

LEAF-TYPE IMAGE CLASSIFICATION USING DEEP LEARNING METHOD CONVOLUTION NEURAL NETWORK

Mikael Reichi Sopany✉, Teny Handhayani

Fakultas Teknologi Informasi, Universitas Tarumanagara, Jakarta, Indonesia

Email: mikael.535210053@stu.untar.ac.id

DOI: <https://doi.org/10.46880/jmika.Vol9No1.pp86-91>

ABSTRACT

One of the most important parts of an ecosystem is plants. Plant life has given us many benefits, including food, oxygen, and medicine. There are many species of plants, each with its unique benefits and utilities. In this paper, we aim to identify plants by their leaves using deep learning. For this research, we utilized the convolutional neural network architecture Xception to classify five different types of leaves. We used 1,075 images of leaves that can be classified into five different types. The classification model achieved an overall accuracy score of 74%. We hope that the results of our research will help people's lives by enabling them to identify the plants they have so that they can use them for their benefit.

Keyword: *Convolutional Neural Network, Classification, Deep Learning, Supervised Learning.*

INTRODUCTION

Plant life is a fundamental part of the history of our planet. Plants have provided resources, created habitats for animals, and influenced the climate on a global scale (Morris, et al., 2018). There are approximately 374,000 accepted species of plants, each with its own characteristics (Christenhusz & Byng, 2016). However, identifying these species remains a significant challenge. Studies show that up to 10–20% of plant species have yet to be scientifically classified, while many known species still lack sufficient identification data (Pimm & Joppa, 2015). Additionally, the Global Strategy for Plant Conservation (GSPC) has highlighted that a significant percentage of plant species are at risk of extinction due to habitat loss, underscoring the urgency of developing efficient identification tools. Because there are so many species of plants, identifying them is very difficult. That is why we need a tool to help us identify them so that we can use them to their fullest potential.

There has been a growing interest in classifying and identifying plants in recent years, notably through the examination of leaf traits. For botanical studies, crop management, and environmental monitoring, leaves are an invaluable source of data and play a key role in plant taxonomy. The ability to precisely categorize leaf types has significant implications for various industries, such as agriculture, forestry, and biodiversity preservation. Despite the advancements in botany, traditional plant identification methods are often labor-intensive, require expert knowledge, and can be prone to human error.

There has been a growing interest in classifying and identifying plants in recent years, notably through the examination of leaf traits. For botanical studies, crop management, and environmental monitoring, leaves are an invaluable source of data and play a key role in plant taxonomy. The ability to precisely categorize leaf types has significant implications for various industries, such as agriculture, forestry, and biodiversity preservation.

The introduction of deep learning has significantly improved image classification tasks, notably with Convolutional Neural Networks (CNNs) (Yu, 2022). CNNs have demonstrated remarkable performance in many visual identification tasks, including object detection, facial recognition, and image classification. Researchers have successfully classified images of leaves using CNNs, paving the way for promising automated plant identification systems. Several studies have compared different CNN architectures for plant classification. For example, (Liu, Yang, Cheng, & Song, 2018), constructed a ten-layer CNN for plant leaf classification and applied sample augmentation to improve accuracy. Their experimental results on the Flavia dataset, which consists of 4,800 leaf images across 32 species, showed that their proposed method achieved an accuracy of 87.92%. These findings suggest that deep learning, specifically CNNs, offers a significant improvement over traditional classification techniques, reinforcing the relevance of this research.

The purpose of this research is to explore the application of deep learning, specifically a

Convolutional Neural Network, for leaf-type image classification. Deep learning is a subset of machine learning that uses complex algorithms and vast amounts of data to train a model (Sarker, 2021). To create a reliable and effective model that can accurately categorize leaf types, we will leverage CNNs' built-in ability to learn hierarchical features from images.

The developed model could be incorporated into mobile apps or web-based systems, allowing users to recognize plant species by taking pictures of their leaves. For this research, we will use a deep learning method called a convolutional neural network with the Xception architecture.

In traditional machine learning, programmers must carefully define specific features that the computer should focus on. This process, known as feature extraction, can be time-consuming and requires expertise. The computer's success rate depends on the programmer's ability to define a feature set accurately. This process is called supervised learning. Deep learning has an advantage over traditional machine learning because the program can build the feature set on its own without the programmer's supervision. This process is called unsupervised learning. Unsupervised learning is usually faster and more accurate, as it eliminates the need for manually engineering features (Diallo, 2022).

Convolutional neural networks (CNNs) are a type of artificial neural network that can learn and extract features from data automatically using a convolution process. Unlike traditional methods that require manual feature engineering, CNNs eliminate the need for human intervention. CNN architecture is inspired by the human visual system. Each artificial neuron in a CNN corresponds to a biological neuron, while the kernels in a CNN represent different receptors that can detect various features. The activation functions in CNNs mimic the biological process in which only neural signals that exceed a certain threshold are transmitted to the next neuron (Li, Liu, Yang, Peng, & Zhou, 2022).

Xception is a deep learning architecture composed of a linear sequence of depthwise separable convolution layers with residual connections. This simplified design makes it easy to create and modify using high-level libraries like Keras or TensorFlow-Slim. It requires only 30–40 lines of code, unlike more complex architectures such as Inception V2 or V3. A publicly available implementation of Xception using Keras and TensorFlow is included in the Keras

Applications module under the MIT license (Chollet, 2016).

METHODS

Algorithm

Xception is different from Inception because Xception has a modified depthwise separable convolution. As shown in Figure 1 The original depthwise separable convolution is comprised of a depthwise convolution followed by a pointwise convolution. Unlike traditional convolution, it avoids performing convolution across all channels, reducing the number of connection and resulting in a smaller, more efficient model.

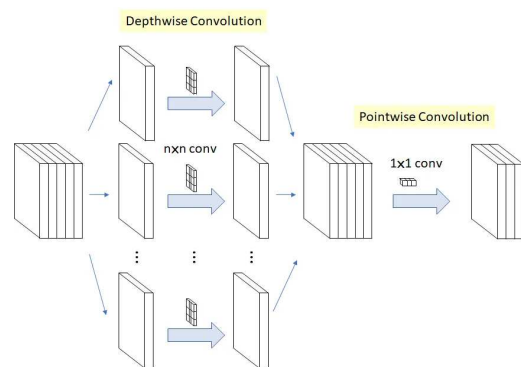


Figure 1. Original Depthwise Separable Convolution

As shown in Figure 2 The modified depthwise separable convolution is a technique that involves applying a pointwise convolution followed by a depthwise convolution. This approach is inspired by the Inception-v3 architecture, where 1×1 convolutions are used to reduce the number of channels before performing $n \times n$ spatial convolutions. This modification is slightly different from the original, which typically uses the depthwise convolution before the pointwise convolution.

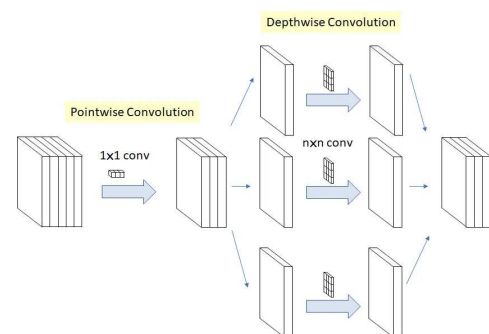


Figure 2. Modified Depthwise Separable Convolution used as an Inception Module in Xception

For this program, we will be using hyperparameters such as ReLU activation with 1024 units and SoftMax activation with 5 units. The optimizer for the model will be the Adam optimizer with a 0.001 learning rate. We will be using 50 epochs, with each epoch having 5 steps. These hyperparameters were chosen based on their effectiveness in previous deep learning studies and their ability to enhance model performance while preventing overfitting.

ReLU (Rectified Linear Unit) Activation are multiple layers of activation that the CNN model uses. Simplicity and the reliability of ReLU activation cause an increase in performance compared to other activation models (Ramachandran, Zoph, & Le, 2017). ReLU function activation performs Thresholding of the zero value on image input pixels. ReLU activation creates value pixels that have a value less than zero then it will be set to 0 in the image.

SoftMax activation or SoftMax Classifier is another type of activation function used in CNN. SoftMax activation is a combination of sigmoid activation. However, SoftMax activation unlike sigmoid activation that is used for binary classification. SoftMax activation usually counts probability in each target category/class of all possible target classes. when used in multi-classification models, SoftMax activation will return each class and the target class which will have the highest probability (Nwankpa, Ijomah, Gachagan, & Marshall, 2018).

The Adam optimizer was chosen due to its adaptive learning rate properties, which help achieve faster convergence compared to traditional optimization algorithms such as Stochastic Gradient Descent (SGD). The learning rate of 0.001 was selected based on prior research indicating that it provides a good balance between model stability and convergence speed. Additionally, 50 epochs were chosen to allow sufficient learning while avoiding overfitting, and each epoch consists of 5 steps to ensure effective weight updates. These hyperparameters were fine-tuned to optimize classification performance while maintaining computational efficiency.

Evaluation

The Evaluation method for this research is by using Confusion Matrix. To do the evaluation we used classification report from the scikit-learn metrics library. the result of classification report is a table composed of precision, recall, f1-score, and accuracy. Precision is the metric that give you the proportion of true positives to the number of total positives that the

model predicts. Recall is the metric that focuses on how good the model is at finding the positives. F1 Score is the metric that combines recall and precision. Accuracy measures how many predictions were true out of all the predictions that were made.

Here is the equation to calculate the number of each result in a classification report (Foody, 2023):

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Data

For this research, the dataset consists of images from five different classes of leaf types: kaffir lime leaves, chive leaves, turmeric leaves, bay leaves, and betel leaves. The dataset was obtained by collecting images from Google Images using a web scraper to gather the first 400 images of each leaf type. After collecting the images, we manually checked each one for irrelevant content that could not be used. After this process, we were left with 200 images of kaffir lime leaves, 219 images of chive leaves, 217 images of turmeric leaves, 221 images of bay leaves, and 218 images of betel leaves, for a total of 1,075 images.

The dataset was then put through a process of image data augmentation. Image data augmentation creates new images from existing ones to artificially increase the size and diversity of the training data. To do this, the images were altered by making small changes, such as rotating them, adjusting their brightness, or shifting the subject vertically or horizontally (Shorten & Khoshgoftaar, 2019). By incorporating these modified images, we aimed to improve the model's ability to generalize and handle variations in real-world leaf images. We used ImageDataGenerator from Keras' preprocessing image library to perform the image data augmentation.

To facilitate the training and evaluation of our model, we divided the dataset into a training and validation set. Seventy percent of the images (754 images) were allocated to the training set, while thirty percent (321 images) were reserved for the validation set. This division enabled us to train the model on a substantial portion of the data and assess its performance on a separate set to prevent overfitting.

Examples of the leaves that will be used in the dataset are shown in the figures below.



