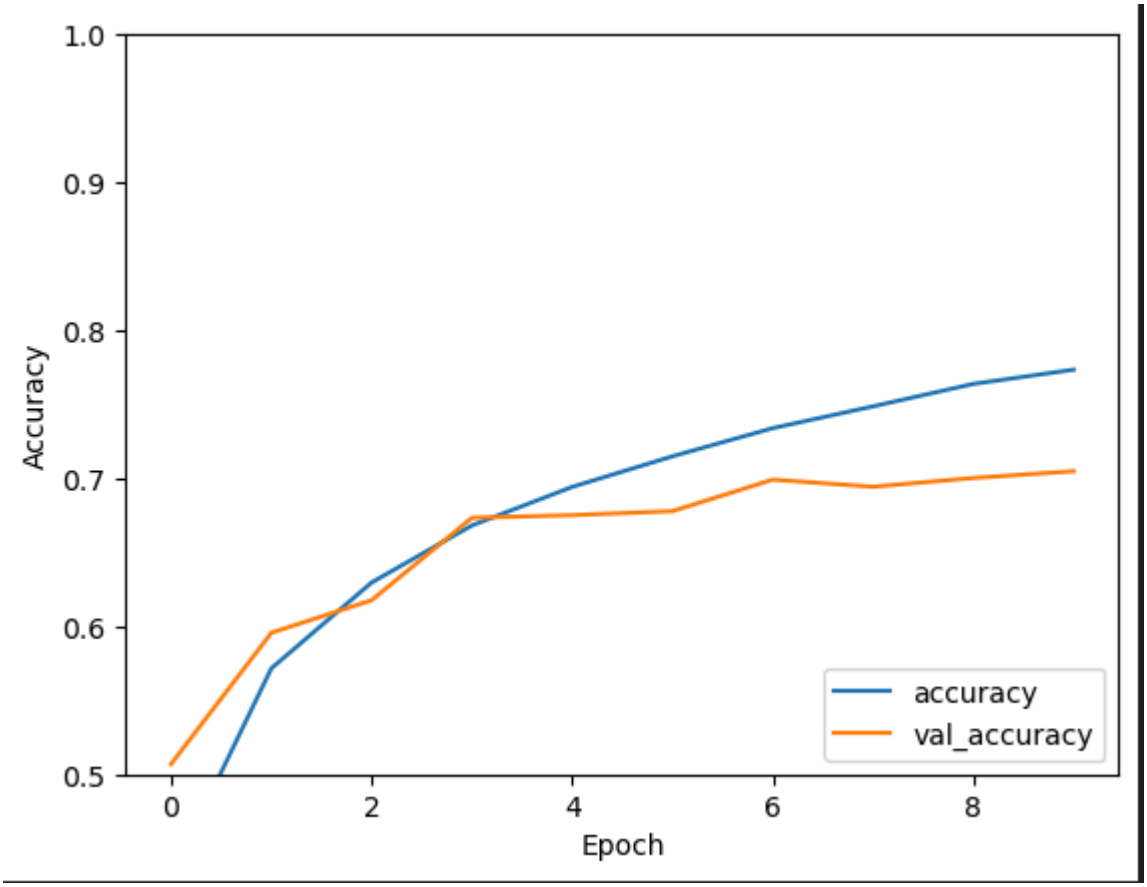
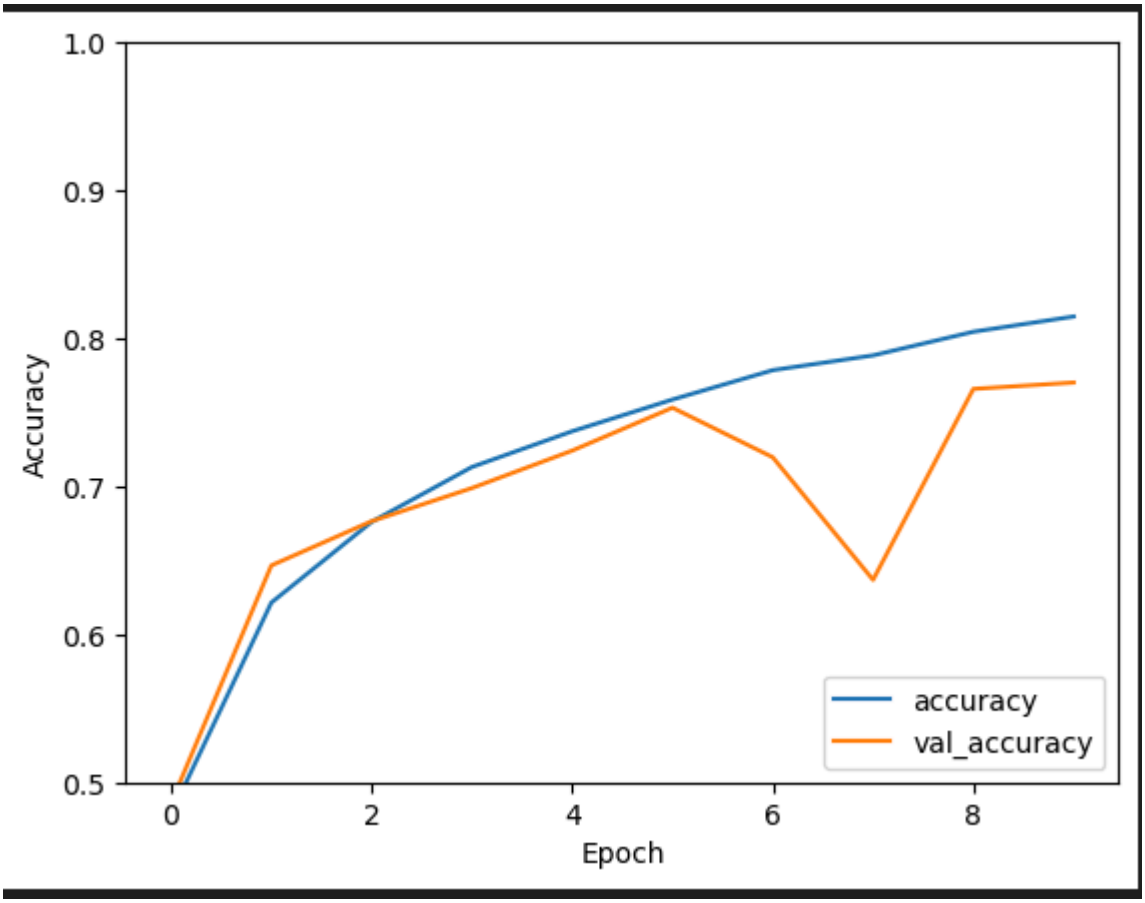
My original model and its accuracy:

```python
def create_model_1():
    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)
    ])
    return model
```



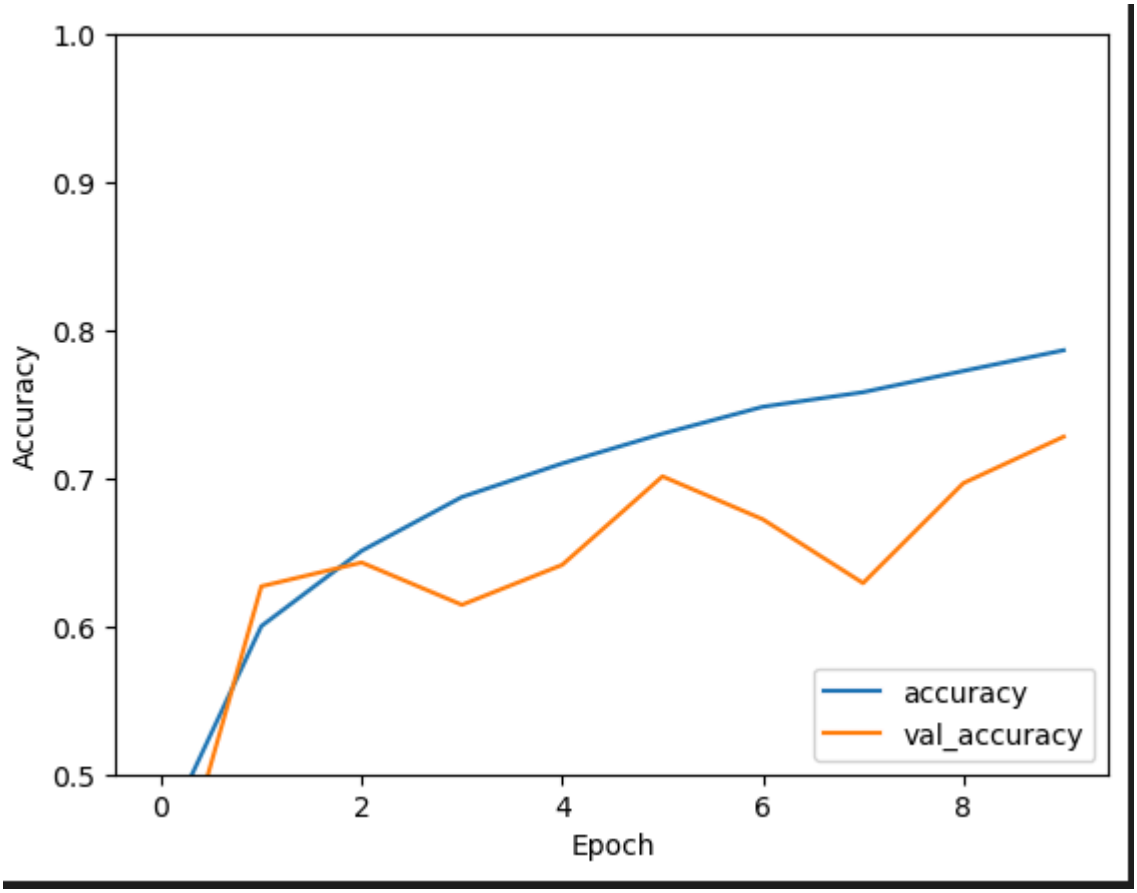

My improved model and its accuracy:

```python
def create_model_2():
    model = models.Sequential([
        layers.Conv2D(64, (3, 3), activation='relu', input_shape=(32, 32, 3)),
        layers.BatchNormalization(),
        layers.MaxPooling2D((2, 2)),
        layers.Dropout(0.3),
        layers.Conv2D(128, (3, 3), activation='relu'),
        layers.BatchNormalization(),
        layers.MaxPooling2D((2, 2)),
        layers.Dropout(0.3),
        layers.Conv2D(256, (3, 3), activation='relu'),
        layers.Flatten(),
        layers.Dense(128, activation='relu'),
        layers.BatchNormalization(),
        layers.Dropout(0.5),
        layers.Dense(10)
    ])
    return model
```



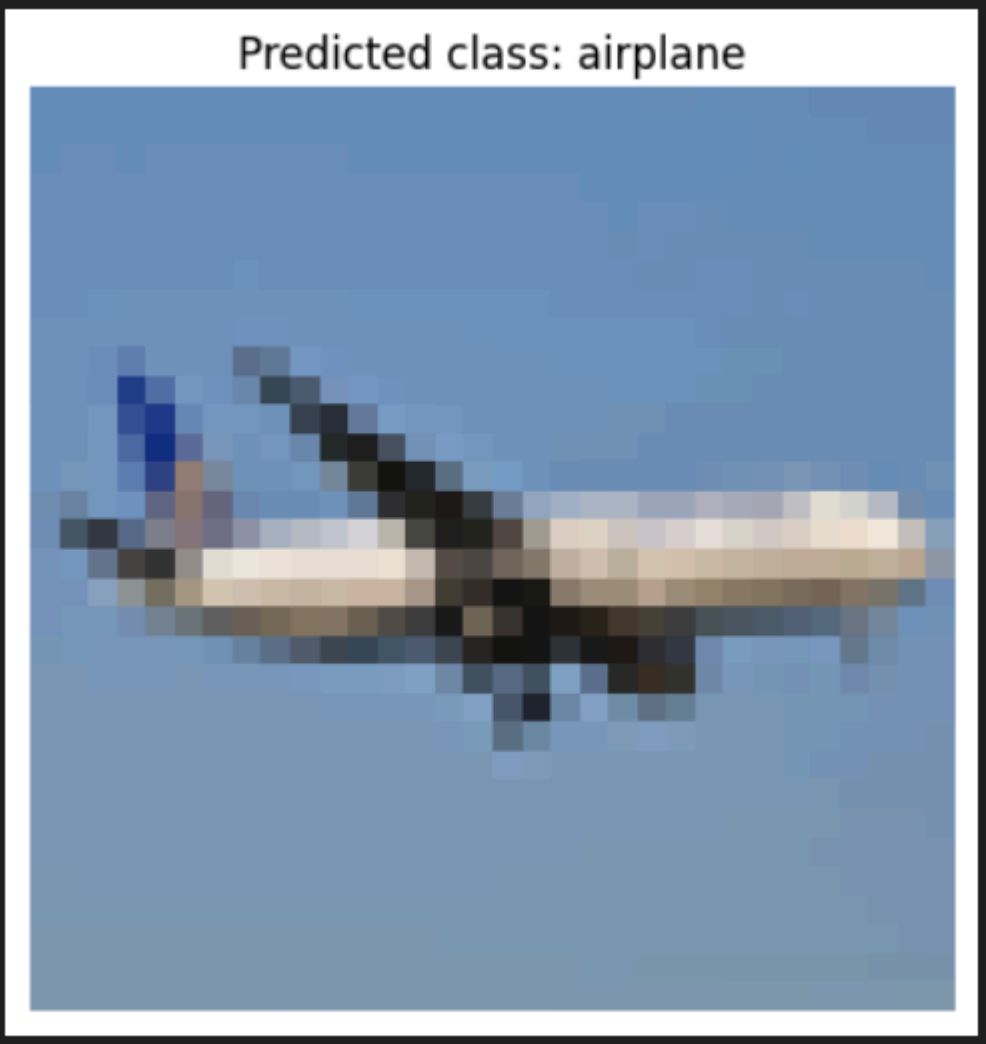

My other improved model and its accuracy:

```python
def create_model_3():
    model = models.Sequential([
        layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)),
        layers.BatchNormalization(),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3), activation='relu'),
        layers.BatchNormalization(),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(128, (3, 3), activation='relu'),
        layers.BatchNormalization(),
        layers.MaxPooling2D((2, 2)),
        layers.GlobalAveragePooling2D(),
        layers.Dense(128, activation='relu'),
        layers.Dropout(0.5),
        layers.Dense(10, activation='softmax')
    ])
    return model
```



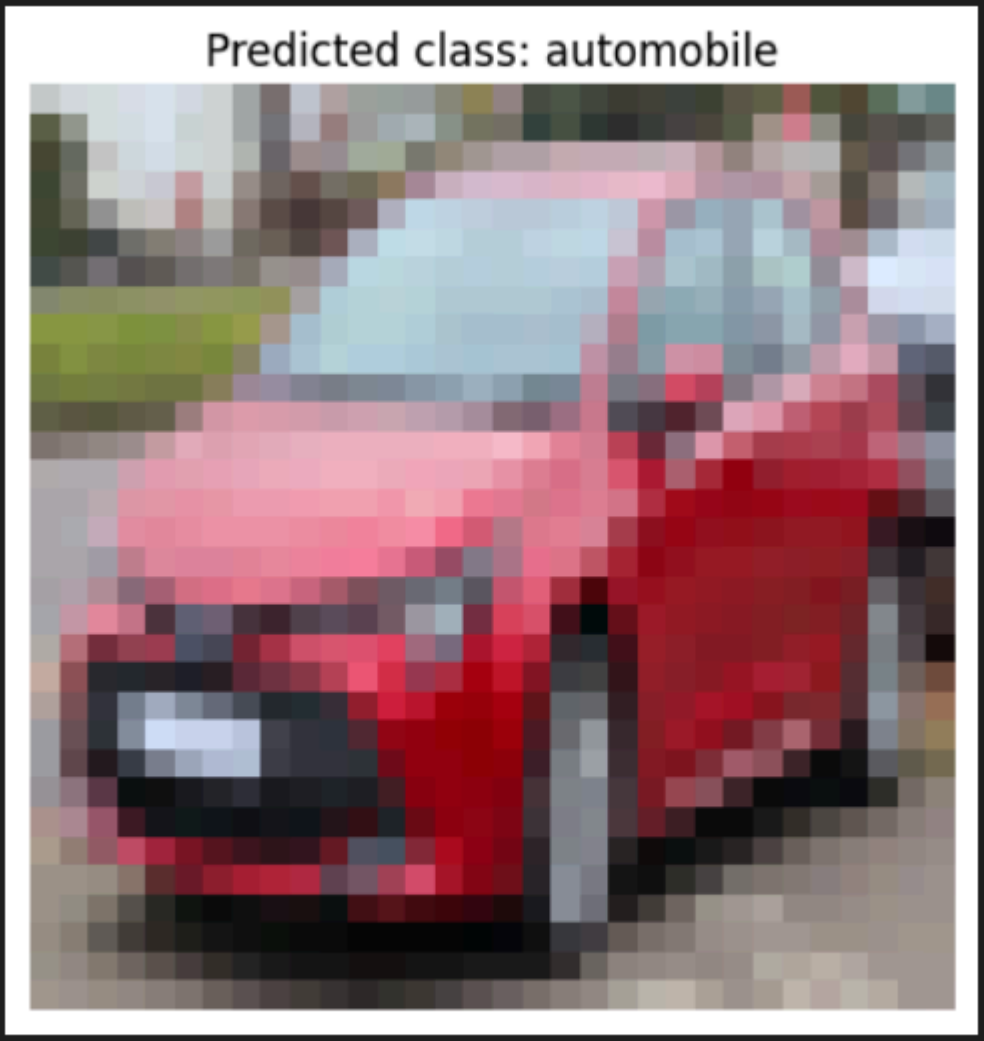

Prediction from model 1:

