KHATULISTIWA SMART : JOURNAL OF ARTIFICIAL INTELLIGENCE
e-ISSN : 3089-9710
p-ISSN : XXXX-XXXX

**Khatulistiwa Smart : Journal of Artificial Intelligence**
Vol. 2 No. 1 (2025) : 126 – 137
*Available online at : https://journal.literasikhatulistiwa.org/index.php/smart*

# Classification of Poisonous Ornamental Plants Using CNN and ResNet Methods

**Ahmad Halimi [1] , Eka Herliana Agustini [2] , Nur Diyana Kholidah [3]**
[1,2,3] Universitas Nurul Jadid
ahmadhalimi@unuja.ac.id [1] , ekaherlianaagustini05@gmail.com [2],
ekaherlianaagustini05@gmail.com [3]

Doi
Submitted: 30 Nov 2025 Revision: 13 Dec 2025 Accepted: 19 Dec 2025 Published: 31 Dec 2025

**Abstract** This research, titled "Classification of Poisonous Ornamental Plants Using CNN and ResNet Methods," defines poisonous ornamental plants as those containing toxic substances that can cause pain, allergies, or even death. These plants come in many different varieties, each with its own unique appeal. Many laypeople still find it difficult to distinguish poisonous from non-poisonous ornamental plants, especially in household environments that pose a risk to children and pets. Therefore, vigilance is needed in recognizing these plants. This study aims to develop a classification system for poisonous and non-poisonous ornamental plants using CNN methods with and without ResNet architecture. Furthermore, this study also aims to implement the trained model into a web application using Flask, so users can easily upload images of ornamental plants and obtain information about their potential toxicity in real-time. This research method uses deep learning techniques, specifically Convolutional Neural Network (CNN) with ResNet-50 and regular Convolutional Neural Network (CNN) or without ResNet with data divided into (70% training, 15% validation, and 15% testing). The test results for ResNet showed an accuracy of 98.25%, while the test results for regular CNN reached an accuracy of 87.47%. These accuracy results indicate that CNN with ResNet is superior for classifying poisonous and non-poisonous ornamental plants compared to CNN without ResNet.

***Keywords*** *Image Classification, Poisonous Ornamental Plants, CNN, ResNet, Deep Learning, Flask.*

## 1. INTRODUCTION

Houseplants have become an important part of home and public space decor because they provide visual beauty while improving air quality. However, not all houseplants are safe. Some contain toxic compounds that can harm human and pet health. For example, daffodils (Narcissus) contain lycorine, caladiums (Caladium) are toxic category 2 and 3, and lilies (Lilium genus) can cause irritation and digestive upset [1]. Even Sansevieria, or snake plants, can cause mild allergic reactions in sensitive skin. These cases demonstrate the importance of identifying toxic houseplants, especially in households with small children or pets [2].

Advances in digital image processing and artificial intelligence technology have opened up new opportunities for automatically classifying poisonous plants. Convolutional Neural Network (CNN) methods have been widely used in image classification research, but specific applications to poisonous ornamental plants remain limited. Furthermore, the use of the ResNet architecture, known for its ability to address vanishing gradient problems and improve accuracy, has not been widely implemented in this context. [3]

Based on this, this study aims to classify poisonous ornamental plants by comparing the performance of conventional CNN and CNN with ResNet architecture. Furthermore, this study also develops a web-based application using the Flask framework to enable users to identify plants in real-time. Thus, this research is expected to contribute to the field of image processing while increasing public awareness of the dangers of poisonous ornamental plants.[4]

## 2. RESEARCH METHODS

In this study, quantitative methods were applied to collect and analyze numerical data. The primary focus was on the model's performance in classifying eggplant diseases using the web-based DenseNet201 algorithm. By utilizing processed digital image data, this study objectively measured the accuracy, precision, recall, and F1-score of the developed model. Figure 1 shows the stages of this study, arranged in the order shown in the flowchart.

