APPLIED ARTIFICIAL INTELLIGENCE AND ENGINEERING JOURNAL
E-ISSN: 3124-1212, DOI: 00.000/XXXX.YYYY.II.NNN
Vol. 2 No. 1 2026 pp. 21-34

Deep Convolutional Neural Network for Multi-Class Flower Image Classification Using the Flowers Recognition Dataset

Nadila Arsyifa[1], Cut Nadia[2], Arif Arrafi[3*], Nurmala[4]

[1,2] Program Studi Teknik Informatika, Jurusan Teknologi Informasi dan komputer, Politeknik Negeri Lhokseumawe, Indonesia
[3] Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Unirversitas Hang Tuah Pekanbaru, Indonesia
[4] Program Studi Agribisnis, Fakultas Pertanian, Universitas Malikussaleh, Indonesia

*Corresponding Author: arifarrafi17@gmail.com

Article info: Received 04/03/2026, Revised 26/04/2026, Accepted 30/04/2026

Abstract

Automatic flower species identification plays a crucial role in agriculture, horticulture, and biodiversity conservation, as manual identification is time-consuming and susceptible to errors caused by high intra-class variability and inter-class visual similarity. This study proposes a Convolutional Neural Network (CNN)-based approach for classifying three flower species—sunflower, daisy, and tulip—using the publicly available Flowers Recognition dataset from Kaggle, comprising 2,481 images. The dataset was divided into training (70%), validation (15%), and testing (15%) subsets. The preprocessing pipeline included image resizing to 224 × 224 pixels, pixel value normalization to the range of 0–1, contrast enhancement, and data augmentation through random rotation (±20°), horizontal flipping, zooming, and shearing to improve model generalization. The proposed CNN architecture consists of four convolutional blocks with 32–256 filters, followed by max-pooling, batch normalization, a dropout layer (0.5), fully connected layers, and a Softmax output layer. The model was optimized using the Adam optimizer with categorical cross-entropy loss and trained for up to 50 epochs using Early Stopping and ReduceLROnPlateau callbacks. Experimental results demonstrate that the proposed model achieved a test accuracy of 90%, with 90% precision, 90% recall, and a 90% F1-score. The sunflower class exhibited the highest recall (93%), while most misclassifications occurred between the daisy and tulip classes due to their similar visual characteristics. Furthermore, the regularization strategy effectively mitigated overfitting, as indicated by the relatively small gap between training and validation accuracy (97% and 90%, respectively). These findings demonstrate the effectiveness of CNNs for automated flower image classification and highlight their potential for supporting intelligent flower recognition systems. Future work will focus on expanding the number of flower categories and integrating transfer learning techniques to further improve classification performance.

Keywords: deep learning, flower classification, image classification, daisy, sunflower, tulip, computer vision

1. Introduction

The rapid advancement of computer vision and artificial intelligence has significantly transformed image-based recognition systems across a wide range of disciplines, including agriculture, horticulture, biodiversity conservation, and environmental monitoring. Among these applications, automatic flower species identification has attracted considerable attention because of its potential to support plant taxonomy, precision agriculture, botanical research, and commercial horticulture. Accurate flower identification is essential for monitoring plant diversity, supporting breeding programs, improving crop management practices, and assisting educational and ecological studies. Traditionally, flower identification is performed manually by botanical experts through visual observation of morphological characteristics such as petal shape, color, texture, and floral structure. Although this approach provides reliable results, it is labor-intensive, time-consuming, and highly dependent on expert knowledge, making it impractical for large-scale applications involving thousands of flower images [1].

The increasing availability of digital imaging devices and publicly accessible image datasets has created new opportunities for developing intelligent image classification systems capable of automatically recognizing flower species. Such systems can reduce human effort, accelerate the identification process, and minimize subjective interpretation errors. Automated flower recognition is particularly valuable in modern agriculture and horticulture, where rapid identification supports plant inventory management, disease monitoring, cultivation planning, and product classification. In biodiversity conservation, automatic recognition systems facilitate species documentation and ecological surveys, enabling researchers to process large collections of plant images more efficiently. Furthermore, intelligent flower recognition has promising applications in mobile plant identification systems, smart farming technologies, educational platforms, and e-commerce services involving ornamental plants. Recent advances in deep learning have revolutionized image classification by enabling models to automatically learn hierarchical feature representations directly from image data. Unlike conventional machine learning approaches that rely on handcrafted feature extraction methods such as color histograms, texture descriptors, and shape features, deep learning models perform feature extraction and classification simultaneously within a unified framework. Among various deep learning architectures, Convolutional Neural Networks (CNNs) have emerged as one of the most successful approaches for visual recognition tasks because of their capability to capture spatial patterns, local textures, and high-level semantic information through multiple convolutional layers [2]. CNN-based models have consistently achieved state-of-the-art performance in numerous computer vision applications, including medical image analysis, plant disease diagnosis, facial recognition, object detection, and species classification.

Despite these remarkable achievements, flower image classification remains a challenging computer vision problem. One of the major difficulties arises from the substantial intra-class variation, where flowers belonging to the same species may exhibit considerable differences in color intensity, blooming stage, lighting conditions, viewing angle, scale, background complexity, and image quality. Simultaneously, many flower species exhibit very similar visual characteristics, resulting in low inter-class variation that makes discrimination considerably more difficult. For example, flowers from different species may share similar petal colors, floral symmetry, and overall morphology, causing classification models to produce ambiguous predictions. Additional challenges include image noise, occlusion, cluttered natural backgrounds, and variations in environmental illumination, all of which may negatively affect classification accuracy [3]. Therefore, developing a robust CNN model capable of learning highly discriminative visual features remains an important research challenge.Several previous studies have demonstrated the effectiveness of deep learning for flower recognition using different CNN architectures. Mete and Ensari [4] investigated the application of VGG16 and several CNN models for multi-class flower classification. Their results demonstrated that deep learning architectures successfully extracted representative visual features from flower images and significantly outperformed conventional machine learning methods. Abbas et al. [5] proposed a flower detection framework based on Faster R-CNN with an Inception V2 backbone, achieving mean Average Precision (mAP) values of 83.3% and 91.3%. Their study confirmed that deep learning-based object detection methods can accurately localize and recognize flowers in complex scenes, thereby extending the applicability of CNNs beyond image classification.

In addition to improving recognition accuracy, several studies have focused on reducing computational complexity for deployment on embedded and mobile devices. Gavai et al. [6] implemented MobileNets using the TensorFlow framework for flower classification and demonstrated that lightweight CNN architectures substantially reduced computational cost and inference time while maintaining competitive classification performance compared with larger models such as Inception-v3. Similarly, Lutimath et al. [7] proposed Enhanced MobileNetV2 (EMobileNet), which improved feature extraction efficiency while maintaining a compact model size suitable for industrial and real-time applications. These studies highlight the importance of balancing classification performance and computational efficiency, particularly for practical deployment scenarios.

Another important factor influencing CNN performance is the availability of sufficient and diverse training data. Since collecting large-scale labeled flower datasets is often difficult and expensive, data augmentation has become an effective strategy for increasing dataset diversity and improving model generalization. Labed, Touati, and Dif [8] demonstrated that combining multiple augmentation techniques—including image rotation, horizontal flipping, brightness adjustment, scaling, and geometric transformation—significantly enhanced classification performance, increasing model accuracy from 84% to 99%. These findings indicate that appropriate preprocessing and augmentation techniques enable CNN models to learn more robust and invariant visual representations while reducing the risk of overfitting.

Although previous studies have reported promising results, several research gaps remain. Most existing works employ datasets containing either a very large number of flower categories or only binary classification problems, making it difficult to investigate the discriminative capability of CNNs for visually similar flower species. Furthermore, many studies emphasize the comparison of different CNN architectures without thoroughly optimizing preprocessing strategies, regularization techniques, and network design for a specific classification task. The visual similarity among sunflower, daisy, and tulip flowers presents a particularly challenging problem because these species often share comparable petal structures, color distributions, and floral arrangements while still exhibiting subtle distinguishing characteristics. Consequently, accurate classification requires CNN models capable of extracting fine-grained visual features that effectively capture these subtle differences.

Motivated by these challenges, this study proposes a CNN-based flower image classification framework for recognizing three flower species—sunflower, daisy, and tulip—using the publicly available Flowers Recognition dataset from Kaggle. The proposed framework integrates comprehensive preprocessing procedures, including image resizing, pixel normalization, contrast enhancement, and extensive data augmentation, together with an optimized CNN architecture consisting of multiple convolutional blocks, batch normalization, dropout regularization, and Softmax classification. Early Stopping and ReduceLROnPlateau strategies are further incorporated to improve training stability and prevent overfitting. The performance of the proposed model is comprehensively evaluated using accuracy, precision, recall, F1-score, confusion matrix analysis, and learning curves. The primary contributions of this study are threefold. First, it develops an optimized CNN architecture specifically designed for multi-class classification of sunflower, daisy, and tulip images. Second, it investigates the effectiveness of preprocessing, image enhancement, and data augmentation techniques in improving classification performance and reducing overfitting. Third, it provides a comprehensive experimental evaluation that demonstrates the capability of CNNs to achieve reliable flower recognition performance using a publicly available benchmark dataset. The findings of this research are expected to contribute to the advancement of intelligent flower recognition systems and provide a practical reference for future applications in precision agriculture, horticulture, biodiversity conservation, and computer vision research.

2. Methods

This study develops a Convolutional Neural Network (CNN)-based framework for flower image classification consisting of two main phases: the training phase and the testing phase. During the training phase, the input images first undergo a preprocessing stage to improve image quality and ensure consistency across the dataset. The preprocessing pipeline includes image resizing, pixel normalization, contrast enhancement, and data augmentation to increase sample diversity and improve the model's generalization capability. The augmented images are subsequently used to train the CNN, where hierarchical visual features are automatically learned through successive convolutional layers. The extracted features are then processed by fully connected layers to optimize the network parameters using supervised learning. During the testing phase, unseen flower images are subjected to the same preprocessing procedures as those applied during training to ensure data consistency. The preprocessed images are then forwarded through the trained CNN model, which automatically extracts discriminative visual features and predicts the corresponding flower category using the Softmax output layer. The predicted class is subsequently compared with the ground-truth labels to evaluate the classification performance using quantitative metrics such as accuracy, precision, recall, and F1-score [9]. The overall workflow of the proposed methodology is illustrated in Fig. 1.

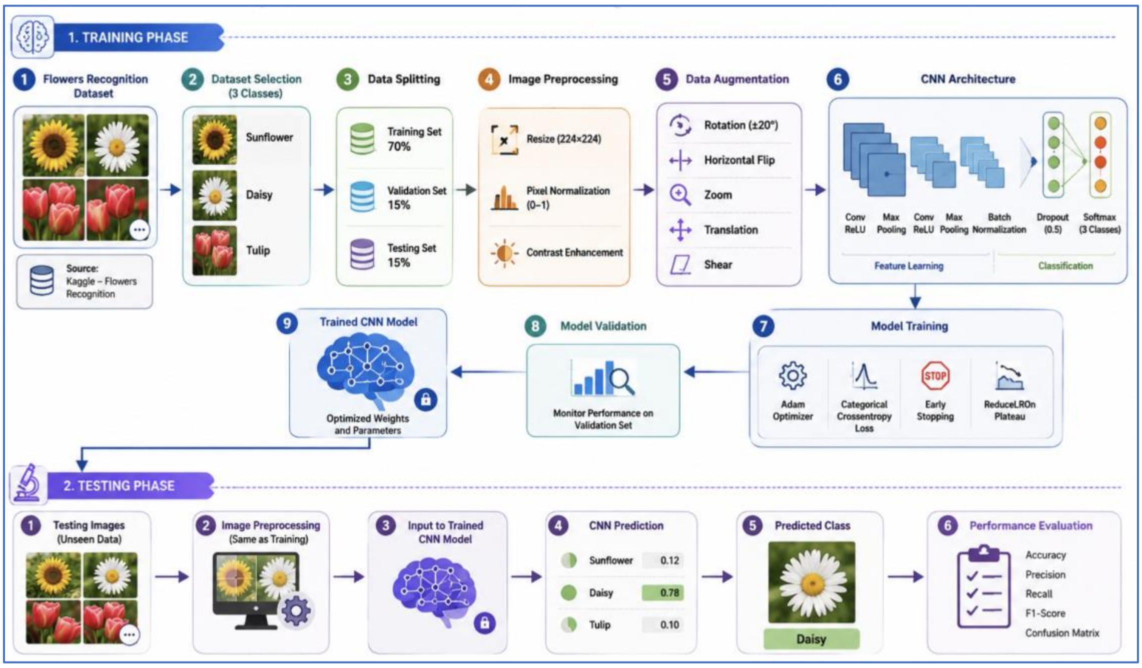


Figure 1. The Proposed Methodology

2.1 Dataset

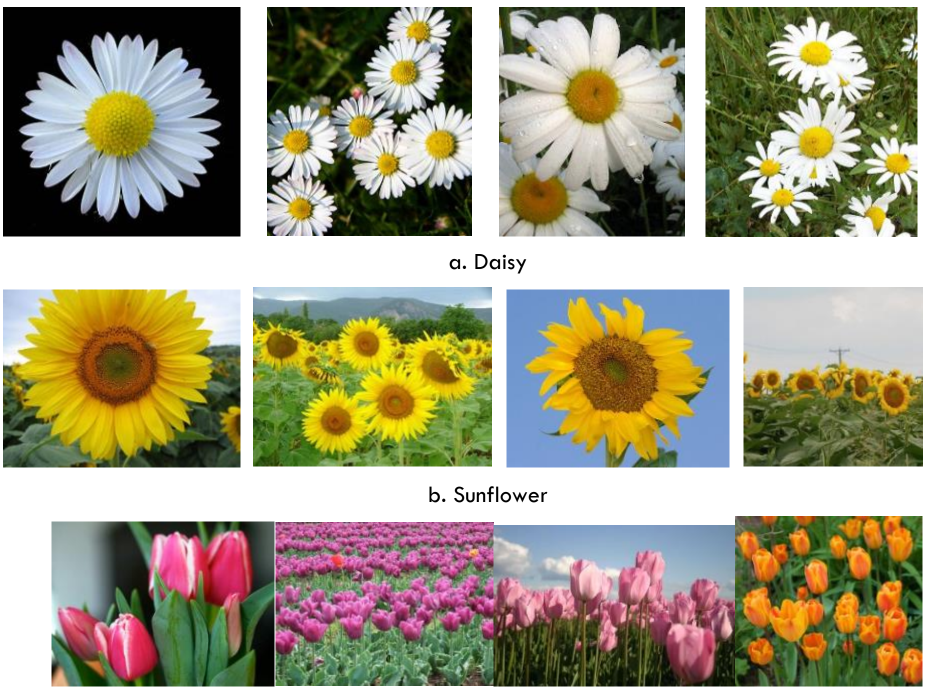
This study employs the Flowers Recognition dataset, a publicly available benchmark dataset widely used for image classification and deep learning research [10]. Owing to its diversity in flower species, imaging conditions, and background complexity, the dataset has become a popular benchmark for evaluating the performance of computer vision algorithms. Several previous studies have demonstrated its effectiveness in assessing convolutional neural network (CNN)-based classification models. For instance, Ayumi et al. [11] investigated flower recognition using ResNet50 and MobileNetV2, demonstrating the capability of transfer learning models to achieve high classification performance. Similarly, another study evaluated several CNN-based transfer learning architectures for flower species classification and reported competitive recognition accuracy [12]. Furthermore, Istiqomah [13] implemented an Inception V4 architecture optimized for deployment on Android devices, highlighting the practical applicability of deep learning models for mobile flower recognition systems. The original Flowers Recognition dataset consists of approximately 4,242 RGB images categorized into five flower species: daisy, dandelion, rose, sunflower, and tulip. The images were collected from various real-world environments, resulting in considerable diversity in image resolution, illumination, viewing angle, object scale, background complexity, and flowering stage. Such variations increase the difficulty of the classification task and provide a realistic benchmark for evaluating the robustness and generalization capability of deep learning models.

Unlike previous studies that considered all five categories, this research focuses exclusively on three flower species, namely sunflower, daisy, and tulip. These classes were selected because they exhibit distinctive yet partially overlapping visual characteristics, making them suitable for evaluating the discriminative capability of CNN models. In total, 2,481 images were utilized in this study, comprising 764 daisy images, 733 sunflower images, and 984 tulip images. Figure 2 presents representative samples from each flower category used in the experiments. Prior to model development, the dataset was randomly partitioned into three mutually exclusive subsets: 70% for training, 15% for validation, and 15% for testing. The training subset was employed to optimize the CNN parameters, the validation subset was used to monitor the learning process and prevent overfitting through model selection, while the testing subset was reserved exclusively for evaluating the final classification performance. This data partitioning strategy ensures an unbiased assessment of the proposed model and is consistent with common practices in deep learning-based image classification. The detailed distribution of images for each subset is presented in Table 1.

Table 1. Distribution of Training, Validation, and Testing Images in the Flowers Recognition Dataset

Class	Images Total	Training (70%)	Validation (15%)	Testing (15%)
Daisy	764	535	114	116
Sunflower	733	513	109	111
Tulip	984	688	147	149
Total	2.481	1.735	370	376

Fig. 2 presents representative images of the sunflower, daisy, and tulip classes from the Flowers Recognition dataset.



a. Daisy

b. Sunflower

c. Tulip

Figure 2. Representative images of the sunflower, daisy, and tulip classes from the Flowers Recognition

2.2 Image Preprocessing

Image preprocessing is a crucial stage in deep learning pipelines, as it enhances image quality and ensures data consistency before model training. The primary objective of preprocessing is to optimize the CNN's ability to learn discriminative visual features while reducing the influence of irrelevant variations such as differences in illumination, image resolution, object scale, and viewing angle [14]. In this study, the preprocessing pipeline consists of image resizing, pixel normalization, and contrast enhancement. These operations standardize the input images and improve the robustness and generalization capability of the proposed CNN model.

2.2.1 Image Resizing

The first preprocessing step involves resizing all flower images to a fixed resolution of 224 × 224 pixels, which corresponds to the input size required by the proposed CNN architecture. Because the original Flowers Recognition dataset contains images with varying spatial resolutions and aspect ratios, resizing is necessary to provide a uniform input dimension for all training and testing samples. Standardizing image dimensions not only simplifies the learning process but also reduces computational complexity and memory consumption during model training [15]. The resizing operation preserves the overall visual characteristics of each flower while ensuring compatibility with the convolutional layers of the CNN. Consequently, the model can efficiently extract hierarchical features from images with a consistent spatial representation. Figure 3(a) illustrates an example of the original flower image before resizing, whereas Figure 3(b) presents the corresponding image after being resized to 224 × 224 pixels. Figure 3. (a) Original flower image before resizing; (b) Flower image after resizing to 224 × 224 pixels.

(a) Original Size (320,240) (b) Resized (224,224)

Figure 3. (a) Original flower image before resizing; (b) Flower image after resizing to 224 × 224 pixels.

2.2.2 Pixel Rescaling (Normalization)

Following the resizing process, all input images were subjected to pixel value rescaling, where each pixel intensity was normalized from the original range of 0–255 to 0–1 by dividing every pixel value by 255. This normalization process standardizes the input data, enabling the CNN to learn more efficiently while improving numerical stability during the optimization process [16]. Pixel normalization offers several advantages in deep learning. First, it prevents large pixel values from dominating the gradient updates during backpropagation, thereby facilitating faster and more stable convergence. Second, normalization reduces the risk of unstable weight updates caused by large variations in input intensity values. Finally, standardized pixel values improve the model's ability to generalize by ensuring that all images share a consistent numerical representation throughout the training and inference stages.

Mathematically, the rescaling operation can be expressed as

$$I_{\text{normalized}} = \frac{I_{\text{original}}}{255}$$

where I_{original} denotes the original pixel intensity in the range of 0–255, and $I_{\text{normalized}}$ represents the normalized pixel value within the interval [0, 1]. Fig. 4 illustrates representative flower images before and after the pixel rescaling process. Although the visual appearance remains almost unchanged to the human eye, the normalized images provide a more suitable numerical representation for CNN training, resulting in improved optimization efficiency and classification performance.

(a) Before Rescaling (b) After Rescaling

Figure 4. Representative flower images before and after pixel value rescaling

2.2.3 Contrast Adjustment

To improve the robustness of the proposed CNN model against illumination variations, contrast adjustment was performed by applying random brightness scaling within the range of 0.8–1.2 during the preprocessing stage. Rather than directly modifying the image contrast using conventional enhancement techniques, brightness variation was introduced to simulate different lighting conditions commonly encountered in real-world image acquisition [17]. This approach enables the model to learn invariant visual representations under varying illumination levels, including images captured in both indoor and outdoor environments. Consequently, the CNN becomes less sensitive to changes in lighting intensity while preserving the discriminative characteristics of flower petals, textures, and color patterns. Improving illumination robustness is particularly important for flower image classification because images in the Flowers Recognition dataset exhibit substantial variability in brightness, shadows, exposure, and environmental conditions. The brightness transformation can be expressed as

$$I_{\text{adjusted}} = \alpha \times I_{\text{original}}$$

where I_{original} represents the original image, I_{adjusted} denotes the brightness-adjusted image, and α is a randomly selected scaling factor within the interval [0.8, 1.2]. Values of $\alpha < 1$ produce darker images, whereas $\alpha > 1$ generate brighter images while preserving the original image structure. Fig. 5 presents representative examples of flower images before and after brightness-based contrast adjustment. The enhanced images exhibit varying illumination levels while maintaining their essential visual characteristics, thereby increasing data diversity and improving the CNN's generalization capability during training.

Figure 5. Contrast Adjustment Results (Range: 0.8 – 1.2)

2.2.4 Image Data Augmentation

Image data augmentation was employed to increase the diversity of the training dataset and improve the generalization capability of the proposed CNN model. Since the Flowers Recognition dataset contains a limited number of images and exhibits considerable variations in flower appearance, augmentation generates additional training samples through label-preserving geometric transformations. This strategy reduces the likelihood of overfitting by exposing the CNN to a broader range of image variations while maintaining the semantic information of each flower class [18]. In this study, augmentation was applied only to the **training dataset**, whereas the validation and testing datasets remained unchanged to ensure an unbiased evaluation of the model. The augmentation process consisted of the following transformations:

- Rotation: Random image rotation within ± 20° to simulate flowers photographed from different viewing angles.
- Translation (Shift): Horizontal and vertical translations of up to 20% of the image dimensions to improve robustness against object displacement.
- Zoom: Random zooming with a scaling factor ranging from 0.8 to 1.2, enabling the model to recognize flowers at different distances and scales.
- Horizontal Flip: Horizontal image flipping to increase orientation diversity while preserving the class label.
- Shear Transformation: Random shearing with a shear intensity of 0.2, introducing slight geometric distortions that improve invariance to perspective changes.

These augmentation techniques enable the CNN to learn more robust and invariant feature representations by simulating realistic image acquisition conditions. Consequently, the trained model becomes less sensitive to variations in object orientation, position, scale, and geometric deformation, thereby improving its ability to generalize to previously unseen flower images. Moreover, data augmentation serves as an effective regularization technique by increasing the variability of the training data without requiring additional image collection. Fig. 6 illustrates representative examples of flower images after applying the augmentation techniques used in this study. The augmented images retain the essential visual characteristics of each flower species while introducing realistic variations that enhance the diversity of the training dataset.

