

# Make predictions in production

```
import numpy as np
import matplotlib.pyplot as plt
import os
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model

# (height, width, channels)
input_shape = (224, 224, 3)
folder_models = '../models'
path_test = '../test'

def prediction(path: 'str', model) -> None:
    test_img = image.load_img(path, target_size=input_shape)
    test_img = image.img_to_array(test_img) / 255.0
    test_img = np.expand_dims(test_img, axis=0)

    prob = model.predict(test_img)[0][0]

    print(f"Probability to be: 🐱 Cat {(1-prob):.4f}, 🐶 Dog: {prob:.4f}")
    print(f" 🐶 Dog\n" if prob >= 0.5 else " 🐱 Cat\n")

model = load_model(os.path.join(folder_models, 'model_v3.h5'))

# Select images to be tested
test = [os.path.join(path_test, 'a.jpeg'),
        os.path.join(path_test, 'b.jpeg'),
        os.path.join(path_test, 'c.jpeg')]
label_test = [1, 1, 0]

# Show test images
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(8, 5))
for ax, img_path, target in zip(axes, test, label_test):
    img = image.load_img(img_path, target_size=input_shape)
    ax.imshow(img)
    ax.set_title(f"Target: {target}")
    ax.axis("off")
plt.tight_layout()
plt.show()
```

Target: 1



Target: 1



Target: 0



```
# Make predictions
for img in test:
    prediction(img, model)

1/1 [=====] - 0s 107ms/step
Probability to be: 🐱 Cat 0.3449, 🐶 Dog: 0.6551
🐶 Dog

1/1 [=====] - 0s 90ms/step
Probability to be: 🐱 Cat 0.3479, 🐶 Dog: 0.6521
🐶 Dog

1/1 [=====] - 0s 101ms/step
Probability to be: 🐱 Cat 0.9186, 🐶 Dog: 0.0814
🐱 Cat
```

Model version 3 is good enough!!.