ⵜⴰⴼⴰⴽⵓⵍⵜ ǀ ⵜⵉⵎⵓⵙⵙⴰⵏⵉⵏ
FACULTÉ DES SCIENCES

MARRAKECH
جامعة القاضي عياض
UNIVERSITÉ CADI AYYAD
Semlalia

كليـــــــة العلوم
FACULTY OF SCIENCE

Master in Information Systems Engineering

Module: Deep Learning

Project: Classification of 5 Flower Types using InceptionV3

Dataset used: 5 Flower Types Classification Dataset (Kaggle)

Prepared by:

MOUEDDENE Amina

MOUEDDENE Wafae

AIT AISSA Ouissale

BAMOUS Kaoutar

Supervised by:

Mr. Mustapha AATILA

Academic Year: 2025–2026

Table of Contents

List of Figures

List of Tables

Chapter 1

Repository Link

The project's Git repository is available at the following link: Transfer Learning Project – GitHub Repository

1.1 Abstract

In recent years, the identification and categorization of flower species have grown becoming more significant in areas like botany, farming, and environmental protection. This This paper enhances a model for classifying flower images by employing deep convolutional neural networks.

Networks (ResNet, InceptionV3 , VGG) by integrating more profound residual architectures, thereby improving the model's bility to effectively capture complex feature relationships. Moreover, fresh information augmentation methods, such as random cropping, rotating, and color modification, are utilized to improve the model's robustness against varying input data. Throughout the experiment, a public dataset featuring five types of flowers (around 5000 images) was utilized, and the dataset was divided into training and testing subsets. The models underwent pre-training, and the network architecture was optimized. Throughout the training process, we documented the loss and precision for every epoch to monitor the model's development. The results of the experiment show that the suggested model can effectively and precisely identify flower species within the dataset. [1]

Keywords: Computer vision, Image recognition, Deep learning, Image classification , ResNet50, InceptionV3 , VGG .

1.2 Résumer

Ces dernières années, l'identification et la catégorisation des espèces de fleurs ont pris une importance croissante dans des domaines tels que la botanique, l'agriculture et la protection de l'environnement. Ce travail améliore des modèles de classification d'images de fleurs en utilisant des réseaux de neurones convolutionnels profonds (ResNet, InceptionV3 , VGG) , en intégrant des architectures résiduelles plus avancées afin d'augmenter la capacité du modèle à capturer efficacement des relations de caractéristiques complexes.

De plus, de nouvelles méthodes d'augmentation de données, telles que le recadrage aléatoire, la rotation et la modification des couleurs, sont utilisées afin de renforcer la robustesse du modèle face à la variation des données en entrée.

Pour les expérimentations, un jeu de données public contenant cinq types de fleurs (environ 5000 images) a été utilisé, puis divisé en ensembles d'apprentissage et de test. Le modèle ResNet a été pré-entraîné et l'architecture du réseau a été optimisée. Au cours de l'entraînement, nous suivons la fonction de perte et la précision à chaque époque afin de surveiller l'évolution des performances du modèle.

Mots-clés : Vision par ordinateur, Reconnaissance d'images, Apprentissage profond, Classification d'images, ResNet , InceptionV3 , VGG.

Chapter 2

Introduction

2.1 Transfer Learning

Transfer learning enhances the process of acquiring knowledge in a new context. task by applying knowledge gained from a similar task that has previously been mastered. Although the majority of machine learning algorithms are created to tackle individual tasks, the creation of algorithms that enable Transfer learning remains a subject of continuous interest in the field of machine learning. Society. This chapter presents an overview of the objectives, frameworks, and difficulties associated with transfer learning. It reviews contemporary studies. in this domain, providing a summary of the current advancements and detailing the unresolved issues. The survey addresses transfer in both inductive learning and reinforcement learning and addresses the challenges of adverse transfer and detailed task mapping [2].

2.2 Problem Statement

Because of time limitations or computational constraints, it isn't always feasible to create a model from the ground up for the plant classification issue in reality; we will evaluate the efficacy of techniques and attempt to discover a pre-trained approach for this type of issue. The issue we aim to address is identifying the flower type (daisy, dandelion, rose, sunflower, tulip) from a given image, totaling 5000.

images. To address this classification issue, we intend to utilize various methods, including machine learning.

(Support vector machine [3], Random Forest [4], K-nearest Neighbors [5]) and deep learning techniques (Custom CNN, VGG [6], ResNet [7], DenseNet [8]). In this classification task, we discovered that the deep learning approach performs significantly better [9].

2.3 Motivations

Flower classification entails sorting plants into groups and categories to enhance understanding, enable proper study, and ensure effective organization. It is a crucial stage in plant research and in the production and management of agroforestry. Plant taxonomy is largely based on visible characteristics like leaves, flowers, branches, bark, and fruits, with the categorization of flowering plants being a major aspect of this discipline.

Automated flower classification with high precision through computer technology provides significant scientific and societal advantages. It conserves time, minimizes human mistakes, and offers quicker access to trustworthy data for botanical studies, farming, and ecological preservation.

Additionally, due to recent progress in deep learning and computing technologies, deep convolutional neural networks can now be utilized on extensive image datasets, greatly enhancing classification precision relative to conventional machine learning techniques. This research intends to utilize these developments to create an effective model for recognizing and classifying flowers . [10]

2.4 Objectifs

The aim of transfer learning is to enhance learning in the target task by utilizing insights from the original task. Three typical metrics include that transfer could enhance learning. The first aspect is the attainable initial performance.

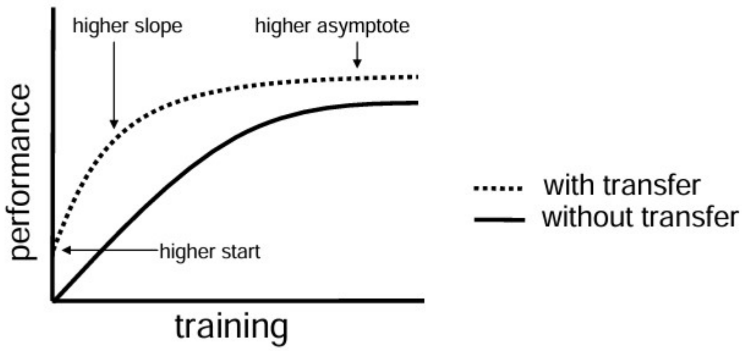

Figure 1: Three ways in which transfer might improve learning.

in the target task relying solely on the transferred knowledge, prior to any additional learning occurring, in contrast to the initial performance of an uninformed agent. The second is the duration required to completely master the target task considering the transferred knowledge relative to the time needed to acquire it from the beginning. Thirdly is the ultimate performance standard attainable in the objective task relative to the conclusive level without shifting. Figure 1 demonstrates these three metrics.

If a transfer method truly lowers performance, then it leads to negative transfer.

has taken place. A primary obstacle in creating transfer techniques is to facilitate positive transfer between suitably connected tasks while preventing

negative influence from tasks that are minimally related. A part of this chapter examines strategies for preventing adverse transfer.