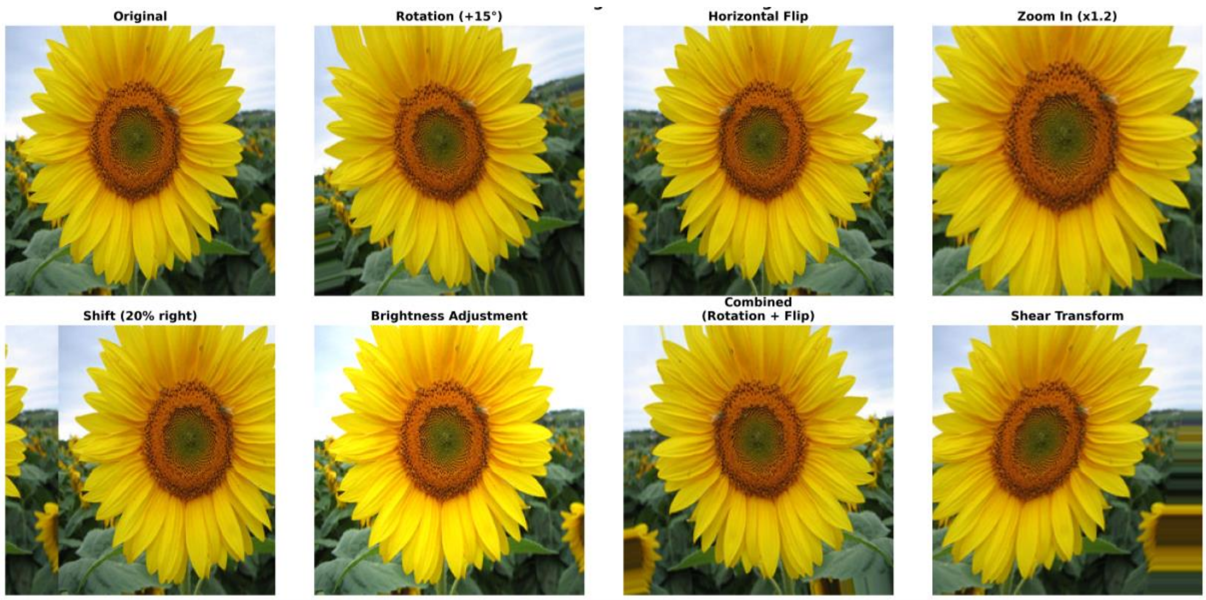


Figure 6. Examples of image augmentation applied to the Flowers Recognition dataset, including rotation, translation, zoom, horizontal flipping, and shear transformation.

2.3 Convolutional Neural Network (CNN)

Following the preprocessing stage, the flower images were classified using a custom Convolutional Neural Network (CNN) specifically designed for three-class flower image classification. CNN is a deep learning architecture that has demonstrated outstanding performance in image recognition tasks because of its capability to automatically learn hierarchical feature representations from raw image data. Unlike conventional machine learning methods that rely on handcrafted feature extraction, CNN progressively learns low-level features, such as edges and textures, followed by more complex semantic representations, including petal structures, flower shapes, and color patterns, through multiple convolutional layers [19].

2.3.1 CNN Architecture

The proposed CNN architecture comprises four convolutional blocks followed by fully connected classification layers. Each convolutional block consists of a 3 × 3 convolutional layer with the ReLU (Rectified Linear Unit) activation function, followed by batch normalization and a 2 × 2 MaxPooling layer. The convolutional layers employ 32, 64, 128, and 256 filters, respectively, enabling the network to progressively learn increasingly discriminative visual representations from low-level image features to high-level semantic information. The convolutional operation uses 3 × 3 kernels with same padding, preserving the spatial dimensions of the feature maps while capturing local spatial patterns effectively. After each convolutional layer, batch normalization is applied to normalize the intermediate feature distributions, thereby improving numerical stability, accelerating convergence, and reducing internal covariate shift during training. Each convolutional block is followed by a MaxPooling2D layer with a 2 ×

2 pooling window, which reduces the spatial resolution of feature maps while retaining the most informative visual features. This operation decreases computational complexity and improves translation invariance by preserving dominant feature responses.

Following the final convolutional block, the extracted feature maps are transformed into a one-dimensional feature vector using a Flatten layer before being forwarded to the fully connected network. Two dense layers are subsequently employed to learn complex nonlinear relationships among the extracted features. To reduce overfitting, a Dropout layer with a dropout rate of 0.5 is inserted after the Flatten layer and between the dense layers. During training, dropout randomly deactivates a subset of neurons, encouraging the network to learn more generalized feature representations rather than memorizing the training samples. Finally, the network terminates with a Softmax output layer consisting of three neurons, corresponding to the three flower classes: sunflower, daisy, and tulip. The Softmax activation function converts the output logits into normalized probability values, enabling the model to assign each input image to the class with the highest predicted probability. The overall architecture of the proposed CNN model is illustrated in Figure 7, while the detailed network configuration is summarized in Table 3.

Table 3. Configuration of the Proposed CNN Architecture

Layer	**Configuration**
Input	224 × 224 × 3
Conv Block 1	Conv (32, 3×3) + BN + ReLU + MaxPool
Conv Block 2	Conv (64, 3×3) + BN + ReLU + MaxPool
Conv Block 3	Conv (128, 3×3) + BN + ReLU + MaxPool
Conv Block 4	Conv (256, 3×3) + BN + ReLU + MaxPool
Flatten	Feature Vector
Dense	512 Neurons + ReLU
Dropout	0.5
Dense	128 Neurons + ReLU
Dropout	0.5
Output	Softmax (3 Classes)

2.3.2 Model Compilation

The CNN model was compiled using the categorical cross-entropy loss function, which is well suited for multi-class classification problems in which each input image belongs exclusively to one class. This loss function measures the discrepancy between the predicted probability distribution generated by the Softmax layer and the corresponding ground-truth class labels. Model optimization was performed using the Adaptive Moment Estimation (Adam) optimizer with an initial learning rate of 0.001. Adam combines the advantages of momentum-based optimization and adaptive learning rates, resulting in faster convergence and improved optimization stability compared with conventional stochastic gradient descent algorithms [20]. To quantitatively evaluate the classification performance throughout the training process, classification accuracy was employed as the primary optimization metric. In addition, the final model performance was assessed using precision, recall, F1-score, and the confusion matrix, providing a comprehensive evaluation of the classifier across all flower categories.

2.3.3 Model Training

The proposed CNN model was trained for a maximum of 50 epochs using a batch size of 32 images. The training process employed several callback mechanisms to improve convergence and prevent overfitting. Early Stopping was utilized to terminate the training process automatically when the validation loss no longer improved after a predefined number of epochs, thereby preventing unnecessary training and reducing overfitting. Model Checkpoint was incorporated to automatically save the model weights corresponding to the best validation performance, ensuring that the optimal model was retained for subsequent evaluation. Furthermore, ReduceLROnPlateau dynamically reduced the learning rate whenever the validation performance stagnated, enabling the optimizer to escape shallow local minima and achieve better convergence [21]. To further enhance model generalization, real-time data augmentation was applied exclusively to the training dataset using the ImageDataGenerator module. During each training iteration, augmented images were generated dynamically through random geometric transformations without permanently increasing the dataset size. In contrast, the validation and testing datasets underwent only the standard preprocessing operations, namely image resizing and pixel normalization, without augmentation. This strategy ensures a fair and unbiased evaluation of the proposed CNN model by assessing its performance on previously unseen images that preserve their original visual characteristics. The complete hyperparameter configuration used for CNN training is summarized in Table 4.

