4.2D - Adversarial Attacks on Computer Vision

August 20, 2022

Welcome to your assignment this week!

To better understand adverse attacks againsts AI and how it is possible to fool an AI system, in this assignment, we will look at a Computer Vision use case.

This assessment creates an *adversarial example* using the Fast Gradient Signed Method (FGSM) attack as described in Explaining and Harnessing Adversarial Examples by Goodfellow *et al*. This was one of the first and most popular attacks to fool a neural network.

1 What is an adversarial example?

Adversarial examples are specialised inputs created with the purpose of confusing a neural network, resulting in the misclassification of a given input. These notorious inputs are indistinguishable to the human eye, but cause the network to fail to identify the contents of the image. There are several types of such attacks, however, here the focus is on the fast gradient sign method attack, which is a *white box* attack whose goal is to ensure misclassification. A white box attack is where the attacker has complete access to the model being attacked.

2 Fast gradient sign method

The fast gradient sign method works by using the gradients of the neural network to create an adversarial example. For an input image, the method uses the gradients of the loss with respect to the input image to create a new image that maximises the loss. This new image is called the adversarial image. This can be summarised using the following expression:

$$adv\_x = x + \epsilon * \text{sign}(\nabla_x J(\theta, x, y))$$

where

- adv_x : Adversarial image.
- x : Original input image.
- y : Original input label.
- ϵ : Multiplier to ensure the perturbations are small.
- θ : Model parameters.
- J : Loss.

An intriguing property here, is the fact that the gradients are taken with respect to the input image. This is done because the objective is to create an image that maximises the loss. A method to accomplish this is to find how much each pixel in the image contributes to the loss value, and

add a perturbation accordingly. This works pretty fast because it is easy find how each input pixel contributes to the loss, by using the chain rule, and finding the required gradients. Hence, the gradients are used with respect to the image. In addition, since the model is no longer being trained (thus the gradient is not taken with respect to the trainable variables, i.e., the model parameters), and so the model parameters remain constant. The only goal is to fool an already trained model.

3 Part 1

So let's try and fool a pretrained model. In this first part, the model is MobileNetV2 model, pretrained on ImageNet.

Run the following cell to load the packages you will need.

[1]:
```
import tensorflow as tf
import matplotlib as mpl
import matplotlib.pyplot as plt
import pandas as pd

mpl.rcParams['figure.figsize'] = (8, 8)
mpl.rcParams['axes.grid'] = False
```

Let's load the pretrained MobileNetV2 model and the ImageNet class names.

[2]:
```
pretrained_model = tf.keras.applications.
 ↪MobileNetV2(include_top=True,weights='imagenet')
pretrained_model.trainable = False

# ImageNet labels
decode_predictions = tf.keras.applications.mobilenet_v2.decode_predictions
```

[3]:
```
# Helper function to preprocess the image so that it can be inputted in␣
 ↪MobileNetV2
def preprocess(image):
    image = tf.cast(image, tf.float32)
    image = tf.image.resize(image, (224, 224))
    image = tf.keras.applications.mobilenet_v2.preprocess_input(image)
    image = image[None, ...]
    return image

#  Helper function to extract labels from probability vector
def get_imagenet_label(probs):
    return decode_predictions(probs, top=5)[0]
```

3.1 Original image

Let's use a sample image of a Labrador Retriever -by Mirko CC-BY-SA 3.0 from Wikimedia Common and create adversarial examples from it. The first step is to preprocess it so that it can be fed as an input to the MobileNetV2 model.

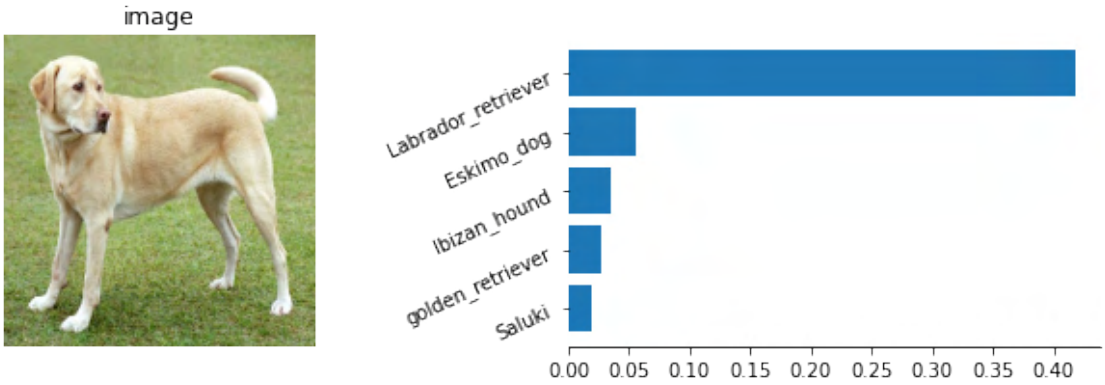
[4]:
```
image_path = tf.keras.utils.get_file('YellowLabradorLooking_new.jpg', 'https://
 ↪storage.googleapis.com/download.tensorflow.org/example_images/
 ↪YellowLabradorLooking_new.jpg')
image_raw = tf.io.read_file(image_path)
image = tf.image.decode_image(image_raw)

image = preprocess(image)
image_probs = pretrained_model.predict(image)
```

```
1/1 [==============================] - 1s 896ms/step
```

Let's have a look at the image.

[5]:
```
top5 = get_imagenet_label(image_probs)
tick_names = [x[1] for x in top5]
print(tick_names)
probs = [x[2] for x in top5]
plt.figure(figsize=(9, 3))
plt.subplot(121)
plt.imshow(image[0] * 0.5 + 0.5)  # To change [-1, 1] to [0,1]
plt.title('image')
ax = plt.gca()
ax.axis('off')

plt.subplot(122)
tick_names = [x[1] for x in reversed(top5)]
probs = [x[2] for x in reversed(top5)]
plt.barh(tick_names, probs)
plt.yticks(rotation=25)
ax = plt.gca()
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)
plt.tight_layout()
plt.show()
```

```
['Labrador_retriever', 'Eskimo_dog', 'Ibizan_hound', 'golden_retriever',
'Saluki']
```



4 Create the adversarial image

4.1 Implementing fast gradient sign method

The first step is to create perturbations which will be used to distort the original image resulting in an adversarial image. As mentioned, for this task, the gradients are taken with respect to the image.

TASK 1: Implement `create_adversarial_pattern()`. You will need to carry out 3 steps:

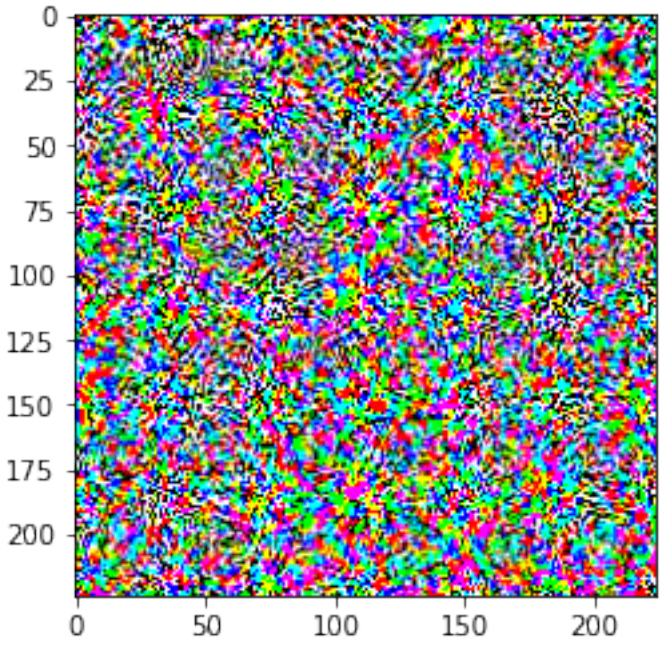
1. Create a loss object using `loss_object` using two arguments: `pretrained_model` and `input_label`.
2. Get the gradients using `tf.gradients` of the `loss` w.r.t to the `input_image`.
3. Get the sign of the gradients to create the perturbation using `tf.sign`.

[6]:
```
loss_object = tf.keras.losses.CategoricalCrossentropy()

def create_adversarial_pattern(input_image, input_label):
    ## START YOU CODE HERE (3 lines)
    with tf.GradientTape() as tape:
        tape.watch(input_image)
        prediction = pretrained_model(input_image)
        loss = loss_object(input_label, prediction)

    # Get the gradients of the loss w.r.t to the input image.
    gradient = tape.gradient(loss, input_image)
    # Get the sign of the gradients to create the perturbation
    signed_grad = tf.sign(gradient)
    # END
    return signed_grad
```

The resulting perturbations can also be visualised.

[7]:
```
# Get the input label of the image.
labrador_retriever_index = 208
```

```
label = tf.one_hot(labrador_retriever_index, image_probs.shape[-1])
label = tf.reshape(label, (1, image_probs.shape[-1]))

perturbations = create_adversarial_pattern(image, label)
plt.imshow(perturbations[0] * 0.5 + 0.5);  # To change [-1, 1] to [0,1]
```



4.2 Fool the AI system

Let's try this out for different values of epsilon and observe the resultant image. You'll notice that as the value of epsilon is increased, it becomes easier to fool the network, however, this comes as a trade-off which results in the perturbations becoming more identifiable.

[57]:
```
def display_images(image, description):
    top5 = get_imagenet_label(pretrained_model.predict(image))
    top5 = list(reversed(top5))
    plt.figure(figsize=(9, 3))
    plt.subplot(121)
    plt.imshow(image[0]*0.5+0.5)
    plt.title(description)
    plt.gca().axis('off')
    plt.subplot(122)
    tick_names = [x[1] for x in top5]
    probs = [x[2] for x in top5]
    plt.barh(tick_names, probs)
    plt.yticks(rotation=25)
    ax = plt.gca()
```

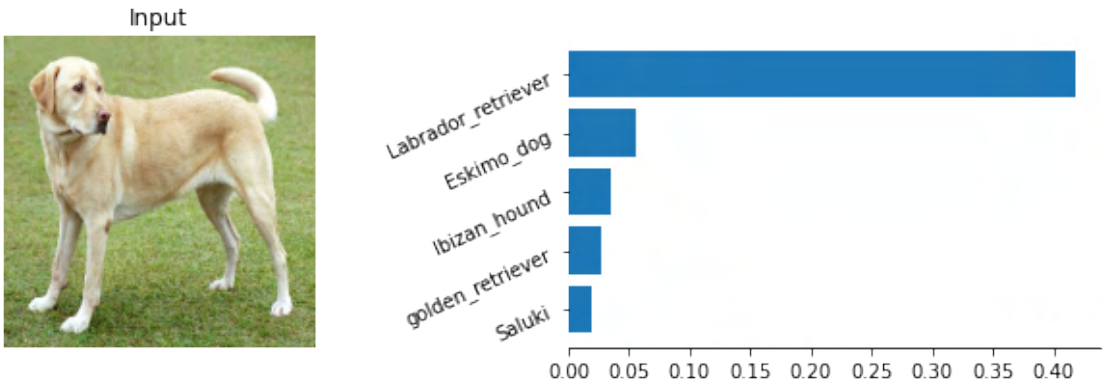
```
    ax.spines['top'].set_visible(False)
    ax.spines['right'].set_visible(False)
    ax.spines['left'].set_visible(False)
    plt.tight_layout()
    plt.show()
```

TASK 2: Generate adverse image using different values for ϵ:

- adv_x = input_image + ϵ * perturbations

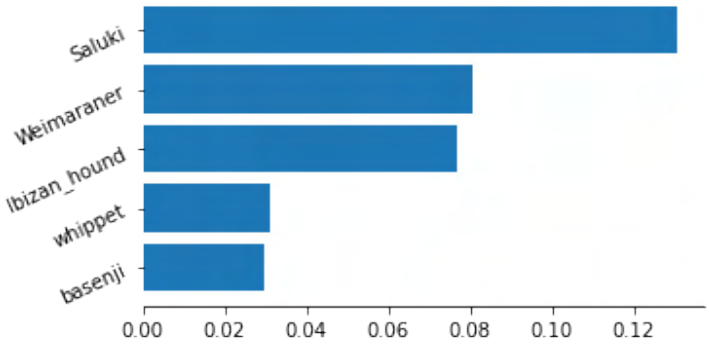
[9]:
```
epsilons = [0, 0.01, 0.1, 0.15, 0.3]
descriptions = [('Epsilon = {:0.3f}'.format(eps) if eps else 'Input')
                for eps in epsilons]

for i, eps in enumerate(epsilons):
    ## START YOU CODE HERE
    adv_x = image + eps * perturbations
    ## End
    adv_x = tf.clip_by_value(adv_x, -1, 1)
    display_images(adv_x, descriptions[i])
```

```
1/1 [==============================] - 0s 47ms/step
```



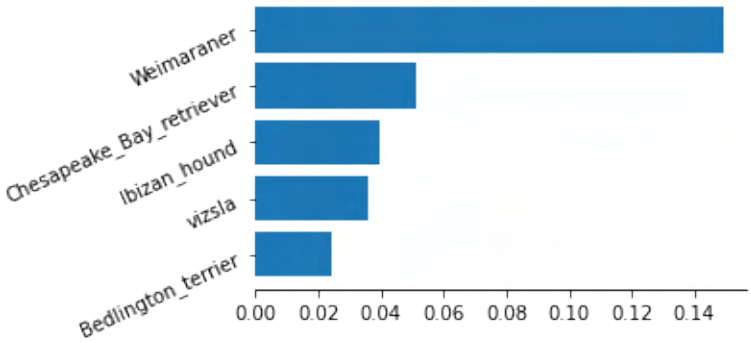

```
1/1 [==============================] - 0s 46ms/step
```

Epsilon = 0.010



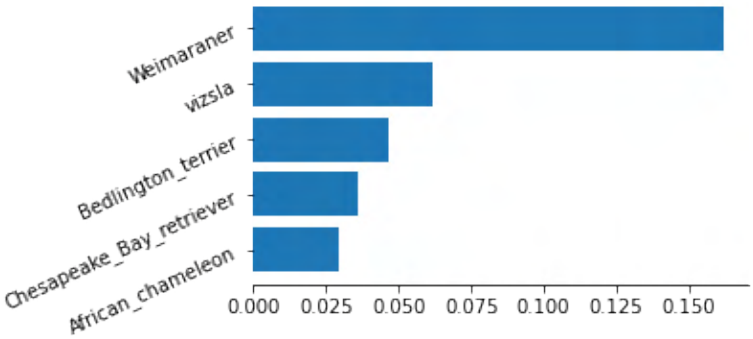

```
1/1 [==============================] - 0s 39ms/step
```

Epsilon = 0.100

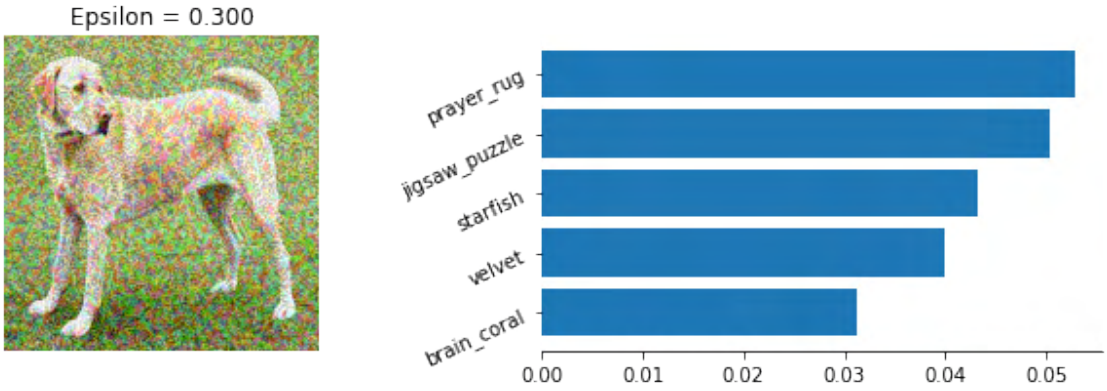



```
1/1 [==============================] - 0s 38ms/step
```

Epsilon = 0.150



```
1/1 [==============================] - 0s 35ms/step
```



TASK 3: What do you observe?

Both the original image and unmodified image are classified by the system. Labrador_retriever is the highest probable class at 0.30 and by a margin of approximately 0.25 to the next classes. The next 4 highest probable classifications are Ibizian hound, saluki, Chesapeake bay retriever and Weimaraner.

As the level of noise increases and distortions to the input image, the correct classification of "Labrador_retriver" is no longer predicted within the top 5. The top 5 probable classes begin to list other (incorrect) dog-breeds, other animal species and at the last iteration of Epsilon at 0.300, seemingly random objects such as "prayer_rug" and "pillow".

To summarise the observations at each iteration of distortion (epsilon):

- 0.000: Model correctly predicts "Labrador_retriever" with approx 0.40 probability
- 0.010: Model incorrectly predicts "Saluki" with approx 0.14 probability
- 0.100: Model incorrectly predicts "Weimaraner" with approx 0.15 probability
- 0.150: Model incorrectly predicts "Weimaraner" with approx 0.1750 probability
- 0.300: Model incorrectly predicts "Prayer_rug" with approx 0.6 probability

Unlike the human eye perspective, when the distortion is introduced, it is possible to identify the input image is of a Labrador_Retriever, however model, the small amount of distortion is considered a significant, which disrupts the model.

---

5 Part 2

Here, you are required to process adversarial attacks using FGSM for a small subset of ImageNet Dataset. We prepared 100 images from different categories (in `./input_dir/`), and the labels are encoded in `./input_dir/clean_image.list`.

For evaluation, each adversarial image generated by the attack model will be fed to an evaluation model, and we will calculate the successful rate of adversarial attacks. **The adversarial images that can fool the evaluation model with ϵ = 0.01 will be considered as a success**.

Task 4: Goal

With the previous FGSM example, you are required to implement an FGSM attack against all examples and calculate the success rate. Also, display the original image with the attacked image as well as the predicted class for each image.

[13]:
```
import numpy as np
import os
```

[40]:
```
epsilon = 0.01  #Set the distortion amount in an attempt to confuse the model
```

[41]:
```
f = open('input_dir/clean_image.list', "r")
clean_image_list = f.read().splitlines()
columns = ['FileName', 'label']
clean_image_list_data = [[str(x) for x in e.split()] for e in clean_image_list]
clean_image_list_df = pd.DataFrame(clean_image_list_data, columns=columns)
print(clean_image_list_df)
```

```
          FileName label
0   n02708093.JPEG   409
1   n03000134.JPEG   489
2   n03384352.JPEG   561
3   n03777754.JPEG   662
4   n03721384.JPEG   642
..             ...   ...
95  n02109525.JPEG   247
96  n07697313.JPEG   933
97  n02489166.JPEG   376
98  n02794156.JPEG   426
99  n04131690.JPEG   773

[100 rows x 2 columns]
```

[42]:
```
import glob
image_files_collection = glob.glob('input_dir/*.JPEG')
image_files_collection = np.char.strip(image_files_collection,␣
 ↪chars='input_dir')
image_files_collection = np.char.strip(image_files_collection, chars='/')
image_files_collection
```

[42]:
```
array(['\\n01443537.JPEG', '\\n01530575.JPEG', '\\n01537544.JPEG',
       '\\n01664065.JPEG', '\\n01688243.JPEG', '\\n01773157.JPEG',
       '\\n01817953.JPEG', '\\n01829413.JPEG', '\\n01877812.JPEG',
       '\\n01955084.JPEG', '\\n02012849.JPEG', '\\n02013706.JPEG',
       '\\n02018207.JPEG', '\\n02033041.JPEG', '\\n02071294.JPEG',
```

```
       '\\n02086240.JPEG', '\\n02093256.JPEG', '\\n02093647.JPEG',
       '\\n02095889.JPEG', '\\n02099267.JPEG', '\\n02107908.JPEG',
       '\\n02109525.JPEG', '\\n02110185.JPEG', '\\n02110341.JPEG',
       '\\n02129604.JPEG', '\\n02177972.JPEG', '\\n02229544.JPEG',
       '\\n02259212.JPEG', '\\n02326432.JPEG', '\\n02422106.JPEG',
       '\\n02443114.JPEG', '\\n02444819.JPEG', '\\n02480495.JPEG',
       '\\n02489166.JPEG', '\\n02655020.JPEG', '\\n02667093.JPEG',
       '\\n02704792.JPEG', '\\n02708093.JPEG', '\\n02769748.JPEG',
       '\\n02777292.JPEG', '\\n02790996.JPEG', '\\n02791270.JPEG',
       '\\n02794156.JPEG', '\\n02814860.JPEG', '\\n02815834.JPEG',
       '\\n02966687.JPEG', '\\n03000134.JPEG', '\\n03016953.JPEG',
       '\\n03042490.JPEG', '\\n03063599.JPEG', '\\n03109150.JPEG',
       '\\n03131574.JPEG', '\\n03160309.JPEG', '\\n03197337.JPEG',
       '\\n03201208.JPEG', '\\n03207941.JPEG', '\\n03216828.JPEG',
       '\\n03379051.JPEG', '\\n03384352.JPEG', '\\n03393912.JPEG',
       '\\n03394916.JPEG', '\\n03424325.JPEG', '\\n03467068.JPEG',
       '\\n03485407.JPEG', '\\n03584254.JPEG', '\\n03673027.JPEG',
       '\\n03721384.JPEG', '\\n03777754.JPEG', '\\n03794056.JPEG',
       '\\n03873416.JPEG', '\\n03956157.JPEG', '\\n03977966.JPEG',
       '\\n04023962.JPEG', '\\n04026417.JPEG', '\\n04049303.JPEG',
       '\\n04065272.JPEG', '\\n04131690.JPEG', '\\n04153751.JPEG',
       '\\n04251144.JPEG', '\\n04311174.JPEG', '\\n04376876.JPEG',
       '\\n04398044.JPEG', '\\n04493381.JPEG', '\\n04525038.JPEG',
       '\\n04536866.JPEG', '\\n04548280.JPEG', '\\n04590129.JPEG',
       '\\n04597913.JPEG', '\\n04606251.JPEG', '\\n07583066.JPEG',
       '\\n07695742.JPEG', '\\n07697313.JPEG', '\\n07717556.JPEG',
       '\\n07754684.JPEG', '\\n07831146.JPEG', '\\n07836838.JPEG',
       '\\n09332890.JPEG', '\\n09399592.JPEG', '\\n12144580.JPEG',
       '\\n12267677.JPEG'], dtype='<U24')
```

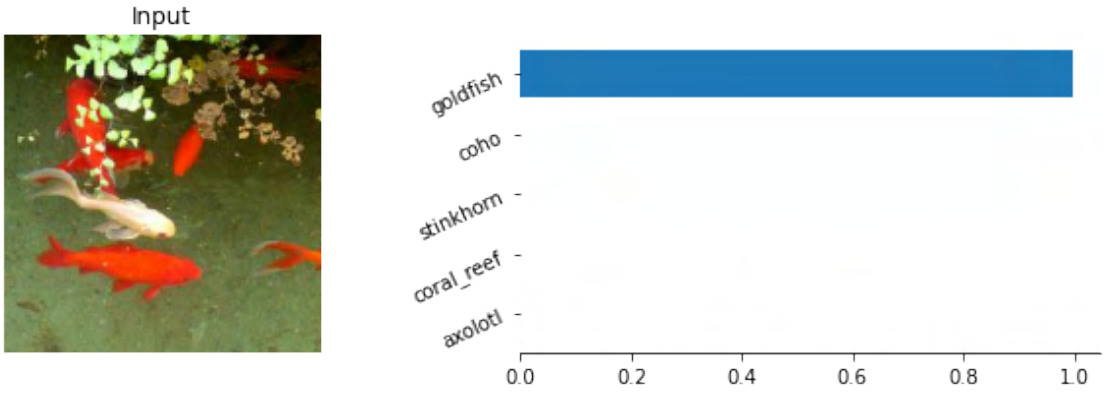
[43]:
```
idx2label = clean_image_list_df['label']
idx2label
```

[43]:
```
0     409
1     489
2     561
3     662
4     642
     ...
95    247
96    933
97    376
98    426
99    773
Name: label, Length: 100, dtype: object
```

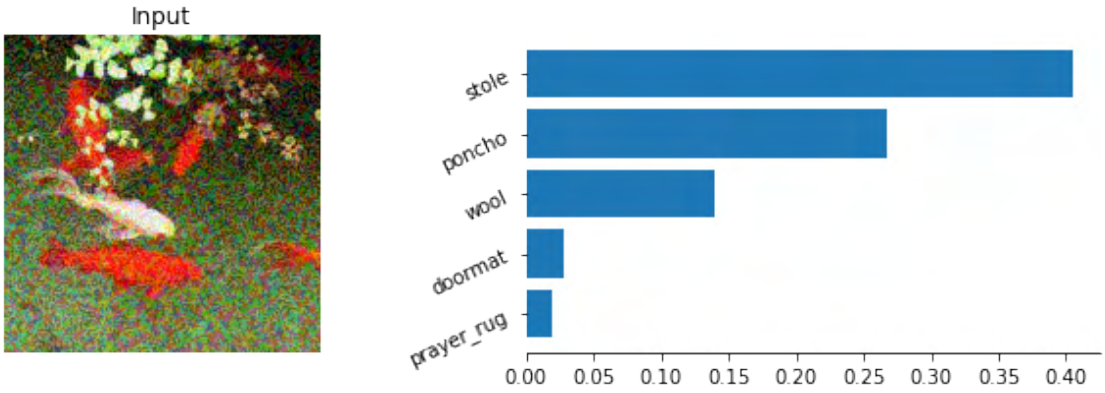
[88]:
```
output = []      #Actuals
output2 = []     #Attack Results
```

```
for idx2label in image_files_collection[:100]:
    print(idx2label)
    path = r"C:\Users\shane\Desktop\input_dir"
    label2 = path + idx2label
    image_raw = tf.io.read_file(label2)
    image = tf.image.decode_image(image_raw)
    image = preprocess(image)
    image_probs = pretrained_model.predict(image)
    top5 = get_imagenet_label(image_probs)
    tick_names = [x[1] for x in top5]
    adv_x = image + eps * perturbations
    adv_x = tf.clip_by_value(adv_x, -1, 1)
    ximage_probs = pretrained_model.predict(adv_x)
    top5x = get_imagenet_label(ximage_probs)
    display_images(image, descriptions[0])
    print(top5[0][1])
    display_images(adv_x, descriptions[0])
    output.append(top5[0][1])
    print(top5x[0][1])
    output2.append(top5x[0][1])
    print()
    print('Loading Next Image')
```

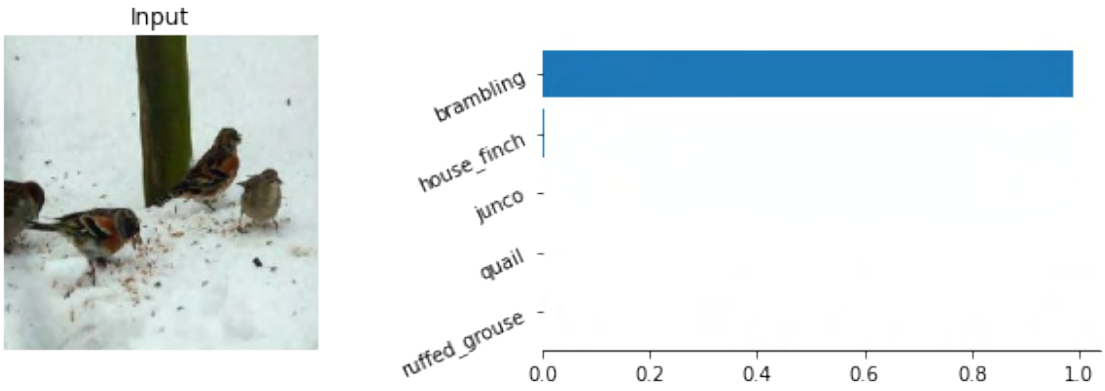
```
\n01443537.JPEG
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 50ms/step
```



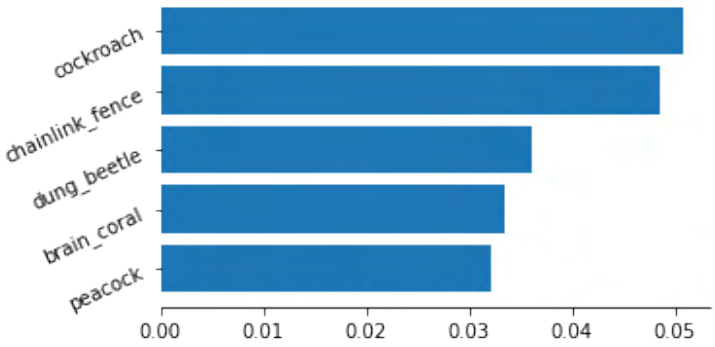

```
goldfish
1/1 [==============================] - 0s 48ms/step
```



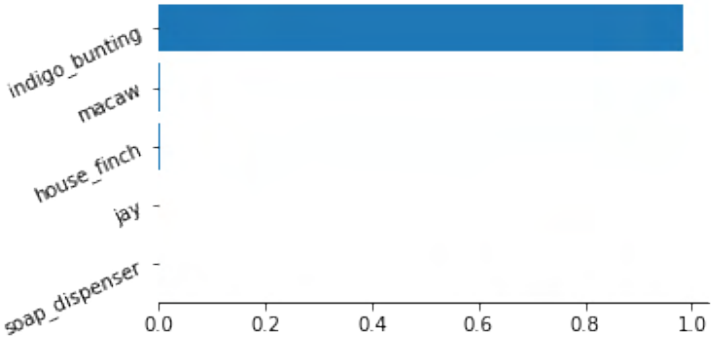

```
stole

Loading Next Image
\n01530575.JPEG
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 46ms/step
```



