Explainability-Driven Leaf Disease Classification Using Adversarial Training and Knowledge Distillation

Sebastian-Vasile Echim[1,*], Iulian-Marius Tăiatu[1,*], Dumitru-Clementin Cercel[1,a] and Florin Pop[1,2,3]

[1]*Computer Science and Engineering Department, National University of Science and Technology POLITEHNICA Bucharest, Romania*
[2]*National Institute for Research and Development in Informatics, ICI Bucharest, Romania*
[3]*Academy of Romanian Scientists, Bucharest, Romania*

Keywords: Adversarial Attacks, Adversarial Defense, Explainable AI, Knowledge Distillation.

Abstract: This work focuses on plant leaf disease classification and explores three crucial aspects: adversarial training, model explainability, and model compression. The models' robustness against adversarial attacks is enhanced through adversarial training, ensuring accurate classification even in the presence of threats. Leveraging explainability techniques, we gain insights into the model's decision-making process, improving trust and transparency. Additionally, we explore model compression techniques to optimize computational efficiency while maintaining classification performance. Through our experiments, we determine that on a benchmark dataset, the robustness can be the price of the classification accuracy with performance reductions of 3%-20% for regular tests and gains of 50%-70% for adversarial attack tests. We also demonstrate that a student model can be 15-25 times more computationally efficient for a slight performance reduction, distilling the knowledge of more complex models.

1 INTRODUCTION

Artificial intelligence (AI) has provided the theoretical knowledge to develop real-world applications for decades. The amalgamation of enhanced computational capacities, refined learning algorithms, and increased data accessibility has propelled significant advances in machine learning, resulting in the use in many domains (Linardatos et al., 2021). Furthermore, the advancement of deep learning techniques, demonstrated by their ongoing refinement, has reinforced their pivotal role in the AI landscape (He et al., 2016; Khan et al., 2022).

Nevertheless, although deep neural networks (DNNs) can attain best performance for various tasks in domains such as legal (Smădu et al., 2022), energy (Năstăsescu and Cercel, 2022), and medicine (Lungu-Stan et al., 2023), it has been demonstrated that these networks are prone to classification errors for data with imperceptible noise (Szegedy et al., 2014; Goodfellow et al., 2015). Thus, the network sensitivity to perturbations is explored using adversarial algorithms (Goodfellow et al., 2015; Kurakin et al., 2017). They are used in network attacks, robustness improvements,

[a]Corresponding author
[*]Equal contributions.

or data augmentations to achieve better performance with adversarial training (Miyato et al., 2016).

The prediction accuracy is frequently attained by elevating DNN architecture complexity (Linardatos et al., 2021; Christian Meske and Gersch, 2022). However, boosting a model's predictive power reduces its ability to explain its inner workings and mechanisms. Consequently, the rationale behind their decisions becomes quite challenging to understand, and, therefore, their predictions take time to interpret. Considerable emphasis has been placed on elucidating the explainability of deep learning, particularly in image analysis, specifically focusing on the concept of saliency maps (Ramirez-Moreno et al., 2013; Simonyan et al., 2014).

We conduct a series of experiments on a publicly available dataset (Geetharamani and Pandian, 2019) to better understand DNNs and their performance in the context of adversarial attacks and defenses (Szegedy et al., 2014), explainability (Simonyan et al., 2014), and knowledge distillation (Buciluă et al., 2006). We bring in this work the following main contributions:

- We demonstrate the performance impact on DNNs when extra training data is generated with adversarial algorithms for achieving better robustness.

- We indicate the effectiveness of adversarial attacks on models not trained with adversarial data and

the defense capability when adversarial training is also performed.

- We show that different shapes and orientations of a plant leaf to be classified can determine different results when using adversarial attacks.

- We offer a closer look into the attentional patterns, demonstrating that models can have significant focus shifts away from the main classification points of interest when attacked.

- We find a good trade-off solution between accuracy and efficiency using knowledge distillation, obtaining only 13-15% classification accuracy reduction for models 2-14 times smaller and 15-25 times more efficient.

The paper is structured in six sections, starting with the introduction and the related work. The third section describes the dataset characteristics. The fourth section covers the methods used in our experiments, indicating the network architectures employed, the adversarial algorithms applied, the performance metrics used to analyze the results, and the main research questions on which we based our experiments. Finally, the work covers the result analysis in Section 5 and closes with a conclusion section.

2 RELATED WORK

Geetharamani and Pandian (Geetharamani and Pandian, 2019) proposed a leaf disease identification model based on DNNs. The authors train their model using a dataset with 39 different classes of plant leaves. This predictive model achieved impressive results, boasting a 96.46% accuracy in classification. This performance surpasses classical machine learning baselines. They anticipate even better results with a more complex model incorporating adversarial training. In this experimental area, You et al. (You et al., 2023) evaluated a set of adversarial algorithms to attack neural networks such as ResNet50 (He et al., 2016) and VGG16. Moreover, using the adversarial examples as data augmentation, the authors achieved performance improvements.

