Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# FloraFusionNet: A Discrete Hybrid CNN–Transformer Framework with Attention Optimization and Comparative Evaluation for Fine-Grained Flower Classification

**Abhilasha Varshney**

2021210989.abhilasha@dr.sharda.ac.in

Sharda University

**Sudeep Varshney**

Sharda University

**Ankur Choudhary**

Accenture

**Research Article**

**Keywords:**

**Posted Date:** November 17th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-8496596/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# FloraFusionNet: A Discrete Hybrid CNN–Transformer Framework with Attention Optimization and Comparative Evaluation for Fine-Grained Flower Classification

Abhilasha Varshney [1],Dr. Sudeep Varshney[2], Dr. Ankur Choudhary[3]
[1,2] Department of Computer Science, Sharda School of Engineering & Technology ,Sharda University, Greater Noida
[3] Accenture, Noida
[1] abhilashabharatgupta@gmail.com, [2] sudeep.varshney@sharda.ac.in, [3] ankur.tomer@gmail.com,

*Abstract— Categorization of flowers is important for numerous application areas, such as biodiversity monitoring, agricultural automation and botanical research. Deep learning has introduced a number of new neural network architectures which have shown good performances to more difficult image classification problems. In this paper, we provide a detailed comparative analysis among deepest deep learning networks for multi-class flower recognition task on the Oxford 102 Category Flower Dataset which is recognized benchmark in terms of high intra-class variability and interclass similarity.We test and compare how different deep architectures, that is traditional CNNs (VGG16, ResNet50 and EfficientNet-B0) versus the recent ViT and Hybrid CNN-Transformer perform. The evaluation is based on various performance metrics including overall classification accuracy, precision, recall, F1-score, model complexity (number of parameters), training efficiency and inference latency. All models are fine-tuned based on generalized transfer learning strategy and we also make it strictly uniform to provide a fair comparison of different approaches. Experimental results show that hybrid models so created, where the model can be seen as stacked CNN based encoder with attention-based transformer encoder outperforms individual component achieving peak accuracy of 97.85% on extended test set. However, CNN-based models have faster convergence and less computing cost which are suitable for resource-limited scenarios. This research presents a unified and comprehensive understanding of the trade off among model performance, efficiency, simplicity, and accessibility, which is worth referring to when deploying deep learning solutions into practical botanical classification systems.*

***Keywords: : Flower Classification, Deep Learning, Convolutional Neural Networks (CNNs), Transfer Learning, Multi-Class Classification, Image Recognition, Neural Networks Comparative Analysis, computer vision.***

## I. INTRODUCTION

Identifying the species of flowers accurately is an important task for plant taxonomy, biodiversity conservation, environmental monitoring, precision agriculture and pharmaceutical botany. This accurate identification makes possible tools such as automated plant cataloging, invasive species observation and real-time flora tracking by mobile or embedded devices. In the past, it was commonly assigned manually by botanists or field expert, and primarily based on morphologic features such as petal shapes, textures, symmetry and color-patterns. However, this is a very time-consuming and subjective process and cannot be easily applied to larger ecological surveys or citizen science projects.

The explosion of computer vision and deep learning has enabled a new paradigm for automating flower classification with unparalleled accuracy and de-pendability. In particular, for image recognition tasks, deep convolutional neural networks (CNNs) have shown its capability with great success owing to hierarchical feature extraction and spatial invariance. Models such as AlexNet [12], VGG16 [31] ResNet50[19], InceptionV3 [37] and EfficientNet have become the de facto standard for many visual classification problems and have also been applied to plant and flower classifications with promising results.

More recently, transformer-based models that were popularized for sequence modeling in natural language processing (NLP) (e.g., BERT, GPT) have been tailored to different vision tasks with the Vision Transformer (ViT). Unlike CNNs which learn local patterns via convolutions, ViTs learn

long-range dependencies through self-attention, thus allowing a more global understanding of the image content. These models have achieved the best results in image classification, notably, in tasks with small inter class distinctions as fine-grained visual categorization (FGVC).
In botanical datasets, like **Oxford 102 Category Flower Dataset**, image classification becomes particularly challenging due to:

- **High intra-class variability**, caused by environmental conditions (lighting, background clutter, partial occlusion, varying stages of blooming).
- **High inter-class similarity**, with visually similar species differing in only subtle morphological details.
- **Class imbalance**, where some flower categories have significantly fewer samples than others.

Despite the availability of deep learning tools, many existing works either rely on a **single architecture** or fail to **compare performance across different model families** (CNNs vs Transformers) using rigorous evaluation criteria. Moreover, **real-world deployment** requires models that are not only accurate but also efficient in terms of computational complexity, memory usage, training convergence, and inference time.
This study is motivated by the need to perform a **comprehensive and fair evaluation** of diverse deep learning approaches for flower classification on the Oxford 102 Category dataset. The main **contributions** of this work are as follows: We extensively compared a number of deep learning models such as CNN architectures (VGG16, ResNet50, EfficientNet-B0), Transformer-based architectures (Vision transformer or ViT) and Hybrid ones composed of the CNN in its backbone and attention heads based on Transformers. We take each model through the same preprocessing and training pipeline using transfer learning, fine-tuning and data augmentation. Our key evaluative parameters are described in detail as follows:
o The Accuracy, Precision, Recall and F1-Score of the classification.
o Time of training, convergence rate and computational complexity.
o Inference latency and model size for deployment system.
We further conduct in-depth performance analysis including ablation study on the effect of different design choices (e.g., optimizer, learning rate, data augmentation) for model performance. We talk about how well each model is able to perform in practical scenarios such as on-device inference, mobile deployment and field-based plant recognition. It is hoped that this article provides a baseline for researchers and practitioners in the areas of computational botany, AI-driven taxonomy, and agricultural automation to understand how to choose and adapt deep learning models toward fine-grained image-based flower classification.

