model did not achieve the best performance at the last stage of classifications, it still had the best overall performance in the form of the SWAT-DCNN. Compared to others, the SWAT-DCNN had the highest overall accuracy of 95.98%, as shown in Table 13. Thus, unlike other DCNNs trained in a stage-wise or even through the conventional approach, the SWAT-DCNN still prevailed as the best overall performing model.

The calculated results had justified that model aggregation of diverse DCNN models like EfficientNetB0, DenseNet121, and VGG16 trained in a stage-wise fashion can yield valuable and better performance in diagnosing various coffee leaf conditions compared to single-stage models or uniformed staged models.

For transparency and future recommendation, this work also highlights the drawback of the proposed model. From the presented results, the developed model only functioned as an image classification model rather than an object detection [59] or segmentation model [60], making it unable to point out the exact localized sections of the leaf's affected areas. Nonetheless, future works can employ the proposed approach for a detection or segmentation model that may yield massive improvements for coffee leaf diagnosis.

5 Conclusion

This study proposed an automated approach to diagnose coffee leaf conditions with DL and computer vision due to the challenging task of coping with various coffee leaf diseases and infestations. With the growing demand for DL solutions, this study contributed a novel approach that classified seven coffee leaf conditions with three aggregated profound DCNN models selected through a benchmark study

Table 9 Comparison of performance with other state-of-the-art models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SWAT-DCNN	95.98	96.27	95.98	95.91
Base T-DCNN	95.93	95.86	95.93	95.86
Base DenseNet121 [37]	94.21	93.95	94.21	94.02
Base VGG16 [34]	93.46	93.30	93.46	93.30
Base InceptionV3 [35]	93.35	93.43	93.35	92.57
Base Xception [38]	92.82	92.47	92.82	92.42
Base ResNet50V2 [39]	92.82	92.51	92.82	92.50
Base EfficientNetB0 [36]	92.07	91.42	92.07	91.39
Base VGG19 [34]	91.96	92.19	91.96	92.04
Base AlexNet [6]	81.35	80.06	81.35	80.08
Base LeNet-5 [40]	75.67	74.38	75.67	74.72

*Entries in bold indicate the highest attained value for a specific criterion. Base models indicate that it classified all seven classes simultaneously or did not train in a stage-wise manner

Table 10 Stage 1 classification results

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
T-DCNN	98.39	98.41	98.39	98.40
EfficientNetB0 [36]	98.18	98.21	98.18	98.18
ResNet50V2 [39]	97.96	98.01	97.96	97.97
InceptionV3 [35]	97.96	98.03	97.96	97.97
Xception [38]	97.86	97.88	97.86	97.86
DenseNet121 [37]	97.32	97.37	97.32	97.33
VGG19 [34]	96.78	96.78	96.78	96.77
VGG16 [34]	92.71	92.91	92.71	92.77
AlexNet [6]	92.07	92.20	92.07	92.11
LeNet-5 [40]	86.50	86.78	86.50	86.60

The bolded entries indicate the highest scores obtained during the Stage 1 classification

*The following models presented only trained with two classes, a healthy and unhealthy leaf

Table 11 Stage 2 classification results

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
T-DCNN	95.20	95.30	95.20	95.16
EfficientNetB0 [36]	95.20	95.21	95.20	95.18
InceptionV3 [35]	95.20	95.21	95.20	95.18
VGG19 [34]	95.04	95.04	95.04	95.04
DenseNet121 [37]	94.72	94.79	94.72	94.68
VGG16 [34]	94.40	94.42	94.40	94.37
ResNet50V2 [39]	94.40	94.42	94.40	94.37
Xception [38]	94.40	94.40	94.40	94.40
AlexNet [6]	89.44	89.68	89.44	89.52
LeNet-5 [40]	87.04	87.08	87.04	87.03

The bolded scores indicate the highest values attained during the Stage 2 classifications

*The following models had trained with three classes, rust, brown spots, and sooty molds

and formed in a stage-wise fashion. It is worth mentioning that this study also used a diverse set of coffee leaves captured in various conditions, which most existing works did not consider. The intuition of this study primarily lies in the concept of narrowing down the classification complexity by simplifying a broad classification task into stages with fewer

options and increasing the number of features in a neural network classifier. Based on the discussed approach, this study aggregated three state-of-the-art pre-trained and fine-tuned DCNNs that included the EfficientNetB0, DenseNet121, and VGG16 formed into an ensemble called the T-DCNN. Once aggregated, the T-DCNN produced a broader set of features,

Table 12 Stage 3 classification results

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
VGG19 [34]	96.05	96.09	96.05	96.02
ResNet50V2 [39]	94.92	94.96	92.56	94.87
T-DCNN	94.35	95.11	94.35	94.18
EfficientNetB0 [36]	94.35	94.75	94.35	94.27
Xception [38]	94.35	94.31	94.35	94.16
InceptionV3 [35]	93.79	93.92	93.79	93.70
DenseNet121 [37]	93.22	93.33	93.22	93.12
VGG16 [34]	92.09	92.11	92.09	91.93
LeNet-5 [40]	89.27	88.83	89.27	88.73
AlexNet [6]	86.44	85.94	86.44	86.05

The scores in bold means that they are the highest scores obtained in Stage 3

*The following models had trained with four classes, CLS, PLS, CLM, and RSM

Table 13 Overall classification performance in a stage-wise manner

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SWAT-DCNN	95.98	96.27	95.98	95.91
SW-VGG19 [34]	95.96	95.97	95.96	95.94
SW-EfficientNetB0 [36]	95.91	96.06	95.91	95.88
SW-ResNet50V2 [39]	95.76	95.80	94.97	95.74
SW-InceptionV3 [35]	95.65	95.72	95.65	95.62
SW-Xception [38]	95.54	95.53	95.54	95.47
SW-DenseNet121 [37]	95.09	95.16	95.09	95.04
SW-VGG16 [34]	93.07	93.15	93.07	93.02
SW-AlexNet [6]	89.32	89.27	89.32	89.23
SW-LeNet-5 [40]	87.60	87.56	87.60	87.45

The bolded values indicate the highest scores obtained when the models are in stage-wise form

*The following models with SW had their performance averaged in a stage-wise approach. The T-DCNN model in the form of a stage-wise manner became the SWAT-DCNN

where it also had an averaging layer attached at the end to achieve less biased classification toward a specific target. Upon evaluation, this study proved that the Base T-DCNN could yield better results than conventionally trained DCNNs when it classified seven coffee leaf conditions simultaneously. However, having only the Base T-DCNN classify all the said conditions led to an abundant case of false classifications. As identified on its P-R curves, the Base T-DCNN generated the lowest 0.664 AUC for the RSM. Fortunately, the SWAT-DCNN managed to alleviate the said problem, as it attained significant improvements that increased the RSM's AUC to 1.00. Overall, the stage-wise approach had shown an increase of 0.5% accuracy, 0.41% precision, 0.05% recall, and a 0.05% F1-score than the Base T-DCNN. In addition, this study also showed that the SWAT-DCNN can outperform commonly used state-of-the-art and classic DCNN models, justifying its classification prowess with a 95.98% accuracy.

In conclusion, the overall performance achieved by the SWAT-DCNN implies that model aggregation and a stage-wise approach can induce significant improvements in the classification of diverse coffee leaf conditions. In addition, such improvements can become vital in developing apps for

real-world scenarios. Based on the SWAT-DCNN's accurate performance with images captured in a laboratory setting and images captured from an uncontrolled environment, farmers and experts can potentially gain better opportunities to perform real-world diagnoses of the identified classes in their coffee farms easier and faster in the future. However, the proposed model cannot identify other coffee leaf conditions and cannot localize the affected areas effectively due to the limitations of a classification scheme. Therefore, as a future study, researchers can add other coffee leaf conditions that can help the model learn beyond the given classes, making it a more viable tool. Also, employing an object detection head or re-constructing the SWAT-DCNN into a segmentation architecture can provide additional capabilities that can help it identify coffee leaf diseases from a distinct perspective. Development and deployment of such in drones or robots can significantly improve coffee farming or even agriculture in general.

Acknowledgment The author thanks Batangas State University for supporting this study and the validation of its results. Without its support, this work would not have become possible.

Authors contributions The author contributed fully to accomplishing this study.

Funding This study did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data and code availability The author believes that research reproducibility can better impact other researchers and the likes that may require such a solution. Therefore, the author provides the SWAT-DCNN code and data sources through this link <https://github.com/francismontalbo/swatdcnn> for future reproduction and improvements.

Declarations

Conflict of interest The author declares no conflict of interest.

