TASK 2

FLOWER IMAGE CLASSIFICATION USING TRANSFER LEARNING

ARNAV GUPTA – RA2211028010107

SPANDAN BASU CHAUDHURI – RA2211028010120

AIM

The aim of this project is to build an efficient flower image classification model capable of identifying different types of flowers using transfer learning based on the InceptionV3 architecture.

OBJECTIVE

1. To download and prepare the flower image dataset for training and validation.
2. To apply data augmentation techniques to improve model generalization.
3. To implement a Transfer Learning approach using the InceptionV3 pretrained model.
4. To train and evaluate the model to classify flowers into five categories.
5. To visually analyze training results through accuracy and loss plots.
6. To save the best performing model and use it for real-time prediction.

INTRODUCTION

Image classification is a key task in computer vision that focuses on assigning labels to images. Deep learning models, particularly Convolutional Neural Networks (CNNs), are widely used for this purpose. Training CNNs from scratch typically requires large datasets and high computational power. To overcome this, **Transfer Learning** is used, where a pretrained model (trained on large datasets like ImageNet) is adapted for a specific task.

In this project, the **InceptionV3** model is used as a feature extractor, and new fully connected layers are added for flower classification. This approach helps achieve higher accuracy and faster training.

The dataset used contains images of 5 types of flowers:

- Daisy
- Dandelion
- Rose
- Sunflower
- Tulip

REQUIREMENTS

Hardware Requirements

- System with 8GB RAM or more
- GPU recommended (e.g., NVIDIA CUDA GPU)

Software Requirements

Software/Library	Version	Purpose
Python	3.8+	Programming environment
TensorFlow / Keras	2.x	Deep learning model implementation
Matplotlib	Latest	Plotting accuracy and loss graph
Pandas and NumPy	Latest	Data handling
KaggleHub	Latest	Dataset retrieval

Dataset

Kaggle Dataset: **imsparsh/flowers-dataset**

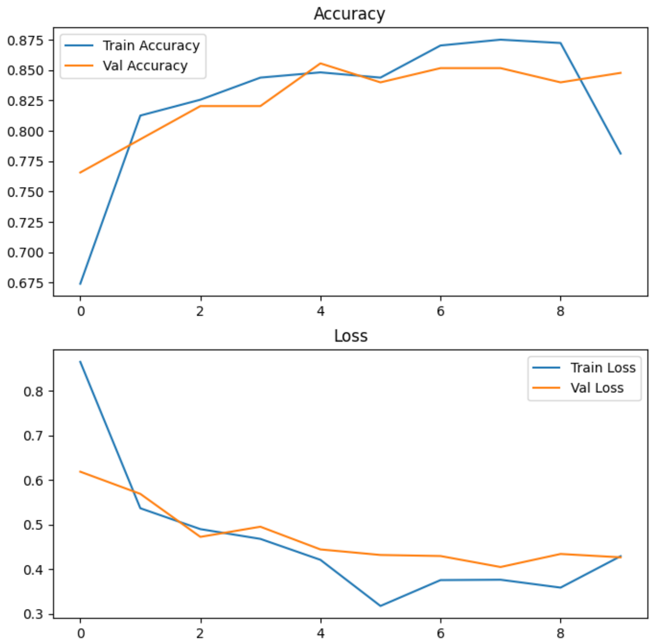
LIBRARIES USED

Library	Purpose
TensorFlow / Keras	Load pretrained InceptionV3 and build model
KaggleHub	Download dataset from Kaggle
ImageDataGenerator	Data augmentation and preprocessing
Matplotlib	Visualization of results

NumPy, Pandas	Numerical and data handling operations

ALGORITHM / METHODOLOGY

1. **Dataset Download**
    - Dataset is downloaded using kagglehub.dataset_download().
2. **Data Preprocessing & Augmentation**
    - Images are resized to **300 × 300**.
    - Pixel values are normalized.
    - Data augmentation applied: rotation, shifting, zooming, flipping.
3. **Model Selection (Transfer Learning)**
    - InceptionV3 pretrained on ImageNet is loaded without top layers.
    - Pretrained layers are frozen to retain learned features.
4. **Model Construction**
    - Added layers:
        - GlobalAveragePooling
        - Fully connected Dense layer with ReLU
        - Dropout to prevent overfitting
        - Output Dense layer with Softmax activation
5. **Model Training**
    - Optimizer: Adam (learning_rate = 0.0001)
    - Loss: Categorical Crossentropy
    - Training for **10 epochs** with validation.
6. **Model Saving**
    - ModelCheckpoint used to save the best model based on validation accuracy.
7. **Performance Visualization**
    - Accuracy and Loss curves plotted to evaluate training behavior.
8. **Testing / Prediction**
    - Model used to classify sample flower images.



INPUT / OUTPUT

Input	**Output**
Flower image (JPEG/PNG)	Predicted flower class + Confidence percentage

Example Output:

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
Predicted: rose | Confidence: 97.3 %
```

RESULTS

- The model successfully learned distinguishing features of flower categories.
- Validation accuracy improved steadily across epochs.
- Data augmentation helped reduce overfitting.
- The final trained model demonstrated **high prediction confidence** during testing.

Training Trend Observations:

- Training and validation accuracy increased progressively.
- Loss curve showed decreasing error, indicating successful model learning.

CONCLUSION

This project successfully demonstrates the application of **Transfer Learning using InceptionV3** for multi-class flower image classification. By leveraging pretrained features, the model achieved high classification performance with reduced training time and computational cost. The final model can be used for:

- Botanical identification applications
- Educational tools
- Automated plant nursery classification systems
- Mobile or web-based flower recognition apps

Future improvements may involve:

- Fine-tuning deeper layers of InceptionV3
- Using larger datasets
- Deploying the model using TensorFlow Lite for mobile devices