Table 4. CNN Training Hyperparameters

Hyperparameter	**Value**
Input Size	224 × 224
Optimizer	Adam
Learning Rate	0.001
Loss Function	Categorical Cross-Entropy
Batch Size	32
Epochs	50
Activation	ReLU
Output Activation	Softmax
Dropout Rate	0.5
Callbacks	Early Stopping, Model Checkpoint, ReduceLROnPlateau

2.4 Model Evaluation

The performance of the proposed CNN model was evaluated using a comprehensive set of quantitative metrics on the testing dataset, which was not used during either the training or validation stages. Evaluating the model on previously unseen data provides an objective assessment of its generalization capability. The evaluation metrics employed in this study include accuracy, precision, recall, F1-score, classification report, and confusion matrix, which are widely used for multi-class image classification problems [22].

2.4.1 Accuracy

Accuracy measures the overall proportion of correctly classified images among all testing samples and is defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP (True Positive) denotes correctly classified positive samples, TN (True Negative) represents correctly classified negative samples, FP (False Positive) indicates incorrectly classified negative samples as positive, and FN (False Negative) refers to positive samples incorrectly classified as negative.

2.4.2 Precision

Precision evaluates the reliability of positive predictions by measuring the proportion of correctly predicted positive samples among all samples predicted as positive. It is calculated as

$$\text{Precision} = \frac{TP}{TP + FP}$$

A high precision value indicates that the classifier produces relatively few false-positive predictions.

2.4.3 Recall

Recall, also known as sensitivity, measures the ability of the classifier to identify all positive samples belonging to a particular class. It is expressed as

$$\text{Recall} = \frac{TP}{TP + FN}$$

Higher recall values indicate that fewer positive samples are missed during classification.

2.4.4 F1-Score

The F1-score combines precision and recall into a single metric by computing their harmonic mean. It provides a balanced evaluation when both false positives and false negatives are important.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

A higher F1-score indicates better overall classification performance by balancing prediction precision and sensitivity.

2.4.5 Classification Report

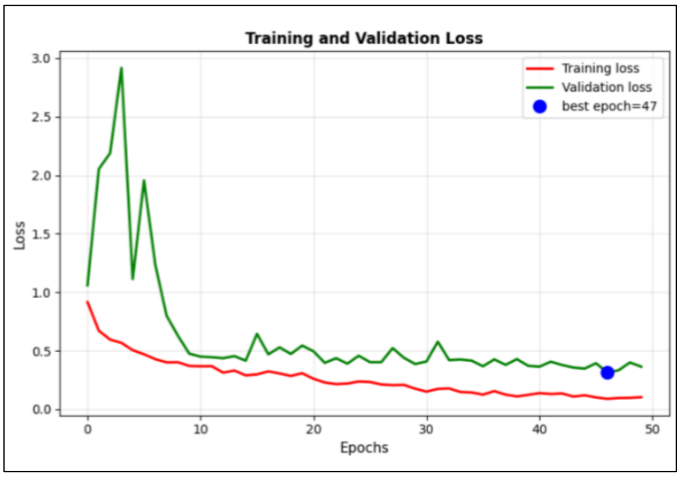
To provide a more detailed performance analysis, a classification report was generated for each flower class. The report summarizes the values of precision, recall, F1-score, and support, where support represents the number of testing samples belonging to each class.

$$\text{Report} = \{\text{Precision}_k, \text{Recall}_k, \text{F1}_k, \text{Support}_k\},\ k = 1, \ldots, n$$

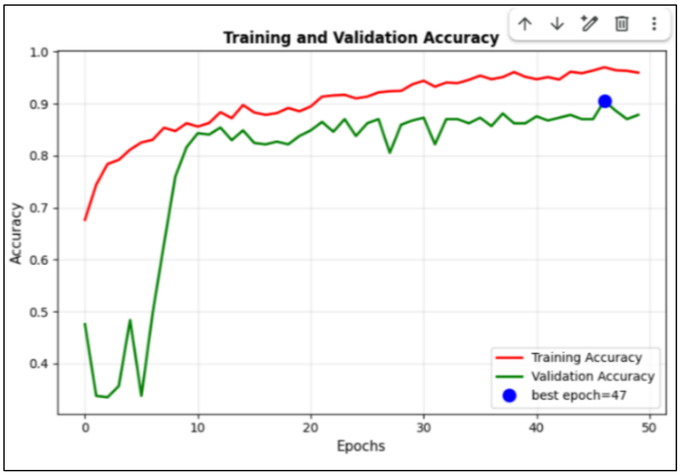
where

$$\text{Support}_k = \sum_{i=1}^{N} I\,(y_i = k)$$

with n denoting the number of classes, N representing the total number of testing samples, and $I(\cdot)$ being the indicator function that counts the samples belonging to class k.



a. Training dan Validation Loss



b. Training dan Validation Accuracy

Figure. 6. Training and Validation Loss and Accuracy

2.4.6 Learning Curves and Confusion Matrix

In addition to the quantitative evaluation metrics, the learning behavior of the proposed CNN model was analyzed through training and validation accuracy as well as training and validation loss curves. These visualizations provide valuable insights into the convergence process and enable the identification of potential overfitting or underfitting during model training. Furthermore, a confusion matrix was generated and visualized as a heatmap to examine the classification performance for each flower category in greater detail. The confusion matrix illustrates the number of correctly and incorrectly classified samples for each class, thereby revealing common misclassification patterns among sunflower, daisy, and tulip images. This analysis provides a more comprehensive understanding of the strengths and limitations of the proposed CNN model beyond the overall accuracy metric.

3. Result and Discussions

The optimal model performance was obtained at epoch 47, where the validation loss reached its minimum value. The learning curves, including the training and validation loss as well as the corresponding accuracy, are presented in Fig. 6. Based on Fig. 6, the model exhibited a positive learning trend throughout the training process. The training loss decreased consistently from approximately 0.90 in the initial epochs to 0.08 at epoch 47, indicating that the model effectively learned the underlying patterns in the dataset. Although the validation loss fluctuated during the first 10 epochs, reaching a peak of approximately 2.90, it gradually stabilized within the range of 0.30–0.60 after epoch 10 and achieved its minimum value of 0.31 at epoch 47. Similarly, the training accuracy improved steadily from approximately 67% to 97%, while the validation accuracy increased substantially from approximately 45% in the initial epochs to 90% at epoch 47, despite minor fluctuations within the 80–90% range during training. Overall, the relatively small gap between the training and validation accuracies (97% versus 90%) indicates good generalization capability with no significant evidence of overfitting.

3.1 Evaluation Results on the Testing Dataset

The model with the lowest validation loss, saved at epoch 47, was subsequently evaluated on the testing dataset comprising 376 images. The evaluation results are summarized in Table 5.

Table 5. Evaluation results on Testing data

Evaluation metrics	Score
Test Loss	0.28
Test Accuracy	0.90

The model achieved a testing accuracy of 90%, demonstrating balanced classification performance across all three classes. The macro-average precision, recall, and F1-score were each 90%, indicating consistent predictive performance without significant bias toward any particular class. The detailed evaluation results for each class are presented in Table 6.

Tabel 6. Clasification report on Testing Data

Class	Precision	Recall	F1-Score	Support
Daisy	0.91	0.88	0.89	116
Sunflower	0.89	0.93	0.91	111
Tulip	0.90	0.90	0.90	149
Macro Avg	0.90	0.90	0.90	376
Weighted Avg	0.90	0.90	0.90	376

3.2 Confusion Matrix Analysis

Based on the confusion matrix analysis, the proposed CNN model demonstrated strong classification performance across the three flower classes. For the Daisy class, which consisted of 116 testing images, the model correctly classified 103 images (88%), while 13 images (11%) were misclassified, including 5 images predicted as Sunflower and 8 images predicted as Tulip. For the Sunflower class, comprising 111 testing images, the model achieved the highest classification performance, correctly identifying 104 images (93%). Only 7 images (6%) were misclassified, with 1 image predicted as Daisy and 6 images predicted as Tulip. The Tulip class contained the largest number of testing images (149 images), of which 133 images (89%) were correctly classified. The remaining

16 images (10%) were incorrectly predicted, including 9 images classified as Daisy and 7 images classified as Sunflower. Overall, the confusion matrix presented in Fig. 7 illustrates that most predictions are concentrated along the main diagonal, indicating a high level of classification accuracy. The relatively small number of off-diagonal predictions suggests that the proposed model effectively distinguishes among the three flower classes, with only minor confusion occurring between visually similar categories.

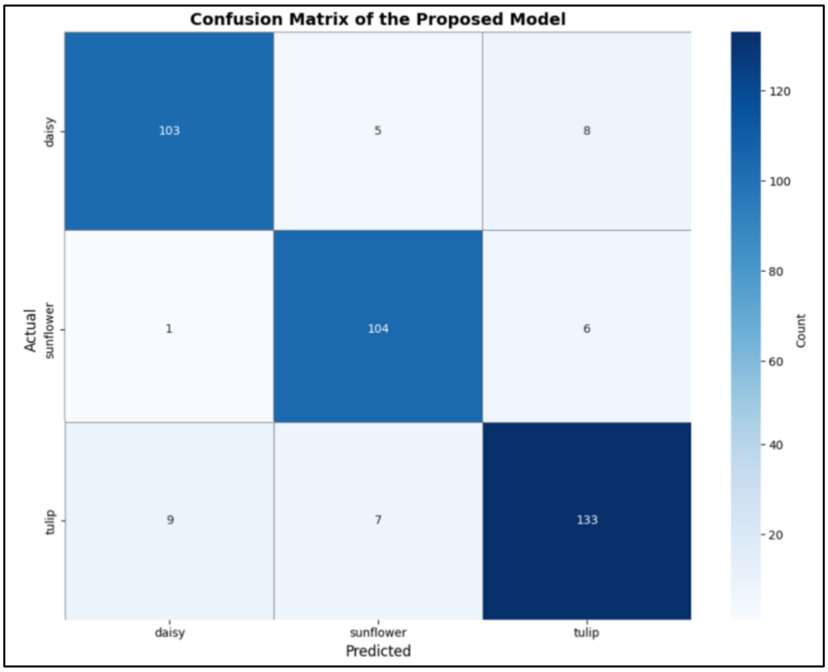


Figure 7. Confusion Matrix Model on Data Testing

Based on the confusion matrix, the highest number of misclassifications occurred between the Daisy and Tulip classes. Several Daisy images were incorrectly classified as Tulip and vice versa. This can be attributed to the similarity of their visual characteristics, particularly in terms of petal shape and flower color, which introduces ambiguity in the features learned by the CNN. In contrast, the Sunflower class exhibited the lowest misclassification rate due to its more distinctive and visually discriminative characteristics, enabling the model to distinguish it more effectively.

4. Conclusion

This study successfully developed a Convolutional Neural Network (CNN) model for classifying three flower species: Daisy, Sunflower, and Tulip. The proposed architecture consists of four convolutional layers with progressively increasing numbers of filters (32, 64, 128, and 256), complemented by batch normalization, max-pooling, and dropout layers to improve learning stability and reduce overfitting. The proposed model achieved a testing accuracy of 90%, with a precision of 90%, recall of 90.58%, and F1-score of 90%. Among the three classes, Sunflower demonstrated the highest classification performance, achieving a recall of 93%, whereas the majority of misclassifications occurred between the Daisy and Tulip classes due to their similar visual characteristics. The experimental results demonstrate that the combination of image preprocessing, data augmentation, and the proposed custom CNN architecture enables accurate flower classification and exhibits good generalization performance on unseen data. Future work may focus on expanding the dataset by incorporating additional flower categories and exploring transfer learning architectures to further improve classification accuracy and model robustness.

Aknowledgements

The authors would like to express their deepest appreciation to Dr. Rahmad Hidayat, S.Kom., M.Cs., as the research supervisor, for his invaluable guidance, constructive feedback, and continuous support throughout this study. His expertise, encouragement, and insightful recommendations have been instrumental in the successful completion of this research.

REFERENCES

[1] C. Narvekar and M. Rao, “Flower classification using CNN and transfer learning in CNN-Agriculture Perspective,” in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, IEEE, Dec. 2020, pp. 660–664. doi: 10.1109/ICISS49785.2020.9316030.

[2] A. V. Sai, M. S. V. Hitesh, V. C. Jadala, S. K. Pasupuleti, S. H. Raju, and M. Shameem, “Flower Identification and Classification applying CNN through Deep Learning Methodologies,” in *2022 International Mobile and Embedded Technology Conference (MECON)*, IEEE, Mar. 2022, pp. 173–180. doi: 10.1109/MECON53876.2022.9752231.

[3] P. Lin, D. Li, Z. Zou, Y. Chen, and S. Jiang, “Deep convolutional neural network for automatic discrimination between Fragaria × Ananassa flowers and other similar white wild flowers in fields,” *Plant Methods*, vol. 14, no. 1, pp. 1–12, 2018, doi: 10.1186/s13007-018-0332-5.

[4] B. R. Mete and T. Ensari, “Flower Classification with Deep CNN and Machine Learning Algorithms,” in *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, IEEE, Oct. 2019, pp. 1–5. doi: 10.1109/ISMSIT.2019.8932908.

[5] T. Abbas *et al.*, “Deep Neural Networks for Automatic Flower Species Localization and Recognition,” *Comput Intell Neurosci*, vol. 2022, pp. 1–9, Apr. 2022, doi: 10.1155/2022/9359353.

[6] N. R. Gavai, Y. A. Jakhade, S. A. Tribhuvan, and R. Bhattad, “MobileNets for flower classification using TensorFlow,” in *2017 International Conference on Big Data, IoT and Data Science (BID)*, IEEE, Dec. 2017, pp. 154–158. doi: 10.1109/BID.2017.8336590.

[7] N. M. Lutimath, B. K. Byregowda, S. V. Seeri, R. V. Manjunath, J. R. Maria Navin, and P. N. Lutimath, “Flower Classification Using Enhanced MobileNetV2 Model for Industrial Application,” 2025, pp. 399–409. doi: 10.1007/978-981-96-2015-9_28.

[8] S. Labed, H. Touati, and R. Dif, “Plant Recognition Using Data Augmentation and Convolutional Neural Network,” 2023, pp. 192–204. doi: 10.1007/978-3-031-18516-8_14.

[9] J. Feng, Z. Wang, M. Zha, and X. Cao, “Flower Recognition Based on Transfer Learning and Adam Deep Learning Optimization Algorithm,” in *Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence*, New York, NY, USA: ACM, Sep. 2019, pp. 598–604. doi: 10.1145/3366194.3366301.

[10] A. Mamaev, “Flowers Recognition,” *Kaggle, Saint Petersburg, Russia.*

[11] V. Ayumi, “Pengenalan Objek Bunga Berbasis Deep Learning Menggunakan Model Resnet50 dan MobileNet-v2,” *JSAI (Journal Scientific and Applied Informatics)*, vol. 5, no. 3, pp. 188–193, Nov. 2022, doi: 10.36085/jsai.v5i3.3613.

[12] A. Mufidatuzzainiya and M. Faisal, “Penggunaan Teknik Transfer Learning pada Metode CNN untuk Pengenalan Tanaman Bunga,” *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 10, no. 2, pp. 195–206, 2025, doi: 10.14421/jiska.2025.10.2.195-206.

[13] A. M. Istiqomah, “Klasifikasi Citra Menggunakan Metode Convolutional Neural Network Arsitektur Inception V4 Berbasis Android Pada Dataset Flower Recognition Ainun Mardiyah Istiqamah,” *Klasifikasi Citra Menggunakan Convolutional Neural Network Dengan Arsitektur Inception V4 Berbasis Android Pada Dataset Flower Recognition*, pp. 13–20, 2020, [Online]. Available: http://repository.unhas.ac.id/id/eprint/804/%0Ahttp://repository.unhas.ac.id/id/eprint/804/2/H13116507_skripsi_ 1-2.pdf

[14] H. Hiary, H. Saadeh, M. Saadeh, and M. Yaqub, “Flower classification using deep convolutional neural networks,” *IET Computer Vision*, vol. 12, no. 6, pp. 855–862, Sep. 2018, doi: 10.1049/iet-cvi.2017.0155.

[15] A. Kutyrev and N. Andriyanov, “Apple Flower Recognition Using Convolutional Neural Networks with Transfer Learning and Data Augmentation Technique,” *E3S Web of Conferences*, vol. 493, p. 01006, Feb. 2024, doi: 10.1051/e3sconf/202449301006.

[16] Y. Wu, X. Qin, Y. Pan, and C. Yuan, “Convolution Neural Network based Transfer Learning for Classification of Flowers,” in *2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP)*, IEEE, Jul. 2018, pp. 562–566. doi: 10.1109/SIPROCESS.2018.8600536.

[17] E. Cengil and A. Cinar, “Multiple Classification of Flower Images Using Transfer Learning,” in *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*, IEEE, Sep. 2019, pp. 1–6. doi: 10.1109/IDAP.2019.8875953.

[18] N. Alipour, O. Tarkhaneh, M. Awrangjeb, and H. Tian, “Flower Image Classification Using Deep Convolutional Neural Network,” in *2021 7th International Conference on Web Research (ICWR)*, IEEE, May 2021, pp. 1–4. doi: 10.1109/ICWR51868.2021.9443129.

[19] C.-H. Ou, Y.-N. Hu, D.-J. Jiang, and P.-Y. Liao, “An Ensemble Voting Method of Pre-Trained Deep Learning Models for Orchid Recognition,” in *2023 IEEE International Systems Conference (SysCon)*, IEEE, Apr. 2023, pp. 1–5. doi: 10.1109/SysCon53073.2023.10131263.
[20] V. S. Dhaka *et al.*, “A Survey of Deep Convolutional Neural Networks Applied for Prediction of Plant Leaf Diseases,” *Sensors*, vol. 21, no. 14, p. 4749, Jul. 2021, doi: 10.3390/s21144749.
[21] P. T. Jackson, A. Atapour-Abarghouei, S. Bonner, T. Breckon, and B. Obara, “Style Augmentation: Data Augmentation via Style Randomization,” 2018, [Online]. Available: http://arxiv.org/abs/1809.05375
[22] E. Ahmed, A. Darwish, and A. E. Hassanien, “Flowers Classification with Low Carbon Footprint Using Deep Learning Pretrained Models,” 2024, pp. 51–69. doi: 10.1007/978-3-031-63451-2_4.