Visual Plant Disease Detection using Deep Learning, Machine learning and eXplainable AI Techniques

Orestis Kotrotsios, Prodromos Kampouridis
Dept. of Digital Systems, University of Piraeus
Institute of Informatics and Telecommunications, National Center for Scientific Research "DEMOKRITOS"
Athens, Greece

30 June 2023

ABSTRACT

In this paper, we investigate the potential of Artificial Intelligence to enhance identifications of plant diseases using Computer Vision and eXplainable AI (XAI). This project is the key component of the Multimodal Machine Learning course curriculum and is part of the Master's program at National Center for Scientific Research "Demokritos" and the University of Piraeus. Our approach involves comparing the results of Machine Learning and Deep Learning models. Additionally, we extract features from the plants using techniques such as Haralick, Gabor, shape and, histogram features which are used to train our Machine Learning models for classification. Moreover, by leveraging the power of XAI we aim to provide transparent and interpretable results to improve decision making in this task. Our results demonstrate the effectiveness of our approach in accurately detecting plant diseases and providing interpretable results.

KEYWORDS

Deep Learning; Machine learning; Computer Vision; Image Classification; Object Detection; Agriculture; Plant Diseases; Explainable AI (XAI); EfficientNet; ResNet50; CNN; SHAP; Grad-CAM;

1 INTRODUCTION

The ability to accurately identify plant diseases is crucial since plants are the fundamental components of our ecosystem. Plants not only provide us with oxygen, food and shelter but also add to the beauty of our surroundings everywhere especially in the urban areas. However, like all living beings, plants are vulnerable to diseases caused by various pathogens. These diseases can have irreversible impacts on both natural and agricultural ecosystems and due to the numerous characteristics and subtle variations among crops it can be a challenge to classify them correctly. Advances in technology such as Deep Learning can play a significant role in addressing this issue. In addition, accurate plant disease classification can also help with understanding health issues and behavior traits related to specific plants.

2 DATASET

In this study, we used the New Plant Diseases Dataset for our computer vision task. The dataset consists of approximately 54.305 images of both healthy and inflected leaves of crop plants, divided into 38 distinct classes. Specifically, this dataset has been generated using offline augmentation techniques from the original PlantVillage Dataset. What's more, we created a new segmented dataset by applying a mask to existing images to remove the background and then we merged the two datasets together.

Figure 1: Sample images from the New Plant Diseases dataset

3 MACHINE LEARNING APPROACH

In this chapter we are using machine learning algorithms to classify the images of plants according to their respective disease.

3.1 FEATURE EXTRACTION

The first step in a Machine Learning image classification task is to extract the features from the images and create a table with all these features, the samples, and the true label of each sample, as well.

The features we decided to extract for our classification model include Haralick texture features, Gabor features, Shape features, Color features and Histogram features.

Firstly, the haralick texture features are calculated using the gray-level co-occurrence matrix (GLCM) of an image. The GLCM describes how often different combinations of pixel intensities occur in an image. We used the mahotas library to calculate 13 different statistics based on the GLCM, including measures of contrast, correlation, energy, and homogeneity.

Furthermore, the Gabor features are calculated by applying Gabor filters to an image. These filters are bandpass filters that can be detect edges and textures in an image at different scales and orientations. With different combinations of these filters, we produced a set of 32 Gabor features.

Moreover, Shape features describe the shape of the plant leaf in an image. We calculated the area, perimeter, and centroid coordinates of the largest contour.

In addition, Color features describe the color distribution of an image and we calculated the mean and standard deviation of pixel values in each color channel for a total of 6 features.

Lastly, Histogram features describe the distribution of pixel values in an image and we calculated histograms for each color channel and concatenated them to produce a single feature vector.

3.2 DATA EXPLORATION

As mentioned in the Section 2 earlier, our dataset contains 38 classes of healthy and unhealthy leaves from various plants. With a total of 54.305 samples, the extracted features resulted in a dataset shape of 54.305x564, including the label column. For our machine learning algorithms, we utilized only the colored pictures without applying any image augmentation or any preprocessing techniques. Instead, we focused on preprocessing the extracted features, as described in the next section.

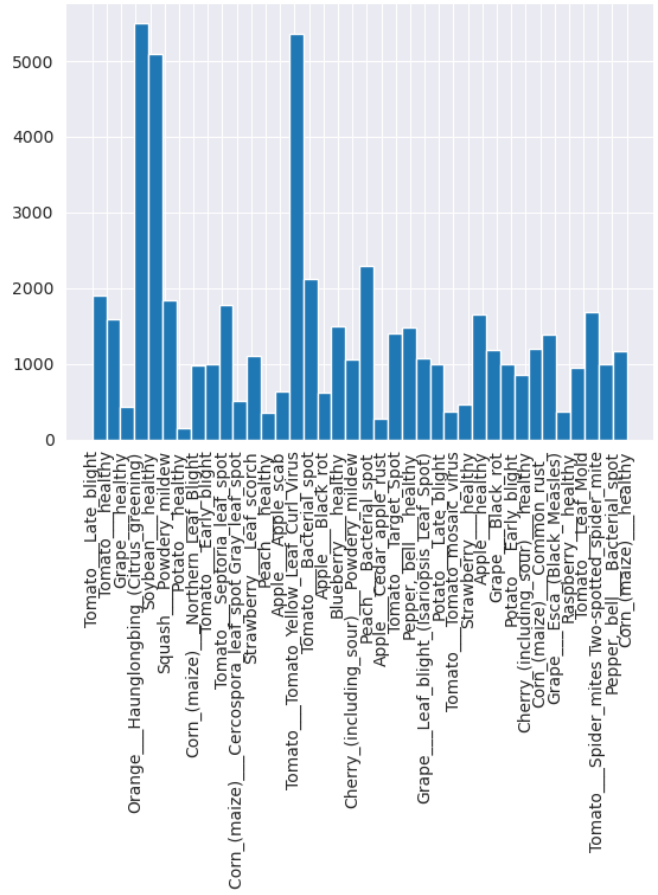

Figure 2: The distribution of the dataset

As shown in Figure 2, the dataset is imbalanced with some classes having approximately 5000 samples while others have only 200. To address this issue, we used synthetic data, which we will discuss in detail.

3.3 DATA PREPROCESSING

Before training the machine learning algorithms, we preprocessed the data to balance it.

To ensure that our preprocessing was not influenced by the test data and that the test data remains independent from the training data, we first split our dataset into separate training and test sets After splitting the data into separate training and test sets, we proceeded with preprocessing. Our initial step was to scale the data to ensure that all features contributed equally to the final result, despite having different value ranges. Next, we addressed the issue of class imbalance by using SMOTE to generate synthetic samples for the minority classes, thereby bringing their sample counts closer to that of the majority class. Then, we applied undersampling to reduce the number of samples to 1000. The main reason was to also reduce the number of synthetic samples and mitigate potential bias in the model from an over representation of synthetic images in some classes.

	Before SMOTE	After SMOTE	UNDERSAMPLING
Orange___Haunglongbing_(Citrus_greening)	4390	4390	1000
Tomato___Tomato_Yellow_Leaf_Curl_Virus	4318	4390	1000
Soybean___healthy	4064	4390	1000
Peach___Bacterial_spot	1870	4390	1000
Tomato___Bacterial_spot	1705	4390	1000
Tomato___Late_blight	1534	4390	1000
Squash___Powdery_mildew	1479	4390	1000
Tomato___Septoria_leaf_spot	1387	4390	1000
Tomato___Spider_mites Two-spotted_spider_mite	1344	4390	1000
Tomato___healthy	1313	4390	1000
Apple___healthy	1271	4390	1000
Blueberry___healthy	1214	4390	1000
Pepper,_bell___healthy	1199	4390	1000
Tomato___Target_Spot	1115	4390	1000
Grape___Esca_(Black_Measles)	1096	4390	1000
Grape___Black_rot	956	4390	1000
Corn_(maize)___Common_rust_	944	4390	1000
Corn_(maize)___healthy	931	4390	1000
Strawberry___Leaf_scorch	875	4390	1000
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	867	4390	1000
Cherry_(including_sour)___Powdery_mildew	842	4390	1000
Potato___Late_blight	816	4390	1000
Pepper,_bell___Bacterial_spot	800	4390	1000
Tomato___Early_blight	793	4390	1000
Corn_(maize)___Northern_Leaf_Blight	791	4390	1000
Potato___Early_blight	791	4390	1000
Tomato___Leaf_Mold	744	4390	1000
Cherry_(including_sour)___healthy	676	4390	1000
Apple___Black_rot	504	4390	1000
Apple___Apple_scab	502	4390	1000
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	399	4390	1000
Strawberry___healthy	366	4390	1000
Grape___healthy	342	4390	1000
Raspberry___healthy	293	4390	1000
Tomato___Tomato_mosaic_virus	290	4390	1000
Peach___healthy	280	4390	1000
Apple___Cedar_apple_rust	223	4390	1000
Potato___healthy	119	4390	1000

Figure 3: Data Balancing

3.4 RESULTS

Our comparison includes several machine learning modelss such as Support Vector Machine, K-Nearest Neighbors and Random Forest Classifier. We used a grid search to find the best parameters for each model and K-fold cross validation to train and evaluate them. Our training data was split into 10 parts, with one part used as a validation set and the rest for training.

We also experimented with training the models, both with and without the use of SMOTE. Moreover, we found that without SMOTE, the recall for minority classes was suboptimal. As a result, we decided to present our results using SMOTE.

The following Figure presents the classification report we produced using the model that achieved the best performance scores which it was the Random Forest. This table displays metrics such as precision, recall, f1 score, and support for each plant disease and the outcomes were quite impressive.

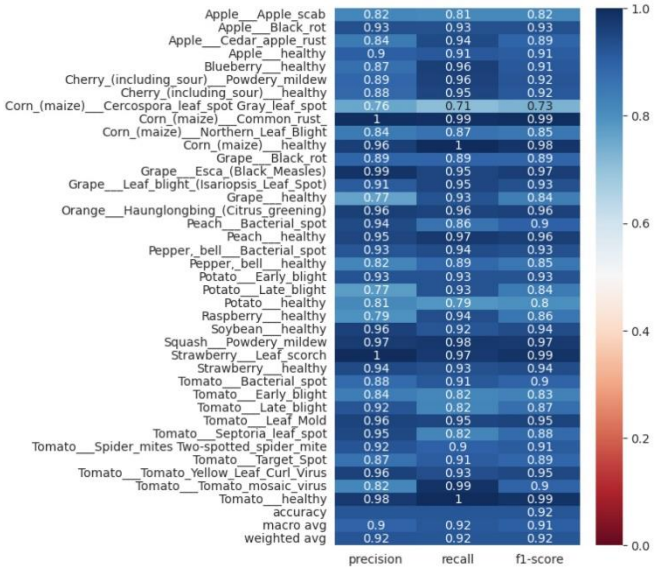

Figure 4: Classification Report of Random Forest

We also produced a confusion matrix again to better understand the model's behavior. While it struggles to accurately classify Gray Leaf SPOT disease in corns, it performs optimally for the remaining diseases. Even minority classes like the Apple Cedar rust have shown significant improvement. Without SMOTE the recall score was about 50% while after applying SMOTE it was increased to 94%.

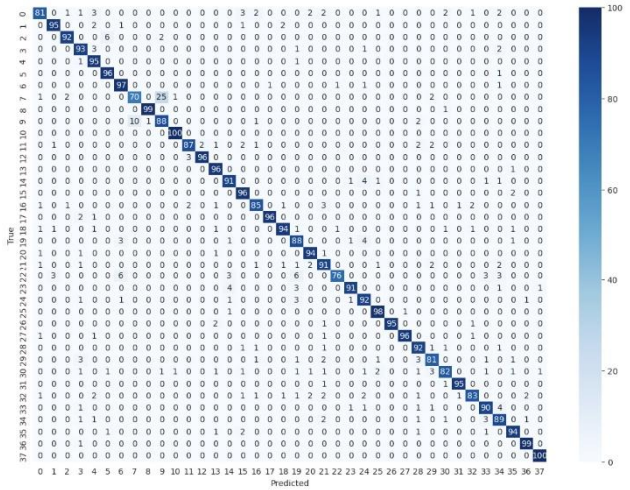

Figure 5: Confusion Matrix of Random Forest

4 DEEP LEARNING APPROACH

In our Deep Learning section, we enhanced the diversity of our training data, by implementing additional augmentation techniques such as horizontal and vertical flipping, as well as zooming in, on the images. It is worth mentioning that we trained each model twice, one time with and one time without segmentation. We tried various model structures, and our principal goal was to compare the results and determine which approach leads to the best performance in plant disease classification. All the training process was monitored using a custom callback that we additionally created.

4.1 CNN

In this subsection, we introduce a simple Convolutional Neural Network (CNN) model that we have created and trained for the task. What's more, we present the training history and evaluate the performance on the test dataset.

The model consists of an input layer followed by a BatchNormalization layer, three Convolutional layers with 32, 64, and 128 filters respectively, each followed by a Max Pooling layer. After the last Max Pooling layer, the output is flattened and passed through a Dense layer with 256 units and finally to an output

layer with 38 units, since we have 38 classes in total, and a softmax function.

Despite its simplicity the model has demonstrated impressive performance, achieving high accuracy on both validation and test sets, while maintaining relatively low losses. The validation accuracy and loss were 95.8% and 0.15, while the test accuracy and test loss were 96.2% and 0.14 respectively.

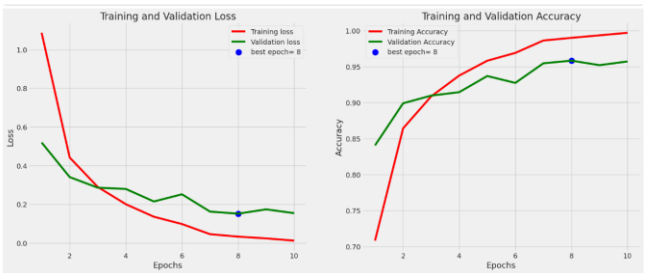

Figure 6: Accuracy and loss for Validation and Training of the CNN model

Furthermore, we generated a classification report which includes again important metrics for our task, for each plant disease. and the outcomes were quite satisfactory.

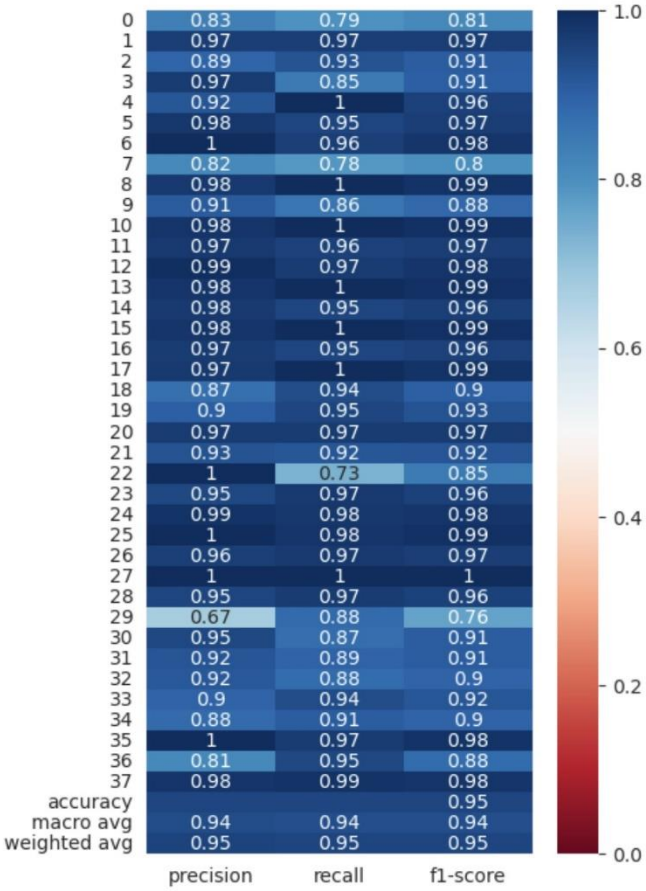

Figure 7: Classification Report of CNN

To gain further understanding of the model's performance we generated a confusion matrix. The results were satisfactory as well and as expected due to the high validation/test accuracy, and our model was not biased towards a specific class.

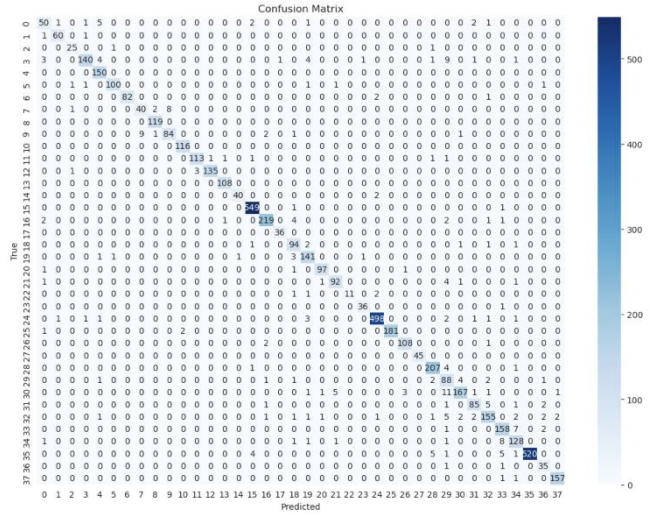

Figure 8: Confusion Matrix

4.2 ResNet50

In addition to the simple CNN model, we also explored the performance of a more complex model but without pretrained weights. We used the ResNet50 architecture as our base model and added a few additional layers. Specifically, we added a Convolutional layer, a Global Average Pooling layer, a Batch Normalization layer, two Dense layers and a Dropout layer.

The highest validation accuracy achieved was 98.6% with a loss of 0.048, while on the test it achieved an accuracy of 99.06% and a loss of 0.05 respectively.

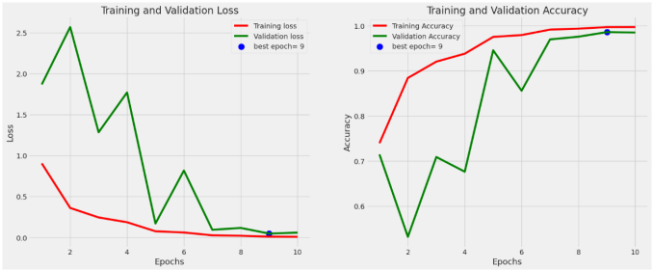

Figure 9: Accuracy and loss for Validation and Training of the ResNet50 model

After printing a classification report, we observed that the results were significantly improved, likely due to the increased number of layers in ResNet50 compared to the simple CNN.

