Advance Sustainable Science, Engineering and Technology (ASSET)
Vol. 6, No.4, October 2024, pp. 0240409-01 ~ 0240409-08
ISSN: 2715-4211 DOI: https://doi.org/10.26877/asset.v6i4.790

# Jasmine Flower Classification with CNN Architectures: A Comparative Study of NasNetMobile, VGG16, and Xception in Agricultural Technology

**Danar Bayu Adi Saputra*, Christy Atika Sari, Eko Hari Rachmawanto**

Faculty of Computer Science, Universitas Dian Nuswantoro, Jl. Imam Bonjol 207 Semarang, 50131, Central Java, Indonesia

*111202113658@mhs.dinus.ac.id

**Abstract**. Jasmine flowers have many benefits and uses such as for traditional medicine, tea, perfume, cosmetics, decoration, and others. in the selection of fresh jasmine flowers for making tea is very important, currently the classification of jasmine flowers for making tea is mostly still using manual methods. Often influenced by individual preferences, opinions, or biases. this causes a lack of objectivity and uncertainty in the classification of jasmine flowers. The manual method is very weak due to human visual limitations and fatigue levels which can result in less than the optimal jasmine flower classification. Therefore, in the research that has been done, a transfer learning system was applied that can classify fresh jasmine flowers with rotten jasmine flowers. This study aims to compare three different Convolutional Neural Network architectures: NasNetMobile, VGG16, and Xception. The results on the three architectures can show maximum results, namely 99.21% for NasNetMobile, 98.69% for VGG16 and 97.91% for Xception. This study provides insight into the classification of good and bad jasmine flowers to encourage further exploration in the field of agriculture.

**Keywords**: jasmine flower, transfer learning, plant classification, CNN, AI in agriculture

***(Received 2024-07-03, Accepted 2024-08-23, Available Online by 2024-07-02)***

## 1. Introduction

Jasmine flowers originate from the genus Jasminum and are commonly known as jasmine or white jasmine. This plant species can be upright-stemmed or shrubby, and it tends to climb and can live for many years. Jasmine flowers can be found in plantations as well as in gardens. Jasmine grows well in locations that receive plenty of sunlight. Its adaptability is very rapid, allowing jasmine to be planted at any time and in any place, whether in pots or in open ground. Besides being favored for its fragrance, jasmine is also popular due to its easy maintenance, making it a favored ornamental plant for many people [1]. The aroma of jasmine flowers is used in beauty products or perfumes that contain phytochemicals extracted from the flowers of Jasminum Sambac. Linalool, eugenol, β-bisabolene, or β-caryophyllene, as well as other active compounds, can be found in jasmine plants. Eugenol, which is semipolar, can be extracted using ethyl acetate as a solvent. Due to its lipophilic nature, eugenol has the ability to penetrate the fatty acid chains in the bilayer membrane, allowing it to alter the permeability level of the cell membrane [2].

Aside from being used in beauty products, jasmine flowers can also be made into tea. Jasmine tea is known for its distinctive aroma and flavor, offering a refreshing and calming tea experience. The process involves steeping fresh jasmine flowers in hot water, resulting in a fragrant and flavorful drink. Jasmine tea is often considered to have soothing properties and is a good choice for breaks or creating a tranquil atmosphere. This plant is also commonly used in traditional medicine to treat fever, acne, diarrhea, influenza, and swelling from insect bites. The plant contains phytochemicals such as flavonoids, saponins, tannins, indole, and benzyl alcohol, which possess antimicrobial properties [3].

The classification of fresh jasmine flowers for tea making is crucial to achieve optimal results. Currently, the classification of jasmine flowers for tea making still relies largely on manual methods, which are often subjective. This subjective classification is influenced by individual preferences, opinions, and biases, resulting in subjectivity, and understanding in the classification process. The main challenges of this manual method are the limitations of human visual perception that can vary between individuals, as well as the level of fatigue that can result in decreased accuracy and consistency in classification. As a result, this manual method is less than optimal and is often inconsistent in determining the quality of jasmine flowers.

Therefore, to distinguish fresh jasmine from stale jasmine, a more efficient method for classification is needed, one of which is using convolutional neural networks (CNN) [4]. CNN is an image processing method that is often used compared to other methods [5].CNN has several benefits, including great accuracy and efficiency in image processing[6]. With image recognition intelligence technology, it is possible to create a machine learning model that can recognize the difference between fresh jasmine and stale jasmine [7]. There are many models available, but the accuracy and stability of each model vary. This study will compare the Xception, NasNetMobile, and VGG16 models in the classification of fresh jasmine.

