Advances and Applications in Statistics

https://pphmjopenaccess.com
https://doi.org/10.17654/0972361724062
Volume 91, Number 9, 2024, Pages 1165-1194 **P-ISSN: 0972-3617**

ENHANCING INTERPRETABILITY AND FIDELITY IN CONVOLUTIONAL NEURAL NETWORKS THROUGH DOMAIN-INFORMED KNOWLEDGE INTEGRATION

Codjo Emile AGBANGBA[1,*], Rodéo Oswald Y. TOHA[2], Abdou Wahidi BELLO[3] and Jamal ADETOLA[2]

[1]Laboratoire de Biomathématiques et d'Estimations Forestières
Université d'Abomey-Calavi
Calavi, Benin
e-mail: agbangbacodjoemile@gmail.com

[2]Ecole Nationale Supérieure de Génie Mathématique et Modélisation
Université Nationale des Sciences, Technologie, Ingénierie et Mathématique
Abomey, Benin
e-mail: rodeooswald@gmail.com
adetolajamal58@yahoo.com

[3]Université d'Abomey-Calavi
Faculté des Sciences et Techniques
Calavi, Benin
e-mail: vrcireip.uac@uac.bj

Received: May 10, 2024; Revised: June 12, 2024; Accepted: June 19, 2024
2020 Mathematics Subject Classification: 68T07, 62R07.
Keywords and phrases: intelligent agriculture, image classification, convolutional neural networks (CNN), plant diseases, initialization, heatmaps.
*Corresponding author

How to cite this article: Codjo Emile AGBANGBA, Rodéo Oswald Y. TOHA, Abdou Wahidi BELLO and Jamal ADETOLA, Enhancing interpretability and fidelity in convolutional neural networks through domain-informed knowledge integration, Advances and Applications in Statistics 91(9) (2024), 1165-1194. https://doi.org/10.17654/0972361724062

Published Online: August 6, 2024

Abstract

This study addresses the need for robust disease detection methods in vegetable crops by introducing a novel initialization method for convolutional neural networks (CNNs). Rather than creating a new CNN architecture, our approach focuses on infusing expert knowledge from phytopathology directly into the model's foundation. This innovative initialization ensures that the CNN possesses a contextual understanding of intricate disease patterns specific to tomatoes. Additionally, our study redefines the role of heatmaps as a dynamic metric for assessing model fidelity in real-time. Unlike traditional post hoc applications, heatmaps are integrated into the model evaluation process, providing insights into decision-making processes and alignment with expert-derived expectations. This dual innovation aims to enhance transparency and fidelity in CNNs, offering a nuanced and effective solution for disease detection in agriculture. The study contributes to advancing artificial intelligence applications in agriculture by providing accurate predictions and a deeper understanding of the underlying decision mechanisms crucial for crop health management.

1. Introduction

Crops play a vital role in global food security and the agricultural economy. Tomatoes, one of the most widely cultivated vegetables worldwide, are susceptible to various diseases and plant health issues that can have a devastating impact on yields. Early and accurate identification of these diseases is essential for managing and preventing crop losses.

In this context, machine learning and convolutional neural networks (CNNs) have emerged as promising tools for classification [1-4], for detection [5-7], and for segmentation [5, 8].

However, there are many circumstances in which purely data-driven approaches may reach their limits or lead to unsatisfactory results. The most obvious scenario is when there is not enough data to train models that are both high-performing and sufficiently generalized. Another important aspect

is that a model solely based on data may not adhere to constraints such as those dictated by natural laws or provided by regulatory or safety guidelines, which are crucial for trustworthy artificial intelligence (AI) [9].

In general, these models are trained for high accuracy, but recently, there has also been a strong demand to understand how a specific model works and the underlying reasons for the decisions it produces. In other words, on one hand, we need to be able to communicate to the machine what we know in a sufficiently precise form; and on the other hand, the machine must be able to communicate to us whatever it has found in a sufficiently understandable form.

In this context, research directions such as explainable artificial intelligence (AI) [10], informed machine learning [11], and intelligible intelligence have emerged. As machine learning models become increasingly complex, there is a growing need to interpret and explain them [12]. These concerns have led to increased research on how to enhance machine learning models by incorporating prior knowledge into the learning process [13].

Many modern machine-learning techniques focus on creating datasets, data preprocessing, and harnessing computational power. However, a wealth of specialized knowledge embedded in equations, models, or established methods can also be leveraged to enhance machine learning applications. In their study on articulated taxonomy for knowledge-based machine learning, [14] notes that this specialized knowledge can be grouped into three subcategories: scientific knowledge, world knowledge, and expert knowledge.

In this study, we focus on integrating the knowledge held by experts in our field of application, which is phytopathology, into convolutional neural networks (CNNs). The goal is to produce a model with a transparent architecture, internally explainable functioning, and interpretable results.

This study diverges from conventional model development approaches, as it does not aim to create a new CNN architecture but instead proposes a groundbreaking initialization method. This novel approach is designed to

infuse expert knowledge from the field of phytopathology directly into the model's foundation, ensuring that the CNN starts with a contextually rich understanding of the intricate patterns associated with tomato diseases.

Moreover, a key innovation lies in the utilization of heatmaps not merely as a visualization tool but as a dynamic metric for evaluating the fidelity of the model. In contrast to traditional approaches, where heatmaps are often applied post hoc for interpretability, our methodology integrates them into the model evaluation process. This shift promises a more nuanced and real-time assessment of the model's performance, offering insights into its decision-making processes and alignment with expert-derived expectations.

By proposing an innovative initialization method and redefining the role of heatmaps in model evaluation, this study seeks to enhance both the transparency and fidelity of CNNs in the critical domain of phytopathology. This nuanced approach holds the potential to elevate the effectiveness of AI applications in agriculture, providing not just accurate predictions but also a deeper understanding of the underlying decision mechanisms.

2. Material and Methods

2.1. Data

We used two datasets for this study, with one serving as the primary dataset and the other enabling easier transfer learning.

(a) Main dataset

The PlantVillage database, presented in [15], hosts the largest collection of accurately processed leaf images for disease diagnosis. This compilation includes 54,309 images from 14 different crops, expertly labeled by phytopathology specialists. Among these, 14,531 are tomato leaf images, covering healthy states and nine disease categories (Figure 1). Using this dataset to train a convolutional neural network (CNN) for identifying tomato plant diseases is crucial. It provides an opportunity to showcase the model's effectiveness while establishing a benchmark for evaluating other

methodologies. The images were in RGB, and variations in sample numbers between classes were observed. However, imbalances existed between classes, presenting challenges such as a less comprehensive view of underrepresented classes. Although these classes have a minor impact on overall accuracy, excluding them can maintain high precision [16]. To address this issue, various under-sampling and oversampling techniques can be applied, ensuring effective handling of all diseases, including the less frequent ones.