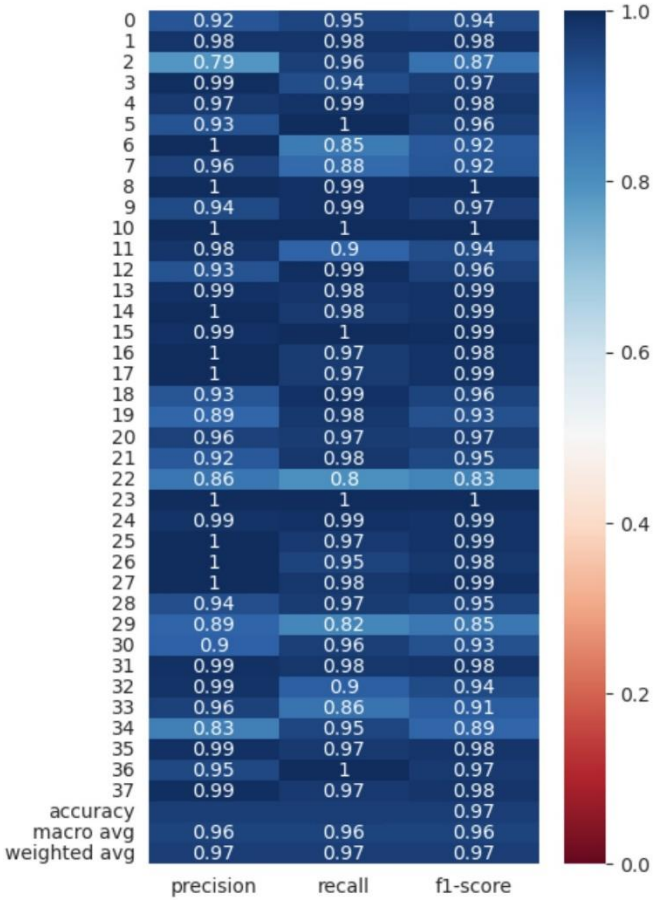

Figure 10: Classification Report of ResNet50

We also produced a confusion matrix again to better understand the model's behavior and we observed that there was no bias towards a specific class, as expected.

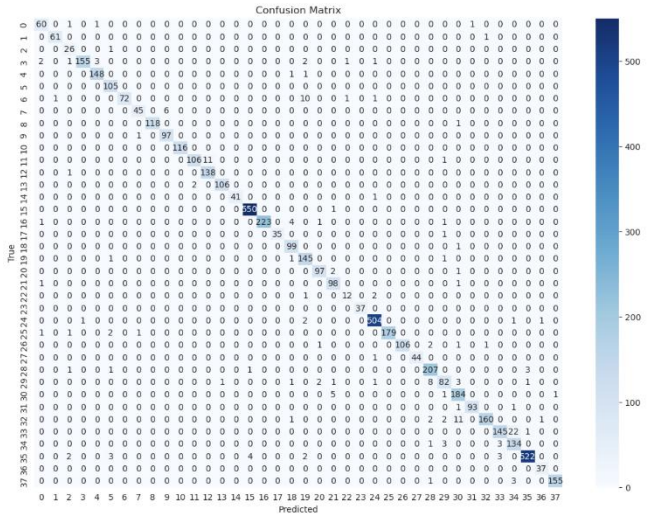

Figure 11: Confusion Matrix of ResNet50

4.3 TRANSFER LEARNING

To improve further the performance of the previous model we explored the use of Transfer Learning. Now, we are using the EfficientNetB3 architecture with the same additional layers, but this time we initialized the model with pretrained weights from ImageNet. Our goal was to compare the result with our previous models and if Transfer Learning will lead to even better performance in this task.

This approach yielded exceptional results, achieving a validation accuracy of 99.74% and a test accuracy of 99.79%. The validation and test losses were 0.129 and 0.125 respectively.

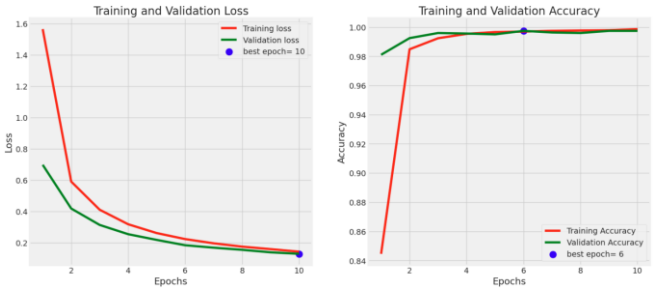

Figure 12: Accuracy and loss for Validation and Training of EfficientNetB3

As illustrated in the following classification report, the use of Transfer Learning has led to near-perfect outcomes. This can definitely be attributed to the use of pretrained weights from ImageNet which improved the performance of our model in classifying the 38 classes we had.

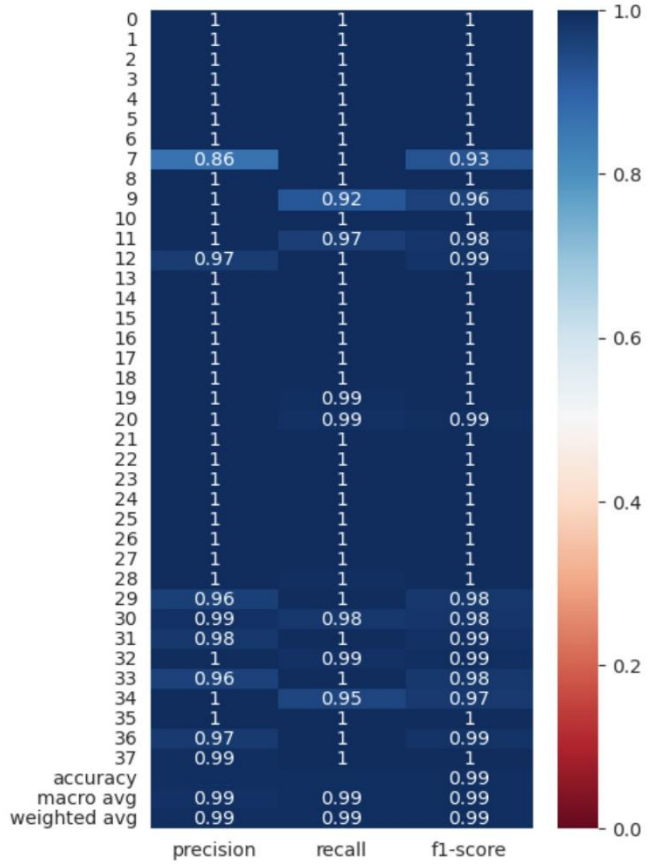

Figure 13: Classification Report of EfficientNetB3

The last confusion matrix we produced still indicates that there is no bias towards a specific class,

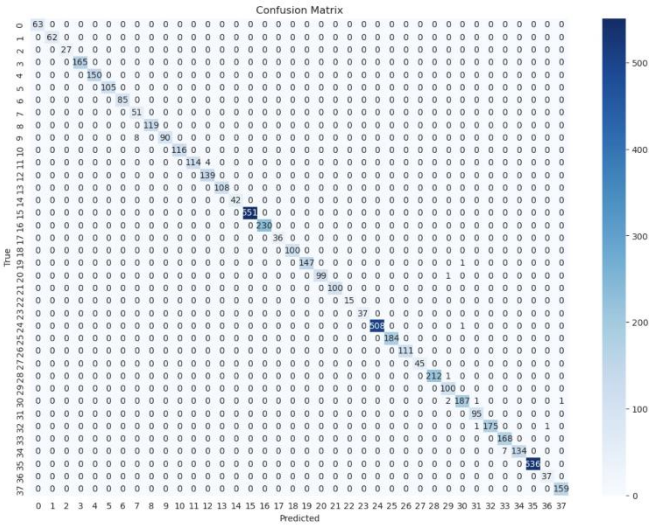

Figure 14: Confusion Matrix of EfficientNetB3

4.4 SIDE-BY-SIDE COMPARISON

Model	Validation Accuracy	Validation Loss	Test Accuracy	Test Loss
CNN	95.83%	0.151	96.28%	0.146
ResNet50	98.60%	0.048	99.06%	0.050
Efficient Net	99.74%	0.129	99.79%	0.125

Table 1: Performance comparison of 3 different models

According to the Table 1 above, EfficientNet exhibited the highest performance with a validation accuracy of 99.74% and a test accuracy of 99.79%. This superior performance can be attributed to the use of transfer learning with pretrained weights from the ImageNet.The ResNet50 model also demonstrated strong performance but with a validation and test accuracy slightly lower than EfficientNet's. Plus, it important to note that resnet had the lowest validation and test accuracy loss among the three models. In contrast, the CNN model had the lowest performance, but its results remain satisfactory.

5 AI EXPLAINABILITY

In an attempt to try and explain why the Deep Learning models made their predictions and also to evaluate if the models really learnt to recognize the important features and not just the background color or other irrelevant details, we used two different eXplainable AI (XAI) techniques, SHAP and Grad-CAM. Both techniques were useful in providing visual explanations for the decision made by our models.

5.1 SHAP

SHAP is a method for explaining the output of machine learning models. It assigns each feature an importance value for a particular prediction, providing insights into how the model arrived at its decision.

In the context of our plant diseases task, we used SHAP to explain the predictions of our neural networks by attributing importance values to the individual pixels in an image. This helped us identify which parts of the image were most influential in the model's decision-making process.

Figure 15 shows the SHAP heatmaps for 3 random images from our dataset using the three models we mentioned earlier. Each subplot contains the original image on the left and its corresponding SHAP heatmaps on the right. The colors represent the importance of each pixel, with warmer colors indicating higher importance and cooler colors indicating lower importance.