References

- Mutandwa, E., Kanuma, N.T., Rusatira, E., Kwiringirimana, T., Mugenzi, P., Govere, I., Foti, R.: Analysis of coffee export marketing in Rwanda: application of the Boston consulting group matrix. *Afr. J. Bus. Manage.* **3**(5), 210–219 (2009). <https://doi.org/10.5897/AJBM09.009>
- Badel, J.L., Zambolim, L.: Coffee bacterial diseases: a plethora of scientific opportunities. *Plant. Pathol.* **68**(3), 411–425 (2019). <https://doi.org/10.1111/ppa.12966>
- Millard, E.: Still brewing: fostering sustainable coffee production. *World Dev. Perspect.* **7**, 32–42 (2017). <https://doi.org/10.1016/j.wdp.2017.11.004>
- LeCun, Y., Kavukcuoglu, K., Farabet, C.: Convolutional networks and applications in vision. In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, Paris, pp. 253–256 (2010). <https://doi.org/10.1109/ISCAS.2010.5537907>
- LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Handwritten digit recognition with a back-propagation network. In: *Advances in Neural Information Processing Systems 2 (NIPS'89)*, pp. 396–404 (1990). <https://papers.nips.cc/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf>
- Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: *Proceedings of the Neural Information Processing Systems*, pp. 1097–1105 (2012). <https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>
- Minetto, R., Pamplona Segundo, M., Sarkar, S.: Hydra: an ensemble of convolutional neural networks for geospatial land classification. *IEEE Trans. Geosci. Remote Sens.* **57**(9), 6530–6541 (2019). <https://doi.org/10.1109/TGRS.2019.2906883>
- Esener, I., Ergin, S., Yuksel, T.: A new feature ensemble with a multistage classification scheme for breast cancer diagnosis. *J. Healthc. Eng.* **2017**(3895164), 1–15 (2017). <https://doi.org/10.1155/2017/3895164>
- Amara, J., Bouaziz, B., Algergawy, A.: A deep learning-based approach for banana leaf diseases classification. *BTW Workshops*, pp. 79–88 (2017). <https://dl.gi.de/handle/20.500.12116/944>
- Zhang, K., Wu, Q., Liu, A., Meng, X.: Can deep learning identify tomato leaf disease? *Adv. Multimedia* **2018**, 1–10 (2018). <https://doi.org/10.1155/2018/6710865>
- Zhang, X., Qiao, Y., Meng, F., Fan, C., Zhang, M.: Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access* **6**, 30370–30377 (2018). <https://doi.org/10.1109/ACCESS.2018.2844405>
- Esgario, J., Krohling, R., Ventura, J.: Deep learning for classification and severity estimation of coffee leaf biotic stress. *Comput. Electron. Agric.* **169**, 105162 (2020). <https://doi.org/10.1016/j.compag.2019.105162>
- Kumar, M., Gupta, P., Madhav, P., Sachin: Disease Detection in coffee plants using convolutional neural network. In: *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, COIMBATORE, India, pp. 755–760 (2020). <https://doi.org/10.1109/ICCES48766.2020.9138000>
- Montalbo, F.J.P., Hernandez, A.A.: Classifying Barako coffee leaf diseases using deep convolutional models. *Int. J. Adv. Intell. Inform.* **6**(2), 197 (2020). <https://doi.org/10.26555/ijain.v6i2.495>