```
Model: 1
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step
```



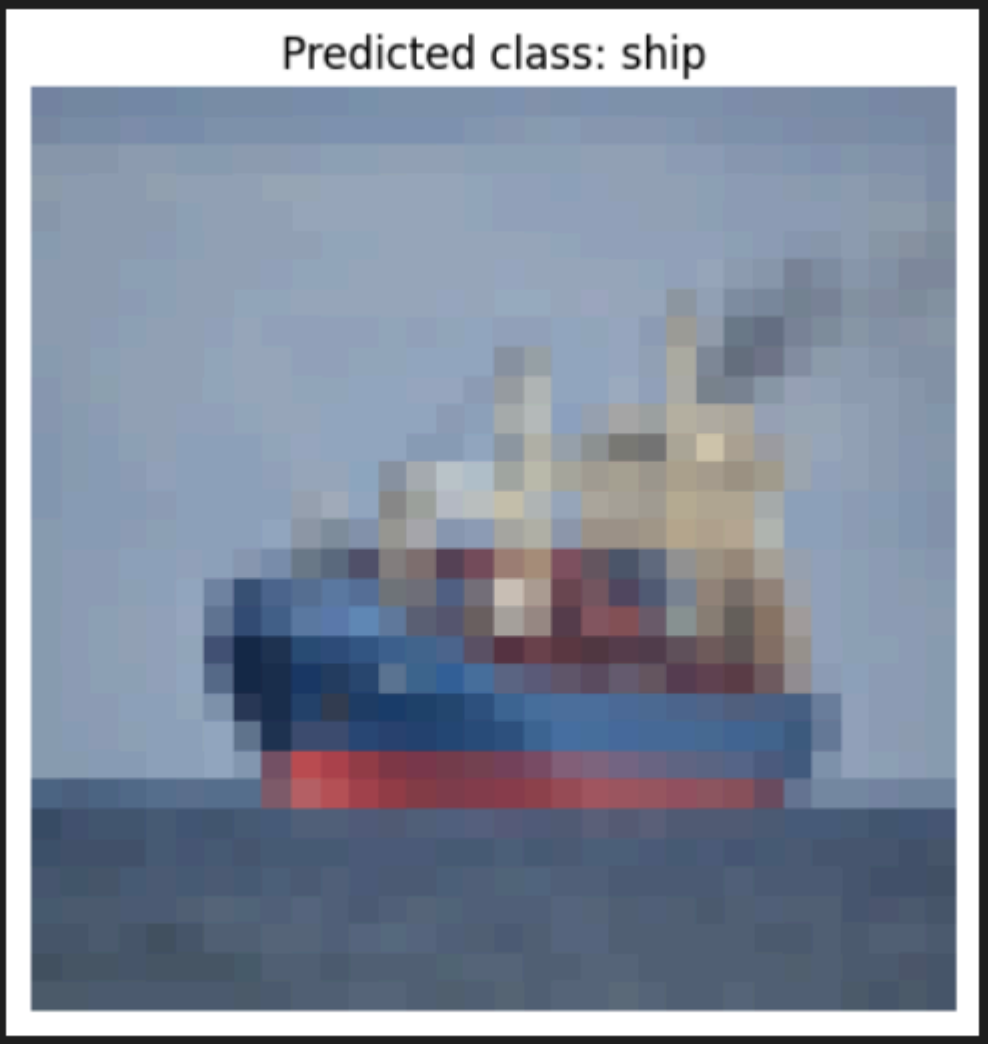

```
Image directory: /content/test/airplane.jpg
Predicted class: airplane
```

```
Model: 1
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step
```




```
Image directory: /content/test/automobile.jpg
Predicted class: automobile
```

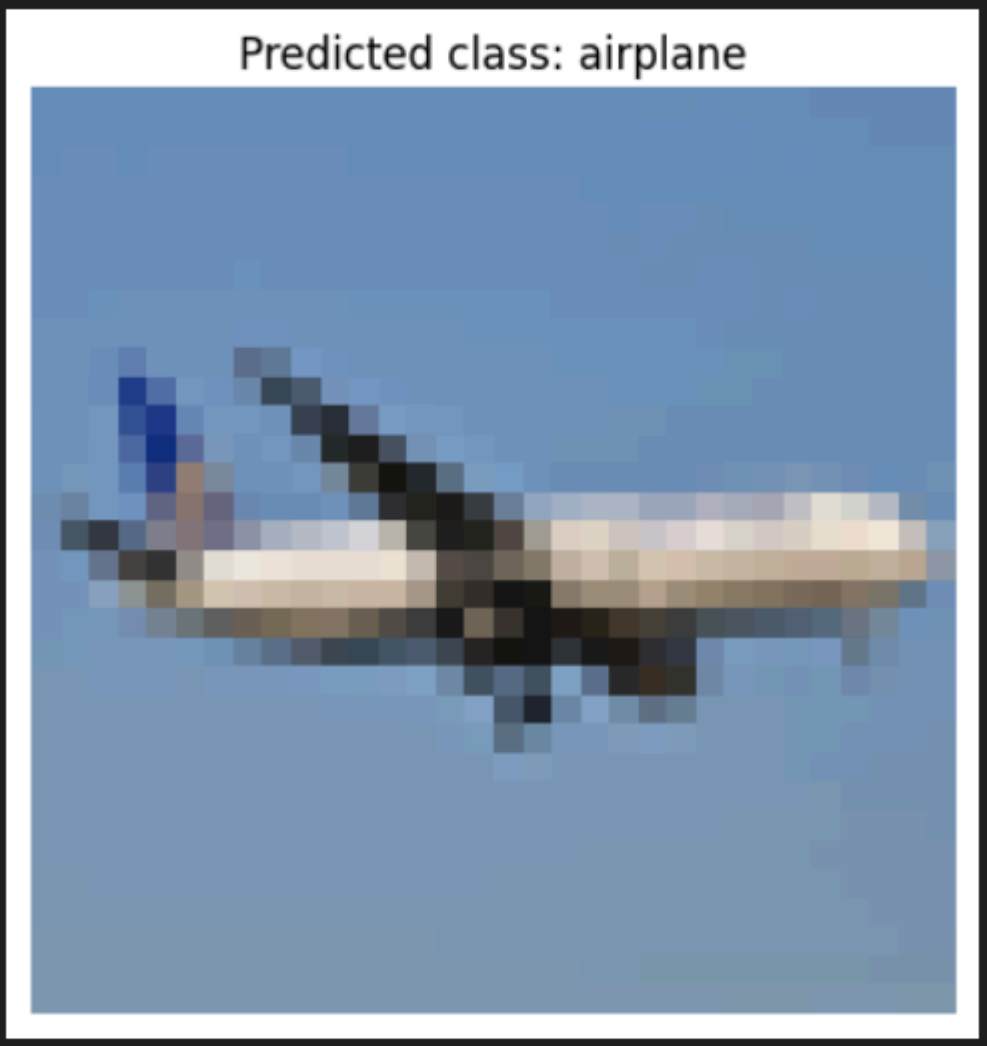
```
Model: 1
