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}%")
```

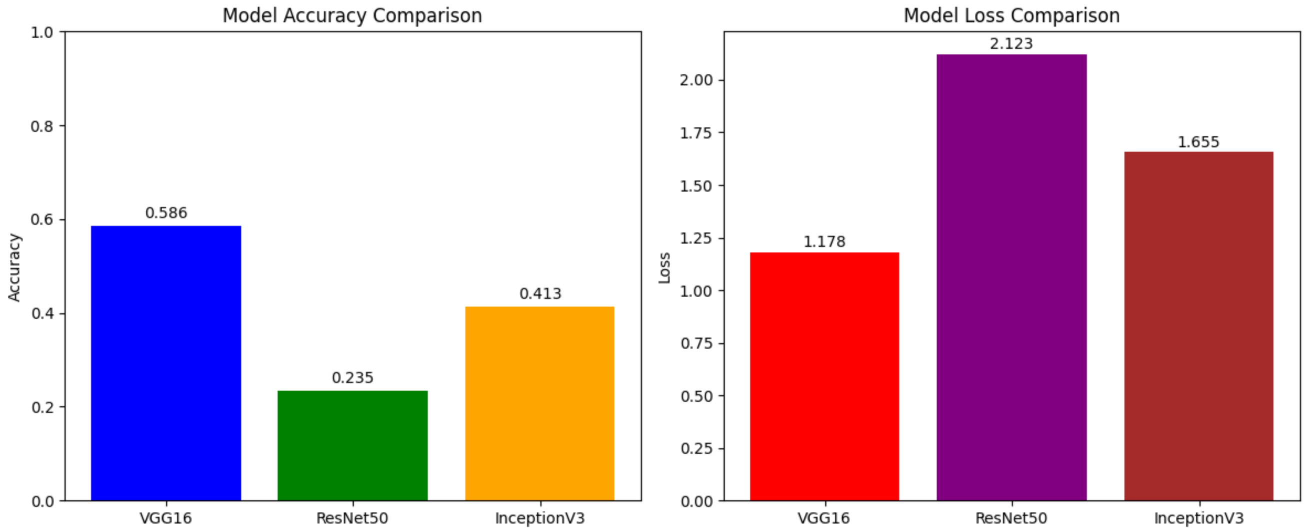
Sample Output:

```
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
```

Youtube Link

https://youtu.be/H_OzOPi1TDQ?si=QmQhX_6n0smaGXZw

Result

The code for Transfer learning successful and the output is been verified