```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.utils import to_categorical
```

```
# Load and preprocess the CIFAR-10 dataset
(X_train, y_train), (X_test, y_test) = cifar10.load_data()
X_train = X_train / 255.0
X_test = X_test / 255.0
y_train = to_categorical(y_train, num_classes=10)
y_test = to_categorical(y_test, num_classes=10)
```

```
# Load a pre-trained model (VGG16 in this example)
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(32, 32, 3))
```

```
# Freeze the pre-trained model layers
for layer in base_model.layers:
  layer.trainable = False
```

```
# Add custom classification layers on top of the pre-trained model
x = Flatten()(base_model.output)
x = Dense(256, activation='relu')(x)
x = Dense(128, activation='relu')(x)
output = Dense(10, activation='softmax')(x)
```

```
# Create the final model
model = Model(inputs=base_model.input, outputs=output)
```

```
# Compile the model
model.compile(optimizer=SGD(learning_rate=0.001, momentum=0.9), loss='categorical_crossentropy', metrics=['accuracy'])
```

```
# Train the model
model.fit(X_train, y_train, batch_size=32, epochs=10, validation_data=(X_test, y_test))
```

```
Epoch 1/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 793s 507ms/step - accuracy: 0.3306 - loss: 1.9110 - val_accuracy: 0.4659 - val_loss: 1.4934
Epoch 2/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 777s 491ms/step - accuracy: 0.5031 - loss: 1.4125 - val_accuracy: 0.5211 - val_loss: 1.3653
Epoch 3/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 822s 504ms/step - accuracy: 0.5366 - loss: 1.3269 - val_accuracy: 0.5367 - val_loss: 1.3182
Epoch 4/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 802s 504ms/step - accuracy: 0.5514 - loss: 1.2837 - val_accuracy: 0.5591 - val_loss: 1.2663
Epoch 5/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 797s 500ms/step - accuracy: 0.5649 - loss: 1.2406 - val_accuracy: 0.5511 - val_loss: 1.2794
Epoch 6/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 765s 489ms/step - accuracy: 0.5718 - loss: 1.2217 - val_accuracy: 0.5587 - val_loss: 1.2577
Epoch 7/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 765s 490ms/step - accuracy: 0.5813 - loss: 1.1975 - val_accuracy: 0.5784 - val_loss: 1.2089
Epoch 8/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 767s 491ms/step - accuracy: 0.5881 - loss: 1.1750 - val_accuracy: 0.5731 - val_loss: 1.2154
Epoch 9/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 808s 494ms/step - accuracy: 0.5925 - loss: 1.1640 - val_accuracy: 0.5782 - val_loss: 1.1967
Epoch 10/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 821s 506ms/step - accuracy: 0.5948 - loss: 1.1520 - val_accuracy: 0.5810 - val_loss: 1.1976
<keras.src.callbacks.history.History at 0x7fc4da507400>
```

```
# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print("Test Loss:", loss)
print("Test Accuracy:", accuracy)
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 129s 411ms/step - accuracy: 0.5847 - loss: 1.1934
Test Loss: 1.1976414918899536
Test Accuracy: 0.5809999704360962
```

```
import numpy as np
```

```
# Make Predictions on New Data
# Predict class for a single test image
img = X_test[1]  # Take one image from the test set
img = np.expand_dims(img, axis=0)  # Expand dimensions to fit the model input shape
```

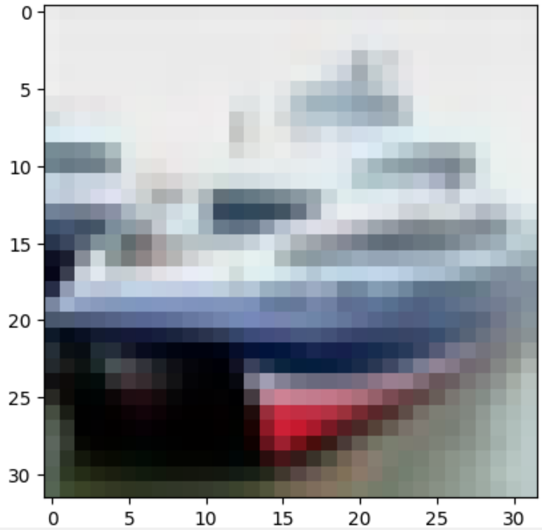
```
# Get prediction
pred = model.predict(img)
predicted_class = np.argmax(pred, axis=1)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 295ms/step
```

```
# Map predicted class index to the class label
class_labels = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck']
print(f"Predicted class: {class_labels[predicted_class[0]]}")
```

```
Predicted class: ship
```

```
import matplotlib.pyplot as plt
img = X_test[1]
#img = img.squeeze(axis=0)
plt.imshow(img)
plt.show()
```



Start coding or generate with AI.