This study aims to compare three different Convolutional Neural Network (CNN) architectures: NasNetMobile, VGG16, and Xception, in the classification of fresh jasmine flowers. In the conducted experiments, we will train and evaluate these models on the same dataset to measure their respective performance. After classifying the images using the CNNs, we will perform performance evaluations and comparisons on these images. By comparing all the CNN architectures, we aim to determine the most effective method for recognizing fresh and non-fresh jasmine flowers.

## 2. Methods

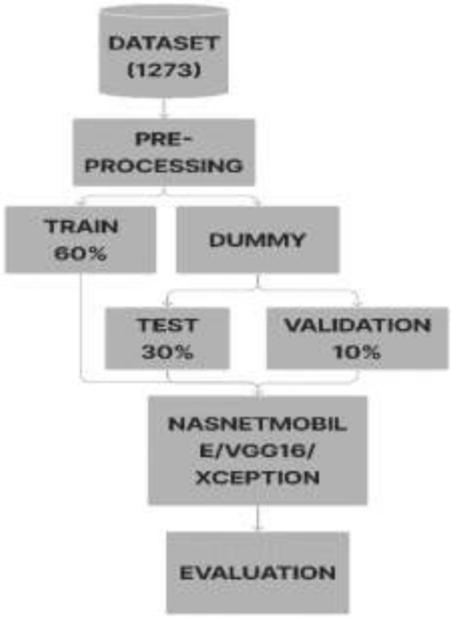



**Figure 1.** Dataset Pre-Processing Flow

Figure 1 shown the dataset process. Firstly, input into the `class df` and then split into two parts: Train and Dummy. Subsequently, the Dummy set is divided into two parts: Test and Validation. The Train, Test, and Validation sets will then be used in training using the proposed architectures: Xception, NasnetMobile, and VGG16. The dummy data is further split into two subsets: test data and validation data, with 30% of the data for testing and 10% for validation. These subsets will then be used to train the CNN architectures.

### *2.1. Dataset*

**Figure 2.** Sample of Datasets

Here, the dataset was directly collected from Mrs. Dasri jasmine garden on May 3, 2024, at 11:00 AM WIB in Tegal Regency using a Poco X6 smartphone camera. This dataset comprises 1,273 images of jasmine flowers, encompassing two classes: fresh jasmine and non-fresh jasmine. The dataset contains different quantities of images for each class, with 670 images of fresh jasmine and 603 images of non-fresh jasmine. A sample of the dataset is shown in Figure 2 below.

### *2.2. Image Preprocessing*

using a configuration that is set at 128x128 pixels with 3 color channels showing RGB format images. with a batch size of 64. Image preprocessing is required in order to enhance the image's quality [8].Some images may have scale differences, such as variations in lighting that can cause problems in model training. The model may not be able to generalize well to different lighting conditions. To overcome this problem and amplify or reduce the bias, image normalization with techniques such as rescale is used. Normalization, which usually involves scaling pixel values from the range 0-255 to 0-1, ensures that all images are at a consistent scale. In this way, the model can process the data uniformly and more effectively, reducing the differences between training, validation, and test images caused by lighting variations.

### *2.3. Model Training*

The architectures NasNetMobile, VGG16, and Xception were utilized in this study. Among them, NasNetMobile stands out as the lightest architecture. NASNetMobile is a neural architecture search network designed for embedded and mobile platforms. Its core architecture structure builds an optimized network framework through reinforcement learning using data-driven smart techniques. By applying repetitive operations on convolutional cells across the architecture, such as reduction cells and normal cells, the architecture generates feature maps. The architecture consists of twelve cells, each containing 5.3 million parameters [9]. The deep CNN network named VGG-16 was released in 2015. In this study, the Visual Geometry Group (VGG) investigated the impact of increasing the depth of convolutional networks on accuracy. Their architecture employed very small 3×3 convolutional filters, which demonstrated significant [10].

Improvement compared to previous configurations. With 3×3 convolutional kernels and 2×2 pooling layers, the VGGNet architecture can be seen as an extended version of AlexNet. Using smaller convolutional layers enhances feature learning and allows for deeper network architectures [11]. François Chollet created the Xception model, which utilizes the concept of Depthwise Separable Convolutions, a modification of regular convolutions aimed at being more efficient in terms of computational time [12]. This combination enhances accuracy in classifying images from the dataset. The feature extraction backbone of the network consists of 36 convolutional layers within the architecture of the Xception neural network layers [13]. The proposed architecture has been shown in Table 1 until Table 3.

**Table 1.** Proposed Architecture NasNetMobile

| Layer(type) | OutputShape | Param # |
|---|---|---|
| NasNetMobile | (None,4,4,1056) | 4269716 |
| Conv2D | (None,4,4,32) | 304160 |
| MaxPooling2D | (None,2,2,32) | 0 |
| Droopout | (None,2,2,32) | 0 |
| Flatten | (None,128) | 0 |
| Dense | (None,2) | 258 |

**Table 2.** Proposed Architecture VGG16

| Layer(type) | OutputShape | Param # |
|---|---|---|
| VGG16 | (None,4,4,512) | 14714688 |
| Conv2D | (None,4,4,32) | 1471488 |
| MaxPooling2D | (None,2,2,32) | 0 |
| Droopout | (None,2,2,32) | 0 |
| Flatten | (None,128) | 0 |
| Dense | (None,2) | 258 |

**Table 3.** Proposed Architecture Xception

| Layer(type) | OutputShape | Param # |
|---|---|---|
| Xception | (None,4,4,2048) | 20861480 |
| Conv2D | (None,4,4,32) | 589856 |
| MaxPooling2D | (None,2,2,32) | 0 |
| Droopout | (None,2,2,32) | 0 |
| Flatten | (None,128) | 0 |
| Dense | (None,2) | 258 |

Here, the architecture was initially trained using a base model pre-trained on ImageNet, and then extended by adding several layers. The Conv2D layer was used to extract local features from the input image, detecting patterns such as edges, textures, and other important elements of the image. The MaxPooling2D layer was employed to reduce the spatial dimensions (width and height) of the input, decreasing dimensionality and the number of parameters, thus helping to prevent overfitting during model training. The Dropout layer was used to further prevent overfitting by randomly omitting neurons, promoting the learning of more robust representations and evenly distributing weights across the network, thereby enhancing the model's generalization capability. The Flatten layer converted the three-dimensional output into a one-dimensional vector, which could then be connected to the dense layer. Finally, the dense layer performed the final classification.

In the Conv2D layer, RELU activation is used in convolutional neural networks which effectively improves the classification performance and learning rate. In addition, softmax activation function is applied to the dense layer to facilitate the final multiclass classification [14]. The Adam algorithm is a stochastic optimization algorithm that updates parameters by combining first and second moment gradients [15]. The optimizer uses Adam with a learning rate of 0.001, loss as categorical_crossentropy, and accuracy as the metric. In testing, a confusion matrix is added to the built model to measure the model's performance in classification.

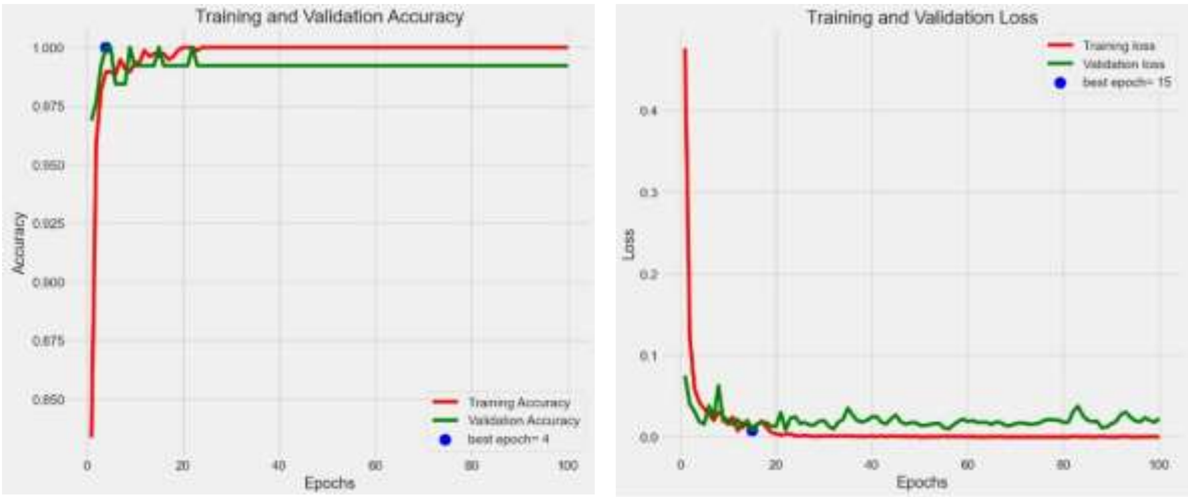
## 3. Results and Discussion

This chapter lays out specific instructions for writing the full text, including the article section, the systematic chapter and its contents. These specific instructions will guide the entire editorial process of the article as shown in Figure 2. This research was conducted on the jupyter notebook platform, trained using an i5-9300H CPU with 8GB of RAM on a 64-bit Windows 11 OS. Using various libraries such as keras, TensorFlow, numpy, os, pandas, matplotlib, and seaborn. The image size used as input data is 128x128 with 3 channels, namely RGB channels with a batch_size of 64 and an epoch of 100.

Table 4 shows the training results of three different deep learning models: NasNetMobile, VGG16, and Xception. NasNetMobile recorded the highest accuracy of 99.21% and also performed very well on other metrics with precision, recall, and f1-score values of 0.99 each. VGG16, although slightly lower in accuracy at 98.69%, still produced impressive results with precision, recall, and f1-score values of 0.99 each. Xception, with 97.91% accuracy, recorded precision, recall, and f1-score values of 0.98 each. Although its accuracy was slightly lower than the other two models, Xception still performed very well. Overall, all three models performed very well with consistently high evaluation metric values, indicating highly accurate and consistent data classification capabilities.

**Table 4.** Result Accuracy Research All Model

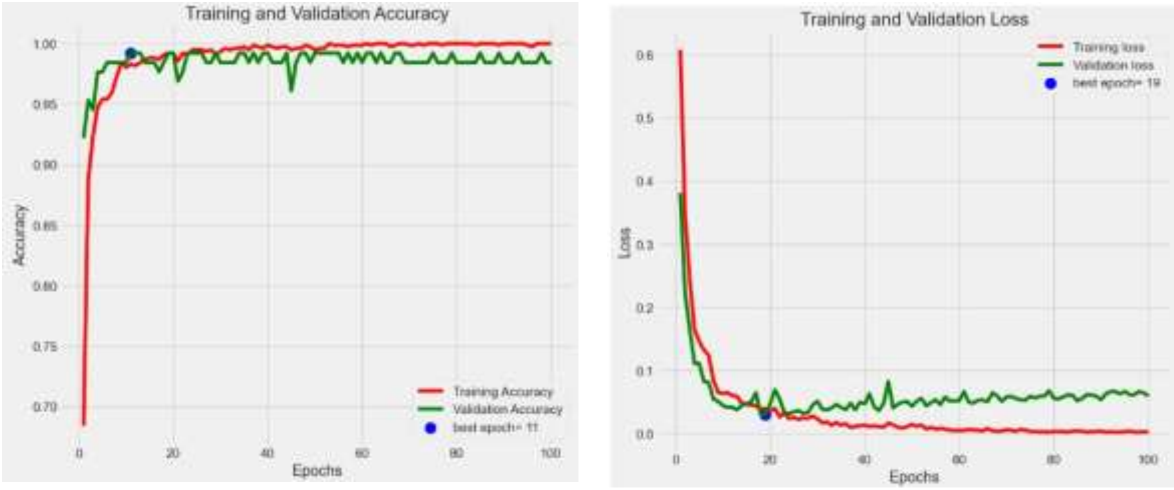
| Model | Accuracy | Precision | Recall | F-1Score |
|---|---|---|---|---|
| NasNet | 0.9921 | 0.99 | 0.99 | 0.99 |
| VGG16 | 0.9869 | 0.99 | 0.99 | 0.99 |
| Xception | 0.9791 | 0.98 | 0.98 | 0.98 |




(a) validation accuracy (b) validation loss

**Figure 3.** NasNetMobile Graph

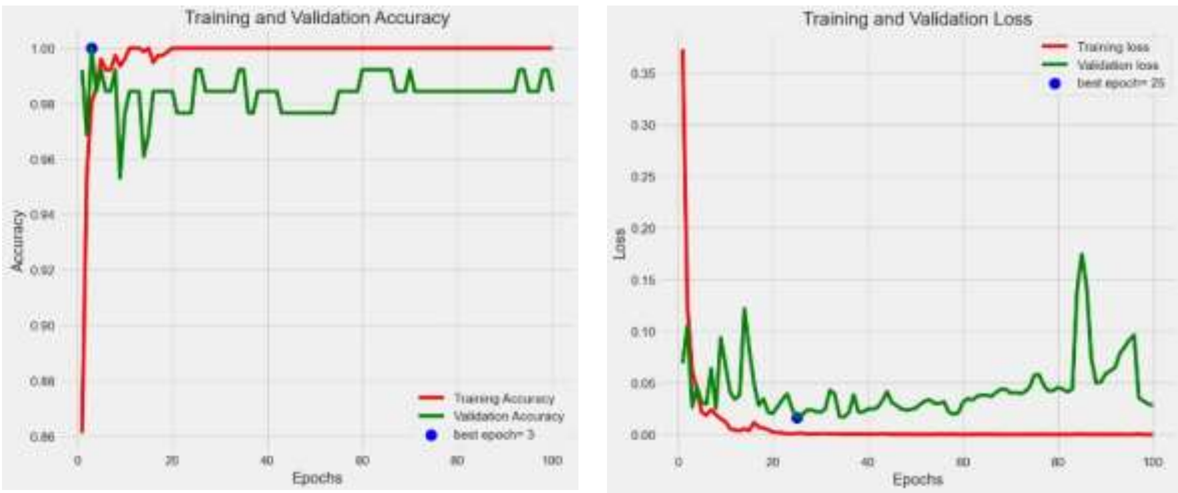
In Figure 3, there are two sub-graphs depicting the performance of the NasNetMobile model. Sub-graph (a) shows the validation pattern of the model during training. In the first epoch, the validation accuracy of NasNetMobile is 96.88%. As training continues, the accuracy increases and peaks at the 4th epoch with a perfect value of 100%. After that, although the accuracy shows performance, the value remains high. The accuracy stabilizes at around 99.22% at the 100th epoch. Sub-graph (b) depicts the decreasing pattern of the validation loss. The validation loss starts at 7.53% at the first epoch and decreases significantly to 0.8% at the 15th epoch. After that, although there are small fluctuations, the general trend shows a steady decrease, with the validation loss slightly increasing to 2.26% at the 100th epoch. These graphs show that NasNetMobile underwent an effective training process, with a consistent increase in accuracy and decrease in validation loss, despite some stress during training.




(a) validation accuracy (b) validation loss

**Figure 4.** VGG16 Graph Result

Figure 4 (a) and Figure 4 (b) show the validation accuracy patterns of VGG16 in Figure 4, which shows the NasNetMobile graph model. In the first epoch, the VGG16 model produces a validation accuracy of 92.19%. This accuracy continues to increase until it reaches its peak in the 11th epoch with a value of 99.22%. After that, although there are some fluctuations, the accuracy remains high until the end of training at the 100th epoch with a value of 98.44%. Graph (b) shows the validation loss pattern of VGG16. In the first epoch, the validation loss value was recorded at 38.12%, which is quite high. However, along with training, this value decreased drastically, reaching its lowest point in the 11th epoch with a value of 3%. Until the end of training at the 100th epoch, the validation loss value increased slightly to 5.95%, but still showed good performance. Overall, these graphs show that the VGG16 model experienced a significant increase in accuracy and a decrease in validation loss during the training process.




(a) validation accuracy (b) validation loss

**Figure 5**. Xception Graph

According to Figure 5, there are two sub-graphs that illustrate the Xception model's performance. Sub-graph (a) shows the validation accuracy, starting at 99.22% in the first epoch, peaking at 100% by the 3rd epoch, and finishing at 98.44% in the 100th epoch. Sub-graph (b) displays the validation loss, beginning at 6.89% in the first epoch, decreasing to its lowest point of 1.68% at the 25th epoch, and slightly increasing to 2.78% by the end of training. Overall, these graphs reveal that the Xception model experienced a notable increase in accuracy and a reduction in validation loss throughout the training process.