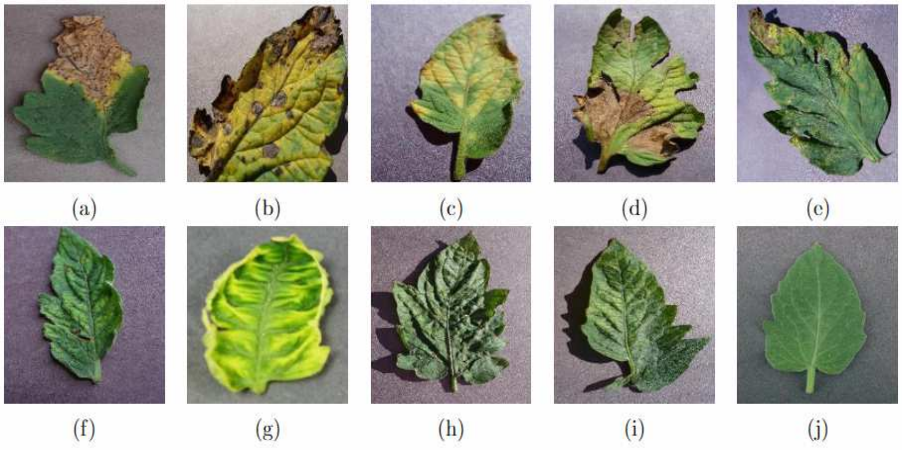

Figure 1. Examples of images of leaves with diseases and pathogen for (a) Bacterial spot, (b) Early blight, (c) Leaf mold disease, (d) Late blight, (e) Septoria leaf disease, (f) Tomato mosaic, (g) Yellow leaf curl disease, (h) Target spot, (i) Two-spotted spider mite, and (j) Healthy.

(b) Secondary dataset

We created a second dataset comprising other vegetable crops (potato, strawberry, bell pepper, and sweet potato) with which we trained the CNN2 model for weight initialization using a knowledge-sharing technique. These data were also sourced from the extensive PlantVillage database in 2022. The dataset consisted of 8,803 images of affected vegetable crop leaves, excluding tomatoes, distributed across 10 categories, all in the RGB color space. Using this dataset provided insights into the detection and

classification of various diseases affecting vegetable crops. This knowledge gained from the additional dataset has been shared with the CNN2 model to enhance its ability to classify diseases affecting tomato plants. Similar to the main dataset, this dataset underwent the same basic processing steps.

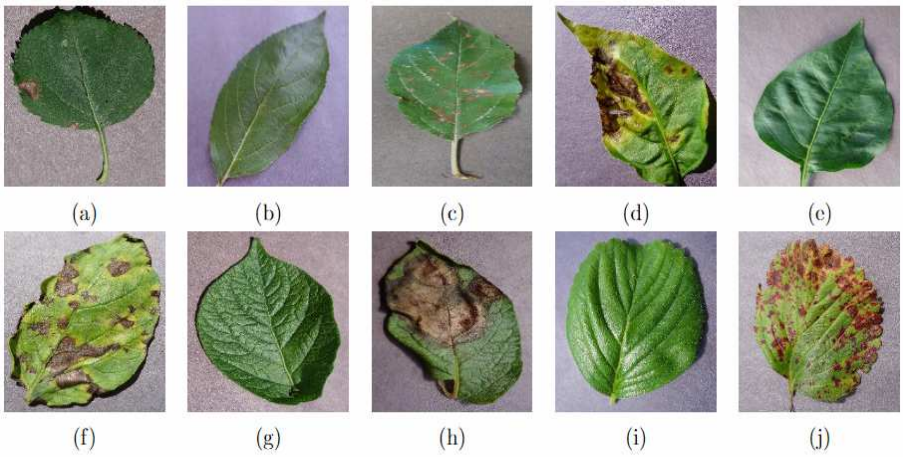

Figure 2. Examples of images of leaves with diseases and pathogen for (a) Black rot of apple, (b) Healthy apple, (c) Apple rust, (d) Pepper bacterial spot, (e) Healthy pepper, (f) Potato blight, (g) Potato late blight, (h) Healthy potato, (i) Healthy strawberry, and (j) Strawberry leaf burn.

2.2. Models

(a) Model's architecture

The three models used in this study shared the same architecture, which is structured as follows: the proposed CNN architecture consisted of three convolutional blocks followed by max-pooling layers. The convolutional layers used 3×3 filters, progressively reducing spatial dimensions. The architecture concluded with a dense layer of 100 units and a final output layer of 10 units with a Softmax activation function for classification. Each convolutional layer was activated using the Rectified Linear Unit (ReLU) activation function. The model takes RGB (Red-Green-Blue) images with dimensions of 227×227 pixels as input.

(b) Model's description

Table 1 presents the detailed architecture of the three models used in this study. The model's input consisted of images with dimensions (227, 227, 3), representing height, width, and the three RGB channels. The network comprised three convolutional layers, each followed by a max-pooling layer to progressively extract salient features from the image. The first convolutional layer (Conv2D) employed 32 filters of size (3, 3) with a ReLU activation function (Table 1). It was then followed by a max-pooling layer (MaxPooling2D) with a kernel size of (2, 2). The convolution and max-pooling steps were repeated twice with successive layers, gradually reducing the spatial dimension of the extracted features. Subsequently, a flattening layer (Flatten) was applied to convert the features into a one-dimensional vector. This vector was then connected to two fully connected layers (Dense) with ReLU activation functions. The first dense layer had 100 neurons, followed by an output layer with 10 neurons corresponding to possible classes, activated by a Softmax function for classification. The values in the "Param" column indicated the number of parameters in each layer, reflecting the model's complexity. This architecture aimed to capture increasingly abstract feature hierarchies, thereby facilitating the classification task.

Table 1. Architecture of CNN model

Layer	Output shape	Parameters	Activation
Input	(227, 227, 3)	0	-
Conv2D (32, (3,3))	(225, 225, 32)	896	ReLU
MaxPooling2D (2,2)	(112, 112, 32)	0	-
Conv2D (18, (3,3))	(110, 110, 18)	5202	ReLU
MaxPooling2D (2,2)	(55, 55, 18)	0	-
Conv2D (9, (3,3))	(53, 53, 9)	1467	ReLU
MaxPooling2D (2,2)	(26, 26, 9)	0	-
Flatten	(6084, 1)	0	-
Dense (100)	(100, 1)	608500	ReLU
Dense (10)	(10, 1)	1010	Softmax

(c) Model initialization methods

The main difference between the three models (CNN1, CNN2, and Ti-CNN) lies in the technique used to initialize the weights of the convolution filters.

CNN1

The CNN1 model utilized Glorot initialization, also known as Xavier initialization [9], defined as follows for a layer with n_{in} input units and n_{out} output units:

Let W be the weight matrix of the layer, and the elements of W be initialized from a distribution centered at zero (mean = 0) with a variance

$$Var(W) = 2/(n_{in} + n_{out}).$$

In other words,

$$W_{ij} \sim N\left(0, \frac{2}{n_{in} + n_{out}}\right),$$

where N represents the normal (Gaussian) distribution.

This ensures that the weights are initialized to have a balanced variance between the inputs and outputs of the layer, helping mitigate issues such as vanishing or exploding gradients during neural network training. The variance formula used in Glorot initialization is a heuristic that has empirically proven effective for various neural network architectures.

CNN2

For this second model, we employed the knowledge-sharing technique (Figure 3). Transfer learning involves reusing a pre-trained model on a new problem. It is currently highly popular in deep learning because it enables the training of deep neural networks with relatively small datasets. This is particularly useful in the field of data science, where real-world problems typically do not have millions of labelled data points to train such complex models. With transfer learning, the basic idea is to leverage what has been

learned in one task to improve generalization in another. The weights learned by a network during "Task 1" are transferred to a new "Task 2". This is especially valuable when "Task 1" has a large set of labelled data, allowing the model to learn meaningful representations, which can then be beneficial for solving "Task 2", even if the latter has a limited amount of labelled data. By transferring knowledge in the form of pre-trained weights, the model starts with a head start, potentially accelerating the learning process on the new task and often leading to better performance than training a model for "Task 2" from scratch.

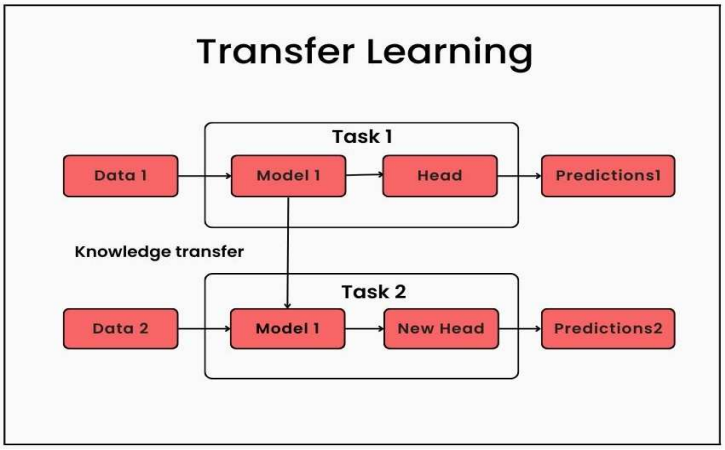

Figure 3. Transfer learning.

Ti-CNN

Regarding our Ti-CNN model, we followed the process of integrating domain-informed knowledge into a CNN model. Since this involved expert knowledge, we utilized it to influence the architecture of the model, specifically at the level of filters in the first convolutional block. The whole process can be described in three steps:

- **First step:** Information acquisition

First, we gathered information about the characteristic symptoms of the different diseases we are dealing with in this study case.

▪ **Second step:** Mathematical analysis of the information gathered

After gathering information about the manifestation of all the diseases, we analyze them from a mathematical standpoint. The key work to do at this step is to find the different filters that could better identify the symptoms.

▪ **Third step:** Integration of the information gathered in the architecture

Lastly, after identifying the different filters that best describe the symptoms, we integrate them into the architecture as initialization to the model.

2.3. Models' performance evaluation metrics

The performance of the developed method for disease classification was evaluated using various parameters such as sensitivity, specificity, and accuracy, defined as follows:

(a) Accuracy

$$Accuracy = \frac{TN + TP}{TP + TN + FP + FN}.$$

Accuracy measures the proportion of images correctly classified by the model compared to the total set of images. It provides an overall view of the model's performance.

(b) Recall

$$Recall = \frac{TP}{TP + FN}.$$

Recall, also known as sensitivity in binary classification, evaluates the model's ability to identify all instances of a given class. It is particularly crucial when missing a positive detection is costly.

(c) Precision

$$Precision = \frac{TP}{TP + FP}.$$

Precision measures the quality of the model's positive predictions. It is particularly useful when minimizing false positives is critical.

- TP (True Positive) is the proportion of correctly identified positive cases. For example, in our disease classification study, a TP is obtained when the model correctly identifies a leaf affected by mildew.

- TN (True Negative) is the proportion of correctly identified negative cases. For example, a TN is obtained when the model correctly identifies a leaf as not affected by Alternaria.

- FP (False Positive) is the proportion of negative cases identified incorrectly as positive. In this study, this error occurs when the model incorrectly detects that a leaf is not affected by Alternaria, which is true in reality.

- FN (False Negative) is the proportion of positive cases identified incorrectly as negative. This occurs when the model incorrectly classifies a leaf affected by Alternaria as not affected by this disease.

2.4. Visualization method

In the context of our study, we visualized heatmaps to understand the rationale behind the deep learning models used.

The heatmap visualization for CNN models relies on the activations of filters in the convolutional layers, which can be formulated mathematically as follows. Let $F_i(x, y, c)$ be the activation value of filter i at position (x, y) in channel c. Then, the heatmap $H(x, y)$ for a given class can be calculated as the weighted sum of filter activations:

$$H(x, y) = \sum_i w_i \cdot F_i(x, y, c),$$

where w_i represents the weight associated with filter i, and c is the channel corresponding to the class of interest. Typically, these weights can be determined using adaptive weighting methods, such as class weights or gradients of the output concerning activations of the last convolutional layer.

Applying an activation function, such as ReLU, to the heatmap produces a visual representation where high values indicate areas of the image that

have a strong influence on the class prediction. This mathematical process provides a quantitative understanding of regions of interest, thereby enhancing the transparency and interpretability of the CNN model. Incorporating this mathematical analysis of heatmaps into model evaluation allows for a more thorough validation of the relevance of extracted features, improving the reliability and efficiency of CNNs in various applications.

2.5. Diagnosing the fidelity of the model

We introduce a new evaluation procedure in this study called "diagnosis". This procedure aims to assess the fidelity of each model to the logic of the application domain or specific expert practices. It allows us to determine the ratio of True Positives obtained while adhering to a certain logic approved by the expert to the total number of True Positives:

$$R = \frac{\text{Number of approved true positives}}{\text{Total number of true positives}}.$$

In our case, for example, if the model correctly classifies a leaf affected by mildew, and upon analysis, we determine that the model adhered to a minimum set of criteria according to the logic of the experts, we consider it an "Approved True Positive". Conversely, if the model's decision cannot be justified, and it is unclear how it arrived at the correct decision, we consider it an "Unapproved True Positive".

This diagnostic approach provides a nuanced evaluation by considering not only the accuracy of predictions but also the adherence to domain-specific logic or expert-approved criteria in disease classification.

3. Results

3.1. Presentation of the convolutional filters at the initialization

We presented a brief visualization of the initialization filters for each of the three models used (Figures 4-6). Since the literature review placed particular emphasis on the hierarchy in CNN learning, we only focused on the filters in the first block of the architecture.

Model CNN1

As previously mentioned, this model utilized the default initialization method in the TensorFlow tool (specifically, the Xavier-Glorot initialization as presented in the previous section), which we used and was also highlighted in the literature.

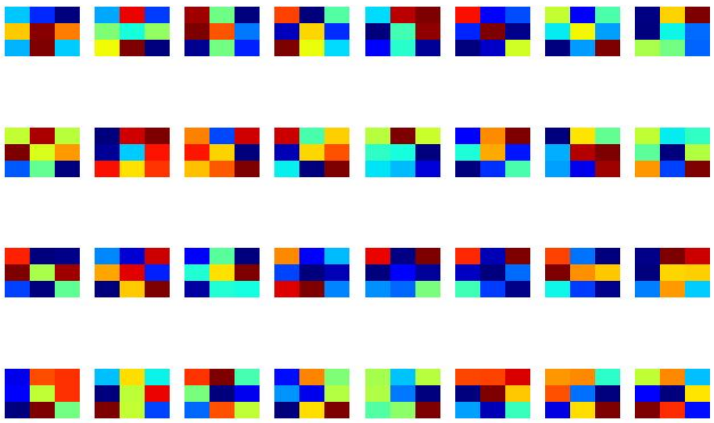

Figure 4. First convolutional layer filter of CNN1.

