|  |  |
| --- | --- |
| **Ex No: 6**  **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 machine learning technique where a model developed for a particular task is reused as the starting point for a model on a second, related task. It leverages the pre-learned knowledge from large datasets to solve new problems with fewer data and resources. In the context of image classification, models like MobileNetV2, trained on vast datasets such as ImageNet, have learned to extract meaningful 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 case. The assignment utilizes transfer learning by taking MobileNetV2 from TensorFlow Hub, removing its top layer, and adding new layers to train it on a smaller flower dataset, enabling efficient training while achieving high accuracy with reduced computation.

**Model Summary:**

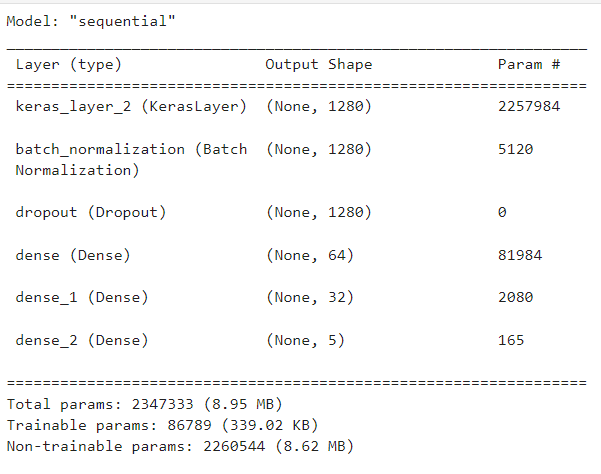


Figure1: A sequential CNN model used in Transfer Learning.

**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 the transfer learning approach in this assignment is that leveraging pre-trained models like MobileNetV2 significantly reduces the computational cost and time required to train a model for a specific task, such as flower classification. By using a model already trained on a large dataset (ImageNet), the model is able to extract general features, which are then fine-tuned to perform well on the new dataset with minimal additional training. This approach demonstrates the effectiveness of transfer learning in handling small datasets and complex tasks efficiently, making it a powerful tool in image classification and other machine learning applications.

**GitHub Link:** [**https://github.com/tulasigr/DeepLearning**](https://github.com/tulasigr/DeepLearning)