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step
```




```
Image directory: /content/test/ship.jpg
Predicted class: ship
```

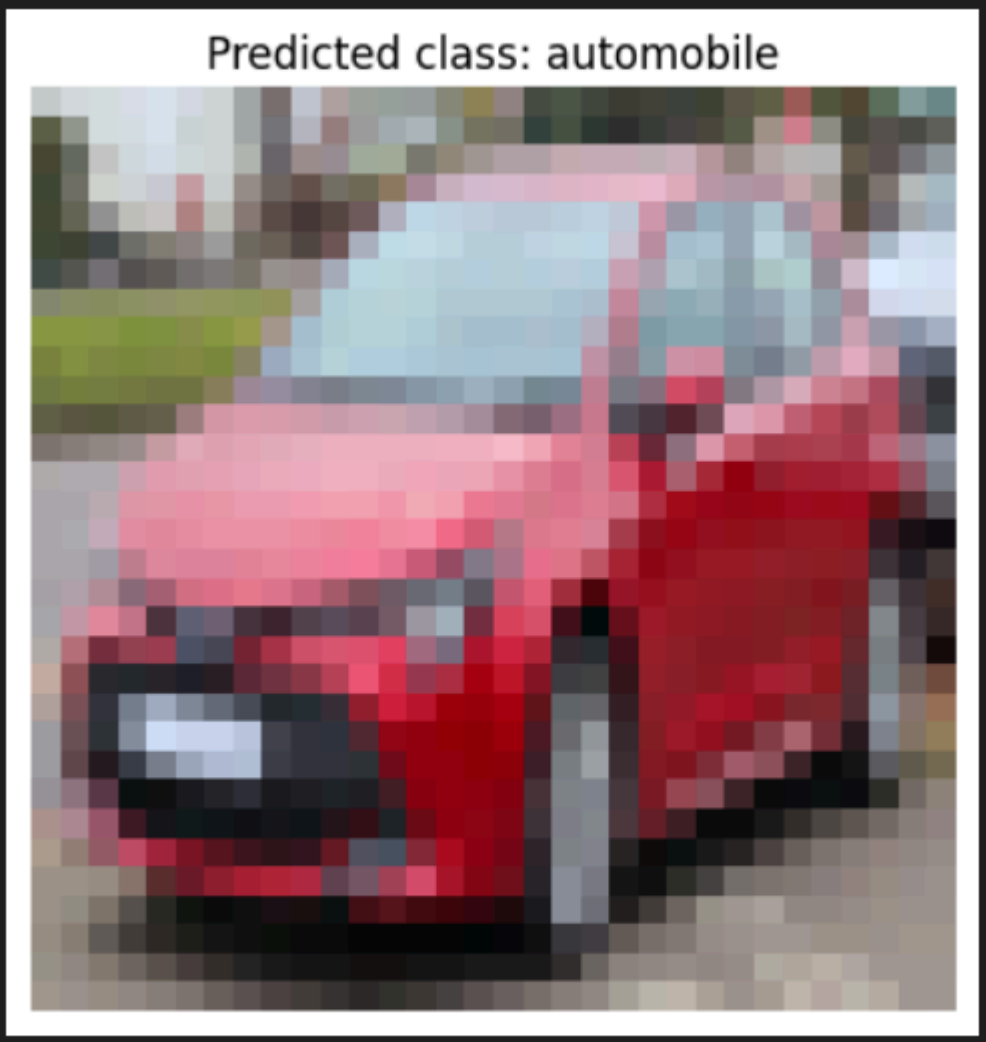
Prediction from model 2:

```
Model: 2
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
```



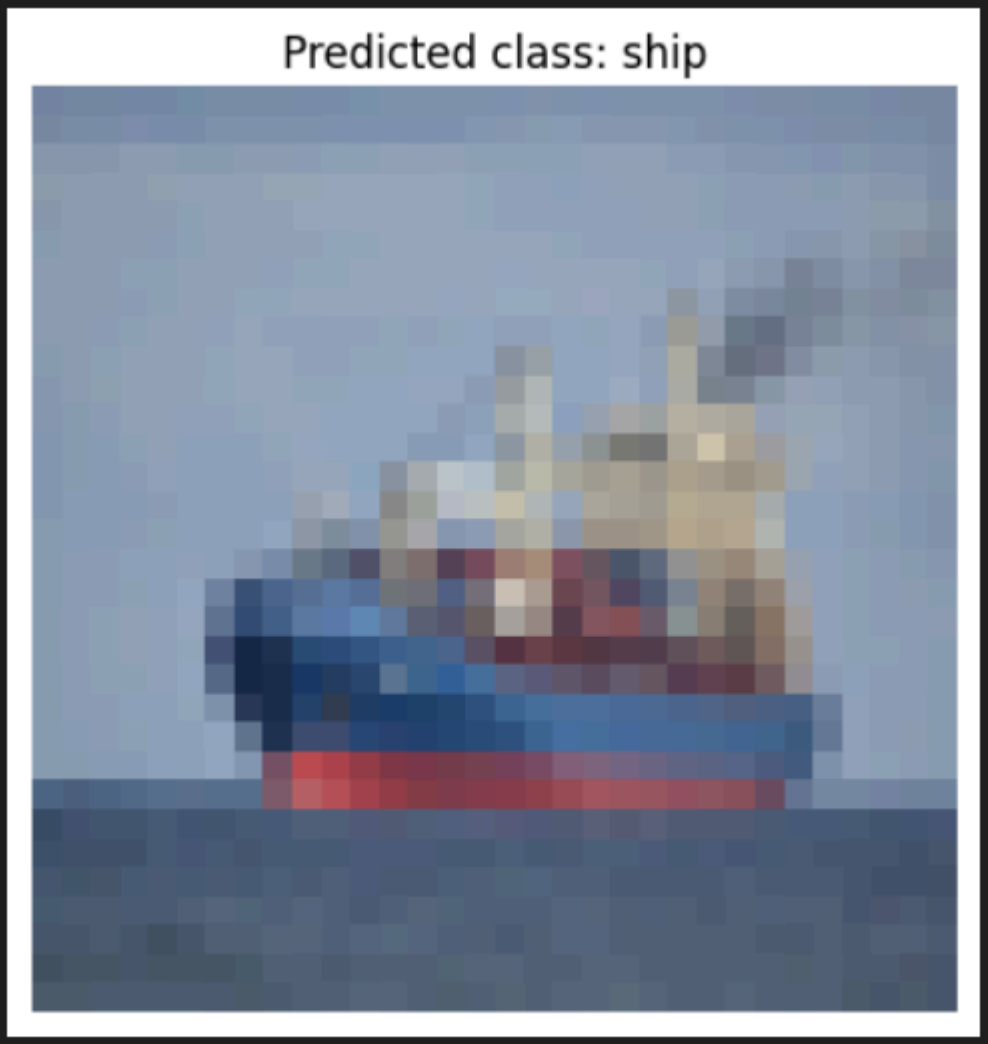

```
Image directory: /content/test/airplane.jpg
Predicted class: airplane
```

```
Model: 2
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step
```




```
Image directory: /content/test/automobile.jpg
Predicted class: automobile
```

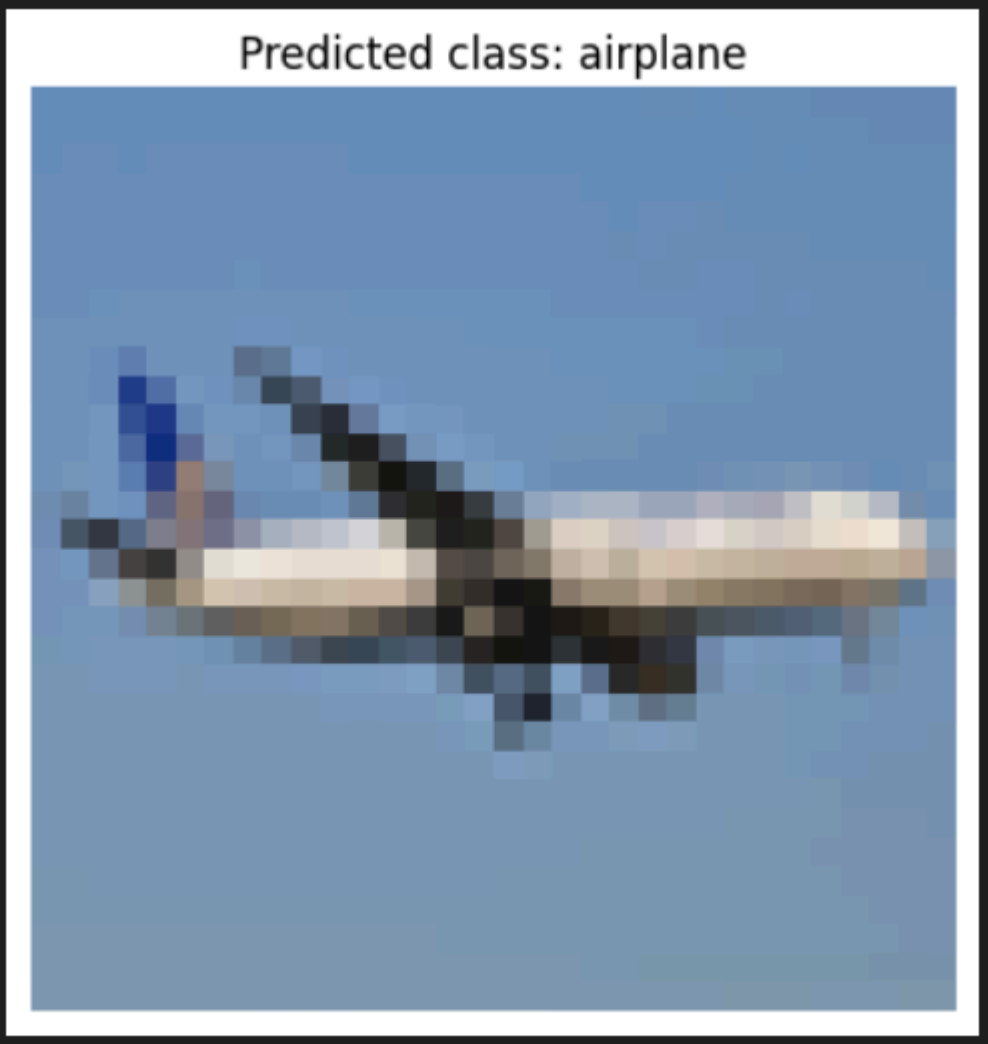
```
Model: 2
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step
```




```
Image directory: /content/test/ship.jpg
Predicted class: ship
```

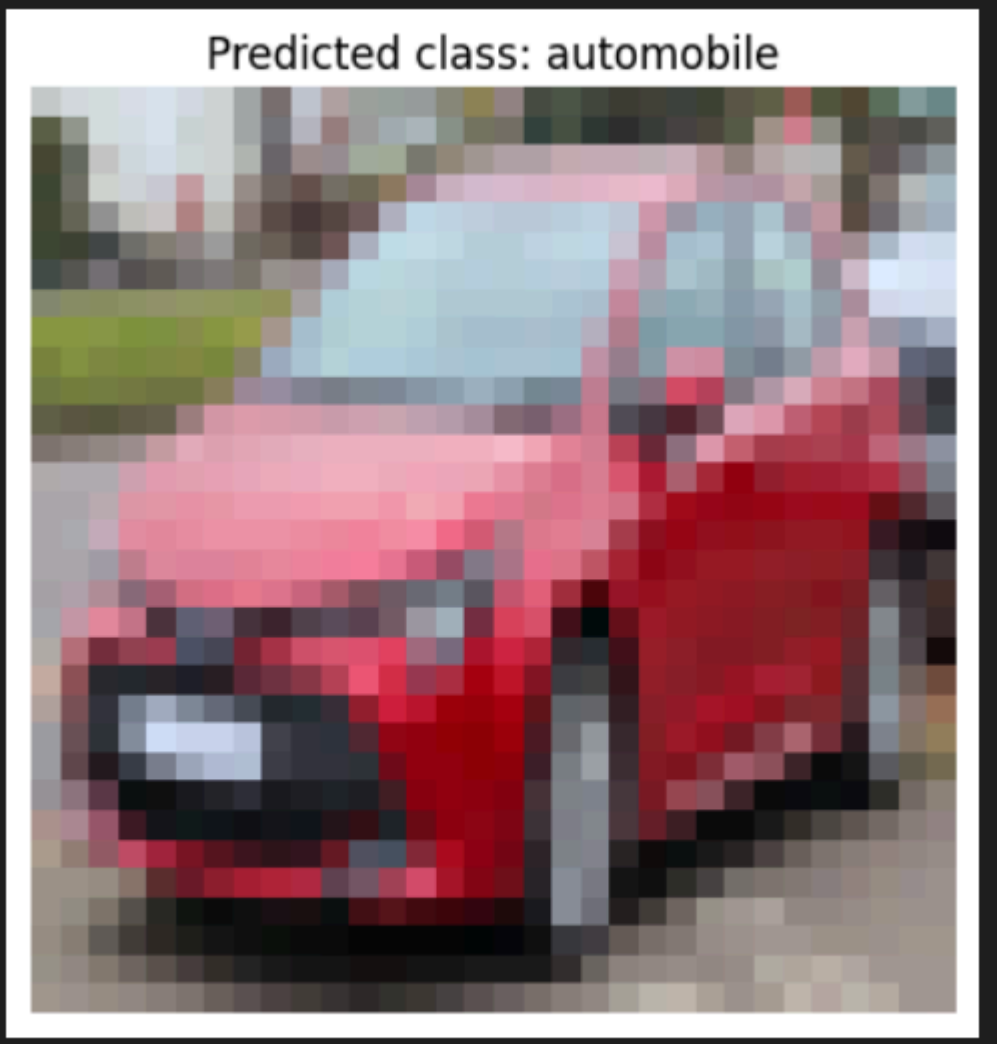
Prediction from model 3:

```
Model: 3
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step