Recently, knowledge distillation has become a significant strategy in plant leaf disease recognition, facilitating the creation of compact and practical models. For example, Ghofrani et al. (Ghofrani and Mahdian Toroghi, 2022) introduced a client-server system based on the knowledge distillation technique to enhance the client's accuracy. In their approach, a robust deep convolutional neural network (CNN) (Kim, 2014) architecture was responsible for classifying diseases on the server side. In contrast, on the client side, a

lightweight deep CNN architecture facilitated classification on mobile devices.

Musa et al. (Musa et al., 2022) explored efficient mobile solutions for AI in agriculture, implementing a low-power DNN for plant disease diagnosis using knowledge distillation. Their teacher and student models are based on the CNN architecture, featuring around 52M parameters for the teacher model and around 1.2M for the student. The system was deployed on a Raspberry Pi device, achieving a 90% power reduction, with a maximum of 6.22W, and a top performance of plant disease detection of 99.4% classification accuracy. Likewise, Huang et al. (Huang et al., 2023) tackled the issue of creating small plant disease classification systems based on knowledge distillation, applicable across various crops. Their solution, called multistage knowledge distillation, upholds performance and reduces the model size. Furthermore, the authors underscored the possibility of extending this technique to diverse computer vision tasks, such as image classification and segmentation, enabling the development of automated plant disease detection models with broader utility in precision agriculture.

3 DATASET

In this paper, we employ a leaf disease dataset (Geetharamani and Pandian, 2019) consisting of images of the leaves of different plants such as apple, cherry, corn, grape, pepper bell, and potato (38 different classes). Moreover, the dataset introduces an extra "false" class, covering several landmark photos without leaves.

The dataset contains 55,448 samples with RGB, 256x256 pixels each. As shown in Figure 1, the dataset is highly imbalanced, with sizes from 100 to over 5,500 per class, thus imposing a classification challenge due to various leaf disease types. The classes with the most significant training footprint are Orange with Huanglongbing, Healthy soybean, and Tomato with yellow leaf curl virus. We expect a better classification accuracy for these categories. In contrast, classes such as Healthy potato, Healthy raspberry, and Apple with cedar apple rust will most likely be unfavoured classification-wise.

4 METHODOLOGY

4.1 Model

The architecture consists of a regular CNN model as a feature extractor built based on different filters with (3,
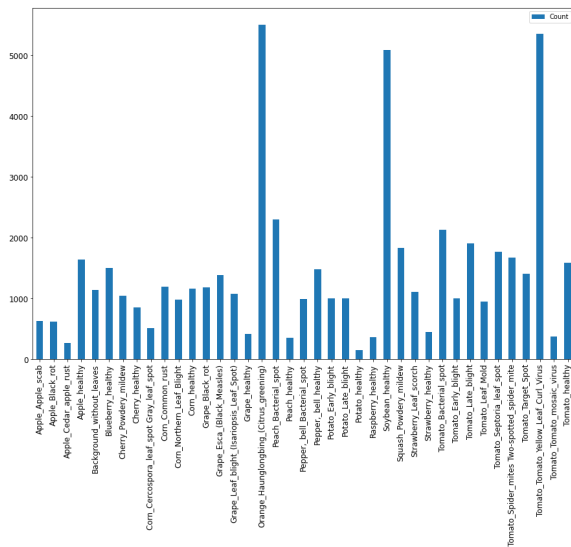
Figure 1: Class distribution of the leaf disease dataset.

32), (32, 64), (64, 128), (128, 256) features with ReLU activations and max-pooling, as well as a classifier represented by two fully-connected layers with 128 inputs to 39 outputs, that is, the number of classes for our dataset. The optimizer is Adam (Kingma and Ba, 2014), whose learning rate is 5e-4 with linear decay, while the loss function is the cross-entropy.

4.2 Adversarial Training and Attack

Regarding adversarial attacks and training, several algorithms were introduced in our experiments (see Table 1). We use these gradient-based algorithms to test the performance of the CNN model under several adversarial configurations. The experiments performed in this paper include regular and adversarial training and testing adversarial attack success analysis (Su et al., 2018). For the classification experiments, we introduce four scenarios: testing against a model trained only on regular data, using adversarial test examples for a model, and checking performance with adversarial training using either regular or adversarial test examples.

4.3 Model Explainability

Introduced by Simonyan et al. (Simonyan et al., 2014), the saliency explanation mechanism is an attribution method based on gradients. In this method, each gradient denotes the impact of a change in each input dimension on predictions within a small neighborhood around the input. As a result, the technique calculates a class-level saliency map for an image based on the gradient of an output neuron concerning the input. This map highlights the areas in the input image,

representing the focus regions for the specified class.

Guided backpropagation (Springenberg et al., 2015), alternatively referred to as guided saliency, employs a novel deconvolution variant (Zeiler and Fergus, 2014) to visualize features learned by various DNN architectures. When applied to small images, using max-pooling in CNNs becomes questionable. Replacing the max-pooling layers with a convolutional layer featuring an increased stride demonstrates no loss of performance on several image processing tasks, underscoring the versatility of guided backpropagation in challenging the conventional use of max-pooling within CNNs for specific image sizes.

Gradient-weighted class activation mapping (Grad-CAM), introduced by Selvaraju et al. (Selvaraju et al., 2017), addresses a fundamental limitation of class activation mapping (CAM) (Zhou et al., 2015) by offering visual explanations for any CNN, irrespective of its architecture. This gradient-based method utilizes class-specific gradient information from the convolutional layer of a CNN to generate a localization map, highlighting the visual attention regions in an image for classification and enhancing the transparency of CNN-based models. Despite its advancements over CAM, GradCAM has certain limitations, such as not being able to localize multiple occurrences of an entity in an image effectively (Chattopadhay et al., 2018).

We interpret the visual performance of the models trained in different setups with visualization methods such as t-SNE (Hinton and Roweis, 2002) for highlighting the classifier capabilities through two-dimensional clustering, GradCAM, Guided GradCAM (Selvaraju et al., 2017), and Guided GradCAM HiRes (Draelos and Carin, 2021). Thus, we also determine the saliency maps linked to the output of the feature extractor.

4.4 Knowledge Distillation

A practical method for compressing models is knowledge distillation, which facilitates information transfer from a large teacher network to a small student network. Initially introduced by Bucila et al. (Buciluă et al., 2006) and later generalized by Hinton et al. (Hinton et al., 2015), this method has gained popularity across multiple machine learning applications. Unlike conventional training, knowledge distillation involves learning the student network to emulate the outputs of the teacher model, typically represented as probability distributions over classes. By minimizing the similarity loss between the student's predictions and the teacher's probabilities, the student network can assimilate the knowledge from the teacher, resulting in superior performance compared to training from

Table 1: Adversarial algorithms used for model defense and attack experiments.

Algorithm	Formulas
FGSM (Goodfellow et al., 2015)	$x' = x + \varepsilon \cdot \text{sgn}\left(\nabla_x \ell(f(x), y)\right)$
RFGSM (Tramèr et al., 2018)	$x'_0 = x + \alpha \cdot \text{sgn}\left(\mathcal{N}(\mathbf{0}^n, \mathbf{I}^n)\right)$ $x'_{t+1} = \text{clip}_{(x,\varepsilon)}\left\{ x'_t + (\varepsilon - \alpha) \cdot \text{sgn}\left(\nabla_{x'_t} \ell\left(f(x'_t), y\right)\right) \right\}$
FFGSM (Wong et al., 2020)	$x'_0 = x + \mathcal{U}(-\varepsilon, \varepsilon)$ $x' = \Pi_{\mathcal{B}(x,\varepsilon)}\left\{ x'_0 + \alpha \cdot \text{sgn}\left(\nabla_{x'_0} \ell\left(f(x'_0), y\right)\right) \right\}$
MIFGSM (Dong et al., 2017)	$g_{t+1} = \mu \cdot g_t + \dfrac{\nabla_{x'_t} \ell(f(x'_t), y)}{\left\| \nabla_{x'_t} \ell(f(x'_t), y) \right\|_1}$ $x'_{t+1} = \Pi_{\mathcal{B}(x,\varepsilon)}\left\{ x'_t + \alpha \cdot \text{sgn}\left(g_{t+1}\right) \right\}$
BIM (Kurakin et al., 2017)	$x'_{t+1} = \text{clip}_{(x,\varepsilon)}\left\{ x'_t + \alpha \cdot \text{sgn}\left(\nabla_{x'_t} \ell\left(f(x'_t), y\right)\right) \right\}$
PGD (Madry et al., 2018)	$x'_0 = x + \mathcal{U}(-\varepsilon, \varepsilon)$ $x'_{t+1} = \Pi_{\mathcal{B}(x,\varepsilon)}\left\{ x'_t + \alpha \cdot \text{sgn}\left(\nabla_{x'_t} \ell\left(f(x'_t), y\right)\right) \right\}$
TPGD (Zhang et al., 2019)	$x'_0 = x + 0.001 \cdot \mathcal{N}(\mathbf{0}^n, \mathbf{I}^n)$ $x'_{t+1} = \Pi_{\mathcal{B}(x,\varepsilon)}\left\{ x'_t + \alpha \cdot \text{sgn}\left(\nabla_{x'_t} \ell_{KL}\left(f_\theta(x), f_\theta(x')\right)\right) \right\}$
EOTPGD (Liu et al., 2018)	$x'_{t+1} = \Pi_{\mathcal{B}(x,\varepsilon)}\left\{ x'_t + \alpha \cdot \text{sgn}\left(\mathbb{E}\left[\nabla_{x'_t} \ell\left(f(x'_t), y\right)\right]\right) \right\}$

scratch without a teacher (Avram et al., 2022).

The architecture of our student consists of multiple stacked convolutional layers with interleaved pooling layers and ReLU activation functions. The classification head of the student model comprises a fully-connected layer followed by a dropout layer with a dropout rate of 50%. The utilization of dropout aids in enhancing the generalization capabilities of the model. The Adam optimizer is adopted for conducting the knowledge distillation experiments. Three teacher models were selected for our goals: ResNet50, DenseNet121 (Huang et al., 2017), and Xception (Chollet, 2017). Additionally, an ensemble approach is employed to leverage the knowledge acquired by these pre-trained teachers on the ImageNet benchmark (Deng et al., 2009). Before beginning the effective distillation process, each teacher model is fine-tuned on the leaf disease classification task.

For the distillation process, the hyperparameters are crucial in controlling the knowledge transfer from the teacher models to the student model. Their values are chosen as follows: the alpha parameter is set to 0.1, and the temperature parameter is set to 3.

