```python
In [1]: from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input, decode_pr
        from tensorflow.keras.preprocessing import image
        import numpy as np
        import matplotlib.pyplot as plt
        import matplotlib.image as mpimg

        from pathlib import Path
        import os

        model = ResNet50(weights='imagenet')

        def process_image(img_path):
            img = image.load_img(img_path, target_size=(224, 224))
            img = image.img_to_array(img)
            img = np.expand_dims(img, axis=0)
            img = preprocess_input(img)
            return(img)

        def predict_image(processed_img):
            preds = model.predict(processed_img)
            prediction = decode_predictions(preds, top=1)[0][0]
            _, description, probability = prediction
            return description, probability
```

2023-04-24 00:00:46.887251: I tensorflow/core/platform/cpu_feature_guard.cc:193] This Te
nsorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the
following CPU instructions in performance-critical operations: SSE4.1 SSE4.2 AVX AVX2 F
MA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler fla
gs.

```python
In [2]: import pandas as pd
        from IPython.display import Image

        current_dir = Path(os.getcwd()).absolute()
        images_dir = current_dir.joinpath('images')

        for root, dirs, pictures in os.walk(images_dir):
            for picture in pictures:
                image_path = Path(root).joinpath(picture)
                img = process_image(image_path)
                description, probability = predict_image(img)
                pic = mpimg.imread(image_path)
                plt.imshow(pic)
                plt.title(f'{picture}\nPrediction: {description:^}\nProbability: {probability:.3
                plt.show()
```

1/1 [==============================] - 1s 685ms/step

dog.jpg
Prediction: standard_poodle
Probability: 0.937

```
1/1 [==============================] - 0s 71ms/step
```


bed.jpeg
Prediction: chest
Probability: 0.641

```
1/1 [==============================] - 0s 72ms/step
```

boat.jpg
Prediction: speedboat
Probability: 0.506


```
1/1 [==============================] - 0s 80ms/step
```

stopsign.jpg
Prediction: street_sign
Probability: 0.771


```
1/1 [==============================] - 0s 76ms/step
```

cat.jpg
Prediction: Egyptian_cat
Probability: 0.610
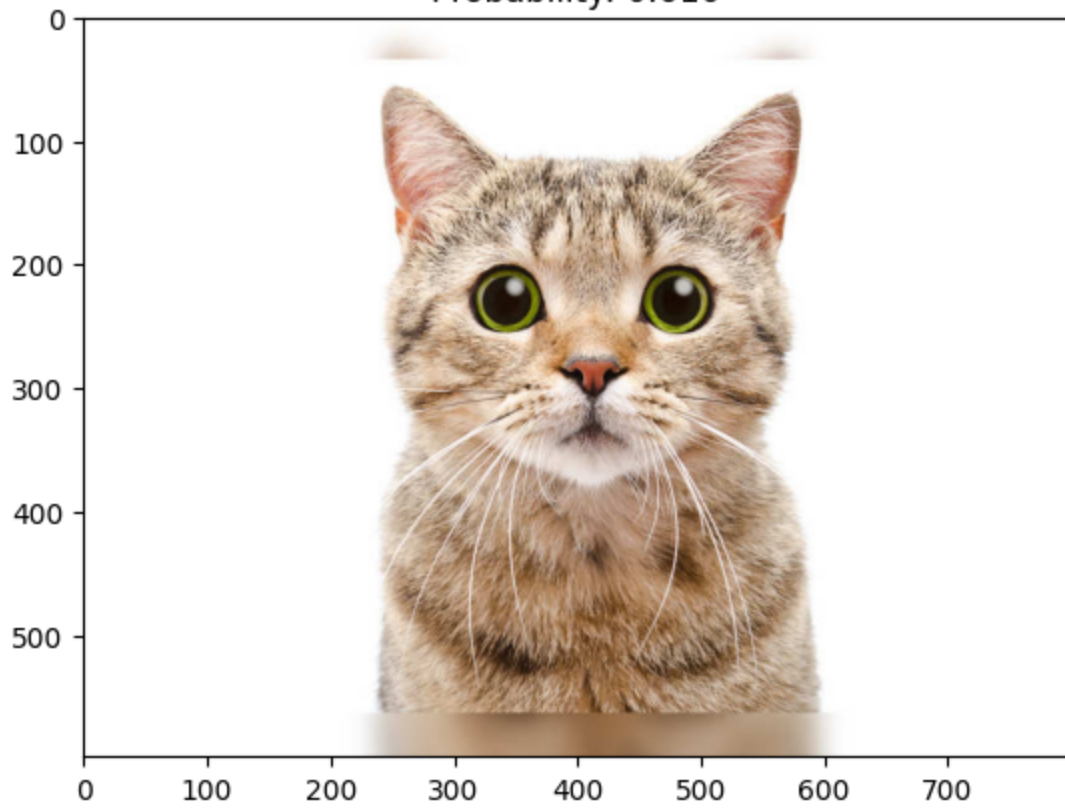
1/1 [==============================] - 0s 72ms/step