Figure 3. Example of Kaffir Lime Leaves



Figure 4. Example of Chive Leaves



Figure 5. Example of Turmeric Leaves



Figure 6. Example of Bay Leaves



Figure 7. Example of Betel Leaves

RESULTS AND DISCUSSION

For the evaluation, the loss function that we use is categorical cross-entropy, and Adam for the optimizer with 0.001 learning rate. And as we can see from Figure 8, the model starts with a very high loss score and very low accuracy score, but after 50 epochs the model keeps getting better.

```
Epoch 1/50  
5/5 [-----] - 64s 13s/step - loss: 188.6714 - accuracy: 0.0000e+00 - val_loss: 69.7132 - val_accuracy: 0.2025  
Epoch 2/50  
5/5 [-----] - 52s 12s/step - loss: 32.7545 - accuracy: 0.0000e+00 - val_loss: 7.6568 - val_accuracy: 0.2025  
Epoch 3/50  
5/5 [-----] - 78s 15s/step - loss: 7.2199 - accuracy: 0.0688 - val_loss: 3.6891 - val_accuracy: 0.3832  
Epoch 4/50  
5/5 [-----] - 68s 15s/step - loss: 2.8585 - accuracy: 0.3375 - val_loss: 2.4428 - val_accuracy: 0.4850  
Epoch 5/50  
5/5 [-----] - 62s 14s/step - loss: 3.7752 - accuracy: 0.1937 - val_loss: 1.9348 - val_accuracy: 0.5171  
Epoch 6/50  
5/5 [-----] - 78s 15s/step - loss: 1.6341 - accuracy: 0.5258 - val_loss: 1.3348 - val_accuracy: 0.4579  
Epoch 7/50  
5/5 [-----] - 85s 15s/step - loss: 1.6338 - accuracy: 0.3803 - val_loss: 1.2523 - val_accuracy: 0.4829  
Epoch 8/50  
5/5 [-----] - 78s 16s/step - loss: 1.3542 - accuracy: 0.4008 - val_loss: 1.2956 - val_accuracy: 0.5295  
Epoch 9/50  
5/5 [-----] - 87s 15s/step - loss: 1.4155 - accuracy: 0.4025 - val_loss: 1.2866 - val_accuracy: 0.5282  
Epoch 10/50  
5/5 [-----] - 64s 14s/step - loss: 1.3804 - accuracy: 0.5085 - val_loss: 1.2721 - val_accuracy: 0.4994  
Epoch 11/50  
5/5 [-----] - 63s 14s/step - loss: 1.0075 - accuracy: 0.5808 - val_loss: 1.3358 - val_accuracy: 0.5421  
Epoch 12/50  
5/5 [-----] - 68s 13s/step - loss: 1.2757 - accuracy: 0.4812 - val_loss: 1.2821 - val_accuracy: 0.5639  
Epoch 13/50  
5/5 [-----] - 59s 13s/step - loss: 1.4653 - accuracy: 0.3812 - val_loss: 1.0768 - val_accuracy: 0.5794  
Epoch 14/50  
5/5 [-----] - 59s 13s/step - loss: 1.2288 - accuracy: 0.5625 - val_loss: 1.1222 - val_accuracy: 0.5327  
Epoch 15/50  
5/5 [-----] - 58s 13s/step - loss: 0.7866 - accuracy: 0.7437 - val_loss: 1.0743 - val_accuracy: 0.5514  
Epoch 16/50  
5/5 [-----] - 59s 13s/step - loss: 0.8571 - accuracy: 0.7312 - val_loss: 1.0515 - val_accuracy: 0.5888  
Epoch 17/50  
5/5 [-----] - 59s 13s/step - loss: 0.9899 - accuracy: 0.5768 - val_loss: 1.0518 - val_accuracy: 0.5483  
Epoch 18/50  
5/5 [-----] - 68s 14s/step - loss: 1.1225 - accuracy: 0.5685 - val_loss: 1.0167 - val_accuracy: 0.6293  
Epoch 19/50  
5/5 [-----] - 57s 13s/step - loss: 0.6858 - accuracy: 0.7748 - val_loss: 0.8859 - val_accuracy: 0.6366  
Epoch 20/50  
5/5 [-----] - 57s 13s/step - loss: 0.7828 - accuracy: 0.7055 - val_loss: 0.8737 - val_accuracy: 0.6293  
Epoch 21/50  
5/5 [-----] - 59s 13s/step - loss: 0.7728 - accuracy: 0.6825 - val_loss: 0.9438 - val_accuracy: 0.6262  
Epoch 22/50  
5/5 [-----] - 68s 13s/step - loss: 0.8583 - accuracy: 0.6849 - val_loss: 0.8555 - val_accuracy: 0.6417  
Epoch 23/50  
5/5 [-----] - 56s 13s/step - loss: 0.6288 - accuracy: 0.6875 - val_loss: 0.6197 - val_accuracy: 0.7913  
Epoch 24/50  
5/5 [-----] - 55s 12s/step - loss: 0.3511 - accuracy: 0.8562 - val_loss: 0.6712 - val_accuracy: 0.7681
```

Figure 8. Training Model Epochs

We can visualize the plot for each iteration of accuracy and loss in Figure 9 and 10.

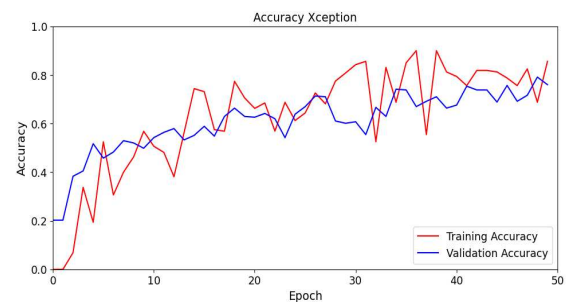


Figure 9. Plot of Accuracy

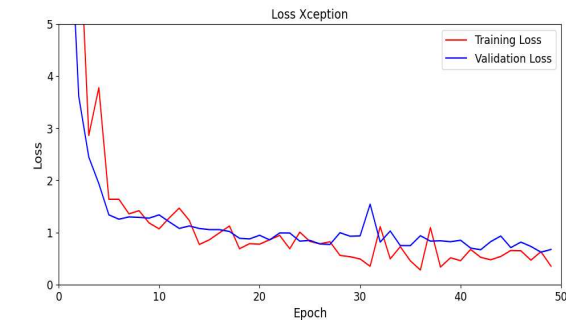


Figure 10. Plot of Loss

As illustrated in Figure 11, the classification model has achieved an accuracy score of 74%. While this result indicates satisfactory performance, a closer examination of the classification report for each class reveals that one particular class exhibits subpar performance. This suboptimal performance can be attributed to several factors, including potential shortcomings in the dataset.