Model CNN2

This model used the knowledge-sharing method, meaning it was already somewhat informed about certain aspects of the primary dataset. Therefore, we anticipated observing some filters that were familiar with extracting useful features or patterns from our dataset.

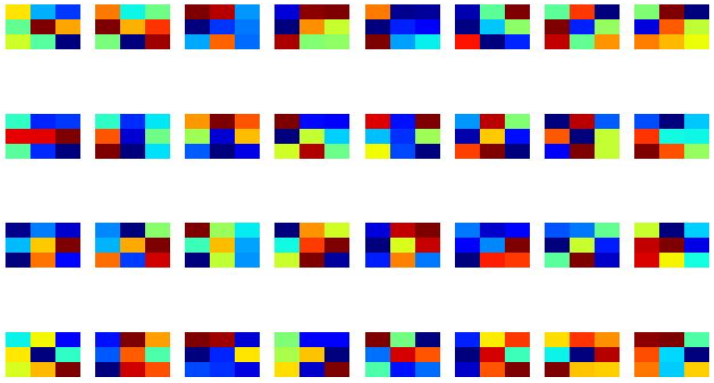

Figure 5. First convolutional layer filter of CNN2.

Model Ti-CNN

This final model, in general, possessed most of the useful and necessary information for accurate detection and classification of various diseases, i.e., for a proper recognition of relevant patterns in the dataset. We, therefore, understood the specific purposes of each of the filters below, as we have selected them based on prior knowledge derived from expert insights.

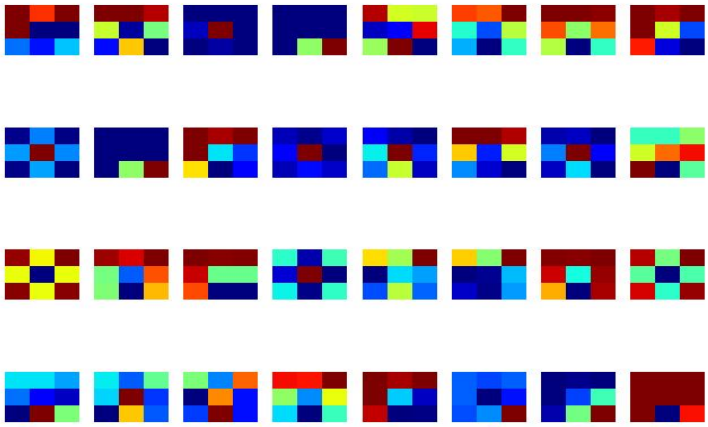

Figure 6. First convolutional layer filter of Ti-CNN.

3.2. Application of the three models on vegetable crops diseases classification

The current subsection aimed to present the results obtained through the application of three distinct models in the context of classifying diseases affecting vegetable crops, with a particular focus on tomato cultivation. The considered models were CNN1, implemented with Xavier Glorot initialization for its filters, CNN2, using transfer learning and trained on a secondary dataset, and Ti-CNN, where the filters of the first layer were initialized using domain-specific knowledge. The results were presented in the form of tables and graphs, highlighting the individual performances of each model, significant differences between them, as well as in-depth analyses of performances per disease class. This subsection aimed to provide a detailed understanding of the models' performances.

3.2.1. Hyperparameters

We presented in the tables below the hyperparameters used to assess the performances, along with the performances based on the metrics previously presented. These hyperparameters consisted of three characteristics:

- Batch size: This refers to the number of samples that are propagated through the entire network to update the weights. The use of batches has the advantage of requiring less memory improving the learning speed of networks after each propagation.

- Epochs: This is the number of times the algorithm works on all training datasets. An epoch is composed of one or more batches.

- Learning rate: The learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while approaching a minimum loss function.

3.2.2. Performances of the models

- **Performance of CNN2 on the secondary dataset**

Table 2 shows that the training of the CNN2 model on the secondary dataset yielded an accuracy of 91.8%. This indicates that the model can correctly classify various diseases found in relevant vegetable crops. The precision of 92.42% showed that out of all instances predicted as positive by the CNN2 model, 92.42% were truly positive (true positives), while 14.01% were false positives (instances predicted as positive but were actually negative). Finally, the recall rate of 91.15% indicates that 91.15% of all truly positive instances (true positives) were correctly identified among all truly positive instances and those that were missed (false negatives).

Table 2. Performance of CNN2 on the secondary dataset

Metrics (%)	Accuracy	Precision	Recall
CNN2	91.8	92.42	91.15

• **Performances of the three models on the main dataset**

All the three models performed well, with CNN2 achieving the highest accuracy of 92.06%, followed by CNN1 with 86.27%, and Ti-CNN with 81.02% (Table 3).

Table 3. Performances of the models on the main dataset

Metrics (%)	Accuracy	Precision	Recall
CNN1	86.27	89.38	84.34
CNN2	92.06	92.71	87.72
Ti-CNN	81.02	85.99	74.12

3.2.3. Performances' graphics

This subsection presented various plots illustrating the variation of metrics such as accuracy, precision, and recall, as well as the loss function over the number of epochs.

• **For CNN1**

Among the presented graphs, the precision graph stands out due to the instability observed in the validation data (Figures 7-10). For instance, at epoch 5, while the CNN1 model detects with 80% precision, it achieves 87% precision on the validation data, which it has never seen before. However, this is not a cause for celebration as the model has actually fallen into the trap of underfitting. However, during epoch 7, the model experienced overfitting issues, resulting in a precision of only 86% on the validation data. Despite this setback, the model has performed well overall, exhibiting excellent sensitivity to various disease categories. The loss function curve indicated a typical minimization of loss, striving for the optimal minimum.

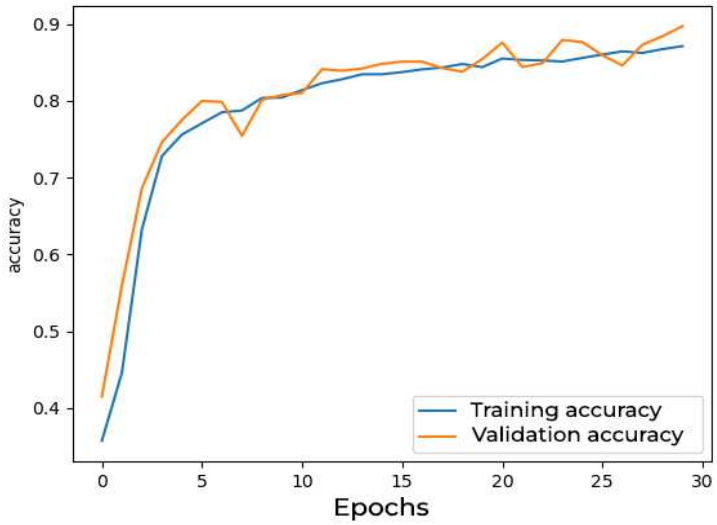

Figure 7. The CNN1 model's accuracy in classifying various diseases on the primary dataset.

