International Journal of Horticultural Science and Technology
Journal homepage: http://ijhst.ut.ac.ir

Hybrid VGG16-Xception Model vs. Single Architecture Transfer Learning for Flower Image Classification

Ravikiran H K [1]*, Jayanth J[2], Wilfred John Vaz[1], Sathisha M S[3], Prashantha S J[3], Madhu K M[4]

1 Department of Electronics and Communication Engineering, Navkis College of Engineering, Hassan-573217, Karnataka, India
2 Department of Electronics and Communication Engineering, GSSSIETW, Mysuru-570016, Karnataka, India
3 Department of Computer Science and Engineering, Navkis College of Engineering, Hassan-573217, Karnataka, India
4 Department of Civil Engineering, Rajeev Institute of Technology, Hassan-573201, Karnataka, India

*Corresponding author's email: Ravikiranhsn@gmail.com

ARTICLE INFO

Article history:

Received: 22 January 2024,
Received in revised form: 12 April 2024,
Accepted: 17 April 2024

Article type:

Research paper

Keywords:

Feature extraction,
Flower classification,
Transfer learning,
VGG16,
Xception

ABSTRACT

In recent years, the application of deep learning models has significantly advanced the field of computer vision, enabling automated recognition and classification of various objects, including flowers. This research begins with exploring two distinct pre-trained convolutional neural networks (CNNs): VGG16 and Xception. Each model has architecture and performance characteristics that are analyzed and compared to establish a baseline for flower species classification. To enhance classification performance further, we introduce a hybrid model that fuses the extracted features from VGG16 and Xception. These features are concatenated and fed into a dense layer with ReLU activation, followed by a softmax classifier, which leverages the combined knowledge of hybrid models to classify various species of flowers accurately. Experimental results are presented on a benchmark flower dataset from Kaggle, demonstrating the effectiveness of the proposed hybrid model in achieving state-of-the-art classification accuracy. The results highlight the performance of the proposed hybrid model for 25 epochs with 512 dense layers, showcasing a remarkable state-of-the-art classification accuracy of 91.20% on the Kaggle flower dataset. The comprehensive evaluation includes quantitative metrics such as accuracy, precision, recall, and F1-score, highlighting how robust the model is and its generalization capabilities. The findings in this research can assist in developing deep learning-based flower species classification systems.

Database: https://www.kaggle.com/datasets/kausthubkannan/5-flower-types-classification-dataset

Introduction

Flowers, with their vast diversity in shapes, colors, and structures, are not just symbols of beauty but also play crucial roles in industries such as pharmaceuticals, cosmetics, and agriculture (Ari Peryanto et al., 2022). Researchers consider classification a principal task due to the numerous uses and wide variety of flowers. Traditional methods of flower classification primarily revolved around manual feature extraction, emphasizing aspects such as color, texture, and shape (Xiaoxue Li et al., 2021). This process, though functional, is labor-intensive, subjective, and can often be limited by the expertise of the individual involved. With more than 250,000 known species of flowering plants, manual classification, even with conventional image processing techniques, becomes a daunting and error-prone task, especially in intricate datasets where flowers share similar features (Burhan Duman et al., 2022).

Enter the era of deep learning and artificial intelligence, which has provided transformative solutions across various domains, including image classification. Unlike conventional image processing methods, deep learning models, especially convolutional neural networks (CNNs), can automatically and adaptively learn and extract hierarchical features from raw image data. Among such models, the deep learning architecture of convolutional neural networks has been identified as a potent tool for large-scale image classification tasks, eliminating the challenges and inefficiencies associated with manual feature extraction (Touqeer Abbas et al., 2022). These networks excel in processing multidimensional signals and, through convolution, extract relevant features that lead to significant performance improvements in tasks such as flower classification.

Various deep learning architectures, like VGG16 and Xception, have emerged recently. They offer advanced capabilities in feature extraction and classification tasks. Moreover, the evolution of hybrid models that leverage multiple networks has further enhanced the accuracy and robustness of classification systems. This work explores the potential of these architectures, individually through transfer learning and in a hybrid setting as a feature extractor, to classify different species of flowers, aiming for a high accuracy, efficient, and robust solution that could revolutionize flower classification across diverse applications.

Thus, the objective of the proposed work is:

• Investigate the performance of prominent deep learning architectures, such as VGG16 and Xception, through transfer learning in flower classification.

• Assess the feature extraction capabilities of hybrid VGG16 and Xception architecture, identifying the strengths and weaknesses pertinent to the classification task.

• Compare the performance of the hybrid VGG16 and Xception architecture with that of the Neural Network, Random Forest Classifier with Histogram-Based Image Features, VGG16 and Xception Transfer Learning models.

The present article maintains the following approach: Section 1 expands on the introduction, Section 2 explores the literature survey, Section 3 details the database, Section 4 outlines basic concepts and methodology, and Sections 5 and 6, respectively, showcase the results and conclude the study.

Flower image classification has witnessed substantial advancements by integrating deep learning methodologies. Numerous studies have highlighted the importance of Convolutional Neural Networks (CNNs), utilizing various models like MobileNet, DenseNet, Xception, Inception, ResNet, AlexNet, and VGG16.

In the context of these studies, the research (Ari Peryanto et al., 2022) delves into a comparative analysis between Convolutional Neural Networks (CNN) and Support Vector Machine (SVM) in image classification. It distinguishes CNN as a deep neural network approach and SVM as a machine learning algorithm, setting the stage for their comparison. The summary of findings highlights CNN's superiority over SVM in classifying flower images, boasting an impressive accuracy, precision, recall, and F1 Score of 91.6%. Additionally, the study details the CNN modeling process, emphasizing incremental enhancements in accuracy and loss across epochs while addressing concerns related to overfitting.

This research study (Burhan Duman et al., 2022) involves flower species detection, leveraging various deep learning models, specifically MobileNet, DenseNet, Inception, and ResNet. Thus, two datasets were employed, i.e., the 5-class Flower Dataset and the 17-class Oxford-17 Dataset. These models were usable via Python and TensorFlow2 on Kaggle and Google Colab platforms. A comparative performance assessment appeared while focusing on different optimizers, i.e., Adam and SGD, across various training epochs. The outcomes illuminated the variability in performance based on the chosen deep learning model, the optimizer, and the dataset size. For instance, the Adam optimizer generally outperformed the SGD in both datasets. Interestingly, several models like Mobilenet-v2, Resnet152v2, Inceptionv3, and DenseNet169 exhibited similar accuracy when comparing the datasets. However, InceptionResnetv2 displayed relatively lower accuracy for the 17-class dataset than the 5-class dataset. Conclusively, the number of classes (5 vs. 17) didn't dramatically influence model accuracy across optimizers. Future research might extend this investigation by incorporating larger datasets like Oxford-102 and experimenting with other deep-learning models and optimizers.

In this study (Zhao Jiantao et al., 1994), Convolutional Neural Networks (CNNs) were harnessed to classify flower images obtained through mobile devices, addressing the inherent complexities of natural conditions, including background interference and the intricate variability among different flower types. The research emphasized the pivotal role of selecting an optimal learning rate during the training process, as it significantly influences the convergence and efficiency of the neural network. Finding the right balance in the learning rate is crucial, as setting it too low can lead to slow convergence while setting it too high risks overshooting the optimal solution. Furthermore, researchers can overcome overfitting challenges via data augmentation as a preventive strategy. However, the study noted that fluctuations in accuracy and loss could occur during training, particularly when maintaining a constant batch size. These fluctuations emanated from the introduction of variations through data augmentation. The problem-solving key involved adjusting batch sizes to accommodate the increased data volume after augmentation. In summary, this study demonstrated the effectiveness of CNNs in improving flower image classification accuracy, offering valuable insights into the importance of learning rates and the benefits of data augmentation for addressing overfitting in limited and diverse datasets.

This study focused on flower recognition using Deep Convolutional Neural Networks with a transfer learning approach. Two popular CNN models, AlexNet and VGG16, were employed and evaluated on a benchmark Kaggle dataset. The results demonstrated the effectiveness of CNNs in object recognition, with VGG16 outperforming AlexNet, achieving an accuracy of 95.02%. Notably, VGG16 excelled in recognizing distinct species, while it showed moderate performance for flowers with inter-class similarity and intra-class variability. The study suggested the potential application of this model in recognizing wildflowers in National Parks, and further improvements can address challenges related to intra-class differences and inter-class similarities in flower photos, possibly involving more advanced algorithms and deep learning expertise. In essence, this research highlighted the successful application of CNNs for flower recognition, showcasing VGG16 as a promising model, especially for distinct flower species, with implications for preserving and cataloging floral diversity in natural ecosystems like National Parks (Mastura Hanafiah et al., 2022).

This study presents a deep learning approach for classifying various flowers using the Visual Geometry Group’s 102-category flower dataset, consisting of 8,189 images across 102 categories. The method involves two main stages: image segmentation and classification. Two popular Convolutional Neural Network (CNN) architectures, GoogleNet and AlexNet, were compared for classification performance. With identical hyperparameters for both models, GoogleNet demonstrated superior results, achieving Top-1 and Top-5 accuracies of 47.15% and 69.17%, respectively, compared to AlexNet’s 43.39% and 68.68%. These results significantly outperformed random classification accuracy, demonstrating the potential of this method for real-time flower classification applications.

Furthermore, the study highlights that deeper networks can yield better performance and regularization, particularly on large datasets. It also underscores the effectiveness of GoogleNet’s Inception module, which reduces parameters without sacrificing model accuracy (Ayesha Gurnani et al., 2017).

In this study (Hiary H et al., 2018), a novel two-step deep learning approach was developed for accurate flower classification, overcoming the challenges posed by the wide variety of flower species with similar shapes and appearances. The first step involved automated flower region segmentation using a fully convolutional network (FCN). This step ensued in creating a robust convolutional neural network (CNN) classifier for flower type differentiation. Notably, the proposed method achieved classification results exceeding 97% accuracy on three well-known flower datasets, outperforming state-of-the-art approaches in this domain. Some essential contributions to obtaining success in this method included CNNs for feature learning, localized flower region detection, and the transfer of weights from pre-trained models. Also, more introductions included gradual CNN learning, optimized weight convergence, and a data augmentation step to enhance robustness and accuracy. This work demonstrates the best flower classification accuracy to date and suggests the potential application of this approach in other image-related tasks facing similar challenges.

In this study (Yeqi Fei et al., 2023), a highly efficient and accurate classification model for fresh-cut flowers was developed, addressing critical factors such as classification speed and accuracy, which are crucial for quality control and pricing in the fresh-cut flower industry. The researchers collected RGB images and depth information data for rose flowers and designed a robust data augmentation strategy to overcome limited sample size constraints. They established a novel architecture based on the ShuffleNet V2 network backbone, performed transfer learning, and incorporated an attention mechanism to classify flowers of five specifications. The results were impressive, with classification accuracies exceeding 97% on all datasets and a rapid overall prediction speed of 0.020 seconds per flower. Compared to existing flower classification methods, this approach demonstrated significant advantages regarding parameter efficiency, classification speed, and accuracy. It holds great promise for developing fresh-cut flower classification and grading systems, offering the potential to enhance efficiency and maintain flower quality throughout the classification process.

Previous Research (Musa Cıbuk et al., 2019) addresses flower species classification using deep convolutional neural networks (DCNNs) for digital flower catalogs. It employs pre-trained AlexNet and VGG16 models for feature extraction, combining their features for efficiency. A feature selection method, mRMR, refines these features. The extracted features are then classified using an SVM with an RBF kernel. Flower17 and Flower102 datasets in place achieved accuracies of 96.39% and 95.70%, respectively. Despite its simplicity, the method proves highly effective, showcasing the potential of this DCNN-based hybrid approach for accurate flower species classification in image-based tasks.

Dataset overview

In our study, the dataset with five distinct flower classes emanated from the Kaggle repository (Kannan K, 2023): Lilly, lotus, sunflower, orchid, and tulip, each class comprising 1000 images. These images were curated to facilitate a multi-class classification approach for accurately classifying flowers within these five categories. We randomly allocated 20% of the images and designated 10% for testing and validation. The remaining ones were for training. Figure 1 depicts some samples of images from the dataset (Database: https://www.kaggle.com/datasets/kausthubkannan/5-flower-types-classification-dataset).

Fig. 1. Sample images from the dataset.

Basic Theory

Basics

VGG16

VGG16 belonged to the Visual Geometry Group (VGG) deep learning architecture series. The architecture of VGG16 appears in Figure 2. Its source was the University of Oxford and comprises 16 layers, blending 3 x 3 convolutional filters and strategically positioned pooling layers. VGG16 begins with an input layer tailored for 224 x 224 RGB images. It embraces 13 convolutional layers, each using a 3 x 3 kernel, followed by ReLU activation functions. Five max-pooling layers systematically reduce spatial dimensions while preserving critical information. This architecture transforms from image analysis to feature extraction. Its hierarchical feature learning is the key to its adaptability. VGG16 excels in tasks beyond image classification, including content-based image retrieval and object localization. For classification, it deploys three fully connected layers. The initial two layers have 4096 channels with ReLU activations. The final layer produces a 1000-dimensional output vector, aligning with ImageNet's 1000 classes, culminating in a softmax activation for class probabilities. Dropout regularization became necessary to combat overfitting after the first two fully connected layers. This randomness guards against overreliance on specific neurons, enhancing the network's robustness to real-world data. VGG16's elegance and hierarchical feature learning make it a potent tool for diverse applications. In its simplicity, we witness the power of deep neural networks, unraveling the complexities of the visual world (Karen Simonyan, 2014; Mesut Toğaçar, 2020).

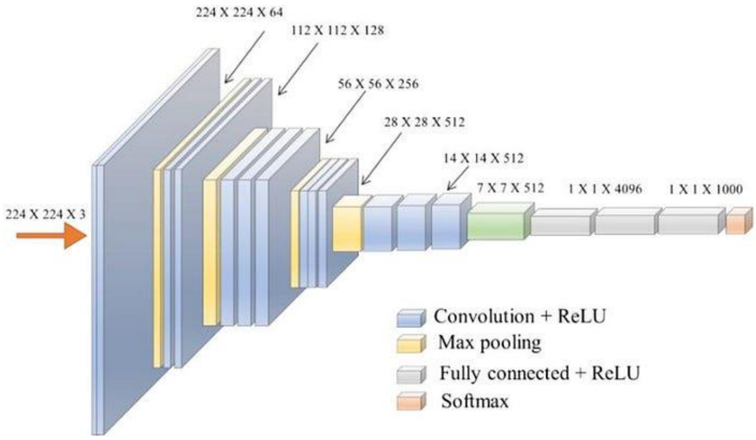


Fig. 2. VGG16 architecture (Vigneashwara Pandiyan et al., 2019).

Xception

Xception (Extreme Inception) reimagines convolutional networks by taking cues from the initial Inception principles and refining them to create an even more powerful architecture (Fig. 3). Typically designed to handle inputs of size 299 x 299 for RGB images, Xception commences with a set of regular convolutional layers to initially pre-process the image. However, the heart and soul of this model lie in its recurrent 'middle flow,' repeated eight times in the architecture. This flow is a distinctive construct that pivots around the concept of depth-wise separable convolutions. Here, spatial and channel-wise features are learned separately. Thus, we effectively capture details with fewer parameters. A batch normalization technique has associations with each depth-wise operation, ensuring stable activations, which subsequently pass through the ReLU activation function, introducing non-linearity and enhancing the model's expressiveness. This innovative decoupling of spatial and channel-wise learning results in greater computational efficiency and empowers the Xception model to set new performance benchmarks across several image-centric tasks (Francois Chollet, 2017).

Methodology

Transfer learning

Transfer learning in machine learning involves repurposing knowledge from solving one task to improve learning efficiency and performance on a different, related task. Utilizing pre-trained models-neural networks trained on extensive datasets like ImageNet-transfer learning leverages their learned features for new tasks. While freezing early layers that capture generic features and fine-tuning later layers to adapt to new data, transfer learning reduces training time and enhances performance. This technique benefits from domain similarity between the original and new tasks, focusing on reusing representations to generalize across different datasets. It excels in scenarios with limited labeled data for the new task, offering improved results and robustness. Transfer learning's essence lies in leveraging existing knowledge, enabling quicker adaptation to new tasks, and contributing to enhanced model performance by utilizing learned features from related domains (Fuzhen Zhuang, 2020; Arun Singh, 2022). This study involves the design of Transfer Learning models based on VGG16 and Xception, both with and without dropout. The methodology (Algorithm 1) illustrates the implementation process.

Algorithm-1: VGG16 / Xception-based Transfer Learning.
Input: Imagedatasets (Training, Validation, Testing), Num_classes: Number of classes.
Output: Classification results, model performance metrics.
Step 1: Import necessary libraries, including TensorFlow and Keras, define the num_classes, and load the pre-trained VGG16 or Xception model.
Step 2: Freeze Pre-Trained Model Layers.
Step 3: Build Model Architecture:
Create a new Sequential model
Add the pre-trained VGG16 base model
Add a Flatten layer to flatten the extracted features
Add a Dense layer with ReLU activation (512 units) to capture complex patterns
Add a Dropout layer to prevent overfitting (dropout rate of 0.5)
Add a final Dense output layer with softmax activation for class probabilities. and Dropout layer to prevent overfitting
Step 4: Compile the Model with data augmentation techniques and Adam optimizer with different learning rate.
Step 5: Generate Training and Validation Data the model.
Step 6: Evaluate the pertained model on Testing Data. Obtain test loss and test accuracy as evaluation metrics.
Step 7: Calculate the confusion matrix using true and predicted classes.
Step 8: Stop.

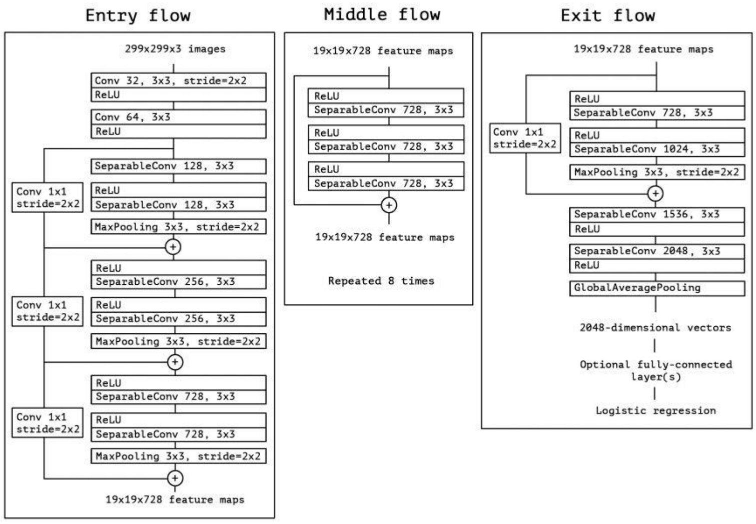


Fig. 3. Xception architecture (Francois Chollet, 2017).

Proposed hybrid model

In this study, we leverage the prowess of pre-trained models, notably VGG16 and Xception, to comprehend and characterize the hierarchical feature spaces of flower images (Fig. 4). Initially, we harness TensorFlow, a versatile deep learning framework, to streamline the modeling process. VGG16 has deep architecture, primarily characterized by 3 x 3 convolutional filters. It commences with an input RGB images, followed by a sequence of convolutional layers. Each convolutional layer employs a 3 x 3 filter with a stride of 1 to preserve spatial resolution and ensure that every pixel in the input contributes to the output. It leads to a Rectified Linear Unit (ReLU) activation function. As these convolution operations unfold, hierarchical features emerge, with initial layers typically capturing basic patterns and textures such as edges and blobs, and deeper layers can extract more intricate features, representing object parts or even entire objects.

The architecture of VGG16 arranges its features by layer depth, organized into blocks where each block includes a sequence of convolutional layers followed by a subsequent max-pooling layer. The spatial dimensions can decrease due to pooling operations as the network delves deeper. The depth increases, signifying the representation of more complex features.

The Xception model rethinks the convolution operation by using depthwise separable convolutions. Instead of performing standard convolutions, it breaks the operation into depthwise and pointwise operations. Initially, it uses standard convolutional layers, but the core of the model consists of its 'middle flow,' repeated eight times. In this middle flow, spatial features and channel-wise features are learned separately, enabling efficient and detailed feature extraction. The Xception model organizes its features using blocks. The middle flows, comprising blocks of depthwise separable convolutions, are the heart of this architecture, grouping features based on the depth of their extraction and the type of convolution used.

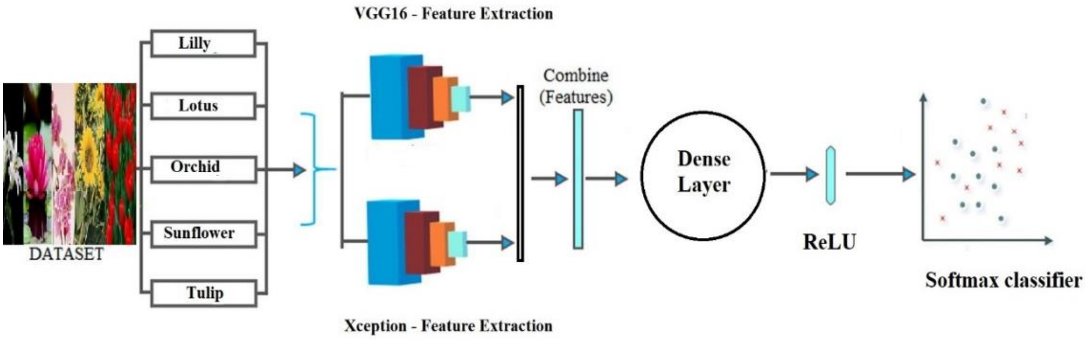


Fig. 4. Proposed hybrid model.

These individual features are maps from each model. They converge into a unified feature space. This amalgamation ensures that the most salient and distinguishing characteristics per model interpretation come together. This consolidated feature space, rich with information and shades from both models, is subsequently channeled through a sequence of dense layers in the hybrid model. These dense layers are responsible for introducing deeper non-linear transformations to the combined features, thus refining and preparing them for an effective classification. The combined feature set passes through a dense layer with a ReLU activation. Within this layer, the features undergo transformations that empower the model and better discern intricate patterns and relationships among classes.

The final step in this intricate process is the classification layer for interpreting the transformed features and assigning them to predefined classes. However, the raw outputs from this layer may not be immediately interpretable, as they may not sum up to one or may have varying magnitudes. These outputs become interpretable and provide a clear ranking of class probabilities, thus passing through a softmax activation function. This function ensures that the output values are normalized and sum up to one, making them directly representable as probabilities. This approach leverages diverse feature extraction capabilities per architecture, potentially leading to more comprehensive representations and enhanced classification accuracy.

The hybrid model emanates from the Adam optimizer and categorical cross-entropy loss, with periodic validation to monitor its performance. Post-training, the model's efficacy is tested on a separate dataset. Algorithm 2 illustrates the implementation process and the proposed hybrid flower image classification.

Algorithm 2: Proposed hybrid flower image classification

Input: Image datasets (Training, Validation, Testing), Num_classes: Number of classes.

Output: Classification results, model performance metrics.

Step 1: Load VGG16 and Xception with ImageNet weights.

Step 2: Freeze layers of each model to retain pre-trained weights.

Step 3: Define a common input layer for image dimensions (128, 128, 3).

Step 4: Extract features.

Step 4.1: Pass image through VGG16; store the output as 'vgg16_features'

Step 4.2: Pass image through Xception; store the output as 'xception_features'

Step 5: Concatenate- 'vgg16_features' and 'xception_features'.

Step 6: Add subsequent layers for classification:

Dense layer with ReLU activation

Dropout layer

Output layer with softmax activation for classification

Step 7: Compile the hybrid model using:

Optimizer: Adam

Loss: Categorical Cross-Entropy

Metrics: Accuracy

Step 8: Initialize-'ImageDataGenerator` such as

rescale = 1.0/255, rotation_range = 40,
zoom_range = 0.2, horizontal_flip = True, fill_mode = 'nearest.
Step 9: Define data generators for:
Training data
Validation data
Testing data
Step 10: Train the hybrid model using the training data, and validate using the validation data.
Step 11: Evaluate the model's performance on the testing data.
Step 12: Generate and display:
Confusion matrix
Classification report
Training - validation loss over epochs
Training - validation accuracy over epochs
Step 13: End.

Results and Discussion

The endeavor to achieve robust and accurate flower classification through deep learning models has witnessed substantial advancements in recent years. This study delves into a comparative analysis of various architectures, including Neural Network, Random Forest Classifier with Histogram-Based Image Features, and Convolutional Neural Networks (CNNs) such as VGG16, Xception, and hybrid models. It explores different configurations, epochs, dropout rates, early stopping techniques, and diverse architectures for flower classification. This exploration aims to establish a benchmark for flower classification models, unraveling insights into the most effective methods for accurate and robust flower species recognition.

This evaluation phase generates a detailed classification report and confusion matrix for insights into class-wise performance. Visualization techniques, employing Matplotlib and Seaborn, provide graphical representations of training-validation loss and accuracy trends across epochs, and a heat-mapped confusion matrix showcases the model's classification ability.

Specifically, the Neural Network architecture employed for comparison includes an input layer representing flattened data, followed by a hidden dense layer containing 512 neurons with ReLU activation. The output layer consists of 5 neurons with softmax activation, making it suitable for multi-class classification tasks. Before training, image data undergoes preprocessing via rescaling. The model is then trained using the Adam optimizer with a learning rate of 0.001 and categorical cross-entropy loss.

In contrast, the Random Forest Classifier with Histogram-Based Image Features consists of 100 decision trees in this implementation, trained to recognize distinct flower classes by discerning patterns within the histogram features. Through analysis of pixel intensity distributions, the model effectively captures key visual traits of flowers, including color and texture, which are crucial for accurate classification.

Table 1 outlines classification metrics for various models with different configurations of dense layers, dropout rates, and early stopping techniques. It includes data for a Neural Network trained over 25 epochs, resulting in a training accuracy of 0.74, with validation and testing accuracies of 0.650 and 0.603, respectively. Additionally, a Random Forest Classifier with Histogram-Based Image Features with 100 trees achieved a training accuracy of 0.999, with validation and testing accuracies of 0.592 and 0.600, respectively. These models are compared to other CNN models such as VGG16, Xception model, and various proposed hybrid models, each featuring different configurations of dense layers, dropout rates, and early stopping techniques. The VGG16 model without dropout displays impressive training accuracy at 96.67%, outperforming the 88.80% testing accuracy, yet with a slight drop in generalization from validation (90.10%) to testing, indicating potential minor overfitting tendencies but still showcasing robust recognition of various flower species. Conversely, the VGG16 model with 0.5 dropout presents lower training accuracy at 83.94%, with validation accuracy close at 89.20%, suggesting that dropout could limit learning complex patterns, affecting performance on unseen data. Whereas the Xception without dropout maintains consistency across training (96.57%), validation (90.50%), and testing (89.39%) accuracies, demonstrating strong generalization capabilities. Conversely, the Xception model with 0.5 dropouts maintains robustness in generalization despite a training accuracy drop of 83.09%, holding steady validation (90.30%) and testing (89.80%) accuracies compared to its non-dropout variant. In the hybrid models, those with no dropout exhibit compelling performance, with higher training accuracies (94.97% to 97.23%) closely aligned with validation (86.50% to 90.90%) and testing (89.99% to 91.20%) accuracies, showcasing robust learning and generalization. However, the introduction of dropout (0.5) in these models leads to a marginal decrease in performance, with slightly lower accuracy (86.59% to 90.10%) compared to their non-dropout counterparts, potentially hindering higher accuracy.

Table 1. Comparison of model performance for flower image classification.

Sl. No.	Model	Epochs	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
1.	Neural Network	25	0.8454	0.74	1.08	0.650	1.12	0.603
2.	Random Forest Classifier with Histogram-Based Image Features	No. of trees = 100	-	0.999	-	0.592	-	0.6
3.	VGG16 with no dropout	25	0.0933	0.9667	0.3754	0.9010	0.4831	0.8880
4.	VGG16 with 0.5 dropout	25	0.4146	0.8394	0.3635	0.8920	0.4265	0.8619
5.	Xception with no dropout	25	0.1100	0.9657	0.3934	0.9050	0.4093	0.8939
6.	Xception with 0.5 dropout	25	0.4513	0.8309	0.3394	0.9030	0.339	0.898
7.	Proposed hybrid model with 512 dense layer, early stopping and no dropout	12	0.1582	0.9497	0.4418	0.8650	0.3311	0.8999
8.	Proposed hybrid model with 512 dense layer, 0.5 dropout and early stopping	21	0.4374	0.8420	0.3194	0.8920	0.3609	0.8659
9.	Proposed hybrid model with 512 dense layer, 0.25 dropout and early stopping	22	0.2118	0.9234	0.3457	0.9050	0.2999	0.8939
10.	Proposed hybrid model for 25 epochs with 512 dense layers and no dropout	25	0.1280	0.9549	0.3853	0.9070	0.4333	0.9120
11.	Proposed hybrid model for 25 epochs with 512 dense layers and 0.5 dropout	25	0.4252	0.8480	0.3108	0.9030	0.3062	0.8920
12.	Proposed hybrid model for 25 epochs with 256 dense layers and no dropout	25	0.0748	0.9723	0.2787	0.9090	0.3594	0.9020
13.	Proposed hybrid model for 25 epochs with 256 dense layers and 0.5 dropout	25	0.1660	0.9386	0.2568	0.9090	0.2931	0.9100

The training accuracy metric evaluates the model's proficiency in fitting the training data, an indicator of the model's learning capacity and potential overfitting tendencies. The Random Forest Classifier with Histogram-Based Image Features achieved the highest training accuracy of 99.9%. Subsequently, the proposed hybrid model trained for 25 epochs with 256 dense layers and no dropout exhibited the highest training accuracy at 97.23%, suggesting robust learning capabilities and effective representation learning. In contrast, the Neural Network and CNN models, excluding the Neural Network itself, such as VGG16 and Xception with a dropout rate of 0.5, exhibited comparatively lower training accuracy, indicating potential challenges in capturing complex patterns within the dataset.

Validation accuracy is a crucial metric that measures a model's ability to generalize well on unseen data. The proposed hybrid model, trained for 25 epochs with 512 dense layers and no dropout, attained the highest validation accuracy of 90.70%, indicating its strong generalization performance. Conversely, the Random Forest

Classifier with Histogram-Based Image Features achieved the lowest validation accuracy of 59.2%. Additionally, the hybrid model with 512 dense layers, utilizing early stopping with no dropout, exhibited a slightly lower validation accuracy of 90.10%, implying possible limitations in its ability to generalize compared to the top-performing model.

The testing accuracy metric provides insights into a model's real-world applicability and performance on completely unseen data. The proposed hybrid model, trained for 25 epochs with 512 dense layers and no dropout, exhibited the highest testing accuracy at 91.20%, signifying its robustness in making accurate predictions on new instances. Conversely, the Neural Network exhibited the lowest testing accuracy of 60%, and VGG16 had 0.5 dropouts and yielded the lowest testing accuracy of 86.19% among the CNN models, indicating potential challenges in generalization and performance on new, unseen flower images.

Tables 2-7 provide compression of accuracy and loss of proposed models. Figures 5-17 represent graphs of loss and accuracy for training and validation, along with confusion matrices of the different proposed models.

Table 2. Classification metrics for neural network and random forest classifier with histogram-based image features.

Neural Network	precision rate	recall rate	f1-sco re	support		Random Forest Classifier	precision rate	recall ra te	f1-score	support
Lilly	0.65	0.31	0.42	100		**Lilly**	0.54	0.64	0.58	100
Lotus	0.65	0.58	0.61	100		**Lotus**	0.65	0.62	0.64	100
Orchid	0.49	0.71	0.58	100		**Orchid**	0.58	0.52	0.55	100
Sunflower	0.68	0.88	0.77	100		**Sunflower**	0.65	0.67	0.66	100
Tupil	0.62	0.54	0.58	100		**Tupil**	0.59	0.55	0.57	100
accuracy			0.60	500		**accuracy**			0.60	500
macro avg	0.62	0.60	0.59	500		**macro avg**	0.60	0.60	0.60	500
weighted avg	0.62	0.60	0.59	500	**av g**	**weighted**	0.60	0.60	0.60	500

Table 3. Classification metrics for VGG16 with no dropout and VGG16 with 0.5 dropout.

VGG16 with no dropout	precision rate	recall rate	f1-scor e	support		VGG16 with 0.5 dropout	precision rate	recall rate	f1-scor e	support
Lilly	0.87	0.76	0.81	100		**Lilly**	0.85	0.72	0.78	100
Lotus	0.76	0.90	0.83	100		**Lotus**	0.75	0.83	0.79	100
Orchid	0.98	0.90	0.94	100		**Orchid**	0.97	0.91	0.94	100
Sunflower	0.88	0.99	0.93	100		**Sunflower**	0.91	0.97	0.94	100
Tupil	0.98	0.89	0.93	100		**Tupil**	0.85	0.88	0.86	100
accuracy			0.89	500		**accuracy**			0.86	500
macro avg	0.90	0.89	0.89	500		**macro avg**	0.86	0.86	0.86	500
weighted avg	0.90	0.89	0.89	500		**weighted avg**	0.86	0.86	0.86	500

Table 4. Classification metrics for Xception with no dropout and Xception with 0.5 dropout.

Xception with no dropout	precision rate	recall rate	f1-score	support		Xception with 0.5 dropout	precision rate	recall rate	f1-score	support
Lilly	0.85	0.80	0.82	100		Lilly	0.89	0.70	0.78	100
Lotus	0.82	0.93	0.87	100		Lotus	0.79	0.94	0.86	100
Orchid	0.91	0.91	0.91	100		Orchid	0.94	0.95	0.95	100
Sunflower	0.95	0.97	0.96	100		Sunflower	0.96	1.00	0.98	100
Tupil	0.95	0.86	0.90	100		Tupil	0.93	0.90	0.91	100
accuracy			0.89	500		accuracy			0.90	500
macro avg	0.90	0.89	0.89	500		macro avg	0.90	0.90	0.90	500
weighted avg	0.90	0.89	0.89	500		weighted avg	0.90	0.90	0.90	500

Table 5. Classification metrics for proposed hybrid model with 512 dense layers, early stopping and no dropout, as well as for the proposed hybrid model with 512 dense layer, 0.5 dropout and early stopping.

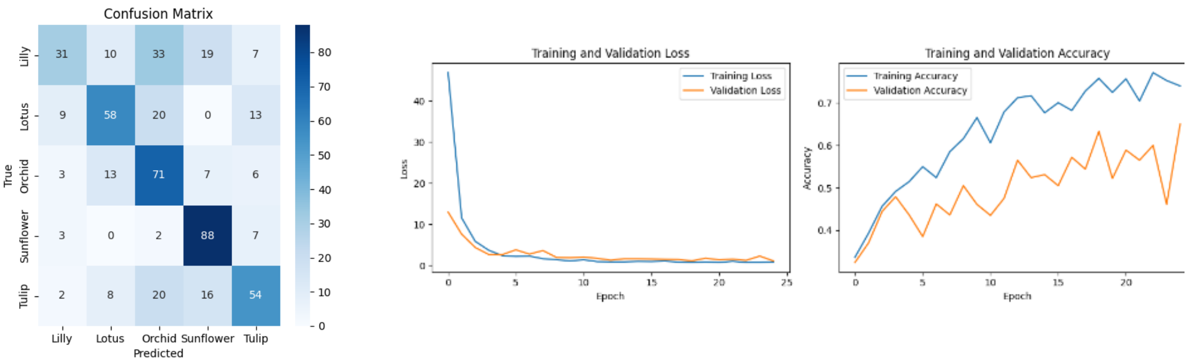
Proposed hybrid model with 512 dense layer, early stopping and no dropout	precision rate	recall rate	f1-score	support		Proposed hybrid model with 512 dense layer, 0.5 dropout and early stopping	precision rate	recall rate	f1-score	support
Lilly	0.85	0.73	0.78	100		Lilly	0.83	0.69	0.75	100
Lotus	0.83	0.92	0.87	100		Lotus	0.74	0.92	0.82	100
Orchid	0.92	0.95	0.94	100		Orchid	0.88	0.92	0.90	100
Sunflower	0.98	0.98	0.98	100		Sunflower	0.97	0.96	0.96	100
Tupil	0.92	0.92	0.92	100		Tupil	0.94	0.84	0.89	100
accuracy			0.90	500		accuracy			0.87	500
macro avg	0.90	0.90	0.90	500		macro avg	0.87	0.87	0.87	500
weighted avg	0.90	0.90	0.90	500		weighted avg	0.87	0.87	0.87	500

Table 6. Classification metrics for proposed hybrid model with 512 dense layer, 0.25 dropout and early stopping, as well as for the proposed hybrid model for 25 epochs with 512 dense layers and no dropout.

Proposed hybrid model with 512 dense layer, 0.25 dropout and early stopping	precision rate	recall rate	f1-score	support		Proposed hybrid model for 25 epochs with 512 dense layers and no dropout	precision rate	recall rate	f1-score	support
Lilly	0.83	0.76	0.79	100		Lilly	0.85	0.81	0.83	100
Lotus	0.85	0.88	0.86	100		Lotus	0.83	0.92	0.87	100
Orchid	0.93	0.96	0.95	100		Orchid	1.00	0.95	0.97	100
Sunflower	0.92	1.00	0.96	100		Sunflower	0.96	0.94	0.95	100
Tupil	0.95	0.87	0.91	100		Tupil	0.93	0.94	0.94	100
accuracy			0.89	500		accuracy			0.91	500
macro avg	0.89	0.89	0.89	500		macro avg	0.91	0.91	0.91	500
weighted avg	0.89	0.89	0.89	500		weighted avg	0.91	0.91	0.91	500

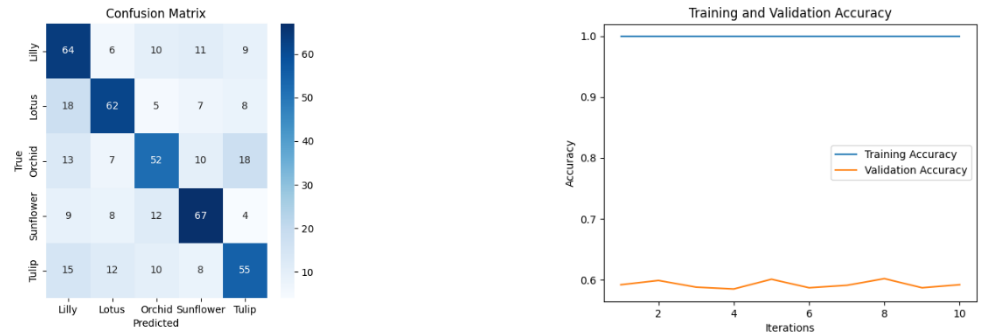
Table 7. Classification metrics for proposed hybrid model for 25 epochs with 256 dense layers and no dropout and early stopping, as well as for the proposed hybrid model for 25 epochs with 256 dense layers and 0.5 dropout.

Proposed hybrid model for 25 epochs with 512 dense layers and 0.5 dropout	precision rate	recall rate	f1-score	support		Proposed hybrid model for 25 epochs with 256 dense layers and no dropout	precision rate	recall rate	f1-score	support
Lilly	0.84	0.70	0.77	100		Lilly	0.91	0.69	0.78	100
Lotus	0.78	0.92	0.84	100		Lotus	0.83	0.90	0.87	100
Orchid	0.96	0.95	0.95	100		Orchid	0.90	0.96	0.93	100
Sunflower	0.96	1.00	0.98	100		Sunflower	0.95	0.99	0.97	100
Tupil	0.93	0.89	0.91	100		Tupil	0.92	0.97	0.95	100
accuracy			0.89	500		accuracy			0.90	500
macro avg	0.89	0.89	0.89	500		macro avg	0.90	0.90	0.90	500
weighted avg	0.89	0.89	0.89	500		weighted avg	0.90	0.90	0.90	500



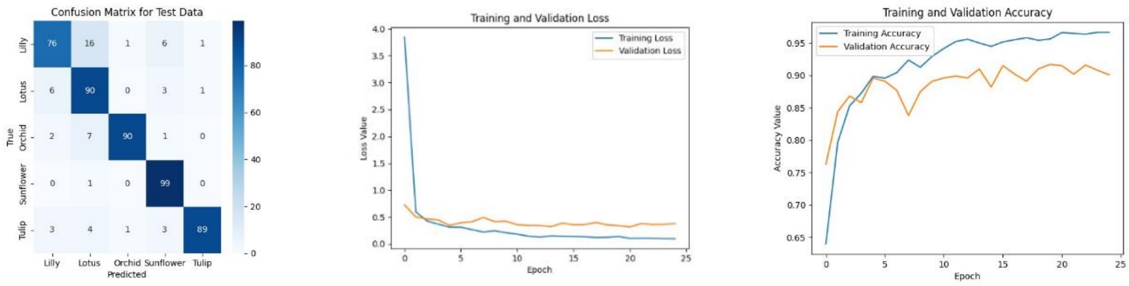
a b c

Fig. 5. Performance evaluation of classification using neural network (a) confusion matrix for classification using neural network based classification, (b) Training-validation loss for classification using neural network based classification and (c) Training-validation accuracy for classification using neural network based classification.



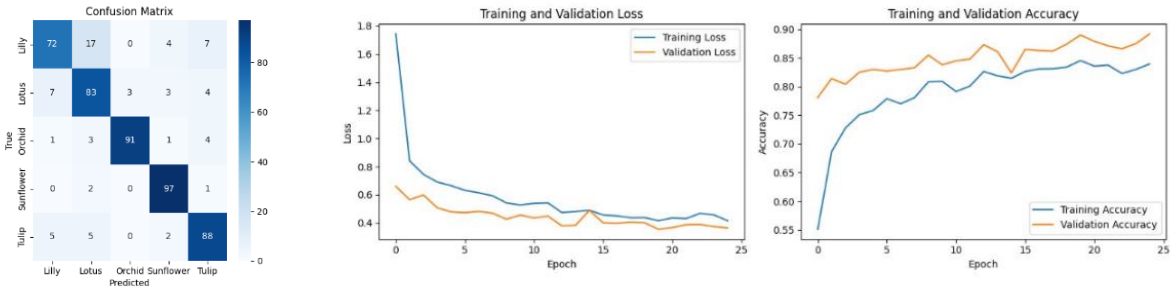
a b

Fig. 6. Performance evaluation of classification using random forest classifier with histogram-based image (a) confusion matrix for classification using random forest classifier with histogram-based image and (b) training-validation accuracy for classification using random forest classifier with histogram-based image.



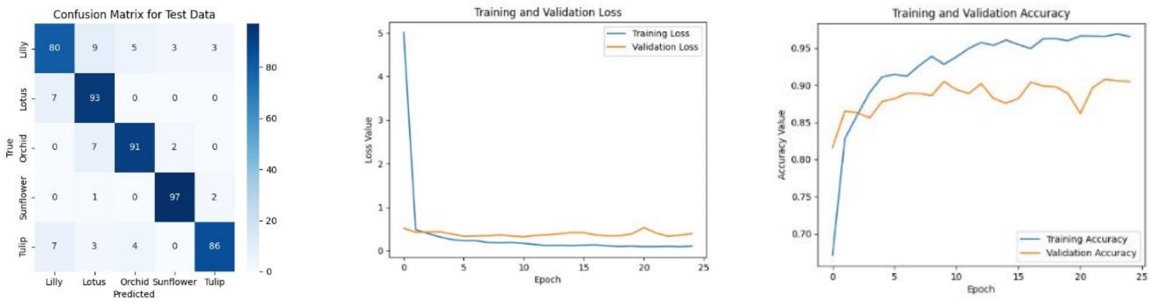
a b c

Fig. 7. Performance evaluation of classification using VGG16 with no dropout (a) confusion matrix for classification using VGG16 with no dropout, (b) training-validation loss for classification using VGG16 with no dropout and (c) training-validation accuracy for classification using VGG16 with no dropout.



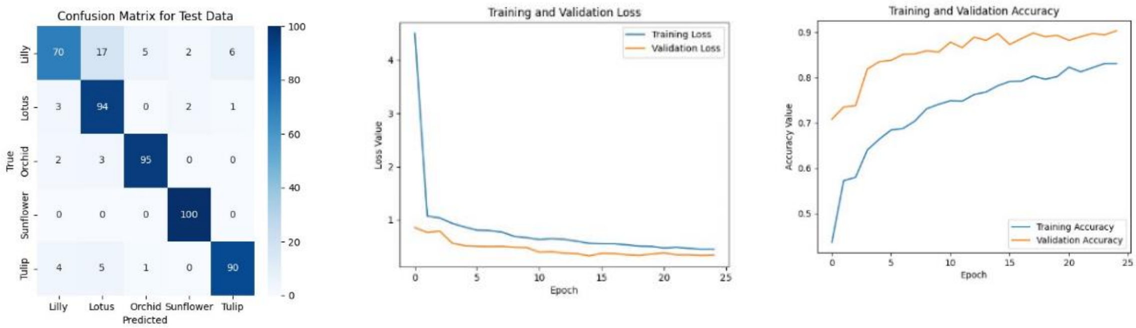
a b c

Fig. 8. Performance evaluation of classification using VGG16 with 0.5 dropout (a) confusion matrix for classification using VGG16 with 0.5 dropout, (b) training-validation loss for classification using VGG16 with 0.5 dropout and (c) training-validation accuracy for classification using VGG16 with 0.5 dropout.



a b c

Fig. 9. Performance evaluation of classification using Xception with no dropout (a) confusion matrix, for classification using Xception with no dropout (b) training-validation loss for classification using Xception with no dropout and (c) training-validation accuracy for classification using Xception with no dropout.

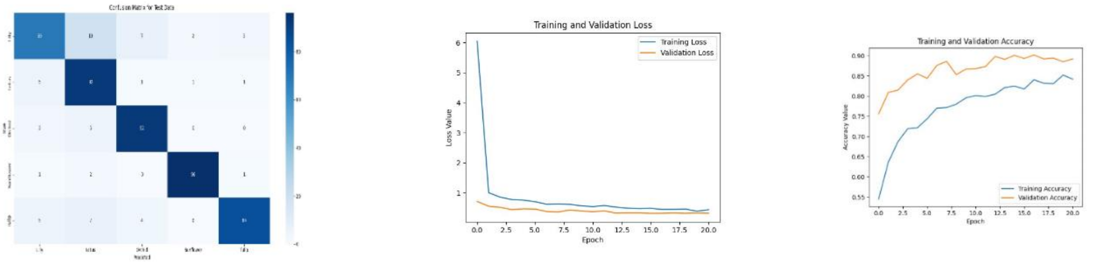


a b c

Fig. 10. Performance evaluation of classification using Xception with 0.5 dropout (a) confusion matrix for classification using Xception with 0.5 dropout, (b) training-validation loss for classification using Xception with 0.5 dropout and (c) training-validation accuracy for classification using Xception with 0.5 dropout.

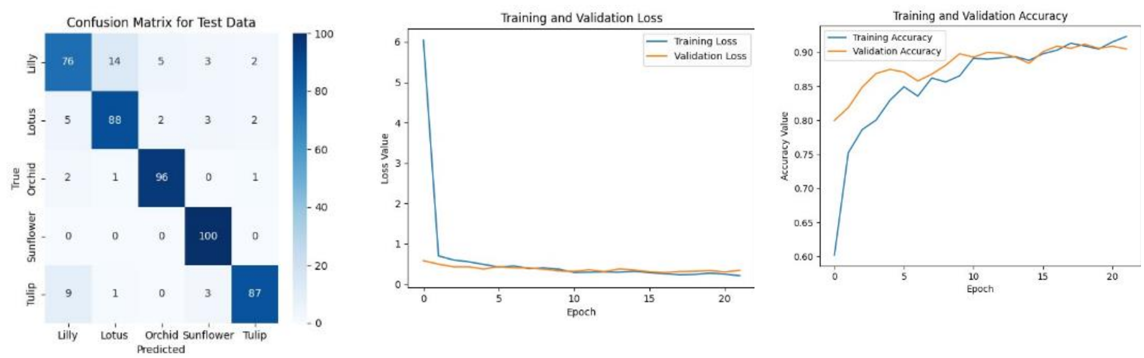
a b c

Fig. 11. Performance evaluation of classification using the proposed hybrid model that includes a 512 dense layers with early stopping and no dropout (a) confusion matrix for classification using the proposed hybrid model that includes a 512 dense layers with early stopping and no dropout, (b) training-validation loss for classification using the proposed hybrid model that includes a 512 dense layers with early stopping and no dropout and (c) training-validation accuracy for classification using the proposed hybrid model that includes a 512 dense layers with early stopping and no dropout.



a b c

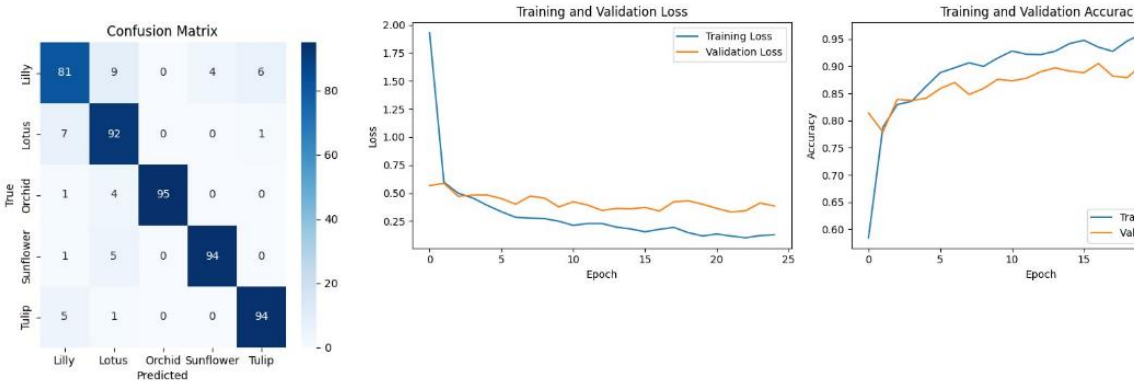
Fig. 12. Performance evaluation of classification using the proposed hybrid model that includes a 512 dense layers with 0.5 dropout and early stopping (a) confusion matrix for classification using the proposed hybrid model that includes a 512 dense layers with 0.5 dropout and early stopping, (b) training-validation loss for classification using the proposed hybrid model that includes a 512 dense layers with 0.5 dropout and early stopping and (c) training-validation accuracy for classification using the proposed hybrid model that includes a 512 dense layers with 0.5 dropout and early stopping.



a b c

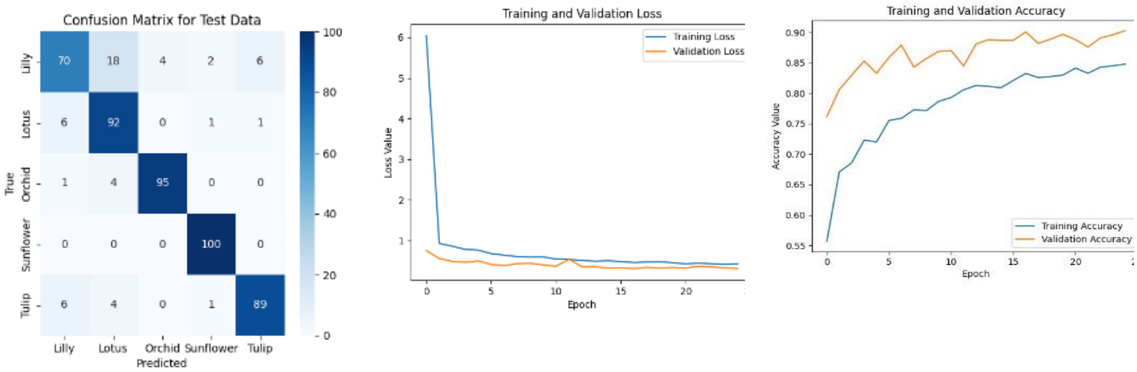
Fig. 13. Performance evaluation of classification using the proposed hybrid model that includes a 512 dense layers with 0.25 dropout and early stopping (a) confusion matrix for classification using the proposed hybrid model that includes a 512 dense layers with 0.25 dropout and early stopping, (b) training-validation loss for classification using the proposed hybrid model that includes a 512 dense layers with 0.25 dropout and early stopping and (c) training-

validation accuracy for classification using the proposed hybrid model that includes a 512 dense layers with 0.25 dropout and early stopping.



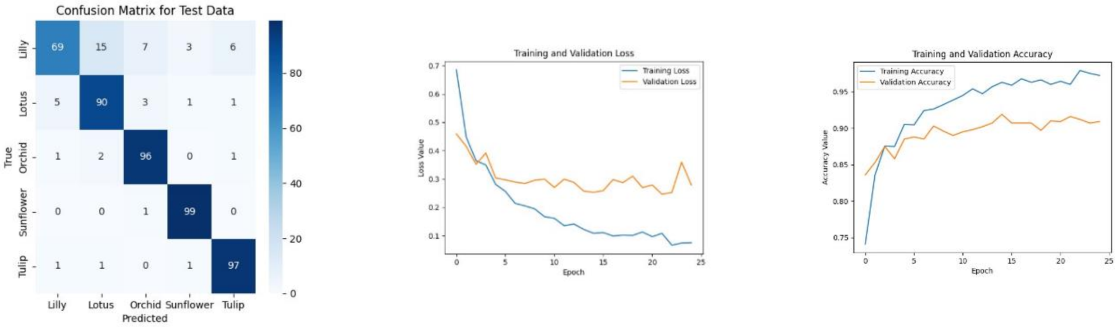
a b c

Fig. 14. Performance evaluation of classification using the proposed hybrid model for 25 epochs with 512 dense layers and no dropout (a) confusion matrix for classification using the proposed hybrid model for 25 epochs with 512 dense layers and no dropout, (b) training-validation loss for classification using the proposed hybrid model for 25 epochs with 512 dense layers and no dropout, and (c) training-validation accuracy for classification using the proposed hybrid model for 25 epochs with 512 dense layers and no dropout.



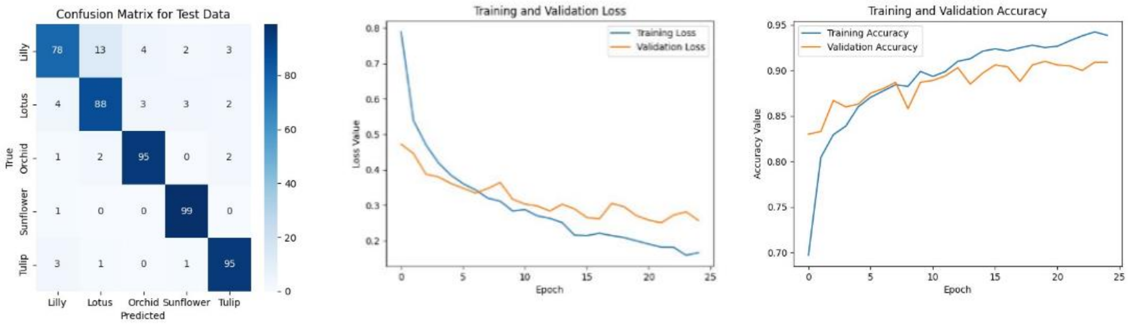
a b c

Fig. 15. Performance evaluation of classification using the proposed hybrid model for 25 epochs with 512 dense layers and no dropout (a) confusion matrix for classification using the proposed hybrid model for 25 epochs with 512 dense layers and no dropout, (b) training-validation loss for classification using the proposed hybrid model for 25 epochs with 512 dense layers and no dropout, and (c) training-validation accuracy for classification using the proposed hybrid model for 25 epochs with 512 dense layers and no dropout.



a b c

Fig. 16. Performance evaluation of classification using the proposed hybrid model for 25 epochs with 256 dense layers and no dropout (a) confusion matrix for classification using the proposed hybrid model for 25 epochs with 256 dense layers and no dropout, (b) training-validation loss for classification using the proposed hybrid model for 25 epochs with 256 dense layers and no dropout, and (c) training-validation accuracy for classification using the proposed hybrid model for 25 epochs with 256 dense layers and no dropout.



a b c

Fig. 17. Performance evaluation of classification using the proposed hybrid model for 25 epochs with 256 dense layers and 0.5 dropout (a) confusion matrix for classification using the proposed hybrid model for 25 epochs with 256 dense layers and 0.5 dropout, (b) training-validation loss for classification using the proposed hybrid model for 25 epochs with 256 dense layers and 0.5 dropout and (c) training-validation accuracy for classification using the proposed hybrid model for 25 epochs with 256 dense layers and 0.5 dropout.

Precision, recall, and F1-score are critical metrics, especially in multiclass classification tasks, which enable assessments of the model's ability to classify each case correctly and avoid misclassifications. Comparing VGG16 models, VGG16 with no dropout displayed marginally higher precision and recall rates across different flower categories compared to VGG16 with 0.5 dropout. This suggests that the absence of dropout layers might contribute to slightly better classification performance for this architecture. Analyzing Xception models, Xception with no dropout consistently outperformed Xception with 0.5 dropouts, showcasing higher precision, recall, and F1 scores across various flower classes. This emphasizes the impact of dropout layers on the model's ability to identify flower types accurately. Evaluating the proposed hybrid models, without dropout layers, exhibited slightly better or at least comparable precision, recall, and F1 scores than their counterparts with dropouts. This suggests that dropout layers might not significantly contribute to the overall performance of these hybrid architectures for flower classification.

Following a comprehensive assessment of training accuracy, validation accuracy, testing accuracy, as well as precision, recall, and F1 scores across multiple models, the proposed hybrid model for 25 epochs with 512 dense layers and no dropout emerges as the most effective model for flower classification. This model consistently demonstrates superior performance across various evaluation metrics, indicating its robustness, high accuracy, and reliable classification capabilities across different flower categories. Table 8 presents the classification metrics for the proposed hybrid model over 25 epochs with 512 dense layers and 0.5 dropouts.

Table 8. Classification metrics for the proposed hybrid model for 25 epochs with 512 dense layers and no dropout.

Proposed hybrid model for 25 epochs with 512 dense layers and no dropout	precision rate	recall rate	f1-score	support
Lilly	0.90	0.78	0.83	100
Lotus	0.85	0.88	0.86	100
Orchid	0.93	0.95	0.94	100
Sunflower	0.94	0.99	0.97	100
Tupil	0.93	0.95	0.94	100
accuracy			0.91	500
macro avg	0.91	0.91	0.91	500
weighted avg	0.91	0.91	0.91	500

Table 9 presents a comparative analysis of flower classification accuracy as documented in the existing literature. In practical scenarios, such assessments serve as valuable benchmarks for evaluating the performance of the proposed classification models.

Table 9. Comparison of classification accuracy with the existing work.

#	Study	Classes	Dataset Size	Method	Accuracy
1.	(Peryanto, 2022)	Aster Mawar (Rose) Tupil	400 images in each class	CNN SVM	91.6% 78.3%
2.	(Nuraini Rini, 2023)	Cherry Rose Velvet Queen Fiesta Del Sol Sunny Smile Teddy Bear Early Russian Red Sun	Total 350 images	Multiclass SVM	79 %
3.	(Qing Lv, 2022)	Oxford-17 dataset comprises 17 categories of flower datasets, including Fritillary, Dandelion, Lily Valley, Daisy, Daffodil, Cowslip, Tulip, Tigerlily, Crocus, Bluebell, etc.	Each category contains 80 images, resulting in a total of 1360 images.	LeNet AlexNet VGGNet- 16	58% 41% 72%
4.	Proposed	Sunflower Orchid Tulip Lilly Lotus	1000 images of each class	Hybrid VGG16 and Xception feature extractor with softmax classifier	91. 2%

Thus, the field of flower classification has seen diverse approaches and datasets explored by researchers. Peryanto et al. (2022) demonstrated the effectiveness of CNN and SVM techniques on a dataset comprising three flower classes, achieving high accuracies. Nuraini Rini (2023) employed Multiclass SVM on a smaller dataset, yielding satisfactory accuracy. Qing Lv (2022) experimented with different architectures on the Oxford-17 dataset, highlighting varying levels of accuracy across models. Finally, the proposed method utilized a hybrid approach, combining VGG16 and Xception feature extractors, resulting in promising classification performance on a dataset with five flower classes. These findings underscore the importance of selecting appropriate methods and datasets for flower classification tasks, with each approach offering its strengths and limitations.

Conclusion

In the rapidly evolving landscape of artificial intelligence and deep learning, the pursuit of more accurate and efficient classification models remains a paramount challenge, particularly resonant in the domain of flower species classification, where automated recognition systems hold immense promise for various applications, including agriculture, environmental monitoring, and botanical research. This quest culminates in a significant milestone in advancing the state-of-the-art in flower species classification by leveraging the power of Convolutional Neural Networks (CNNs) and innovative hybrid models. The journey begins with meticulously examining various models, including Neural Networks, Random Forest Classifier with Histogram-Based Image Features, and various CNN models, each endowed with unique capabilities and nuances. These models undergo training and evaluation across multiple epochs to comprehensively gauge their performance, facilitated by a rich dataset comprising training loss, training accuracy, validation loss, validation accuracy, test loss, and test accuracy metrics, enabling a nuanced understanding of each model behavior across different stages of training and evaluation. The proposed hybrid model for 25 epochs with 512 dense layers and no dropout unequivocally demonstrates unparalleled prowess, exhibiting a remarkable state-of-the-art classification accuracy of 91.20% on the Kaggle flower dataset. Moreover, the success of the proposed hybrid model highlights the potency of feature fusion from multiple CNNs, offering insights that transcend the realm of flower classification. Summarizing the findings, the Neural Network achieves a moderate classification accuracy across different flower species, while the Random Forest Classifier with Histogram-Based Image Features shows comparable performance but with a slightly lower accuracy. VGG16 with no dropout and Xception with no dropout exhibit high accuracy rates, outperforming models with dropouts. The proposed hybrid model with 512 dense layers and no dropout consistently demonstrates superior accuracy, with notable improvements compared to models with dropouts. The hybrid model with 512 dense layers and 0.5 dropout also performs well but slightly lower than its counterpart without dropout. Models with early stopping generally exhibit better performance in terms of accuracy and convergence. Across different configurations, the hybrid model consistently achieves high accuracy rates, showcasing the effectiveness of feature fusion from multiple CNNs.

Future research could delve into experimenting with diverse combinations of pre-trained CNN architectures, probing the impact of merging features from various models on classification accuracy. This exploration could encompass architectures such as ResNet, Inception, or EfficientNet, aiming to discern the optimal fusion that elevates performance in classification tasks.

Acknowledgments

We extend our heartfelt gratitude to the management and staff of GSSSIETW, Navkis College of Engineering, and Rajeev Institute of Technology for their invaluable support, which was pivotal in completing this work. This research project received funding from the 'Consolidated University Research for Innovation and Excellence CORE Grant for Women PG College' (DST-CURIE), under the aegis of the Department of Science and Technology, Government of India, New Delhi, with reference number 'DST/CURIE-PG/2022/71'.

Conflict of Interest

The authors indicate no conflict of interest in this work.

References

Ayesha G, Viraj M. 2017. Flower categorization using deep convolutional neural networks. arXiv preprint arXiv:1708.03763, 4321-4324 (2017). https://doi.org/10.48550/arXiv.1708.03763

Ari P, Anton Y, Rusydi U. 2022. Convolutional neural network and support vector machine in classification of flower images. Khazanah Informatika : Jurnal Ilmu Komputer dan Informatika 8(1), 1-7. https://doi.org/10.23917/khif.v8i1.15531

Arun S, Akansha P, Manik R, Dalwinder S, Gurasis S, Omdev D. 2022. An iris recognition system using CNN & VGG16 technique. In 2022 10th international conference on reliability, infocom technologies and optimization 1-6. https://doi.org/10.1109/ICRITO56286.2022.9965172

Burhan D, Ahmet Ali S. 2022. A study on deep learning based classification of flower images. International Journal of Advanced Networking and Applications 14(02), 5385–5389. https://doi:10.35444/ijana.2022.14209

Francois C. 2017. Xception: Deep learning with depthwise separable convolutions. In proceedings of the IEEE conference on computer vision and pattern recognition

1251-1258. https://doi: 10.1109/CVPR.2017.195

Fuzhen Z, Zhiyuan Q, Keyu D, Dongbo X, Yongchun Z, Hengshu Z. 2020. A comprehensive survey on transfer learning. Proceedings of the IEEE. 109 (1), 43-76. https://doi.org/10.1109/JPROC.2020.3004555

Hiary H, Saadeh H, Saadeh M, Yaqub M. 2018. Flower classification using deep convolutional neural networks. IET Comput. Vis. 12(6), 855-862. https://doi.org/10.1049/iet-cvi.2017.0155

Karen S, Andrew Z. 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv 1409 1556, 1-14. https://doi.org/10.48550/arXiv.1409.1556

Kannan K. 2023. Five flower types classification dataset, Kaggle. https://www.kaggle.com/datasets/kausthubkannan/5-flower-types-classification-dataset

Musa C, Umit B, Yanhui G, Cevdet I, Abdulkadir S. 2019. Efficient deep features selections and classification for flower species recognition. Measurement 137, 7-13. https://doi.org/10.1016/j.measurement.2019.01.041

Mastura H, Mohd Azraei A, Shuzlina A, Sofianita M, Ariff Md Ab M, Mohd Razif S. 2022. Flower recognition using deep convolutional neural networks. IOP Conference Series: Earth and Environmental Science 1013 (1), 1-12. https://doi.org/10.1088/1755-1315/1019/1/012021.

Mesut T, Burhan E, Zafer C. 2020. Classification of flower species by using features extracted from the intersection of feature selection methods in convolutional neural network models. Measurement 158, 107703. https://doi.org/10.1016/j.measurement.2020.107703

Qing L, Suzhen Z, Yuechun W. 2022. Deep learning model of image classification using machine learning. Advanced Pattern Recognition Systems for Multimedia Data 1-12. https://doi.org/10.1155/2022/3351256

Rini N, Rachmat D, Desi N, Yeni D, Allan Desi A. 2023. Sunflower image classification using multiclass support vector machine based on histogram characteristics. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi) 7(1), 146-152. https://doi.org/10.29207/resti.v7i1.4673

Touqeer A, Abdul R, Muhammad Azam Z, Imran M, Muhammad Asim S, Wasif A, Muhammad Ahmad K, Gulzar A, Casper Shikali S. 2022. Deep neural networks for automatic flower species localization and recognition. Computational Intelligence and Neuroscience 1-19. https://doi.org/10.1155/2022/9359353

Vigneashwara P, Pushparaja M, Tegoeh T, Wahyu C, Omey Mohan M, David Jin HT. 2019. In-process virtual verification of weld seam removal in robotic abrasive belt grinding process using deep learning. Robotics and Computer-Integrated Manufacturing 57, 477-487. https://doi.org/10.1016/j.rcim.2019.01.006

Xiaoxue L, Rongxin L, Yanzhen Y, Kangkang X, Zeyuan L, Zhongzhi L. 2021. Flower image classification based on generative adversarial network and transfer learning. In IOP conference series: Earth and environmental science 647 (1), 012180. https://doi.org/10.1088/1755-1315/647/1/012180

Yeqi F, Zhenye L, Tingting Z, Chao N. 2023. A lightweight attention-based convolutional neural networks for fresh-cut flower classification. IEEE Access 11, 17283-17293. https://doi.org/10.1109/ACCESS.2023.3244386

Zhao J, Chu S. 2021. Research on flower image classification algorithm based on convolutional neural network. Journal of Physics: Conference Series 1994 (1), 012034. https://doi.org/10.1088/1742-6596/1994/1/012034