Faktor Exacta

Vol. 19, No. 1, March 2026, pp. 1–15
eISSN: 2502-339X, pISSN: 1979-276X, https://doi.org/10.30998/faktorexacta.v19i1.207

Comparison of Convolutional Neural Network (CNN) Architectures for Sunflower Image Classification

Indah Dwi Febriani[1], Kahfi Heryandi Suradiradja[2]
[1] Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Pamulang, Indonesia
[2] Program Pascasarjana, Program Studi Teknik Informatika, Universitas Pamulang, Indonesia

Article Info

Article history:

Received 04/09/2025
Revised 15/06/2026
Accepted 21/06/2026

Keywords:

Sunflower
CNN
Convolutional Neural Network
Deep Learning
ResNet50

ABSTRACT

Florists must classify sunflower species with similar visual characteristics when customers place orders using flower images as references. The development of deep learning, especially convolutional neural networks (CNNs), is one solution for image classification. This study compares five CNN architectures-ResNet50, VGG16, InceptionV3, NASNet, and MobileNetV2-for classifying images of four sunflower species: Helianthus annuus, Helianthus debilis, Teddy Bear, and Red Sunflower. The research methodology comprises a literature review, data collection, preprocessing, model comparison, evaluation, and selection of the best model from the five CNN architectures. The final stage was the development of an application prototype using the selected model. The results show that the ResNet50 architecture achieved the best performance, with a validation accuracy of 91.66% and the lowest validation *Loss* (0.14). The ResNet50 model was used to develop a web-based application prototype for classifying images of sunflower species. This research supports the use of technology in the creative industry, specifically for florists, in sunflower image classification. Suggestions for further research include exploring InceptionV3, which has the highest validation accuracy (95%); however, its validation *Loss* indicates overfitting (1.86). Further research is needed, including stronger regularization techniques such as dropout and weight decay, as well as a more dynamic learning-rate scheduler.

Corresponding Author:

Kahfi Heryandi Suradiradja,
Fakultas Ilmu Komputer Program Studi Teknik Informatika,
Universitas Pamulang,
Jl. Raya Puspiptek, Buaran, Kec. Pamulang, Kota Tangerang Selatan, Banten
Email: dosen01514@unpam.ac.id

1. INTRODUCTION

The integration of digital imaging technology has become crucial in various domains, including the creative industry, healthcare, and agriculture. Within the creative sector, particularly in floriculture and flower retail, digital image processing plays a significant role in facilitating flower identification and selection. The use of digital images has become an essential part of various sectors, including the creative industry, healthcare, and agriculture. In the creative industry, particularly in floriculture and flower retail, digital image processing technology is highly useful for supporting flower identification and selection.

Florists frequently encounter challenges when customers provide only photographs of flowers without indicating their species or other details. This difficulty is exacerbated by numerous flower varieties that share similar visual characteristics, complicating their identification. One of the main challenges is that customers provide only flower images without specific information on the species. This poses a challenge for florists, as many flower varieties exhibit strong visual similarities [1], [2]. To mitigate this issue, deep learning technology, particularly for the automatic identification and classification of flowers, offers significant advantages in assisting florists in discerning consumer preferences. To address this issue, the application of *deep learning* technology, particularly for automatic flower identification and classification, is highly beneficial for helping florists determine customer needs [3], [4]. The convolutional neural network (CNN) architecture has demonstrated its effectiveness in image recognition and classification, expediting the identification process and reducing human errors. Moreover, CNNs reduce reliance on manual identification, which is both time-intensive and requires specialized knowledge, thereby enhancing efficiency, particularly during periods of high demand. The use of a *convolutional neural network* (CNN) architecture has proven effective for recognizing and classifying images, thereby accelerating the identification process and minimizing human error. Using CNNs also reduces the need for manual identification, which is time-consuming and requires specialized expertise, thereby improving work efficiency, particularly in high-demand situations [1] [5].

In this investigation, the sunflower was selected as the subject due to its distinct and intricate visual features, including variations in petal shape, color, and texture. The primary aim is to evaluate the effectiveness of five convolutional neural network (CNN) architectures—VGG16, ResNet-50, InceptionV3, NASNet Mobile, and MobileNetV2-in classifying images of four sunflower varieties: Red Sunflower, *Teddy Bear*, *Helianthus annuus*, and *Helianthus debilis*. The results will inform the choice of the most effective architecture for a prototype application. The evaluation focuses on three key metrics: validation accuracy, training time, and computational efficiency. These architectures, including VGG16, ResNet-50, InceptionV3, MobileNetV2, and NASNet Mobile, are widely used for visual object classification and have demonstrated strength in various contexts, such as flower classification. In this study, sunflower was selected as the subject because it has unique and complex visual characteristics, including variations in petal shape, color, and texture. The objective of this study is to compare the performance of five CNN architectures—VGG16, ResNet-50, InceptionV3, NASNet Mobile, and MobileNetV2—in classifying images of four types of sunflower: Red Sunflower, *Teddy Bear*, *Helianthus annuus*, and *Helianthus debilis*; the evaluation results will then be used to determine the best-performing architecture to be implemented in a prototype application. The evaluation is conducted based on three main metrics: validation accuracy, training time, and computational efficiency. Architectures such as VGG16, ResNet-50, InceptionV3, MobileNetV2, and NASNet Mobile have been widely used in visual object classification and have demonstrated their respective advantages across various use cases, including flower classification [2], [6], [7], [8], [9] and under different lighting conditions as well as under diverse lighting conditions [1], [10]. This study not only expands the application of artificial intelligence in agriculture and floriculture [11] but also introduces an adaptive subspecies identification system that can enhance operational productivity and business sustainability in the creative industry of florists. and floriculture, this study concretely delivers an adaptive subspecies identification system that can boost operational productivity and business sustainability in the creative industry of flower shops (*florists*).

Previous research has demonstrated the effectiveness of deep learning algorithms in classifying ornamental plant types. However, these studies predominantly address species differentiation at a broad taxonomic level, such as distinguishing between roses, jasmine, and sunflowers. Although several prior studies have successfully classified ornamental plant types using *deep learning* algorithms, most of these studies focus only on species that differ at a macro-taxonomic level (e.g., distinguishing between roses, jasmine, and sunflowers) [1], [2], [12], [13], [14]. A notable gap exists in the exploration of subspecies-level classification, particularly for sunflower varieties (*Helianthus*), where fine-grained visual classification (FGVC) remains largely unexplored despite the high visual similarity among subspecies. There is a *research gap* in subspecies-level classification, with very high visual similarity (fine-grained visual classification, or FGVC) remaining rarely explored, particularly for sunflower varieties (*Helianthus*).

The novelty of this study lies in the formulation of a multi-scale CNN architecture comparison (ranging from lightweight to large-parameter models) specifically designed to address micro-morphological variation across four sunflower subspecies (*Red Sunflower, Teddy Bear, Helianthus annuus,* and *Helianthus debilis*). This approach is arguably crucial because general-purpose flower classification models often fail to capture subtle spatial feature boundaries between these subspecies. In addition, this study not only compares architectural models but also integrates the selected model into a real web-application prototype to support productivity in the creative florist industry.

This study introduces a novel approach by comparing multi-scale CNN architectures, encompassing models ranging from lightweight to large-parameter, specifically designed to address micro-morphological variation across four sunflower subspecies: Helianthus annuus *Red Sunflower, Helianthus annuus Teddy Bear, Helianthus annuus,* and *Helianthus debilis*. This approach is essential because general-purpose flower classification models often fail to capture subtle spatial feature boundaries between these subspecies. Furthermore, this study compares architectural models and integrates the selected model into a prototype of a real web application to enhance productivity in the creative florist industry.

2. METHODS

This study employed a quantitative experimental approach to develop and analyze a sunflower image classification system by comparing the performance of five *convolutional neural network* (CNN) architectures: ResNet50V2, VGG16, NASNetMobile, MobileNetV2, and InceptionV3. The main objective of this study was to evaluate the effectiveness of each model at classifying sunflower images using several evaluation metrics. Figure 1 illustrates the workflow of this study. This study adopted a quantitative experimental design to construct and evaluate a system for classifying sunflower images. It involved a comparative analysis of five convolutional neural network (CNN) architectures: ResNet50V2, VGG16, NASNetMobile, MobileNetV2, and InceptionV3. The primary goal was to assess the classification accuracy of each model using a range of evaluation metrics. The workflow of this study is presented in Figure 1.

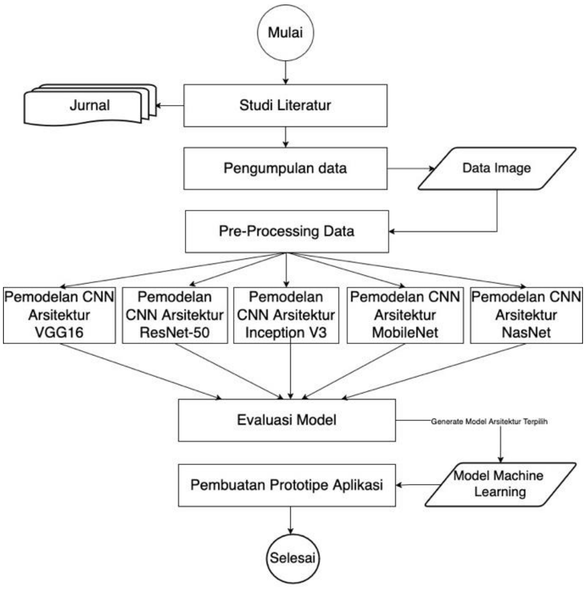


Figure 1. Research Methods

2.1. Literature Review

This phase aimed to gain a comprehensive understanding of the core principles of Convolutional Neural Networks (CNNs), data preprocessing methods, and insights and methodologies from previous research pertinent to image classification. This stage was conducted to obtain an in-depth understanding of the fundamental concepts of *convolutional neural networks (CNNs)*, data preprocessing techniques, and the findings and approaches of prior studies relevant to image classification.

2.2. Data Collection

The sunflower image dataset was sourced from credible platforms and comprised four distinct categories: red sunflower, sunflower *teddy bear*, *Helianthus annuus*, and *Helianthus debilis*. Each category is represented by 70 images, resulting in a total of 280 images (Figure 2). The images, formatted as PNG files, were collected from *Kaggle* and *Google Images. The* sunflower image dataset was collected from valid sources. The dataset comprised four classes: red sunflower, sunflower teddy bear, Helianthus annuus, and Helianthus debilis, with 70 images per class, for a total of 280 images (Figure 2). Furthermore, the dataset was collected from Kaggle and Google Images as PNG-format photos.

Figure 2. Sunflower Dataset Collection Results

2.3. Data Preprocessing

Before using the collected image data for model training, it is essential to preprocess it. This step is crucial for enhancing data quality and ensuring it conforms to the format required by the CNN model. The preprocessing involves several steps that each image must undergo before being introduced into the model. The collected image data requires preprocessing before it can be used for model training. This preprocessing process aims to improve data quality and align it with the format required by the CNN model. Before being fed into the model, all images undergo the following preprocessing steps:

a) *Resizing*, adjusting the image dimensions to 224 × 224 pixels in size.
b) Normalization, which converts pixel values from the range [0, 255] to [0, 1].
c) Real-time Image Augmentation: to improve the model's generalization robustness to variations in lighting and rotation in the field, the following transformations were applied: *random rotation* up to 30 degrees, *width/height shift* of 20%, *shear range* of 20%, *zoom range* of 20%, and *horizontal flip*.

2.4. CNN Modeling

This phase entailed developing five convolutional neural network (CNN) architectures-ResNet50, VGG16, MobileNetV2, InceptionV3, and NASNet-to build various classification models. Each model was trained using an identically preprocessed dataset. The experimental procedure and model training were executed using TensorFlow 2.12 and Keras within the Google Colab environment. All model architectures (ResNet50, VGG16, InceptionV3, NASNetMobile, and MobileNetV2) employed transfer learning with a feature-extraction scheme. The initial weights for each model were derived from ImageNet, with the following architecture. This stage involves developing five CNN architectures-ResNet50, VGG16, MobileNetV2, InceptionV3, and NASNet-to build different classification models. Each model was trained using the same preprocessed dataset. The experimental procedure and model training in this study were conducted using the *framework* TensorFlow 2.12 and Keras on Google Colaboratory, and all model architectures (ResNet50, VGG16, InceptionV3, NASNetMobile, and MobileNetV2) were implemented using a transfer learning strategy with a feature extraction scheme. The *initial weights* for each model were based on ImageNet, with the following architecture:

a) A 2D Global Average Pooling layer was used to reduce the spatial dimensionality of the features.
b) Fully connected Dense layer with 128 neurons and the *rectified linear unit* (ReLU) activation function.
c) A *dropout (rate* = 0.5) layer was used to reduce the risk of *overfitting* during adaptation to new data.
d) An *output dense* layer with four neurons (representing four sunflower subspecies classes), equipped with a softmax activation function to produce a class probability distribution.

The hyperparameters for all model architectures were uniformly standardized as follows: The hyperparameters for all model architectures were standardized as follows:

a) *Optimizer*: Adam with an initial *learning rate* of 0.001.
b) Loss Function: Categorical cross-entropy.
c) *Batch Size*: 32.
d) Epochs: 100 (with an *early stopping* mechanism if the *validation Loss* does not decrease for 20 consecutive *epochs* to avoid unnecessary computation).

2.5. Model Evaluation

Following the training phase, it is crucial to assess the model's effectiveness on test data. This assessment focused on determining the model's ability to accurately classify previously unseen sunflower images. Evaluation metrics such as accuracy, precision, recall, and F1-score were commonly employed. The model demonstrating the best performance based on these metrics is subsequently selected for further implementation. After training, the model's performance needs to be evaluated using test data. This evaluation aimed to measure how well the model could classify sunflower images that it had not previously seen. The commonly used evaluation metrics include accuracy, precision, recall, and F1-score. Based on the evaluation results, the model with optimal performance was selected for the subsequent implementation stage.

2.6. Web Application Prototype Development

The selected model was *generate*d in the Keras JSON model format and then integrated into a web-based application prototype, enabling users to upload sunflower images and automatically obtain classification prediction results.

3. RESULTS AND DISCUSSION

3.1 Results

This study evaluated five CNN architectures-ResNet50V2, VGG16, NASNetMobile, MobileNetV2, and InceptionV3-for classifying sunflower images into four different species. The assessment was conducted using four primary evaluation metrics: accuracy, precision, recall, and F1-score, while also considering training duration and computational efficiency.

During the training stage, a neural network's learning in one iteration (epoch) involves backpropagation, the process of updating weights and biases to minimize the overall loss. *The optimizer* used in this process is Adam, which is known for efficiently accelerating training convergence. The smaller the *Loss* value, the better the model quality; likewise, the higher the accuracy value, the better the model.

After the training process is completed, the model is saved in a file format with a specific extension. The model performance was evaluated using several methods, including *the confusion matrix, mean squared error* (MSE)*, root mean squared error* (RMSE)*, and mean absolute percentage error* (MAPE). In addition, other evaluation metrics, including precision, recall, F1-score, and accuracy, were used to provide a comprehensive overview of each model's classification performance. Based on these evaluation results, the following was obtained:

a) CNN model with the VGG16 architecture

The research observations for the CNN modeling process using the VGG16 architecture are presented in Table 1, followed by a description of the observation results.

Table 1. Evaluation Results for *VGG16*

Epoch	Accuracy (%)	Validation Accuracy (%)	*Loss*	Validation *Loss*
1	0.3335	0.3167	1.7640	1.2406
2	0.5952	0.3833	1.0642	1.0245
3	0.7110	0.5167	0.7332	0.8845
4	0.7958	0.6500	0.5211	0.7239
5	0.9047	0.7333	0.3128	0.5962
…	…	…	…	…
79	0.9229	0.8999	0.2226	0.2799

Figure 3 illustrates the increase in training accuracy, which began at 33.35% during the first epoch and reached 92.29% by the 79th epoch. This progression demonstrates the model's enhanced ability to learn from the training dataset. Similarly, validation accuracy shows a steady rise, starting at 31.67% in the initial epoch and reaching 89.99% by epoch 79, indicating the model's effective generalization beyond mere memorization of the training data. As shown in Figure 3, training accuracy increased from 33.35% in the first epoch to 92.29% at epoch 79. This indicates that the model gradually improved its ability to learn patterns from the training data. The validation accuracy also shows a consistent improvement, starting from 31.67% in the first epoch and reaching 89.99% at epoch 79. This indicates that the model generalizes well and does not merely memorize training data.

Figure 4 shows a progressive reduction in *Loss*, beginning at 1.7640 during the initial epoch and reaching 0.2226 by epoch 79. This decline reflects the model's enhanced accuracy in output prediction, with minimized errors. Concurrently, the Validation *Loss* showed a steady decrease from 1.2406 in the first epoch to 0.2799 at epoch 79, signifying the model's growing proficiency in mitigating overfitting and effectively processing new data. Meanwhile, Figure 4 shows that *Loss* continues to decrease from 1.7640 in the first epoch to 0.2226 at epoch 79, indicating that the model becomes increasingly accurate at predicting outputs as errors decrease. Validation *Loss* also decreases consistently, from 1.2406 in the first epoch to 0.2799 at epoch 79, indicating that the model is increasingly able to avoid overfitting and handle previously unseen data.

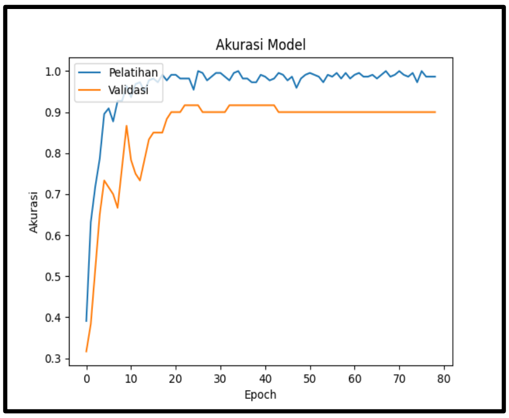


Figure 3. VGG16 Model Accuracy

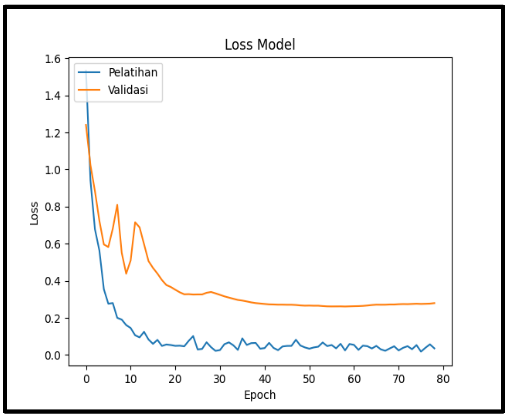


Figure 4. VGG16 Model *Loss*

The evaluation results of the CNN model with the VGG16 architecture in the form of a confusion matrix are shown in Figure 5, and the precision, recall, and F1-score results are presented in Figure 6.

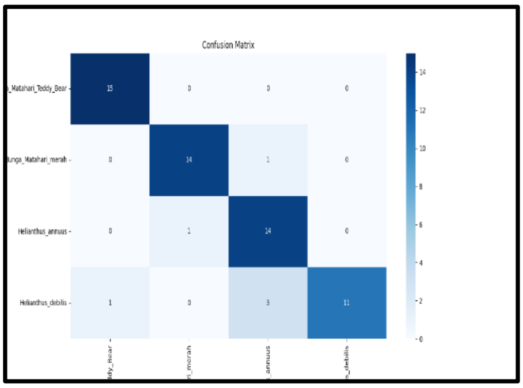
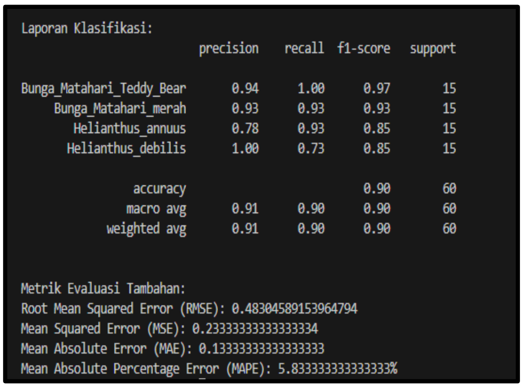


Figure 5. VGG16 Confusion Matrix


Laporan Klasifikasi:

	precision	recall	f1-score	support
Bunga_Matahari_Teddy_Bear	0.94	1.00	0.97	15
Bunga_Matahari_merah	0.93	0.93	0.93	15
Helianthus_annuus	0.78	0.93	0.85	15
Helianthus_debilis	1.00	0.73	0.85	15
accuracy			0.90	60
macro avg	0.91	0.90	0.90	60
weighted avg	0.91	0.90	0.90	60

Metrik Evaluasi Tambahan:
Root Mean Squared Error (RMSE): 0.48304589153964794
Mean Squared Error (MSE): 0.23333333333333334
Mean Absolute Error (MAE): 0.13333333333333333
Mean Absolute Percentage Error (MAPE): 5.833333333333333%

Figure 6. VGG16 Test Report Results

b. CNN Model with the ResNet50v2 architecture

The research observations for the CNN modeling process using the ResNet50v2 architecture are presented in Table 2, followed by an explanation of the results.

Table 2. Accuracy Results for *ResNet50v2*

Epoch	Accuracy (%)	Validation Accuracy (%)	*Loss*	Validation *Loss*
1	0.5667	0.3167	1.4256	4.7293
2	0.9350	0.2500	0.2455	6.6962
3	0.9023	0.2667	0.3301	165.14
4	0.7546	0.3000	0.7060	934.70
5	0.8455	0.2667	0.4425	1167.0
…	…	…	…	…
81	0.9236	0.9166	0.1272	0.1494

Figure 7 reveals that the training accuracy initially undergoes significant fluctuations, beginning at 56.67% in the first epoch and surging to 93.36% by epoch 81. This pattern suggests that the model effectively learned from the training data after an initial period of instability. In contrast, the validation accuracy starts at a low 31.67% in the first epoch but shows consistent improvement, ultimately reaching 91.66% by epoch 81. Despite the early low validation accuracy, the model eventually demonstrated strong generalization capabilities. As shown in Figure 7, training accuracy fluctuated during the initial epochs, with a sharp increase from 56.67% in the first epoch to 93.36% at epoch 81. This indicates that the model made substantial progress in learning patterns from the training data after an inconsistent initial period. The validation accuracy initially decreased significantly to 31.67% in the first epoch, then steadily improved to 91.66% at epoch 81. Although validation accuracy is lower in the early epochs, the model ultimately generalizes well to the data.

Figure 8 shows that the Loss metric exhibits significant fluctuations at the start of training, with notably high values, such as 4.7293 in the first epoch and 165.14 in the third epoch. This pattern suggests that the model initially struggled to learn the appropriate patterns. Nevertheless, from the fourth epoch onward, the *Loss* metric consistently declined, reaching 0.1272 by epoch 81, indicating improved predictive accuracy on the training data. Similarly, the Validation *Loss* also exhibited substantial initial fluctuations, with values such as 4.7293 in the first epoch and 934.70 in the fourth epoch, suggesting poor generalization of the data at the start. However, as with the Loss metric, the validation Loss decreased significantly as training progressed, reaching 0.1494 by epoch 81, demonstrating the model's ability to avoid overfitting and generalize effectively. Figure 8 shows that *Loss* exhibits extreme fluctuations at the beginning of training, with very high *Loss* values (e.g., 4.7293 in the first epoch and 165.14 in the third epoch), which may indicate that the model had difficulty learning appropriate patterns. However, after the fourth epoch, *the Loss* began to decrease steadily, reaching 0.1272 at epoch 81, indicating that the model was becoming increasingly accurate in predicting the training data. Validation *Loss* also shows large fluctuations initially (e.g., 4.7293 in the first epoch and 934.70 in the fourth epoch), indicating that the model did not generalize well to the data at the beginning. However, as with *Loss*, the validation *Loss* also decreased substantially as training progressed, reaching 0.1494 at epoch 81, indicating that the model avoided overfitting and generalized well.

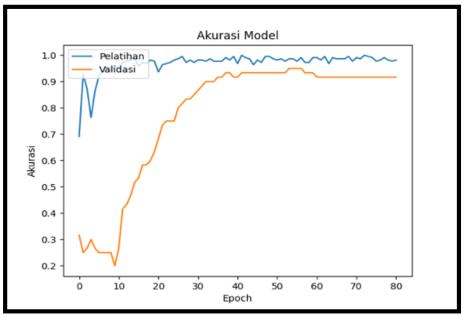


Figure 7. ResNet50v2 Model Accuracy

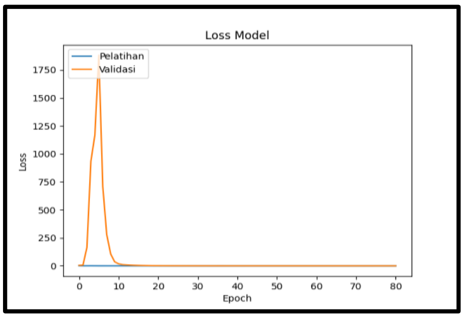


Figure 8. ResNet50v2 Model *Loss*

Figure 8 illustrates the confusion matrix showing the evaluation outcomes of the CNN model using the ResNet50v2 architecture. Figure 9 presents the results for precision, recall, and F1-score. The evaluation results of the CNN model with the ResNet50v2 architecture, presented as a confusion matrix, are shown in Figure 8, and the precision, recall, and F1-score are shown in Figure 9.

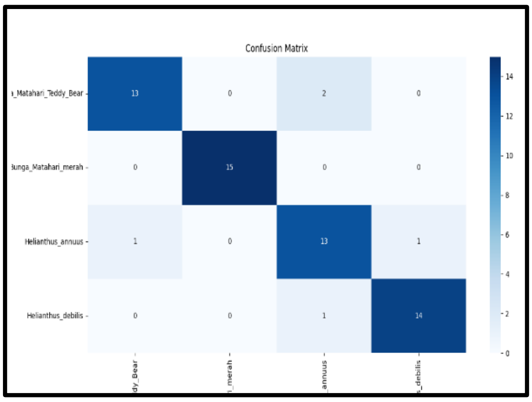


Figure 8. ResNet50v2 Confusion Matrix

```
Laporan Klasifikasi:
                         precision    recall  f1-score   support

Bunga_Matahari_Teddy_Bear     0.93      0.87      0.90        15
     Bunga_Matahari_merah     1.00      1.00      1.00        15
        Helianthus_annuus     0.81      0.87      0.84        15
       Helianthus_debilis     0.93      0.93      0.93        15

                 accuracy                         0.92        60
                macro avg     0.92      0.92      0.92        60
             weighted avg     0.92      0.92      0.92        60


Metrik Evaluasi Tambahan:
Root Mean Squared Error (RMSE): 0.48304589153964794
Mean Squared Error (MSE): 0.23333333333333334
Mean Absolute Error (MAE): 0.13333333333333333
Mean Absolute Percentage Error (MAPE): 9.722222222222223%
```

Figure 9. ResNet50 Model Test Report

c. CNN Model with the InceptionV3 architecture

The research observations for the CNN modeling process using the InceptionV3 architecture are presented in Table 3, followed by an explanation of the results.

Table 3. InceptionV3 Model Evaluation Results

Epoch	Accuracy (%)	Validation Accuracy (%)	*Loss*	Validation *Loss*
1	0.5971	0.3000	1.4351	2.5471
2	0.8935	0.3000	0.4981	4.3764
3	0.8693	0.2500	0.5652	8.7341
4	0.8430	0.3333	0.6335	34.504
5	0.8813	0.2500	0.3489	244.73
…	…	…	…	…
79	0.9642	0.9500	0.0981	0.1866

Figure 10 shows an upward trend in both the training and validation accuracies. Initially, the model's training accuracy was 59.71% in the first epoch, but it significantly increased to 96.42% by epoch 79. This progression signifies the model's enhanced ability to learn from the training data. Concurrently, validation accuracy improved to 95% at epoch 79, reflecting the model's proficiency in generalizing to new, unseen data during training. As shown in Figure 10, both training and validation accuracies improved, with the model achieving a significant increase in training accuracy from 59.71% in the first epoch to 96.42% at epoch 79. This improvement indicates that the model becomes increasingly effective in learning patterns from training data over time. The validation accuracy also showed a positive trend, reaching 95% at epoch 79, indicating the model's ability to generalize well to unseen data during training.

Figure 8 illustrates a significant reduction in both training and validation *Loss*. Specifically, the training *Loss* decreased markedly from 1.4351 in the initial epoch to 0.0981 by epoch 79, suggesting that the model progressively approaches an optimal solution by minimizing errors on the training data. Similarly, the validation *Loss* declined from 2.5471 in the first epoch to 0.1866 at epoch 79, demonstrating that the model not only learns effectively from the training data but also generalizes well to the validation data. This consistent reduction indicates that the model can learn effectively and achieve high performance on both training and validation datasets. Meanwhile, Figure 8 shows a decrease in training and validation *Loss*: the training *Loss* decreased drastically from 1.4351 in the first epoch to only 0.0981 at epoch 79, indicating that the model is increasingly approaching an optimal solution by minimizing errors on the training data. Likewise, the validation *Loss* dropped from 2.5471 in the first epoch to 0.1866 at epoch 79, indicating that the model not only learns well on the training data but also generalizes well to the validation data. This consistent decrease indicates that the model can learn effectively and achieve very good performance on both the training and validation data.

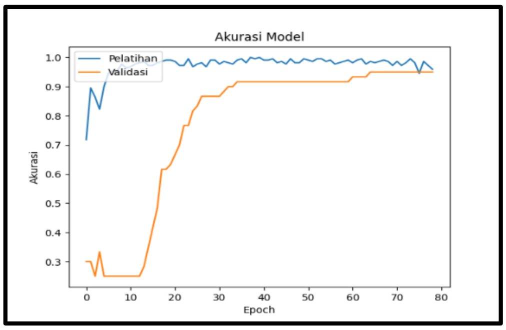


Figure 10. Inception-V3 Model Accuracy

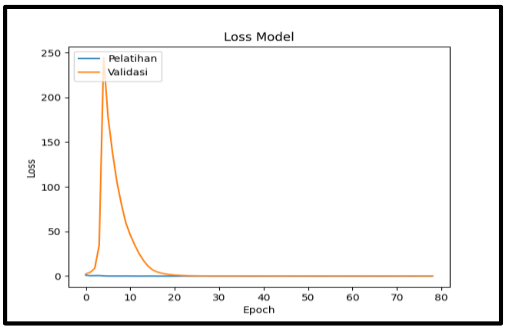


Figure 11. Inception-V3 Model Loss

The evaluation results of the CNN model with the Inception-V3 architecture, in the form of a *confusion matrix*, are shown in Figure 12, and the *precision*, *recall*, and F1-*score* results are shown in Figure 13.

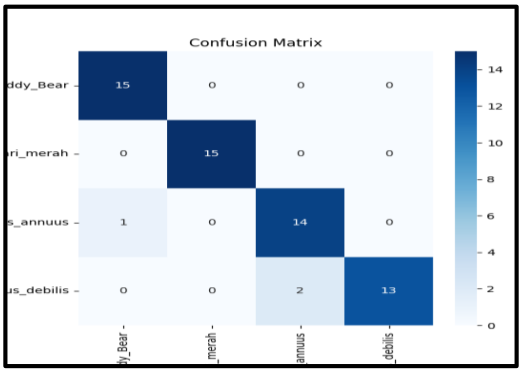
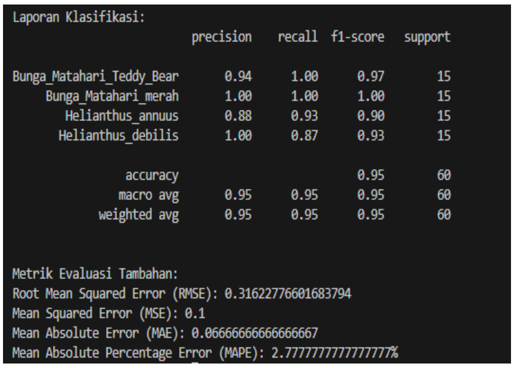


Figure 12. Confusion Matrix Inception-V3



```
Laporan Klasifikasi:
                          precision    recall  f1-score   support

Bunga_Matahari_Teddy_Bear      0.94      1.00      0.97        15
     Bunga_Matahari_merah      1.00      1.00      1.00        15
        Helianthus_annuus      0.88      0.93      0.90        15
       Helianthus_debilis      1.00      0.87      0.93        15

                 accuracy                          0.95        60
                macro avg      0.95      0.95      0.95        60
             weighted avg      0.95      0.95      0.95        60

Metrik Evaluasi Tambahan:
Root Mean Squared Error (RMSE): 0.31622776601683794
Mean Squared Error (MSE): 0.1
Mean Absolute Error (MAE): 0.06666666666666667
Mean Absolute Percentage Error (MAPE): 2.7777777777777777%
```

Figure 13. Inception-V3 Test Report

d. CNN Model with the NASNet Architecture

The research observations for the CNN modeling process using the NASNet architecture are presented in Table 4, followed by a description of the results.

Table 4. NASNet Evaluation Results

Epoch	Accuracy (%)	Validation Accuracy (%)	*Loss*	Validation *Loss*
1	0.5059	0.6000	1.6317	0.8064
2	0.9813	0.7000	0.1931	0.7716
3	0.9490	0.7833	0.1931	0.6382
4	0.9676	0.6667	0.0695	0.6449
5.	0.9775	0.7500	0.0435	0.6594
…	…	…	…	…
39	0.8583	0.8500	0.5384	0.5867

Figure 14 illustrates the progression of training and validation accuracy. The model demonstrated a marked enhancement in training accuracy, beginning at 50.59% in the initial epoch and escalating swiftly to 97.75% by the fifth epoch. Although the validation accuracy exhibited fluctuations, it followed a positive trajectory, rising from 60% in the first epoch to 85% by the 39th epoch. These results suggest that the model effectively learns from the training data and begins to generalize adequately to the validation data, despite some variability during the training process. As shown in Figure 14, training and validation accuracy increased: the model shows a significant improvement in training accuracy, starting at 50.59% in the first epoch and increasing rapidly to reach 97.75% by epoch 5. Validation accuracy also shows a positive trend, although fluctuating, from 60% in the first epoch to 85% at epoch 39. This indicates that the model can learn well on the training data and begins to generalize reasonably well to the validation data, although there are some variations throughout the training process.

Figure 15 illustrates the reduction in both the training and validation *Loss*. The training *Loss* notably decreased from 1.6317 in the initial epoch to 0.0435 by epoch 5, demonstrating the model's effective error reduction on the training dataset. Similarly, the validation *Loss* exhibited a consistent decline, despite some fluctuations, decreasing from 0.8064 in the first epoch to 0.5867 by epoch 39. This trend suggests that the model began to generalize more effectively to the validation data. Figure 15 shows a decrease in training and validation *Loss*: the training *Loss* decreases significantly from 1.6317 in the first epoch to 0.0435 at epoch 5, indicating that the model successfully reduced errors on the training data. The validation *Loss* also showed a steady decline, although there were some fluctuations. The validation *Loss* decreased from 0.8064 in the first epoch to 0.5867 at epoch 39, indicating that the model was starting to generalize better to the validation data.

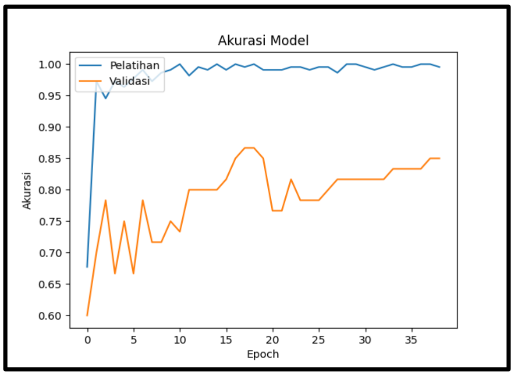


Figure 14. NASNet Model Accuracy

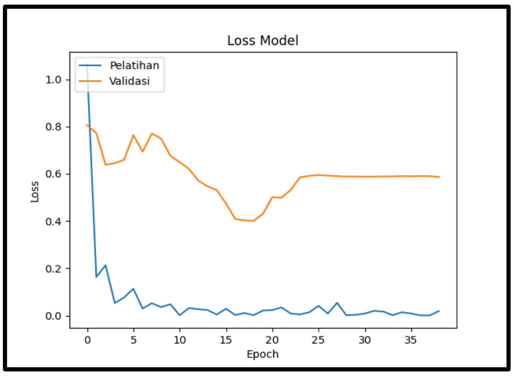


Figure 15. NASNet Model *Loss*

The evaluation results of the CNN model with the NASNet architecture, in the form of a *confusion matrix*, are shown in Figure 16, and the *precision*, *recall*, and F1-*score* results are shown in Figure 17.

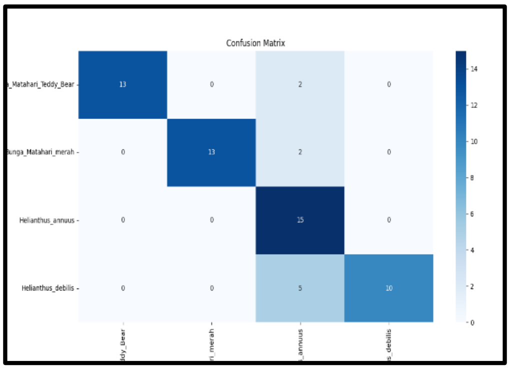
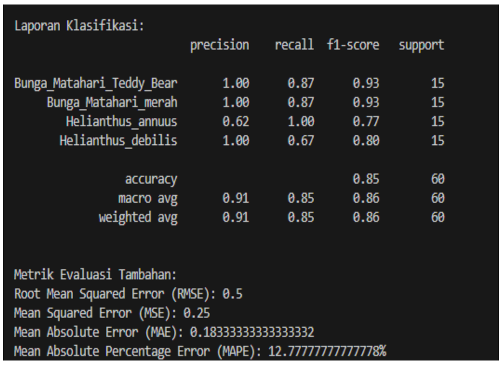


Figure 16. NASNet Confusion Matrix



```
Laporan Klasifikasi:
                          precision    recall  f1-score   support

Bunga_Matahari_Teddy_Bear      1.00      0.87      0.93        15
     Bunga_Matahari_merah      1.00      0.87      0.93        15
       Helianthus_annuus      0.62      1.00      0.77        15
      Helianthus_debilis      1.00      0.67      0.80        15

                accuracy                          0.85        60
               macro avg      0.91      0.85      0.86        60
            weighted avg      0.91      0.85      0.86        60


Metrik Evaluasi Tambahan:
Root Mean Squared Error (RMSE): 0.5
Mean Squared Error (MSE): 0.25
Mean Absolute Error (MAE): 0.18333333333333332
Mean Absolute Percentage Error (MAPE): 12.777777777777778%
```

Figure 17. NASNet Model Test Report

e. CNN model with the *MobileNet*V2 architecture

The research observations for the CNN modeling process using the MobileNet V2 architecture are shown in Table 5, followed by a presentation of the observation results below.

Table 5. MobileNet V2 Evaluation Results

Epoch	Accuracy (%)	Validation Accuracy (%)	*Loss*	Validation *Loss*
1	0.2670	0.2500	2.2169	6.2427
2	0.4232	0.2500	1.3992	6.3680
3	0.4215	0.2500	1.3019	6.6655
4	0.4235	0.2500	1.3607	6.5968
5	0.4934	0.2500	1.1522	6.5650
…	…	…	…	…
16	0.5875	0.2500	0.9929	20.817

Figure 18 shows that training accuracy increased from 26.70% in the initial epoch to 58.75% by epoch 16, suggesting that the model began learning from the training data, albeit slowly. In contrast, the validation accuracy remained stable at 25% during the initial epochs, implying potential challenges in the model's ability to generalize to unseen data. Despite the observed improvements in training accuracy, the lack of significant progress in validation accuracy may indicate overfitting or inadequate generalization. As shown in Figure 18, the training accuracy increased gradually from 26.70% in the first epoch to 58.75% in epoch 16. This indicates that the model began to learn from the training data, albeit with slow improvement. The validation accuracy remained at 25% for the first several epochs, suggesting the model may have difficulty generalizing to unseen data. Despite improvements in training accuracy, validation accuracy did not show significant improvement, suggesting overfitting or insufficient generalization.

Figure 19 illustrates a decline in training *Loss*, which fell from 2.2169 at the first epoch to 0.9929 by the 16th epoch, signifying improved error minimization on the training dataset. In contrast, the validation *Loss* did not show a similar reduction. Instead, it experienced a marked increase during the 16th epoch, reaching 20.817. This increase suggests potential overfitting, where the model may have become too tailored to the training data, thus reducing its ability to generalize to the validation dataset. Meanwhile, Figure 19 shows that the training *Loss* decreased from 2.2169 in the first epoch to 0.9929 in epoch 16, indicating that the model became better at minimizing errors on the training data. Validation *Loss*: The validation *Loss* did not exhibit a consistent decrease. Instead, there was a significant spike in epoch 16, with the validation *Loss* reaching 20.817, which may indicate that the model began to overfit the training data and could not generalize well to the validation data.

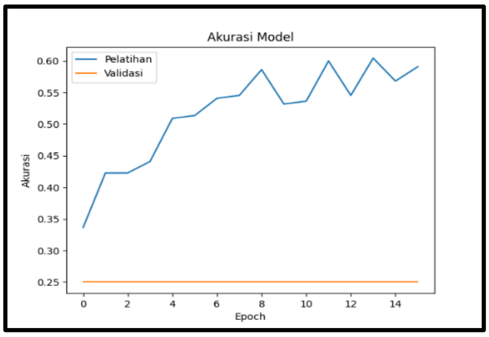


Figure 18. MobileNet Model Accuracy

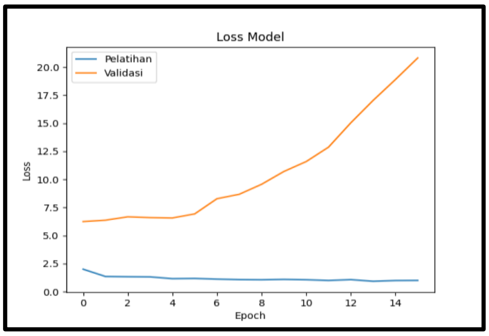


Figure 19. MobileNet Model *Loss*

The evaluation results of the MobileNet CNN architecture in the form of a *confusion matrix* are shown in Figure 20, while the *precision*, *recall*, and F1-*score* results are shown in Figure 21.

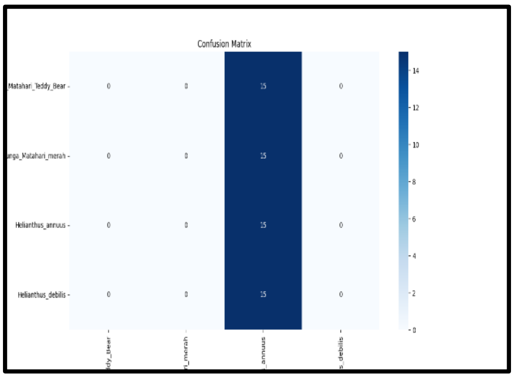
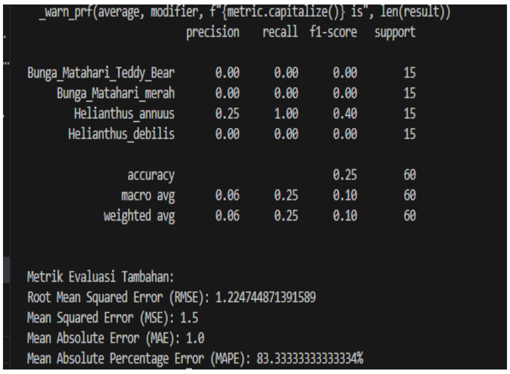


Figure 20. MobileNet Confusion Matrix



```
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
                        precision    recall  f1-score   support

Bunga_Matahari_Teddy_Bear       0.00      0.00      0.00        15
     Bunga_Matahari_merah       0.00      0.00      0.00        15
        Helianthus_annuus       0.25      1.00      0.40        15
       Helianthus_debilis       0.00      0.00      0.00        15

                 accuracy                           0.25        60
                macro avg       0.06      0.25      0.10        60
             weighted avg       0.06      0.25      0.10        60


Metrik Evaluasi Tambahan:
Root Mean Squared Error (RMSE): 1.224744871391589
Mean Squared Error (MSE): 1.5
Mean Absolute Error (MAE): 1.0
Mean Absolute Percentage Error (MAPE): 83.33333333333334%
```

Figure 21. MobileNet Classification Results

From the modeling results of the five convolutional neural network (CNN) architectures, a selection process was conducted to choose the model with the highest accuracy and the lowest *Loss*. The selected model was then generated into the Keras JSON format to be integrated into a web-based application for predicting sunflower type classification (Figure 22).

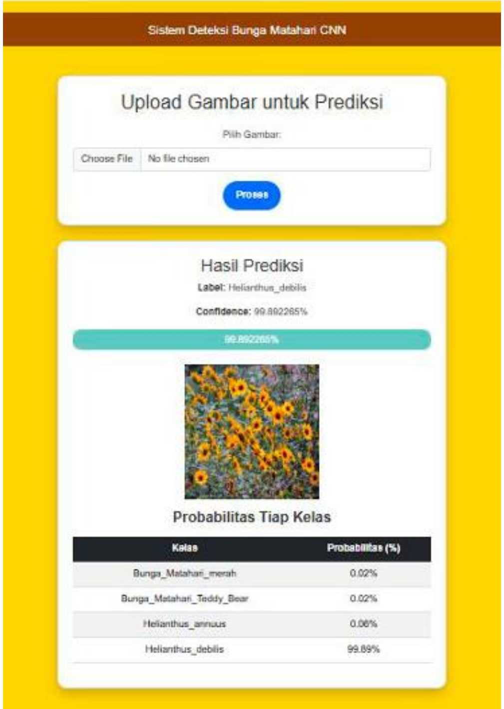


Figure 22. Prototype Application of the Sunflower Detection System

3.2 Discussion

The analysis conducted in this study identifies ResNet50 as the most suitable architecture for classifying sunflower images. The model's validation accuracy of 91.66% and a low validation *Loss* of 0.14, as detailed in Table 6, underscore its proficiency in learning distinct features and its ability to generalize effectively to new data. In contrast, while InceptionV3 achieved a higher validation accuracy of 95% and the lowest error rate, its larger validation *Loss* raises concerns about potential overfitting and less robust generalization compared to ResNet50. Despite the longer training time required for ResNet50, this is considered a reasonable trade-off for its enhanced generalization performance.The comparative analysis reveals that ResNet50 is the most promising architectural choice for sunflower image classification in this study. A substantial validation accuracy of 91.66%, accompanied by a minimal validation *Loss* of 0.14, as shown in Table 6, indicates the model's capability to learn discriminative features effectively and to generalize well to unseen data. Although **InceptionV3** achieved a higher validation accuracy of 95% and the lowest error value, it is important to note that its larger validation *Loss* indicates potential *overfitting* or weaker generalization robustness compared with ResNet50. The *trade-off* of a longer training duration for ResNet50 is considered acceptable given the significant improvement in generalization capability.

Table 6. Comparison Results for Each CNN Model

CNN Architecture	Validation Accuracy (%)	Validation *Loss*	MSE	RMSE	MAE	MAPE (%)	Training Duration
ResNet50	91.66%	**0.1494**	0.2333	0.4830	0.1333	9.722%	60 min 42 s
VGG16	89.99%	0.2799	0.2333	0.4830	0.1333	5.833%	120 min 2 s
InceptionV3	**95.00%**	1.8660	0.1000	**0.3162**	**0.0660**	**2.770%**	60 min 31 s
NASNet Mobile	85.00%	0.5867	0.2500	0.5000	0.1833	12.770%	**25 min 11 s**
MobileNetV2	25.00%	20.8170	1.5000	1.2247	1.0000	83.330%	5 min 36 s

In theoretical terms, while ResNet50 and VGG16 achieve comparable mean squared error values (MSE = 0.2333), ResNet50 demonstrates notably greater stability in validation loss (0.1494) compared to VGG16 (0.2799). This discrepancy arises from the architectural design of VGG16, which consists of a deep linear sequence of traditional convolutional layers lacking shortcut paths, also known as identity shortcut connections. Such a linear configuration is vulnerable to accuracy degradation and the vanishing gradient problem, especially when applied to a small, specific dataset. In contrast, ResNet50 utilizes the principle of Residual Learning, which facilitates direct gradient flow by bypassing one or more layers through an identity addition operation (y = F(x) + x). This approach ensures that essential spatial feature information from the initial layers is maintained up to the final classification layers, thereby promoting more stable generalization of fine-grained visual similarities in sunflowers, while also reducing computation time by approximately half (60 min 42 s).Theoretically, although ResNet50 and VGG16 yield similar squared-error metrics (MSE = 0.2333), ResNet50 exhibits substantially better validation-Loss stability (0.1494) than VGG16 (0.2799). This is attributable to the VGG16 architecture, which employs a very deep linear stack of conventional convolutions without shortcut paths (*identity shortcut connections*). This linear structure is prone to accuracy degradation and the *vanishing gradient* phenomenon when trained on a small, specific dataset. In contrast, ResNet50 leverages the concept of *Residual Learning*, which enables direct gradient propagation to skip one or more layers via an identity addition operation (y = F(x) + x). This mechanism ensures that basic spatial feature information from early layers is preserved up to the final classification layers, thereby enabling much more stable generalization of fine-grained visual similarities in sunflowers, while requiring roughly half the computation time (60 min 42 s).

In this study, two significant anomalies in *Loss* values were identified in the validation data for the InceptionV3 and MobileNetV2 architectures.

a) InceptionV3 *Loss* Anomaly (1.8660)
   InceptionV3 achieved the highest validation accuracy (95.00 %) and the lowest mean squared error (MSE) of 0.1000; however, its validation loss increased significantly. The Categorical Cross-Entropy loss function imposes a substantial logarithmic penalty on incorrect predictions made with high confidence, particularly those nearing 100%. The intricate architecture of InceptionV3, characterized by multi-scale modules that perform parallel extraction using 1×1, 3×3, and 5×5 kernels, contributes to local overfitting. This overfitting arises from the limited diversity in the training dataset, which comprises 280 images. Consequently, the model assigns excessively high confidence probabilities to certain misclassified test samples, thereby elevating the overall accumulated loss value despite maintaining high accuracy. Although InceptionV3 recorded the highest validation accuracy (95.00%) and the smallest MSE (0.1000), its validation *Loss* increased substantially. Mathematically, the *Categorical Cross-Entropy* loss function applies a very heavy logarithmic penalty to *confident incorrect predictions* (misclassifications made with confidence close to 100%). The highly complex InceptionV3 structure with multi-scale modules (parallel extraction using 1×1, 3×3, and 5×5 kernels) causes the model to experience local *overfitting* due to the limited variation in the training data (280 images). This leads the model to assign excessively extreme confidence probabilities to some misclassified test samples, thereby inflating the overall accumulated loss value despite the high accuracy.

b) Extreme MobileNetV2 *Loss* Anomaly **(**20.8170**)**
   The validation loss of MobileNetV2 increased significantly to 20.8170, while the validation accuracy remained notably low at 25.00%. This failure in convergence is intrinsically linked to the architectural design of MobileNetV2, which extensively employs Depthwise Separable Convolutions and Linear Bottlenecks to minimize computational parameters. In the context of a Fine-Grained Classification task with a very limited dataset, the reduction in spatial representational capacity within the depthwise layer leads to a collapse in representation. Lacking sufficient spatial parameters to distinguish subtle inter-subspecies characteristics, the model becomes disoriented in its classification and is confined to a local minimum during the initial epochs.The MobileNetV2 validation loss surged dramatically to 20.8170, while the validation accuracy was extremely low at 25.00%. This convergence failure is closely related to the MobileNetV2 architectural design, which heavily relies on *Depthwise Separable Convolutions* and *Linear Bottlenecks* to reduce computational parameters. In a *Fine-Grained Classification* task with a very limited dataset volume, the reduction in spatial representational capacity in the *depthwise layer* instead triggers *representation collapse*. Without sufficient spatial parameters to distinguish subtle inter-subspecies details, the model becomes disoriented in classification and gets trapped in a local minimum from the early epochs.

4. CONCLUSION

4.1. Conclusion

One of the main limitations of this study is the relatively small dataset, comprising 280 images. Nevertheless, the research offers significant insights into enhancing computer vision systems, particularly in the area of fine-grained image classification. The study demonstrates that, under conditions of limited data, architectures incorporating residual blocks, such as ResNet50, exhibit superior structural resilience to the degradation of micro-feature information compared to ultra-lightweight models like MobileNetV2 or multiscale architectures lacking sufficient spatial regularization, such as InceptionV3. These findings can guide future researchers in selecting the most appropriate base model before compressing it for deployment on smart devices or edge computing. Through the analysis and testing of five convolutional neural network (CNN) architectures for sunflower image classification, ResNet50 was identified as the most effective model. Although InceptionV3 achieved the highest accuracy, it also posed a significant risk of overfitting. Consequently, ResNet50 is recommended as the most reliable model for sunflower image classification, as it offers an optimal balance of accuracy, generalization, and resilience to data variation, making it a strong candidate for practical application in image classification tasks.This study's primary limitation is the relatively small dataset size (280 images), and its findings provide important scientific implications for the development of *computer vision* systems in fine-grained (*fine-grained*) image classification. This study empirically demonstrates that under limited-data conditions, architectures with *residual blocks* (e.g., ResNet50) exhibit substantially better structural robustness against micro-feature information degradation than ultra-lightweight models (e.g., MobileNetV2) or multiscale architectures without adequate spatial regularization (e.g., InceptionV3). These findings can serve as a reference for future researchers in selecting the most suitable base model prior to model compression for deployment on smart devices or *edge computing*. Based on the results of the analysis and testing of five *convolutional neural network (CNN)* architectures for sunflower image classification, ResNet50 was selected as the best-performing model, even though InceptionV3 achieved the highest accuracy but also carried a very high risk of *overfitting*. Therefore, ResNet50 is recommended as the most reliable model for sunflower image classification because it demonstrates the best balance among accuracy, generalization, and robustness to data variation, making it a strong candidate for practical implementation in image classification applications.

4.2. Recommendations

Future research should consider the following recommendations: Despite InceptionV3 achieving the highest validation accuracy at 95%, the elevated validation loss of 1.86 suggests a significant overfitting issue. Consequently, it is advisable to implement more stringent regularization techniques. These may include increasing the dropout rate, applying weight decay, or utilizing a more dynamic learning rate scheduler. Such measures aim to stabilize model performance, particularly to optimize InceptionV3, which, while exhibiting high accuracy and low error values, also presents the highest validation loss.Recommendations for future research include the following: although InceptionV3 produced the highest validation accuracy (95%), the inflated validation loss (1.86) indicates a serious *overfitting* problem. Therefore, it is recommended to apply stricter regularization techniques, such as increasing the *dropout rate*, applying *weight decay*, or using a more dynamic *learning rate scheduler* to stabilize model performance, particularly to optimize InceptionV3, which has high accuracy and a low *error* value but the highest validation *Loss*.

REFERENCES

[1] W. Nengsih and S. Yulina, "Optimization of a CNN Model for Flower Type Identification Based on the Color Spectrum," J. Komput. Terap., vol. 10, no. 1, pp. 57–66, 2024, doi: 10.35143/jkt.v10i1.6274.

[2] A. N. R. Munandar and A. F. Rozi, "Analysis of Convolutional Neural Network Architecture for Flower Image Classification," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 6, no. 3, pp. 522–531, 2024, doi: 10.47233/jteksis.v6i3.1413.

[3] D. Efendi, J. Jasril, S. Sanjaya, F. Syafria, and E. Budianita, "Application of the Convolutional Neural Network Algorithm with the ResNet-50 Architecture for Beef and Pork Image Classification," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 3, p. 607, 2022, doi: 10.30865/jurikom.v9i3.4176.

[4] R. A. Kurniawan, A. Sunyoto, and A. Nasiri, "The Effect of Convolutional Neural Network Architecture on Tomato Leaf Disease Classification (Effect of Convolutional Neural Network Architecture for Tomato Leaf Disease Classification)," pp. 1–6, 2021, [Online]. Available: https://www.kaggle.com/datasets/emmarex/plantdiseas

[5] E. Predianto and B. Sutomo, "Classification of Flower Types Using a Convolutional Neural Network (CNN)," *Tek. Teknol. Inf. dan Multimed.*, vol. 2, no. 2, pp. 64–68, 2021, doi: 10.46764/teknimedia.v2i2.39.

[6] A. Rahman, M. Salim, and I. Riadi, "Classification of Flower Species Images in Indonesia Based on a Convolutional Neural Network Using Transfer Learning Techniques," vol. 2, no. 2, pp. 92–100.

[7] F. Felix, J. Wijaya, S. P. Sutra, P. W. Kosasih, and P. Sirait, "Implementation of a Convolutional Neural Network for Plant Type Identification Through Leaves," *J. SIFO Mikroskil*, vol. 21, no. 1, pp. 1–10, 2020, doi: 10.55601/jsm.v21i1.672.

[8] I. Gilang Perwati, N. Suarna, and T. Suprapti, "Analysis of Lily Flower Image Classification Using the Convolutional Neural Network (CNN) Method in Image Processing," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 2908–2915, 2024, doi: 10.36040/jati.v8i3.9193.

[9] A. F. Setyawan, R. A. Hasani, and E. R. Arumi, "Classification System for Food Crop Diversity Using

Transfer Learning with a CNN Approach and Architecture Models," vol. 6, no. 1, pp. 66–75, 2024, doi: 10.47065/josh.v6i1.5577.

[10] A. Pratiwi and A. Fauzi, "Implementation of Deep Learning for Flower Classification," *J. Tek. Inform.*, vol. 5, no. 2, pp. 487–495, 2024.

[11] K. H. Suradiradja, I. S. Sitanggang, L. Abdullah, and I. Hermadi, "Convolutional neural network for estimation of harvest time of forage sorghum (Sorghum bicolor) cultivar Samurai-1," International Journal of Electrical and Computer Engineering (IJECE), vol. 14, no. 2, p. 1730, Apr. 2024, doi: 10.11591/ijece.v14i2.pp1730-1738.

[12] R. Kaur, A. Jain, and S. Kumar, "Optimized Classification of Sunflower Recognition Through Machine Learning," *Materials Today: Proceedings*, vol. 51, no. 1, pp. 207–211, 2022, doi: 10.1016/j.matpr.2021.05.182.

[13] F. R. Sams, S. K. Supti, S. B. Hamid, R. Junayed, K. M. F. A. Bari, M. J. Ali, R. Gani, K. Shams, M. R. Rashid, and R. U. Islam, "Real-time sunflower detection using semi-supervised and self-supervised deep learning for precision agriculture," *Smart Agricultural Technology*, vol. 13, p. 101684, 2026, doi: 10.1016/j.atech.2025.101684.

[14] M. Barrio-Conde, M. A. Zanella, J. M. Aguiar-Perez, R. Ruiz-Gonzalez, and J. Gomez-Gil, "A Deep Learning Image System for Classifying High Oleic Sunflower Seed Varieties," *Sensors (Basel, Switzerland)*, vol. 23, no. 5, p. 2471, 2023, doi: 10.3390/s23052471.