4.5 Performance Metrics

We use accuracy and macro-F1 as performance metrics for adversarial training and attack experiments. Knowledge distillation experiments are evaluated using accuracy and macro-F1 for classification, the number of model parameters for model size comparison, and floating-point operations per second (FLOPs) for model efficiency evaluation.

4.6 Research Questions

[RQ1] Is robustness the cost of accuracy?
[RQ2] How is the classification affected when we use adversarial examples?
[RQ3] Does the leaf shape matter?
[RQ4] How do adversarial attacks change focus?
[RQ5] Is knowledge distillation effective for leaf disease classification?

5 RESULTS

5.1 Results for RQ1

Table 2 shows outstanding performance for the regular scenario, without adversarial training and adversarial attack, with 97.78% accuracy. In contrast, when we attack the network, the actual label of adversarial examples is hardly identified, thus a lower accuracy ranging from 0.78% to 18.58%.

When we train the network with adversarial examples, we observe the trade-off between robustness and accuracy (Su et al., 2018). Table 3 shows that for a network trained on adversarial examples and tested on the initial data, we determine lower accuracy for all the

models, ranging from 77.17% to 94.70%. In the attack scenario for adversarially trained models, we gain a robustness improvement for all algorithms. BIM is the best adversarial algorithm in both types of tests, with 94.70% accuracy for initial test data and 83.46% for adversarial test data.

Table 2: Accuracy (Acc) and macro-F1 (F1) for our CNN model without adversarial training.

Attack	Algorithm	Acc (%)	F1 (%)
No	-	**97.75**	**97.03**
Yes	FGSM	5.64	4.69
	RFGSM	0.78	1.17
	FFGSM	6.02	4.99
	MIFGSM	0.11	0.17
	BIM	10.95	9.24
	PGD	0.51	0.76
	TPGD	**18.58**	**17.81**
	EOTPGD	0.81	1.20

Table 3: Accuracy (Acc) and macro-F1 (F1) for our CNN model with adversarial training.

Attack	Algorithm	Acc (%)	F1 (%)
No	FGSM	79.98	70.06
	RFGSM	83.99	75.52
	FFGSM	83.46	74.89
	MIFGSM	77.17	66.84
	BIM	**94.70**	**91.86**
	PGD	83.52	74.09
	TPGD	83.37	77.36
	EOTPGD	84.36	76.53
Yes	FGSM	56.12	41.93
	RFGSM	60.79	47.53
	FFGSM	61.06	47.32
	MIFGSM	51.29	36.92
	BIM	**83.46**	**76.58**
	PGD	51.31	39.79
	TPGD	64.76	57.78
	EOTPGD	60.99	47.93

5.2 Results for RQ2

Figure 2 depicts the classification performance for the models as a 2D projection. We see a clear separation for the model trained and tested on initial data. Moreover, in contrast with the other adversarial models, the t-SNE plot also highlights improved performance for the BIM algorithm. Classes Orange with Huanglong-

bing (brown), Healthy soybean (lavender), and Tomato with yellow leaf curl virus (blue) have the most considerable volume in this dataset. Despite the correct classification, we can observe that their distribution is highly affected by adversarial training, even in the case of BIM, where the Tomato with yellow leaf curl virus is split into two separate clusters. We can still determine a slight separation for this class in regular training and testing scenarios. Thus, we conclude that adversarial training significantly affects class distributions.

5.3 Results for RQ3

Figures 3, 4, 5, and 6 support an analysis focused on the classification of one image in all adversarial scenarios, providing a view on the saliency map, the test image either regular or adversarial, as well as the superimposition of them for better visualization. An initial observation is that maps for adversarial examples often describe the shape of the leaves. However, in addition to highlighting the shape and potential focus points, they feature additional heat in the regions outside the leaves due to the introduced noise. Figure 4 depicts an example where we get four misclassifications out of nine despite the similarities in shapes.

Figure 6a attempts to explain the Strawberry with leaf scorch classification to the expected Squash with powdery mildew. The shapes of the saliency map match in significant proportion; therefore, the round features might have determined the confusion. This impact is even more evident in Figure 5, with only three of nine correctly classified examples. A Peach with bacterial spot example is shown in Figure 6b, featuring a similar shape and orientation to the Tomato with septoria leaf spot. We conclude that the shape and orientation of the leaves in the dataset can be prone to misclassifications when they are subject to adversarial attacks.

5.4 Results for RQ4

The analysis of the model's cognitive processes during classification encompasses a comprehensive exploration of three distinct visualization techniques (i.e., GradCAM, Guided GradCAM, and Guided GradCAM HiRes), each designed to shed light on the intricate mechanisms governing its decision-making. Figure 7 presents these visualization methods aimed at elucidating the underlying reasoning employed by the model when faced with a given instance for classification.