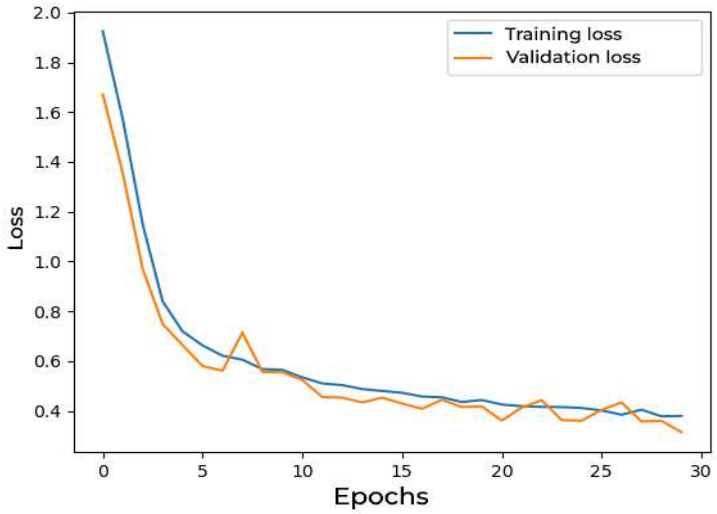

Figure 8. The CNN1 model's loss in classifying various diseases on the primary dataset.

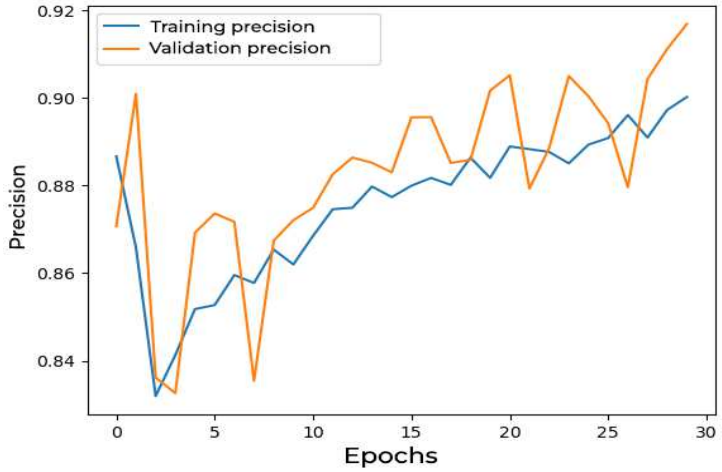

Figure 9. The CNN1 model's precision in classifying various diseases on the primary dataset.

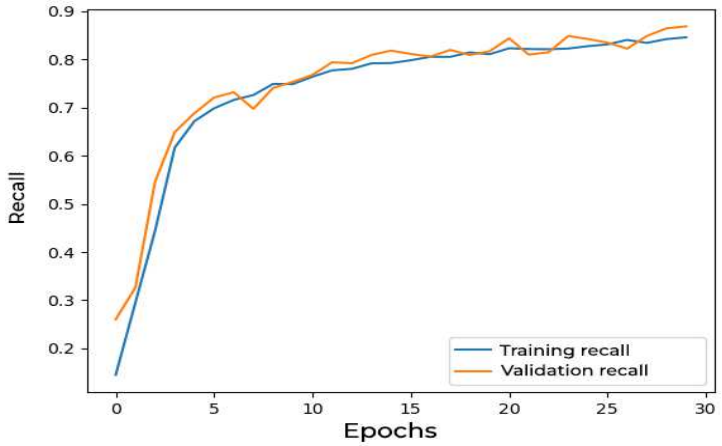

Figure 10. The CNN1 model's recall in classifying various diseases on the primary dataset.

- **For CNN2**

The metrics show a normal evolution and the cost function was minimized. This was expected as the model was well-trained on a secondary dataset related to diseases affecting vegetable crops. The model

demonstrated a good balance between sensitivity to errors and precision in detecting different disease categories.

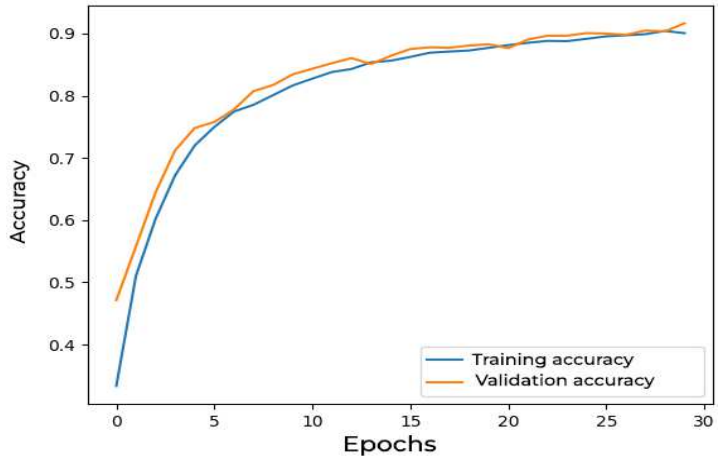

Figure 11. The CNN2 model's accuracy in classifying various diseases on the primary dataset.

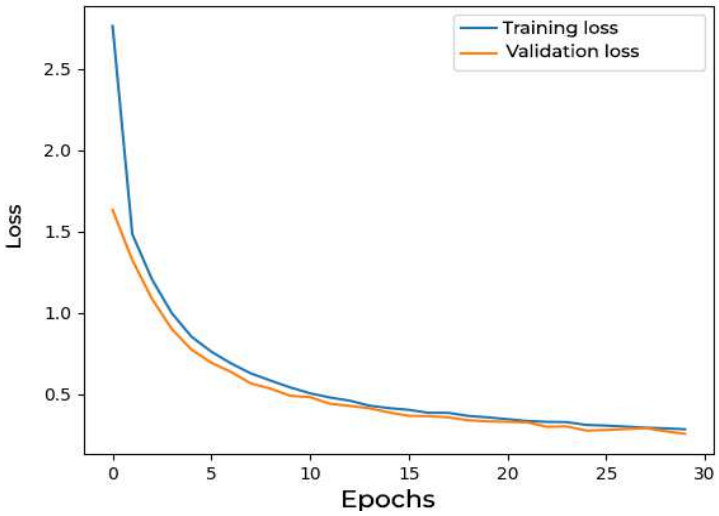

Figure 12. The CNN2 model's loss in classifying various diseases on the primary dataset.

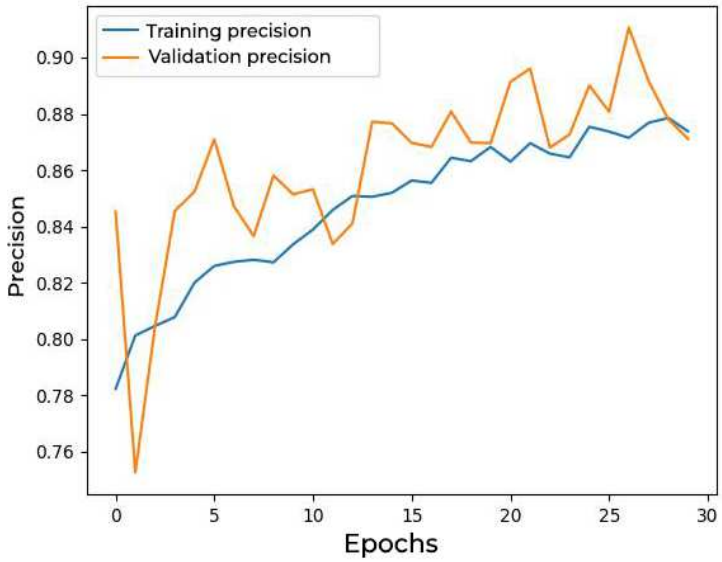

Figure 13. The CNN2 model's precision in classifying various diseases on the primary dataset.

