Image Classification using CNN Model from popular CIFAR-10 dataset

In [1]:

```
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.utils import to_categorical
```

In [2]:

```
(X_train, y_train), (X_test, y_test) = cifar10.load_data()
```

In [3]:

```
X_train, X_test = X_train / 255.0, X_test / 255.0
```

In [4]:

```
y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)
```

In [5]:

```
model = models.Sequential([
    layers.Input(shape=(32, 32, 3)),
    layers.Conv2D(32, (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, activation='softmax') #Output layer (10 classes for CIFAR-10)
])
```

In [6]:

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

In [10]:

```
history = model.fit(X_train, y_train, epochs=10, batch_size=64, validation_data=(X_test, y_test))
```

```
Epoch 1/10
782/782 ━━━━━━━━━━━━━━━━━━━━ 11s 14ms/step - accuracy: 0.8767 - loss: 0.3565 - val_accuracy: 0.6879 - val_loss: 1.0603
Epoch 2/10
782/782 ━━━━━━━━━━━━━━━━━━━━ 11s 14ms/step - accuracy: 0.8983 - loss: 0.3033 - val_accuracy: 0.6862 - val_loss: 1.1424
Epoch 3/10
782/782 ━━━━━━━━━━━━━━━━━━━━ 11s 14ms/step - accuracy: 0.9125 - loss: 0.2604 - val_accuracy: 0.6820 - val_loss: 1.2664
Epoch 4/10
782/782 ━━━━━━━━━━━━━━━━━━━━ 11s 14ms/step - accuracy: 0.9251 - loss: 0.2209 - val_accuracy: 0.6721 - val_loss: 1.4504
Epoch 5/10
782/782 ━━━━━━━━━━━━━━━━━━━━ 11s 14ms/step - accuracy: 0.9316 - loss: 0.1990 - val_accuracy: 0.6740 - val_loss: 1.4041
Epoch 6/10
782/782 ━━━━━━━━━━━━━━━━━━━━ 11s 14ms/step - accuracy: 0.9477 - loss: 0.1582 - val_accuracy: 0.6780 - val_loss: 1.5323
Epoch 7/10
782/782 ━━━━━━━━━━━━━━━━━━━━ 11s 14ms/step - accuracy: 0.9558 - loss: 0.1356 - val_accuracy: 0.6630 - val_loss: 1.7589
Epoch 8/10
782/782 ━━━━━━━━━━━━━━━━━━━━ 11s 14ms/step - accuracy: 0.9574 - loss: 0.1259 - val_accuracy: 0.6744 - val_loss: 1.7462
Epoch 9/10
782/782 ━━━━━━━━━━━━━━━━━━━━ 11s 14ms/step - accuracy: 0.9597 - loss: 0.1145 - val_accuracy: 0.6668 - val_loss: 1.9338
Epoch 10/10
782/782 ━━━━━━━━━━━━━━━━━━━━ 11s 14ms/step - accuracy: 0.9670 - loss: 0.0977 - val_accuracy: 0.6694 - val_loss: 2.0289
```

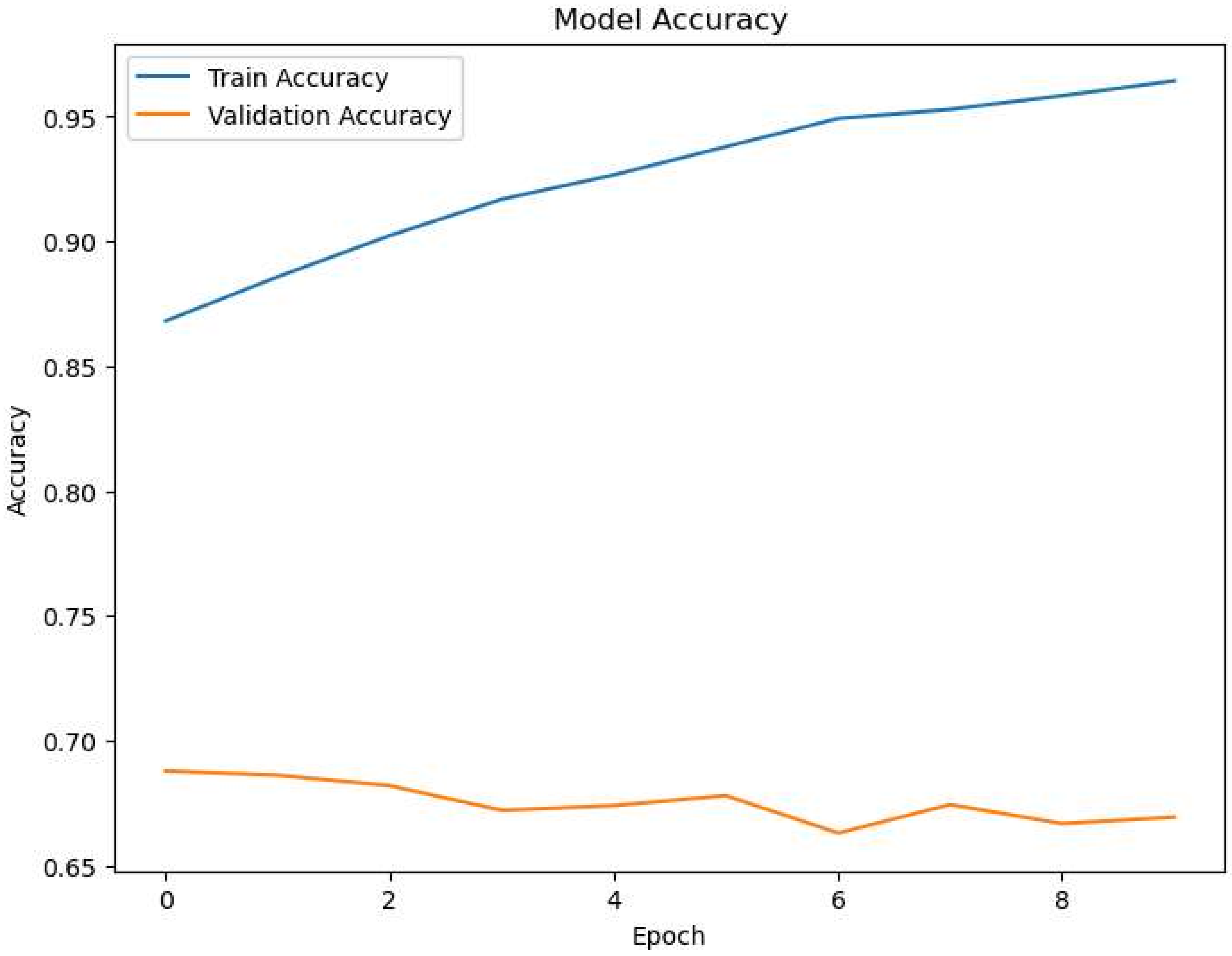
In [11]:

```
test_loss, test_accuracy = model.evaluate(X_test, y_test)
print(f"Test Accuracy: {test_accuracy*100:.2f}%")
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.6681 - loss: 2.0284
Test Accuracy: 66.94%
```

In [12]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 6))
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Model Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```


Model Accuracy
Train Accuracy
Validation Accuracy
Accuracy
0.95
0.90
0.85
0.80
0.75
0.70
0.65
Epoch
0
2
4
6
8

In [ ]: