```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.applications import
VGG16
from tensorflow.keras.models import
Model
from tensorflow.keras.layers import Dense,
Flatten
from tensorflow.keras.optimizers import
Adam
from
tensorflow.keras.preprocessing.image
import ImageDataGenerator
import numpy as np

train_dir = "C:/Users/arpit/DL all/DL
all/cifar-10-img/train"
test_dir = "C:/Users/arpit/DL all/DL
all/cifar-10-img/test"
train_datagen = ImageDataGenerator(
    rescale=1.0 / 255,
)

test_datagen = ImageDataGenerator(
    rescale=1.0 / 255,
)

train_batch_size = 5000
train_generator =
train_datagen.flow_from_directory(
    train_dir,
    target_size=(32, 32),
    batch_size=train_batch_size,
    class_mode='categorical'
)

test_batch_size = 1000
test_generator =
test_datagen.flow_from_directory(
    test_dir,
    target_size=(32, 32),
    batch_size=test_batch_size,
    class_mode='categorical'
)

x_train, y_train =  train_generator[0]
x_test, y_test = test_generator[0]

print(len(x_train))
print(len(x_test))

weights_path = "C:/Users/arpit/DL all/DL
all/vgg16_weights_tf_dim_ordering_tf_ker
nels_notop.h5"
base_model =
VGG16(weights=weights_path,
include_top=False, input_shape=(32, 32,
3))

for layer in base_model.layers:
    layer.trainable = False
x = Flatten()(base_model.output)
x = Dense(256, activation='relu')(x)
x = tf.keras.layers.Dropout(0.3)(x)
x = Dense(256, activation='relu')(x)
x = tf.keras.layers.Dropout(0.3)(x)
predictions = Dense(10,
activation='softmax')(x)

model = Model(inputs=base_model.input,
outputs=predictions)

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

model.fit(x_train, y_train, batch_size=64,
epochs=10, validation_data=(x_test,
y_test))
```

```
base_model =
VGG16(weights=weights_path,
include_top=False, input_shape=(32, 32,
3))

for layer in base_model.layers:
  layer.trainable = False

for layer in
base_model.layers[len(base_model.layers)
- 4:]:
  layer.trainable = True

x = Flatten()(base_model.output)
x = Dense(256, activation='relu')(x)
x = tf.keras.layers.Dropout(0.3)(x)
x = Dense(512, activation='relu')(x)
x = tf.keras.layers.Dropout(0.3)(x)
predictions = Dense(10,
activation='softmax')(x)

model = Model(inputs=base_model.input,
outputs=predictions)

model.compile(optimizer=Adam(learning_
rate=0.001),
loss='categorical_crossentropy',
metrics=['accuracy'])

model.fit(x_train, y_train, batch_size=64,
epochs=10, validation_data=(x_test,
y_test))

import matplotlib.pyplot as plt
predicted_value = model.predict(x_test)

labels =
list(test_generator.class_indices.keys())

n = 890
plt.imshow(x_test[n])
print("Preditcted:
",labels[np.argmax(predicted_value[n])])
print("Actual: ",
labels[np.argmax(y_test[n])])
```