```



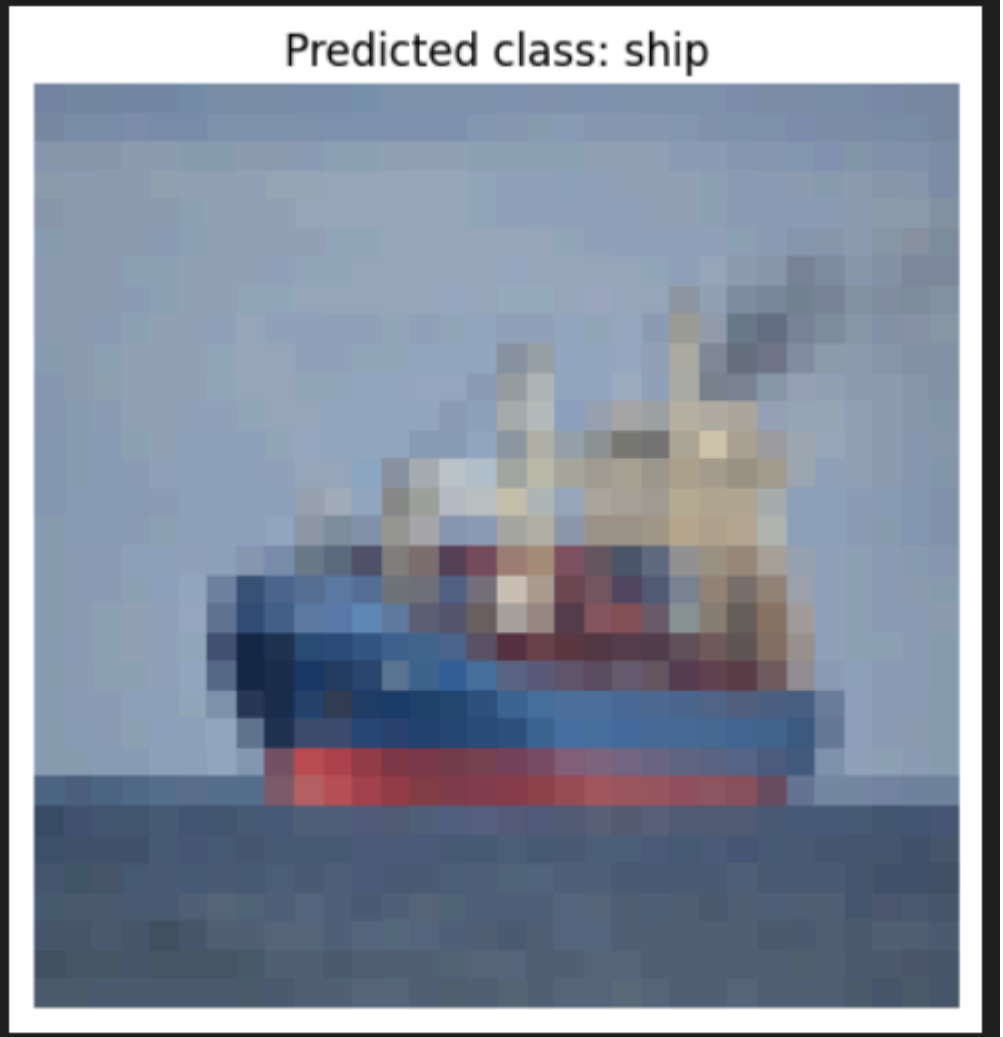

```
Image directory: /content/test/airplane.jpg
Predicted class: airplane
```

```
Model: 3
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step
```




```
Image directory: /content/test/automobile.jpg
Predicted class: automobile
```

```
Model: 3
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step
```




```
Image directory: /content/test/ship.jpg
Predicted class: ship
```