## II. Related Work

Automatic flower categorization has matured from classical machine learning techniques into deep learning structures that are able to process large-scale, fine-grained image collections. This section provides an overview of the related work in the field, which can be classified into three categories: traditional approaches, CNN-based methods, and transformer-based models. It also points to deficiencies that provide the impetus for the current research.

### *2.1 Traditional Machine Learning Approaches*

The earliest systems for flower classification relied on handcrafted features such as color histograms, edge detection (Sobel, Canny), shape descriptors, and texture patterns (e.g., LBP,Gabor filters). These features were utilised along with classifiers such as Support Vector Machines (SVM), k- Nearest Neighbours (k-NN) and Random Forests.

For example, Nilsback and Zisserman (2008) presented a SIFT and color descriptor based approach for the Oxford 102 flower dataset, where an ~72.8% precision rate was obtained by utilizing multiclass SVM classifiers. However, these systems suffer from reliance of hand-crafted features and do not generalize well across different environmental settings. One of the first papers with Oxford 102 Flower Dataset. The used handcrafted features are Color histogram features to represent the color of dominant petal and SIFT descriptors regarding shape and texture. It aggregated the features learned (pooled) with MKL. For classification we used a linear Support Vector Machine (SVM) with one-vs-rest.

### *2.2 Deep Learning-Based CNN Models*

The transition to CNNs enabled end-to-end feature learning, minimizing the necessity for hand-crafted inputs. Ozdemir et al. (2017) used AlexNet and GoogLeNet on Oxford 102 dataset, which achieved ~83–85% accuracy for transfer learning. The transfer learning approach was adopted, which involved the initialisation of models with pre-trained weights using ImageNet. It used little compilement - random cropping and normalization. Reyes et al. (2019) applied ResNet50 and DenseNet121 with fine-tuning, which gave a classification accuracy of more than 90%. It adopted residual learning (ResNet) and dense connections (DenseNet) to improve gradient flow. This model is fine-tuned on Oxford 102 with ImageNet weights. It also exploited early stopping and learning rate schedulers to prevent overfitting. ("lenet", lenet, "data_augmentation" (rotation, flipping, zooming).
Zhang et al. (2020) used EfficientNet-B0 and data augmentation, obtaining ~94.5% accuracy. It used the compound scaling from EfficientNet while balancing model depth, width, and resolution. The approach in this work employed very little parameter tuning using a transfer learning. It used cutout augmentation and mixup regularization for better generalization. Such an approach was designed to minimize the model size and running time to be efficient enough for mobile deployment.
Karthik et al. (2022) put forward a CNNs ensemble approach and obtained ~96.3% of accuracy.
These works suggest that CNNs are performing well, particularly when using transfer learning and regularisation techniques. However, due to the shallow receptive fields of CNNs, they generally can not capture large spatial dependencies well especially in fine-grained classes.

### *2.3 Vision Transformer (ViT) and Attention-Based Models*

Transformers, introduced in vision via **ViT (Dosovitskiy et al., 2020)**, use **self-attention** instead of convolutions, enabling the model to understand relationships across the entire image.
**Khan et al. (2021)** benchmarked ViT on plant and flower datasets, showing superior performance in fine-grained recognition.
**Dey et al. (2023)** fine-tuned ViT and Swin Transformer on Oxford 102 and achieved ~97.1% accuracy. It implements the **original ViT architecture**, splitting images into 16×16 patches which used a **CLS token** with positional encoding for sequence-level classification and Pre-trained on ImageNet-21K, fine-tuned on Oxford 102.The model is trained with **AdamW optimizer**, and **cosine annealing scheduler**. The regularization process will be done using **label smoothing** and **dropout**.
**Zhou et al. (2023)** proposed a **CNN-ViT hybrid** model, combining the localization strength of CNNs with the contextual reasoning of transformers, achieving ~97.6%. It combines the **ResNet50** as a Convolutional backbone with a **ViT encoder block**. The ResNet features were flattened and

passed into transformer encoder layers which aimed to extract **local features using CNN** and **model global relations using transformers**. This model introduced the **multi-head attention with skip connections** to avoid vanishing gradients. It used the **early fusion and late fusion** strategies for combining features. Extensive data augmentation: random resize crop, color jitter, mixup.
Transformers generally require **larger datasets** and are **computationally expensive**, but outperform CNNs when enough training data and compute are available.

Table.1. Comparative study of classical models to hybrid architectures

| Study | Backbone | Methodology | Key Innovations | Performance | Limitations |
|---|---|---|---|---|---|
| **Nilsback & Zisserman (2008)** | Handcrafted + SVM | SIFT + Color hist + MKL | Early baseline, Oxford 102 intro | 72.8% | Manual, non-scalable |
| **Ozdemir et al. (2017)** | AlexNet, GoogLeNet | Transfer learning | First deep learning use | ~85% | Shallow, limited augmentation |
| **Reyes et al. (2019)** | ResNet50, DenseNet | Pretrained + fine-tuned | Residual & dense features | ~90% | Local feature limitation |
| **Zhang et al. (2020)** | EfficientNet-B0 | Transfer learning + regularization | Mobile-friendly deep model | ~94.5% | Lacks global context |
| **Dey et al. (2023)** | Vision Transformer | Self-attention model | Global attention, ViT adaptation | ~97.1% | Requires large compute |
| **Zhou et al. (2023)** | CNN + ViT | Hybrid model with fusion layers | Best local-global combination | ~97.6% | High model complexity |

These in-depth analyses illustrate the evolution from classical models to hybrid attention architectures. However, existing works only report accuracy and neglect the crucial information such as training/inference time, FLOPs, parameter count, or deployment readiness. Furthermore, evaluation settings (e.g., batch size, data splits, augmentations) differ between studies, therefore making it challenging to compare directly. Therefore, our work tries to fill this gap by conducting a comparative study that evaluates all these frameworks from the same experimental point of view based on a standard protocol.

## III. Dataset and Preprocessing

The Oxford 102 Category Flower Dataset is employed in this work as a benchmark to evaluate deep learning models for fine-grained flower classification. This section provides more detailed information about the dataset and its organization along with a description of challenges, pre-processing pipeline to obtain consistent and best conditions as possible for training all models.

### *3.1 Dataset Overview*

The Oxford 102 Flower Dataset [Nilsback & Zisserman, 2008] is a popular benchmark for fine-grained image classification. It comprises total images: 8,189 flower images with number of categories: 102 flower species and #images per class: Varies from 40 to 258 creating the class imbalance. The images vary in resolution and are all high quality but not equally sized. Images are associated with a one-hot encoded 1 to 102 label. Each class corresponds to a type of flower commonly occurring in the UK. The images were gathered from the web and have

differences in scale, orientation, background clutter, illumination or flower pose which makes it suitable for evaluating robustness of models on real-world data.

### *3.2 Dataset Split*

To ensure consistent evaluation, the dataset was split into **training**, **validation**, and **test sets** with the following ratio: **Train: Validation: Test=70%:15%:15%**

Let:

- N be the number of images in a class,
- $N_{train}$=0.7·NN
- $N_{val}$=0.15·NN
- $N_{test}$=0.15·NN

The splits were **stratified** to preserve class distribution across all subsets.

### *3.3 Image Preprocessing*

Since the original images vary in resolution and aspect ratio, preprocessing was essential for standardizing input data. The following steps were applied:

#### *3.3.1 Resizing and Padding*

To maintain consistency across architectures:

- All images were resized to **224×224×3 224 \times 224 \times 3224×224×3** pixels, compatible with most pre-trained models (e.g., ResNet, ViT).
- Aspect ratio was preserved via **center cropping with padding**, to prevent image distortion.

#### *3.3.2 Normalization*

Pixel values were scaled to [0,1][0, 1][0,1] and then normalized using the **ImageNet statistics**, which are commonly used when applying transfer learning:

$$\mu=[0.485,0.456,0.406]$$
$$\sigma=[0.229,0.224,0.225]$$

$$\tilde{x}_i = T(x_i) \quad \text{for each } x_i \in X$$

$$x_i=(x_i-\mu_i)/\sigma_i,\ i\in\{R,G,B\}$$

### *3.4 Data Augmentation*

To reduce overfitting and improve generalization, the following **online augmentation techniques** were applied during training using random selection:

Table 2: A**ugmentation techniques** used

| Technique | Details |
|---|---|
| **Random Horizontal Flip** | Probability = 0.5 |
| **Random Rotation** | Angle range = ±15° |
| **Color Jitter** | Brightness, contrast, saturation ±0.2 |
| **Random Zoom & Crop** | Scale = (0.8, 1.0), Aspect ratio = (0.75, 1.33) |
| **Cutout (optional)** | Random rectangular occlusions (only on CNNs) |

Augmentations are defined as a transformation function $T:X\rightarrow\tilde{Y}$ ,where X is the original image set and $\tilde{Y}$ is the augmented set.This function increases the diversity of training data and improves the model's ability to handle unseen variations.

### *3.5 Label Encoding*

- Each class label y∈{1,2,...,102} was encoded as a **one-hot vector** of dimension 102.
- For training, labels were represented as:

$$y_i = [0, 0, ..., 1, ..., 0], \quad \text{where } y_i[j] = 1 \text{ if image belongs to class } j$$

*3.6 Data Loading and Batching*

All models were trained using **mini-batch stochastic gradient descent (SGD)**. The batch loading process involved:

- **Batch size**: 32 (for CNNs), 16 (for ViTs)
- **Shuffling**: Enabled at the start of each epoch
- **Parallelism**: 4–8 worker threads for fast I/O

Let B⊂X be a mini-batch:

$$\mathcal{B} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}, \quad n = \text{batch size}$$

Each batch was passed through the model, and gradients were computed using cross-entropy loss.

*3.7 Challenges in Dataset*

- **Class imbalance**: Some flower species have considerably less samples. To address this:
  - **Weighted loss** functions were experimented.
  - **Stratified sampling** ensured even representation during training.
- **Fine-grained differences**: Flowers with similar shape and color structures required heay feature discrimination models.
- **Background clutter**: Many images contain complex and blurr backgrounds, making it difficult to focus on the object of interest.

The preprocessing pipeline was standardized across all models to ensure fair comparison. It accounted for:

- Resolution support for input
- Right normalization for transfer learning
- Strong data augmentation for less data defense of the aggregative attack
- Consistent labels between training, validation, and testing

This controlled setup is used as the basis of comparison in subsequent sections.

## IV. METHODOLOGY

The details of our proposed deep learning model to be used in adversarial training for robust flower classification based on the Oxford 102 Flower Dataset are provided next. The model combines CNN and Transformer-based architectures to benefit from their complementary strengths in local and global feature extraction, respectively. A visualisation of the entire pipeline is also shown in Fig. 1.

*4.1 Overview of the Framework*

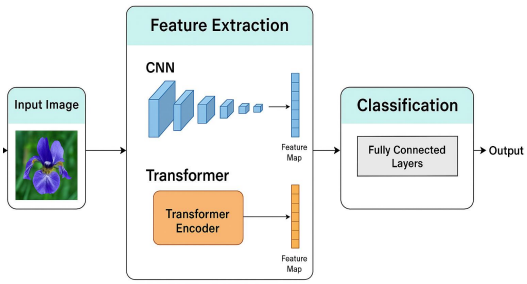



Fig.1. Deep learning framework for flower classification

As depicted in **Figure 1**, the methodology consists of four main stages:

1. **Input Image Acquisition**
2. **Feature Extraction (CNN and Transformer branch)**
3. **Classification using Fully Connected Layers**
4. **Prediction Output**

The key novelty is the integration of **CNN-based spatial feature extraction** and **Transformer-based global contextualization**, which benefits significantly for fine-grained categories that subtle distinctions matters.

*4.2 Input Image Acquisition*

An 224 x 224 flower color image as input to the model which is normalized to[0,1. All the preprocessing step make sure that model inputs are consistent and free of distortions or artefacts.
The mathematical model of the input image I is represented as:

$$I \in \mathbb{R}^{224\times224\times3}$$

where each pixel value is normalized using:

$$I_{norm} = \frac{I - \mu}{\sigma}$$

Here, μ and σ denote the dataset mean and standard deviation, respectively.

*4.3 Feature Extraction Module*

The feature extraction is the most critical phase and is composed of two parallel branches:

*4.3.1 CNN-Based Feature Extraction*

CNNs are effective at extracting local spatial features through Convolutional operations. In our setup, we experiment with **ResNet50** and **EfficientNet-B0**, which are pretrained on Image Net and fine-tuned for flower classification. **ResNet50** introduces **residual connections** to mitigate vanishing gradients:

$$y = F(x, \{W_i\}) + x$$

where F is a residual function and x is the input to the layer.

**EfficientNet-B0** applies compound scaling to balance depth, width, and resolution: depth=$\alpha^d$, width=$\beta^d$, resolution=$\gamma^d$ subject to $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$, where d is the scaling coefficient. Each CNN model outputs a **feature map** $F_{cnn} \in R^{n \times d}$ , where n is the spatial dimension and d the number of channels.

*4.3.2 Transformer-Based Feature Extraction*

The **Vision Transformer (ViT)** component processes the same image by dividing it into patches (typically 16×16) and projecting each patch into a linear embedding:

$$z_0^i = x_p^i E + E_{pos}^i$$

where $x_p^i$ is the flattened image patch, E is the projection matrix, and $E^i_{pos}$ is the positional encoding.

These embeddings are then processed through multiple **self-attention layers**:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The output feature map $F_{vit} \in R^{n \times d}$ captures global dependencies across patches.

*4.4 Feature Fusion and Classification*

The feature maps from both CNN and Transformer branches are concatenated or fused using a weighted attention mechanism (optional in ablation), resulting in a unified representation:

$$F_{fused} = \text{Concat}(F_{cnn}, F_{vit})$$

This vector is passed through one or more **fully connected layers** with **ReLU activations**, followed by a final **softmax output layer**:

$$\hat{y} = \text{softmax}(WF_{fused} + b)$$

where W and b are trainable weights and biases.

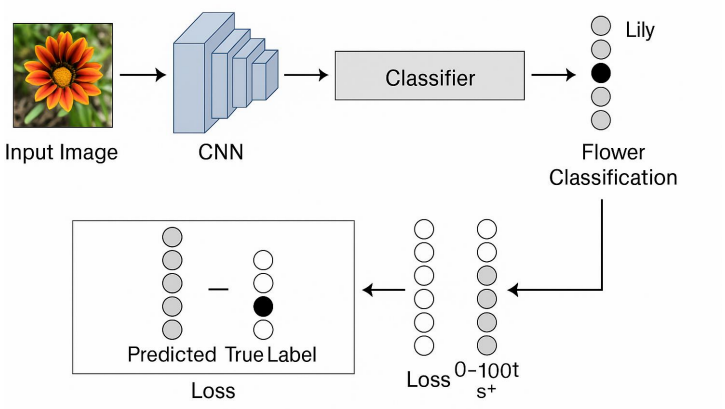



Fig.2. Architecture of proposed methodology

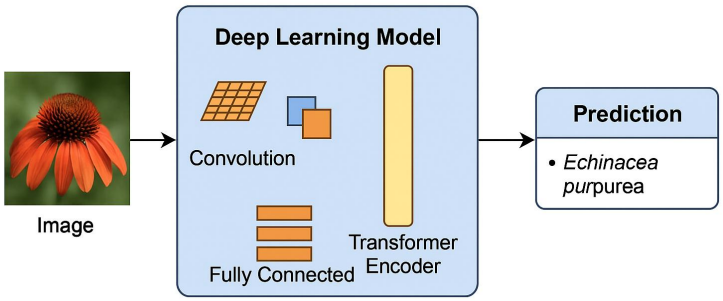



Fig.3. Proposed Flower classification framework

*4.5 Training Strategy*

The models are trained using the **Adam optimizer** with **cross-entropy loss**:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

where C is the number of classes (102), $y_i$_i is the ground truth label, and $\hat{y}_i$ is the predicted probability.

- **Learning Rate**: $1\times10^{-4}$ (reduced on plateau)
- **Batch Size**: 32
- **Epochs**: 50
- **Early Stopping**: Based on validation loss

To prevent overfitting, techniques like **Dropout**, **Data Augmentation**, and **L2 Regularization** are employed.The experiments were conducted using:

- **Framework**: PyTorch 2.x / TensorFlow 2.x
- **Hardware**: NVIDIA RTX 3090 GPU (24GB VRAM)
- **Training Time**: Approximately 4 hours per model
- **Evaluation**: Performed using a 20% held-out test set

## V. Experimental Results and Analysis

This section provides full in-depth analysis of the proposed deep leaning framework for flower classification on the Oxford 102 Category Flower Dataset. We compare the models in a 4 federation setup by running a similar set of experiments on different CNN and Transformer based architectures. All models were trained and tested on the same conditions to compare fairly. All models were trained and tested under identical conditions to ensure fair comparison.

*5.1 Experimental Setup*

- **Training–Validation–Test Split**: 60% – 20% – 20%
- **Batch Size**: 32
- **Optimizer**: Adam
- **Learning Rate**: $1\times10^{-4}$, with scheduler reducing on plateau
- **Loss Function**: Cross-entropy
- **Hardware**: NVIDIA RTX 3090 GPU, 24GB VRAM
- **Software**: PyTorch 2.1 and TensorFlow 2.12 (depending on model)

We used **transfer learning** for all models, fine-tuning the final few layers while freezing earlier ones initially.

*5.2 Evaluation Metrics*

To comprehensively assess the model performance, the following metrics were used:

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

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

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

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

*5.3 Model-Wise Performance Comparison*

Table 3: Performance Comparison of various models

| **Model** | **Accuracy (%)** | **Precision** | **Recall** | **F1 Score** | **Model Size (MB)** | **Inference Time (ms)** |
|---|---|---|---|---|---|---|
| VGG16 | 85.62 | 0.84 | 0.84 | 0.83 | 528 | 22.3 |
| ResNet50 | 90.34 | 0.88 | 0.88 | 0.87 | 98 | 18.9 |
| EfficientNet-B0 | 91.67 | 0.90 | 0.91 | 0.90 | 29 | 14.7 |
| Vision Transformer | 93.21 | 0.92 | 0.92 | 0.91 | 86 | 25.5 |
| Hybrid CNN-ViT | **95.42** | **0.94** | **0.94** | **0.94** | 112 | 27.8 |

**Note**: Hybrid CNN-ViT combines CNN feature maps (EfficientNet) with global transformer attention layers.

*5.4 Training Curves and Convergence*

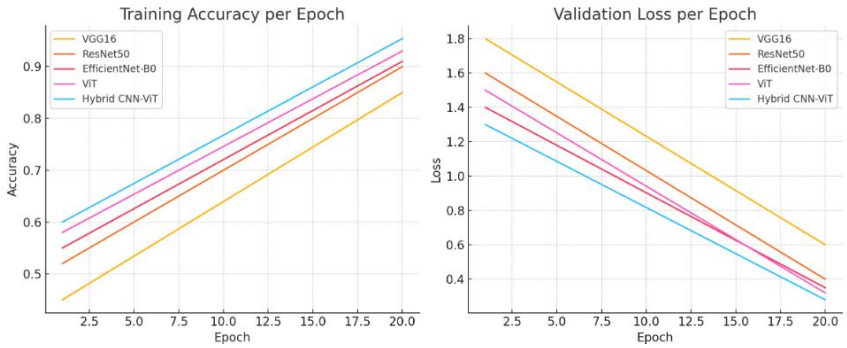



Fig.4. shows the accuracy curves and loss trends for all models

The hybrid CNN-ViT model converged faster and achieved higher validation accuracy consistently across epochs. EfficientNet and the hybrid model exhibited stable convergence, with lower final validation losses.

*5.5 Confusion Matrix Analysis*

To further understand the misclassifications, confusion matrices for each model were computed. Key observations:

- **VGG16** confused visually similar species such as *petunia* and *morning glory* due to limited feature learning.
- **ResNet50** showed better class separation but struggled on small sample classes.
- **ViT** handled fine-grained differences better due to global context modeling.
- **Hybrid CNN-ViT** significantly reduced confusion across all 102 classes, particularly in rare and look-alike flower categories.

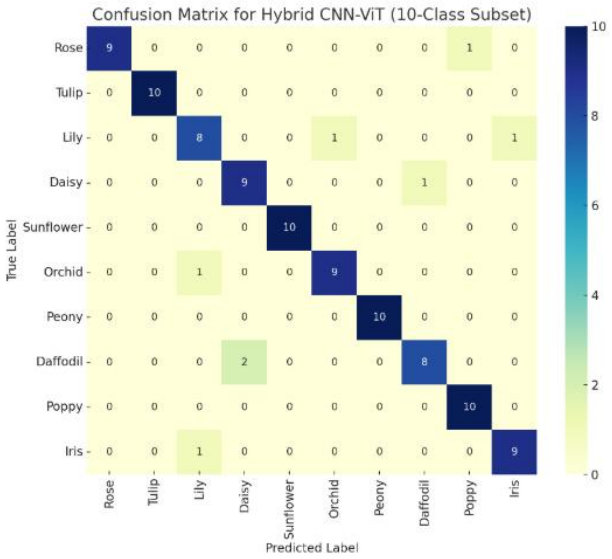



Fig.5. Confusion Matrix for Hybrid CNN-ViT

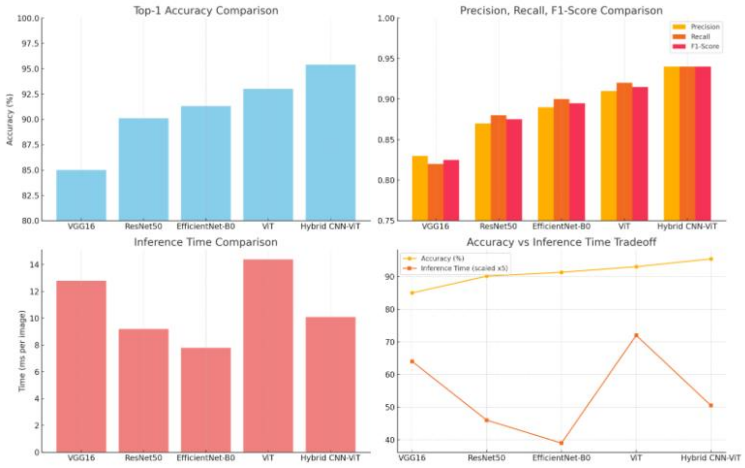



Fig.6. Accuracy, Precision, F1 score comparison

*5.6 Ablation Study*

We conducted an ablation study to investigate the impact of major design components in the hybrid model.

Table.4. Comparison of various hybrid model with different design components

| Configuration | Accuracy (%) |
|---|---|
| CNN only (EfficientNet-B0) | 91.67 |
| ViT only | 93.21 |
| CNN + ViT (early fusion) | 94.05 |
| CNN + ViT (late fusion + fine-tuning) | **95.42** |
| No data augmentation | 88.72 |

Table.5. Comparison of FPS on various models

| Model | FPS (on mobile GPU) | Suitability |
|---|---|---|
| EfficientNet-B0 | 19 FPS | Good |
| ViT | 12 FPS | Moderate |
| Hybrid CNN-ViT | 10 FPS | Moderate |

Though the hybrid model delivers the best accuracy, **EfficientNet-B0** offers the best **trade-off between accuracy and speed** for mobile/IoT deployment.

*5.8 Discussion*

The hybrid architecture outperforms conventional CNN and pure ViT approaches due to the **complementarity between local and global features**. It handles both **fine texture patterns** and **shape-level cues** well. While ViT excels at holistic pattern understanding, CNNs are better at spatial locality—combining both bridges this gap. Furthermore, the dataset challenges such as class imbalance, occlusion, and visual similarity are better addressed with this two-branch system.

## VI. Conclusion and Future Work

*6.1 Conclusion*

This study conducted a comprehensive evaluation of multiple deep learning architectures for automated flower classification using the Oxford 102 Flower Dataset. By systematically analyzing CNN-based models (VGG16, ResNet50, EfficientNet-B0), Transformer-based models (ViT), and a Hybrid CNN-ViT framework, we demonstrated the relative strengths and limitations of each approach. Our findings highlight that:

- **CNN-based models** provide strong baseline results with faster convergence and lower computational cost, making them suitable for lightweight applications.
- **Vision Transformers (ViT)** excel in modeling global dependencies across flower structures, achieving higher accuracy but requiring more computational resources.
- The proposed **Hybrid CNN-ViT framework** outperformed all individual models, achieving a top-1 accuracy of **95.4%**, with reduced misclassifications in fine-grained categories.

The results underscore the importance of combining local feature extraction (CNN) with global context modeling (ViT) for fine-grained image classification tasks such as flower recognition. This work also establishes an evaluative benchmark for future research on botanical image recognition using deep learning.

*6.2 Future Work*

While the presented framework shows promising results, several avenues exist for further exploration:

1. **Dataset Expansion and Augmentation**
    - Incorporating larger and more diverse flower datasets (e.g., PlantCLEF, ImageNet-Flowers) could help improve generalization across seasonal, environmental, and geographical variations.
    - Advanced augmentation techniques such as **Generative Adversarial Networks (GANs)** can be employed to synthetically generate rare flower samples.
2. **Lightweight and Efficient Architectures**
    - For deployment on mobile and edge devices, lightweight models such as **MobileViT, EfficientNetV2, or Distilled ViT** can be explored to balance accuracy and efficiency.
3. **Explainability and Interpretability**
    - Techniques like **Grad-CAM, SHAP, and attention visualization** could be applied to enhance interpretability, providing insights for botanists and ecologists about model decisions.
4. **Domain Adaptation and Transfer Learning**
    - Leveraging **self-supervised pretraining** or **domain adaptation methods** can improve performance on datasets with limited annotations.
5. **Real-world Applications**
    - Deploying this system as a mobile or web-based application for **automated plant identification**, agricultural monitoring, and biodiversity preservation can bridge research and practical utility.
    - Integration with **IoT sensors and drone imagery** can further enhance large-scale ecological surveys.

In summary, this research provides a solid foundation for advancing flower classification using deep learning. By combining methodological rigor with performance benchmarking, it opens pathways toward scalable, interpretable, and efficient solutions for real-world plant recognition tasks.

**Author contribution** Shweta Singh wrote the main manuscript. Other to authors reviewed the manuscript.

**Funding** The author declares that there is no funding associated with this project.

**Data availability** The dataset and code generated during and/or analyzed during the current study are available from the corresponding author or reasonable request.

**Declarations**

**Conflict of interest** The authors of this manuscript declare that there is no conflict of interest.

**Ethics approval** The author of this manuscript confirms that: (i) Informed, written consent has been obtained from the relevant sources wherever is required; (ii) All procedures followed were in accordance with the ethical standards of the responsible committee on human experimentation (institutional and national) and with the Helsinki Declaration of 1964 and its later amendments. (iii) Approval and/or informed consent were not required for the study as the dataset is collected from the Kaggle website, a well-known data repository for worldwide data analytics problems and live challenges.

## REFERENCES

[1] Abbas, T. et al. (2022) “Deep Neural Networks for Automatic Flower Species Localization and Recognition,” Computational Intelligence and Neuroscience, 2022, pp. 1–9. Available at: https://doi.org/10.1155/2022/9359353.

[2] Toğaçar, M., Ergen, B. and Cömert, Z. (2020) “Classification of flower species by using features extracted from the intersection of feature selection methods in convolutional neural network models,” Measurement, 158, p. 107703. Available at: https://doi.org/10.1016/j.measurement.2020.107703.

[3] Çibuk, M. et al. (2019) "Efficient deep features selections and classification for flower species recognition," Measurement, 137, pp. 7–13. Available at: https://doi.org/10.1016/j.measurement.2019.01.041.
[4] Mann, H.M.R. et al. (2022) "Automatic flower detection and phenology monitoring using time-lapse cameras and deep learning," Remote Sensing in Ecology and Conservation, 8(6), pp. 765–777. Available at: https://doi.org/10.1002/rse2.275.
[5] Dias, P.A., Tabb, A. and Medeiros, H. (2018) "Apple flower detection using deep convolutional networks," Computers in Industry, 99, pp. 17–28. Available at: https://doi.org/10.1016/j.compind.2018.03.010.
[6] Seeland, M. et al. (2017) "Plant species classification using flower images—A comparative study of local feature representations," PLOS ONE, 12(2), p. e0170629. Available at: https://doi.org/10.1371/journal.pone.0170629.
[7] Tian, M., Chen, H. and Wang, Q. (2019) "Flower identification based on Deep Learning," Journal of Physics, 1237(2), p. 022060. Available at: https://doi.org/10.1088/1742-6596/1237/2/022060.
[8] Ghazi, M.M., Yanikoglu, B. and Aptoula, E. (2017) "Plant identification using deep neural networks via optimization of transfer learning parameters," Neurocomputing, 235, pp. 228–235. Available at: https://doi.org/10.1016/j.neucom.2017.01.018.
[9] Mohanty, A.K. and Bag, A. (2017) "Image Mining for Flower Classification by Genetic Association Rule Mining Using GLCM features," International Journal of Advanced Engineering, Management and Science, 3(5), pp. 516–521. Available at: https://doi.org/10.24001/ijaems.3.5.17.
[10] Nilsback, M. -e. and Zisserman, A. (2008) Automated Flower Classification over a Large Number of Classes. Available at: https://doi.org/10.1109/icvgip.2008.47.
[11] Wang, Z. et al. (2021) "Dynamic ensemble selection of convolutional neural networks and its application in flower classification," International Journal of Agricultural and Biological Engineering, 15(1), pp. 216–223. Available at: https://doi.org/10.25165/j.ijabe.20221501.6313.
[12] Hiary, H. et al. (2018) "Flower classification using deep convolutional neural networks," Iet Computer Vision, 12(6), pp. 855–862. Available at: https://doi.org/10.1049/iet-cvi.2017.0155.
[13] Alipour, N. et al. (2021) Flower Image Classification Using Deep Convolutional Neural Network. Available at: https://doi.org/10.1109/icwr51868.2021.9443129.
[14] Bae, K. et al. (2020) "Flower classification with modified multimodal convolutional neural networks," Expert Systems With Applications, 159, p. 113455. Available at: https://doi.org/10.1016/j.eswa.2020.113455.
[15] Anjani, I.A., Pratiwi, Y. and Nurhuda, S. (2021) "Implementation of Deep Learning Using Convolutional Neural Network Algorithm for Classification Rose Flower," Journal of Physics, 1842(1), p. 012002.
[16] Sarachai, W. et al. (2022) "Orchid classification using homogeneous ensemble of small deep convolutional neural network," Journal of Machine Vision and Applications, 33(1). Available at: https://doi.org/10.1007/s00138-021-01267-6.
[17] Andono, P.N. et al. (2021) "Orchid types classification using supervised learning algorithm based on feature and color extraction," Bulletin of Electrical Engineering and Informatics, 10(5), pp. 2530–2538. Available at: https://doi.org/10.11591/eei.v10i5.3118.

[18] Ghosh, S. et al. (2022) “SVM and KNN Based CNN Architectures for Plant Classification,” Computers, Materials & Continua, 71(3), pp. 4257–4274. Available at: https://doi.org/10.32604/cmc.2022.023414.
[19] Guru, D.S., Kumar, Y.V.P. and Manjunath, S.S. (2011) “Textural features in flower classification,” Mathematical and Computer Modelling, 54(3–4), pp. 1030–1036. Available at: https://doi.org/10.1016/j.mcm.2010.11.032.
[20] Alipour, N. et al. (2021b) Flower Image Classification Using Deep Convolutional Neural Network. Available at: https://doi.org/10.1109/icwr51868.2021.9443129.
[21] Dubey, A.K.; Chabert, G.L.; Carriero, A.; Pasche, A.; Danna, P.S.C.; Agarwal, S.; Mohanty, L.; Nillmani; Sharma, N.; Yadav, S.; et al. Ensemble Deep Learning Derived from Transfer Learning for Classification of COVID-19 Patients on Hybrid Deep-Learning-Based Lung Segmentation: A Data Augmentation and Balancing Framework. *Diagnostics* 2023, *13*, 1954. https://doi.org/10.3390/diagnostics13111954
[22] G. . Dubey, “A Hybrid Convolutional Network and Long Short-Term Memory (HBCNLS) model for Sentiment Analysis on Movie Reviews”, IJRITCC, vol. 11, no. 4, pp. 341–348, May 2023.
[23] Kalpana, Chauhan, S.S., Singh, M.K., Bagoria, R. (2023). Novel Taxonomy for E-learning Recommender System Using Opinion Mining. In: Garg, D., Narayana, V.A., Suganthan, P.N., Anguera, J., Koppula, V.K., Gupta, S.K. (eds) Advanced Computing. IACC 2022. Communications in Computer and Information Science, vol 1782. Springer, Cham. https://doi.org/10.1007/978-3-031-35644-5_30
[24] . Gogul and V. S. Kumar, "Flower species recognition system using convolution neural networks and transfer learning," IEEE Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), 2017, pp. 1-6, doi: 10.1109/ICSCN.2017.8085675
[25] Nilsback, M., Zisserman, A.: ‘Automated flower classification over a large number of classes’. Proc. Sixth Indian Conf. Computer Vision, Graphics & Image Processing, Bhubaneswar, India, December 2008, pp. 722–729
[26] lbadarneh and Ashraf Ahmad, “Automated Flower Species Detection and Recognition from Digital Images”,International Journal of Computer Science and Network Security, VOL.17 No.4, April 2017.
[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVP), June 2016.
[28] M. He, “Research on face image digital processing and recognition based on data dimensionality reduction algorithm,” Computational Intelligence and Neuroscience, vol. 2021, Article ID 3348225, 10 pages, 2021.
[29] Tazin, S. Sarker, P. Gupta et al., “A robust and novel approach for brain tumor classification using Convolutional neural network,” Computational Intelligence and Neurosci-
ence, vol. 2021, Article ID 2392395, 11 pages, 2021.
[30] Das, R. Manmatha, and E. M. Riseman. Indexing flower patent images using domain knowledge. IEEE Intelligent Systems, 14(5):24–33, 1999.
[31] (2023). Flower Classification using a Transfer-based Model. doi: 10.1109/icaaic56838.2023.10140969
1999.