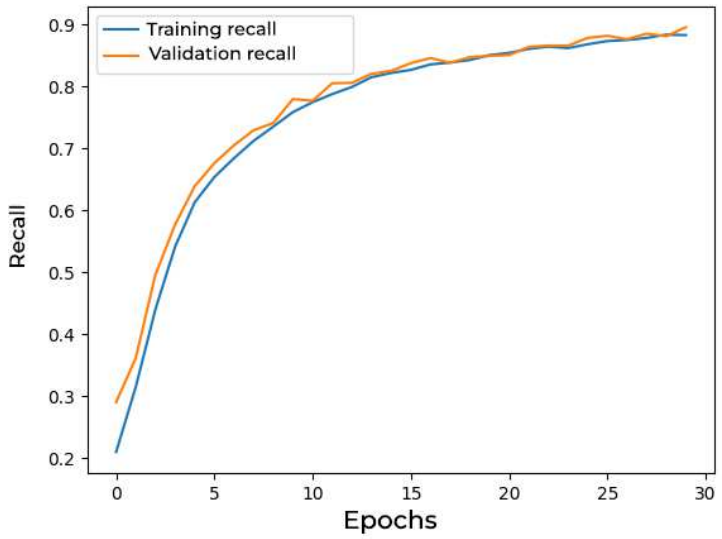

Figure 14. The CNN2 model's recall in classifying various diseases on the primary dataset.

• **For Ti-CNN**

Finally, the model used expert knowledge to initialize the weights of the first layer, subtly sharing certain knowledge. Prior information on diseases was used to manually identify convolutional filters. The model detected essential features hierarchically, starting with the most basic and progressing to the most complex. The reason for initializing only the first layer is that all the other layers depend on it. This innovation enabled the model to achieve an accuracy of 81.02%. The graphs indicated many instabilities in variations regarding precision in detection (Figures 15-18). Our initial goal was not to build a super-powerful model, but rather one that we could control and explain its decisions.

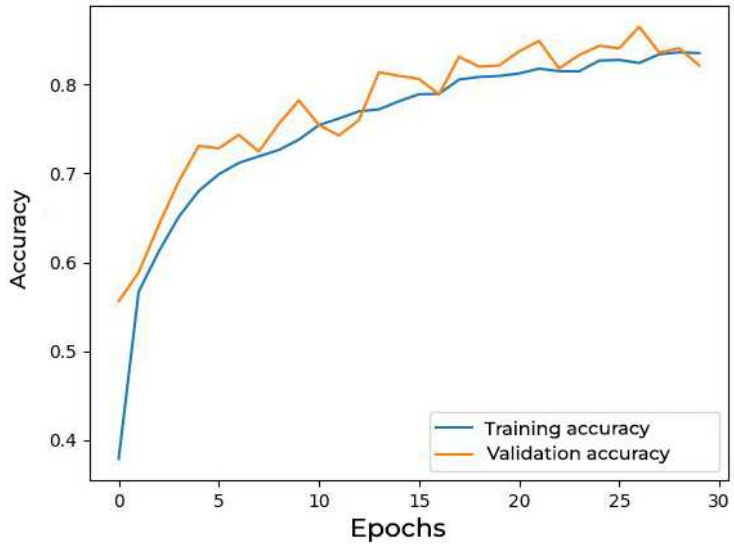

Figure 15. The Ti-CNN model's recall in classifying various diseases on the primary dataset.

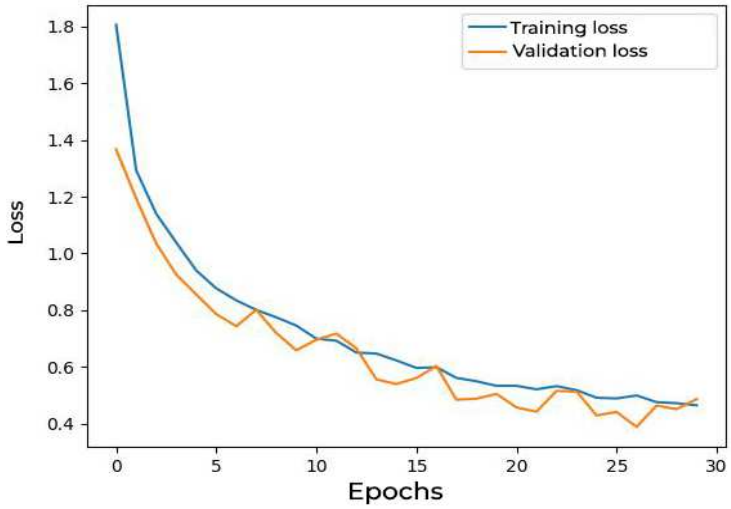

Figure 16. The Ti-CNN model's loss in classifying various diseases on the primary dataset.

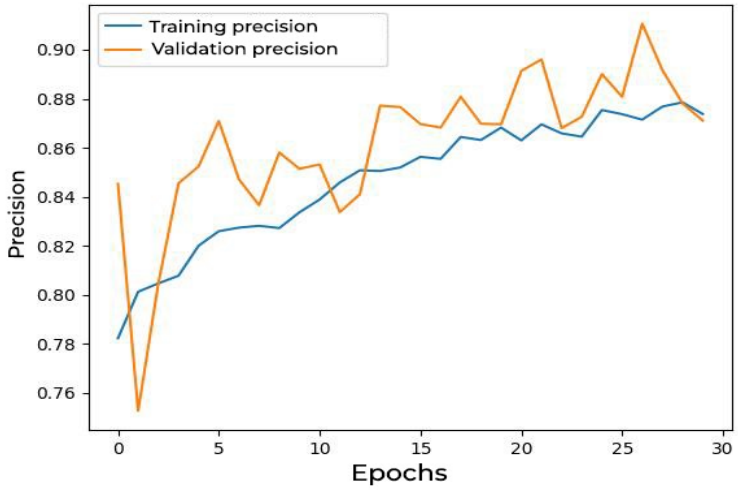

Figure 17. The Ti-CNN model's precision in classifying various diseases on the primary dataset.

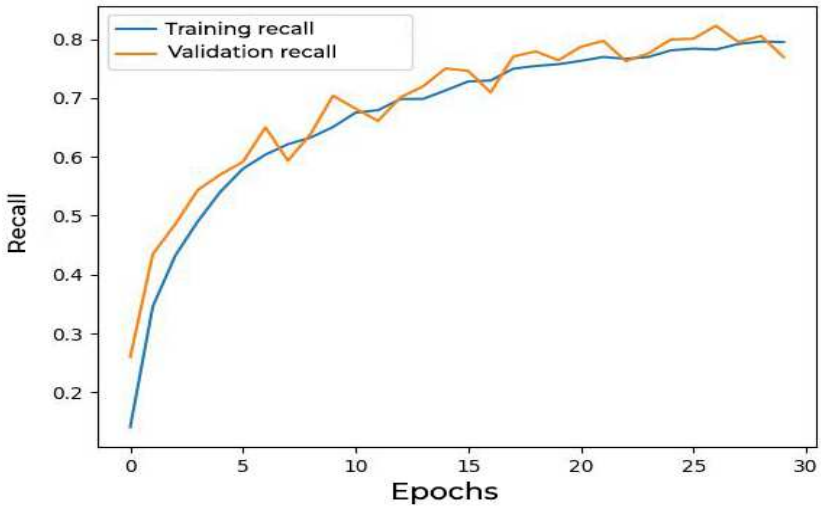

Figure 18. The Ti-CNN model's recall in classifying various diseases on the primary dataset.