car.jpg
Prediction: sports_car
Probability: 0.533

1/1 [==============================] - 0s 72ms/step

carboat.jpg
Prediction: amphibian
Probability: 0.754
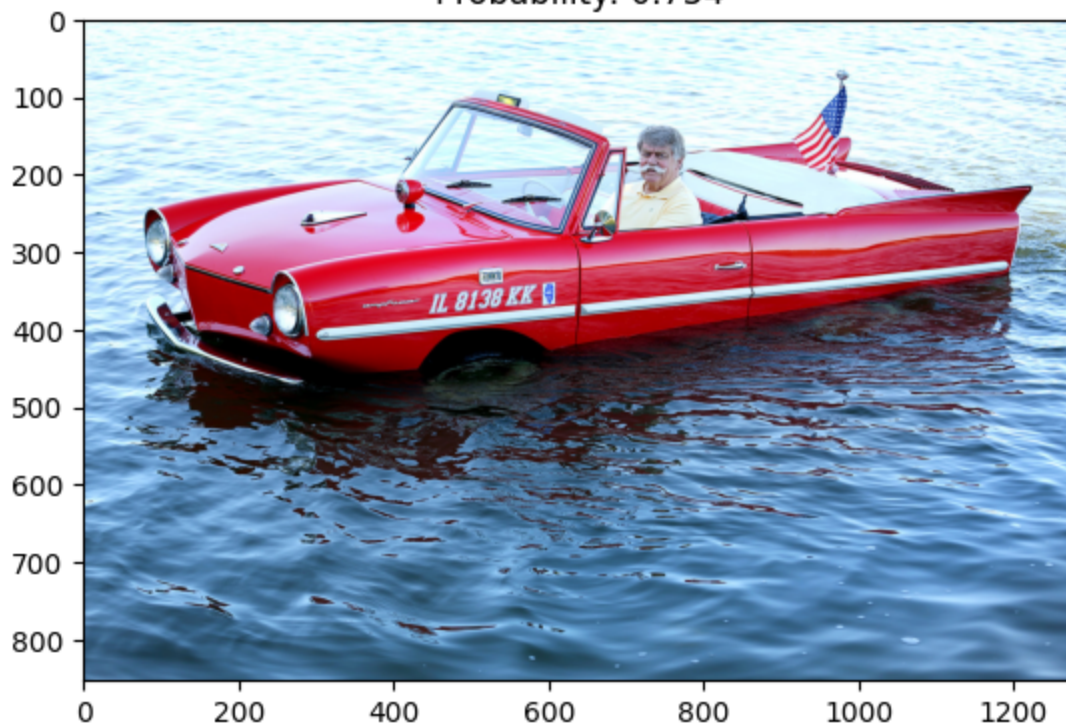

```
1/1 [==============================] - 0s 70ms/step
```