One possible reason for the subpar performance is the quality of the images within the dataset. If the images are of low resolution, have insufficient detail, or contain artifacts, it can hinder the model's ability to make accurate classifications. To address this issue, future research efforts should focus on acquiring a dataset with higher quality images. By providing the model with more informative and visually distinct images, we can expect to improve its accuracy and overall performance.

	precision	recall	f1-score	support
0	0.50	0.80	0.62	60
1	0.91	0.98	0.95	65
2	0.88	0.78	0.83	65
3	0.71	0.36	0.48	66
4	0.83	0.80	0.81	65
accuracy			0.74	321
macro avg	0.76	0.75	0.74	321
weighted avg	0.77	0.74	0.74	321

Figure 11. Classification Report Result

CONCLUSION

The classification model we developed demonstrated promising results, achieving a commendable 74% accuracy score based on a comprehensive evaluation using the confusion matrix. While the model effectively classified leaf types, a detailed analysis of precision, recall, and F1-score indicated suboptimal performance in one particular class, likely due to dataset limitations such as insufficient sample size or class imbalance. To enhance accuracy, future research should focus on acquiring a

larger, more diverse dataset, increasing training epoch counts to improve learning, and refining hyperparameter tuning for additional performance gains. By pursuing these directions, we aim to develop a more robust classification model that advances leaf identification and related fields.

REFERENCES

Chollet, F. (2016, 10). *Xception: Deep Learning with Depthwise Separable Convolutions*. doi:10.1109/CVPR.2017.195

Christenhusz, M., & Byng, J. (2016, 5). The number of known plants species in the world and its annual increase. *Phytotaxa*, 261(3), 201-217. doi:https://doi.org/10.11646/phytotaxa.261.3.1

Diallo, A. (2022). *Artificial Intelligence-Based Medical PredictionS*. Retrieved from https://uniselinus.education/sites/default/files/2022-06/abdoulaye%20diallo.pdf

Foody, G. (2023, 10). Challenges in the real world use of classification accuracy metrics: From recall and precision to the Matthews correlation coefficient. *PLoS ONE*, 18(10 October). doi:https://doi.org/10.1371/journal.pone.0291908

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022, 12). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12), 6999-7019. doi:10.1109/TNNLS.2021.3084827

Liu, J., Yang, S., Cheng, Y., & Song, Z. (2018). Plant Leaf Classification Based on Deep Learning. *2018 Chinese Automation Congress (CAC)*, (pp. 3165-3169). China. doi:http://dx.doi.org/10.1109/CAC.2018.8623427

Morris, J., Puttick, M., Clark, J., Edwards, D., Kenrick, P., Pressel, S., . . . Donoghue, P. (2018, 3). The timescale of early land plant evolution. *Proceedings of the National Academy of Sciences of the United States of America*, 115(10), E2274-E2283. doi:https://doi.org/10.1073/pnas.1719588115

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018, 11). Activation Functions: Comparison of trends in Practice and Research for Deep Learning. doi:https://doi.org/10.48550/arXiv.1811.03378

Peter, W., Shengping, Z., Sharat, C., Stefan A., L., Scott L., W., & Thomas Serre. (2016). Computer vision cracks the leaf code. *Proceedings of the National Academy of*

- Sciences*, 113(12), 3305-3310.
doi:<https://doi.org/10.1073/pnas.1524473113>
- Pimm, S. L., & Joppa, L. N. (2015). How Many Plant Species are There, Where are They, and at What Rate are They Going Extinct? *Annals of the Missouri Botanical Garden*, 100(3), 170-176. doi:10.3417/2012018
- Ramachandran, P., Zoph, B., & Le, Q. (2017, 10). Searching for Activation Functions. doi:<https://doi.org/10.48550/arXiv.1710.05941>
- Sarker, I. (2021, 11). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, 2(6). doi:<https://doi.org/10.1007/s42979-021-00815-1>
- Shorten, C., & Khoshgoftaar, T. (2019, 12). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1). doi:<https://doi.org/10.1186/s40537-019-0197-0>
- Yu, Y. (2022, 10). Deep Learning Approaches for Image Classification. *ACM International Conference Proceeding Series*, 1494-1498. doi:10.1109/ICDSBA.2018.00101