|  |  |
| --- | --- |
| **Ex No: 6**  **Date: 18th September 2024** | **Transfer Learning in Image Classification** |

**Objective:** The goal is to apply transfer learning to classify images of flowers by using a pre-trained model from TensorFlow Hub, like MobileNetV2, and fine-tuning it on a flower dataset. This approach aims to save time and computational resources by utilizing an existing model for a new classification task focused on five flower categories.

**Description:**

Transfer learning is a machine learning approach where a model developed for one task serves as the foundation for a model on a related task, allowing it to leverage knowledge from large datasets to solve new problems with fewer data and resources. In image classification, models like MobileNetV2, trained on extensive datasets like ImageNet, learn to extract useful features from images. By using these pre-trained models as feature extractors, they can be fine-tuned 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 on a smaller flower dataset, achieving efficient training and high accuracy with reduced computational demands.

**Model Summary:**

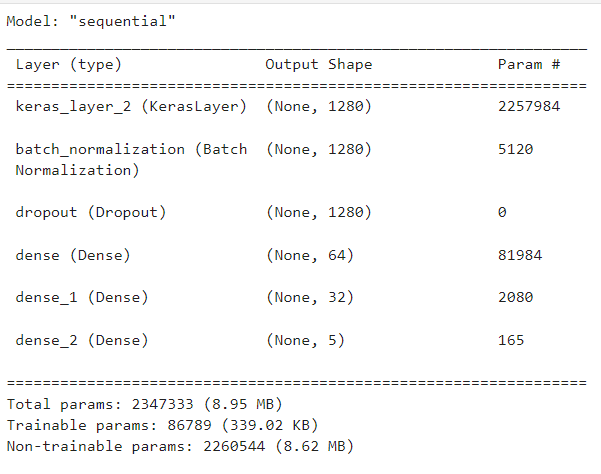


Figure1: A sequential CNN model used in Transfer Learning.

**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:**

In conclusion, this transfer learning approach shows that using pre-trained models like MobileNetV2 can greatly reduce the computational cost and time needed to train a model for specific tasks, such as flower classification. By starting with a model already trained on a large dataset like ImageNet, the model can leverage general feature extraction, which is then fine-tuned to perform effectively on the new dataset with minimal extra training. This demonstrates transfer learning’s effectiveness in efficiently handling small datasets and complex tasks, making it a valuable technique for image classification and broader machine learning applications.

**GitHub Link:** [**https://github.com/SyedHashirA/deeplearning**](https://github.com/SyedHashirA/deeplearning)