bird1.jpg
Prediction: jacamar
Probability: 0.134
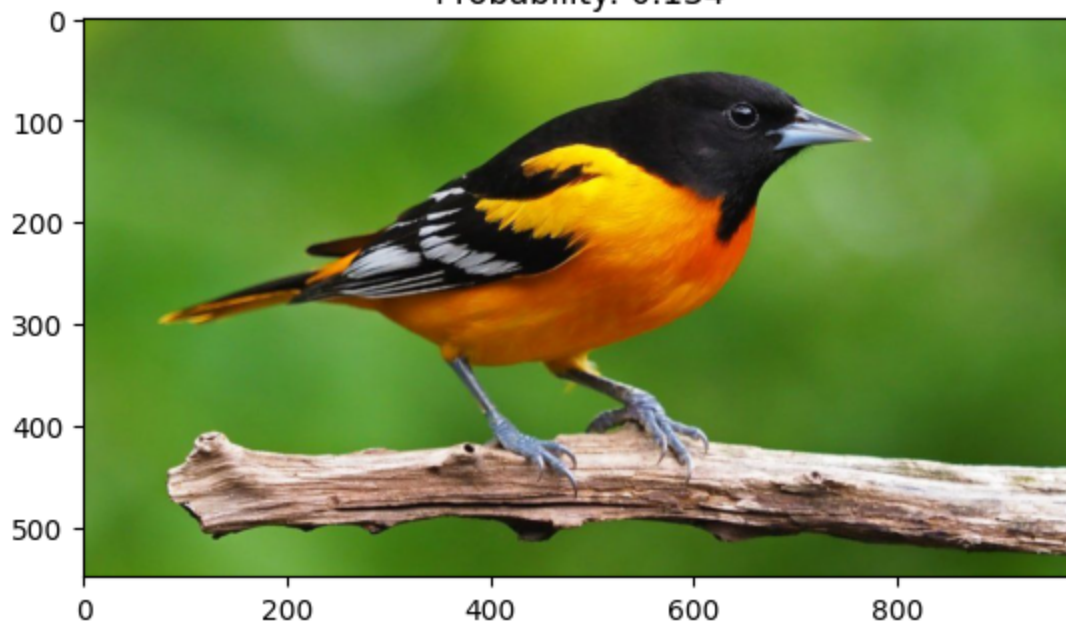

```
1/1 [==============================] - 0s 71ms/step
```

bird2.jpg
Prediction: toucan
Probability: 0.993
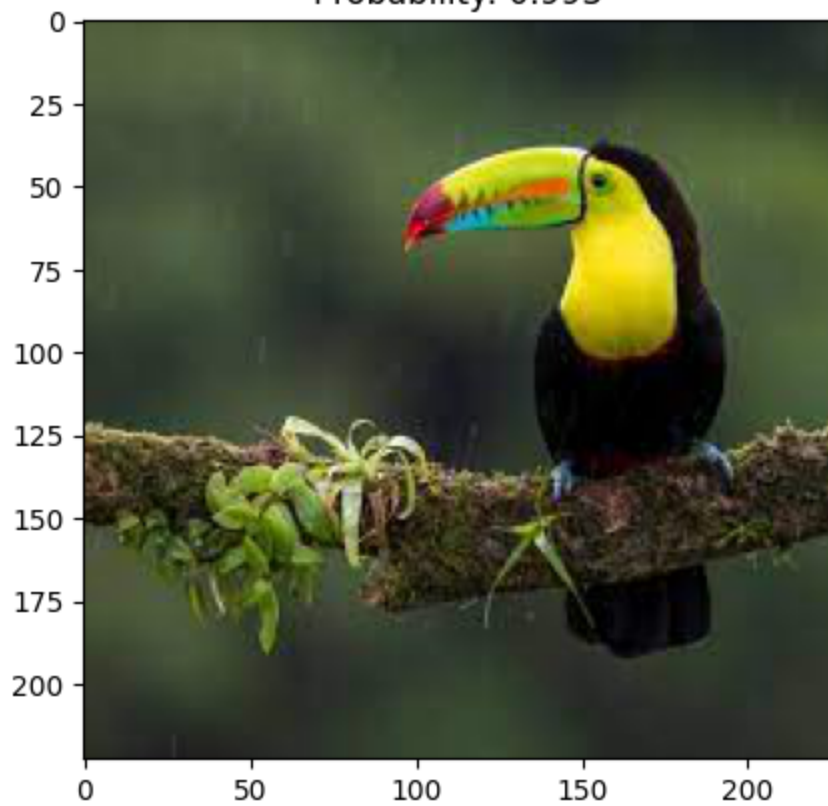
```
1/1 [==============================] - 0s 76ms/step
```


snake.jpg
Prediction: Indian_cobra
Probability: 1.000

In [ ]: