Implementation of Xception method for Early Detection of Plant Diseases

[1]**Gilbertus Kristo Anugerah Adviandito**, [2]**Albertus Dwiyoga Widiantoro**
[1]Informatics Engineering [2]Information System
[1]Semarang State University, Semarang, Indonesia
[2]Soegijapranataa Catholic University, Semarang, Indonesia
[1]gilbertuskristo@students.unnes.ac.id, [2]yoga@unika.ac.id

Abstract- Plant diseases pose a significant threat to global food security, reducing both crop quality and productivity. To address this challenge, this study proposes an enhanced deep learning approach using a modified Xception architecture for plant disease detection. The model integrates Global Average Pooling and a Dropout layer to reduce overfitting and improve generalization performance on complex agricultural data. A large-scale multi-class dataset of 54,305 images across 38 plant species was employed, with preprocessing steps including resizing, augmentation, and normalization. The model was trained using the Adam optimizer with a batch size of 32 over 5 epochs, and its performance was compared against CNN, InceptionV3, InceptionResNetV2, and XGBoost. Experimental results demonstrate that the modified Xception achieved the highest accuracy, with 97.61% on training data and 97.63% on validation data, outperforming the other models under identical experimental settings.
The novelty of this research lies in the application of a modified Xception network on a large-scale, multi-class plant disease dataset combined with a systematic comparative evaluation against multiple architectures, an approach that has been rarely explored in previous studies. The findings not only confirm the robustness of Xception in handling complex agricultural imagery but also provide a practical framework for developing early disease detection systems in precision agriculture.

Keywords—Accuracy, Detection, Disease, Plants, Xception

I. INTRODUCTION

The agricultural sector is an important sector in ensuring food security for the world's population, the agricultural sector itself has a large contribution to the economic development of many countries[1]. In today's modern era, the agricultural sector itself has a major challenge, one of which is increasing crop productivity while maintaining plant health[2]. Of the several key factors that influence agricultural productivity, crop management and prevention are very important[2].

Every year, problems such as pests, environmental problems, and leaf problems make it difficult for farmers to produce a variety of healthy crops[2]. Plant diseases can cause significant economic losses due to decreased crop quality and quantity. Many developed countries face food quality challenges. According to projections from the World Health Organization, ¬food production will need to increase by 70% by 2050 to meet the needs of the global population.

The presence of problems such as plant diseases can have a major impact in reducing crop yields and quality, therefore it is important to identify plant diseases. Therefore, the development of Machine Learning and Deep Learning is used to detect diseases in plants [3]. The use of deep learning techniques in computer vision has opened up new opportunities to detect plant diseases quickly thanks to technological advances [4]. As one of the Deep architectures. The most advanced learning, Xception has the remarkable ability to recognize early signs of plant disease through image processing [5]. By implementing the Xception model into an image-based plant disease detection system, it is expected that, compared to conventional methods, the diagnosis of plant conditions will be more accurate and faster. This method allows farmers to track plant conditions more effectively and efficiently and allows for rapid intervention to prevent the spread of disease. The use of technology in plant disease detection not only provides an efficient solution but also has the potential to reduce losses caused by plant infections [6].

The challenges of maintaining plant health extend beyond disease identification and include managing environmental factors and agricultural practices [7]. Factors such as extreme weather, humidity, and soil quality can also impact plant growth and development, increasing susceptibility to disease. Therefore, a holistic approach is necessary to ensure optimal plant health and productivity [7].

In this regard, this study aims to develop and test the Xception model to rapidly detect diseases in fruiting plants [5]. Through the use of image processing and machine learning technologies [7]. This is expected to contribute to providing reliable and easy-to-use solutions for farmers to manage plant health, thereby assisting in efforts to increase agricultural productivity sustainably [3].

The novelty of this research lies in the application of the modified Xception model with Global Average Pooling and Dropout on a large-scale multi-class dataset across plant species, as well as its comparative evaluation with other popular architectures. This approach has not been widely used in the agricultural domain, so the results of this study not only strengthen the role of Xception as a more robust baseline but also provide a practical contribution to the implementation of AI-based plant disease detection systems in the field.

II. METHOD

Xception Model is a popular model for detect objects and classification image. This model modified with one-layer average collection or global average pooling that is for reduce dimensions spatial from feature before classification image [8]. there is One linear layer followed with function activation ReLU that helps introducing non- linearity to in the model, increasing ability from features complex [3].

For prevent occurrence overvitting and improve generalization model, one dropout layers are also included in the section end [9]. Layer dropout Work with random for turn off number of units on a layer certain during phase training [10]. Implementation elements This aim for increase Xception model accuracy in processing pictures, and allows this model for give more predictions accurate in very varied and complex data conditions [11].

2.1.Research Methods

2.1.1 Load Images

In this study, a modified Xception architecture was developed to detect plant diseases from leaf images. The methodological workflow of the proposed approach is illustrated in **Figure 1**.

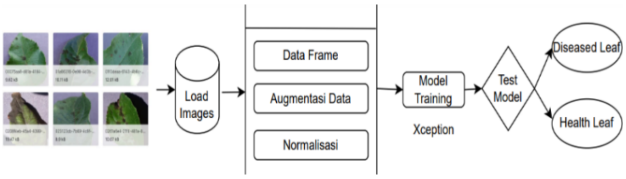

Figure 1. Modified Xception architecture for plant disease detection

In Figure 1, the Xception architecture is presented, illustrating its multi-stage image processing capability for detecting plant diseases[12] [13]. Identification process condition health leaf use technology Xception is one of the significant progresses in technology agriculture[14]. By leveraging deep learning models, researchers and practitioners are able to obtain deeper insights into plant health, thereby enabling earlier interventions to prevent the spread of diseases or to mitigate conditions that may negatively impact crop productivity[15]. This process, as depicted in the diagram, begins with the loading of input images.

In first step, leaf images under various health conditions are typically captured using high-quality cameras and related imaging devices[16]. However, in this study, we employed a dataset carefully curated from the open-access Kaggle platform, which comprises both real-time and ground-truth annotated images. This comprehensive dataset contains a total of 54,305 leaf images organized into 38 distinct classes, with 43,444 images allocated for training and 10,861 images for validation. The dataset encompasses a wide variety of plant species, including apple, blueberry, cherry, corn, grape, orange, peach, pepper, potato, pumpkin, soybean, raspberry, strawberry, and tomato. To ensure consistency during the training process, all images were uniformly resized to a resolution of 139 × 139 pixels.

In this study, a modified Xception architecture was developed to detect plant diseases from leaf images. The methodological workflow of the proposed approach is illustrated in Figure 1.

2.1.2 Preprocessing

In the image processing pipeline, the inclusion of a feature extraction stage is essential. Within the Xception architecture, the feature extraction process is inherently performed during the training phase of the model. The features here are abstract representations of the information contained in the image itself. Image data initially changed to in form matrix with a process called vectorization[17].

Figure 2 presents examples of leaf images, some of which are affected by disease and some of which are healthy leaves. Once the image dataset is uploaded, the subsequent stage involves data preprocessing. This process consists of several steps, including data frame creation, data augmentation, and normalization[17].

Figure 2. Example Leaf Image

After the image dataset is uploaded, the next step is data processing. The data processing process involves several steps, namely creating a data frame, data augmentation, and normalization [17]. Creating a data frame is a data structure used to store image information, the data frame also allows the storage of additional attributes such as category labels, namely by using healthy leaf and diseased leaf labels used in the Xception model training process and Xception model testing [18]. Data augmentation is a technique used to generate additional training data from existing data through modification, the data augmentation technique has the aim of making the model more robust to variations in new data [11][19].

Normalization is the process of adjusting the scale of pixel values in an image to a standard range, this step is important to help machine learning models process data more efficiently and effectively [20][21].

2.1.3 Train Model

Tabel 1. HyperParameters

Hyper-parameter	Values
Classes	38
Optimizer	Adam
Epochs	5
Learning Rate	0,01
Batch Size	32
Poolling	Global Average Pooling
Activation	Relu

Table 1 presents the hyperparameter configuration used during the model training. The proposed model employs the Xception architecture for image classification, implemented using the TensorFlow and Keras frameworks. The model structure begins with the definition of input images and leverages pre-trained Xception weights as the foundation of the architecture[22]. Xception is recognized as one of the most efficient and effective convolutional neural network (CNN) architectures, primarily due to its capability to extract hierarchical features from complex images[23].

Average Pooling as shown in table 1. Global Average Pooling have objective for reduce size Xception output feature that can produce representation more features solid and can used by layers next in the model. For extracting feature more next, layer Dense added for introducing non- linearity into the model with use activation Useful ReLU For learn deeper relationship complex between class labels and input features.

During training, dropout layer disables some units/neurons of layer previously for learn more features common and useful for avoid excessive dependence. After determine model structure, steps next that is compile the model. With use optimizer Adam for optimize the training process [17]. Optimizer Adam have ability For adapt rate learning in a way adaptive for every parameters. In addition, the function categorical crossentropy chosen as measurements to be taken optimized during training.

The process of loading training data started after the model is compiled with use Image Data Generator, which allows for data augmentation such as rotation, shift, and horizontal flip. For allows effective evaluation during training data sharing into training and validation subsets. For adapt model weights with training data, training process started with call function fit() on a model with parameters of training data, number of epochs, and validation data. This process ongoing with using 5 epochs for customize model weights with training data [24].

2.1.4 Layer Model

To further illustrate the architecture of the proposed model, Table 2 provides a detailed summary of the layers used in the modified Xception network for leaf image recognition and classification. The table presents information regarding the name of each layer, the corresponding output shape, and the number of parameters involved in the computation. This representation enables a clearer understanding of how the model processes input images through successive transformations before generating the final classification output.

Tabel 2. Layer Model

Layer	Output Shape	Param
Input Layer	(None, 139, 139, 3)	0
Conv2D	(None, 69, 69, 32)	864
BatchNormalization	(None, 69, 69, 32)	128
Activation	(None, 69, 69, 32)	0
...		
GlobalAveragePooling2D	(None, 2048)	0
Dense	(None, 128)	262272
Dropout	(None, 128)	0
Dense	(None, 38)	4902
Total params: 21128654 (80.60 MB)		
Trainable params: 21074126 (80.39 MB)		
Non-trainableparams : 54528 (213.00 KB)		

Input Layer the first model to receive input image and generates a tensor with the same shape with image . This image usually consists of from arrangement pixels , with mark pixels that indicate light intensity at a point certain in image . Layer second of the model is Conv2D which is layer convolution and layers This apply a number of filters to the input tensor [17]. Layer the three models are BatchNormalization , output layer previously normalized by BatchNormalization . This layer subtracts the mean output and divides it by the standard deviation. Activation This layer applies a non-linear function to the output of the previous layer. This function introduces non- linearity into the model, allowing for analysis between input data and output labels .

GlobalAveragePooling2D This layer takes the output of the previous layer and calculates the average value for each channel of the tensor , which reduces the dimensionality to a single value. Dense This layer takes the output of the previous layer and applies a linear transformation to it [25]. The dropout layer protects the model from overfitting the training data by regularizing the tensor elements. input randomly to zero during training [7]. Dense This final output layer is the layer that applies the softmax function to the output of the previous layer and converts it into a probability distribution over the output labels .

This table also shows the total number of model parameters, as well as the number of trainable and non-trainable parameters. Trainable parameters are those updated during training, while non-trainable parameters are those not updated during training.

2.1.5 Test Model

Model testing or data validation is step important for evaluate how much both models have been trained and able apply knowledge gained to new data that has not been once seen previously. The primary objective of this process is to verify the generalization capability of the model, i.e., its ability to perform reliably beyond the training dataset used for parameter optimization. Testing is crucial as it ensures that the model not only fits the training data but is also capable of predicting or classifying unseen samples with high accuracy.

In this study, The dataset that has been trained will tested For evaluate how much good Xception model This data validation consists of 38 image classes matched to the model's output classes. To ensure model consistency, the model's input specifications were resized to 299 x 299 pixels.

2.2.Mathematical Equations

The Softmax activation function is used in the output layer to classify leaf images into 38 different plant disease classes. This function converts raw scores (logits) into probabilities that can be summed together. The softmax function is formulated as follows:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{1}$$

where $\hat{y}_i$ is the predicted probability of the i-th class, z_i is the logit score for that class, and N is the total number of classes.

$$\mathcal{L} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{2}$$

Where $\hat{y}_i$ is the target label of the i-th class, and $\hat{y}_i$ is the model's prediction for the i-th class. This loss will be minimized during training using the Adam optimization algorithm.

$$g_k = \frac{1}{H\ x\ W}\sum_{i=1}^{H}\sum_{j=1}^{W} f_{i,j,k} \tag{3}$$

In Equation (3), $f_{i,j,kf}$, i,j,kf,$f_{i,j,k}$ is the activation of the kkk-th feature channel at pixel positions i,ji, ji,j, dan H,WH, WH,W is the height and width of the feature map. This technique helps avoid overfitting and simplifies the classification process.

III. RESULTS AND DISCUSSION

This section presents the training and validation results of the Xception model developed to detect plant diseases. The training process was conducted over five epochs using data that had been previously divided into training and validation subsets.

```
Epoch 1/5
1358/1358
[==============================]    -
1043s  740ms/step  -  loss:  0.4451  -  accuracy:
0.8809 - val_loss: 0.7783 - val_accuracy: 0.8394
Epoch 2/5
1358/1358
[==============================]    -
996s 733ms/step - loss: 0.1735 - accuracy: 0.9528
- val_loss: 0.2115 - val_accuracy: 0.9374
Epoch 3/5
1358/1358
[==============================]    -
991s 730ms/step - loss: 0.1306 - accuracy: 0.9645
- val_loss: 0.2110 - val_accuracy: 0.9442
Epoch 4/5
1358/1358
[==============================]    -
995s 732ms/step - loss: 0.1099 - accuracy: 0.9706
- val_loss: 0.0971 - val_accuracy: 0.9733
Epoch 5/5
1358/1358
[==============================]    -
1003s  738ms/step  -  loss:  0.0877  -  accuracy:
0.9761 - val_loss: 0.0872 - val_accuracy: 0.9763
```

The test results show that in epoch 1, the Xception model started with a relatively high loss and achieved relatively low accuracy during training. In epochs 2-4, the Xception model experienced a very significant decrease in loss and a significant increase in accuracy. At the end of the 5th epoch , the model achieved an accuracy of 0.9761 on the training data and an accuracy of 0.9763 on the validation data.

Figure 3 illustrates the training and validation accuracy of the model across multiple epochs. An epoch refers to a complete iteration of the training dataset through the model. The objective of training is to achieve high accuracy not only on the training dataset, which is used to optimize the model parameters, but also on the validation dataset, which evaluates the model's ability to generalize to unseen data. Training accuracy represents the proportion of correctly classified instances in the training dataset, while validation accuracy reflects the model's performance on independent validation samples.