- Zambolim, L.: 'Current status and management of coffee leaf rust in Brazil.' *Tropic. Plant Pathol.* **41**(1), 1–8 (2016). <https://doi.org/10.1007/s40858-016-0065-9>
- Talhinhas, P., et al.: The coffee leaf rust pathogen *Hemileia vastatrix*: one and a half centuries around the tropics. *Mol. Plant Pathol.* **18**(8), 1039–1051 (2017). <https://doi.org/10.1111/mpp.12512>
- Nelson, S.: *Cercospora Leaf Spot and Berry Blotch of Coffee*. University of Hawai'i at Manoa, College of Tropical Agriculture and Human Resources, Cooperative Extension Service, Honolulu (2008). <http://www.ctahr.hawaii.edu/oc/freepubs/pdf/PD-41.pdf>
- Silva Júnior, M., et al.: Foliar fertilizers for the management of phoma leaf spot on coffee seedlings. *J. Phytopathol.* **166**(10), 686–693 (2018). <https://doi.org/10.1111/jph.12745>
- Maghuly, F., Jankowicz-Cieslak, J., Bado, S.: Improving coffee species for pathogen resistance. *CAB Rev.* **15**(9), 1–18 (2020). <https://doi.org/10.1079/PAVSNR202015009>
- Silva, M., et al.: Coffee resistance to the main diseases: leaf rust and coffee berry disease. *Braz. J. Plant. Physiol.* **18**(1), 119–147 (2006). <https://doi.org/10.1590/s1677-04202006000100010>
- Sanders, M.: Breeding for coffee leaf rust resilience in *Coffea* sp. *Nat. Sci. Educ.* **48**(1), 190102 (2019). <https://doi.org/10.4195/nse2019.01.0102>
- Roy, S., Muraleedharan, N., Mukhopadhyay, A.: The red spider mite, *Oligonychus coffeae* (Acari: Tetranychidae): its status, biology, ecology and management in tea plantations. *Exp. Appl. Acarol.* **63**(4), 431–463 (2014). <https://doi.org/10.1007/s10493-014-9800-4>
- Androcioni, H., Hoshino, A., Menezes Júnior, A., Morais, H., Bianco, R., Caramori, P.: Coffee leaf miner incidence and its predation by wasp in coffee intercropped with rubber trees. *Coffee Sci.* **13**(3), 389–400 (2018). <https://doi.org/10.25186/cs.v13i3.1487>
- Nelson, S.: Sooty Mold. University of Hawaii, Honolulu (2008). <https://scholarspace.manoa.hawaii.edu/handle/10125/12424>
- Savary, S., Ficke, A., Aubertot, J.-N., Hollier, C.: Crop losses due to disease and their implications for global food production losses and food security. *Food Secur.* **4**(2), 519–537 (2012). <https://doi.org/10.1007/s12571-012-0200-5>
- Bentley, J., Thiele, G.: Bibliography: farmer knowledge and management of crop disease. *Agric. Hum. Values* **16**, 75–81 (1999). <https://doi.org/10.1023/a:1007558919244>
- Nelson, R., et al.: Working with resource-poor farmers to manage plant diseases. *Plant Dis.* **85**(7), 684–695 (2001). <https://doi.org/10.1094/pdis.2001.85.7.684>
- Ngugi, L., Abelwahab, M., Abo-Zahhad, M.: Recent advances in image processing techniques for automated leaf pest and disease recognition—a review. *Inf. Process. Agric.* (2020). <https://doi.org/10.1016/j.inpa.2020.04.004>
- Tarr, S.A.J.: Plant injury due to insects, mites, nematodes, and other pests. In: Tarr, S.A.J. (ed.) *Principles of Plant Pathology*, pp. 126–137. Springer, Berlin (1972). https://doi.org/10.1007/978-1-349-00355-6_9

