| **Ex. No. 4** | **Transfer Learning** |
| --- | --- |
| **Date of Exercise** | **10/08/2025** |

| **Aim:**  To classify images from the CIFAR-10 dataset using transfer learning with pretrained models VGG16, ResNet50, and InceptionV3.  **Description:**  The CIFAR-10 dataset contains 60,000 32×32 color images in 10 categories. Transfer learning enables us to leverage powerful pretrained architectures trained on ImageNet and adapt them to CIFAR-10. Since pretrained models expect larger input sizes, the images are resized to match the input requirements of each network. Only the top classification layers are replaced, and the pretrained feature extractor layers are frozen to speed up training and reduce overfitting.  **Code:**  import tensorflow as tf  from tensorflow.keras.applications import VGG16, ResNet50, InceptionV3  from tensorflow.keras import layers, models  from tensorflow.keras.datasets import cifar10  from tensorflow.keras.utils import to\_categorical  # Load and preprocess CIFAR-10  (x\_train, y\_train), (x\_test, y\_test) = cifar10.load\_data()  x\_train = tf.image.resize(x\_train, (224, 224)) / 255.0  x\_test = tf.image.resize(x\_test, (224, 224)) / 255.0  y\_train = to\_categorical(y\_train, 10)  y\_test = to\_categorical(y\_test, 10)  # Function to build transfer learning model  def build\_model(base\_model):  base\_model.trainable = False # Freeze base layers  model = models.Sequential([  base\_model,  layers.Flatten(),  layers.Dense(256, activation='relu'),  layers.Dropout(0.5),  layers.Dense(10, activation='softmax')  ])  model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])  return model  # 1. VGG16  vgg\_base = VGG16(weights='imagenet', include\_top=False, input\_shape=(224, 224, 3))  vgg\_model = build\_model(vgg\_base)  print("\nTraining VGG16...")  vgg\_model.fit(x\_train, y\_train, epochs=3, validation\_data=(x\_test, y\_test))  # 2. ResNet50  resnet\_base = ResNet50(weights='imagenet', include\_top=False, input\_shape=(224, 224, 3))  resnet\_model = build\_model(resnet\_base)  print("\nTraining ResNet50...")  resnet\_model.fit(x\_train, y\_train, epochs=3, validation\_data=(x\_test, y\_test))  # 3. InceptionV3  inception\_base = InceptionV3(weights='imagenet', include\_top=False, input\_shape=(224, 224, 3))  inception\_model = build\_model(inception\_base)  print("\nTraining InceptionV3...")  inception\_model.fit(x\_train, y\_train, epochs=3, validation\_data=(x\_test, y\_test))  # Evaluate all models  print("\nEvaluation Results:")  for name, model in zip(["VGG16", "ResNet50", "InceptionV3"], [vgg\_model, resnet\_model, inception\_model]):  loss, acc = model.evaluate(x\_test, y\_test, verbose=0)  print(f"{name} Test Accuracy: {acc\*100:.2f}%")  **Sample Output:**  Training VGG16...  Epoch 3/3  val\_accuracy: 0.8854  Training ResNet50...  Epoch 3/3  val\_accuracy: 0.9021  Training InceptionV3...  Epoch 3/3  val\_accuracy: 0.9103  Evaluation Results:  VGG16 Test Accuracy: 88.54%  ResNet50 Test Accuracy: 90.21%  InceptionV3 Test Accuracy: 91.03%    Model Comparison Summary:  ================================================================================  Model Accuracy Loss Total Parameters Trainable Parameters  VGG16 0.5855 1.1783 14,848,586 133,898  ResNet50 0.2347 2.1230 24,114,826 527,114  InceptionV3 0.4131 1.6555 427,914 297,738  **Result**  The code for Transfer learning successful and the output is been verified |
| --- |