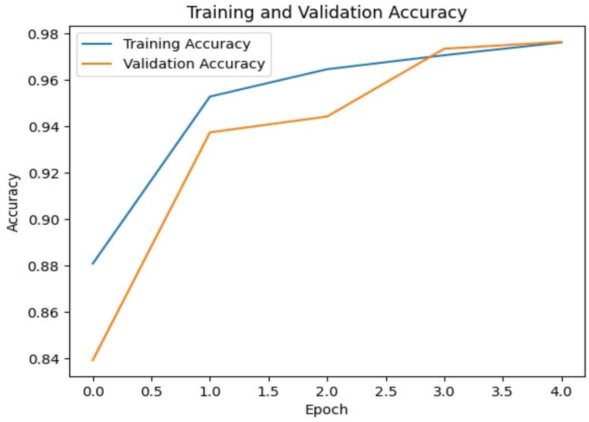

Figure 3. Training and validation accuracy of the Xception model across epochs

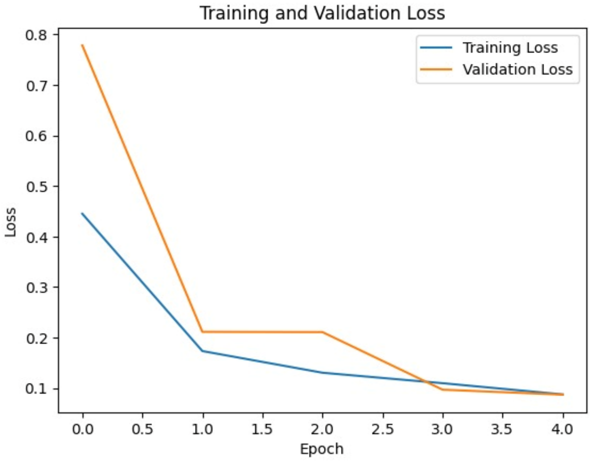

Figure 4. Training and validation loss of the model across epochs.

The graphs (Fig 4) show the training and validation losses of a machine learning model over several epochs . An epoch is one full iteration of the training data through the model. The validation loss indicates how well the model performs on unseen (validation) data. A lower validation loss indicates a model 's ability to generalize well and avoid overfitting . training shows how well the model performs on the training data. The lower the loss, the training , indicating the model is better at fitting the training examples. The goal is to reduce learning and validation loss. In the best-case scenario, both lines consistently fall and move toward low values.

Tabel 3. Comparative performance of CNN, Xception, Inception, InceptionResNetV2, and XGBoost

Network	Batch size	Epoch	Learning rate	Training Accuray	Validation Accuracy
CNN	32	5	0.001	0.9758	0.9173
Xception	**32**	**5**	**0.001**	**0.9761**	**0.9763**
Inception	32	5	0.001	0.9511	0.8832
Inception-ResnetV2	32	5	0,001	0.9371	0.9377
XGBoost	32	5	0,001	0.9137	0.9604

Table 3. summarizes the comparative performance of five deep learning architectures CNN, Xception, Inception, InceptionResNetV2, and XGBoost evaluated under identical experimental settings. All models were trained using a learning rate of 0.001, categorical crossentropy as the loss function, and a batch size of 32, with the dataset divided into training and validation subsets in an 80:20 ratio.

Among the evaluated models, Xception achieved the highest performance, with a training accuracy of 97.61% and a validation accuracy of 97.63%. This result demonstrates its superior generalization capability and robustness when handling unseen data. CNN exhibited comparable training accuracy (97.58%) but lower validation accuracy (91.73%), suggesting potential overfitting despite its strong training performance. The Inception model yielded a validation accuracy of 88.32%, which was lower than both Xception and CNN, indicating limited generalization in this configuration. InceptionResNetV2 achieved relatively balanced results (93.71% training accuracy and 93.77% validation accuracy), reflecting stable learning yet not surpassing Xception. Interestingly, XGBoost produced a relatively high validation accuracy of 96.04% despite its lower training accuracy of 91.37%, implying strong generalization but reduced ability to capture complex feature representations compared to CNN-based models.

Overall, the results confirm that Xception consistently outperforms the other architectures in both training and validation accuracy, establishing it as the most effective and reliable model for the classification task under the given experimental configuration.

IV. CONCLUSION

This study addresses challenges in agriculture by implementing a CNN-based Xception transfer learning model for plant disease detection. Through the utilization of image processing technology and machine learning techniques, the model's performance was evaluated over five training epochs. Initially, the Xception model experienced high failure rates and relatively high loss. However, after three additional epochs, there was a noticeable improvement in accuracy and a significant reduction in loss. By the end of the training process, the model achieved excellent performance, with an accuracy of 0.9760 on the training data and 0.9763 on the validation data.