This study also refers to a study conducted by [16] in the context of Plant Disease Diagnostic using the NasNetMobile architecture. In the study, a 128x128 pixel image dataset was used, consisting of 54,309 images divided into 14 classes. With this large and diverse dataset, the NasNetMobile model managed to achieve an accuracy of 92.47%. In related research conducted by [17], regarding Transfer Learning VGG16 for Orange Fruit Image Classification using 1000 datasets with testing on the VGG16 architecture model, an accuracy of 97.5% was obtained.in a study conducted by [18] on Online recognition of peanut leaf diseases based on the data balance algorithm and deep transfer learning. the number of datasets 652 with the use of the Xception architecture obtained a high accuracy of 99%. Overfitting can occur on noise from training data [19], a model that has too many layers or parameters compared to the complexity of the training data can learn noise or specific patterns from the training data. So it loses the ability to generalize to the test data. As a result, the model will show very good performance on the training data but bad on new data. The limited dataset used can also affect the processed model because it does not have enough information to learn general patterns so it learns irrelevant patterns. Dropout a simple way to prevent neural networks from overfitting. One method that deals with both of these problems is dropout. It offers an effective means of combining exponentially many different neural network topologies and eliminates overfitting. The units in a neural network that are eliminated (both visible and hidden) are referred to as "dropout" [20].

## 4. Conclusion

The three architecture models tested showed excellent performance with high accuracy. The NasNetMobile model achieved the highest accuracy among the three models at 99.21%, demonstrating exceptional data classification ability. The VGG16 model had the second-highest accuracy, performing excellently but slightly below NasNetMobile. The last model, namely Xception, achieved an accuracy of 97.91% which is still considered very good although slightly below the other two models. The three models tested, namely NasNetMobile, VGG16, and Xception, all showed excellent performance in the classification task. The main findings of this study provide valuable insights into agricultural technology that can classify fresh and rotten jasmine flowers. The application of this classification system can help in the speed of jasmine flower classification in the field of agricultural technology. The application of this study can be used as a basis for further development in optimizing the CNN model that will be created by subsequent researchers on the use of technology for agriculture.

## References

[1] L. Gustira Annisa, T. Akbar, and P. Studi Kriya Seni Fakultas Seni Rupa dan Desain, “Bunga Melati Sebagai Motif Pada Baju Kuruang Basiba,” *Style J. Fash. Des.*, vol. II, no. Ii, pp. 68–76, 2023, doi: DOI:10.26887/style.v2i2.3740.

[2] F. N. Hikmah, S. Malahayati, and D. F. Nugraha, “Formulasi Dan Evaluasi Sediaan Serum Gel Ekstrak Bunga Melati (Jasminum sambac L.),” *J. Pharm. Care Sci.*, vol. 3, no. 2, pp. 93–108, 2023, doi: 10.33859/jpcs.v3i2.248.

[3] R. Setyowati, “Aktivitas Antioksidan Teh Herbal Kombinasi Daun Alpukat (Parsea americana Mill.), Daun Teh (Camelia sinnesis) dan Bunga Melati (Jasminum sambac L.) dengan Metode DPPH,” *J. Formil (Forum Ilmiah) KesMas Respati*, vol. 9, no. 2, pp. 132–147, 2022, doi: https://doi.org/10.35842/formil.v9i2.550.

[4] M. Rafid, A. Luthfiarta, M. Naufal, M. D. Al Fahreza, and M. Indrawan, “The Effect of LAB Color Space with NASNetMobile Fine-tuning on Model Performance for Crowd Detection,” *Adv. Sustain. Sci. Eng. Technol.*, vol. 6, no. 1, p. 02401014, 2024, doi: 10.26877/asset.v6i1.17821.

[5] A. N. R. Munandar and A. F. Rozi, “Analisis Arsitektur Convolutional Neural Network Untuk Klasifikasi Citra Bunga,” *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 6, no. 3, pp. 522–531, 2024, doi:

