|  |  |
| --- | --- |
| **Ex No: 4**  **Date: 18th September 2024** | **Transfer Learning in Image Classification** |

**Objective:** The objective is to utilize transfer learning for classifying images of flowers. Specifically, the task involves using a pre-trained model from TensorFlow Hub, such as MobileNetV2, and re-training it on a flower dataset. The objective is to save time and computational resources by leveraging pre-trained models for a new classification problem, which focuses on five classes of flowers.

**Description:**

Transfer learning is a technique in machine learning where a model trained for one task serves as the foundation for a model tackling a different, yet related task. This approach leverages knowledge gained from large datasets to solve new problems with fewer data and resources. In image classification, models like MobileNetV2, which have been trained on extensive datasets like ImageNet, are skilled at extracting relevant features from images. By using these pre-trained models as feature extractors, we can fine-tune them for specific tasks, such as classifying flower images. In this assignment, transfer learning is applied by importing MobileNetV2 from TensorFlow Hub, removing its top layer, and adding custom layers to train it on a smaller flower dataset, allowing efficient training with high accuracy while minimizing computation.

**Model Summary:**

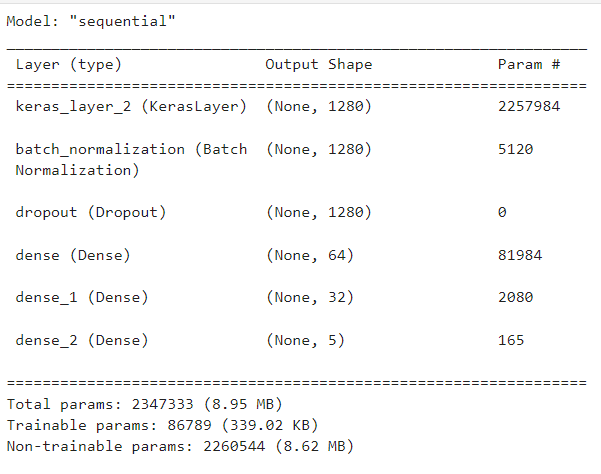


Figure1: A sequential CNN model used in Transfer Learning.

**Building the parts of the algorithm**

Here are the steps involved in building each part of the algorithm:

1. **Imports and Installation**:

* Libraries such as TensorFlow, TensorFlow Hub, Keras, OpenCV, and Matplotlib are used for model construction, training, and visualization.
* Dependencies are installed using pip, including TensorFlow Hub and OpenCV.

1. **Pre-trained Model Selection**:

* A pre-trained **MobileNetV2 model** is downloaded from TensorFlow Hub using hub.KerasLayer. This model is designed for transfer learning and used as a feature extractor.
* The model's top (classification) layer is removed, and custom dense layers are added to adapt the model for the flower classification task.

1. **Model Construction**:

* The architecture includes the pre-trained feature extractor, followed by custom layers: batch normalization, dropout, and dense layers for classification. The final layer has five output neurons, one for each flower class.

1. **Model Compilation and Training**:

* The model is compiled using the Adam optimizer and Sparse Categorical Crossentropy as the loss function (since this is a multi-class classification problem).
* It is trained on the flower dataset with 10 epochs, using a batch size of 64 and a validation split of 20%.

1. **Evaluation and Prediction**:

* The model is evaluated on test data.
* A prediction function is implemented that loads an image, preprocesses it, and predicts the flower class using the trained model.

**Conclusion:**

The conclusion of this assignment’s transfer learning approach is that utilizing pre-trained models like MobileNetV2 drastically cuts down the computational cost and training time needed for specific tasks, such as flower classification. By starting with a model trained on a large dataset like ImageNet, it can extract broad features, which are then fine-tuned to perform well on a new dataset with minimal additional training. This showcases how effective transfer learning is for efficiently managing small datasets and complex tasks, making it a powerful technique for image classification and other machine learning applications.

**GitHub Link:** [**https://github.com/jihanrjbtech22/DeepLearningLab/blob/main/Lab4\_Asgt\_Jihan.ipynb**](https://github.com/jihanrjbtech22/DeepLearningLab/blob/main/Lab4_Asgt_Jihan.ipynb)