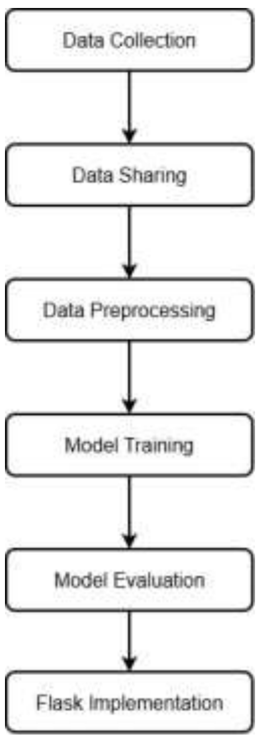



**Figure 1.** Stages of research methods

### 2.1. Data Sharing

The first stage of this research was data preparation, which began with collecting datasets from various Kaggle platforms to obtain a suitable dataset. The dataset used consisted of 3,320 images with a resolution of 224x224. Of these, approximately 664 images were original images of various poisonous ornamental plants, while the remainder were obtained through augmentation techniques such as zooming and rotation to increase the sample size [5].

### 2.2. Data Preprocessing

After the dataset was collected, the next step was to divide the augmented data into three main subsets: 1859 training data, 398 validation data, and 399 testing data. The datasets remained the same because both models needed to be tested on identical data to ensure fair and valid comparisons. In this study, the data was divided in a 70:15:15 ratio, with 70% of the total dataset used for training, 15% for validation, and the remaining 15% for testing.[6] Then, both models (simple CNN and CNN + ResNet) were trained and tested on the same data to accurately compare their performance.

### 2.3. Data Preprocessing

Before data is used for model training, a preprocessing stage is performed to improve the quality of the data that will be fed into the artificial neural network. One of the main techniques used in this stage is normalization, particularly in the validation dataset [7]. The data preprocessing stage includes model and processing: Using CNN and ResNet50 models trained with ImageNet weights, establishing standard image sizes, and creating preprocessing functions to prepare images for input into the model; data generator: Setting a data generator for training that performs image augmentation; and generators for validation and testing. The training and validation data are randomized for variation, while the testing data is unrandomized to compare predictions with the actual labels; and batch size: Determining the optimal batch size for the testing data to ensure efficient model evaluation.

### 2.4. Model Training

After the data has been properly processed through a series of preprocessing steps to ensure quality and consistency, a deep learning model is trained to effectively classify images. The model used in this study is ResNet-50, a convolutional neural network (CNN) architecture that has proven highly efficient in detecting and recognizing objects in various types of images with high accuracy. This model was chosen not only for its ability to achieve an optimal balance between performance and computational efficiency, but also because its architecture utilizes a residual learning mechanism, which helps address degradation issues in very deep networks. Thus, ResNet-50 is highly suitable for application in various scenarios, especially in environments with limited computing resources, while maintaining good generalization ability and sufficient inference speed for real-time or large-scale applications. To optimize the learning process, the Adam optimizer is used, known for its adaptive ability to adjust the learning rate during training [8].

### 2.5. Model Evaluation

After the entire model training process is complete, the next crucial step is to conduct an in-depth evaluation to measure the performance and effectiveness of the developed model. This evaluation is conducted using the Classification Report, a comprehensive tool that provides important metrics such as precision, recall, and F1-score for each class in the dataset.[9] These metrics provide a detailed overview of the model's ability to correctly predict the positive class, capture all positive instances, and balance these two aspects. Furthermore, the classification results are clearly visualized in the form of a Confusion Matrix. This matrix is very useful because it visually shows the distribution of the number of correct and incorrect predictions for each category of poisonous ornamental plants, allowing researchers to quickly identify specific error patterns and understand the model's weaknesses in distinguishing certain plant types.

In this study, two deep learning model architectures were used for the purpose of classifying poisonous ornamental plant images: a regular CNN and a CNN combined with a ResNet architecture (CNN using ResNet). The regular CNN model is built with a standard architecture consisting of several convolutional layers, ReLU activation, pooling, and ending with a fully connected layer. This architecture is commonly used for simple image classification tasks and has advantages in terms of ease of implementation and computational efficiency. However, a regular CNN has limitations when the number of layers increases, as it can experience the problem of vanishing gradients, which causes a decrease in accuracy.

In comparison, CNN models using ResNet employ a residual learning approach, where each network block has a shortcut connection that allows information from the previous layer to be passed directly to the next layer. This simplifies the training process of deeper networks by smoothing gradient flow and reducing the risk of performance degradation. In this study, the ResNet architecture was used, which has proven to perform well on various image classification tasks. By comparing the two architectures, it is hoped that it will be possible to determine which model is more effective and accurate in classifying images of poisonous ornamental plants.

### 2.6. Flask Implementation

After model evaluation, we proceeded to implement the best model of the two using Flask to create a user-friendly web application. In this application, users can easily upload images to be classified. Once the image is uploaded, the system processes it and prepares the data to the required format for the model. The model then makes predictions and returns the classification results to the user. These results are presented in an easy-to-understand manner, allowing users to clearly see the predicted class for the uploaded image. Using Flask, we were able to create an interactive and responsive interface, allowing users to interact directly with the model. [10]

## 3. RESULTS AND DISCUSSION

### 3.1 Presentation of Trial Data

This study used a dataset of poisonous and non-poisonous ornamental plants. The dataset consists of a collection of images of various ornamental plants, with a total of 664 original images that have been specifically categorized into 5 classes relevant to identification purposes: Lily, Daffodil, Amazonian Caladium, Sansevieria, and non-poisonous ornamental plants.[11]

Table 1 Sample Dataset

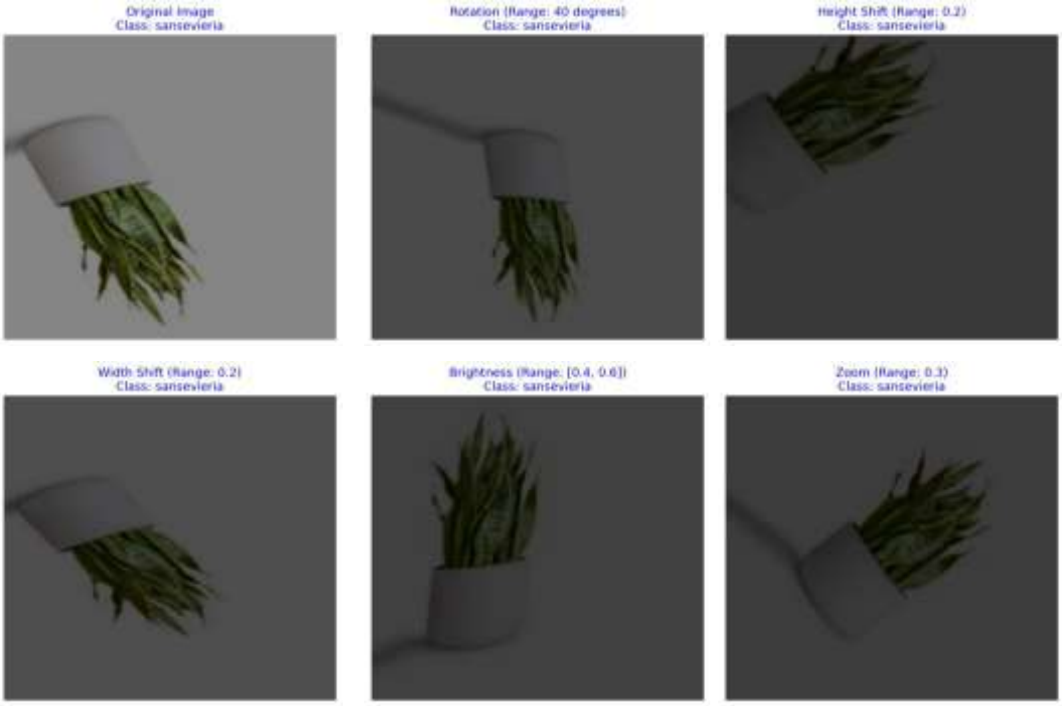
| Dataset Name | Image Dataset |
|---|---|
| Lily |  |
| Daffodils |  |
| Amazon Caladium |  |
| Sensevieria |  |
| Non-Toxic Ornamental Plants |  |

The image data for poisonous and non-poisonous ornamental plants cannot yet be processed into the system due to their different sizes. The citrus image data must be pre-processed.

**1. Preprocessing**

In this section, the obtained data is preprocessed to obtain good image quality to facilitate further processing. This research uses the ImageDataGenerator in Keras. Augmentation is performed by rotating up to 40°, horizontal and vertical shifts of up to 20%, adjusting brightness (0.4–0.6%), rescaling up to 30%, and flipping the image horizontally and vertically. The following is the output of the augmentation visualization on the training data, shown in Figure 2.

Figure 2 Augmentation Results




**2. Model Training**

The training process for a poisonous ornamental plant classification model was conducted to compare the performance of two convolutional neural network architectures: a conventional Convolutional Neural Network (CNN) and a CNN integrated with a Residual Network (ResNet-50) architecture. The dataset used, consisting of 2,656 images of poisonous and non-poisonous ornamental plants, underwent preprocessing and data augmentation to increase diversity and prevent overfitting. The data was divided into 70% for training, 15% for validation, and 15% for testing. Experimental Configuration: Both models were trained using the Adamax optimizer with a learning rate of 0.002 and a categorical cross-entropy loss function. The batch size used was 32, and the training was conducted for 30 epochs.

