Project Task 2

Deep Learning (CS F425)

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

November 2024

Team Members

Akshay Shukla	2022A7PS0087P
Ishaan Kale	2022A7PS0084P
Siddhartha Gotur	2022A7PS0070P

Table of Contents

Description of Dataset

The dataset given to us consists of images of flowers belonging to 60 different classes. The train dataset contains 3000 images with 50 images belonging to each of the classes. There is also a separate text file for each of the train and validation datasets which contains the labels corresponding to each of the images. The images are 3 channel images (RGB) and have a dimension of 256x256. The validation dataset has 600 images with 10 images for each class.

Model Architecture

We used the pretrained VGG19 model for this task. The fully connected layers were removed and we added our own Flatten and two Dense Layers (256 and 60) with leaky ReLU and sigmoid activations respectively.

Initially the VGG base model weights were frozen and we trained the FC layers on the task. Then we unfroze the top 4 layers of VGG for finetuning.

The **VGG architecture** is known for its simplicity and uniformity, using small 3×3 convolutional kernels throughout the network, which allows it to capture spatial hierarchies effectively. VGG stacks multiple convolutional layers with ReLU activations, followed by max-pooling layers for down-sampling. The depth of the network increases significantly, enabling it to learn rich features. Finally, fully connected layers are used for classification tasks. Although computationally heavy due to a large number of parameters, VGG is a highly influential model and serves as a strong baseline for transfer learning.

Hyperparameters used for Training

Loss Function - Cross Entropy

Optimizer - Adam

Batch Size - 32

Here is a description of some of the methods we used -

Cross-entropy is a loss function commonly used in classification tasks that measures the dissimilarity between the predicted probability distribution and the true distribution of labels. It quantifies how well the predicted probabilities match the actual class labels, with lower values indicating better model performance.

Adam (Adaptive Moment Estimation) is an optimization algorithm that combines the benefits of two other methods: AdaGrad and RMSProp, by maintaining separate learning rates for each parameter and using both first and second moments of the gradients. It adapts the learning rate for each parameter based on the estimates of first and second moments, enabling faster convergence and improved performance on various types of data.

Final Result after Training

We tested out this methodology on multiple models and settled on VGG as it gave us the best accuracy on the validation data.

Model Name	Validation Accuracy
InceptionV3	83.33%
MobileNetV2	87.17%
ResNet152	52.1%
Xception	81.67%
VGG19	90.5 %