30. Parraga-Alava, J., Cusme, K., Loor, A., Santander, E.: RoCoLe: a robusta coffee leaf images dataset for evaluation of machine learning based methods in plant diseases recognition. *Data Brief* **25**, 104414 (2019). <https://doi.org/10.1016/j.dib.2019.104414>
31. Mikołajczyk, A., Grochowski, M.: Data augmentation for improving deep learning in image classification problem. In: 2018 International Interdisciplinary Ph.D. Workshop (IIPHDW), Swinoujście, pp. 117–122 (2018). <https://doi.org/10.1109/IIPHDW.2018.8388338>
32. Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. *J. Big Data* **6**(1), 27 (2019). <https://doi.org/10.1186/s40537-019-0192-5>
33. Saravanan, N., Sathish, G., Balajee, J.M.: Data wrangling and data leakage in machine learning for healthcare. *Int. J. Emerg. Technol. Innov. Res.* **5**(8), 553–557 (2018)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint, [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
35. Szegedy, C., et al.: Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, pp. 1–9 (2015). <https://doi.org/10.1109/CVPR.2015.7298594>
36. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning, pp. 6105–6114 (2019). <http://proceedings.mlr.press/v97/tan19a.html>
37. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, pp. 2261–2269 (2017). <https://doi.org/10.1109/CVPR.2017.243>
38. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, pp. 1800–1807 (2017). <https://doi.org/10.1109/CVPR.2017.195>
39. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *Computer Vision—ECCV 2016*. Lecture Notes in Computer Science, vol. 9908, pp. 630–645. Springer, Amsterdam (2016). https://doi.org/10.1007/978-3-319-46493-0_38
40. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998). <https://doi.org/10.1109/5.726791>
41. Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., Hasan, M., Van Essen, B.C., Awwal, A.A.S., Asari, V.K.: A state-of-the-art survey on deep learning theory and architectures. *Electron* **8**(3), 292 (2019). <https://doi.org/10.3390/electronics8030292>
42. Chowdhury, N.K., Rahman, M., Rezoana, N., Kabir, M.A.: ECOV-Net: an ensemble of deep convolutional neural networks based on EfficientNet to detect COVID-19 from chest X-rays. arXiv preprint, [arXiv:2009.11850](https://arxiv.org/abs/2009.11850)
43. Hansen, L.K., Salamon, P.: Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(10), 993–1001 (1990). <https://doi.org/10.1109/34.58871>
44. Pan, S.J., Yang, Q.: A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **22**(10), 1345–1359 (2010). <https://doi.org/10.1109/TKDE.2009.191>
45. Too, E.C., Yujian, L., Njuki, S., Yingchun, L.: A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* **161**, 272–279 (2019). <https://doi.org/10.1016/j.compag.2018.03.032>
46. Lin, M., Chen, Q., Yan, S.: Network in network. In: Proceedings of the International Conference on Learning Representations (2014). <https://arxiv.org/abs/1312.4400>
47. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* (2014). <https://doi.org/10.5555/2627435.2670313>
48. Bengio, Y.: Practical recommendations for gradient-based training of deep architectures. In: Montavon, G., Orr, G.B., Muller, K.R. (eds.) *Neural Networks: Tricks of the Trade*. Lecture Notes in Computer Science, vol. 7700, 2nd edn., pp. 437–478. Springer, Berlin (2012). https://doi.org/10.1007/978-3-642-35289-8_26
49. Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. *SIAM Rev.* **60**(2), 223–311 (2018). <https://doi.org/10.1137/16M1080173>
50. Hinton, G., Srivastava, N., Swersky, K.: *Neural Networks for Machine Learning—Lecture 6a: Overview of Mini-Batch Gradient Descent*. University of Toronto, Toronto, ON (2012). https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
51. Feurer, M., Hutter, F.: Hyperparameter optimization. In: Hutter, F., Kotthoff, L., Vanschoren, J. (eds.) *Automated Machine Learning*. The Springer Series on Challenges in Machine Learning, pp. 3–33. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05318-5_1
52. ML Cheatsheet: Loss functions (2019). https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html. Accessed 2020 Nov 24
53. Mohammad, H., Sulaiman, M.N.: A review on evaluation metrics for data classification evaluations. *Int. J. Data Min. Knowl. Manage. Process* **5**(2), 01–11 (2015). <https://doi.org/10.5121/ijdkp.2015.5201>
54. Caruana, R., Lawrence, S., Giles, C.L.: Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping. In: Proceedings of the Advances in Neural Information Processing Systems, pp. 402–408 (2000). <https://doi.org/10.5555/3008751.3008807>
55. Jia, F., Lei, Y., Lu, N., Xing, S.: Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization. *Mech. Syst. Signal Process.* **110**, 349–367 (2018). <https://doi.org/10.1016/j.ymssp.2018.03.025>
56. Dhingra, G., Kumar, V., Joshi, H.D.: A novel computer vision based neutrosophic approach for leaf disease identification and classification. *Measurement* **135**, 782–794 (2019). <https://doi.org/10.1016/j.measurement.2018.12.027>
57. Fuentes, A.F., Yoon, S., Lee, J., Park, D.S.: High-performance deep neural network-based tomato plant diseases and pests diagnosis system with refinement filter bank. *Front. Plant Sci.* **9**, 1162 (2018). <https://doi.org/10.3389/fpls.2018.01162>
58. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV), Venice, pp. 618–626 (2017). <https://doi.org/10.1109/ICCV.2017.74>
59. Buzzy, M., Thesma, V., Davoodi, M., Mohammadpour Velni, J.: Real-time plant leaf counting using deep object detection networks. *Sensors* **20**(23), 6896 (2020). <https://doi.org/10.3390/s20236896>
60. Singh, V., Misra, A.K.: ‘Detection of plant leaf diseases using image segmentation and soft computing techniques.’ *Inf. Process. Agricult.* **4**, 41–49 (2017). <https://doi.org/10.1016/j.inpa.2016.10.005>



Francis Jesmar P. Montalbo acquired his Doctorate from the Technological Institute of the Philippines and Master of Science degree from Batangas State University in the field of Information Technology focusing on deep learning. Currently, he works as a full-time research faculty for the College of Informatics and Computing Sciences, Batangas State University. He also published and presented works in various international conferences and journals based on his research interests,

including deep learning, computer vision, and medical imaging.