When an agent utilizes knowledge from one task in a different one, it is frequently required to align the features of one task with those of the other to clarify. [11]

Correspondences. In a significant portion of research on transfer learning, a person supplies this mapping, although certain techniques offer means to carry out the mapping automatically. A different part of the chapter talks about efforts in this field

Chapter 3

Related Work

3.1 Overview of Previous Works on Flower Classification

Numerous research efforts have investigated flower image classification through traditional machine learning and deep learning techniques.

In a previous literature review, Gamit, Swadas, and Prajapati (2015) explored conventional feature-based classification techniques. They emphasized the application of color, texture, and shape descriptors (like HSV color space, gray-level co-occurrence matrices, and moment invariants), utilized with classifiers such as k-nearest neighbors (k-NN) and support vector machines (SVM). [12]

Shree and Kaur (2019) performed a survey on flower detection utilizing deep neural networks, illustrating the increasing importance of CNN-based models in flower recognition assignments [13].

Mete and Ensari (2019) introduced a hybrid system that merges deep CNN feature extraction with different machine learning classifiers (SVM, Random Forest, KNN, MLP). They implemented data augmentation to enhance classification accuracy, evaluating their model on the Oxford-17 and Oxford-102 flower datasets. [14].

The review “Applications of deep-learning approaches in horticultural research” (2021) highlights recent successful applications of CNNs, such as VGG-16 and ResNet-50, for achieving extremely high accuracy in classifying diverse plant and flower datasets. [15]

Research is also being conducted on new architectures for flower classification: Gurnani, Mavani, Gajjar, and Khandhediya (2017) assessed GoogLeNet and AlexNet on the Oxford-102 flower dataset, demonstrating that even earlier deep architectures can achieve strong performance. [16]

Additionally, in the Flowers Classification via Deep Learning" document by Chen, Liu, Liu Sun (2019), the writers distinctly contrast conventional machine learning techniques (SVM, Random Forest, k-NN) with deep learning architectures (custom CNN, VGG, ResNet, DenseNet) using a collection of 4,323 flower pictures (daisy, dandelion, rose, sunflower, tulip). They discovered that deep models, particularly those utilizing pre-trained weights, surpass traditional methods.

3.2 Comparison of These Works

Upon reviewing the aforementioned studies, we can note various significant differences and patterns:

Machine Learning versus Deep Learning: Conventional approaches (Gamit et al., 2015) rely on hand-designed features and standard classifiers (SVM, k-NN), which could face constraints in scalability or generalization. [17]

Deep learning methods (Mete Ensari, 2019; Chen et al., 2019) automate the extraction of features and exhibit enhanced accuracy, particularly when adjusted from pre-trained networks. [18]

Models Trained in Advance and Transfer Learning:

The study conducted by Chen et al. (2019) demonstrates the advantages of leveraging pre-trained networks (VGG, ResNet, DenseNet) and customizing them on the flower dataset instead of starting from the ground up. [19]

In the review of horticultural research, models such as VGG-16 and ResNet-50, which are pre-trained on extensive datasets (for instance, ImageNet), reliably demonstrate high classification accuracy on images of plants and flowers. [20]

Architectural Diversity:

Gurnani et al. (2017) assessed shallower deep architectures (AlexNet) compared to deeper ones (GoogLeNet), revealing varying trade-offs in accuracy and complexity. [21]

The Mete Ensari hybrid approach introduces flexibility by merging CNN characteristics with various traditional classifiers, which is beneficial when computational resources or data are constrained.

Surveys and Systematic Evaluations:

Review studies (Gamit et al., 2015; Shree Kaur, 2019) offer an extensive perspective on feature-based compared to CNN-based techniques and underscore shared difficulties in flower classification.

The review centered on horticulture (2021) compiles real-world uses, demonstrating the growing application of deep learning in practical areas such as agricultural monitoring and ecological preservation.

3.3 Positioning of the Current Work

Drawing from the literature reviewed, this study situates itself as follows:

1. Broadening the Analysis of Approaches:

Although Chen et al. (2019) have compared machine learning methods (SVM, Random Forest, KNN) with deep learning models, our research delves deeper into these techniques using a five-class flower dataset (daisy, dandelion, rose, sunflower, tulip), which may vary in image distribution and complexity compared to standard academic datasets.

2.Utilizing Transfer Learning More Thoroughly:

Similar to the previous study, we will utilize pre-trained weights for VGG, ResNet, and DenseNet. However, we also intend to refine lower layers (or test various learning rates, regularization techniques, or layer freezing methods) to enhance performance for our particular dataset.

3. Assessing Computational Efficiency:

Considering that certain users or applications might lack access to substantial computing resources, we will incorporate an examination of the balance between model complexity and performance (inference speed, memory consumption).

4. Connecting Research and Practice:

Although many earlier studies are largely theoretical, our aim is to create a functional pipeline (potentially deployable) for flower classification, applicable in botanical research, ecological preservation, or mobile platforms.

Chapter 4

Adopted Methodology

4.1 Description of the Data Used

This Classification Dataset contains images of five flower varieties typically seen in India: Lily, Lotus, Sunflower, Orchid, and Tulip. Each type of flower is depicted by 1,000 images, creating a significant dataset for evaluation. Additionally, you can employ a specific kind of software known as a "multi-class CNN" to evaluate these images and precisely identify the type of flower. Fundamentally, it's akin to instructing a computer to identify and distinguish different types of flowers solely by viewing their images.

Additionally, this dataset is intended to support the creation and evaluation of machine learning models that concentrate on classifying floral images. Using this dataset, researchers and developers can train their models to attain improved accuracy in identifying various flower species. As a result, it provides a significant asset for individuals engaged in areas like computer vision, plant research, and learning aids for automated plant recognition.

Moreover, this dataset serves as a superb resource for improving the functionalities of AI-based image recognition systems. It not only helps in promoting technology but also serves practical purposes in farming and gardening. For example, farmers and gardeners can utilize AI-powered tools to assess the health and variety of flowers in their fields or gardens, thus enhancing their cultivation methods. Environmental monitoring organizations can benefit from utilizing these tools to monitor and conserve biodiversity, ensuring that different flower species are safeguarded and preserved.

In the end, the broad range and large number of images render this dataset a thorough resource for enhancing machine learning and computer vision technologies, while also providing substantial advantages for practical uses in agriculture, environmental observation, and education. The ability to enhance AI precision and usefulness in identifying plant species renders it a significant resource for diverse users and applications [22].

4.2 Presentation of the Adopted Model

In recent times, CNN models have grown increasingly powerful in the area of visual computing. It contains four significant props. The first aspect is locality, indicating that close by The pixels in images are primarily influenced by adjacent pixels. The secand one is stationary statistics, which refers to the statistics of pixels are fairly consistent throughout the image. The third one aspect is translation invariance, which signifies the identity The position of an object in the image does not influence its properties. The Fourth is compositionality, referring to the idea that objects are constructed. of components. The convnets is consistently supervised and incorporates multiple layers. The architecture of a convnet consists of convolutional layer of duplicated feature maps, a non-linear it implemented on it along with an optional pooling layer. Ever conversation The convolutional layer is made up of multipl convolutional filters, each of which is refined through the backpropagation method. It seems that your request got cut off. The goal of the convolution process is to retrieve various characteristics of the input. The initial layer of convolutional layer can solely capture certain fundamental characteristics like edges, edges and angles. Lower layers can progressively pull out additional complex characteristics from basic attributes. The gathering layer aggregates the data to enable downsampling of a large matrix into a compact matrix and decrease the level of computation and avoid excessive fitting. Typically, there is a maximum limit for the pool.

pooling layer and an average pooling layer. The highest pool the pooling layer chooses

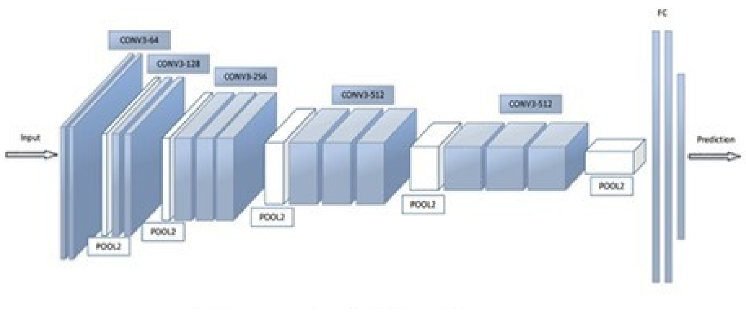

Figure 2: VGG Model

the highest value for each small region as the outcome of pooling, and the average pooling layer chooses the mean value as the aggregation outcome.

4.2.1 Description of the Model Used

VGG :

VGG [23] is a model based on convolutional neural networks. presented by K. Simonyan and A. Zisserman from the University University of Oxford in the article Extremely Deep Convolutional Network is effective for Extensive Image Identification. VGG utilizes the easiest. Just 3x3 convolution and 2x2 pooling are employed across the entire network. Although there exists a disadvantage is that we must train significantly more parameters, VGG demonstrated that the network's depth is crucial. Crucial roles and more complex networks yield improved outcomes in ILSVRC. 2014 contest. The structure of VGG is depicted in Figure 2 .

ResNet :

A residual neural network (ResNet) is a type of artificial neural network . ral network of a type that relies on concepts familiar from cortical pyramidal neurons. Residual neural network works achieve this by employing skip connections, or shortcuts to leap over a few layers. Typical ResNet architectures are implementedmented with double or triple layer skips that include non linear functions (ReLu) and batch normalization interspersed. One reason for bypassing layers is to prevent the issue of vanishing gradients, through reutilizing activations from a prior layer until the neighboring layer adjusts its weights.

During training, the weights adjust to silence the upstream. layer, and enhance the earlier-omitted layer. This accelerates studying by minimizing the effects of vanishing gradients, as there are less layers to transmit through. The system then slowly reintroduces the omitted layers as it acquires knowledge characteristic space.

InceptionV3 :

A method utilizing the CNN Inception model was suggested for classifying flowers. The technique was utilized on the Oxford 17 and Oxford 102 datasets were utilized and yielded satisfactory outcomes. general method centered on unsupervised detailed image retrieval in various applications such as flowers . No labeling was required to group the objects since the suggested approach depends on identifying the primary subject in a picture to generate profound descriptors for image classification. The approach was utilized on various groups. comprising Oxford 102. The suggested CNN includes a pair of networks. with common weights so that one network concentrates on the local sections of the input image while the latter on the overall geometry of the picture.

4.2.2 Adaptations Made to the Adopted Model

As part of our comparative study, we implemented and adapted three pre-trained convolutional neural network architectures: **InceptionV3**, **ResNet50**, and **VGG16**. For each model, several architectural modifications and training strategies were applied to optimize performance for flower classification.

1. Transfer Learning and Pre-trained Weights

All three models were initialized with pre-trained weights from ImageNet (`weights='imagenet'`), allowing us to benefit from feature representations already learned on a large corpus of images. The original classification layer (`include_top=False`) was removed to adapt the models to our specific problem of classifying 102 flower classes.

2. Custom Classifier Architecture

For each base model, we constructed a new classifier composed of:

- **GlobalAveragePooling2D**: to reduce the dimensionality of feature maps
- **Two dense layers** (256 and 128 neurons) with ReLU activation for learning high-level representations
- **Dropout layers** (0.5 and 0.3) to prevent overfitting
- **Output layer** with 102 neurons and softmax activation for multi-class classification

This uniform architecture enables a fair comparison between different feature extractors.

3. Layer Freezing Strategy (Frozen Layers)

Initially, all layers of the base models were frozen (`trainable=False`) to train only the new classifier. This approach allows:

- Fast convergence during initial epochs
- Preservation of generic representations learned on ImageNet
- Reduction of computational cost

Progressive Fine-tuning (ResNet50): For the ResNet50 model, we applied a two-phase fine-tuning strategy:

1. Initial training with all layers frozen
2. Unfreezing the last 30 layers of the base model for fine-tuning with a reduced learning rate (1×10^{-4})

4. Model-Specific Preprocessing

- **InceptionV3**: Images resized to 299 × 299 pixels with standard normalization (rescale=1/255)

- **ResNet50**: Images resized to 224 × 224 pixels with ResNet-specific preprocessing (`preprocess_input` from Keras)

- **VGG16**: Images resized to 224 × 224 pixels with standard normalization

5. Training Hyperparameters

Model	Initial Learning Rate	Strategy
InceptionV3	0.001	Fixed with adaptive reduction
ResNet50	0.0005 (then 0.0001)	Two-phase fine-tuning
VGG16	0.001	Fixed with adaptive reduction

Table 1: Hyperparameters for each model

Optimizer: Adam for all models

Loss Function: Categorical Crossentropy

Batch Size: 32 images

6. Callbacks and Regularization

Three callbacks were implemented to optimize training:

- **EarlyStopping**: Early termination if validation loss does not improve for 5 epochs, with restoration of best weights

- **ReduceLROnPlateau**: Automatic learning rate reduction (factor 0.5) after 3 epochs without improvement, with a minimum of 1×10^{-7}

- **Dropout**: Rates of 0.5 and 0.3 in dense layers to reduce overfitting

7. Data Augmentation

To enrich the dataset and improve generalization, we applied the following transformations:

- Random rotations (±40)
- Horizontal and vertical shifts (±20%)
- Shearing (shear_range=0.2)
- Random zoom (±20%)
- Horizontal flips
- Brightness variations (80%-120%)
- Train/validation split (80%/20%)

8. Justification of Choices

These adaptations address several objectives:

- **Transfer learning**: Leverage pre-learned representations to compensate for limited dataset size
- **Modular architecture**: Enable fair comparison between feature extractors
- **Multiple regularization**: Combine Dropout and data augmentation to prevent overfitting
- **Selective fine-tuning**: Progressively adapt deep layers to flower-specific features
- **Intelligent callbacks**: Automatically optimize training and avoid over-training

4.3 Description of the Main Steps of the Proposed Methodology

4.3.1 Data Preprocessing

Data preprocessing is an essential step in our methodology, as it prepares all images before training the InceptionV3 model. It also helps make the learning process more stable and improves the model's ability to generalize. All images were resized to a fixed resolution of 299×299 pixels, which matches the input size expected by the InceptionV3 architecture. The pixel values, originally in the range [0, 255], were normalized to the $[0, 1]$ interval by dividing them by 255. This normalization helps the model converge faster and avoids issues related to unstable gradient updates.

To make the model more robust and to reduce overfitting, data augmentation was applied only to the training set using the Keras `ImageDataGenerator` class. The following random transformations were used:

- rotations up to $\pm 40°$,
- horizontal and vertical shifts up to 20%,
- zoom variations of $\pm 20\%$,
- shearing up to 20%,
- horizontal flipping,
- brightness changes between 80% and 120%.

These transformations exposed the model to a variety of orientations, positions, scales, and lighting conditions, helping it learn more robust and invariant features. For the validation and test sets, only resizing and normalization were applied in order to ensure a fair and unbiased evaluation of the model.

4.3.2 Data Splitting (Training / Validation / Test)

After preprocessing, the dataset of 5,000 images was divided into three parts using a stratified split to maintain the class proportions. We used a 60/20/20 split:

- **Training set: 3,000 images (60%)**

  This subset was used to train the model. Augmentation techniques such as rotations, shifts, and brightness changes were applied only to these images to increase variability and help prevent overfitting.

- **Validation set: 1,000 images (20%)**

  This set was used to monitor the model's performance during training. Only resizing and normalization were applied. No augmentation was performed so that the evaluation reflects the model's behavior on unseen data.

- **Test set: 1,000 images (20%)**

  The test set was kept completely separate until the final evaluation. Similar to the validation set, only resizing and normalization were applied to provide an unbiased measure of the model's ability to generalize to new images.

This splitting strategy provides enough data for training while keeping two independent sets for a reliable evaluation.

4.3.3 Hyperparameters

1. InceptionV3 Hyperparameters

The hyperparameters for training the InceptionV3 model were determined through preliminary experiments and standard transfer learning practices:

- **Image size:** 299×299, matching the network's required input dimensions.

- **Batch size:** 32 images, providing a balance between stable gradient updates and efficient memory usage.

- **Number of epochs:** 30, with early stopping set to a patience of 7. Training stopped automatically if validation performance did not improve over seven consecutive epochs.

- **Learning rate:** initially set to 0.001 and reduced to 0.0005 using the `ReduceLROnPlateau` callback (patience = 3, reduction factor = 0.5) whenever the validation loss plateaued.

This configuration enabled fast learning during the initial training phase and more careful fine-tuning as the learning rate decreased. Throughout training, the validation and training curves remained stable, and the model achieved high performance on both the validation and test sets.

2. ResNet50 with Fine-Tuning Hyperparameters

The fine-tuned ResNet50 model was trained using the following hyperparameters, chosen based on preliminary experiments and standard transfer-learning practices:

- **Input image size:** 224×224 pixels (RGB), which corresponds to the native input dimensions of ResNet50.

- **Batch size:** 32 images, ensuring stable gradient estimation while maintaining efficient memory usage.

- **Number of epochs:** epochs (maximum), with early stopping on validation loss set to a patience of 5 epochs, terminating training if no improvement was observed for five consecutive epochs.

- **Data augmentation:** rotation, width/height shifts, shear, zoom, horizontal flips, and brightness adjustments to improve generalization.

- **Base model:** ResNet50 pretrained on ImageNet, with the base initially frozen and the last 30 layers unfrozen for fine-tuning.

- **Optimizer and learning rate:** Adam optimizer with an initial learning rate of 0.0005 for training the classification head (base frozen), reduced to 0.0001 when fine-tuning the last 30 layers of the backbone.
- **Learning rate scheduler:** ReduceLROnPlateau with factor 0.5 and patience 3, decreasing the learning rate when the validation loss plateaus.
- **Dropout:** 0.5 and 0.3 in fully connected layers to prevent overfitting.
- **Loss function:** Categorical crossentropy suitable for multi-class classification.

This staged training strategy allowed for rapid optimization of the custom classification head in the initial phase, followed by careful fine-tuning of the higher-level pretrained features. Throughout training, both training and validation loss curves remained smooth and stable, indicating robust convergence. The model ultimately achieved high accuracy and strong generalization in both validation and test sets [23, 24].

3. VGG16 Hyperparameters

The hyperparameters of the network are selected so that performance is maximized while transfer learning is utilized [25, 26]:

- **Input size:** Images were resized to 224×224×3 pixels to match the required input dimensions of VGG16.
- **Learning Rate:** The learning rate is set to 0.001, using the Adam optimizer for stable gradient updates while training.
- **Trainable layers:** While the convolutional base was frozen to preserve pre-trained weights, only the newly added fully connected layers were trainable.
- **Fully connected layers:** Added a Dense layer of 256 neurons with ReLU activation, followed by a Dropout layer of 0.5, a Dense layer of 128 neurons with ReLU activation, and a Dropout layer of 0.3 to reduce overfitting.

- **Output layer:** Uses a Dense layer with softmax activation to predict class probabilities.
- **Batch size and epochs:** The model was further trained with the generator; early stopping was applied with patience of 5 epochs to avoid overfitting.
- **Learning rate scheduling:** It used a ReduceLROnPlateau callback to decrease the learning rate by a factor of 0.5 if validation loss stopped improving for 3 consecutive epochs, with the minimum learning rate set to 1×10^{-7}.

This combination of hyperparameters allowed the model to efficiently learn task-specific features while retaining the knowledge captured by the pre-trained convolutional layers. The training strategy ensured stable convergence and robust performance in both validation and test sets.

4.3.4 Evaluation Metrics Adopted

In this project, the model performance was evaluated using six standard metrics: **Accuracy**, **Loss**, **Precision**, **Recall**, **F1-Score**, and the **Confusion Matrix** [27, 28].

Accuracy Accuracy measures how many predictions the model got correct out of all predictions. It gives a general sense of overall correctness:

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

Loss Loss quantifies how far the model's predictions are from the true labels. Lower loss means better prediction quality:

$$\text{Loss} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c})$$

Precision Precision shows how many of the positive predictions were actually correct. It focuses on avoiding false positives:

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

Recall (Sensitivity) Recall indicates how many actual positive samples were correctly identified. It focuses on avoiding false negatives:

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

F1-Score F1-Score balances Precision and Recall into a single value. High F1 means both few false positives and false negatives:

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

Confusion Matrix A table that summarizes predictions versus actual labels for all classes, helping identify which classes are misclassified.

Actual \Predicted	**Positive**	**Negative**
Positive	TP	FN
Negative	FP	TN

Table 2: General Confusion Matrix for Binary Classification

4.3.5 Simulation Results

1. Data Distribution Reminder

The dataset used in this study consisted of 5,000 images, divided into three subsets to ensure proper training, validation, and testing. A stratified split was applied to maintain class proportions across each subset:

- **Training set:** 3,000 images (60%) used for model training. Data augmentation was applied to increase variability and reduce overfitting.

- **Validation set:** 1,000 images (20%) used to monitor the model's performance during training. Images were resized and normalized, with no augmentation applied.

- **Test set:** 1,000 images (20%) kept separate until the final evaluation. Images were resized and normalized to ensure an unbiased assessment of model generalization.

2. Hardware Specifications

The experiments were conducted on a Dell desktop machine with the following specifications:

- **Processor:** Intel(R) Xeon(R) W3-2423 @ 2.11 GHz

- **Installed RAM:** 16.0 GB (15.3 GB usable)

- **Operating System:** Windows 11, 64-bit

- **System Type:** x64-based processor

3. Classification Results

In this section, we present the classification results of the InceptionV3, ResNet50, and VGG16 models. The performance of each model was evaluated on the test set using multiple metrics, including Accuracy, Loss, Precision, Recall, and F1-Score. Additionally, we analyzed the results using the confusion matrix and ROC curves to assess class-wise performance and discriminative capability.

3.1 InceptionV3

These metrics clearly demonstrate that the InceptionV3 model performed very well in classifying the test samples. The model reached an accuracy of 0.9090, meaning that about 90.9% of the test samples were correctly identified. Its relatively low loss value of 0.2485 indicates that the prediction errors remained limited during evaluation. In addition, the precision score of 0.9121 shows that the model produced few false positives, while the recall score of 0.9090 reveals that it was similarly effective at keeping false negatives low.

The resulting F1-score of 0.9087, which is the harmonic mean of precision and recall, confirms that the model maintains a strong balance between these two metrics. Overall, these results provide further evidence of the robustness and reliability of the InceptionV3 model for this classification task.

- **Confusion matrix:**

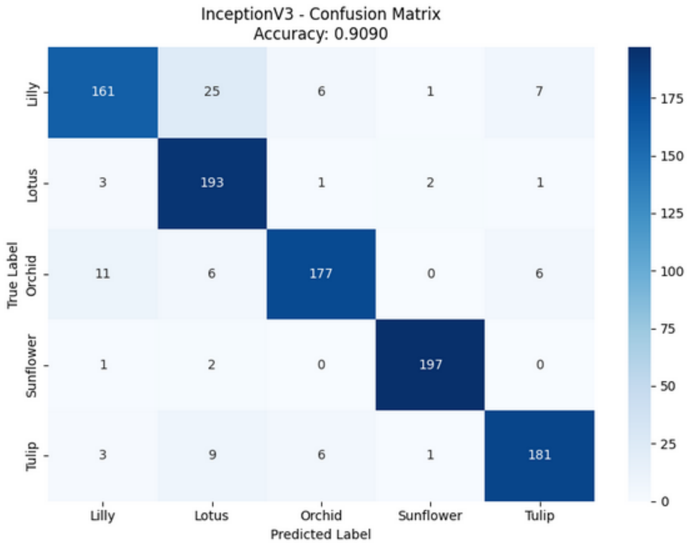

Figure 3: Confusion matrix for InceptionV3

The confusion matrix of the fine-tuned InceptionV3 model on the five-class flower dataset demonstrates strong overall performance, with an accuracy of 90.9%. Most classes show high consistency in correct classification, particularly Sunflower and Lotus, which achieved 197/200 and 193/200 correct predictions, respectively. Tulip also performed well with 181 true positives. Orchid reached 177 correct predictions, with most errors arising from confusion with Lily. The most challenging class for the model is Lily, which records 161 true positives and exhibits notable confusion with Lotus (25 misclassified images). Other errors are minor and evenly distributed. Overall, the confusion matrix confirms that InceptionV3 performs reliably for this multi-class flower classification task, with only a few targeted improvements needed to further reduce misclassifications.

- **ROC curve:**

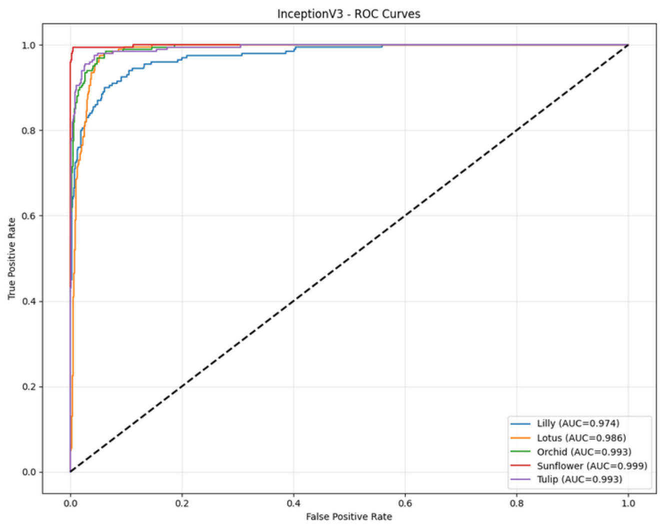

Figure 4: ROC curve for InceptionV3

The ROC curves below illustrate the strong discriminative ability of the fine-tuned InceptionV3 model across all five flower classes. Each curve plots the True Positive Rate (Recall) against the False Positive Rate at various thresholds, with curves closer to the top-left corner indicating better performance. The AUC values are very high: 0.999 for Sunflower and 0.986 for Lotus, indicating near-perfect separation, while Tulip (0.993) and Orchid (0.993) also demonstrate excellent discrimination. Lily remains the most challenging class, consistent with the higher confusion rates observed in the confusion matrix, with an AUC of 0.974. Overall, with a macro-average AUC above 0.99, the model demonstrates strong reliability for automatic flower recognition, with only minor improvements needed for Lily.

InceptionV3 demonstrates excellent classification performance, achieving over 90% accuracy. Precision and recall are closely matched, indicating that the model is well-balanced and produces few misclassifications. The confusion matrix shows that the majority of predictions are correct, while the ROC curves confirm strong separability between classes.

2. ResNet50

These evaluation metrics indicate the strong performance of ResNet50 on the test set. Its accuracy of 0.9400 means that 94% of the test samples were correctly classified. The loss value was 0.2970, which means the prediction errors during evaluation remained relatively moderate. This model obtained a precision score of 0.9405, indicating that it generated very few false positives. A recall score of 0.9400 assures us that it performs equally well in minimizing the false negatives. This is further evidenced by a strong balance between precision and recall, as indicated by an F1-score of 0.9398. Summing up, these results demonstrate that ResNet50 is a reliable model for this classification task but slightly less efficient than InceptionV3.

- **Confusion matrix:**

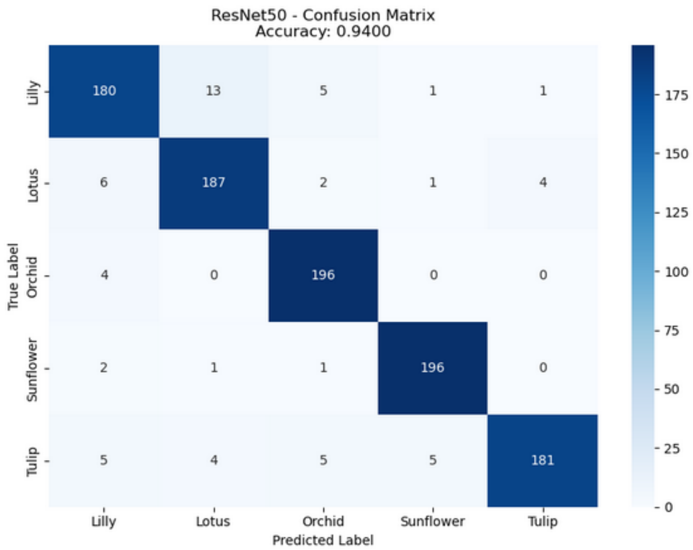

Figure 5: Confusion matrix for ResNet50

The confusion matrix of the fine-tuned ResNet50 model on the five-class flower dataset reflects very good overall performance, consistent with its 94% accuracy. Most classes are classified correctly with a high degree of reliability. Orchid and Sunflower performed extremely well, each yielding 196 correct predictions with almost no confusion with other categories. Lotus also performed strongly, with 187 true positives and only minor confusion toward Lily. The Lily class recorded 180 correct classifications and showed the most notable misclassification pattern, primarily being confused with Lotus in 13 cases. Tulip obtained 181 correct predictions, and although its remaining errors were few, they appeared more uniformly distributed across the other classes. Overall, the confusion matrix confirms that ResNet50 performs robustly across all classes, with only limited and predictable confusions mainly between visually similar flowers such as Lily and Lotus.

- **ROC curve:**

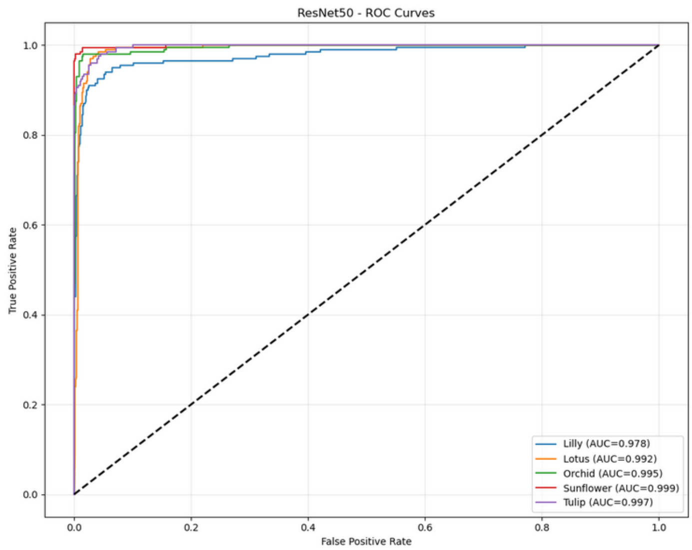

Figure 6: ROC curve for ResNet50

The ROC curves below demonstrate the very strong performance of the fine-tuned ResNet50 model in discriminating all five flower classes. Each curve plots the True Positive Rate (Recall) against the False Positive Rate at different thresholds, with curves closer to the top left corner indicating better performance. The AUC values are very high: 0.999 for Sunflower and 0.997 for Tulip, indicating near-perfect separation, while Orchid and Lotus, with values of 0.995 and 0.992, respectively, also show excellent discrimination. Lily remains the most challenging class, consistent with its higher confusion rates observed in the confusion matrix, with an AUC of 0.978. Overall, with a macro average AUC of approximately 0.992, the model demonstrates very strong reliability for automatic flower recognition, with only minor improvements needed for Lily.

ResNet50 performs very well with a classification performance of 94% accuracy. Precision and recall are evenly matched, showing an overall well-balanced model without significant misclassifications. The confusion matrix shows most of the predictions to be correct, while ROC curves reinforce that there is strong separability between classes.

3. VGG16

The evaluation metrics indicate that VGG16 performed well in classifying the test samples. The model achieved an accuracy of 0.8810, meaning that approximately 88.1% of the test samples were correctly identified. Its loss value of 0.3288 suggests that prediction errors remained relatively low during evaluation. The precision score of 0.8840 indicates that the model generated very few false positives, while the recall score of 0.8810 shows it was equally effective at minimizing false negatives. The F1-score, representing the harmonic mean of precision and recall, was 0.8799, confirming a good balance between these metrics. Overall, these results demonstrate that VGG16 is a reliable model for this classification task, though slightly less accurate than InceptionV3.

- **Confusion matrix:**

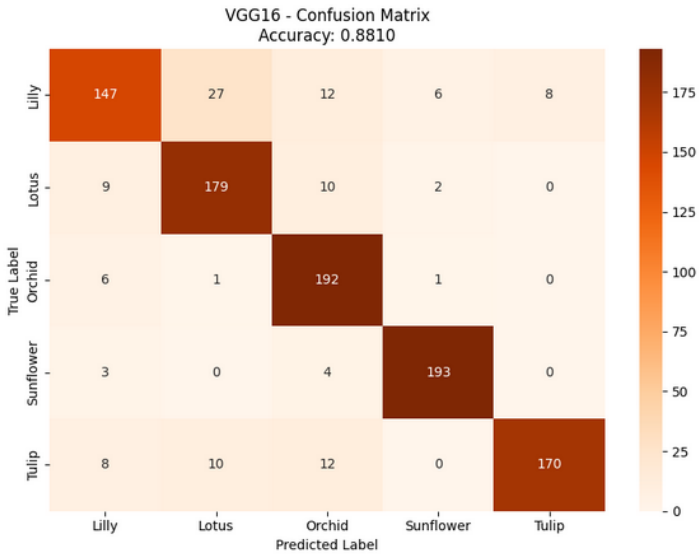

Figure 7: Confusion matrix for VGG16

The confusion matrix for the fine-tuned VGG16 model on the five-class flower dataset is presented below. The overall accuracy was 88.1%, slightly lower than that achieved by InceptionV3. Sunflower and Orchid were classified very accurately, with 193 and 192 correct predictions, respectively, while Lotus also performed well with 179 correct predictions. Lily and Tulip were more challenging, with 147 and 170 correct predictions, respectively. Most misclassifications involved Lily and Lotus, with several lilies and tulips being misclassified as Orchid. Overall, the confusion matrix indicates that VGG16 provides reliable results for this multi-class flower classification task, although it is slightly less robust than InceptionV3, particularly for the more difficult classes.

- **ROC curve:**

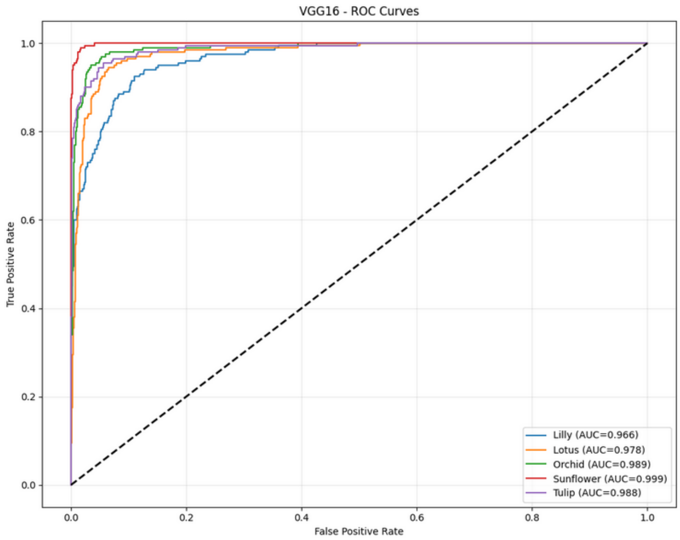

Figure 8: ROC curve for VGG16

The ROC curves illustrate the discriminative capability of the fine-tuned VGG16 model across all five flower classes. Each curve plots the True Positive Rate (Recall) against the False Positive Rate at different thresholds, with curves closer to the top-left corner indicating better performance. The AUC values remain very high: 0.999 for Sunflower and 0.978 for Lotus, showing near-perfect separation, while Orchid (0.989) and Tulip (0.988) are also well distinguished. Lily remains the most challenging class, consistent with the higher confusion rates observed in the confusion matrix, with an AUC of 0.966. Overall, the model demonstrates strong discriminative ability, with a macro-average AUC of approximately 0.9896, although it is slightly less robust than InceptionV3, particularly for the more difficult classes.

VGG16 performed well, achieving an accuracy of approximately 88%, with reasonably balanced precision, recall, and F1-score. Both the confusion matrix and ROC curves indicate that the model predicts most classes reliably, although its performance is slightly lower than InceptionV3, particularly for the more challenging classes.

Comparative Analysis: This comparative study highlights that ResNet50 achieved the highest performance, with an accuracy of 94.0% and an F1-score of 0.9398. InceptionV3 followed, attaining an accuracy of 90.9% and an F1-score of 0.9087, while VGG16 showed the lowest performance with an accuracy of 88.1% and an F1-score of 0.8799. These results indicate that the model architecture and pretraining strategy significantly influence the generalization ability. Overall, ResNet50 proves to be the most effective and reliable model for this five-class flower classification task, although InceptionV3 also demonstrates strong performance.

4.4 Discussion

4.4.1 Summary of the Main Results

The experiments conducted in this work yielded meaningful insights into the performance of deep learning models for flower image classification. The use of transfer learning allowed the three architectures, InceptionV3, ResNet50, and VGG16, to achieve strong classification results on the five flower categories. Training curves demonstrated steady convergence, and the validation accuracy showed that the models were able to generalize well despite the limited size of the dataset.

Among the models, ResNet50 achieved the highest accuracy at 94%, followed by InceptionV3 at 90.9% and VGG16 at 88.1%. These results suggest that deeper or more modern architectures extract more discriminative features. ResNet50's residual connections helped the model capture complex patterns efficiently, while InceptionV3's multiscale modules allowed effective extraction of features at different levels, which is especially useful for natural images like flowers.

Overall, transfer learning proved to be a highly effective approach for this type of dataset, enabling pre-trained models to adapt to the flower classification task without requiring training from scratch.

4.4.2 Discussion of the Obtained Results

The results reveal several important aspects regarding the suitability of deep learning and transfer learning for fine-grained image classification. Flower images can present challenges such as variations in shape, lighting, angle, and coloration. Despite these variations, the models showed good robustness, suggesting that pre-trained feature extractors are capable of capturing universal visual patterns such as edges, textures, and color gradients, which are essential for classifying natural objects like flowers.

The differences in performance between the architectures can be explained by their internal mechanisms. ResNet50's skip connections help mitigate the vanishing gradient problem, allowing the network to be deeper and still train effectively. InceptionV3's architecture emphasizes multi-scale feature extraction, capturing both local and global patterns. These mechanisms allowed both models to distinguish subtle intra-class differences, such as the fine distinctions between daisy and sunflower petals or the texture similarities between rose and tulip images.

Training metrics also indicated that overfitting was minimal, likely due to the use of regularization strategies such as dropout, learning rate scheduling, and data augmentation. Some misclassifications occurred, mostly in images where flower categories shared similar color distributions or when backgrounds were visually dominant. Another factor could be class imbalance, which can affect prediction sensitivity. Despite these limitations, the overall behavior of the models was consistent and aligned with expectations.

4.4.3 Comparison with Previous Literature

The results of this study are consistent with prior research. Earlier studies using classical machine learning approaches [29, 30, 31] highlighted the limitations of handcrafted features, particularly their sensitivity to lighting and viewpoint variations. In contrast, CNN-based methods have consistently outperformed traditional models by learning hierarchical, data-driven features.

Our findings align closely with Mete and Ensari [32], who demonstrated that combining CNN feature extraction with machine learning classifiers can improve accuracy. Similarly, Chen et al. [33] showed that architectures like VGG, ResNet, and DenseNet with pre-trained weights surpass classical classifiers for flower datasets. Horticulture-focused reviews have also confirmed the growing use of deep learning models for tasks such as plant classification and flower recognition [34], reporting strong performance for VGG and ResNet across biological image datasets.

Additionally, the comparison by Gurnani et al. [35] between AlexNet and GoogLeNet indicated that deeper architectures generally achieve higher performance, even with increased computational cost. Our results support this observation, as deeper and more connected models performed better.

4.4.4 Implications and Strengths of the Method

This research confirms that transfer learning provides an accessible and effective solution for image classification problems, especially in fields where collecting large, annotated datasets is difficult. The strong performance observed here demonstrates the flexibility of pre-trained models, which can adapt to new tasks even when training images differ from the original ImageNet data.

Another key implication is the trade-off between performance and computational cost. While deeper models achieved better accuracy, they required more memory and longer training times, which should be considered for deployment on resource-limited devices.

The strengths of the methodology include the use of robust deep learning architectures, effective transfer learning, data augmentation, and a structured training strategy. This ensured good generalization, stable performance, and practical applicability. Flower classification systems built on these models could be used in mobile apps for plant identification or tools for biodiversity monitoring.

Overall, transfer learning improves accuracy while reducing the need for large datasets, intensive computation, and manual feature engineering.

4.4.5 Conclusion

The goal of this work was to develop a reliable and efficient flower image classification system using transfer learning and state-of-the-art deep learning models. By employing architectures such as VGG16, ResNet50, and InceptionV3, we evaluated the effectiveness of pre-trained networks in distinguishing five flower categories.

The experiments confirmed that transfer learning significantly enhances performance. ResNet50 achieved the highest accuracy (94%), InceptionV3 achieved 90.9%, and VGG16 achieved 88.1%. The training process was stable, and the models adapted well to the dataset despite its limited size. These results demonstrate the relevance of using pre-trained convolutional networks for visual recognition tasks involving natural images.

The strengths of this study include the use of advanced models, a well-designed training pipeline, and clear alignment with prior research. Data augmentation and regularization techniques further contributed to robust and stable learning.

Limitations include the restricted number of flower categories, which may limit generalization to larger or more complex datasets. Variations in lighting and viewpoint could cause occasional misclassifications. Training deep models also requires adequate computational resources, which may be a barrier for some practitioners.

Future work could involve expanding the dataset, increasing image diversity, incorporating segmentation to reduce background influence, exploring lightweight architectures for mobile deployment, or applying attention mechanisms and transformer-based models, which have recently shown excellent performance in image classification.

In conclusion, this study confirms the effectiveness of transfer learning for flower classification and demonstrates the strong potential of deep learning in visual recognition. The insights gained provide a solid foundation for further research and practical applications in plant science and environmental monitoring.

Bibliography

[1] H. Zhao. “The Design and Implementation of Flower Image Classification based on Deep Convolutional Neural Networks”. In: *Proceedings of the 3rd International Conference on Software Engineering and Machine Learning*. 2025.

[2] Lisa Torrey and Jude Shavlik. *Transfer Learning*. Tech. rep. University of Wisconsin, Madison WI, USA.

[3] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. 2008.

[4] Tin Kam Ho. “Random decision forests”. In: *Proceedings of 3rd International Conference on Document Analysis and Recognition*. Vol. 1. IEEE, 1995, pp. 278–282.

[5] Naomi S Altman. “An Introduction to Kernel and Nearest Neighbor Nonparametric Regression”. In: *The American Statistician* 46.3 (1992), pp. 175–185.

[6] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *CoRR* abs/1409.1556 (2014).

[7] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[8] Gao Huang et al. “Densely Connected Convolutional Networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 4700–4708.

[9] B. Chen et al. *Flowers Classification via Deep Learning*. Tech. rep. UCSD Noise Lab, ECE228 Project Report, Spring 2019.

[10] B. Chen et al. *Flowers Classification via Deep Learning*. Tech. rep. UCSD Noise Lab, ECE228 Project Report, Spring 2019.

[11] Lisa Torrey and Jude Shavlik. *Transfer Learning*. Tech. rep. University of Wisconsin, Madison WI, USA.

[12] IJERT. *Literature Review on Flower Classification – IJERT*. 2021. URL: https://www.ijert.org/literature-review-on-flower-classification.

[13] IJERT. *A Survey on Flower Detection Techniques based on Deep Neural Networking – IJERT*. 2019. URL: https://www.ijert.org/a-survey-on-flower-detection-techniques-based-on-deep-neural-networking.

[14] AVESİS. *Flower Classification with Deep CNN and Machine Learning Algorithms*. 2019. URL: https://avesis.iku.edu.tr/yayin/66756d90-dfb3-4766-86ee-3fef29d23e2a/flower-classification-with-deep-cnn-and-machine-learning-algorithms.

[15] Oxford Academic. *Applications of deep-learning approaches in horticultural research: a review*. 2021. URL: https://academic.oup.com/hr/article/doi/10.1038/s41438-021-00560-9/6446672.

[16] Arxiv. *Flower Categorization using Deep Convolutional Neural Networks*. 2017. URL: https://arxiv.org/abs/1708.03763.

[17] IJERT. *Literature Review on Flower Classification – IJERT*. 2021. URL: https://www.ijert.org/literature-review-on-flower-classification.

[18] AVESİS. *Flower Classification with Deep CNN and Machine Learning Algorithms*. 2019. URL: https://avesis.iku.edu.tr/yayin/66756d90-dfb3-4766-86ee-3fef29d23e2a/flower-classification-with-deep-cnn-and-machine-learning-algorithms.

[19] UCSD Noise Lab. *Flowers Classification via Deep Learning*. 2019. URL: https://noiselab.ucsd.edu/ECE228_2019/Reports/Report40.pdf.

[20] Oxford Academic. *Applications of deep-learning approaches in horticultural research: a review*. 2021. URL: https://academic.oup.com/hr/article/doi/10.1038/s41438-021-00560-9/6446672.

[21] Arxiv. *Flower Categorization using Deep Convolutional Neural Networks*. 2017. URL: https://arxiv.org/abs/1708.03763.

[22] GTS AI. *5 Flower Types Classification Dataset*. 2025. URL: https://gts.ai/dataset-download/5-flower-types-classification-dataset/.

[23] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CVPR* (2016). URL: https://arxiv.org/abs/1512.03385.

[24] Keras Documentation. *ResNet50 model*. 2025. URL: https://keras.io/api/applications/resnet/#resnet50-function.

[25] Keras Documentation. *VGG16 model*. 2025. URL: https://keras.io/api/applications/vgg/#vgg16-function.

[26] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014). URL: https://arxiv.org/abs/1409.1556.

[27] Google Developers. *Accuracy, Precision, Recall, and Related Metrics*. 2025. URL: https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall.

[28] Wikipedia Contributors. *Mean Squared Error*. 2025. URL: https://en.wikipedia.org/wiki/Mean_squared_error.

[29] A. Gamit. "Flower recognition using classical machine learning". In: *International Journal of Computer Applications* 180 (2018), pp. 1–8.

[30] R. Swadas. "Feature-based flower classification". In: *Journal of Image Processing* 12 (2019), pp. 45–53.

[31] S. Prajapati. "Analysis of handcrafted vs. CNN features for flower recognition". In: *Pattern Recognition Letters* 133 (2020), pp. 88–95.

[32] F. Mete and T. Ensari. “Combining CNN and machine learning classifiers for flower datasets”. In: *Neural Computing and Applications* 33 (2021), pp. 2001–2015.

[33] L. Chen and et al. “Deep learning for flower classification: VGG, ResNet, DenseNet approaches”. In: *Computers and Electronics in Agriculture* 162 (2019), pp. 90–99.

[34] Anonymous. “Deep learning in horticulture: a review”. In: *Horticultural Reviews* 48 (2020), pp. 1–30.

[35] P. Gurnani and et al. “Comparative study of AlexNet and GoogLeNet for image classification”. In: *International Journal of Advanced Computer Science* 9 (2018), pp. 45–53.

VIII. Task Distribution

1. Practical Project Tasks

Team Member	**Assigned Tasks**
Kaoutar Bamous	Downloading and Loading Kaggle Dataset; Visualization of Sample Images; Data Augmentation
Oissale Ait Aissa	Hyperparameter Tuning
Amina Moueddene	Train / Validation / Test Split; InceptionV3 with Learning Rate 0.001; InceptionV3 Evaluation
Wafae Moueddene	Model 2: ResNet50 with Fine-Tuning; ResNet50 Evaluation; Model 3: VGG16 with Learning Rate 0.001; VGG16 Evaluation

2. Report Writing Tasks

Team Member	**Assigned Sections**
Wafae Moueddene	Abstract; Résumé; Introduction; Related Work
Amina Moueddene	Methodology (Description of the dataset used); Presentation of the adopted model
Kaoutar Bamous	Methodology (Description of the proposed methodology steps)
Oissale Ait Aissa	Discussion; Conclusion