The CNN model was developed from scratch with a basic CNN architecture that includes a series of convolutional layers, ReLU activations, and pooling, ending with a fully connected layer. The ResNet-50 model adopted a transfer learning approach using a ResNet-50 architecture pre-trained on the ImageNet dataset. Only the last fully connected layer was retrained to adapt to the five classified ornamental plant classes (Lily, Daffodil, Amazonian Caladium, Sansevieria, and Non-Toxic Ornamental Plants). The ResNet-50 architecture used follows a standard design, including convolutional and identity blocks with shortcut connections that allow for better information flow in deep networks.[9]

Training Performance During the training process, the performance of both models was monitored using accuracy and loss metrics on the training and validation data. Figure 3 shows the accuracy and loss trends for the ResNet-50 model. This model exhibits fast and stable convergence, with validation loss steadily decreasing and validation accuracy consistently increasing. Although the graph for the conventional CNN is not explicitly presented, the final comparison shows that the conventional CNN has a higher loss and is prone to overfitting compared to ResNet-50.

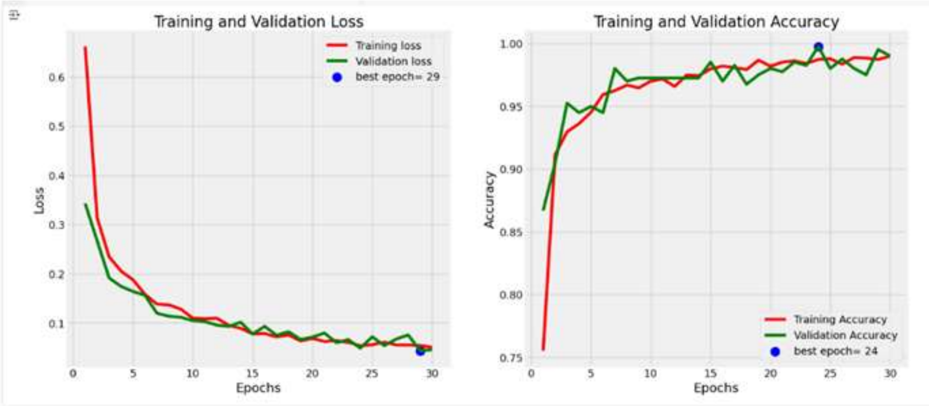



Figure 3 Graph of Accuracy and Loss of Training and Validation of ResNet-50 Model

### 3. Model Evaluation

After the training process was completed, a comprehensive evaluation was conducted to measure the performance and generalization ability of both models on previously unseen test data. This evaluation used standard image classification metrics: accuracy, precision, recall, and F1-score. Model Performance Metrics: Table 2 presents a summary of the key performance metrics for both models on the test data.

Table 2 Comparison of Model Performance on Test Data

| Model | Train / Validation Accuracy | Test Accuracy | Lost (Train, Validation / Test) | Generalization | Overfitting | Additional Information |
|---|---|---|---|---|---|---|
| ResNet 50 | 99,55% | 98,25% | Low | Very Good (Small Difference) | No | Stable Prediction, Low Loss |
| CNN | 91,96% | 87,47% | Tall | Not Good (Big Difference) | A little | Frequently Wrong Predictions, High Losses |

Table 2 clearly demonstrates that the ResNet-50 model significantly outperforms the conventional CNN model in all performance metrics. ResNet-50 achieved a test accuracy of 98.25%, demonstrating its superior ability to classify poisonous and non-poisonous ornamental plants. This accuracy is significantly higher than the 87.47% achieved by the conventional CNN. The high precision, recall, and F1-score for ResNet-50 (each 0.98) also indicate an excellent balance between the model's ability to correctly predict positive classes and its ability to detect all positive instances.

Confusion Matrix Analysis To gain a deeper understanding of the classification performance per class, we analyzed the confusion matrix of the ResNet-50 model. Figure 4 displays the confusion matrix for the ResNet-50 model.

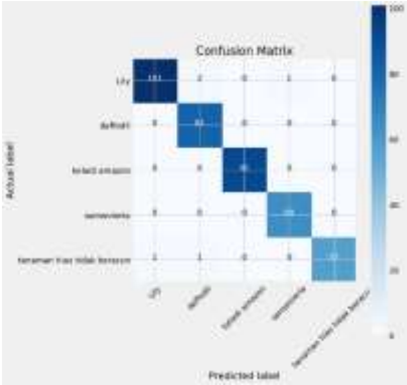



Figure 4 Confusion Matrix of ResNet-50 Model on Test Data

Figure 4 shows that the majority of correct predictions lie along the main diagonal, indicating that the ResNet-50 model is highly effective in identifying each plant class. The number of false positives (FP) and false negatives (FN) is very minimal, confirming the model's reliability.

Classification Report: Further details on performance per class are presented in the classification report. Figure 5 shows the output of the classification report for the ResNet-50 model, demonstrating high precision, recall, and F1-score values for each category of poisonous and non-poisonous ornamental plants. This confirms that the model not only performs well overall but is also able to distinguish between classes with high accuracy.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Lily | 0.99 | 0.97 | 0.98 | 104 |
| daffodil | 0.96 | 1.00 | 0.98 | 81 |
| keladi amazon | 1.00 | 1.00 | 1.00 | 91 |
| sansevieria | 0.98 | 1.00 | 0.99 | 64 |
| tanaman hias tidak beracun | 1.00 | 0.97 | 0.98 | 59 |
| accuracy | | | 0.99 | 399 |
| macro avg | 0.99 | 0.99 | 0.99 | 399 |
| weighted avg | 0.99 | 0.99 | 0.99 | 399 |

Figure 5 Output classification report

## 4. Flask Implementation

After the CNN and ResNet models were successfully saved, the next step was to implement the higher-accuracy model, ResNet 50, into the web using the Flask framework. This application has a function to determine whether a plant is poisonous or not using the previously trained and saved model.[12]


Figure 6. Prediction Results

The image above illustrates the interface of the poisonous houseplant classification app. On the left, there is an option to upload a plant image, which shows a beautiful lily with white petals and green leaves. On the right, the classification results show that the uploaded plant, a lily, was identified as poisonous with an accuracy rate of 87.38%. Important warnings are included, noting that lilies are poisonous to pets and children, indicating that consumption of this plant should be avoided and kept out of their reach. This app is useful for raising awareness about potentially dangerous houseplants.

After the prediction results appear, as shown in Figure 6, this is the final display of the model implementation in Flask development. In the image above, one of the poisonous ornamental plant datasets, the lily, has an accuracy of 87.38% and is proven to be poisonous. Next, the test data will be displayed with incorrect predictions, i.e., when the classification results do not match the original label.

beracun Salah

Table 3. results of incorrect predictions

| Picture | Label | Prediction | Information |
|---|---|---|---|
|  | Amazonian taro | Amazonian taro | Correct |
|  | Lily | Daffodil | Wrong |
|  | Amazonian taro | Non-toxic ornamental plants | Wrong |
|  | Daffodil | Non-toxic ornamental plants | Wrong |
|  | Ornamental Plants Non-toxic | Lily | Wrong |

| Picture | Label | Prediction Results | Information |
|---|---|---|---|
|  | Lily | Non-toxic ornamental plants | Wrong |

After performing the prediction results as in Table 3, it can be concluded that some plants that were initially poisonous became non-toxic, as did the names of the plants. And based on the research results, the ResNet-50 model successfully achieved a classification accuracy of 98.25% in identifying various types of ornamental plants, including poisonous categories such as Lily, Daffodil, Amazonian Caladium, and Sansevieria, as well as non-toxic plants. The advantage of this architecture lies in its effective residual learning mechanism, which is able to minimize overfitting. This is seen from the minimal difference between training accuracy (0.9898) and validation accuracy (0.9899), indicating that this model can generalize well on data that has never been seen before.

The practical implementation of this research was carried out through the development of a web-based application using the Flask framework. The application is designed with a user-friendly interface, making it easy for the public to upload plant images and immediately obtain information about their potential toxicity. This real-time classification feature provides easy access for the wider public to identify poisonous plants, which is crucial for raising awareness of the dangers of certain plants.

While this model demonstrates high performance in controlled environments, it has several limitations that should be considered. The relatively small dataset size and variations in lighting conditions and shooting angles can affect prediction accuracy. Therefore, further development can be done by diversifying the training data, including images from different lighting conditions and shooting angles.

By expanding the dataset and increasing the diversity of training data, it is hoped that the model's performance will improve when faced with a wider variety of real-world conditions. This will enhance the model's generalizability, making it more accurate in various situations and helping the public recognize and avoid poisonous plants more effectively.

## 5. CONCLUSIONS

This study successfully evaluated the performance of the Convolutional Neural Network (CNN) method with and without the ResNet architecture in classifying poisonous ornamental plants, and its implementation in a web application using Flask. The ResNet-50 model demonstrated significant superiority with a test accuracy of 98.25% compared to the conventional CNN model, which only achieved 87.47%, thanks to the effective residual learning mechanism and the transfer learning capabilities of ImageNet weights. The model was integrated into a web application, allowing users to identify poisonous ornamental plants in real time by uploading images. This application contributes to raising public awareness of the risks of poisoning, while strengthening empirical evidence regarding the effectiveness of residual architectures in complex image classification, providing valuable insights for researchers and developers in the field of deep learning. Overall, this study confirms the significant potential of deep learning in creating accurate and reliable visual identification systems, as well as its relevance in the context of public safety.

To improve the classification accuracy of poisonous ornamental plants, it is recommended to develop advanced models using more sophisticated network architectures such as DenseNet or EfficientNet. Furthermore, expanding the dataset with more variations will allow the model to learn from a wider range of examples. Integrating the model into a mobile application using Flask is also important to raise public awareness about the risks of poisoning. Furthermore, future research could compare the performance of this model with other machine learning methods to evaluate its effectiveness.

## 6. REFERENCES


[1] M. M. Hossain, M. B., & Rahman, "Toxicity of Common Houseplants," *J. Environ. Sci. Technol*., vol. 13, no. 5, pp. 345–353, 2020.

[2] A. FIRNANDO, "Penerapan Deep Learning Menggunakan Convolutional Neural Network Dengan Arsitektur Efficientnet Untuk Klasifikasi," *Repository.Uin-Suska.Ac.Id/57861/*, 2021.

[3] S. J. Manek and Supatman, "Classification of Toxic Plants Based on Leaf Image Using Backpropagation Method," 2021.

[4] M. F. NAZULI, M. Fachrurrozi, and M. Q. Rizqie, "Klasifikasi Citra Tanaman Beracun Menggunakan Metode Convolutional Neural Network (Cnn)," *J. Teknol. dan Sist. Komput*., vol. 9, no. 2, pp. 123–130, 2023, [Online]. Available: https://repository.unsri.ac.id/88889/%0Ahttps://repository.unsri.ac.id/88889/3/RAMA_55201_09021181 924004_0222058001_0203128701_01_front_ref.pdf

[5] S. Rizki, M., Pratama, A., & Dewi, "Klasifikasi Tanaman Hias Beracun Menggunakan Convolutional Neural Network dengan Augmentasi Data," *J. Teknol. Inf. dan Ilmu Komput., 2023*.

[6] G. Thiodorus, A. Prasetia, L. A. Ardhani, and N. Yudistira, "Klasifikasi citra makanan/non makanan menggunakan metode Transfer Learning dengan model Residual Network," *Teknologi*, vol. 11, no. 2, pp. 74–83, 2021, doi: 10.26594/teknologi.v11i2.2402.

[7] X. Sun, J., Wang, L., Zhang, Y., & Li, "The Impact of Data Normalization on Deep Learning Model Performance," *J. Mach. Learn. Res*., 2020.

[8] F. Hassan, M., Ali, R., Khan, S., & Ahmed, "Adaptive Learning Rate Optimization using Adam for Deep Neural Networks," *Int. J. Artif. Intell*., 2023.

[9] R. B. Anshori, H. F. Tsp, and T. S. Siadari, "Klasifikasi Citra Kanker Serviks Menggunakan Deep Residual Network," *e-Proceeding Eng*., vol. 8, no. 6, pp. 3163–3170, 2022.

[10] A. Saxena, "An Introduction to Convolutional Neural Networks," *Int. J. Res. Appl. Sci. Eng. Technol*., vol. 10, no. 12, pp. 943–947, 2022, doi: 10.22214/ijraset.2022.47789.

[11] R. Dijaya, "Buku Ajar Pengolahan Citra Digital," *Buku Ajar Pengolah. Citra Digit*., 2023, doi: 10.21070/2023/978-623-464-075-5.

[12] A. Almalki, A. Alzahrani, and M. Alshahrani, "A Comparative Study of Code Editors: Visual Studio Code, Sublime Text, and Atom," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 5, pp. 123–130, 2021, doi: 10.14569/IJACSA.2021.0120516.