Figure 15: SHAP heatmaps for 3 random diseased plants

As we can see, our models correctly identified the diseased areas in all images, as indicated by the high importance values in those regions. For the third plant however, the CNN model seems to have focused on the shape of the grape and have missed some esca spots. Undoubtedly, the EfficientNet outperformed in performance the other two models since the diseased areas are warmer.

5.2 GRAD-CAM

After analyzing the results of our SHAP experiment, we further investigated the performance of our models using Grad-CAM. This technique produces visual explanations for decisions made by CNNs by using gradients of the target class flowing into the last Convolutional layer. It helped us visualize the most important regions of the same three images we tested with SHAP in the previous subsection.

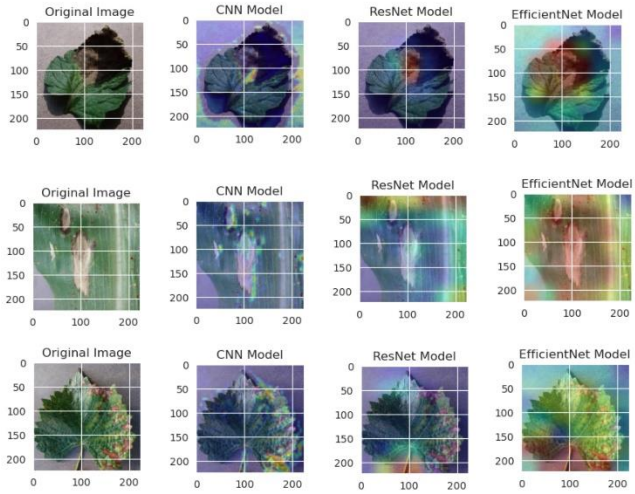

Figure 16: Grad-CAM heatmaps for the same 3 random samples

In this experiment, we observed that the CNN model was unable to accurately identify the diseased regions in all images. In contrast, the ResNet model successfully recognized the affected area in the first plant while in third plant recognized the petiole. The EfficientNet model outperformed again both the CNN and ResNet as it effectively highlighted the diseased areas in all three images. In addition, EfficientNet also gave significant importance to the petiole of the grape. These results demonstrate again the superior performance of models that are created using Transfer Learning.

5.3 XAI TECHNIQUES COMPARISON

Comparing the results of our SHAP and Grad-CAM experiments we noted that both techniques were effective in identifying the diseased regions. The EfficientNet model consistently outperformed the other two models in both experiments. Interestingly, the CNN performed better with SHAP than with Grad-CAM. Furthermore, even all the three models had impressive performance scores and correctly classified the three random plants with their respective diseases, only the results generated by SHAP appeared to be clearer and more easily understandable from a human interpretability perspective.

6. CONCLUSION

To conclude, our study demonstrates the potential of deep learning and transfer learning techniques to enhance the accuracy of plant disease classification. By comparing multiple Deep Learning models and extracting features from plants, we were able to detect correctly several plant diseases. What's more, our results show the effectiveness of both our approaches and with the utilization of eXplainable AI techniques we provide transparent and interpretable results. In future work, we aim to broaden the scope of our research by incorporating a more diverse range of plant species and diseases and contribute to the agriculture community.

7. ACKNOWLEDGMENTS

This work was supported by Professor Maglogiannis Ilias of the Department of Digital Systems at the University of Piraeus, and the Institute of Informatics and Telecommunications of NCSR "Demokritos". For more details, please visit the website of the Cross - Institutional MSc programme in Artificial Intelligence, https://msc-ai.iit.demokritos.gr/

REFERENCES

[1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning (Adaptive Computation and Machine Learning series). The MIT Press.

[2] Chollet, F. (2017). Deep Learning with Python. Manning Publications.

[3] Yang, Q., Zhang, Y., Dai, W., & Pan, S. J. (2020). Transfer Learning. Cambridge University Press.

[4] Wang, J., & Chen, y. (2023). Introduction to Transfer Learning: Algorithms and Practice. Springer

[5] Sarkar, D., Bali, R., & Ghosh, T. (2018). Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras. Packt Publishing Ltd.