

# Project Summary: Transfer Learning for Flower Classification using TensorFlow

## Customizations Introduced

### 1. Custom Optimizer – MySGDOptimizer

- A user-defined variant of **Stochastic Gradient Descent (SGD)**.
- Simplified structure: manually updates weights using `var.assign_sub(lr * grad)`.
- Offers fine control over training and integrates seamlessly into Keras API via `@register_keras_serializable`.

### 2. Custom Loss Function – Focal Loss

- Tackles **class imbalance** by dynamically scaling loss for difficult examples.
- Controlled by:
  - $\gamma$  (gamma) – focuses learning on hard examples.
  - $\alpha$  (alpha) – balances classes.
- Especially useful in cases where **some flower categories dominate**.

### 3. Advanced Activation – Swish

- Defined as  $\text{swish}(x) = x * \text{sigmoid}(x)$ .
- Non-monotonic, smooth, and retains small negative values (unlike ReLU).
- Used in deeper layers to **enhance learning of complex floral patterns**.
- **ReLU** used later to add **sparsity and performance** in final stages.

## Model Architecture

Based on **MobileNetV2** (pre-trained on ImageNet) with frozen base layers for feature extraction.

Custom classification head:

- **GlobalAveragePooling2D**
- Dense layers:  $[256 \rightarrow 128 \rightarrow 64]$
- Activations: Swish, Swish, ReLU
- BatchNormalization and Dropout after each dense layer.
- Final layer uses Softmax for multi-class classification (`num_classes = 5`).

## **Training Procedure**

### **Phase 1 – Feature Extraction**

- Base model is frozen.
- Only custom head is trained.
- Optimizer: MySGDOptimizer with LR = 0.01
- Loss: Focal Loss
- Trained for 5 epochs.

### **Phase 2 – Fine-tuning**

- **Unfreezes the last 30 layers** of the base model.
- Fine-tunes with a smaller LR =  $1e-5$  to **retain general features** while learning flower-specific patterns.
- Same optimizer and loss used.
- Trained for 5 additional epochs.

## **Inference System**

- Allows **image upload by file path** for prediction.
- Displays:
  - Predicted flower class
  - Confidence score
  - Original image
  - Bar chart of class-wise probabilities
- **Confidence threshold (70%)** is used to filter out uncertain predictions.

## **Evaluation**

Model evaluated on validation set (20% of dataset).

Metrics:

- Batch-wise accuracy plot
- Per-class accuracy bar chart
- Overall accuracy and class-wise report printed.

## Output

Please enter the image path:  
dandelion.jpg  
1/1 0s 138ms/step

Predicted Flower: dandelion (96.47%)

Prediction: dandelion (96.47%)



Class Probabilities:  
dandelion: 96.47  
daisy: 0.10  
tulips: 0.70  
sunflowers: 0.53  
roses: 2.19

Overall Accuracy: 96.47%

## Visualizations