10.47233/jteksis.v6i3.1413.
[6] I. P. Kamila, C. A. Sari, E. H. Rachmawanto, and N. R. D. Cahyo, “A Good Evaluation Based on Confusion Matrix for Lung Diseases Classification using Convolutional Neural Networks,” *Adv. Sustain. Sci. Eng. Technol.*, vol. 6, no. 1, p. 0240102, 2023, doi: 10.26877/asset.v6i1.17330.
[7] R. A. Firmansah, H. Santoso, and A. Anwar, “Transfer Learning Implementation on Image Recognition of Indonesian Traditional Houses,” *J. Tek. Inform.*, vol. 4, no. 6, pp. 1469–1478, 2023, doi: 10.52436/1.jutif.2023.4.6.767.
[8] M. M. I. Al-Ghiffary, C. A. Sari, E. H. Rachmawanto, N. M. Yacoob, N. R. D. Cahyo, and R. R. Ali, “Milkfish Freshness Classification Using Convolutional Neural Networks Based on Resnet50 Architecture,” *Adv. Sustain. Sci. Eng. Technol.*, vol. 5, no. 3, p. 0230304, 2023, doi: 10.26877/asset.v5i3.17017.
[9] T. H. Chun, U. R. Hashim, S. Ahmad, L. Salahuddin, N. H. Choon, and K. Kanchymalay, “Efficacy of the Image Augmentation Method using CNN Transfer Learning in Identification of Timber Defect,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 5, pp. 107–114, 2022, doi: 10.14569/IJACSA.2022.0130514.
[10] D. Theckedath and R. R. Sedamkar, “Detecting Affect States Using VGG16, ResNet50 and SE-ResNet50 Networks,” *SN Comput. Sci.*, vol. 1, no. 2, pp. 1–7, 2020, doi: 10.1007/s42979-020-0114-9.
[11] Z. P. Jiang, Y. Y. Liu, Z. E. Shao, and K. W. Huang, “An improved VGG16 model for pneumonia image classification,” *Appl. Sci.*, vol. 11, no. 23, 2021, doi: 10.3390/app112311185.
[12] Rismiyati, S. N. Endah, Khadijah, and I. N. Shiddiq, “Xception Architecture Transfer Learning for Garbage Classification,” *ICICoS 2020 - Proceeding 4th Int. Conf. Informatics Comput. Sci.*, 2020, doi: 10.1109/ICICoS51170.2020.9299017.
[13] J. O. Carnagie, A. R. Prabowo, E. P. Budiana, and I. K. Singgih, “Essential Oil Plants Image Classification Using Xception Model,” *Procedia Computer Science*, vol. 204. pp. 395–402, 2022, doi: 10.1016/j.procs.2022.08.048.
[14] A. M. Alhassan and W. M. N. W. Zainon, “Brain tumor classification in magnetic resonance image using hard swish-based RELU activation function-convolutional neural network,” *Neural Comput. Appl.*, vol. 33, no. 15, pp. 9075–9087, 2021, doi: 10.1007/s00521-020-05671-3.
[15] N. Kadek, D. Adnyaswari, A. Luthfiarta, P. Langgeng, and W. Ellwid, “OPTIMIZING BUTTERFLY CLASSIFICATION THROUGH TRANSFER LEARNING : FINE-TUNING APPROACH WITH NASNETMOBILE AND MOBILENETV2,” *J. Tek. Inform.*, vol. 5, no. 3, pp. 685–692, 2024, doi: https://doi.org/10.52436/1.jutif.2024.5.3.1583.
[16] A. O. Adedoja, P. A. Owolawi, T. Mapayi, and C. Tu, “Intelligent Mobile Plant Disease Diagnostic System Using NASNet-Mobile Deep Learning,” *IAENG Int. J. Comput. Sci.*, vol. 49, no. 1, pp. 216–231, 2022.
[17] D. M. Asriny and R. Jayadi, “Transfer Learning VGG16 for Classification Orange Fruit Images,” *J. Syst. Manag. Sci.*, vol. 13, no. 1, pp. 206–217, 2023, doi: 10.33168/JSMS.2023.0112.
[18] Q. Feng *et al.*, “Online recognition of peanut leaf diseases based on the data balance algorithm and deep transfer learning,” *Precis. Agric.*, vol. 24, no. 2, pp. 560–586, 2023, doi: 10.1007/s11119-022-09959-3.
[19] J. Chen, Y. Cao, and Q. Gu, “Benign Overfitting in Adversarially Robust Linear Classification,” *Proc. Mach. Learn. Res.*, vol. 216, no. Uai, pp. 313–323, 2023, doi: 10.48550/arXiv.2112.15250.
[20] B. Sabiri, B. El Asri, and M. Rhanoui, “Mechanism of Overfitting Avoidance Techniques for Training Deep Neural Networks,” *Int. Conf. Enterp. Inf. Syst. ICEIS - Proc.*, vol. 1, no. May, pp. 418–427, 2022, doi: 10.5220/0011114900003179.