USN NUMBER: 1RVU22CSE128
NAME: Reethu RG Thota

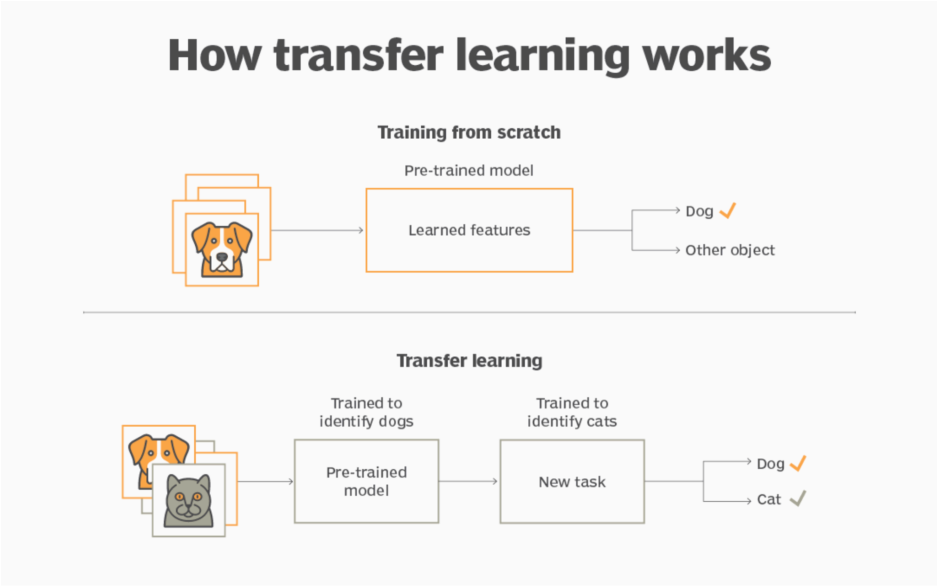
Ex No: 5 **Date:** 21/08/24	**Transfer Learning Image Classification**

Objective

To train a pre-trained Convolutional Neural Network (CNN) model using transfer learning to classify images of flowers into different categories.

Descriptions

In this experiment, we aim to implement transfer learning for image classification. The model will use a pre-trained CNN, specifically ***MobileNetV2***, from TensorFlow Hub, which will be fine-tuned on a new dataset, the flowers dataset. Transfer learning leverages the features learned by a model trained on a large dataset and applies them to a different but related problem, saving time and computational resources.



Transfer learning is a technique where a model developed for a particular task is reused as the starting point for a model on a second task. For image classification, transfer learning can involve using a pre-trained model on a large dataset like ImageNet and fine-tuning it on a smaller dataset specific to the new task. The pre-trained model acts as a feature extractor, and a new classification layer is added to adapt the model to the specific categories of the new task.

USN NUMBER: 1RVU22CSE128
NAME: Reethu RG Thota

Model

The model structure involves the following layers:

1. **Input Layer:** Accepts images as input.
2. **Base Model (MobileNetV2):** The pre-trained MobileNetV2 model from TensorFlow Hub is used as the base model. This model is pre-trained on a large dataset and serves as a fixed feature extractor or is fine-tuned for the specific task.
3. **Output Layer:** A fully connected layer with softmax activation to classify images into multiple categories.

Building the parts of algorithm

The steps involved in building the CNN with transfer learning are:

1. **Install Dependencies:** Install the necessary libraries, including TensorFlow Hub.

   !pip install tensorflow_hub

2. **Load Pre-trained Model:** Load the *MobileNetV2* model from *TensorFlow Hub*, which acts as the base model.

3. **Build the Model:** Define the model architecture by adding a new classification layer on top of the MobileNetV2 model.

   model = tf.keras.Sequential([
   base_model,
   tf.keras.layers.Dense(5, activation='softmax')])

4. **Compile the Model:** Specify the loss function, optimizer, and metrics for the model.

   model.compile(
   optimizer='adam',
   loss='sparse_categorical_crossentropy',
   metrics=['accuracy'])

5. **Train the Model**: Train the model on the flowers dataset.

   model.fit(train_dataset, epochs=10, validation_data=validation_dataset)

6. **Evaluate the Model:** Assess the performance of the model on the validation set.

   model.evaluate(validation_dataset)

USN NUMBER: 1RVU22CSE128
NAME: Reethu RG Thota

Result

Training Accuracy: 92.48% after 5 epochs.

Validation Accuracy: 86.38%, indicating good generalization.

Loss: Training loss decreased steadily; validation loss remained stable.

Model Performance: High accuracy and low loss demonstrate effective transfer learning with MobileNetV2.

GitHub Link

https://github.com/reethuthota/Deep_Learning/tree/main/Lab5