```
import tensorflow as tf
from PIL import Image
import numpy as np
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.utils import to_categorical



(x_train, y_train), (x_test, y_test) = cifar10.load_data()
train_subset_size = 5000
test_subset_size = 1000
x_train = x_train[:train_subset_size]
y_train = y_train[:train_subset_size]
x_test = x_test[:test_subset_size]
y_test = y_test[:test_subset_size]
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0
num_classes = 10
y_train = to_categorical(y_train, num_classes)
y_test = to_categorical(y_test, num_classes)

base_model = MobileNetV2(weights='imagenet', include_top=False,
input_shape=(32, 32, 3))
```

```
for layer in base_model.layers:
    layer.trainable = False
model = Sequential()
model.add(base_model)
model.add(GlobalAveragePooling2D())
model.add(Dense(256, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
model.fit(x_train, y_train, batch_size=32, epochs=5, validation_data=(x_test,
y_test))
_, accuracy = model.evaluate(x_test, y_test, verbose=0)
print('Test accuracy:', accuracy)



image_path = 'cat.jpg'
image = Image.open(image_path)
image = image.resize((32, 32))
image = np.array(image)
image = image.astype('float32') / 255.0
image = np.expand_dims(image, axis=0)
predictions = model.predict(image)
predicted_labels = tf.argmax(predictions, axis=1)
class_names = [
'airplane', 'automobile', 'bird', 'cat', 'deer',
'dog', 'frog', 'horse', 'ship', 'truck'
]
```

```
print('Predicted class:', class_names[predicted_labels[0]])
```

```
<PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=220x230 at
0x7EA99ABE4910>

<PIL.Image.Image image mode=RGB size=32x32 at 0x7EA99ABE4190>

[[[ 4  0  1] [ 4  0  1] [ 5  0  0] ... [ 9  0  0] [ 9  0  0] [ 9  0  0]] [[ 4  0  1] [ 5
1  2] [ 6  0  1] ... [ 9  0  0] [ 9  0  0] [ 9  0  0]] [[ 5  0  0] [ 6  0  1] [ 8  0  0]
... [ 9  0  0] [ 9  0  0] [10  0  1]] ... [[39 25 19] [38 28 25] [67 46 36] ...
[78 61 49] [16 13 11] [24 18 13]] [[59 34 25] [36 22 21] [53 36 30] ...
[48 33 28] [33 22 17] [48 39 31]] [[53 27 15] [59 36 29] [53 33 24] ...
[20 12 10] [64 50 40] [60 50 41]]]



[[[0.01568628 0.         0.00392157]
[0.01568628 0.         0.00392157] [0.01960784 0.         0.        ] ... [0.03529412 0.         0.        ]
[0.03529412 0.         0.        ] [0.03529412 0.         0.        ]] [[0.01568628 0.         0.00392157]
[0.01960784 0.00392157 0.00784314] [0.02352941 0.         0.00392157] ...
[0.03529412 0.         0.        ] [0.03529412 0.         0.        ] [0.03529412 0.         0.        ]] [[0.01960784
0.         0.        ] [0.02352941 0.         0.00392157] [0.03137255 0.         0.        ] ... [0.03529412 0.
0.        ] [0.03529412 0.         0.        ] [0.03921569 0.         0.00392157]] ... [[0.15294118
0.09803922 0.07450981] [0.14901961 0.10980392 0.09803922] [0.2627451
0.18039216 0.14117648] ... [0.30588236 0.23921569 0.19215687] [0.0627451
0.05098039 0.04313726] [0.09411765 0.07058824 0.05098039]] [[0.23137255
0.13333334 0.09803922] [0.14117648 0.08627451 0.08235294] [0.20784314
0.14117648 0.11764706] ... [0.1882353  0.12941177 0.10980392] [0.12941177
0.08627451 0.06666667] [0.1882353  0.15294118 0.12156863]] [[0.20784314
0.10588235 0.05882353] [0.23137255 0.14117648 0.11372549] [0.20784314
0.12941177 0.09411765] ... [0.07843138 0.04705882 0.03921569] [0.2509804
0.19607843 0.15686275] [0.23529412 0.19607843 0.16078432]]]


[[[[0.01568628 0.         0.00392157] [0.01568628 0.         0.00392157] [0.01960784 0.         0.
] ... [0.03529412 0.         0.        ] [0.03529412 0.         0.        ] [0.03529412 0.         0.        ]]
[[0.01568628 0.         0.00392157] [0.01960784 0.00392157 0.00784314] [0.02352941
0.         0.00392157] ... [0.03529412 0.         0.        ] [0.03529412 0.         0.        ] [0.03529412 0.
0.        ]] [[0.01960784 0.         0.        ] [0.02352941 0.         0.00392157] [0.03137255 0.         0.        ]
... [0.03529412 0.         0.        ] [0.03529412 0.         0.        ] [0.03921569 0.         0.00392157]]
... [[0.15294118 0.09803922 0.07450981] [0.14901961 0.10980392 0.09803922]
[0.2627451  0.18039216 0.14117648] ... [0.30588236 0.23921569 0.19215687]
[0.0627451  0.05098039 0.04313726] [0.09411765 0.07058824 0.05098039]]
[[0.23137255 0.13333334 0.09803922] [0.14117648 0.08627451 0.08235294]
[0.20784314 0.14117648 0.11764706] ... [0.1882353  0.12941177 0.10980392]
[0.12941177 0.08627451 0.06666667] [0.1882353  0.15294118 0.12156863]]
[[0.20784314 0.10588235 0.05882353] [0.23137255 0.14117648 0.11372549]
```

[0.20784314 0.12941177 0.09411765] ... [0.07843138 0.04705882 0.03921569]
[0.2509804 0.19607843 0.15686275] [0.23529412 0.19607843 0.16078432]]]]