|  |  |
| --- | --- |
| **Ex No: 5**  **Date: 04-092024** | **Image Classification with MobileNetV2** |

**Objective:**

To classify an input image using a pre-trained MobileNetV2 model from TensorFlow Hub and evaluate the model's prediction accuracy.

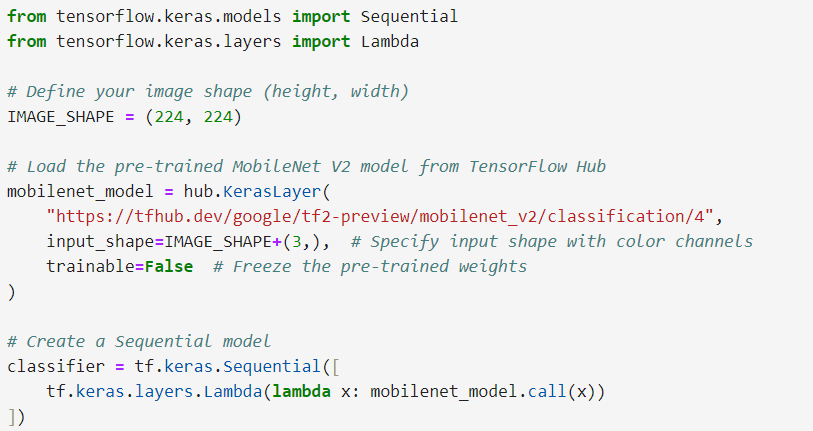
**Descriptions:**

In this experiment, we utilized a pre-trained MobileNetV2 model from TensorFlow Hub to classify an image. The MobileNetV2 model is a lightweight deep learning model designed for efficient image classification tasks, particularly on devices with limited computational resources.

The image used in this experiment was resized and normalized before being passed through the model for classification. The model's output, a vector of logits, was then decoded to identify the predicted class label. Finally, the accuracy of the prediction was assessed by comparing the predicted label to the known label.

**Model:**

* **Model Used**: Pre-trained MobileNetV2 from TensorFlow Hub
* **Input Shape**: (224, 224, 3)
* **Frozen Layers**: All layers (model is used without additional training)



|  |  |
| --- | --- |
| **Ex No: 5**  **Date: 04-092024** | **Custom Flower Classification Using Transfer Learning** |

**Objective:**

To create a custom classifier for flower images using transfer learning, leveraging a pre-trained MobileNetV2 model, and retraining the top layer to recognize specific flower classes from a custom dataset.

**Description:**

In this experiment, we applied transfer learning to classify images of flowers into five distinct categories: daisy, dandelion, roses, sunflowers, and tulips. We used a pre-trained MobileNetV2 model as a feature extractor and added a custom dense layer to adapt the model for our specific classification task.

The TensorFlow flowers dataset was used as the training and testing dataset. The dataset consists of images categorized into the aforementioned flower types. The MobileNetV2 model, pre-trained on the ImageNet dataset, was fine-tuned to recognize these specific flower categories by retraining the top layer.

**Data Preprocessing**

* Images were resized to (224, 224) pixels.
* Pixel values were normalized to the [0, 1] range.
* The dataset was split into training and testing sets.

**Model**

* **Model Used**: Pre-trained MobileNetV2 from TensorFlow Hub (used as a feature extractor)
* **Input Shape**: (224, 224, 3)
* **Custom Output Layer**: A dense layer with 5 units for classifying the flowers.

The model evaluation results for your custom flower classifier are as follows:

* **Loss:** 0.3853
* **Accuracy:** 86.38%

**GitHub Link:**