1: CNN FOR IMAGE CLASSIFICATION

AIM:

To demonstrate the working of a simple Convolutional Neural Network (CNN) for image classification using CIFAR-10 dataset.

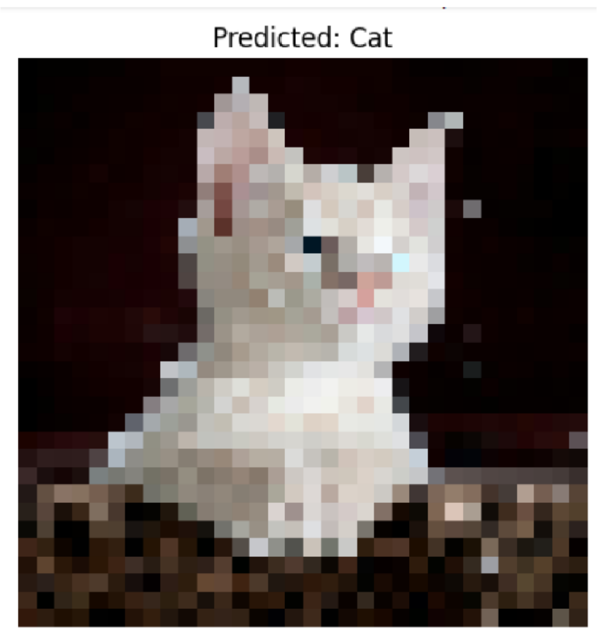
PROCEDURE:

1. Load and normalize the CIFAR-10 dataset.
2. Define a CNN model with convolutional, pooling, and dense layers.
3. Compile the model using Adam optimizer and sparse categorical crossentropy loss.
4. Train the model on the training dataset.
5. Evaluate the model on the test dataset and display accuracy.

CODE:

```
import tensorflow as tf
from tensorflow.keras import layers, models
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.cifar10.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
model = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(10)
])
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
model.fit(x_train, y_train, epochs=10, validation_data=(x_test, y_test))
test_loss, test_acc = model.evaluate(x_test, y_test)
print(f'Test accuracy: {test_acc:.2f}')
```

OUTPUT:



RESULT:

CNN model achieved around 70–80% test accuracy on CIFAR-10 dataset.