3.3. Heatmaps visualization

To diagnose the internal functioning of the model, we selected a sample of leaves that exhibited characteristic symptoms of each disease. This allowed us to examine what each model was learning beyond the performance metrics mentioned above. For an effective and efficient analysis, we focused on strong activation zones. The intense color areas on the heatmap indicated the regions of the image that strongly contributed to the prediction of the class. These were important areas where the model focused its attention. Furthermore, we compared the activated zones on the heatmap with the known symptoms of tomato plant diseases. If certain areas corresponded to parts of the plant where symptoms were frequently observed, then it strengthened confidence in the model's ability to identify relevant features.

An analysis of the grid provided insights into the behavior of the CNN1 model regarding the diseases in the dataset. The observation was generalized, as each selected leaf represented the symptoms of each disease

in a representative manner (Figure 19). The model was more sensitive to different symptoms presented in the form of shapes. The intensification of colors at the locations of lesions on certain leaves was well illustrated. However, the model appeared to have difficulty in identifying features represented in the form of texture, such as leaf curling, despite its remarkable overall performance.

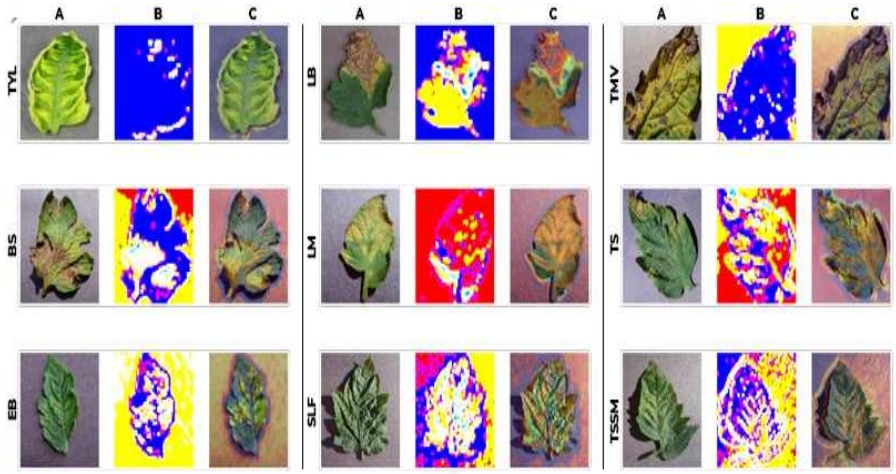

Figure 19. Heatmaps for a number of affected leaves (CNN1).

We observed the behavior of the CNN2 model on our dataset. Since this model was the best performer among the three models used in this study, we expected it to be more effective in learning representations of each symptom manifested in our dataset. Indeed, the model showed some insight in identifying most of the diseases (5 out of 9 cases), as illustrated (Figure 20). However, the model seemed to struggle in its reasoning for four disease cases where the symptoms were more represented by textural features.

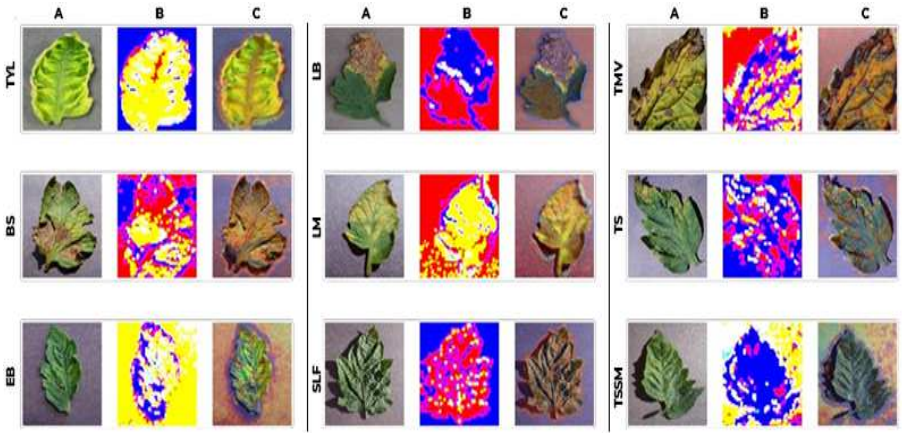

Figure 20. Heatmaps for a number of affected leaves (CNN2).

Finally, we have the heatmaps for the last model, Ti-CNN, which we used with filter initialization in the first convolutional layer based on expert knowledge. Our previous performance results showed that this model performed less well than the other two in this classification task. However, we observed that where the other two models seemed somewhat unreasonable, this model proved to be more effective. We can see some accuracy in identifying the characteristics of leaf curling (Figure 21).

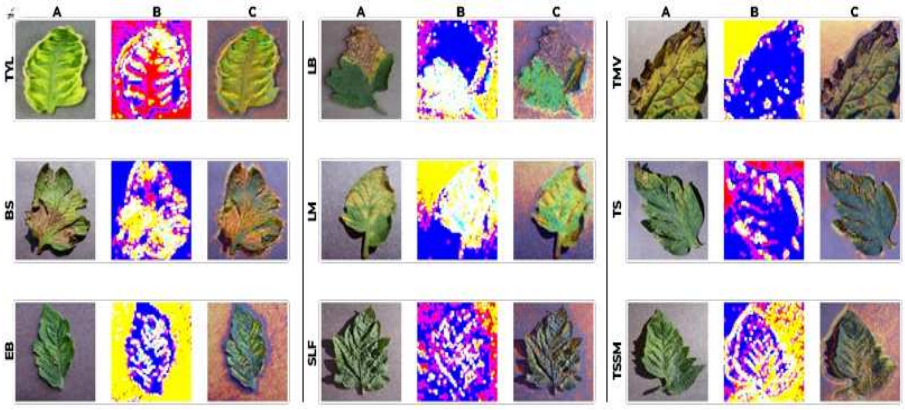

Figure 21. Heatmaps for a number of affected leaves (Ti-CNN).

3.4. Fidelity analysis

This diagnosis, which is highly dependent on the different visualization methods proposed in the literature to understand what neural network models learned, shows that of all the good decisions made by the CNN2 model to identify different diseases, only 51% seem to agree according to our analyses of different heatmap visualizations. In contrast, the Ti-CNN model outperformed the CNN1 model with a calculated proportion of 47% compared to 32%. This means that, although the Ti-CNN model is less accurate than the CNN1 model, it is recognized as being more faithful to the logic of the domain than the CNN1 model, which was 10% more sensitive than the Ti-CNN model. This diagnosis, based on our observations and comparisons of strong activation zones in heatmaps, shows that a high-performance model may well ignore all the rules of the application domain. This justifies the black-box nature of CNNs.