Although the Xception model is successful in detecting plant diseases, the model is highly dependent on the quality of the dataset used. And the Xception model Xception itself has a complex model consisting of many convolutional layers and many parameters. This complexity makes the model slower and requires more computation, but this complexity can be handled by using a computer GPU during training, which reduces training time. However, with the large number of parameters in the Xception model , there is a risk of overfitting , where the model is too complex for the training data and its performance decreases when it receives new data.

Despite these limitations, the quality and quantity of the dataset play a crucial role in the model’s performance. A high-quality, diverse, and sufficiently large dataset enables the model to learn relevant features and generalize to new data effectively. By focusing on the development of the dataset and continuous evaluation of the model's performance, the potential of the Xception model can be maximized to aid in plant disease detection. It is anticipated that technologies like Xception will continue to evolve, becoming increasingly effective tools in the agricultural sector.

REFERENCES

[1] K. G. Panchbhai, M. G. Lanjewar, V. V. Malik, and P. Charanarur, “Small size CNN (CAS-CNN), and modified MobileNetV2 (CAS-MODMOBNET) to identify cashew nut and fruit diseases,” *Multimed Tools Appl*, 2024, doi: 10.1007/s11042-024-19042-w.

[2] M. Srivastava and J. Meena, “Plant leaf disease detection and classification using modified transfer learning models,” *Multimed Tools Appl*, Apr. 2023, doi: 10.1007/s11042-023-16929-y.

[3] A. Abisha and N. Bharathi, “Modified Xception-RF: A biotic stress prediction model in plant leaves using feature extraction and classification,” *Multimed Tools Appl*, Apr. 2024, doi: 10.1007/s11042-024-18534-z.

[4] J. U. M. Akbar, S. F. Kamarulzaman, A. J. M. Muzahid, M. A. Rahman, and M. Uddin, “A Comprehensive Review on Deep Learning Assisted Computer Vision Techniques for Smart Greenhouse Agriculture,” *IEEE Access*, vol. 12, pp. 4485–4522, 2024, doi: 10.1109/ACCESS.2024.3349418.

[5] N. Yao *et al.*, “L2MXception: an improved Xception network for classification of peach diseases,” *Plant Methods*, vol. 17, no. 1, Dec. 2021, doi: 10.1186/s13007-021-00736-3.

[6] “764-2559-1-PB”.

[7] S. Mishra, R. Sachan, and D. Rajpal, “Deep Convolutional Neural Network based Detection System for Real-time Corn Plant Disease Recognition,” in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 2003–2010. doi: 10.1016/j.procs.2020.03.236.

[8] N. An, “Xception Network for Weather Image Recognition Based on Transfer Learning,” in *Proceedings - 2022 International Conference on Machine Learning and Intelligent Systems Engineering, MLISE 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 330–333. doi: 10.1109/MLISE57402.2022.00072.

[9] S. A. Joshi, A. M. Bongale, P. O. Olsson, S. Urolagin, D. Dharrao, and A. Bongale, “Enhanced Pre-Trained Xception Model Transfer Learned for Breast Cancer Detection,” *Computation*, vol. 11, no. 3, Mar. 2023, doi: 10.3390/computation11030059.

[10] H. H. Farag, L. A. A. Said, M. R. M. Rizk, and M. A. E. Ahmed, “Hyperparameters optimization for ResNet and Xception in the purpose of diagnosing COVID-19,” *Journal of Intelligent and Fuzzy Systems*, vol. 41, no. 2, pp. 3555–3571, 2021, doi: 10.3233/JIFS-210925.

[11] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions.”

[12] W. Muhammad, S. Aramvith, and T. Onoye, “Multi-scale Xception based depthwise separable convolution for single image superresolution,” *PLoS One*, vol. 16, no. 8 August, Aug. 2021, doi: 10.1371/journal.pone.0249278.

[13] B. S. Abunasser, M. Rasheed, J. Al-Hiealy, I. S. Zaqout, and S. S. Abu-Naser, “Breast Cancer Detection and Classification using Deep Learning Xception Algorithm.” [Online]. Available: www.ijacsa.thesai.org

[14] Universitas Diponegoro. Faculty of Science and Mathematics. Department of Informatics, Institute of Electrical and Electronics Engineers. Indonesia Section, and Institute of Electrical and Electronics Engineers, *Proceeding : the 4th International Conference on Informatics and Computational*

Sciences : ICICoS 2020, 10-11 November 2020.

[15] S. Sarkar, J. Adhikari Ray, C. Mukherjee, S. Ghosh, N. Jayanthi, and K. R. Chairma Lakshmi, "Plant Leaf Disease Classification Based on SVM Based Densenets," in *2023 International Conference on Advances in Computation, Communication and Information Technology, ICAICCIT 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 636–641. doi: 10.1109/ICAICCIT60255.2023.10465886.

[16] Ö. POLAT, "Detection of Covid-19 from Chest CT Images using Xception Architecture: A Deep Transfer Learning based Approach," *Sakarya University Journal of Science*, vol. 25, no. 3, pp. 800–810, Jun. 2021, doi: 10.16984/saufenbilder.903886.

[17] X. Wu, R. Liu, H. Yang, and Z. Chen, "An Xception Based Convolutional Neural Network for Scene Image Classification with Transfer Learning," in *Proceedings - 2020 2nd International Conference on Information Technology and Computer Application, ITCA 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 262–267. doi: 10.1109/ITCA52113.2020.00063.

[18] H. Madeira, J. G. Silva, and J. Carreira, "Xception: Software Fault Injection and Monitoring in Processor Functional Units Xception: Software Fault Injection and Monitoring in Processor Functional Units 1," 2001. [Online]. Available: https://www.researchgate.net/publication/2396771

[19] F. Salim, F. Saeed, S. Basurra, S. N. Qasem, and T. Al-Hadhrami, "DenseNet-201 and Xception Pre-Trained Deep Learning Models for Fruit Recognition," *Electronics (Switzerland)*, vol. 12, no. 14, Jul. 2023, doi: 10.3390/electronics12143132.

[20] N. Yao *et al.*, "L2MXception: an improved Xception network for classification of peach diseases," *Plant Methods*, vol. 17, no. 1, Dec. 2021, doi: 10.1186/s13007-021-00736-3.

[21] Ö. POLAT, "Detection of Covid-19 from Chest CT Images using Xception Architecture: A Deep Transfer Learning based Approach," *Sakarya University Journal of Science*, vol. 25, no. 3, pp. 800–810, Jun. 2021, doi: 10.16984/saufenbilder.903886.

[22] N. Jinsakul, C. F. Tsai, C. E. Tsai, and P. Wu, "Enhancement of deep learning in image classification performance using Xception with the swish activation function for colorectal polyp preliminary screening," *Mathematics*, vol. 7, no. 12, Dec. 2019, doi: 10.3390/MATH7121170.

[23] M. T. Do, H. N. Huynh, T. N. Tran, and T. L. Hoang, "Prediction of Retina Damage in Optical Coherence Tomography Image Using Xception Architecture Model," in *2023 IEEE 5th Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability, ECBIOS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 58–61. doi: 10.1109/ECBIOS57802.2023.10218586.

[24] R. K. Dubey and D. K. Choubey, "Feature selection with Optimized XGBoost model-based paddy plant leaf disease classification," *Multimed Tools Appl*, 2024, doi: 10.1007/s11042-024-18524-1.

[25] N. M. Pandiyan, "Digital Image Forgery Detection Using Pre-Trained Xception Model as Feature Extractor," in *2023 International Conference for Advancement in Technology, ICONAT 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICONAT57137.2023.10080127.