In particular, the second row of visualizations offers valuable insights into the model's attentional patterns when confronted with an un-attacked image. This
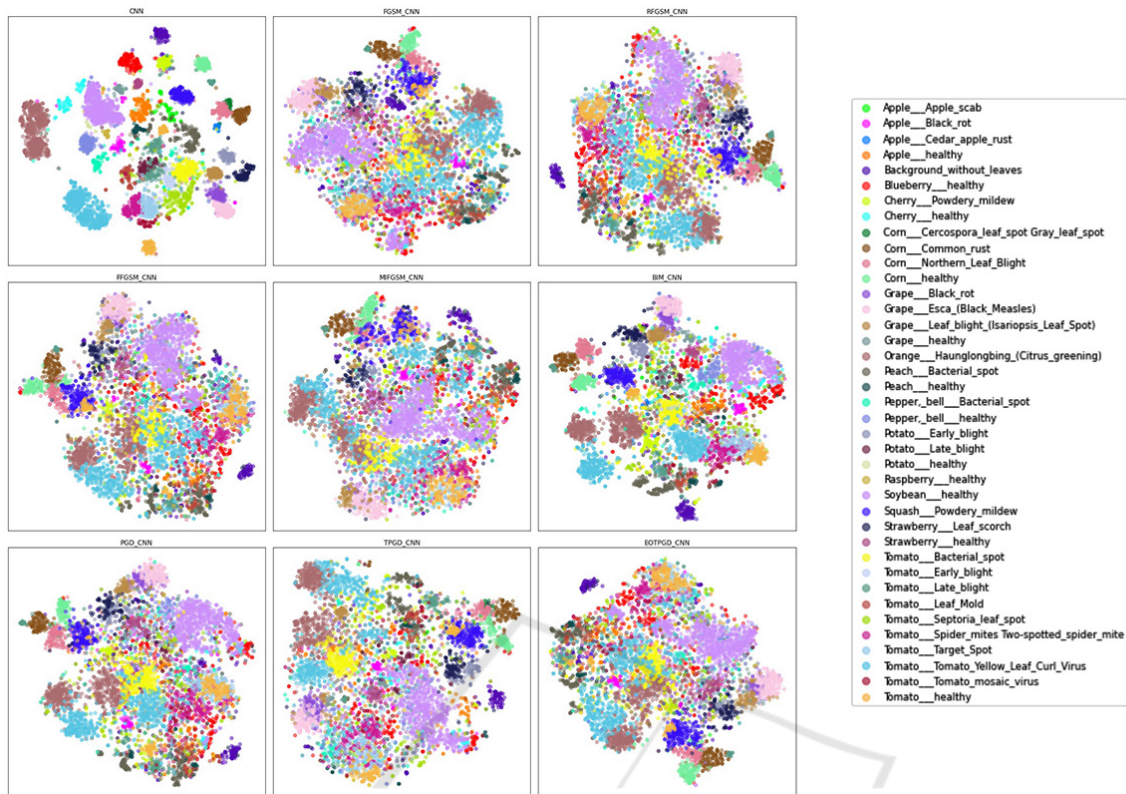
Figure 2: Class t-SNE distribution for the models trained with adversarial training. The first subfigure, labeled "CNN", depicts the classifications of the model trained without adversarial examples, while subfigures 2-9 show the classification clustering for the models using the FGSM, RFGSM, FFGSM, MIFGSM, BIM, PGD, TPGD, and EOTPGD algorithms to create more training data.

investigation unveils crucial information regarding the areas of interest within the input image that the model prioritizes when making its predictions.

Furthermore, the last row of visualizations is a pivotal component of this analysis, seeking to reveal the precise focal points that capture the model's attention during the classification process. In this example, the FGSM attack demonstrates its efficiency by skillfully diverting the model's focus away from significant points within the leaf where the disease is present.

The ability of the FGSM attack to redirect the model's focus towards less critical regions of the image highlights potential vulnerabilities in the model's robustness and susceptibility to adversarial perturbations. Such insights are essential contributions to model explainability and adversarial machine learning, facilitating the development of more robust and dependable AI systems for critical applications.

5.5 Results for RQ5

Before Knowledge Distillation. Table 4 presents the performance of the models before the knowledge dis-

tillation process. The student model achieved an accuracy of 85.38% and a macro-F1 score of 79.17%. In comparison, the teacher models, namely ResNet50, DenseNet121, Xception, and Ensemble, displayed significantly higher accuracies and macro-F1 scores, ranging from 97.16% to 99.04%.

After Knowledge Distillation. Table 5 depicts the results after the knowledge distillation process. The student model demonstrates accuracy and macro-F1 score improvements when distilled with each teacher model. When distilled with ResNet50, the student model achieved an accuracy of 89.43% (+4.05) and a macro-F1 score of 85.09% (+5.92). Similarly, distillation with DenseNet121 and Xception resulted in accuracies of 90.11% (+4.73) and 88.88% (+3.50), as well as macro-F1 scores of 85.23% (+6.06) and 83.42% (+4.25). Furthermore, distillation with the Ensemble model led to an accuracy of 89.72% (+4.34) and a macro-F1 score of 84.95% (+5.78).

Number of Parameters. Table 6 outlines the number of parameters for each model, both the student and teacher models. The student model contains only 3.7M parameters. In comparison, the teacher models have significantly more parameters, ranging from 7M
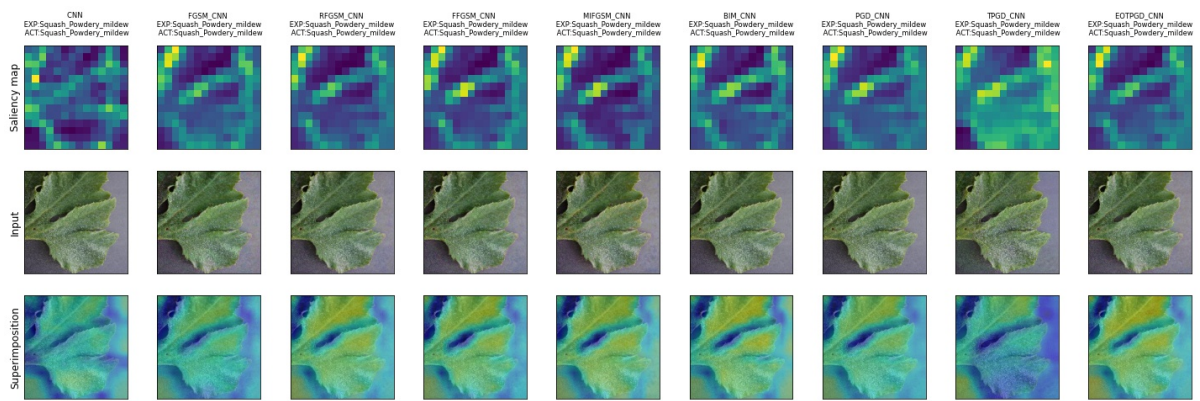
Figure 3: Squash GradCAM saliency maps without adversarial attack.
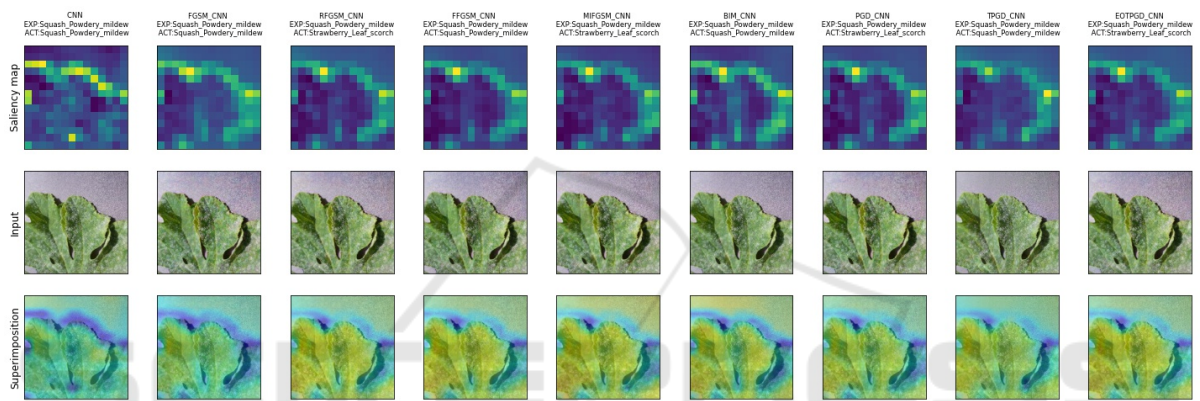

Figure 4: Squash GradCAM saliency maps with adversarial attack.
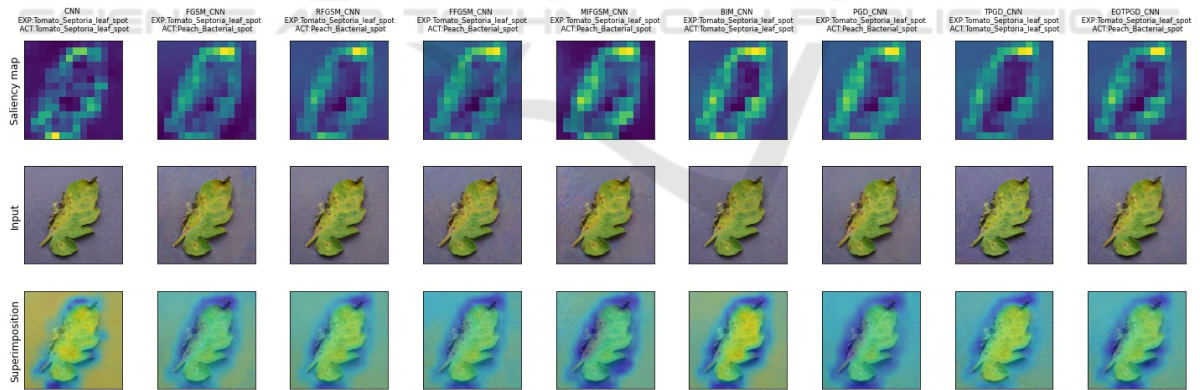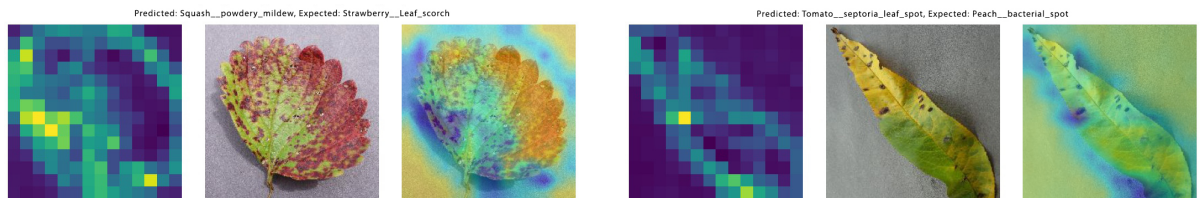

Figure 5: Tomato GradCAM saliency maps with adversarial attack.