If we look at the calculated proportion for CNN1, which is 32%, this suggested that 68% of the correct decisions come from the model's discoveries during training (Table 4). While this may not seem problematic, in domains where decision-making is highly critical, most people would prefer a less powerful model that is much more faithful than a powerful model that is potentially unfaithful.

Table 4. Fidelity ratio

Models	CNN1	CNN2	Ti-CNN
Ratio (%)	32	51	47

4. Discussion

This study aims to explore a new methodology to improve the explainability and interpretability of results produced by convolutional neural network (CNN) models for the task of detecting and classifying diseases affecting certain vegetable crops. Three CNN models with the same architecture but different filter initializations were used and evaluated. The results showed that CNN2, which uses knowledge transfer, outperformed

the other two models - CNN1 initialized with the Xavier-Glorot method, and Ti-CNN, which integrates domain-specific information. In terms of performance, if one were to recommend a model based solely on performance, CNN2 would be the clear choice due to its superior performance.

However, given the context in which the study was conducted, performance is not the only criterion for recommending a model. Instead, the focus is on the logic [18], explainability [19], and interpretability [9] of the decisions made by the model. As highlighted by the authors in [20], the current concern in the scientific community is to understand the logic or reasoning behind the model's decisions. By providing the model with understandable knowledge that is critical to identifying a particular disease, the model should fully rely on this knowledge in its decision-making process and communicate it to us. This crucial detail was missing in the two best-performing models (CNN1 and CNN2). Despite the integration of useful information in the form of filter initialization, the Ti-CNN model still seems to lack clarity, which could be attributed to various factors such as the input data, the level of information integration, the backpropagation algorithm, and many other reasons.

5. Conclusion

This study investigated the potential of convolutional neural network (CNN) models in the context of disease detection and classification in vegetable crops, with a particular focus on tomatoes' diseases. Three different approaches were used, namely, CNN1 with Xavier initialization, CNN2 with transfer learning and Ti-CNN with manual initialization of filters in the first layer based on expertise. Our novel approach, implemented in Ti-CNN, used expert knowledge to manually initialize the filters in the first layer, which made an innovative contribution in improving the model performance. However, it is worth noting that despite this innovation, the CNN2 model emerged as the best performing model with an impressive score of 92%. Heatmap analysis was a key element of our methodology,

providing an in-depth understanding of the decision-making processes of each model. This visualization approach helped to interpret the choices made by the models in disease detection and provided important insights for further improvement and optimization of our approaches. In conclusion, our results demonstrate the effectiveness of CNN models in disease detection in vegetable crops, especially tomatoes. Although our innovative approach in Ti-CNN showed promise, CNN2 outperformed the other models in terms of performance. Looking ahead, we plan to explore various ways of integrating knowledge to improve the interpretability and explainability of neural network models, intending to alleviate concerns about their use in decision-critical domains.

Acknowledgement

We thank the anonymous referees for their comments and feedback on earlier version of this document.

References

[1] R. Ahmad, I. Alsmadi, W. Alhamdani and L. A. Tawalbeh, Zero-day attack detection: a systematic literature review, Artificial Intelligence Review 56(10) (2023), 10733-10811.

[2] J. M. Alonso Moral, C. Castiello, L. Magdalena and C. Mencar, Toward explainable artificial intelligence through fuzzy systems, Explainable Fuzzy Systems: Paving the Way from Interpretable Fuzzy Systems to Explainable AI Systems, Springer, 2021, pp. 1-23.

[3] N. R. Ashwin, Z. Cao, N. Muralidhar, D. Tafti and A. Karpatne, Deep learning methods for predicting fluid forces in dense particle suspensions, Powder Technology 401 (2022), 117303.

[4] M. Brundage et al., Toward trustworthy AI development: mechanisms for supporting verifiable claims, 2020. arXiv preprint arXiv:2004.07213.

[5] T. Chauhan and S. Sonawane, Explicable AI for surveillance and interpretation of coronavirus using X-ray imaging, 2023 International Conference on Emerging Smart Computing and Informatics (ESCI), IEEE, 2023, pp. 1-6.

[6] V. A. G. da Cunha, J. Hariharan, Y. Ampatzidis and P. D. Roberts, Early detection of tomato bacterial spot disease in transplant tomato seedlings utilising remote sensing and artificial intelligence, Biosystems Engineering 234 (2023), 172-186.

[7] Y. Deng, L. Wang, C. Zhao, S. Tang, X. Cheng, H. W. Deng and W. Zhou, A deep learning-based approach to extracting periosteal and endosteal contours of proximal femur in quantitative CT images, 2021.
arXiv preprint arXiv:2102.01990.

[8] G. Geetharamani and A. Pandian, Identification of plant leaf diseases using a nine-layer deep convolutional neural network, Computers and Electrical Engineering 76 (2019), 323-338.

[9] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249-256.

[10] S. M. Javidan, A. Banakar, K. A. Vakilian and Y. Ampatzidis, Tomato leaf diseases classification using image processing and weighted ensemble learning, Agronomy Journal 116(3) (2024), 1029-1049.

[11] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder and N. Seliya, A survey on addressing high-class imbalance in big data, Journal of Big Data 5(1) (2018), 1-30.

[12] D. Minh, H. X. Wang, Y. F. Li and T. N. Nguyen, Explainable artificial intelligence: a comprehensive review, Artificial Intelligence Review 55 (2022), 3503-3568.

[13] S. G. Paul, A. A. Biswas, A. Saha, M. S. Zulfiker, N. A. Ritu, I. Zahan, M. Rahman and M. A. Islam, A real-time application-based convolutional neural network approach for tomato leaf disease classification, Array 19 (2023), 100313.

[14] P. J. Phillips, C. Hahn, P. Fontana, A. Yates, K. K. Greene, D. A. Broniatowski and M. A. Przybocki, Four principles of explainable artificial intelligence, NISTIR 8312, 2021, 36 pp.

[15] A. M. Roy, J. Bhaduri, T. Kumar and K. Raj, WilDect-YOLO: an efficient and robust computer vision-based accurate object localization model for automated endangered wildlife detection, Ecological Informatics 75 (2023), 101919.

[16] R. Thangaraj, P. Pandiyan, S. Anandamurugan and S. Rajendar, A deep convolution neural network model based on feature concatenation approach for classification of tomato leaf disease, Multimedia Tools and Applications 83(7) (2024), 18803-18827.

[17] L. Von Rueden et al., Informed machine learning – a taxonomy and survey of integrating prior knowledge into learning systems, IEEE Transactions on Knowledge and Data Engineering 35(1) (2021), 614-633.

[18] Y. Xu, S. Kohtz, J. Boakye, P. Gardoni and P. Wang, Physics-informed machine learning for reliability and systems safety applications: state of the art and challenges, Reliability Engineering and System Safety 230 (2023), 108900.

[19] A. D. Selbst and S. Barocas, The intuitive appeal of explainable machines, Fordham L. Rev. 87 (2018), 1085.

[20] J. ArunPandian and G. Gopal, Data for: Identification of plant leaf diseases using a 9-layer deep convolutional neural network, 2019. Retrieved from https://api.semanticscholar.org/CorpusID:192568744.