World Journal of Advanced Engineering Technology and Sciences

eISSN: 2582-8266

Cross Ref DOI: 10.30574/wjaets

Journal homepage: https://wjaets.com/

(RESEARCH ARTICLE)

Check for updates

Explaining image classification model (CNN-based) predictions with lime

Eray ÖNLER *

Department of Biosystem Engineering, Faculty of Agriculture, Tekirdag Namik Kemal University, Tekirdag, Turkiye.

World Journal of Advanced Engineering Technology and Sciences, 2022, 07(02), 275–280

Publication history: Received on 17 November 2022; revised on 26 December 2022; accepted on 29 December 2022

Article DOI: https://doi.org/10.30574/wjaets.2022.7.2.0176

Abstract

Convolutional neural network models are black-box methods. The black-box in artificial intelligence means that model insights are based on the dataset, but the user does not know how. However, to obtain better classification models, it is important to monitor and understand how these models make decisions. In this way, classification success can be increased and improved. In this study, we examine how a model that classifies the disease in cassava leaves focuses on the image when making a decision. We used the cassava leaf dataset of five classes: Cassava Bacterial Blight, Cassava Brown Streak Disease (cbsd), Cassava Greem Mite (cgm), and Cassava Mosaic Disease (cmd). We used imagenet-trained Xception architecture for the base model to be used in transfer learning. The LIME library v 0.2.0 was used to examine which parts of the image affect the predictions made with the CNN model. With the LimeImageExplainer function, the superpixels are divided according to the weights of the model and then visualized. We can visually understand how the model decides on the predictions.

Keywords: Convolutional neural networks; Explainable AI; Artificial intelligence; Computer vision

1. Introduction

Machine learning models based on convolutional neural networks (CNN) have been widely used for computer vision problems in recent years due to their automatic feature extraction capabilities [1]. CNN models used for computer vision, similar to human perception, first detect low-level features (line, corner, edge, etc.) and then continue to perceive more general features in sequential layers [2]. In this study, we used an algorithm that highlights superpixels that contain high-level features that the network uses for decision making, which have a positive or negative effect on the model's decision process.

We aimed to examine a black box image classification model with the LIME (Local Interpretable Model-agnostic Explanations) library [3]. We examine where a model that classifies the disease in Cassava leaves focuses on the image when making a decision and how these regions affect the decision.

2. Material and methods

2.1. Dataset

In the study, we used a cassava leaf dataset from five classes: Cassava Bacterial Blight (cbb), Cassava Brown Streak Disease (cbsd), Cassava Greem Mite (cgm), Cassava Mosaic Disease (cmd) and healthy (Figure 1) [4]. There are 9430 labeled images in total in the dataset. These images are split into 5656 images for training, 1889 for validation, and 1885 for testing. We used training and validation dataset during the training and fine-tuning of the deep neural network model. After training the model, we examined the overall success using the test dataset.

* Corresponding author: Eray ÖNLER

Figure 1 Cassava Leaf Dataset Samples

2.2. Convolutional Neural Network Model

We created a CNN model that finds which class the images belong to. This model was trained with training and validation datasets, and then the accuracy of the model was carried out with a test dataset. When creating the CNN model, we used the transfer learning method. We used imagenet-trained **[5]**. Xception architecture **[6]** for the base model to be used in transfer learning.

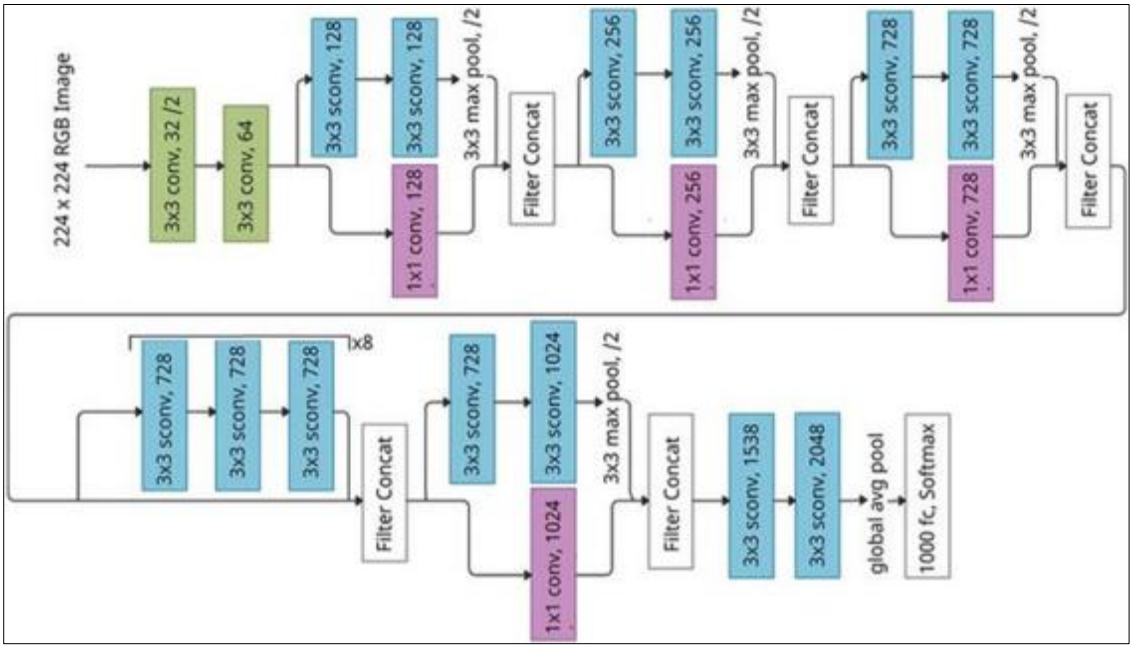


Figure 2 The architecture of the Xception deep CNN model [7]

We did not use the last layer of the Xception model and used it as a feature extractor. The RGB image that enters the model with dimensions of (224,224) is subjected to augmentation in the augmentation layer (random rotation and random horizontal flip) layer, feature extraction in the transfer learning layer, and then converted into a vector consisting of 2048 elements in the Global Average Pooling 2D layer. By applying dropout regularization to the vector coming out of the Global Pooling layer, it is estimated which class it belongs to in the last layer (Table 1).

Table 1 Architecture of the Deep Learning Model Used in the Study

Layer	**Output Shape**	**# Parameters**
Input Layer	[(None, 224, 224, 3)]	0
Augmentation Layer	(None, 224, 224, 3)	0
Xception (Transfer Learning Layer)	(None, None, None, 2048)	20.861.480
Global Average Pooling 2D Layer	(None, 2048)	0
Dropout Layer (0.2)	(None, 2048)	0
Output Layer	(None, 5)	10.245

During the training of model Adam optimizer [8] is used as optimizer function, the learning rate is set to 0.0001. Sparse categorical cross entropy is used as a loss function. All weights in the Xception layer used for transfer learning in the first 10 epochs were frozen and closed to training.

From the first 10 epochs, we opened the last 100 layers of the transfer learning layer to training. The optimizer learning rate was reduced by a factor of 10 and the training continued for 30 epochs. Table 2 shows the increase in the number of trainable parameters.

Table 2 Number of trainable parameters in the model before and after the fine-tuning

	10 Epoch lr=0.0001	**30 Epoch lr=0.00001**
Total Parameters	20.871.725	20.871.725
Trainable Parameters	10.245	9.488.589
Non-trainable Parameters	20.861.480	11.383.136

We used the TensorFlow 2.10.0 library to create artificial neural networks and the NVIDIA A100 SXM4 40GB GPU to accelerate training. We coded with Python 3.8.8 in Google Colab environment.

2.3. Explaining the Results with LIME

The LIME library v 0.2.0 was used to examine which parts of the image affect the result in the predictions made with the CNN model. With the LimeImageExplainer function in this library, the images used for evaluation are divided into superpixels according to the weights in the model and then visualized.

3. Results

Figure 3 shows the graphs of the accuracy and loss values obtained in the first 10 epochs.

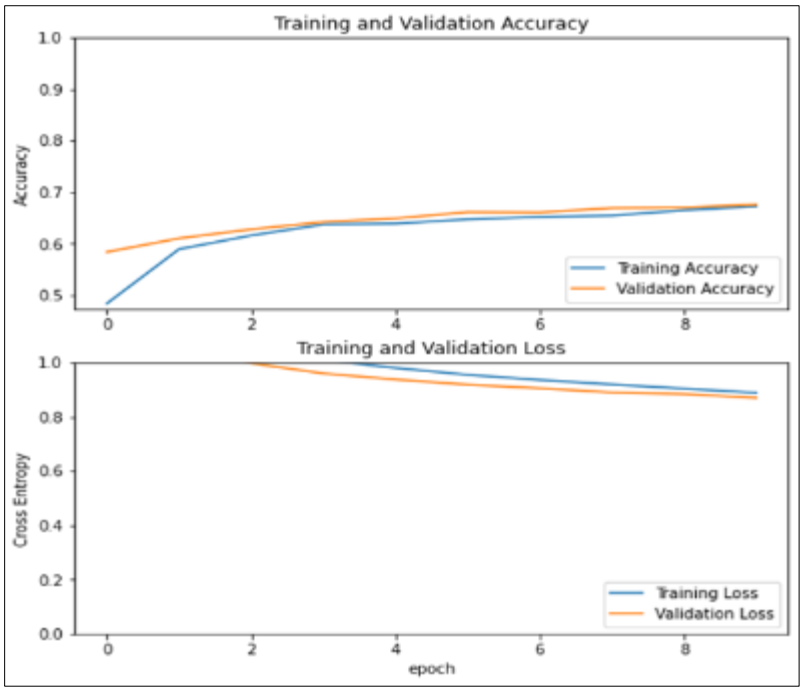


Figure 3 Accuracy and Loss Graphs before the fine-tuning

When we tried the model obtained after the first 10 epochs on the test data, the loss and accuracy values were 0.88 and 0.68, respectively.

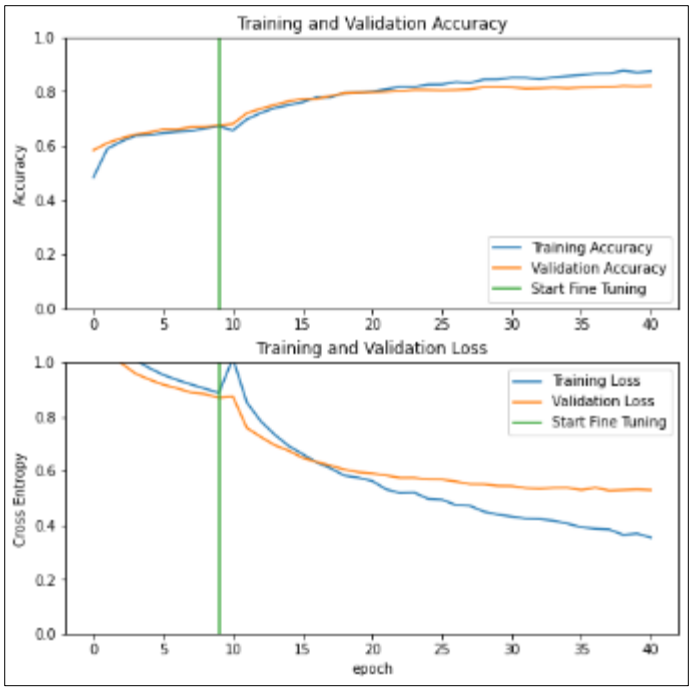
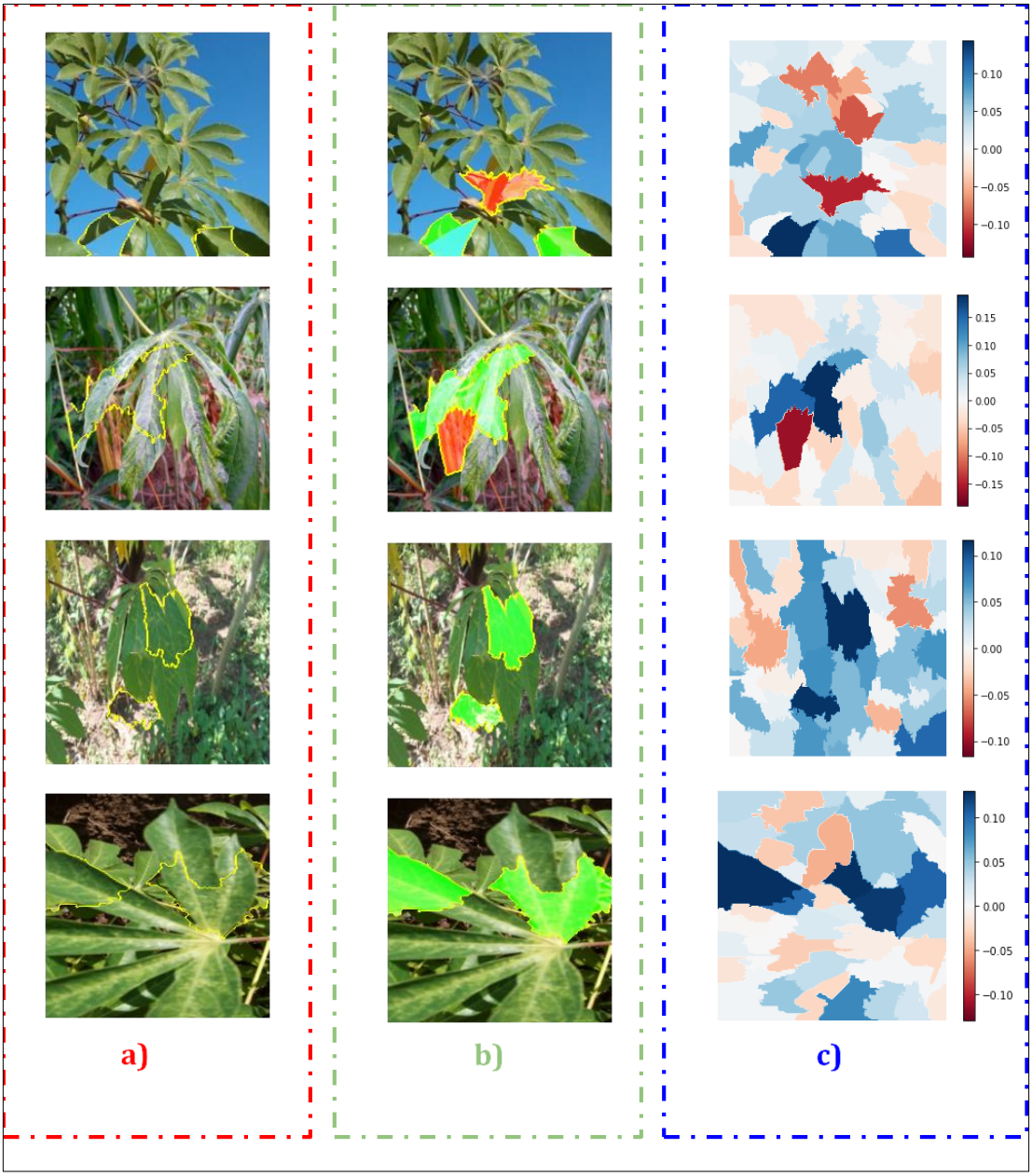


Figure 4 Accuracy and loss graphs after fine-tuning

The accuracy and loss graphs obtained after the fine-tuning process are shown in Figure 4. When we tested the model on the test dataset after fine tuning, the loss and accuracy values were 0.55 and 0.84, respectively.



a)Most important features for prediction are outlined with yellow line; b)The most important parts are highlighted with green and the less important parts are highlighted with red color; c)Plot weights as heatmap based on importance (red: less important, blue: most important)

Figure 5 Model explainer for image classification

The above samples show us how we can outline or highlight the super-pixels to identify the region of interest used by the model to make the prediction. What we see from here does make sense, and does allow us to increase trust toward black-box models. We can also form a heat map to show how important each superpixel is to get more granular explainability.

4. Conclusion

In this study, we evaluated the results of a CNN model used for the classification of diseased cassava leaves using the LIME library. CNN models have a black box structure. In order to obtain better classification models, it is important to monitor and understand how these models make decisions. In this way, classification success can be increased and improved.

As we clearly saw in this study, visual explanations of model predictions are important for building better models and understanding model decisions much better. The LIME library can be easily used to explain image classifiers. We can

use LIME to explain our model and find out if the model is looking into right areas of the image to make the final prediction.

Compliance with ethical standards

Funding

I have not received any external funding for the research underlying my paper.

References

[1] Li, D., & Du, L. (2022). Recent advances of deep learning algorithms for aquacultural machine vision systems with emphasis on fish. Artificial Intelligence Review, 55(5), 4077-4116.

[2] Gururaj, N., Vinod, V., & Vijayakumar, K. (2022). Deep grading of mangoes using Convolutional Neural Network and Computer Vision. Multimedia Tools and Applications, 1-26.

[3] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). why should I trust you?: Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, san francisco, ca, usa, august 13-17, 2016 (pp. 1135–1144)

[4] Mwebaze, E., Gebru, T., Frome, A., Nsumba, S., & Tusubira, J. (2019). icassava 2019fine-grained visual categorization challenge.

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

[6] Chollet, F. (2016). Xception: Deep learning with depthwise separable convolutions. arXiv. Retrieved from https://arxiv.org/abs/1610.02357 doi: 10.48550/ARXIV.1610.02357

[7] Srinivasan, K., Garg, L., Datta, D., Alaboudi, A. A., Jhanjhi, N. Z., Agarwal, R., & Thomas, A. G. (2021). Performance comparison of deep cnn models for detecting driver's distraction. CMC-Computers, Materials & Continua, 68(3), 4109-4124.

[8] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.