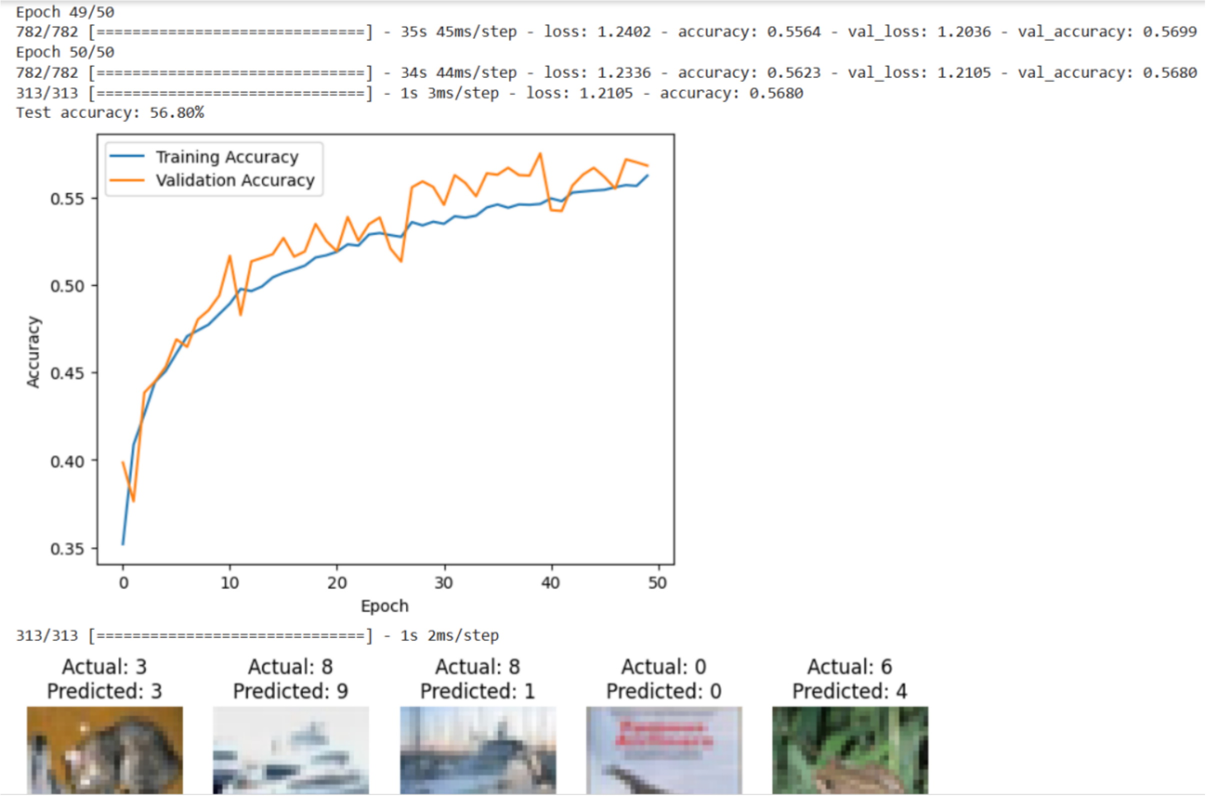
PROGRAM

```
import tensorflow as tf
from tensorflow.keras import layers, models, initializers
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.preprocessing.image import
ImageDataGenerator
import matplotlib.pyplot as plt

(train_images, train_labels), (test_images, test_labels) =
cifar10.load_data()
train_images, test_images = train_images / 255.0, test_images
/ 255.0
train_labels = to_categorical(train_labels, 10)
test_labels = to_categorical(test_labels, 10)
datagen = ImageDataGenerator(
    rotation_range=15,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
)
datagen.fit(train_images)

model = models.Sequential()
model.add(layers.Flatten(input_shape=(32, 32, 3)))
model.add(layers.Dense(512, activation='relu',
kernel_initializer='glorot_uniform'))
model.add(layers.BatchNormalization())
model.add(layers.Dense(256, activation='relu',
kernel_initializer='glorot_uniform'))
model.add(layers.BatchNormalization())
model.add(layers.Dense(128, activation='relu',
kernel_initializer='glorot_uniform'))
model.add(layers.BatchNormalization())
model.add(layers.Dense(10, activation='softmax',
kernel_initializer='glorot_uniform'))
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate
=0.0001),
              loss='categorical_crossentropy',metrics=['accura
cy'])
history = model.fit(datagen.flow(train_images, train_labels,
batch_size=64),epochs=50,validation_data=(test_images,
test_labels))
test_loss, test_acc = model.evaluate(test_images, test_labels)
print(f'Test accuracy: {test_acc * 100:.2f}%')
```

```
plt.plot(history.history['accuracy'], label='Training
Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation
Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()

import numpy as np
predictions = model.predict(test_images)
predicted_labels = np.argmax(predictions, axis=1)
plt.figure(figsize=(10, 5))
for i in range(5):
    plt.subplot(1, 5, i + 1)
    plt.imshow(test_images[i])
    plt.title(f"Actual:
{np.argmax(test_labels[i])}\nPredicted:
{predicted_labels[i]}")
    plt.axis('off')
plt.show()
```

OUTPUT:



```
Epoch 49/50
782/782 [==============================] - 35s 45ms/step - loss: 1.2402 - accuracy: 0.5564 - val_loss: 1.2036 - val_accuracy: 0.5699
Epoch 50/50
782/782 [==============================] - 34s 44ms/step - loss: 1.2336 - accuracy: 0.5623 - val_loss: 1.2105 - val_accuracy: 0.5680
313/313 [==============================] - 1s 3ms/step - loss: 1.2105 - accuracy: 0.5680
Test accuracy: 56.80%
```

```
313/313 [==============================] - 1s 2ms/step
```