(a) Strawberry classification GradCAM.

(b) Peach classification GradCAM.

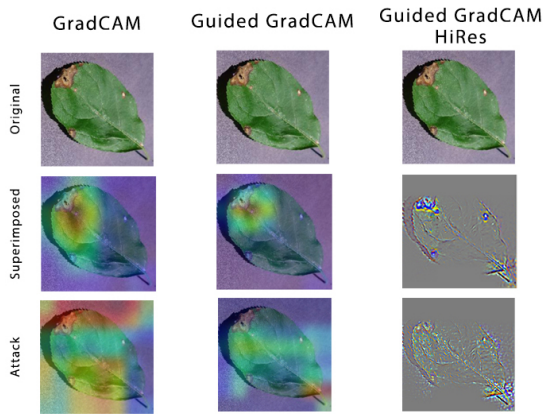Figure 6: Misclassification GradCAM for different leaf diseases.

Figure 7: Model's attentional focus after the FGSM attack.

for DenseNet121 to 51.6M for the Ensemble (comprising the parameters from ResNet50, DenseNet121, and Xception). Despite the knowledge transfer from the teacher models during distillation, the student model maintains a substantially lower number of parameters than the teacher models. Surprisingly, the Ensemble does not obtain the best enhancement despite the notable improvements achieved through the knowledge distillation process with various teacher models.

Model Computational Efficiency. The computational efficiency of the models, as indicated by the FLOPs for each one, is shown in Table 7. The student model achieves 0.4B FLOPs. In contrast, the teacher models perform significantly more FLOPs, ranging from 7.4B for DenseNet121 to 11.9B for Xception. Even with the knowledge transfer from the teacher models during the distillation process, the student model retains its computational efficiency by operating at fewer FLOPs than the teacher models.

The results demonstrate the effectiveness of the knowledge distillation process, where the student model exhibits improved performance while acquiring knowledge from the teacher models. Moreover, the student model achieves these performance gains while keeping its parameters and computational complexity low, making it an efficient and effective solution for the leaf disease classification.

Student's Performance. To gain insights into the student's learning progress from the teacher, we utilize t-SNE visualizations for both the teacher and the student after the knowledge distillation process. The 2D visualizations, showcased in Figure 8, allow us to compare their abilities in understanding the data and clustering the leaves effectively.

The teacher demonstrates a higher capacity to comprehend the data and adeptly cluster the leaves. On the other hand, the student's performance is commendable, having learned to capture a substantial amount

of knowledge from the teacher, enabling it to group the leaves accurately. However, due to the limited capacity, the student overlaps some clusters slightly.
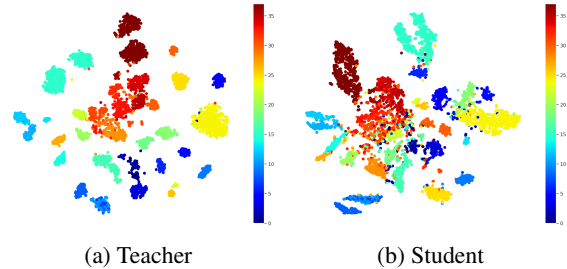

(a) Teacher   (b) Student

Figure 8: t-SNE visualizations after knowledge distillation.

Table 4: Performance before knowledge distillation.

Model	Accuracy (%)	Macro-F1 (%)
Student	85.38	79.17
ResNet50	**99.04**	**98.50**
DenseNet121	98.55	97.99
Xception	97.16	96.34
Ensemble	98.73	98.11

Table 5: Student's accuracy (Acc) and macro-F1 (F1) after knowledge distillation.

Teacher	Acc (%)	Δ Acc	F1 (%)	Δ F1
ResNet50	89.43	+4.05	85.09	+5.92
DenseNet121	**90.11**	**+4.73**	**85.23**	**+6.06**
Xception	88.88	+3.50	83.42	+4.25
Ensemble	89.72	+4.34	84.95	+5.78

Table 6: Comparison of student size with teacher models. The number of parameters is written in millions. The Ensemble model parameters are represented as the sum of the ResNet50, DenseNet121, and Xception number of parameters.

Model	#Params	% of Teacher Params
Student	**3.71**	-
ResNet50	23.66	15.68
DenseNet121	7.06	52.54
Xception	20.93	17.72
Ensemble	51.65	7.18

6 CONCLUSIONS

In this work, we addressed three vital pillars in machine learning: adversarial attacks, knowledge distillation, and explainability, explicitly focusing on plant

Table 7: Comparison of student computational efficiency with teacher models. The FLOPs values are written in gigabytes. The Ensemble model FLOPs is the maximum value of the ResNet50, DenseNet121, and Xception FLOPs.

Model	FLOPs	% of Teacher FLOPs
Student	**0.48**	-
ResNet50	10.09	4.75
DenseNet121	7.45	6.44
Xception	11.90	4.03
Ensemble	11.90	4.03

leaf disease classification. Thus, we explored adversarial attacks and their impact on the robustness of CNNs. Through rigorous experimentation, we successfully employed adversarial training to enhance the resilience of the CNNs against adversarial perturbations, reinforcing their ability to perform accurate leaf disease classification.

In machine learning explainability, we recognized the critical importance of understanding how models arrive at their decisions, especially in the context of leaf disease classification. Additionally, we ventured into the domain of model compression, aiming to retain the performance of the leaf disease classification models while reducing their computational complexity. Our findings demonstrated promising results, paving the way for deploying efficient and resource-friendly models, particularly crucial for applications with limited computing resources.

The following steps in our experiments include how successful adversarial attacks of an adversarial algorithm are on a model previously trained on examples generated with a different one. The quest for a better understanding of the inner workings of knowledge distillation and its interplay with other regularization techniques also opens up intriguing directions for future exploration.

ACKNOWLEDGEMENTS

REFERENCES

Avram, A.-M., Catrina, D., Cercel, D.-C., Dascalu, M., Rebedea, T., Păiș, V., and Tufiș, D. (2022). Distilling the knowledge of romanian berts using multiple teachers. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 374–384.

Buciluă, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

Chattopadhay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. (2018). Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision*, pages 839–847.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258.

Christian Meske, Enrico Bunde, J. S. and Gersch, M. (2022). Explainable artificial intelligence: Objectives, stakeholders, and future research opportunities. *Information Systems Management*, 39(1):53–63.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Dong, Y., Liao, F., Pang, T., Hu, X., and Zhu, J. (2017). Discovering adversarial examples with momentum. *CoRR*, abs/1710.06081.

Draelos, R. L. and Carin, L. (2021). Use hirescam instead of grad-cam for faithful explanations of convolutional neural networks.

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

Ghofrani, A. and Mahdian Toroghi, R. (2022). Knowledge distillation in plant disease recognition. *Neural Computing and Applications*, 34(17):14287–14296.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pages 770–778. IEEE.

Hinton, G. E. and Roweis, S. (2002). Stochastic neighbor embedding. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press.

Hinton, G. E., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Huang, Q., Wu, X., Wang, Q., Dong, X., Qin, Y., Wu, X., Gao, Y., and Hao, G. (2023). Knowledge distillation facilitates the lightweight and efficient plant diseases detection model. *Plant Phenomics*, 5:0062.

Khan, A., Vibhute, A. D., Mali, S., and Patil, C. (2022). A systematic review on hyperspectral imaging technology with a machine and deep learning methodology for agricultural applications. *Ecological Informatics*, 69:101678.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kurakin, A., Goodfellow, I. J., and Bengio, S. (2017). Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Linardatos, P., Papastefanopoulos, V., and Kotsiantis, S. (2021). Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1).

Liu, X., Li, Y., Wu, C., and Hsieh, C.-J. (2018). Adv-bnn: Improved adversarial defense through robust bayesian neural network. In *International Conference on Learning Representations*.

Lungu-Stan, V.-C., Cercel, D.-C., and Pop, F. (2023). Skindistilvit: Lightweight vision transformer for skin lesion classification. In *International Conference on Artificial Neural Networks*, pages 268–280. Springer.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Miyato, T., Dai, A. M., and Goodfellow, I. (2016). Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Musa, A., Hassan, M., Hamada, M., and Aliyu, F. (2022). Low-power deep learning model for plant disease detection for smart-hydroponics using knowledge distillation techniques. *Journal of Low Power Electronics and Applications*, 12(2).

Năstăsescu, G.-S. and Cercel, D.-C. (2022). Conditional wasserstein gan for energy load forecasting in large buildings. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Ramirez-Moreno, D., Schwartz, O., and Ramirez-Villegas, J. (2013). A saliency-based bottom-up visual attention model for dynamic scenes analysis. *Biological cybernetics*, 107.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Visualising image classification models and saliency maps. *Deep Inside Convolutional Networks*, 2.

Smădu, R.-A., Dinică, I.-r., Avram, A.-M., Cercel, D.-C., Pop, F., and Cercel, M.-C. (2022). Legal named entity recognition with multi-task domain adaptation. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 305–321.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. A. (2015). Striving for simplicity: The all convolutional net. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.

Su, D., Zhang, H., Chen, H., Yi, J., Chen, P.-Y., and Gao, Y. (2018). Is robustness the cost of accuracy? – a comprehensive study on the robustness of 18 deep image classification models. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, pages 644–661, Cham. Springer International Publishing.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2014). Intriguing properties of neural networks. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I. J., Boneh, D., and McDaniel, P. D. (2018). Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Wong, E., Rice, L., and Kolter, J. Z. (2020). Fast is better than free: Revisiting adversarial training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

You, H., Lu, Y., and Tang, H. (2023). Plant disease classification and adversarial attack using simam-efficientnet and gp-mi-fgsm. *Sustainability*, 15(2).

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In Fleet, D. J., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. (2019). Theoretically principled trade-off between robustness and accuracy. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR.

Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., and Torralba, A. (2015). Learning deep features for discriminative localization. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929.