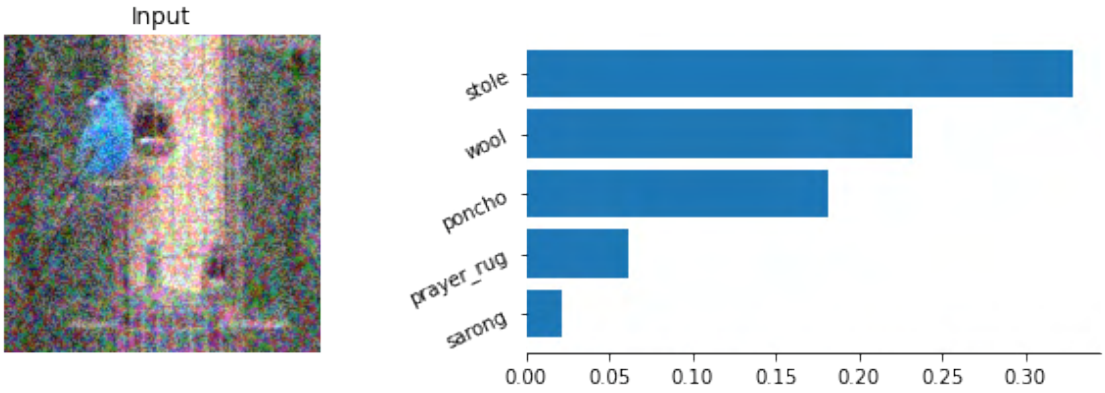

```
brambling
1/1 [==============================] - 0s 63ms/step
```



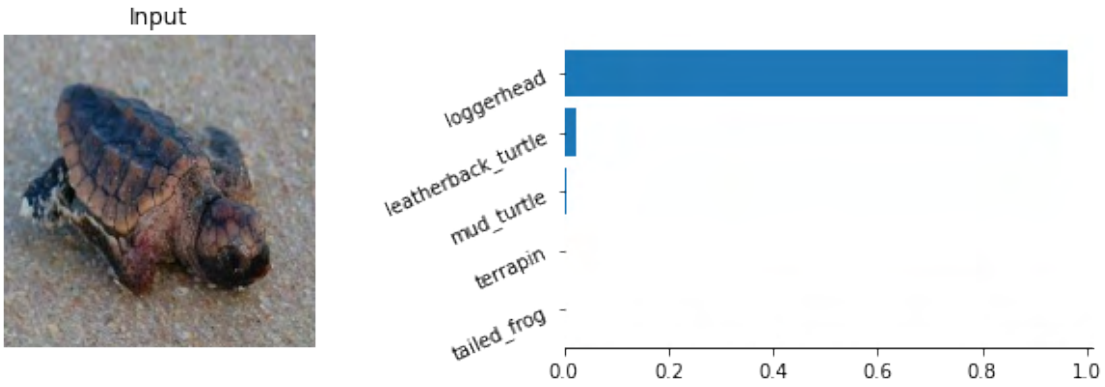

```
cockroach

Loading Next Image
\n01537544.JPEG
1/1 [==============================] - 0s 60ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 38ms/step
```



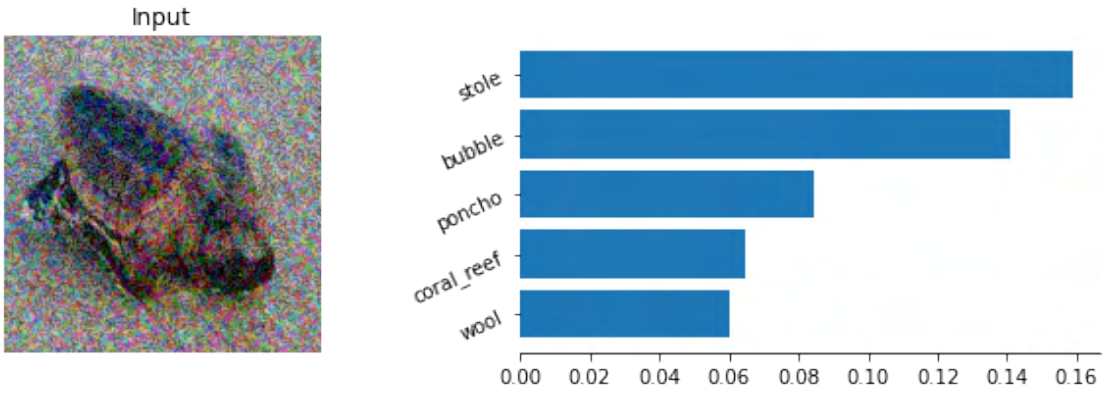

```
indigo_bunting
1/1 [==============================] - 0s 62ms/step
```



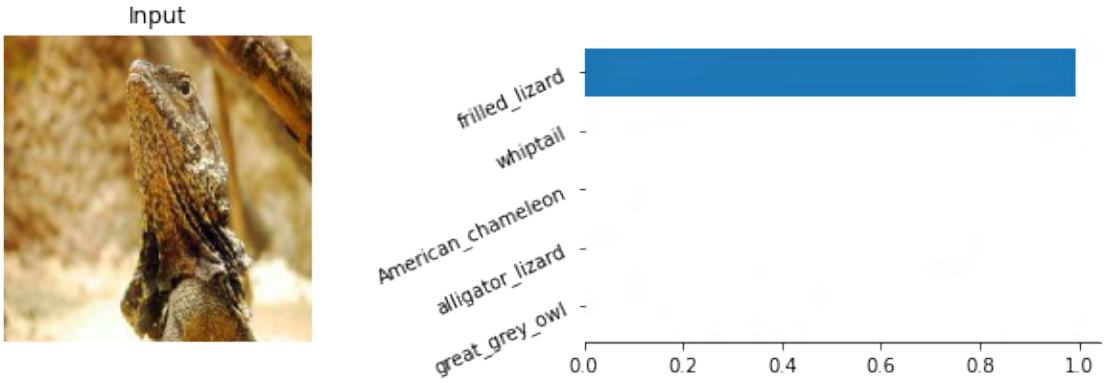

```
stole

Loading Next Image
\n01664065.JPEG
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 41ms/step
```



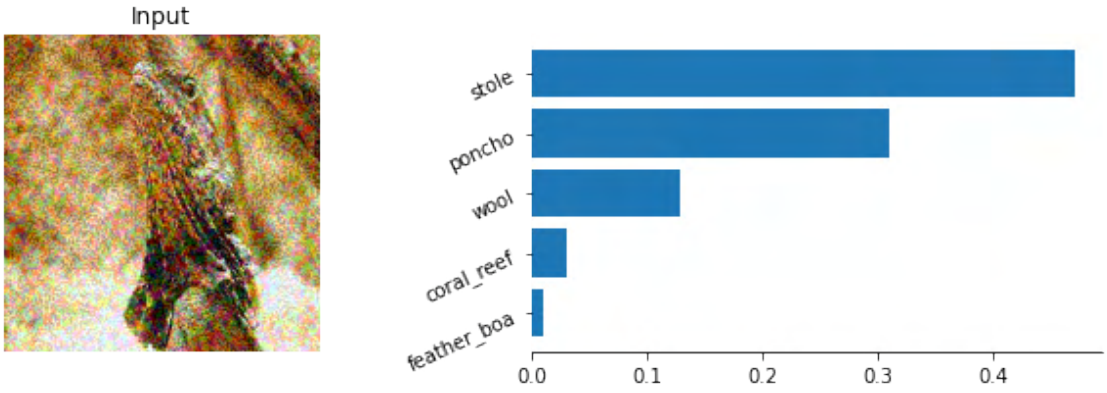

```
loggerhead
1/1 [==============================] - 0s 48ms/step
```



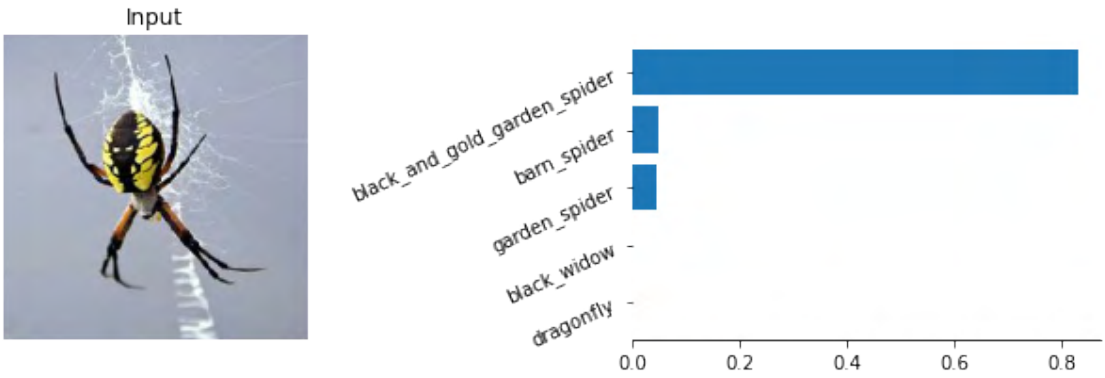

```
stole

Loading Next Image
\n01688243.JPEG
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 42ms/step
```



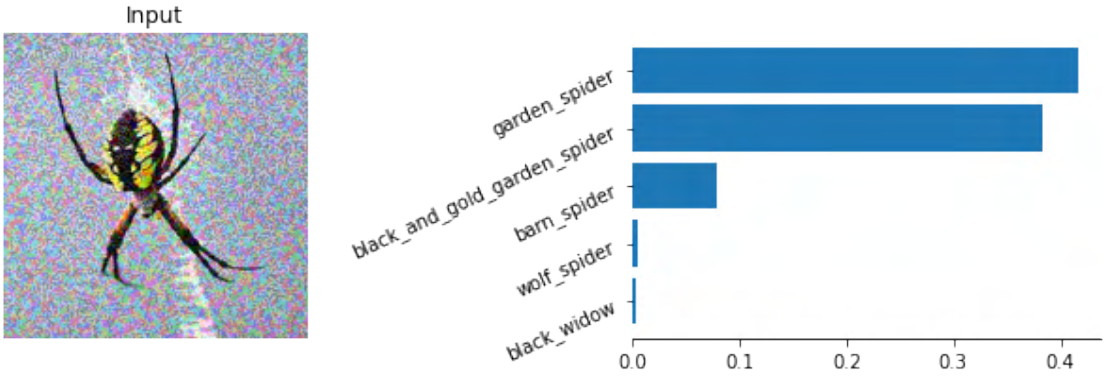

```
frilled_lizard
1/1 [==============================] - 0s 51ms/step
```



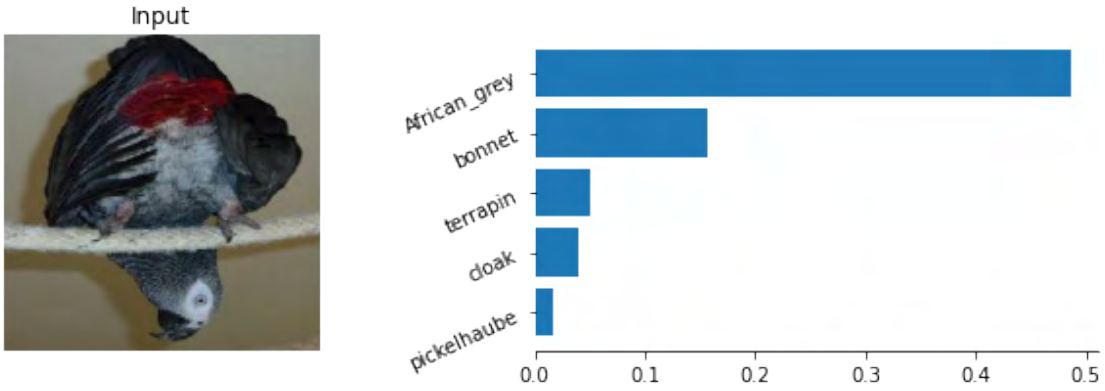

```
stole

Loading Next Image
\n01773157.JPEG
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 39ms/step
```



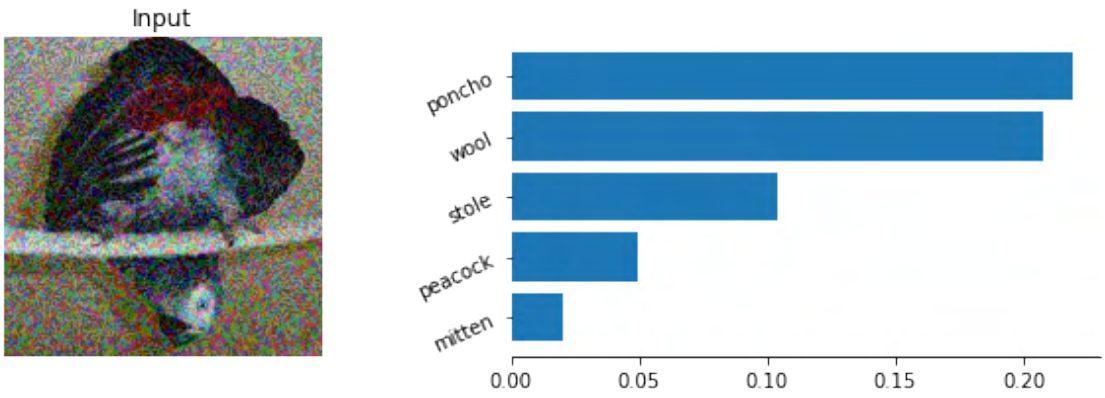

```
black_and_gold_garden_spider
1/1 [==============================] - 0s 50ms/step
```



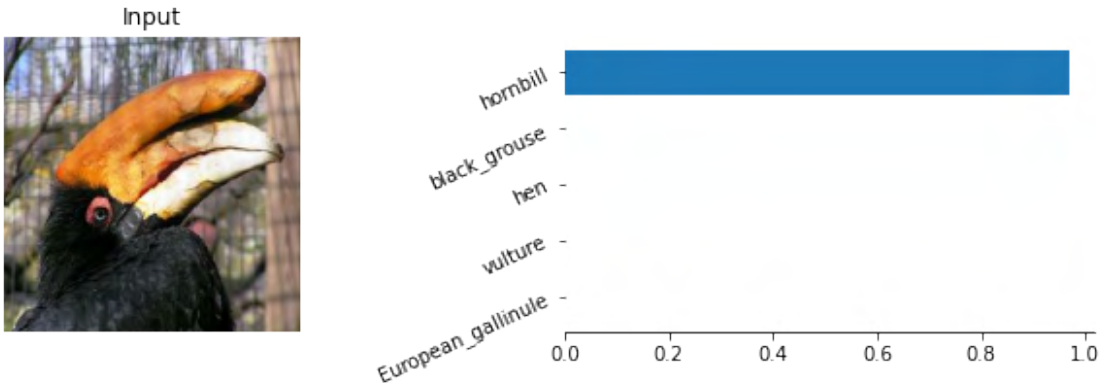

```
garden_spider

Loading Next Image
\n01817953.JPEG
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 56ms/step
```



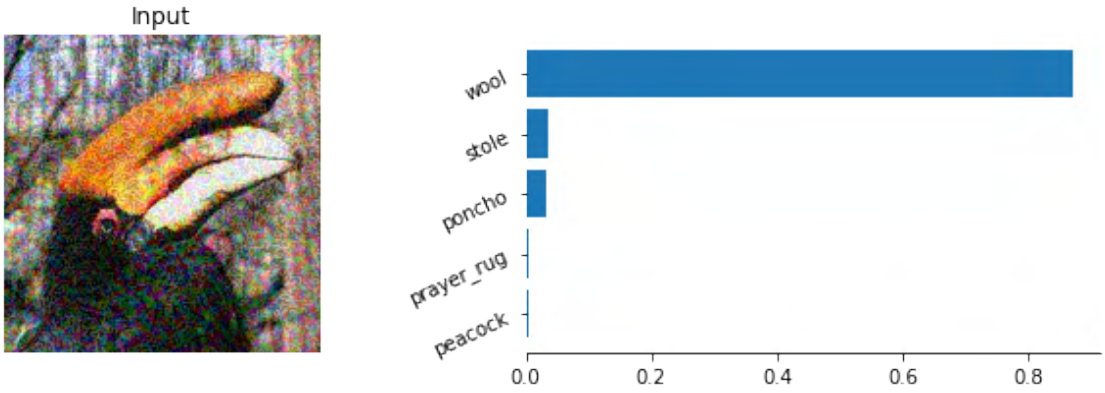

```
African_grey
1/1 [==============================] - 0s 57ms/step
```



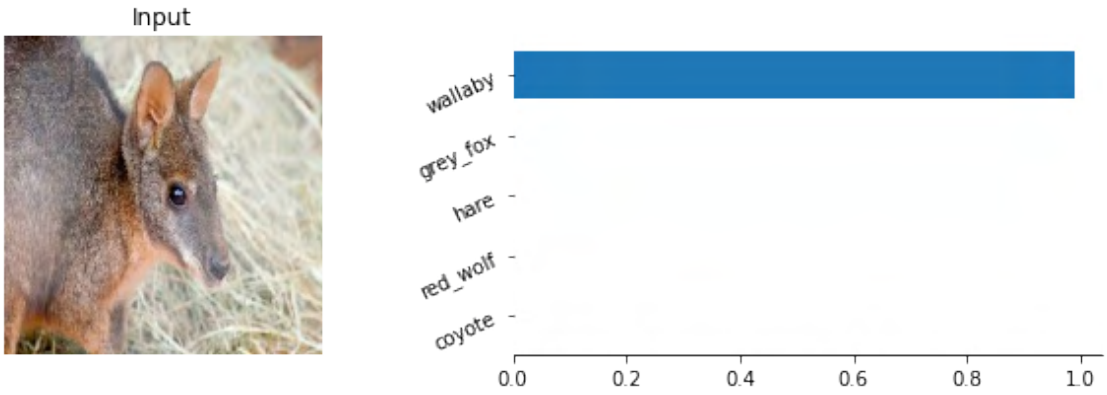

```
poncho

Loading Next Image
\n01829413.JPEG
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 49ms/step
```



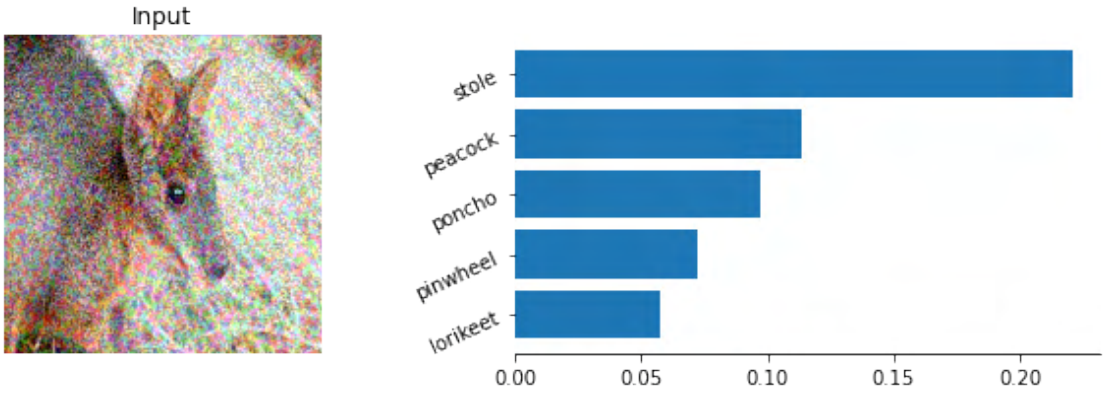

```
hornbill
1/1 [==============================] - 0s 56ms/step
```



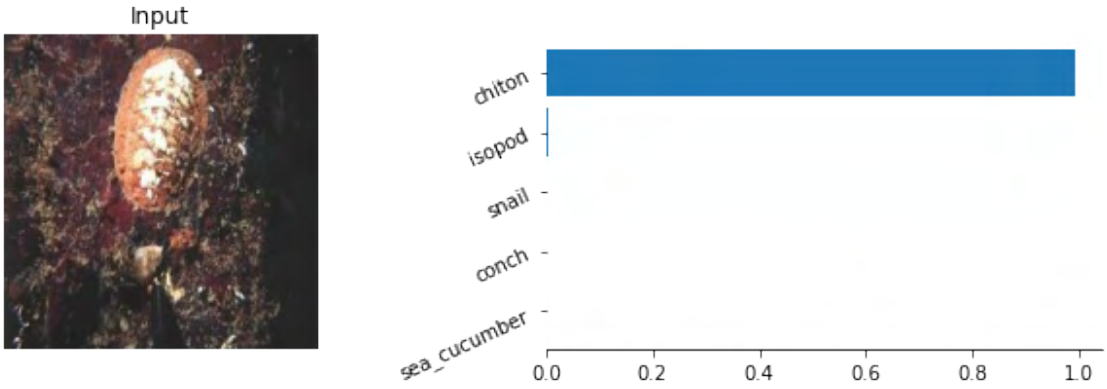

```
wool

Loading Next Image
\n01877812.JPEG
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 51ms/step
```



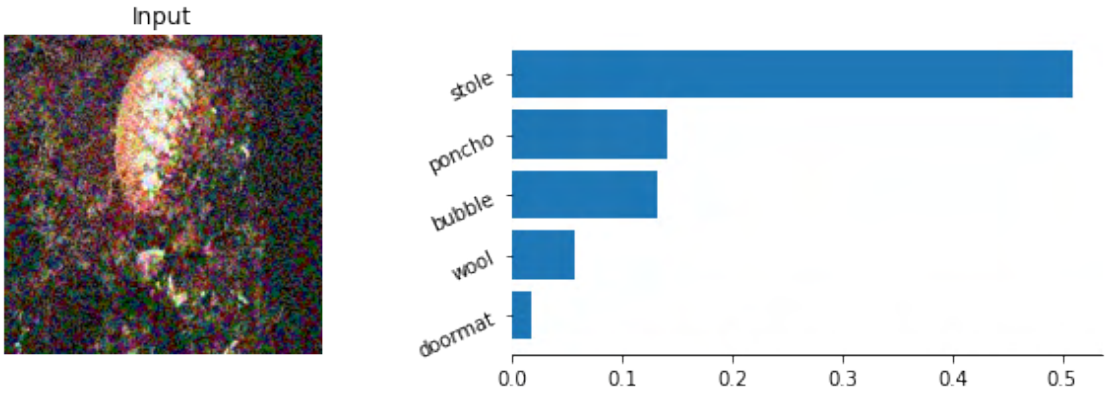

```
wallaby
1/1 [==============================] - 0s 46ms/step
```



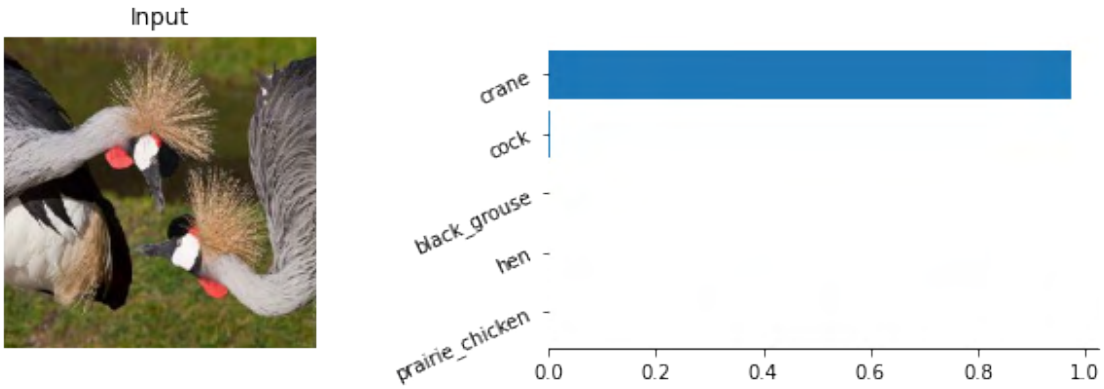

```
stole

Loading Next Image
\n01955084.JPEG
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 55ms/step
```



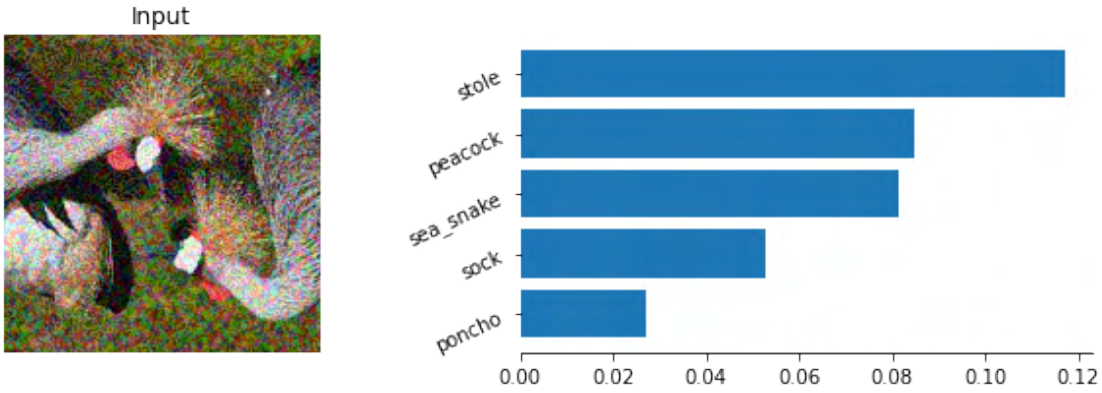

```
chiton
1/1 [==============================] - 0s 58ms/step
```



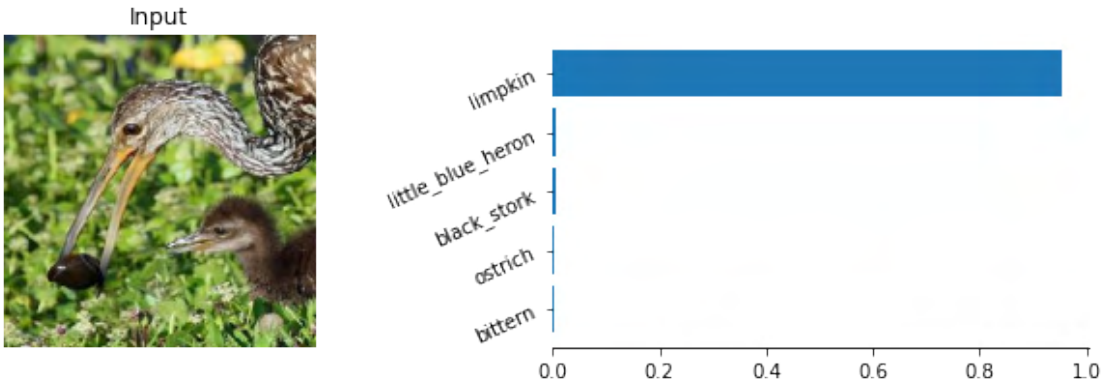

```
stole

Loading Next Image
\n02012849.JPEG
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 45ms/step
1/1 [==============================] - 0s 44ms/step
```



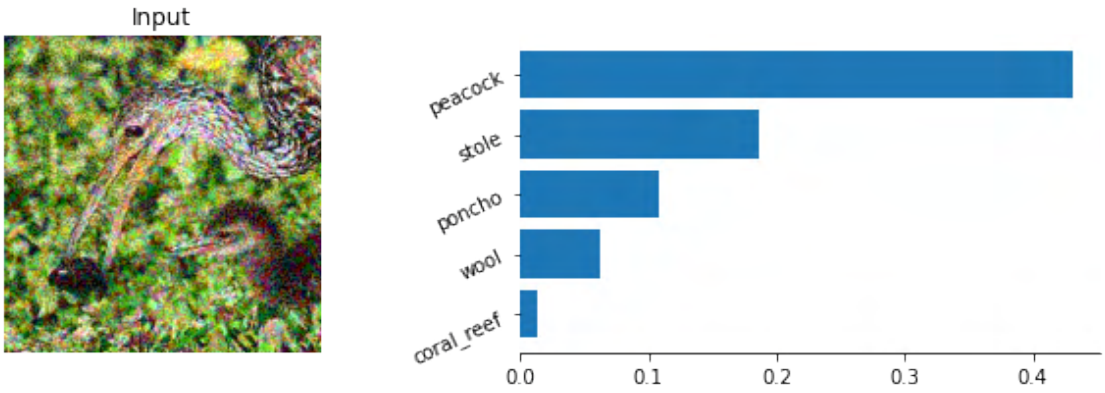

```
crane
1/1 [==============================] - 0s 58ms/step
```



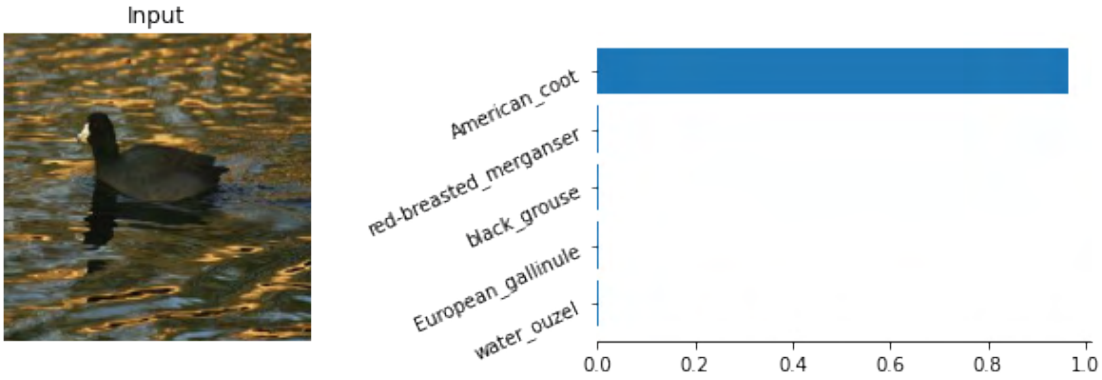

```
stole

Loading Next Image
\n02013706.JPEG
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 48ms/step
```



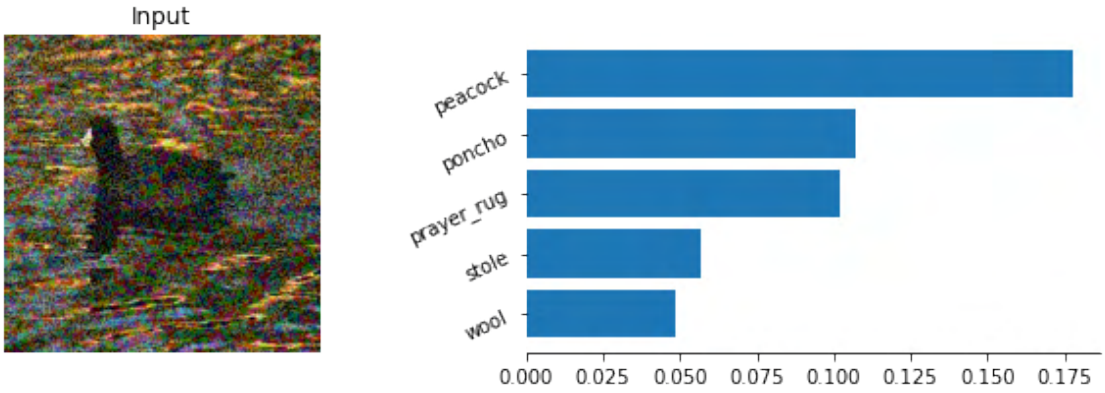

```
limpkin
1/1 [==============================] - 0s 54ms/step
```



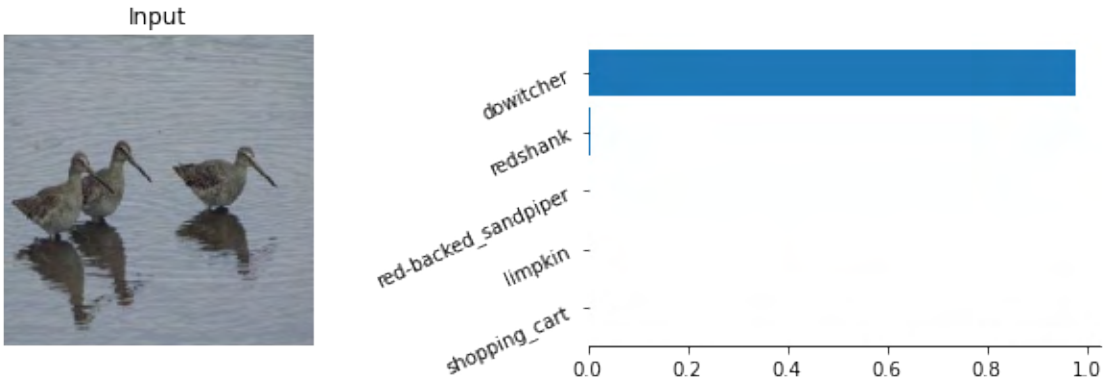

```
peacock

Loading Next Image
\n02018207.JPEG
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 58ms/step
```



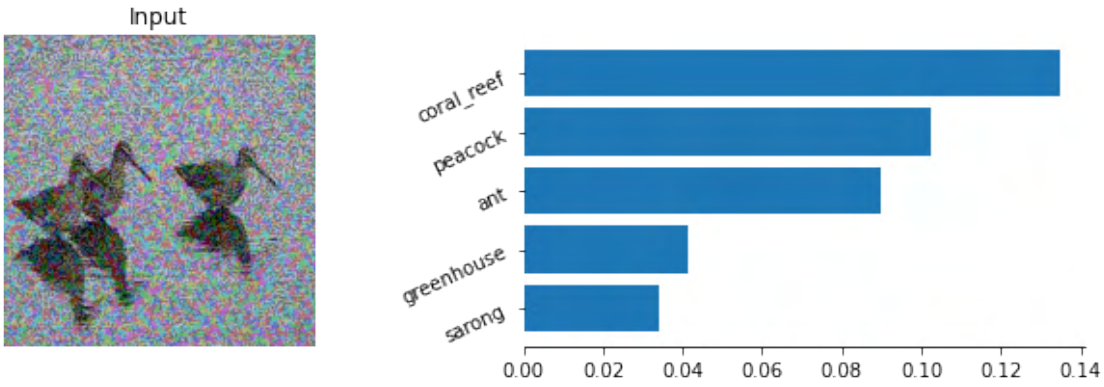

```
American_coot
1/1 [==============================] - 0s 48ms/step
```



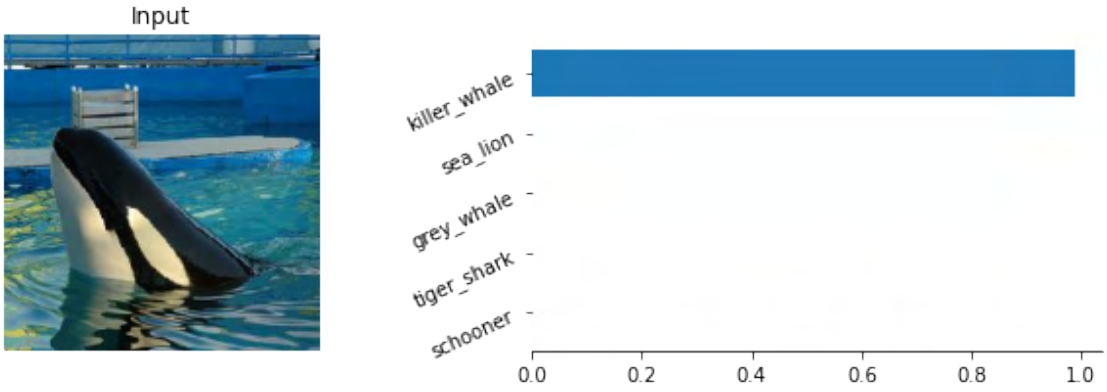

```
peacock

Loading Next Image
\n02033041.JPEG
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 53ms/step
```



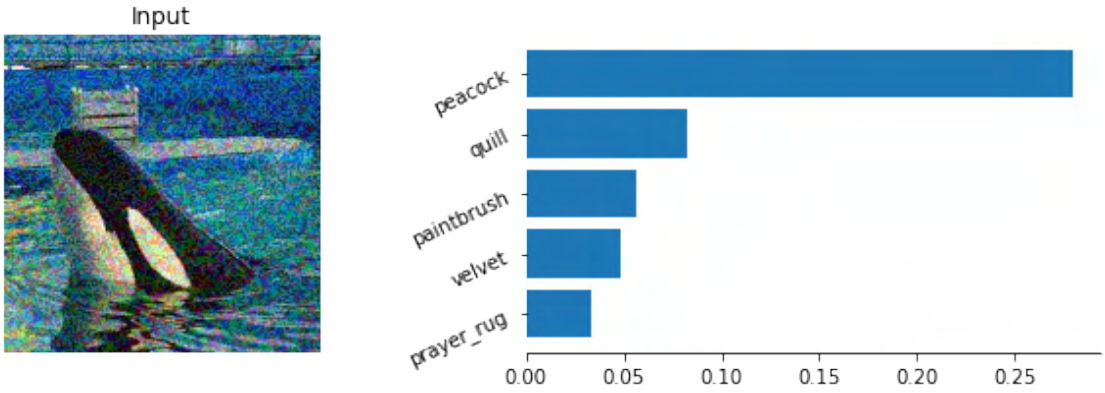

```
dowitcher
1/1 [==============================] - 0s 51ms/step
```



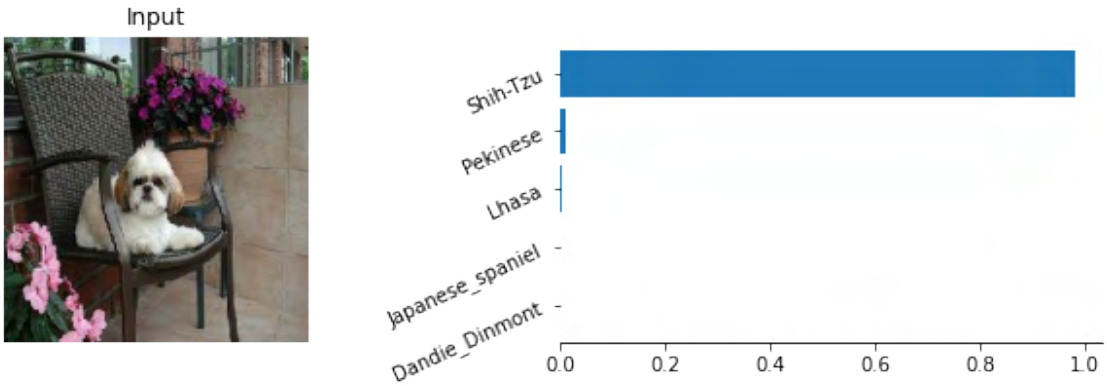

```
coral_reef

Loading Next Image
\n02071294.JPEG
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 42ms/step
```



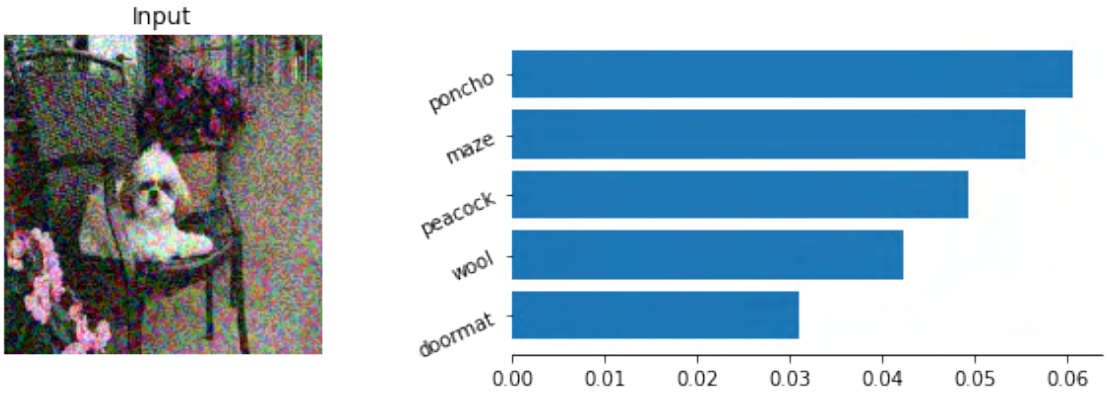

```
killer_whale
1/1 [==============================] - 0s 52ms/step
```



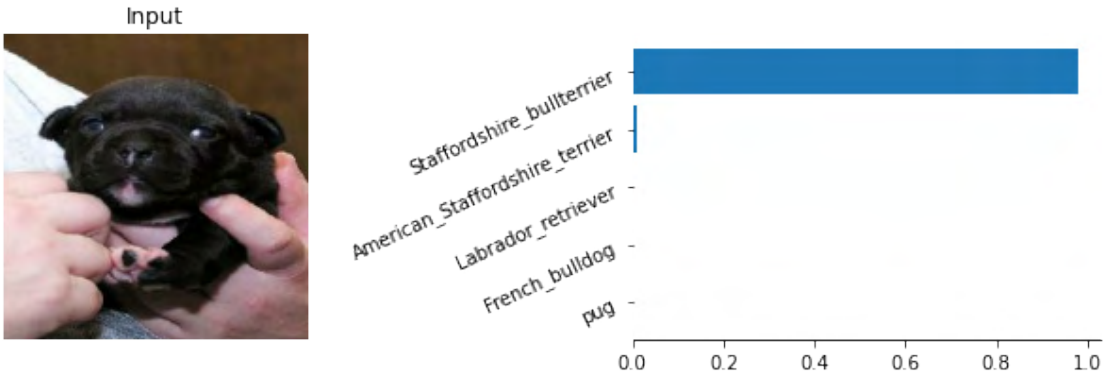

```
peacock

Loading Next Image
\n02086240.JPEG
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 51ms/step
```



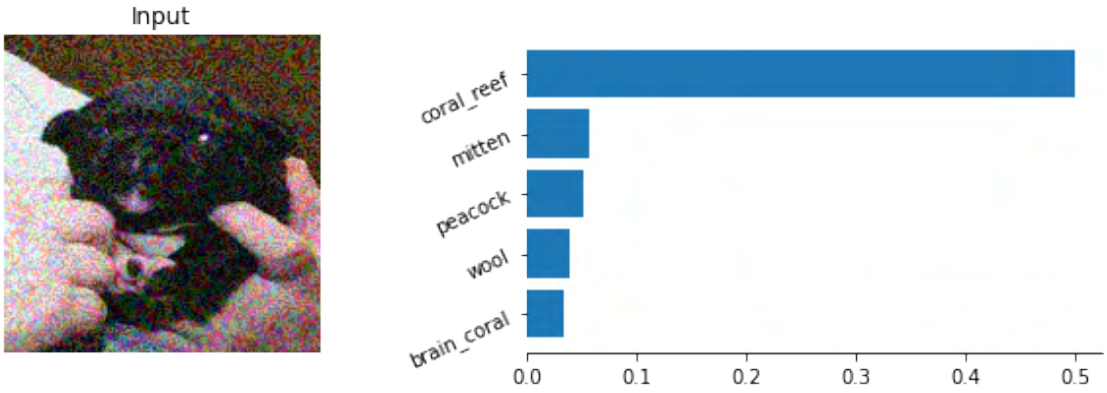

```
Shih-Tzu
1/1 [==============================] - 0s 49ms/step
```



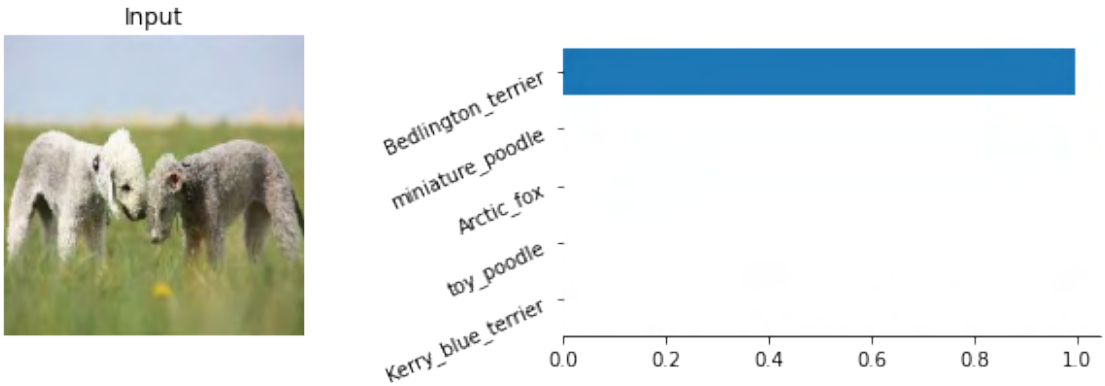

```
poncho

Loading Next Image
\n02093256.JPEG
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 55ms/step
```



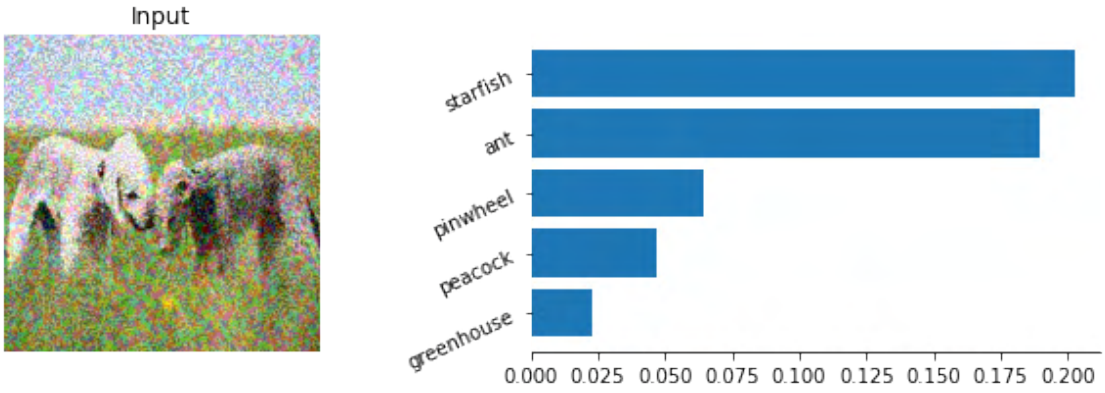

```
Staffordshire_bullterrier
1/1 [==============================] - 0s 37ms/step
```



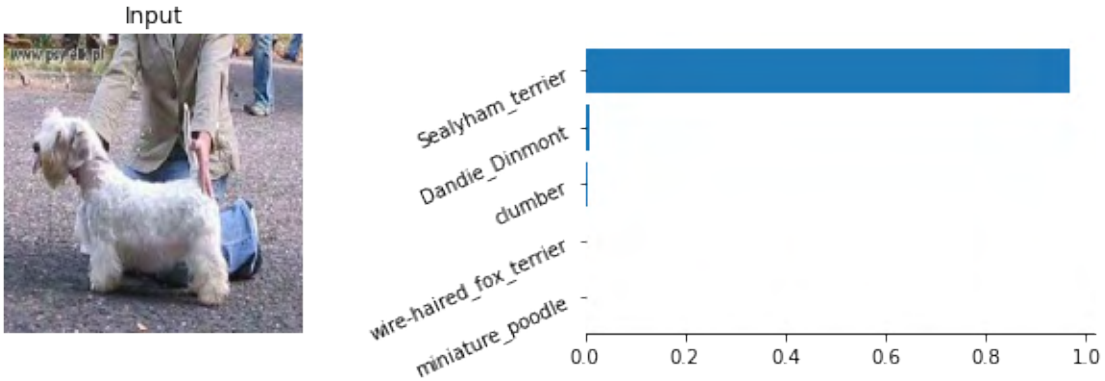

```
coral_reef

Loading Next Image
\n02093647.JPEG
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 60ms/step
1/1 [==============================] - 0s 50ms/step
```



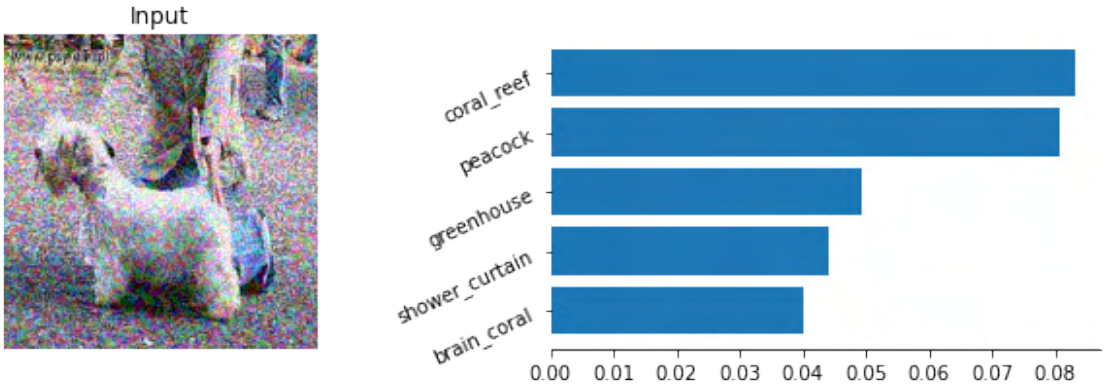

```
Bedlington_terrier
1/1 [==============================] - 0s 58ms/step
```



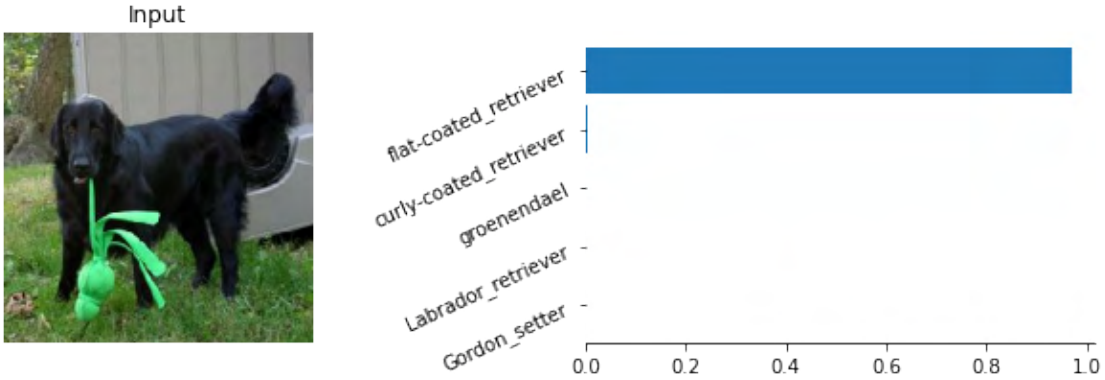

```
starfish

Loading Next Image
\n02095889.JPEG
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 40ms/step
```



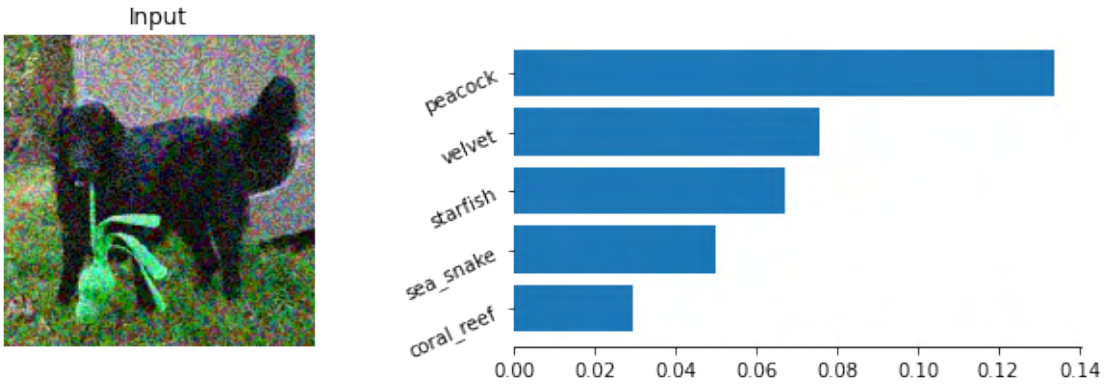

```
Sealyham_terrier
1/1 [==============================] - 0s 62ms/step
```



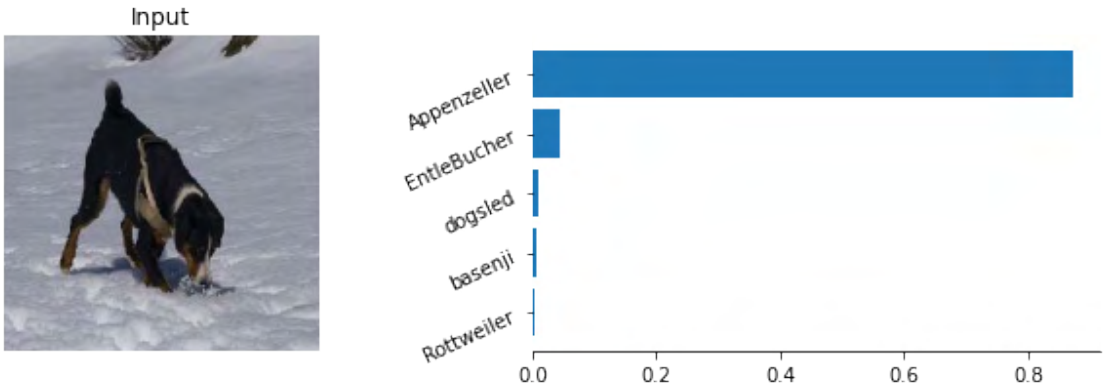

```
coral_reef

Loading Next Image
\n02099267.JPEG
1/1 [==============================] - 0s 60ms/step
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 54ms/step
```



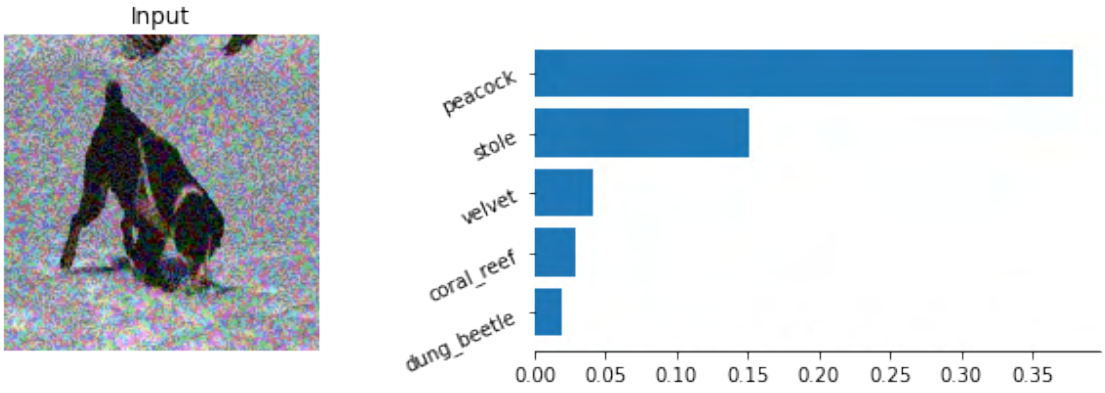

```
flat-coated_retriever
1/1 [==============================] - 0s 57ms/step
```



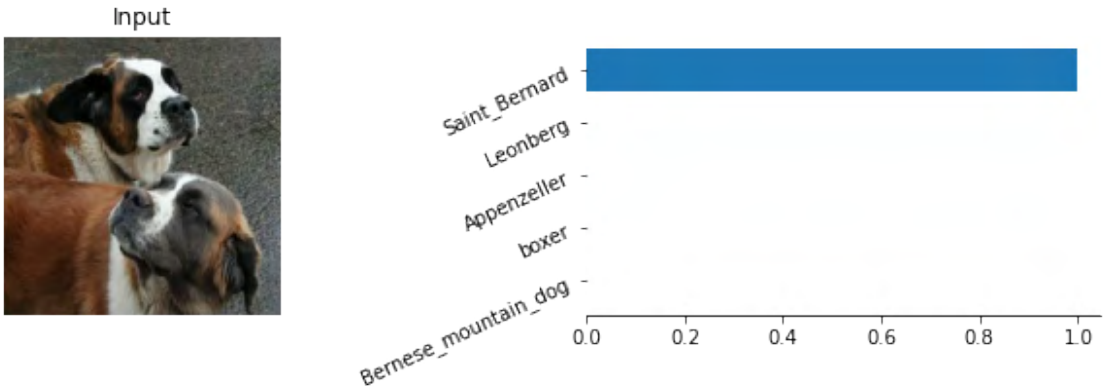

```
peacock

Loading Next Image
\n02107908.JPEG
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 53ms/step
```



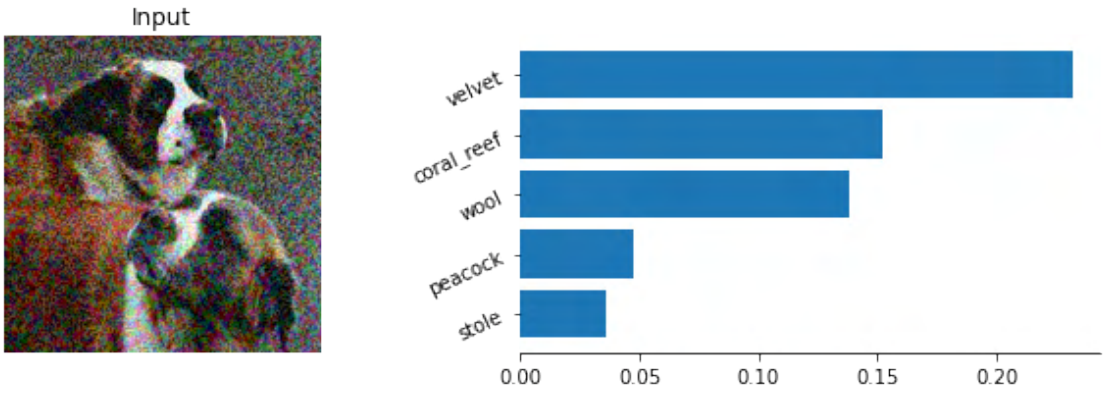

```
Appenzeller
1/1 [==============================] - 0s 52ms/step
```



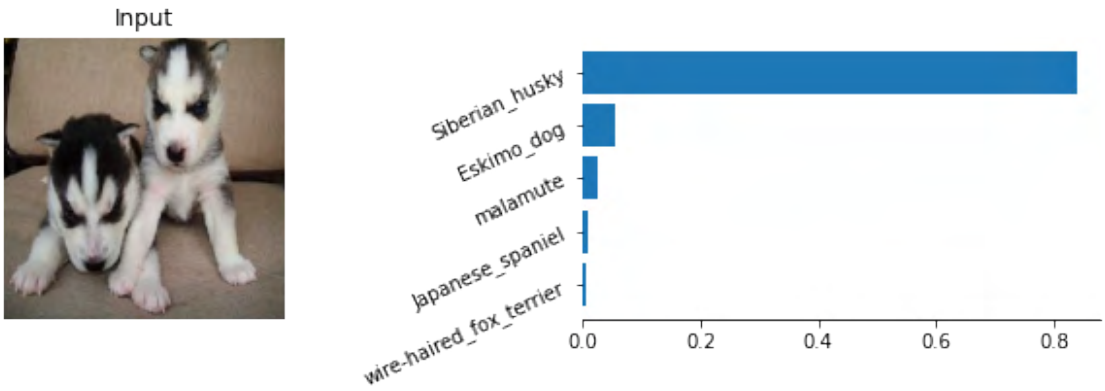

```
peacock

Loading Next Image
\n02109525.JPEG
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 54ms/step
```



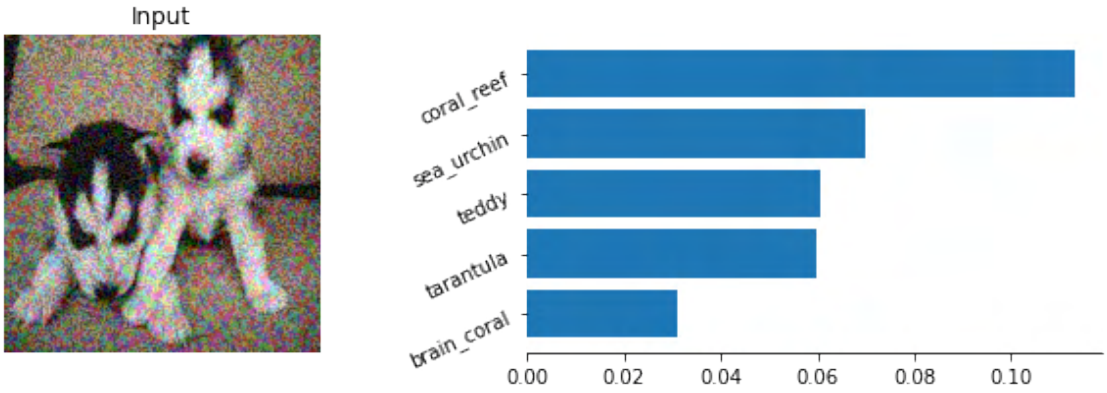

```
Saint_Bernard
1/1 [==============================] - 0s 55ms/step
```



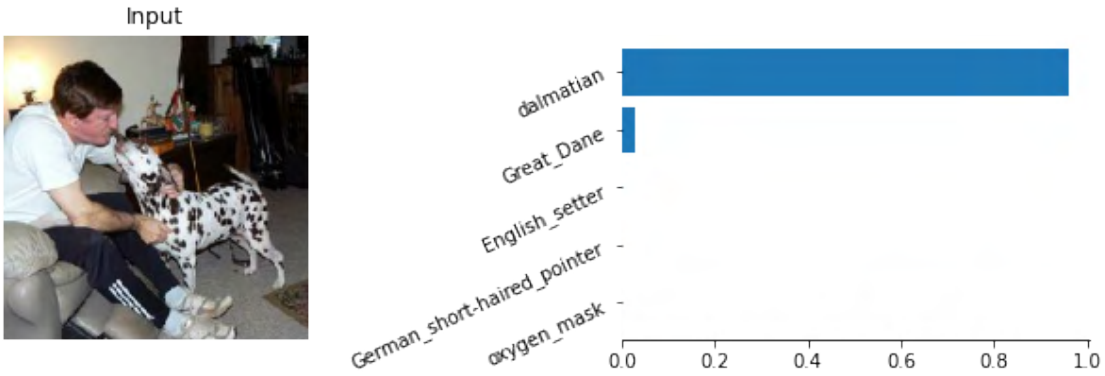

```
velvet

Loading Next Image
\n02110185.JPEG
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 55ms/step
```



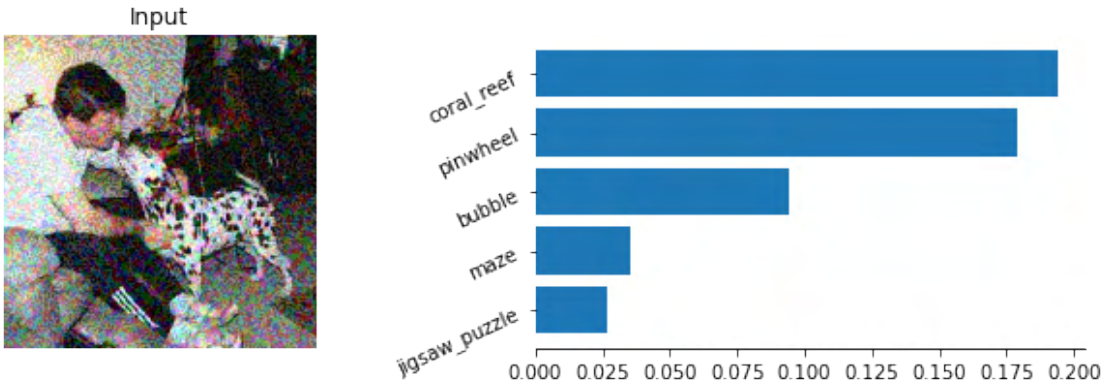

```
Siberian_husky
1/1 [==============================] - 0s 43ms/step
```



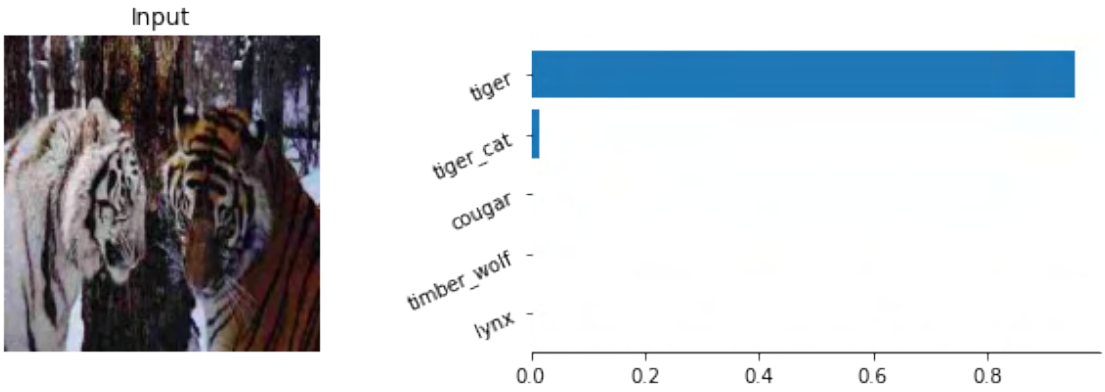

```
coral_reef

Loading Next Image
\n02110341.JPEG
1/1 [==============================] - 0s 61ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 39ms/step
```



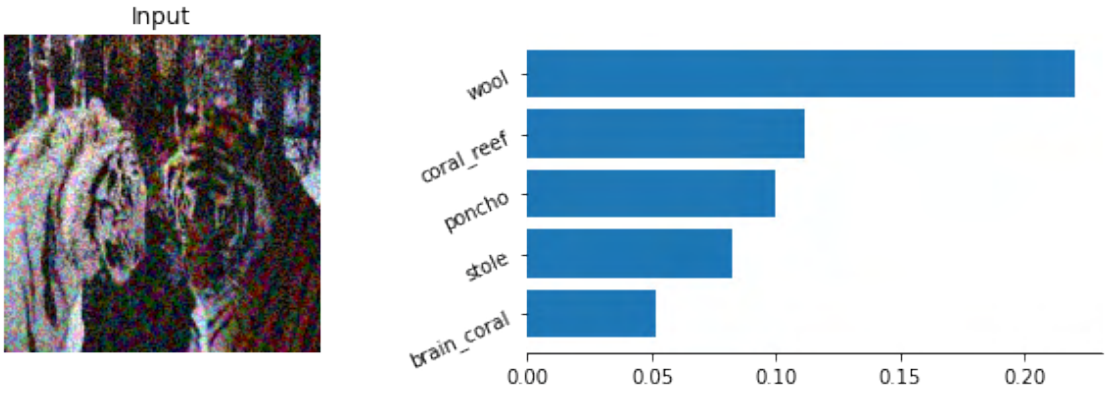

```
dalmatian
1/1 [==============================] - 0s 50ms/step
```



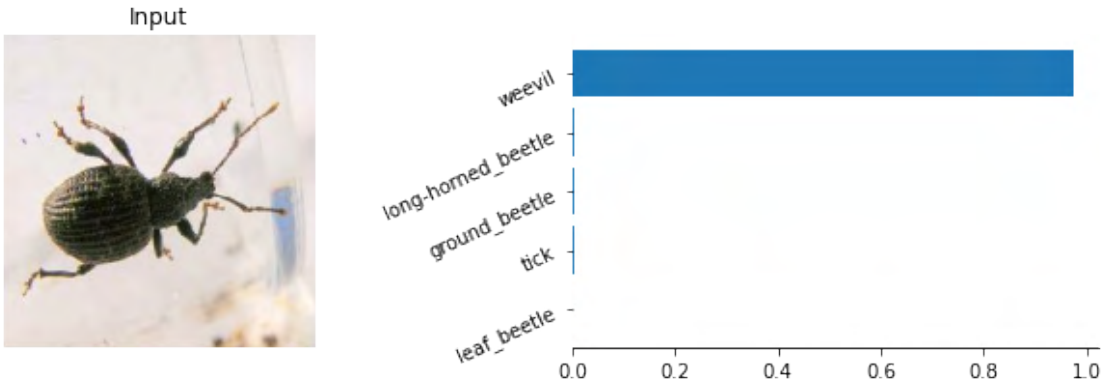

```
coral_reef

Loading Next Image
\n02129604.JPEG
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 61ms/step
```



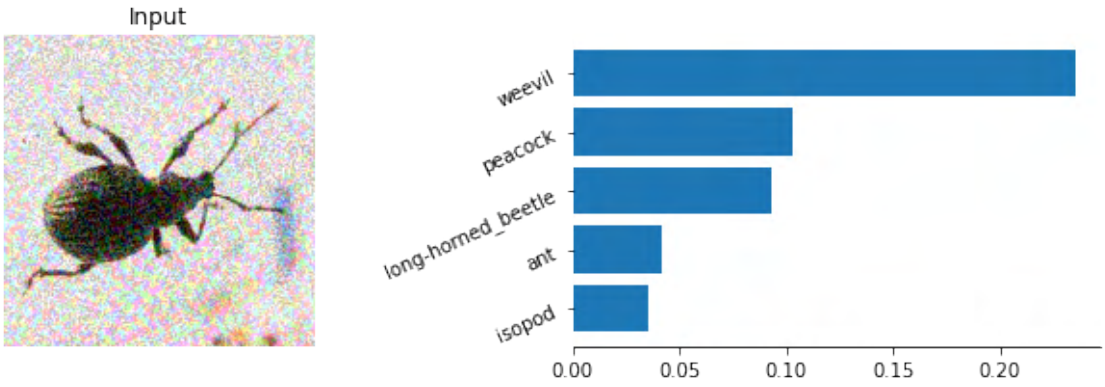

```
tiger
1/1 [==============================] - 0s 50ms/step
```



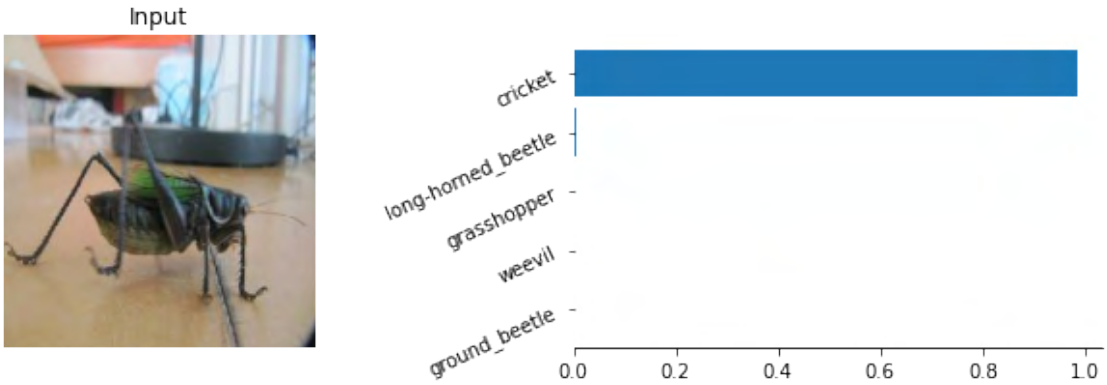

```
wool

Loading Next Image
\n02177972.JPEG
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 42ms/step
```



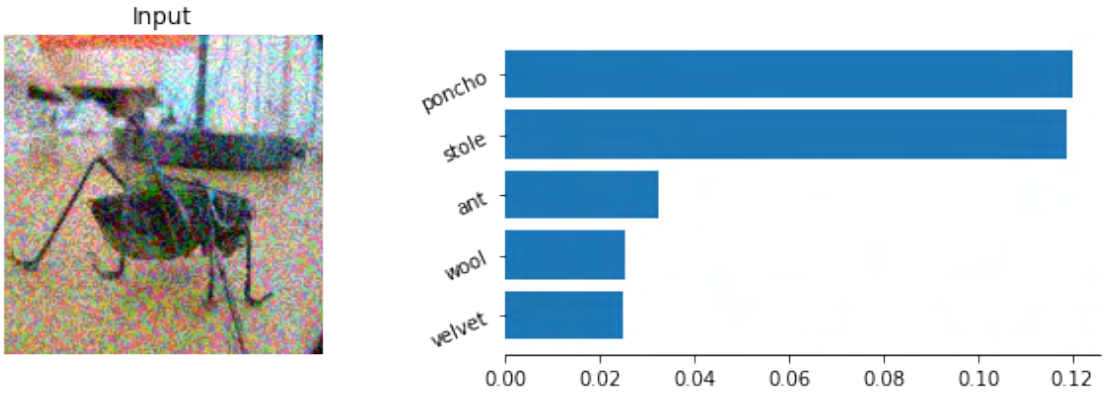

```
weevil
1/1 [==============================] - 0s 49ms/step
```



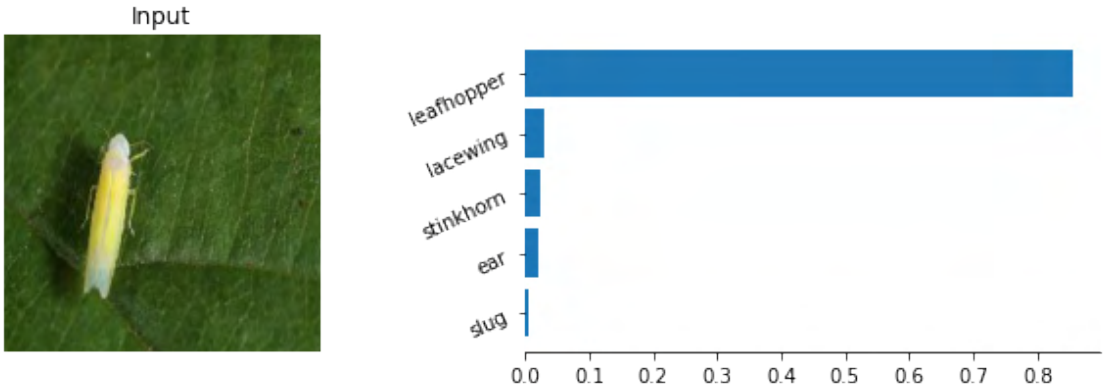

```
weevil

Loading Next Image
\n02229544.JPEG
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 40ms/step
```



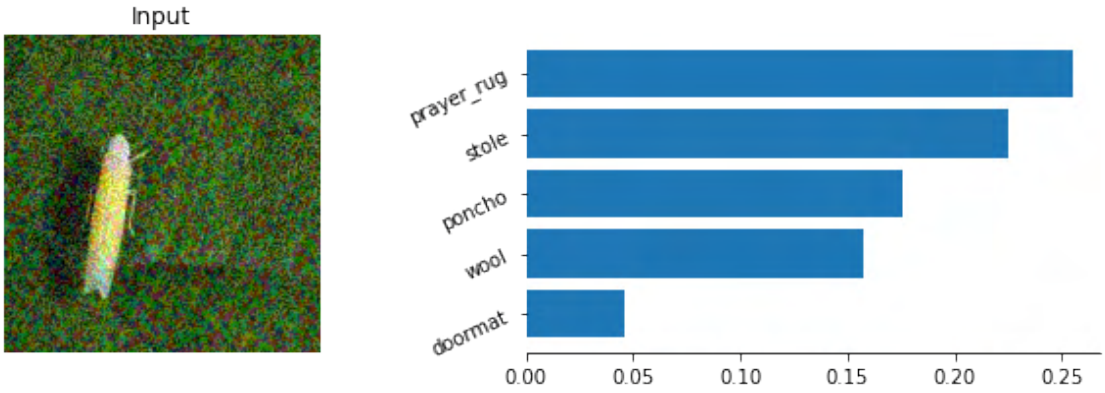

```
cricket
1/1 [==============================] - 0s 49ms/step
```



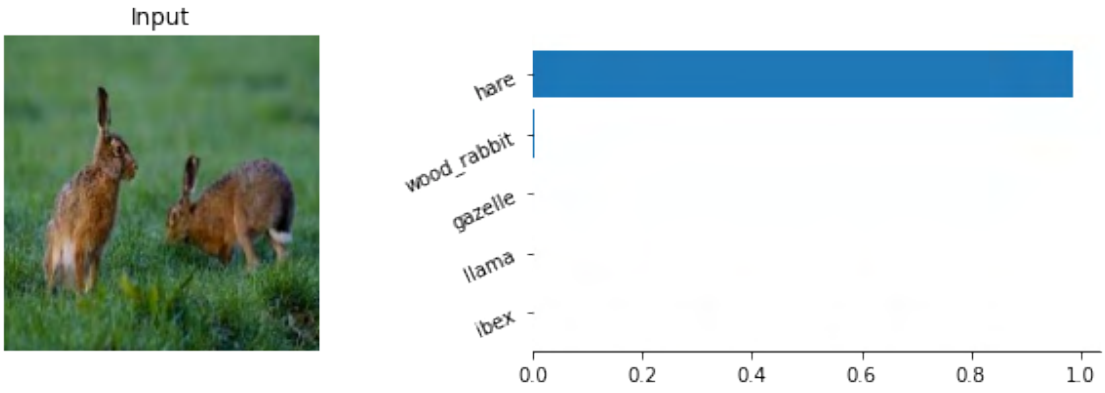

```
poncho

Loading Next Image
\n02259212.JPEG
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 44ms/step
```



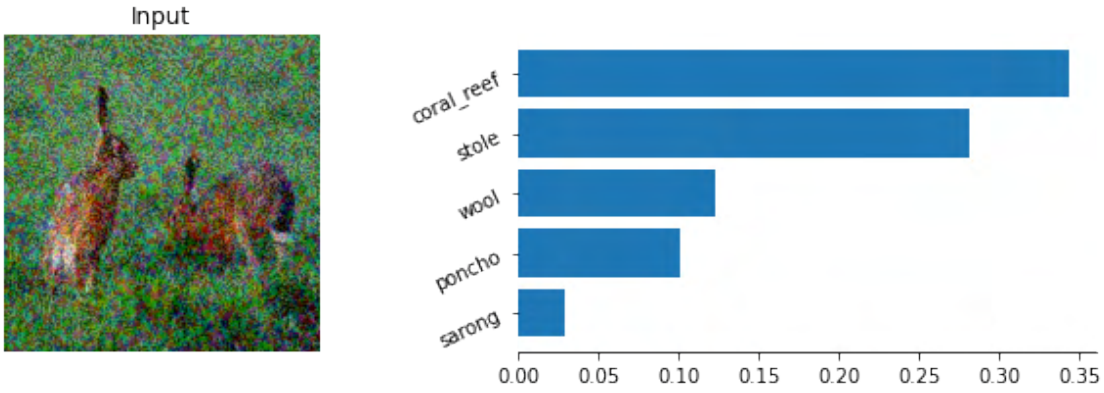

```
leafhopper
1/1 [==============================] - 0s 46ms/step
```



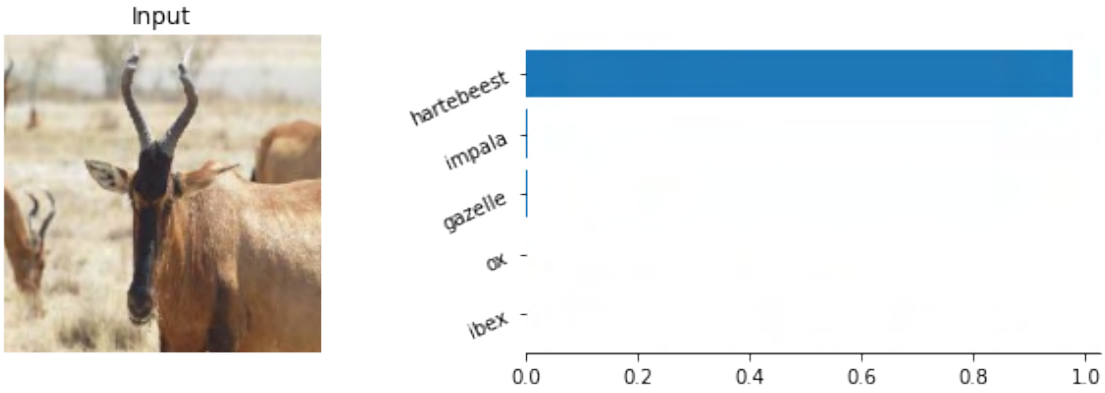

```
prayer_rug

Loading Next Image
\n02326432.JPEG
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 38ms/step
```



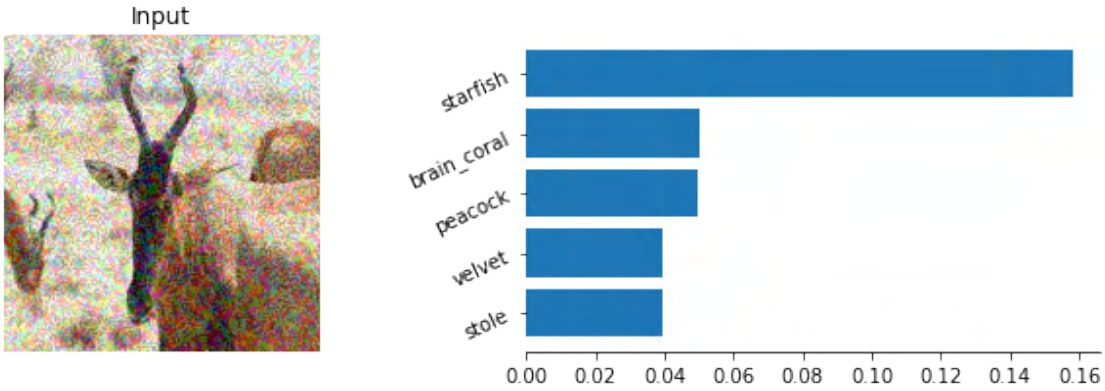

```
hare
1/1 [==============================] - 0s 58ms/step
```



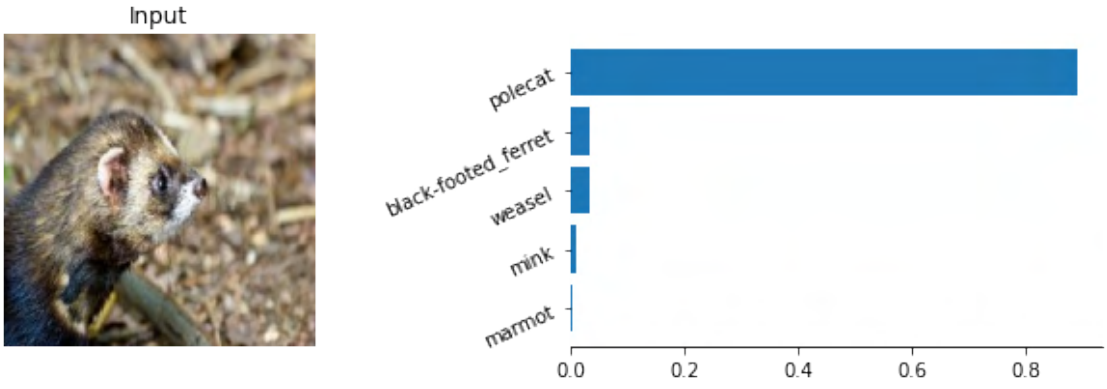

```
coral_reef

Loading Next Image
\n02422106.JPEG
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 39ms/step
```



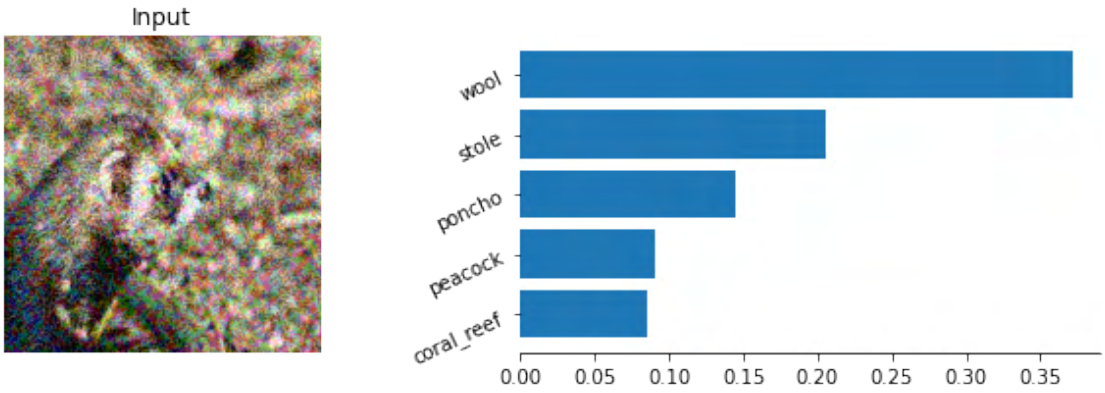

```
hartebeest
1/1 [==============================] - 0s 49ms/step
```



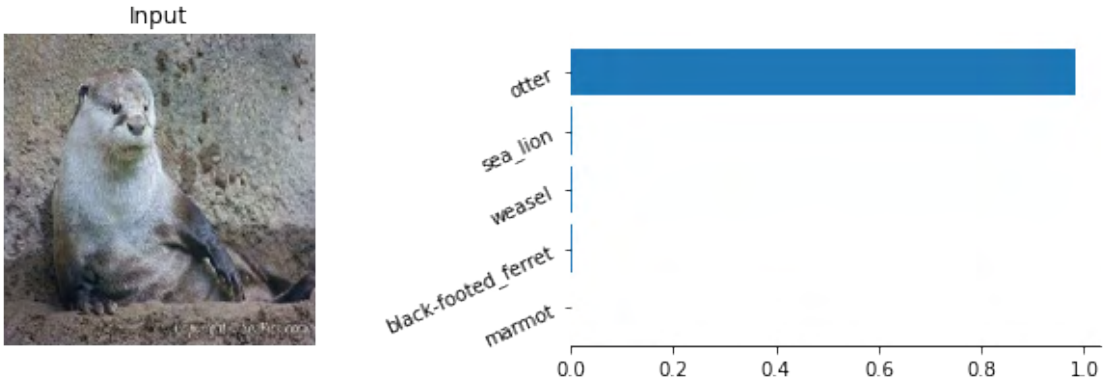

```
starfish

Loading Next Image
\n02443114.JPEG
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 60ms/step
1/1 [==============================] - 0s 46ms/step
```



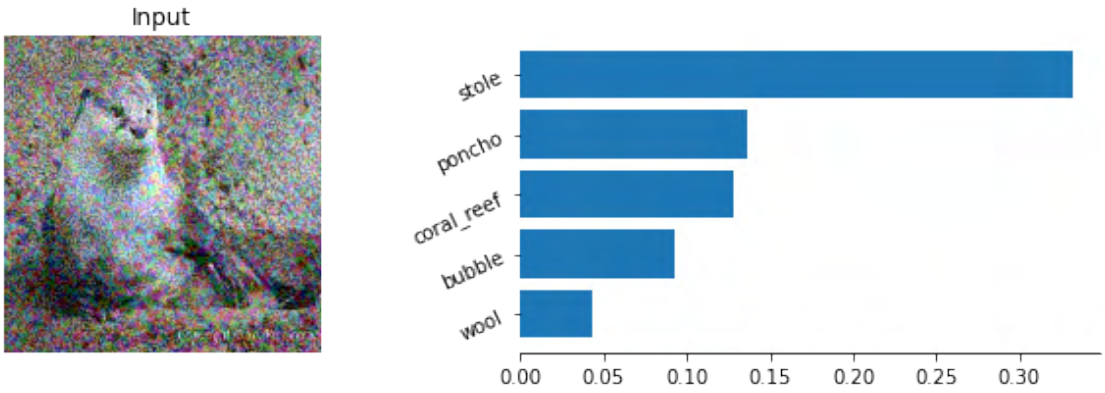

```
polecat
1/1 [==============================] - 0s 36ms/step
```



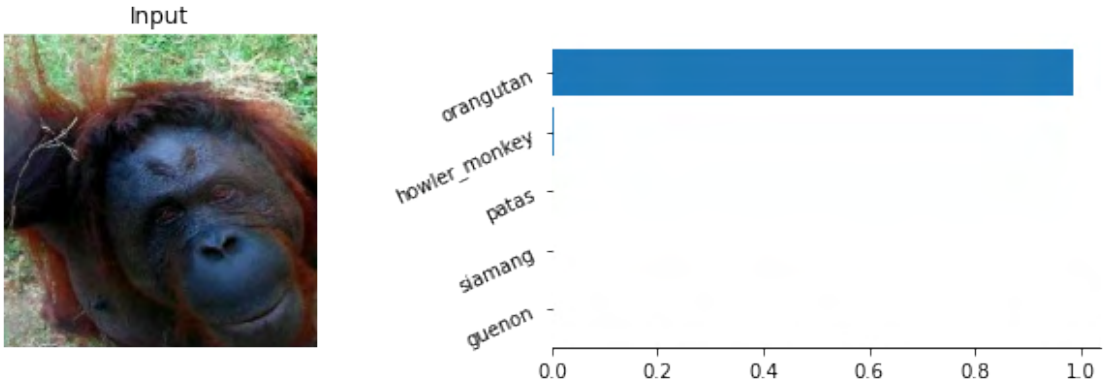

```
wool

Loading Next Image
\n02444819.JPEG
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 42ms/step
```



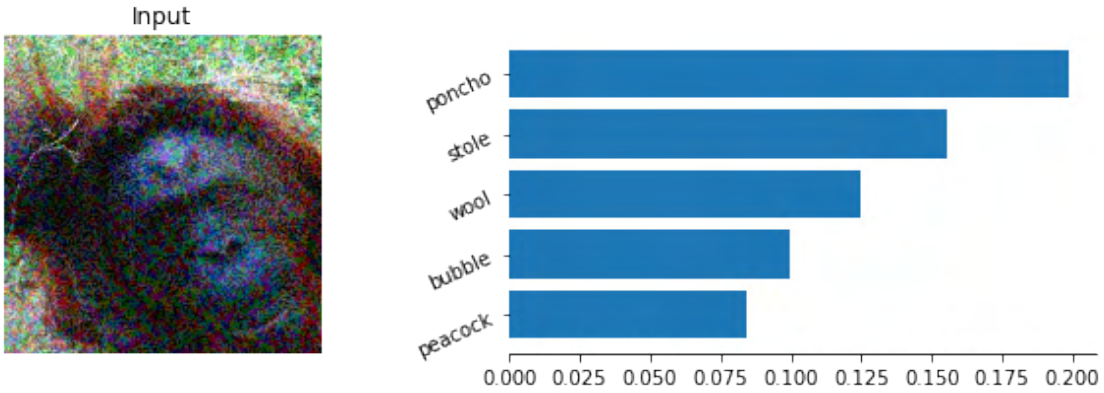

```
otter
1/1 [==============================] - 0s 58ms/step
```



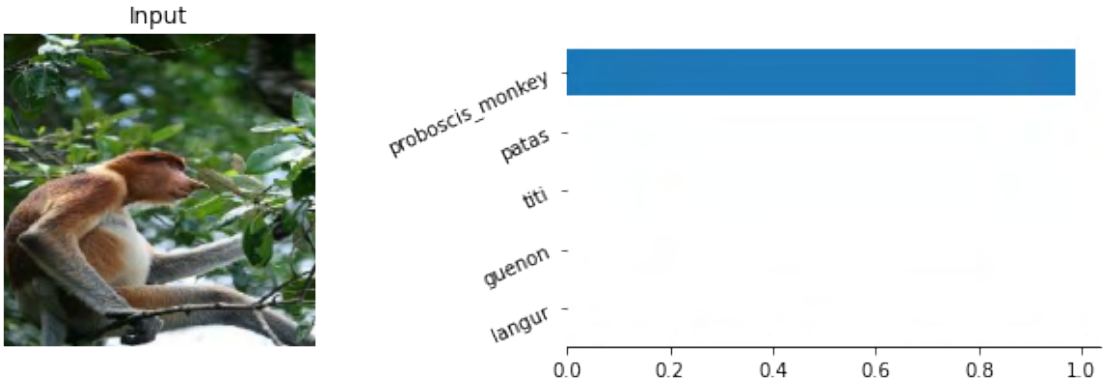

```
stole

Loading Next Image
\n02480495.JPEG
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 39ms/step
```



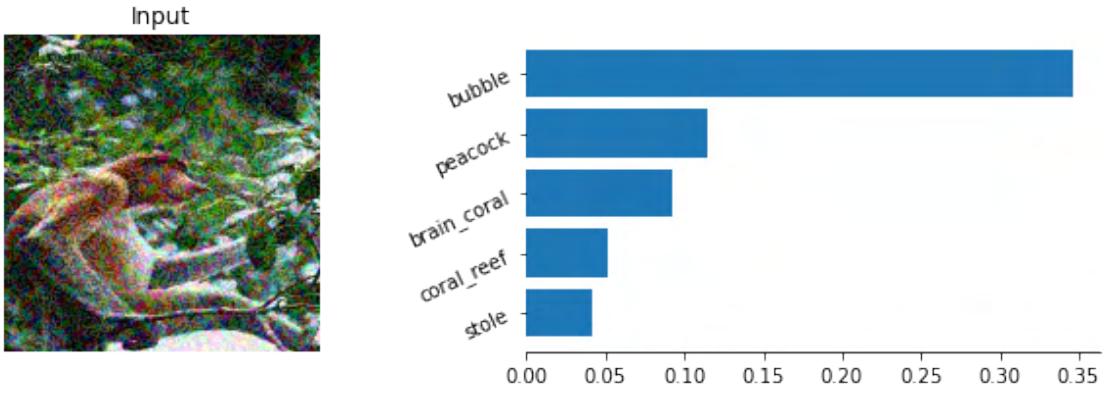

```
orangutan
1/1 [==============================] - 0s 47ms/step
```



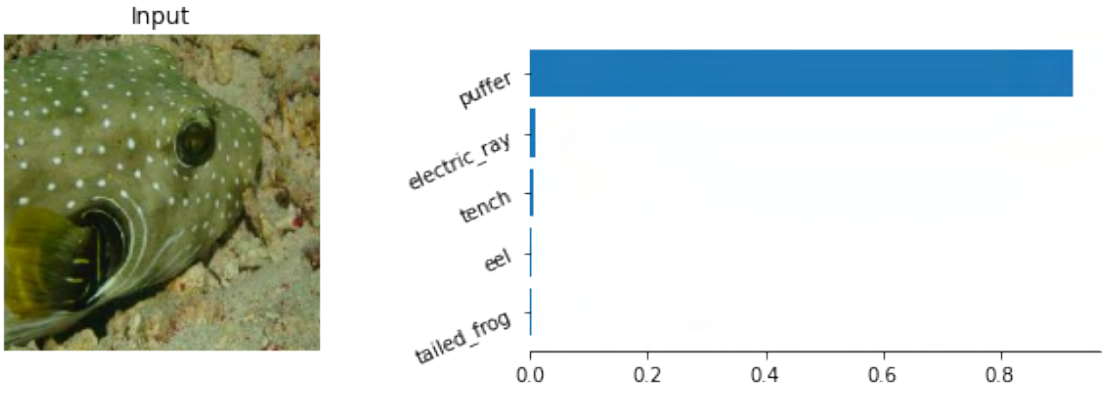

```
poncho

Loading Next Image
\n02489166.JPEG
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 58ms/step
1/1 [==============================] - 0s 56ms/step
```



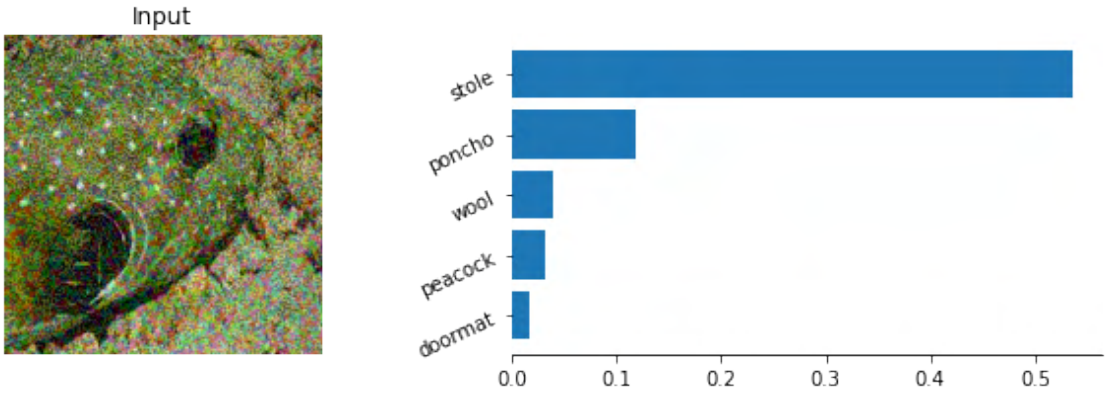

```
proboscis_monkey
1/1 [==============================] - 0s 35ms/step
```



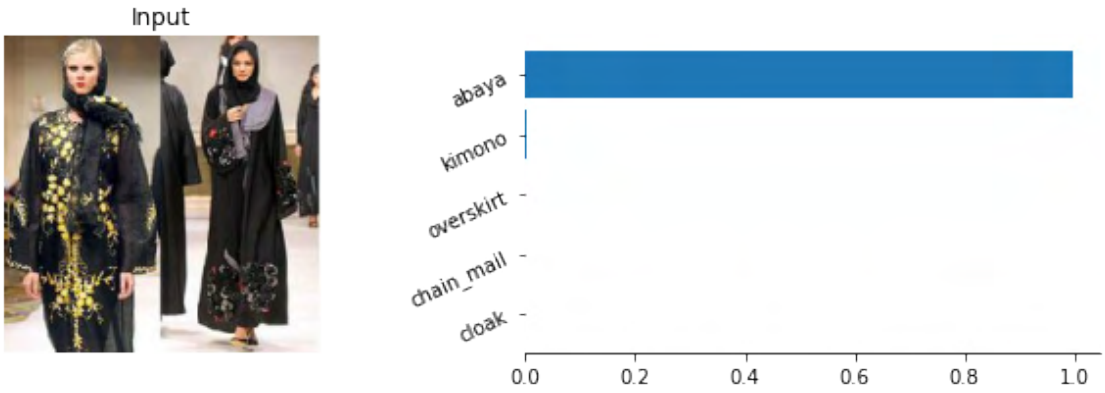

```
bubble

Loading Next Image
\n02655020.JPEG
1/1 [==============================] - 0s 65ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 50ms/step
```



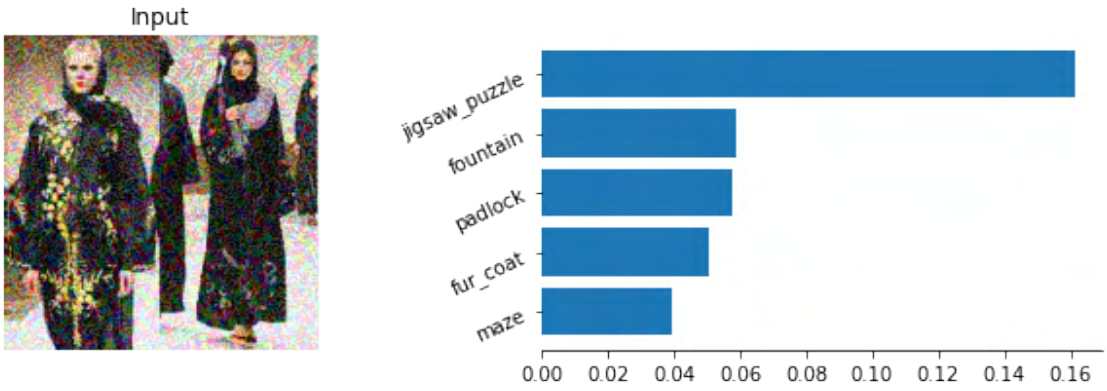

```
puffer
1/1 [==============================] - 0s 50ms/step
```



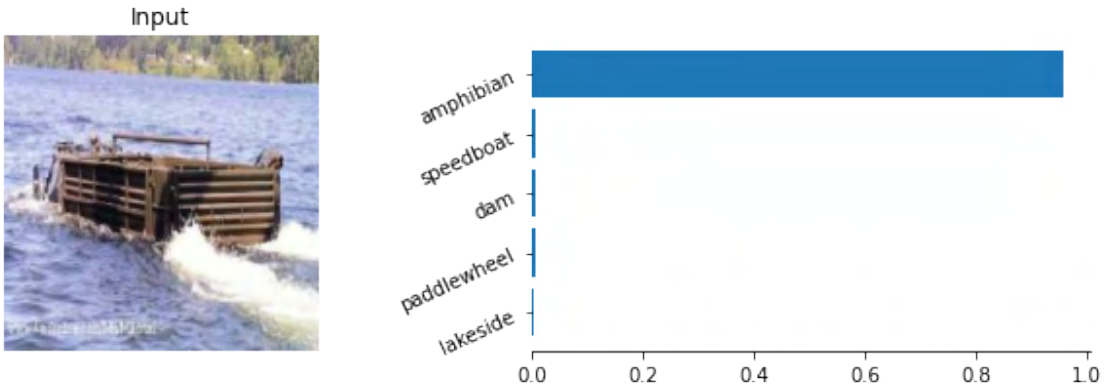

```
stole

Loading Next Image
\n02667093.JPEG
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 36ms/step
```



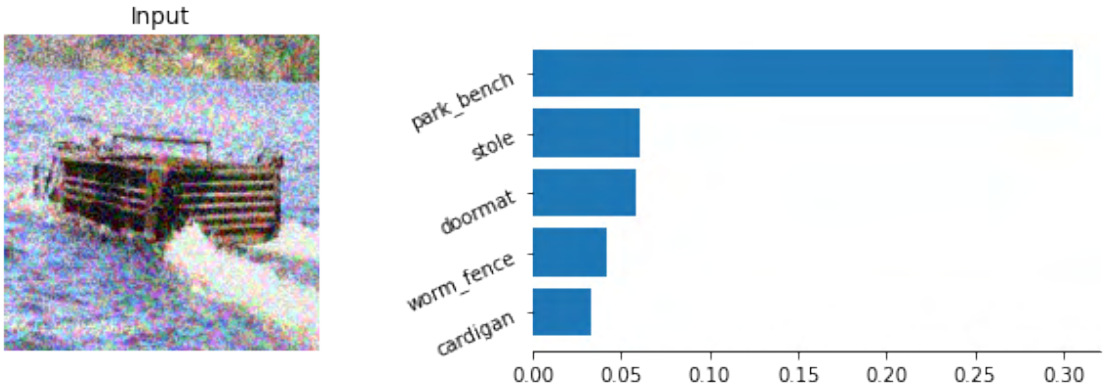

```
abaya
1/1 [==============================] - 0s 48ms/step
```



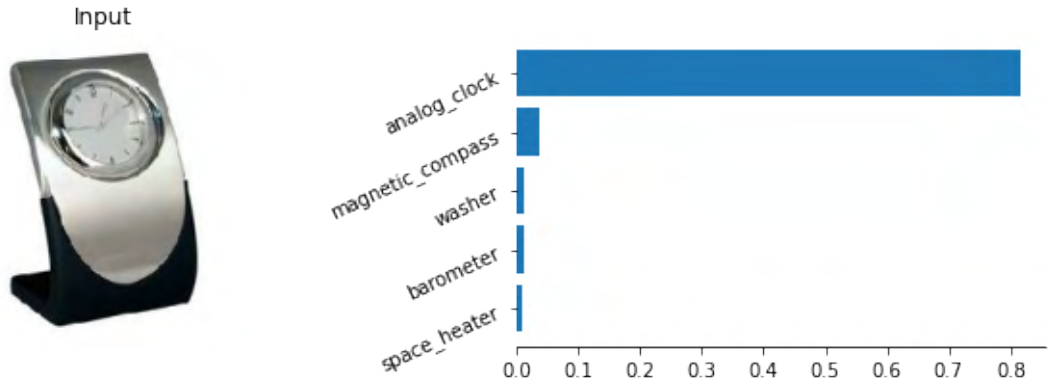

```
jigsaw_puzzle

Loading Next Image
\n02704792.JPEG
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 58ms/step
```



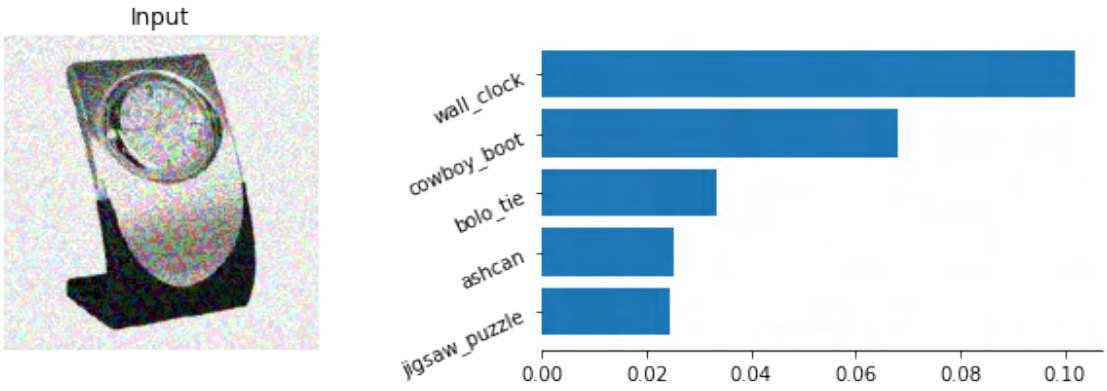

```
amphibian
1/1 [==============================] - 0s 49ms/step
```



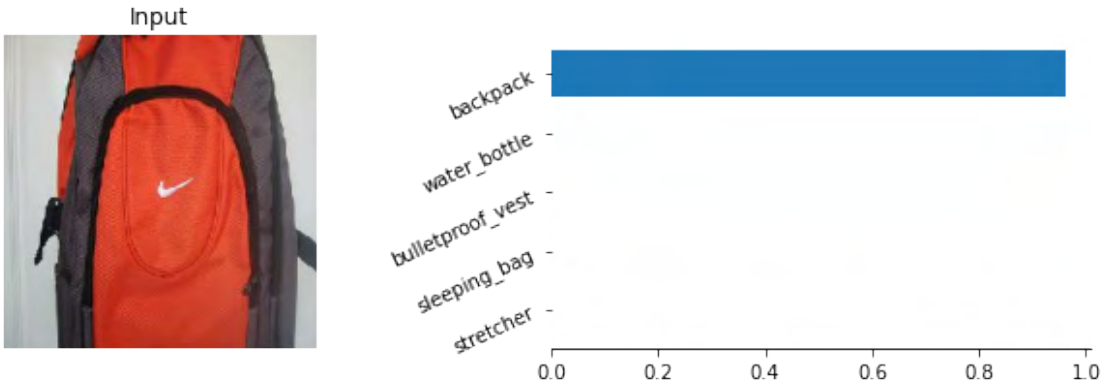

```
park_bench

Loading Next Image
\n02708093.JPEG
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 58ms/step
1/1 [==============================] - 0s 40ms/step
```



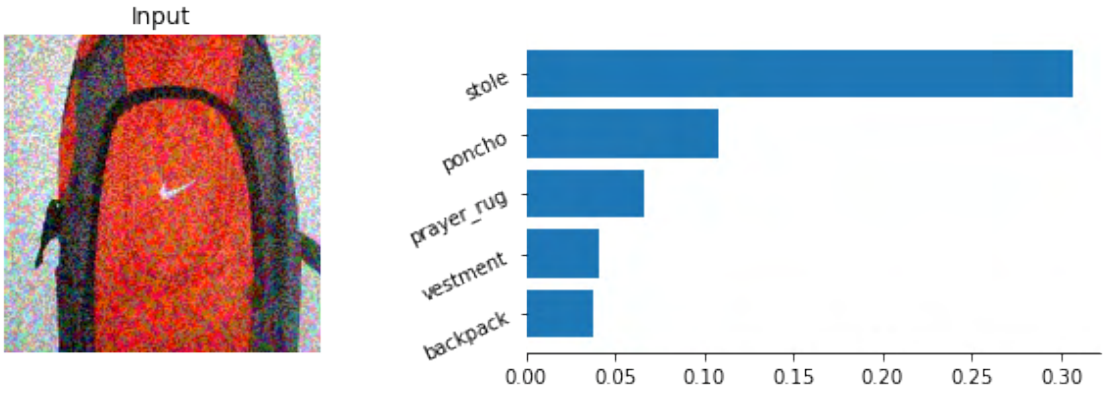

```
analog_clock
1/1 [==============================] - 0s 49ms/step
```



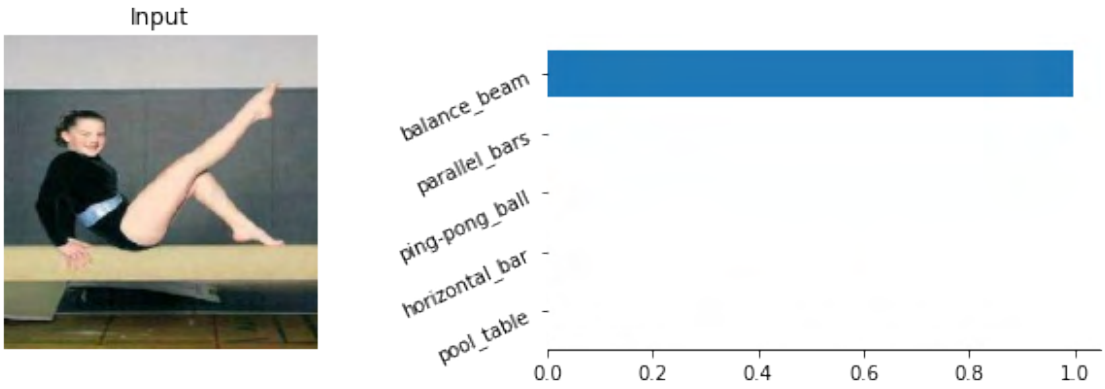

```
wall_clock

Loading Next Image
\n02769748.JPEG
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 58ms/step
```



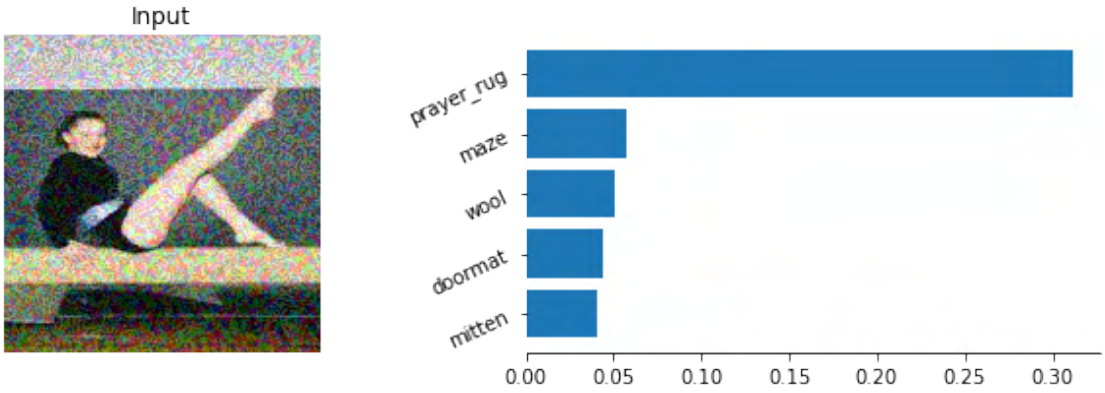

```
backpack
1/1 [==============================] - 0s 51ms/step
```



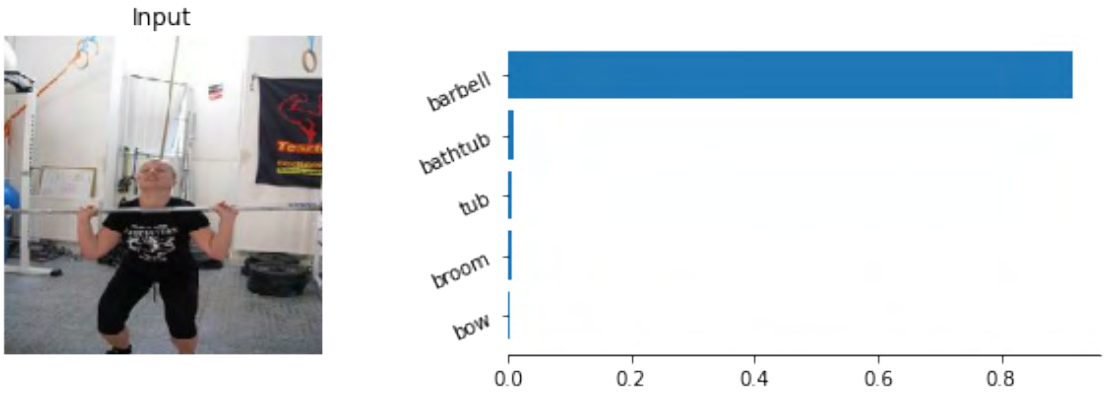

```
stole

Loading Next Image
\n02777292.JPEG
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 47ms/step
```



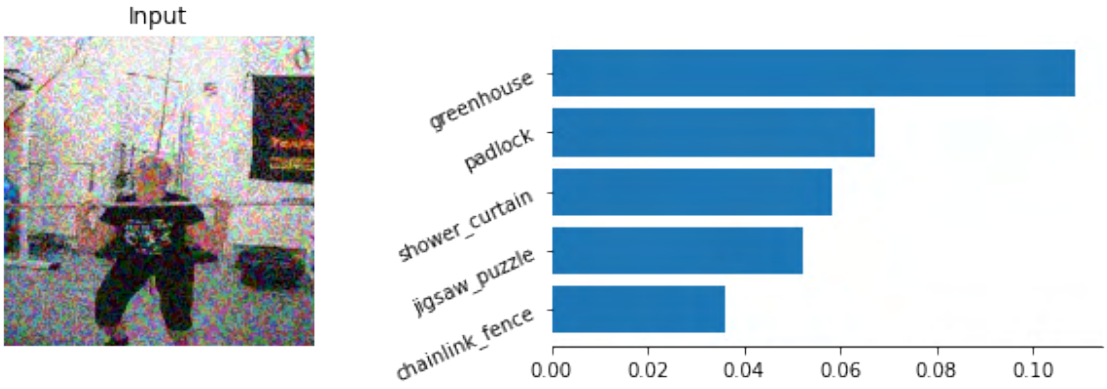

```
balance_beam
1/1 [==============================] - 0s 34ms/step
```



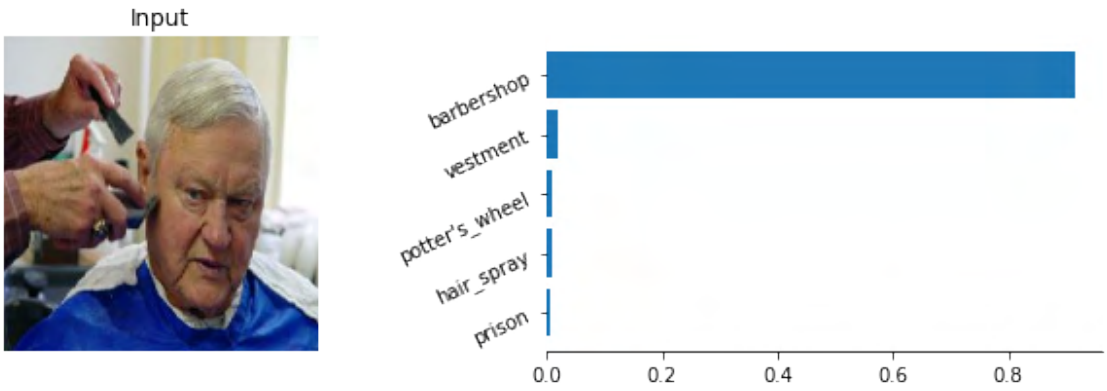

```
prayer_rug

Loading Next Image
\n02790996.JPEG
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 36ms/step
```



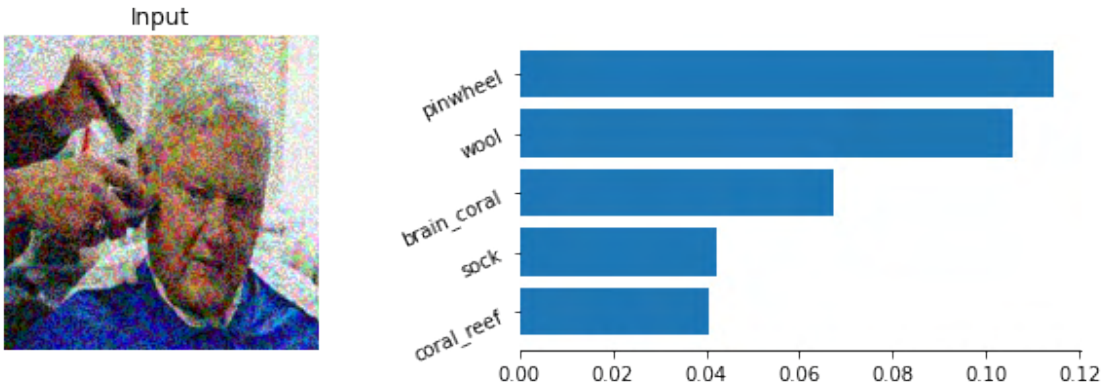

```
barbell
1/1 [==============================] - 0s 50ms/step
```



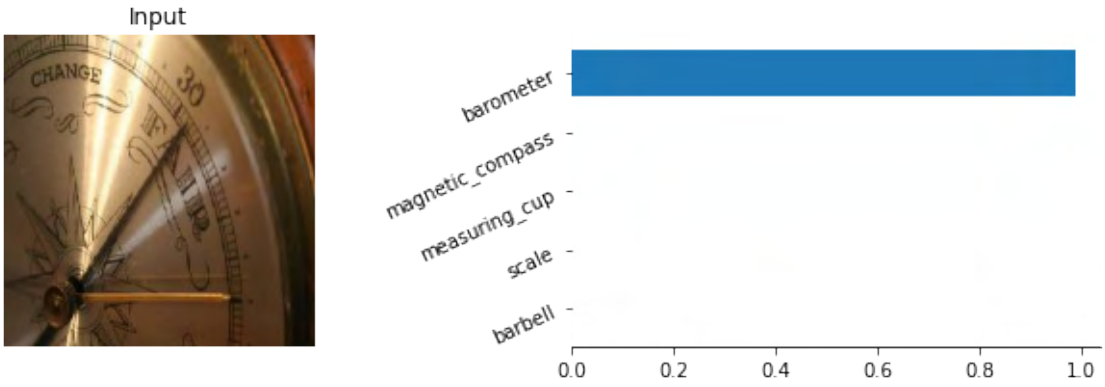

```
greenhouse

Loading Next Image
\n02791270.JPEG
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 41ms/step
```



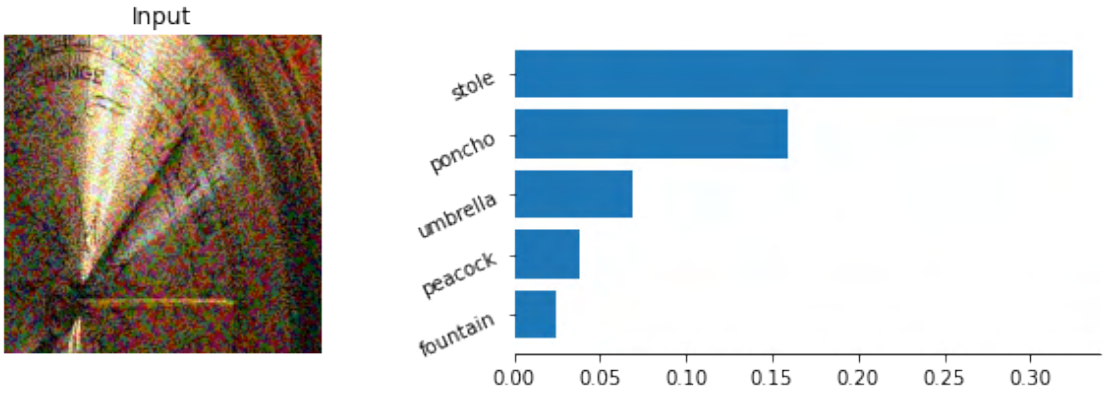

```
barbershop
1/1 [==============================] - 0s 48ms/step
```



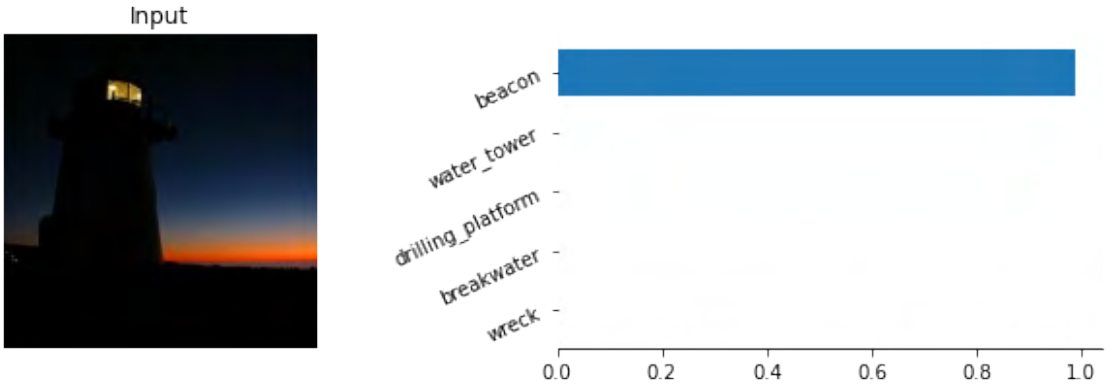

```
pinwheel

Loading Next Image
\n02794156.JPEG
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 63ms/step
```



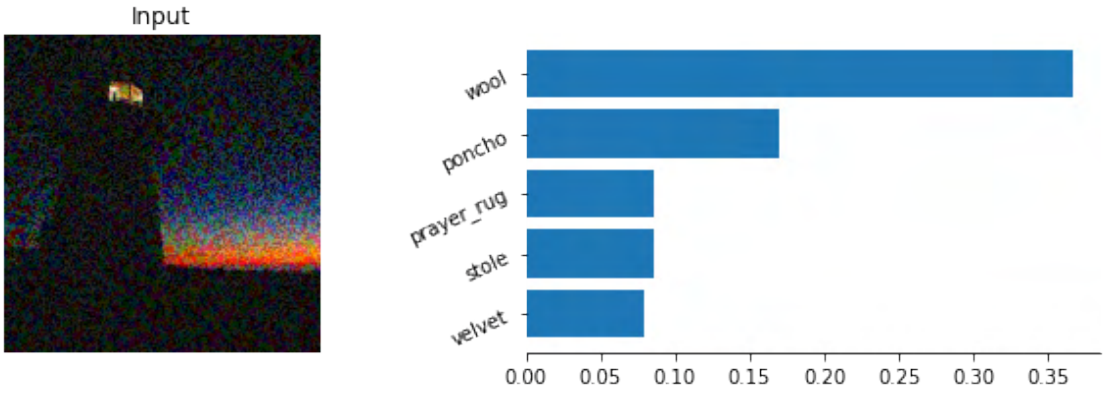

```
barometer
1/1 [==============================] - 0s 39ms/step
```



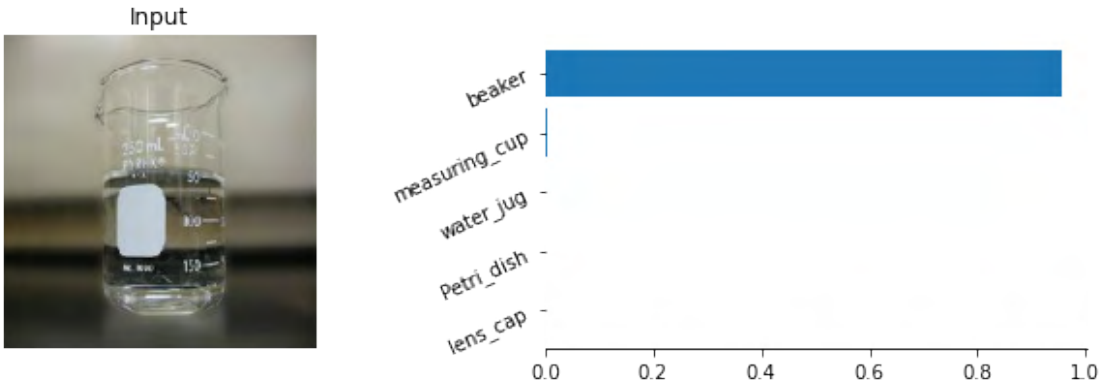

```
stole

Loading Next Image
\n02814860.JPEG
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 62ms/step
1/1 [==============================] - 0s 57ms/step
```



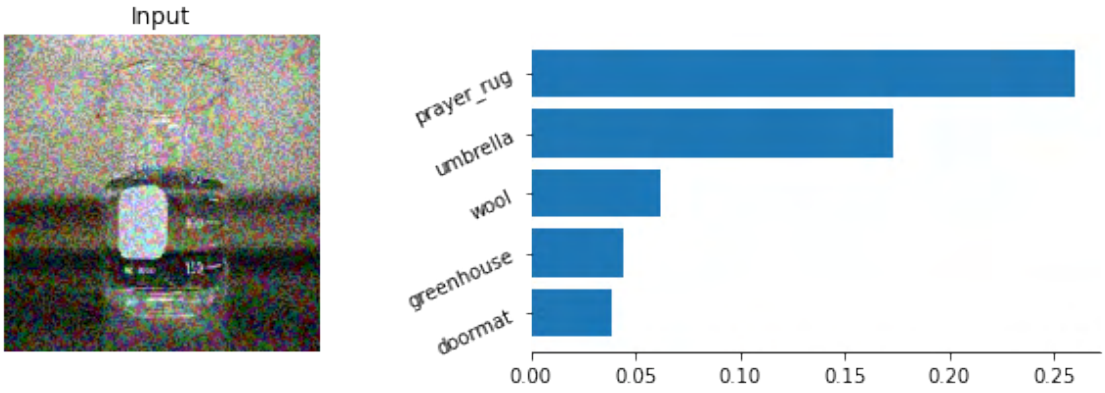

```
beacon
1/1 [==============================] - 0s 33ms/step
```



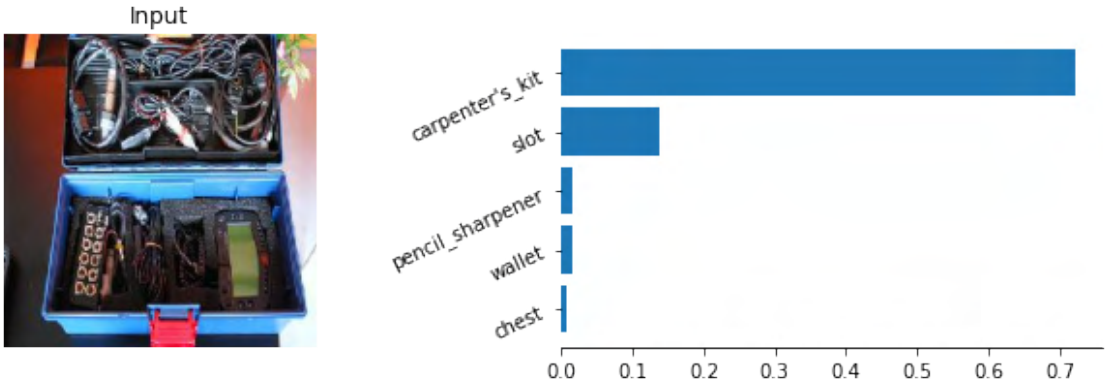

```
wool

Loading Next Image
\n02815834.JPEG
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 50ms/step
```



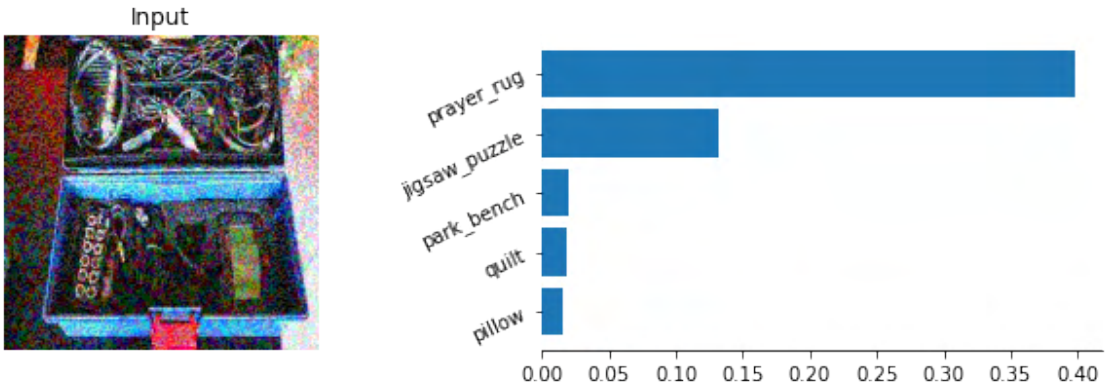

```
beaker
1/1 [==============================] - 0s 51ms/step
```



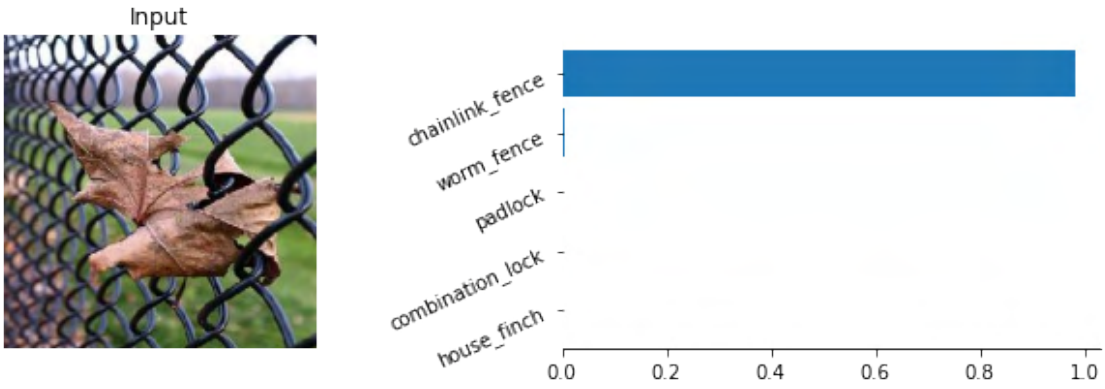

```
prayer_rug

Loading Next Image
\n02966687.JPEG
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 39ms/step
```



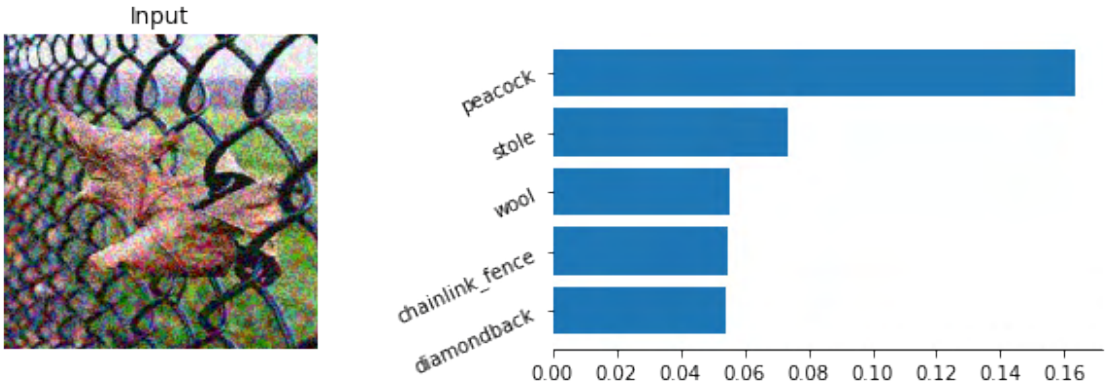

```
carpenter's_kit
1/1 [==============================] - 0s 50ms/step
```



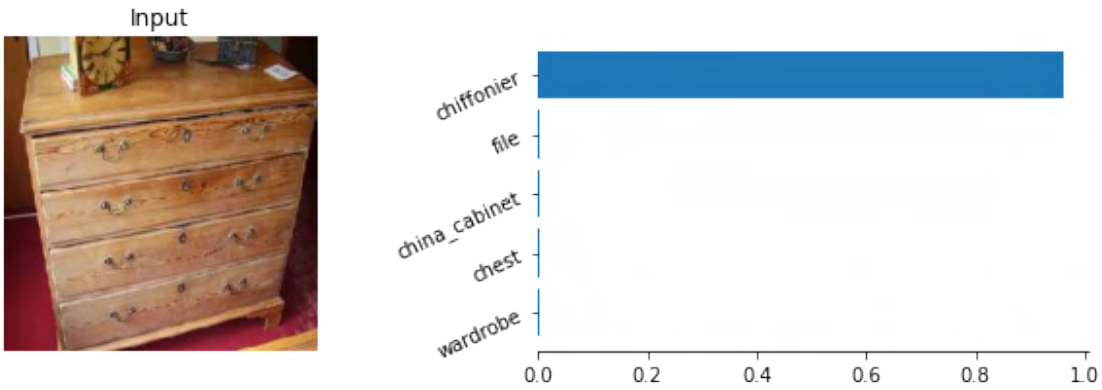

```
prayer_rug

Loading Next Image
\n03000134.JPEG
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 37ms/step
```



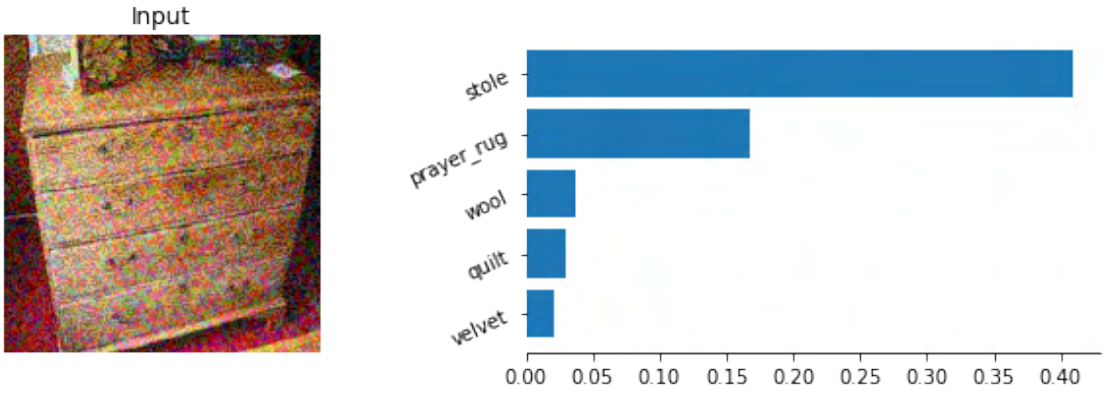

```
chainlink_fence
1/1 [==============================] - 0s 51ms/step
```



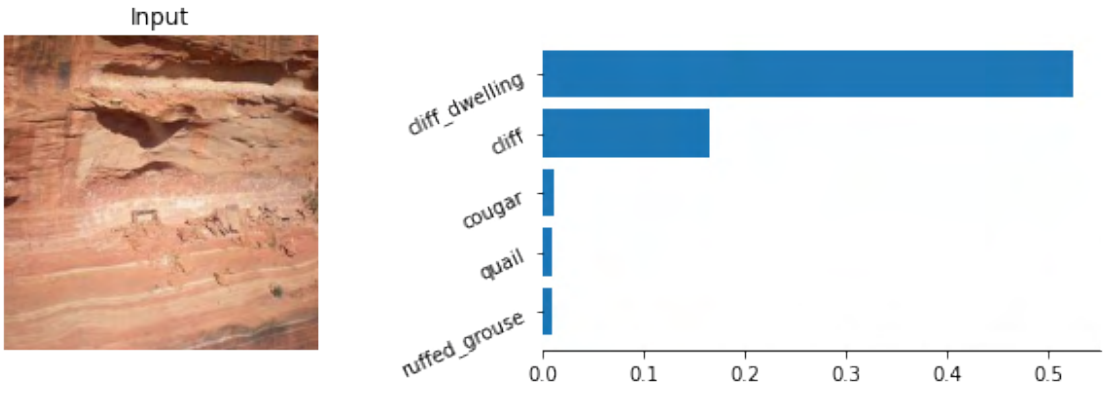

```
peacock

Loading Next Image
\n03016953.JPEG
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 60ms/step
```



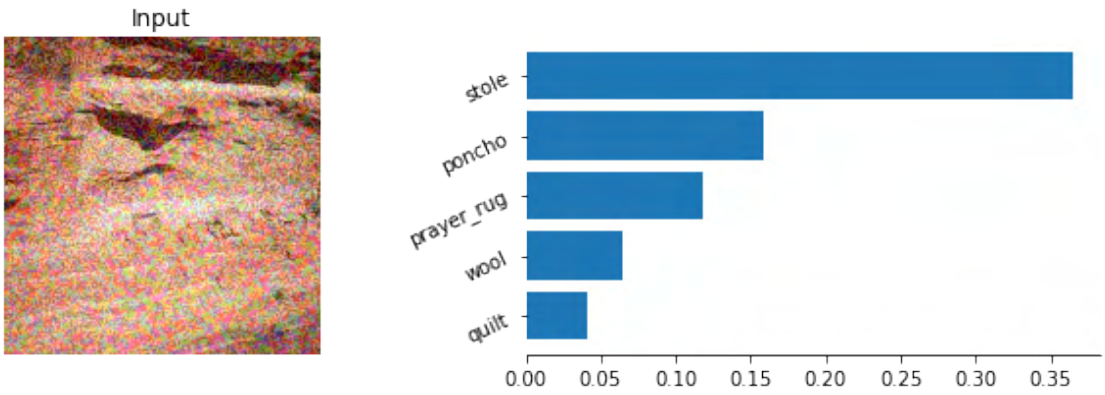

```
chiffonier
1/1 [==============================] - 0s 49ms/step
```



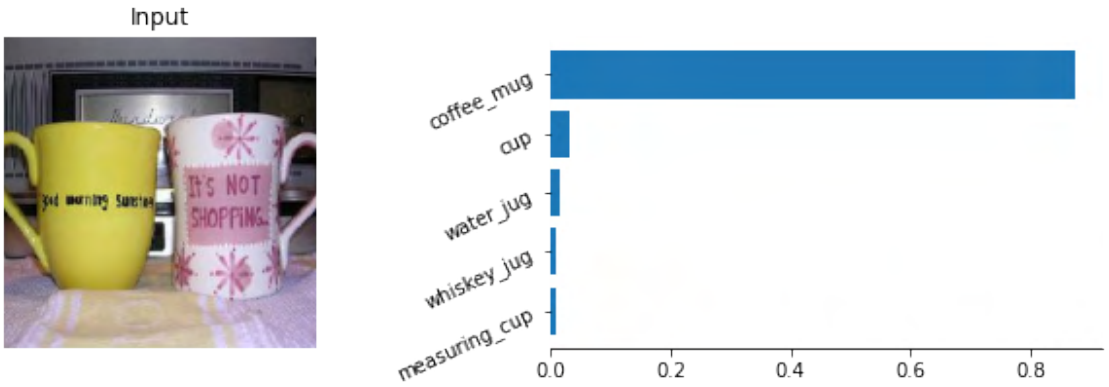

```
stole

Loading Next Image
\n03042490.JPEG
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 57ms/step
```



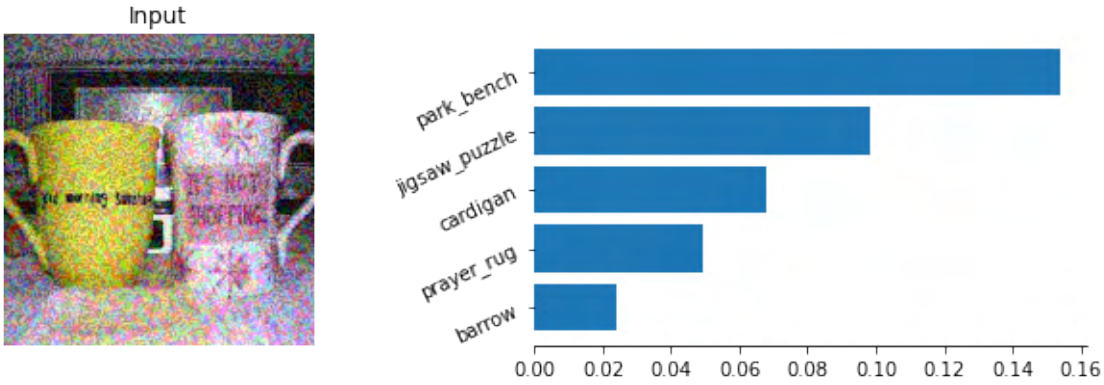

```
cliff_dwelling
1/1 [==============================] - 0s 44ms/step
```



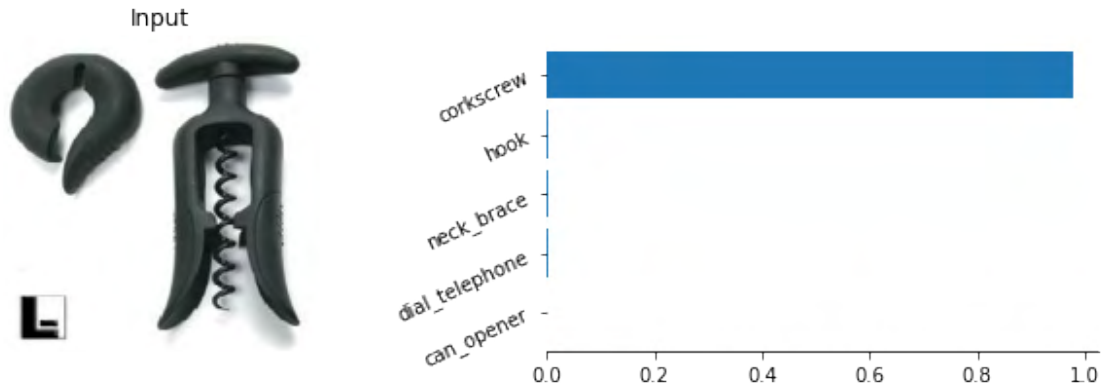

```
stole

Loading Next Image
\n03063599.JPEG
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 55ms/step
```



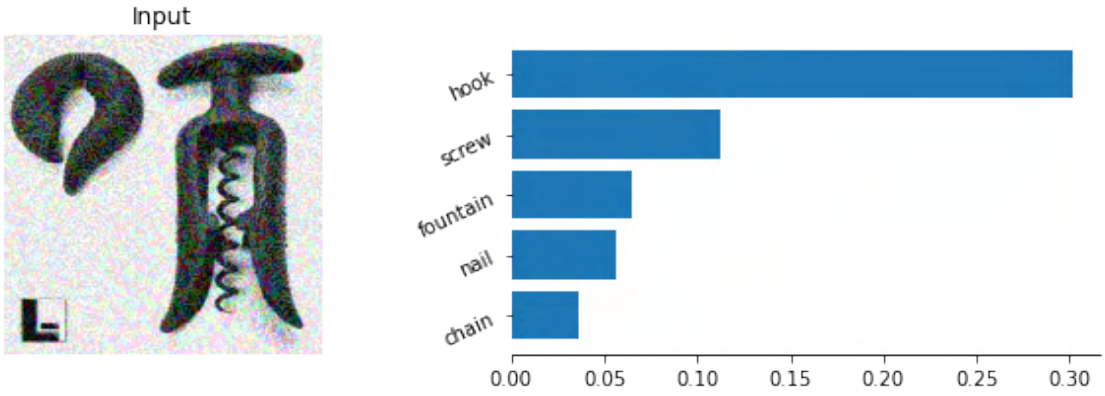

```
coffee_mug
1/1 [==============================] - 0s 49ms/step
```



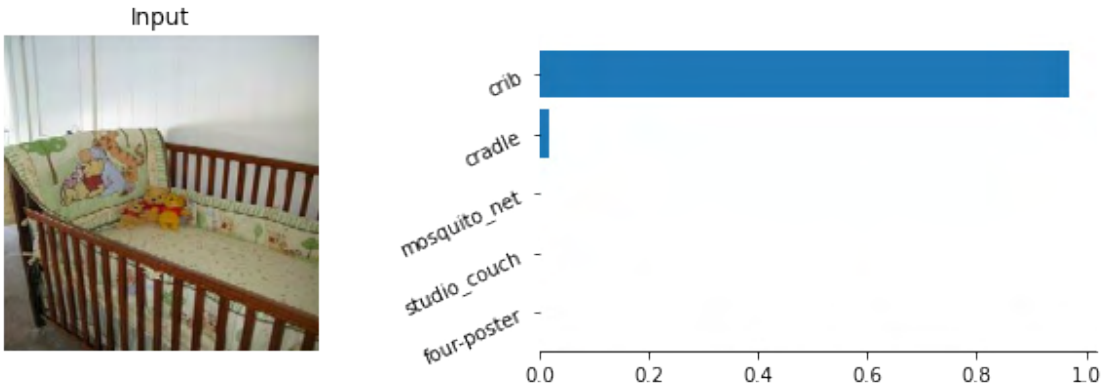

```
park_bench

Loading Next Image
\n03109150.JPEG
1/1 [==============================] - 0s 58ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 43ms/step
```



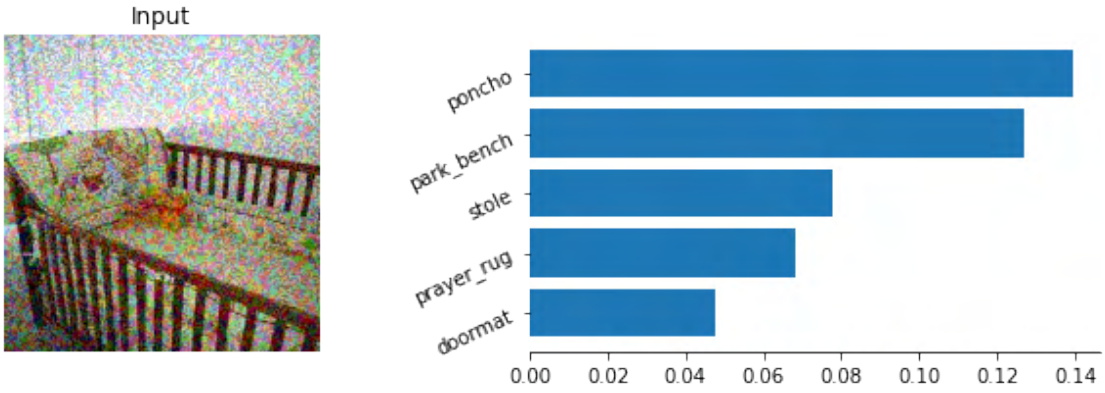

```
corkscrew
1/1 [==============================] - 0s 48ms/step
```



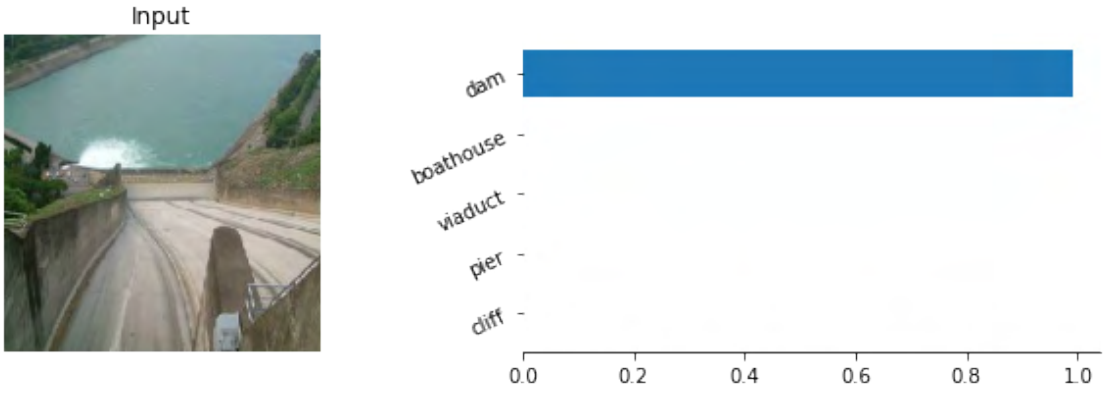

```
hook

Loading Next Image
\n03131574.JPEG
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 50ms/step
```



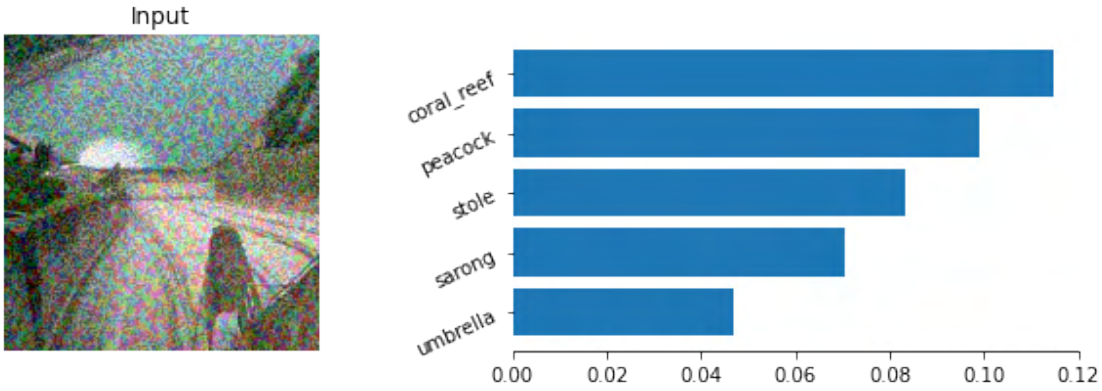

```
crib
1/1 [==============================] - 0s 70ms/step
```



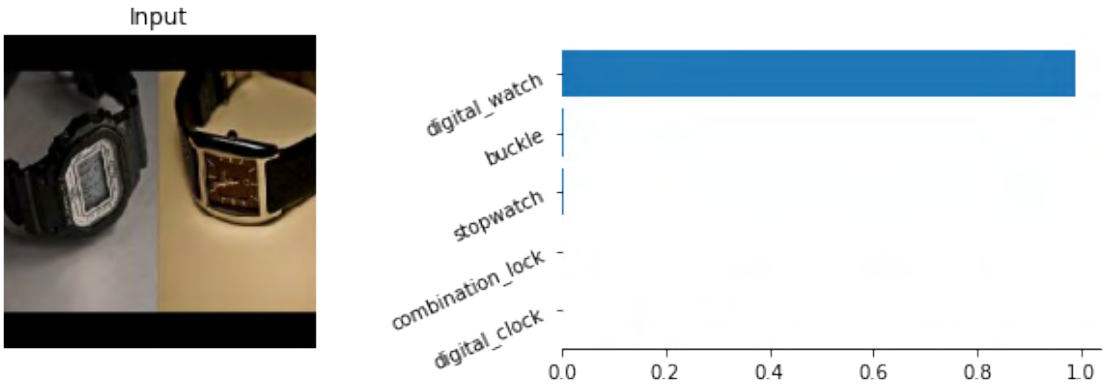

```
poncho

Loading Next Image
\n03160309.JPEG
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 33ms/step
```



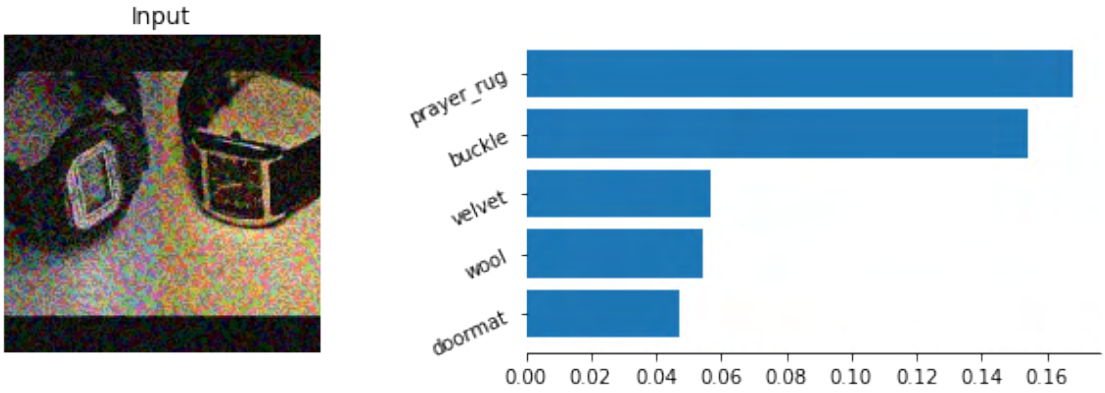

```
dam
1/1 [==============================] - 0s 55ms/step
```



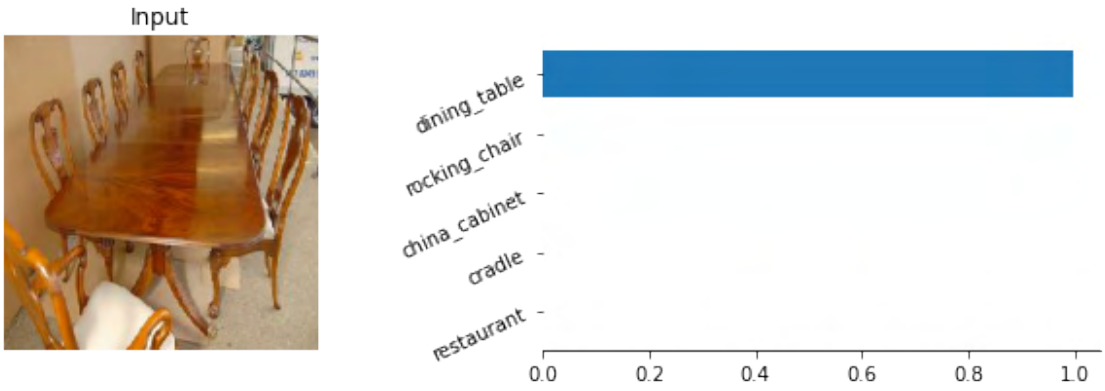

```
coral_reef

Loading Next Image
\n03197337.JPEG
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 55ms/step
```



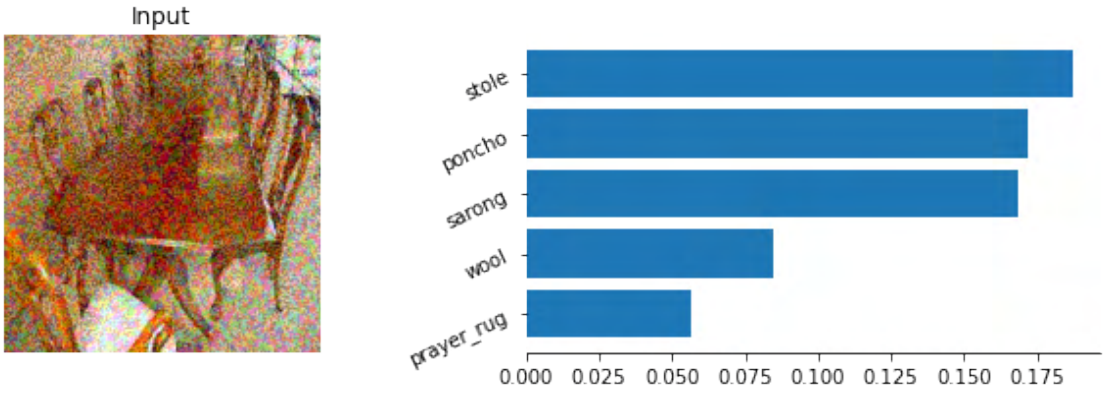

```
digital_watch
1/1 [==============================] - 0s 54ms/step
```



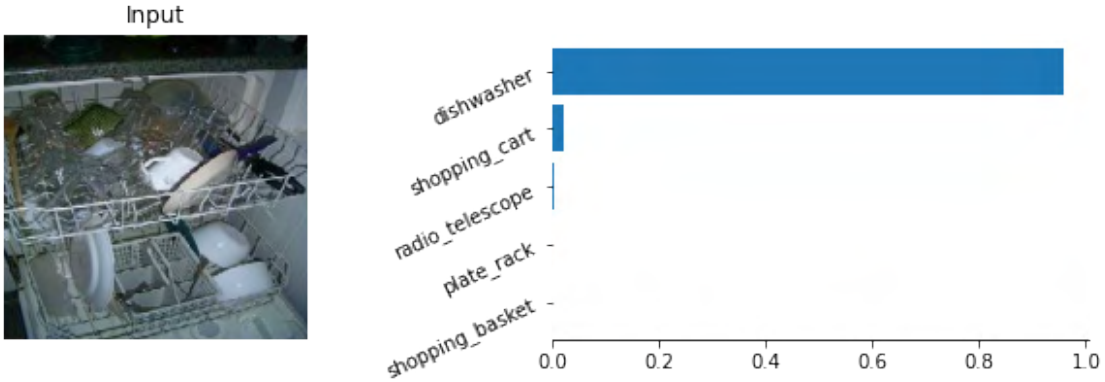

```
prayer_rug

Loading Next Image
\n03201208.JPEG
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 56ms/step
```



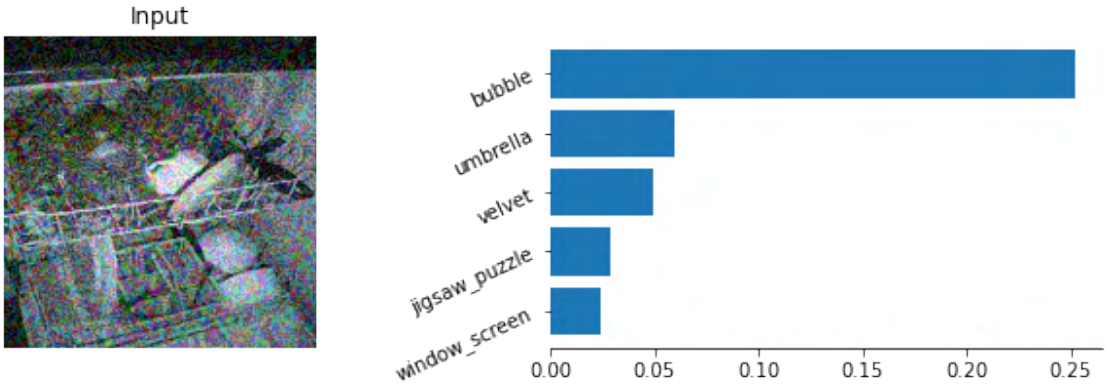

```
dining_table
1/1 [==============================] - 0s 52ms/step
```



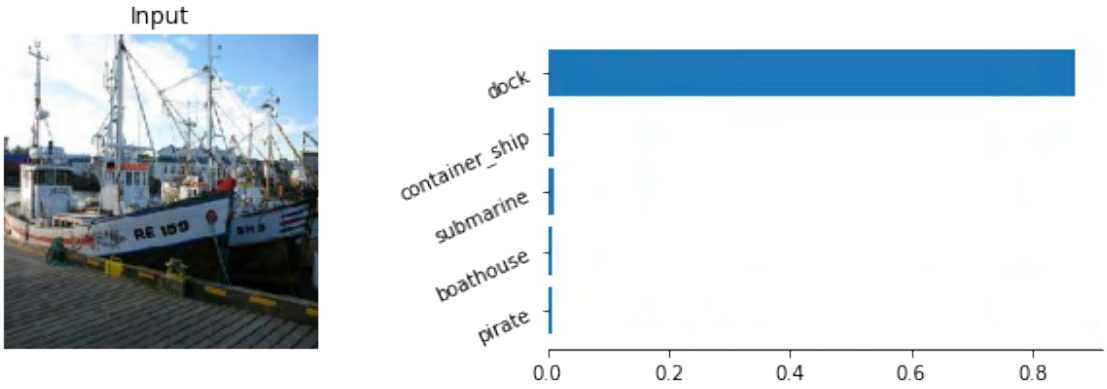

```
stole

Loading Next Image
\n03207941.JPEG
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 53ms/step
```



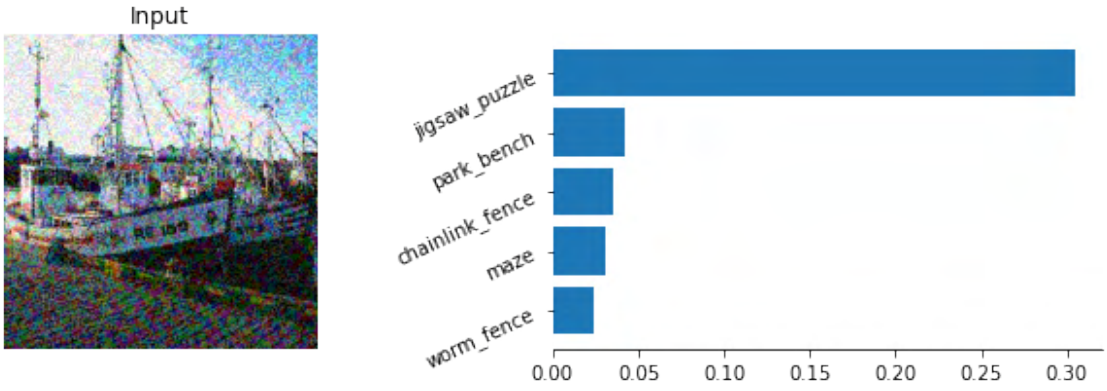

```
dishwasher
1/1 [==============================] - 0s 53ms/step
```



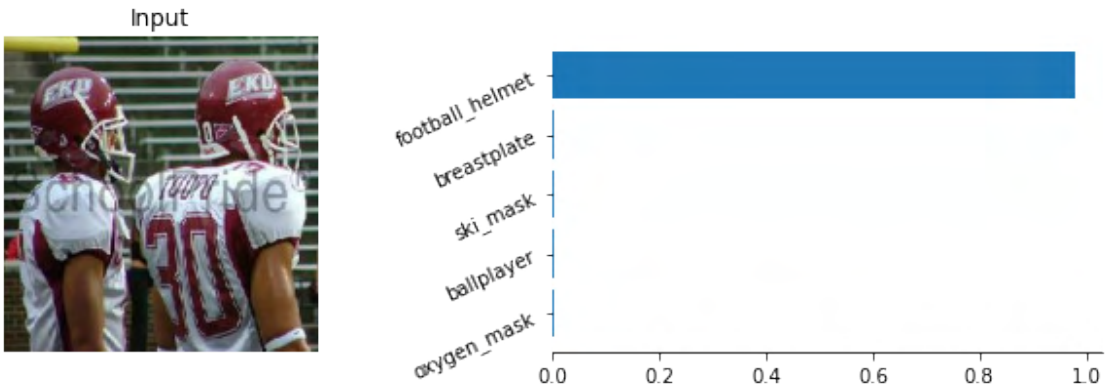

```
bubble

Loading Next Image
\n03216828.JPEG
1/1 [==============================] - 0s 68ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 53ms/step
```



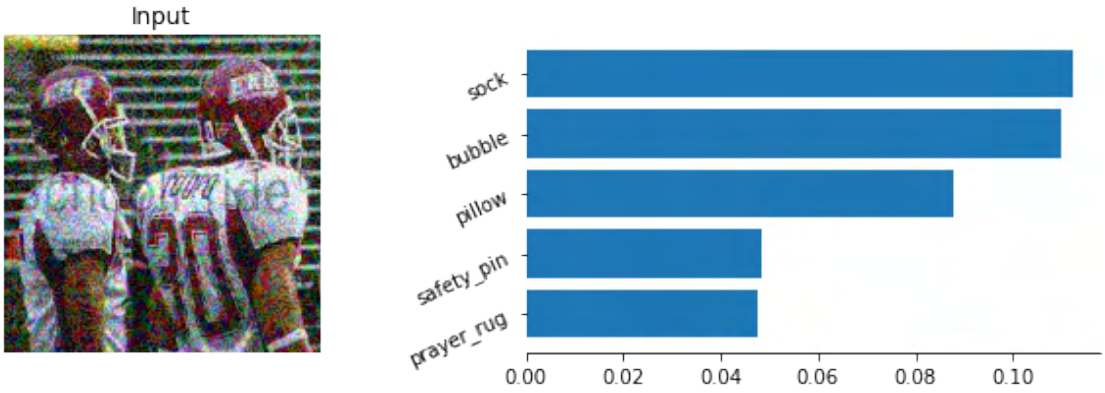

```
dock
1/1 [==============================] - 0s 59ms/step
```



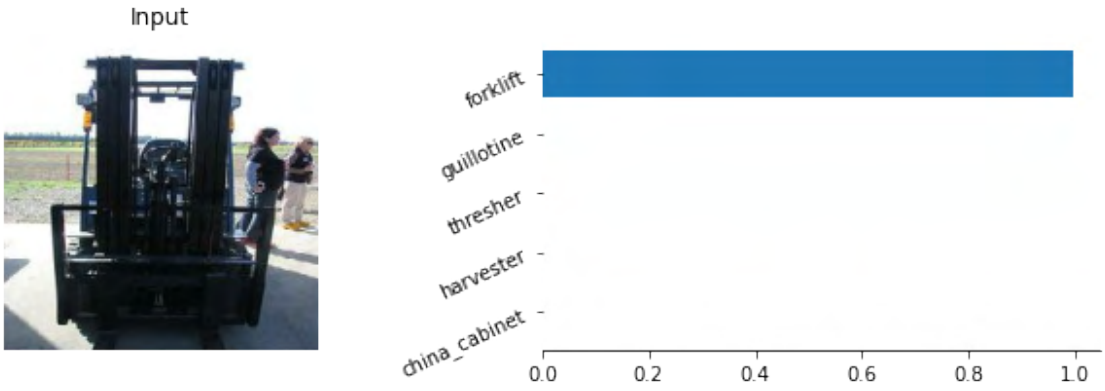

```
jigsaw_puzzle

Loading Next Image
\n03379051.JPEG
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
```



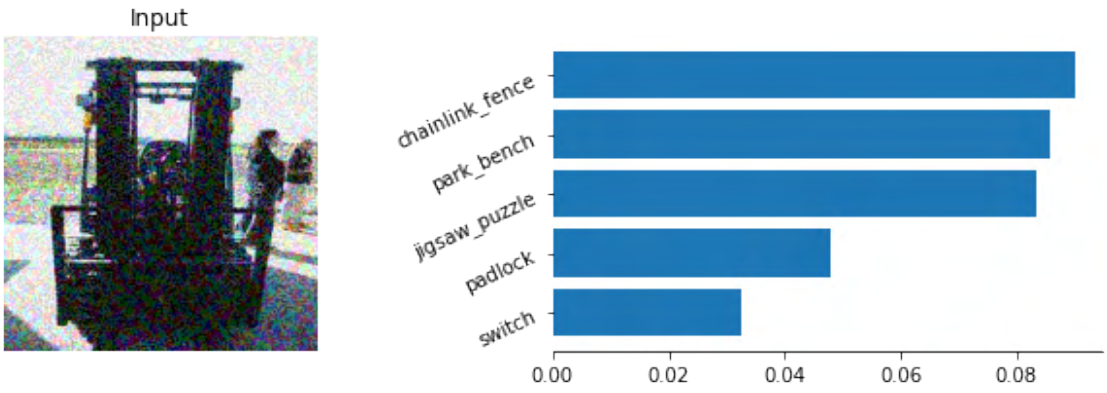

```
football_helmet
1/1 [==============================] - 0s 57ms/step
```



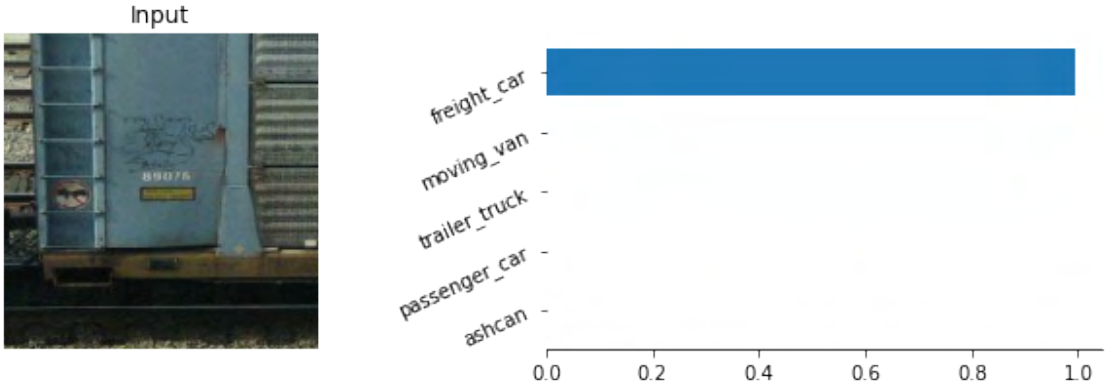

```
sock

Loading Next Image
\n03384352.JPEG
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 41ms/step
```



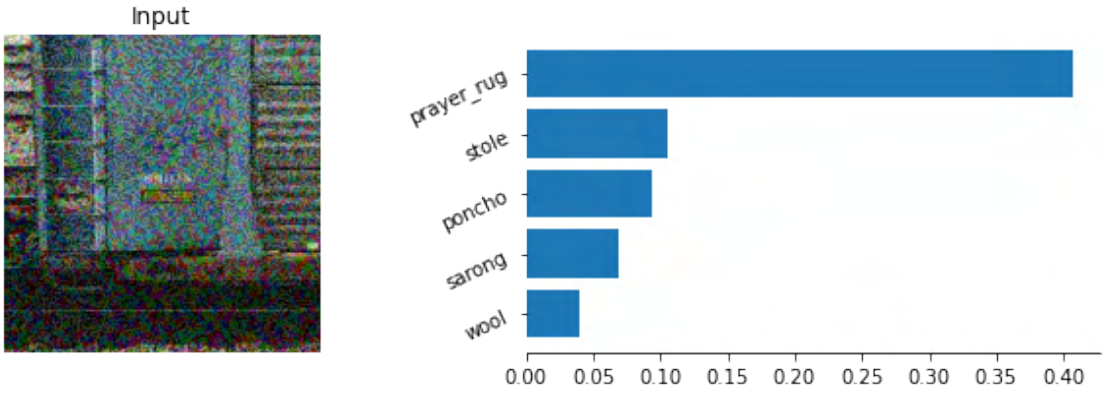

```
forklift
1/1 [==============================] - 0s 50ms/step
```



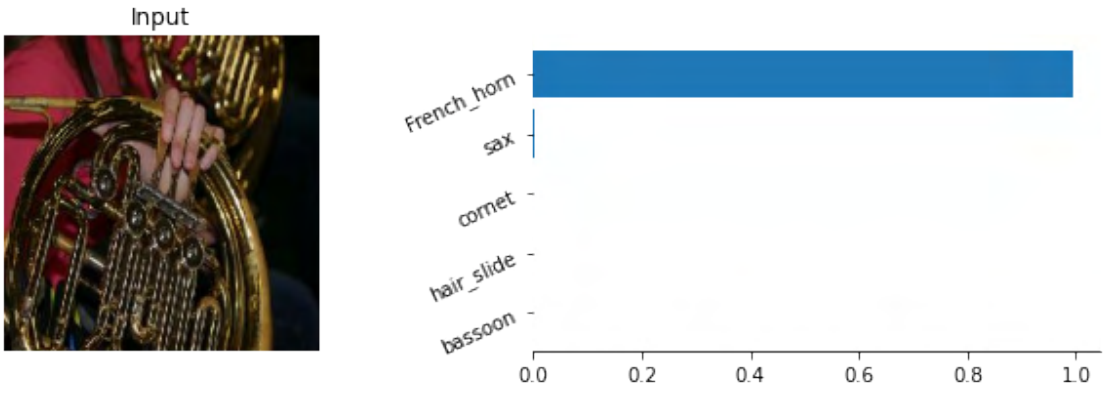

```
chainlink_fence

Loading Next Image
\n03393912.JPEG
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 46ms/step
```



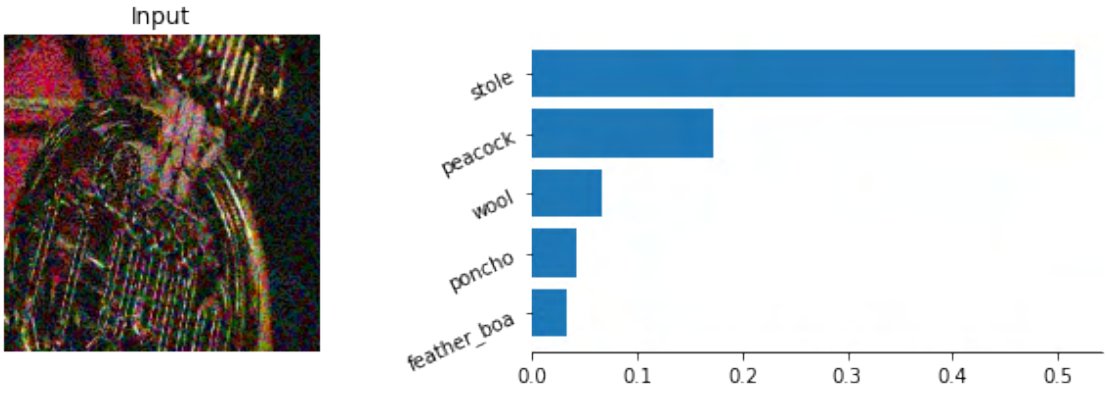

```
freight_car
1/1 [==============================] - 0s 46ms/step
```



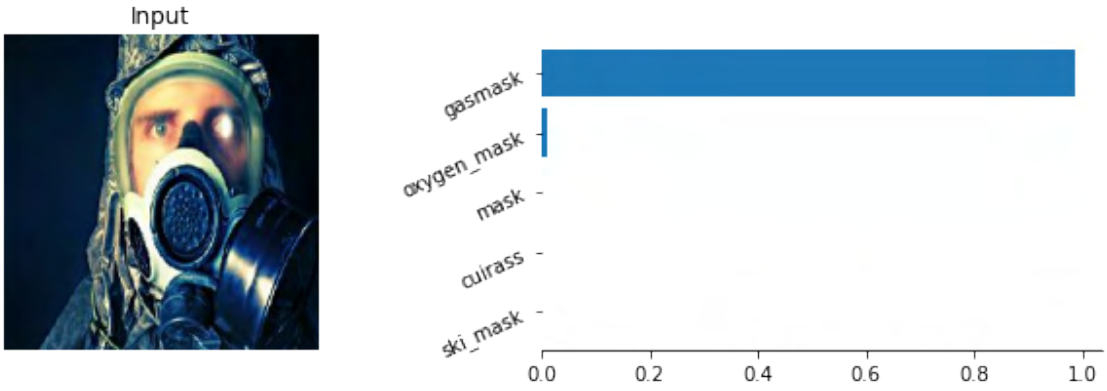

```
prayer_rug

Loading Next Image
\n03394916.JPEG
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 41ms/step
```



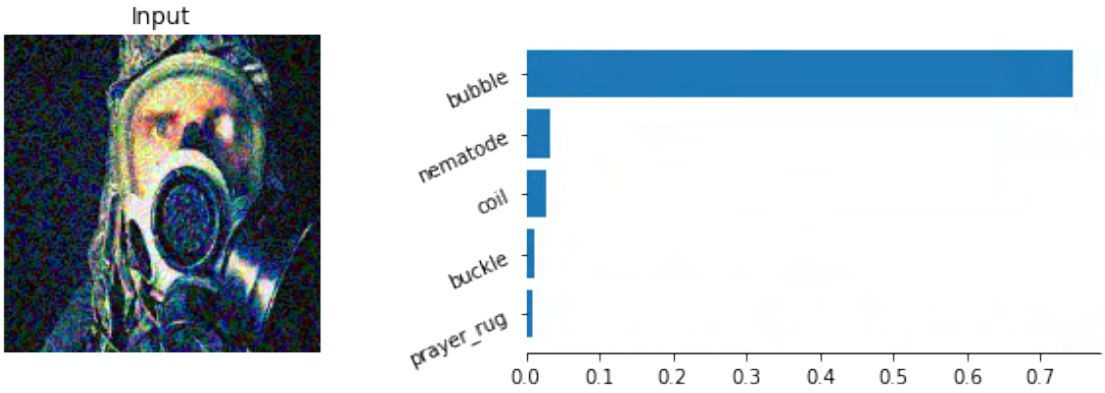

```
French_horn
1/1 [==============================] - 0s 48ms/step
```



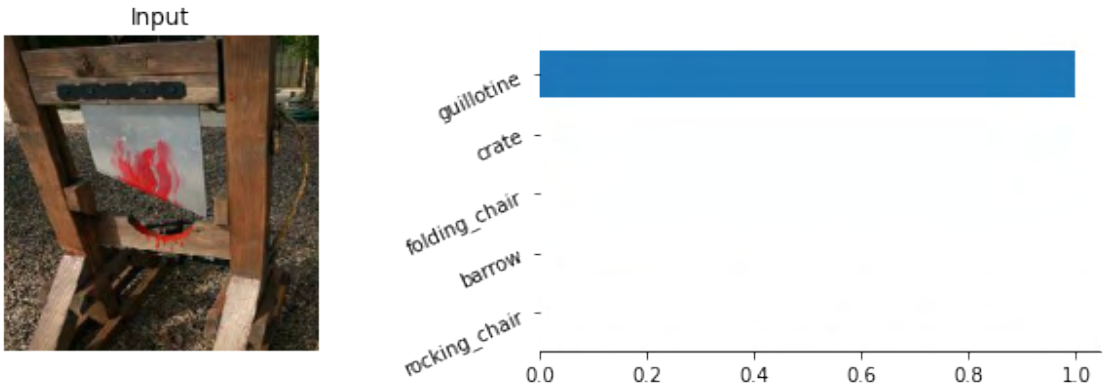

```
stole

Loading Next Image
\n03424325.JPEG
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 49ms/step
```



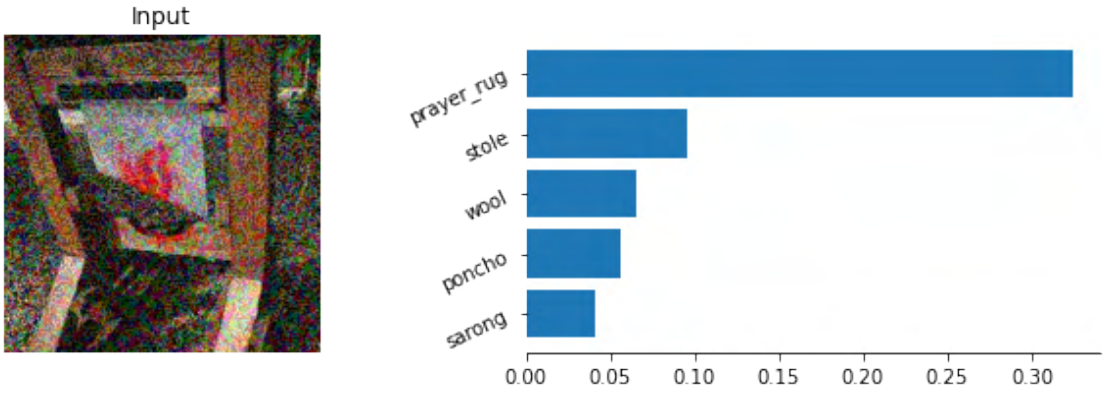

```
gasmask
1/1 [==============================] - 0s 39ms/step
```



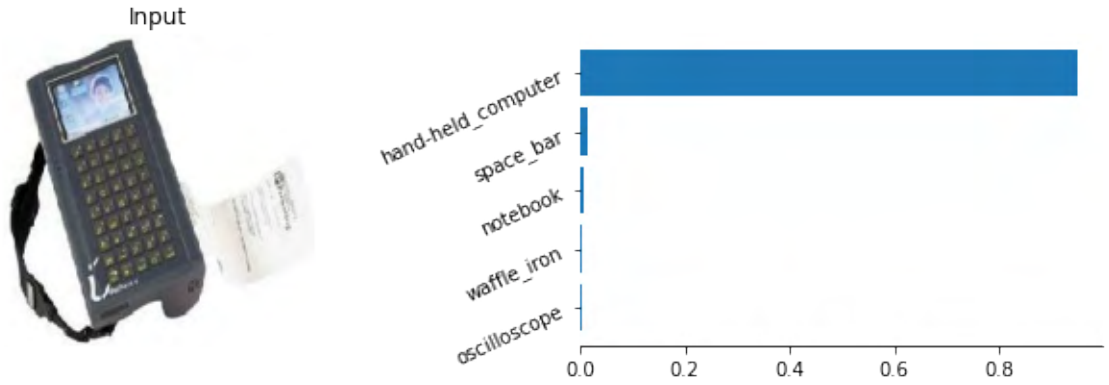

```
bubble

Loading Next Image
\n03467068.JPEG
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 49ms/step
```



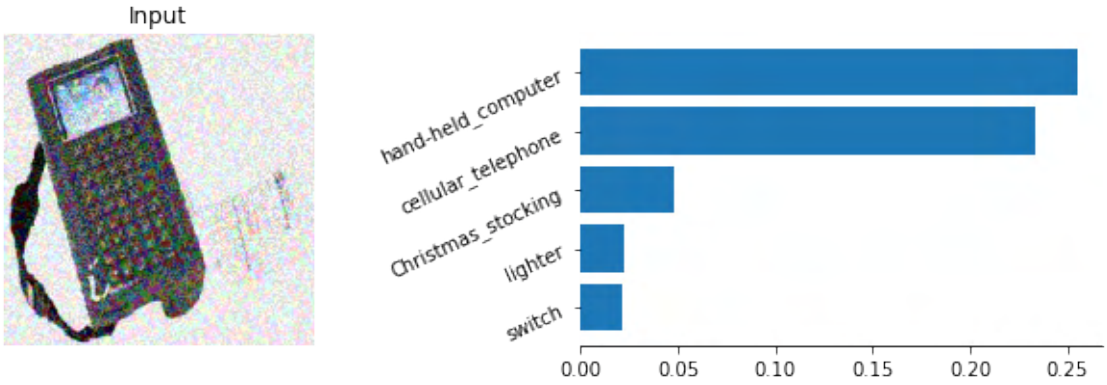

```
guillotine
1/1 [==============================] - 0s 34ms/step
```



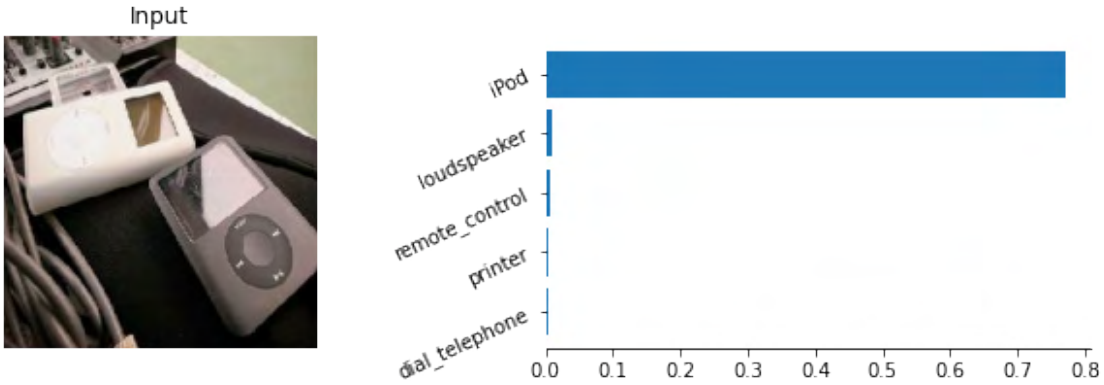

```
prayer_rug

Loading Next Image
\n03485407.JPEG
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 40ms/step
```



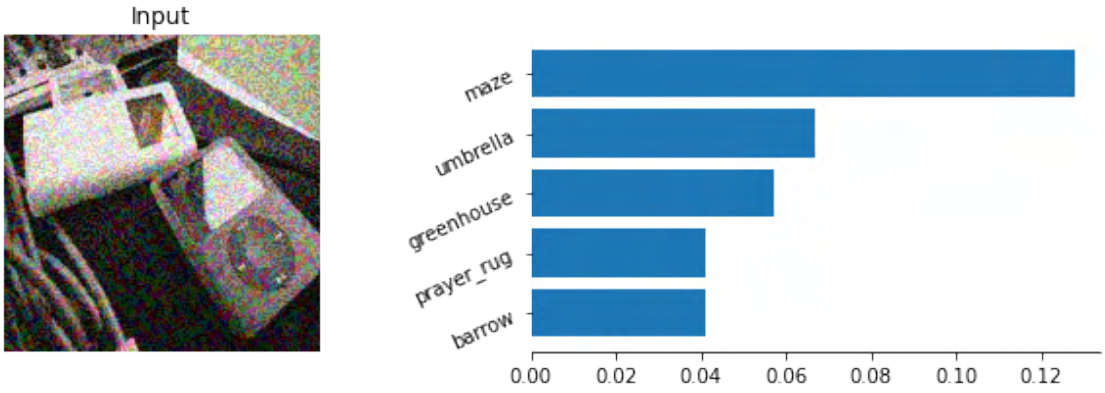

```
hand-held_computer
1/1 [==============================] - 0s 50ms/step
```



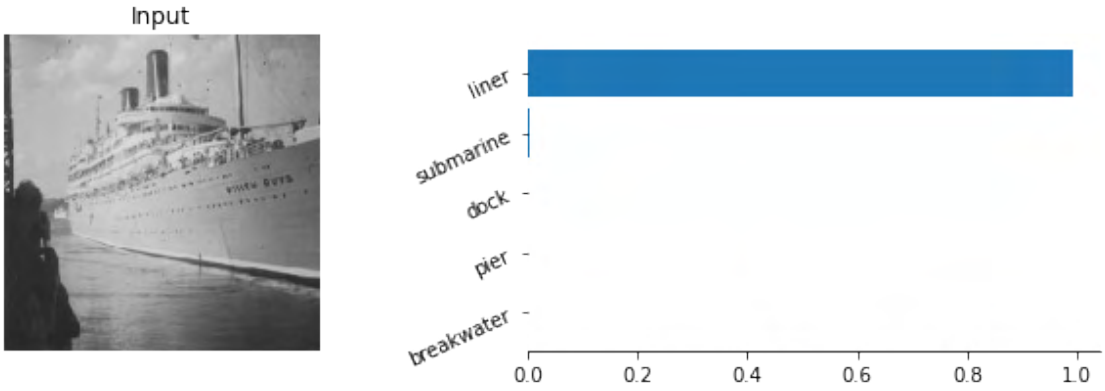

```
hand-held_computer

Loading Next Image
\n03584254.JPEG
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 45ms/step
1/1 [==============================] - 0s 44ms/step
```



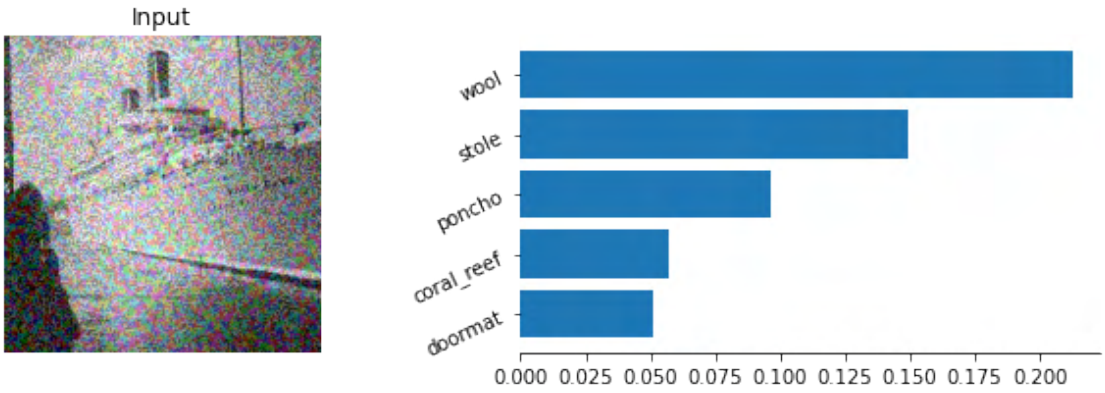

```
iPod
1/1 [==============================] - 0s 55ms/step
```



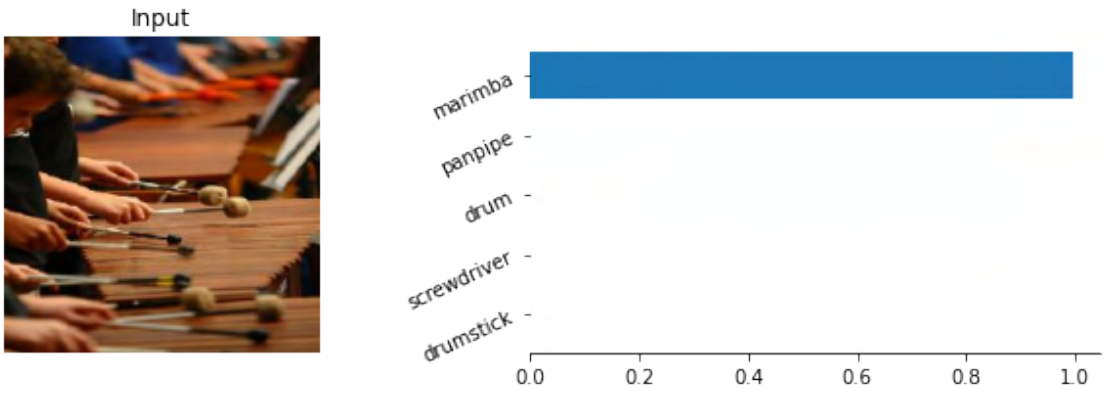

```
maze

Loading Next Image
\n03673027.JPEG
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 47ms/step
```



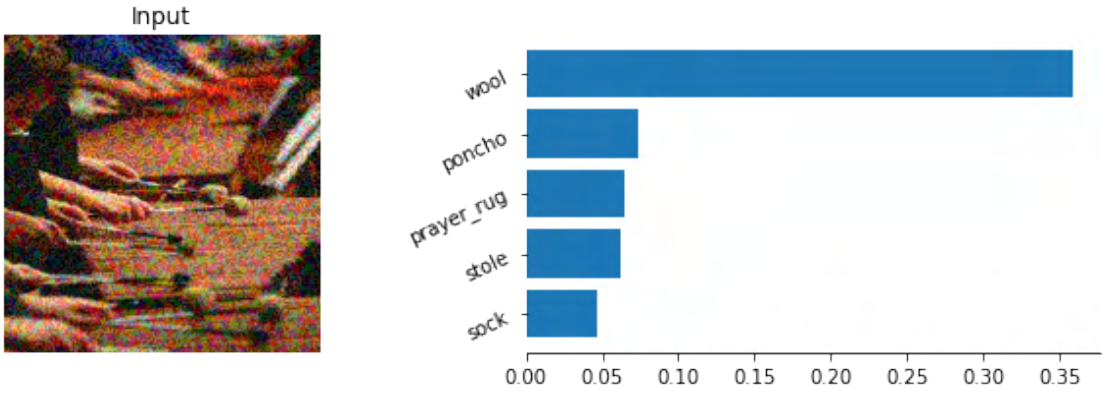

```
liner
1/1 [==============================] - 0s 49ms/step
```



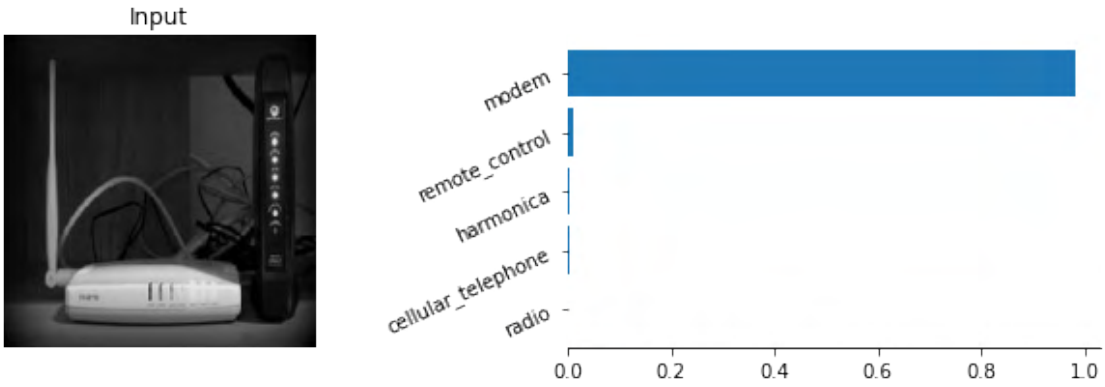

```
wool

Loading Next Image
\n03721384.JPEG
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 37ms/step
```



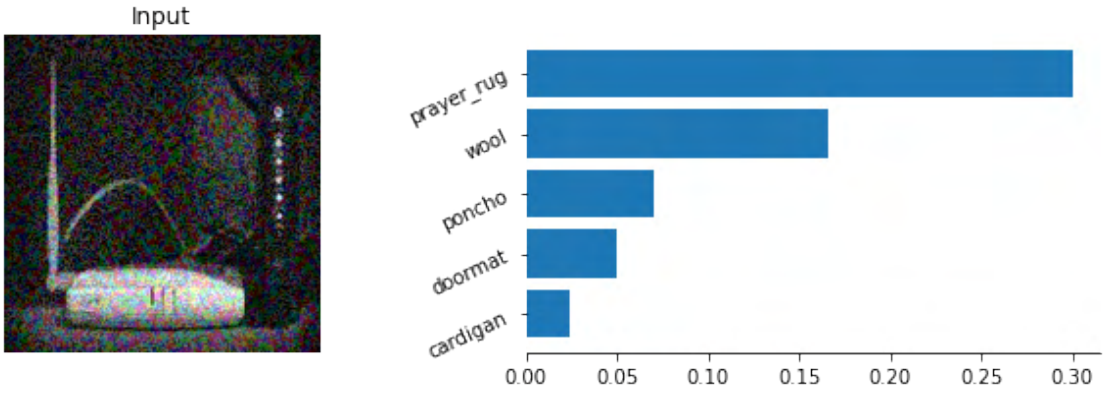

```
marimba
1/1 [==============================] - 0s 48ms/step
```



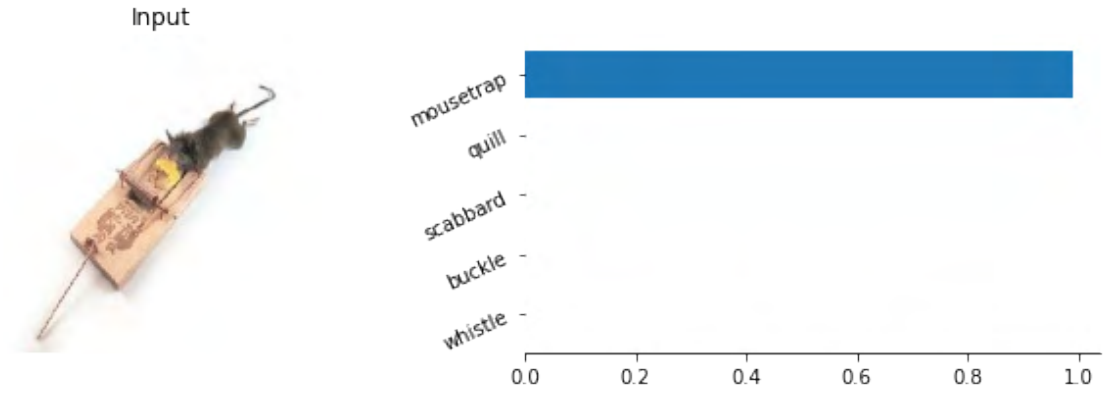

```
wool

Loading Next Image
\n03777754.JPEG
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 57ms/step
```



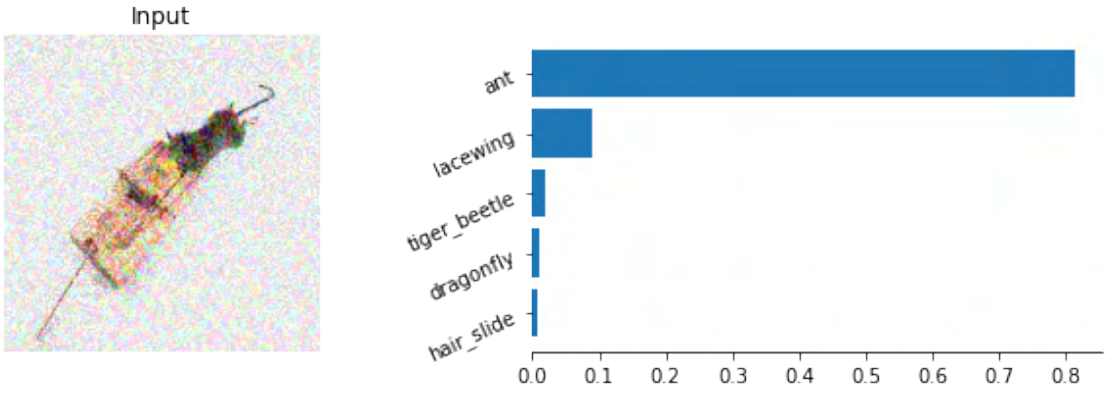

```
modem
1/1 [==============================] - 0s 53ms/step
```



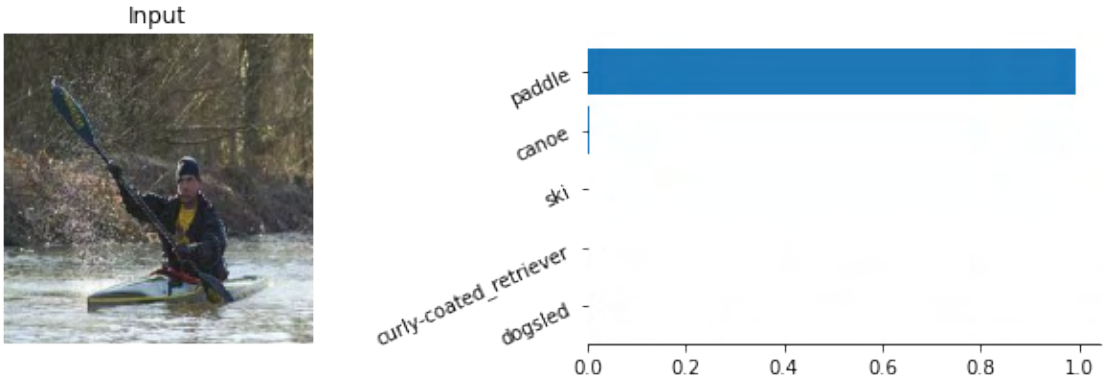

```
prayer_rug

Loading Next Image
\n03794056.JPEG
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 42ms/step
```



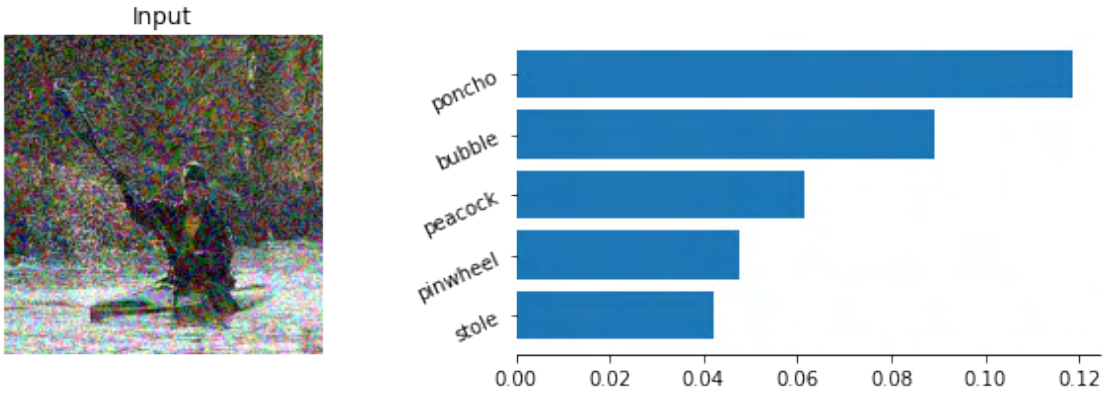

```
mousetrap
1/1 [==============================] - 0s 52ms/step
```



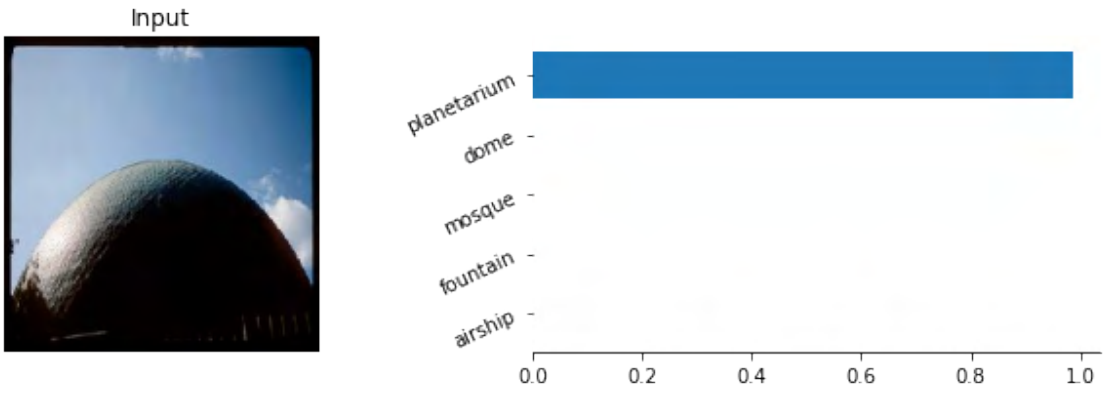

```
ant

Loading Next Image
\n03873416.JPEG
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 47ms/step
```



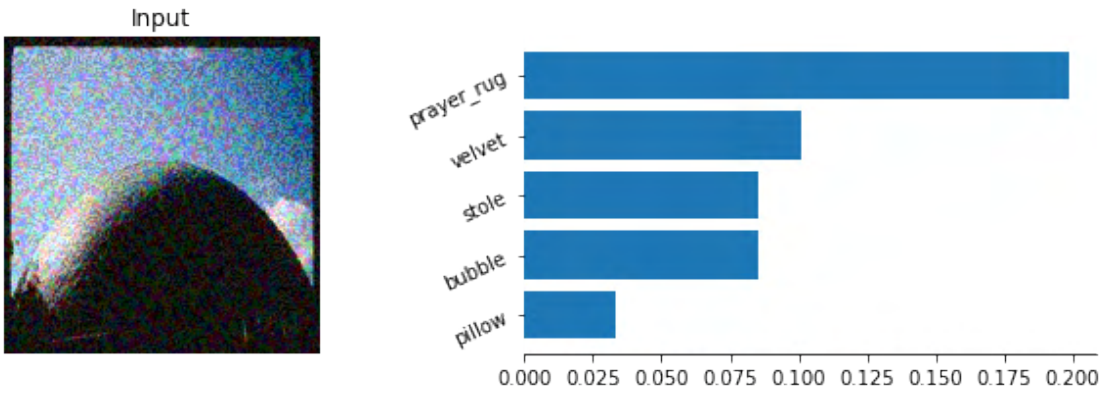

```
paddle
1/1 [==============================] - 0s 48ms/step
```



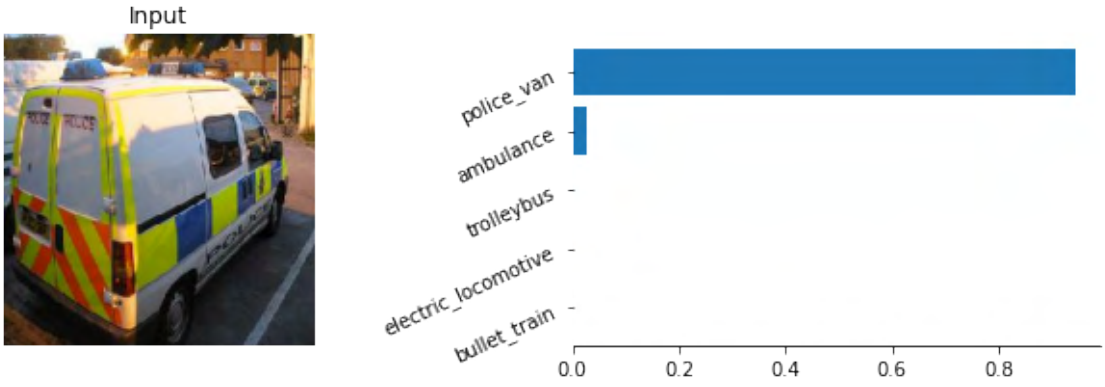

```
poncho

Loading Next Image
\n03956157.JPEG
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 38ms/step
```



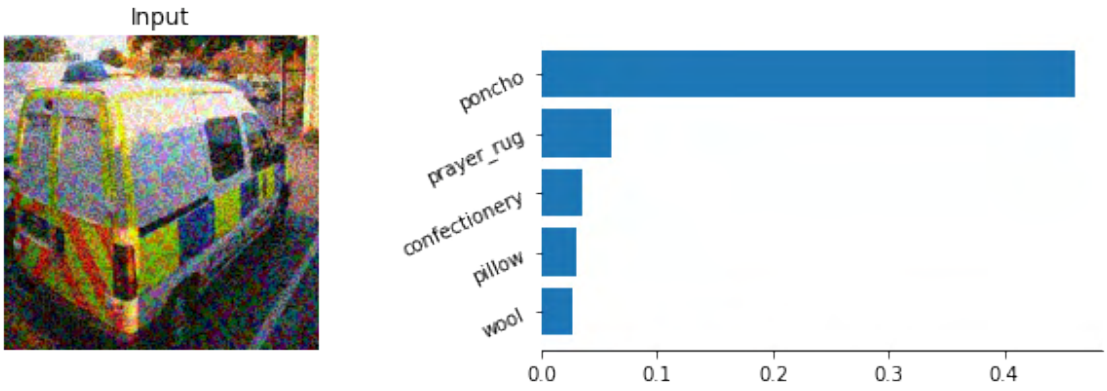

```
planetarium
1/1 [==============================] - 0s 35ms/step
```



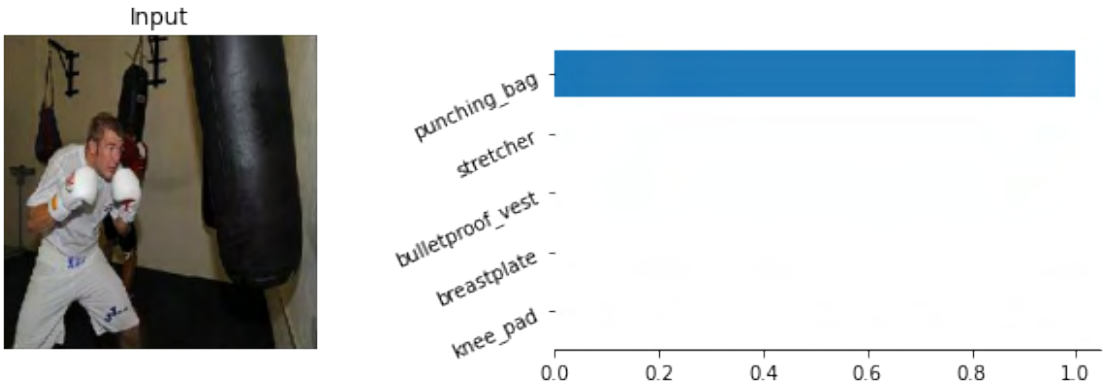

```
prayer_rug

Loading Next Image
\n03977966.JPEG
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 58ms/step
1/1 [==============================] - 0s 57ms/step
```



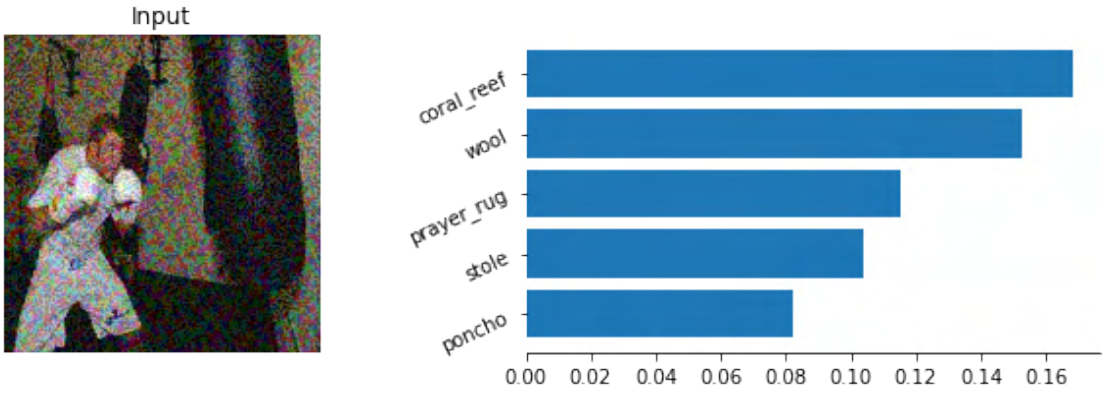

```
police_van
1/1 [==============================] - 0s 53ms/step
```



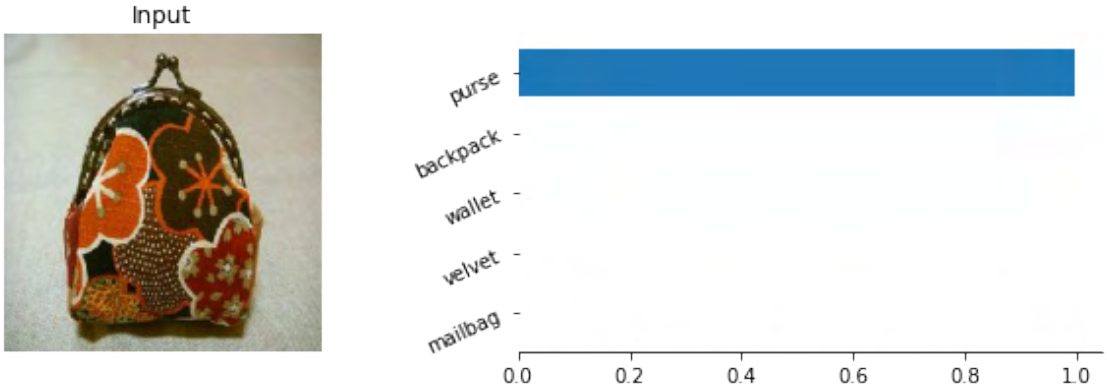

```
poncho

Loading Next Image
\n04023962.JPEG
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 41ms/step
```



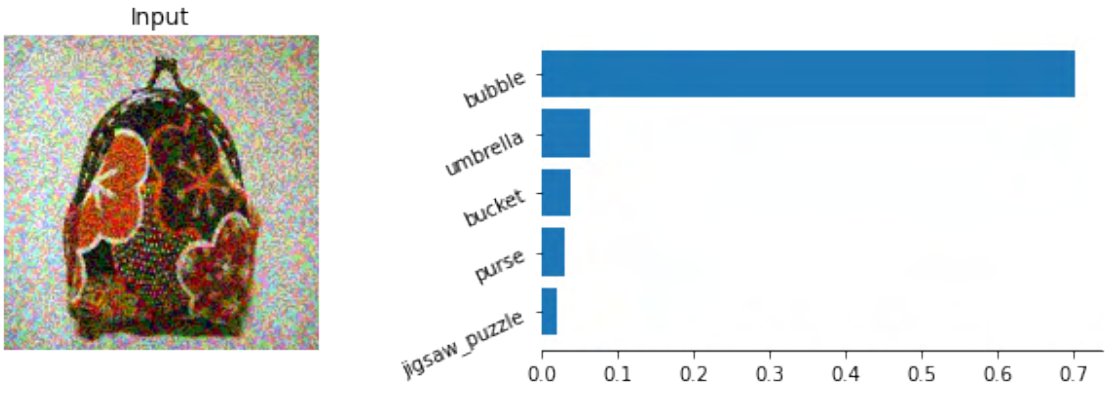

```
punching_bag
1/1 [==============================] - 0s 47ms/step
```



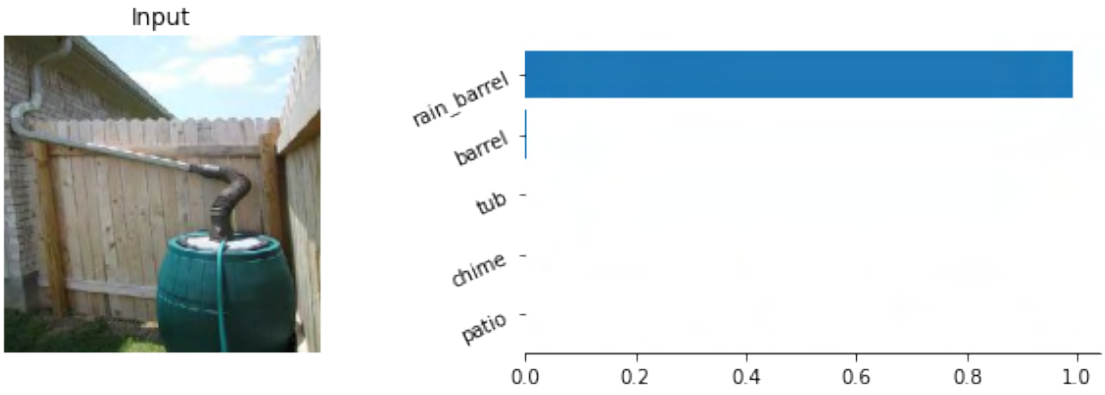

```
coral_reef

Loading Next Image
\n04026417.JPEG
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 46ms/step
```



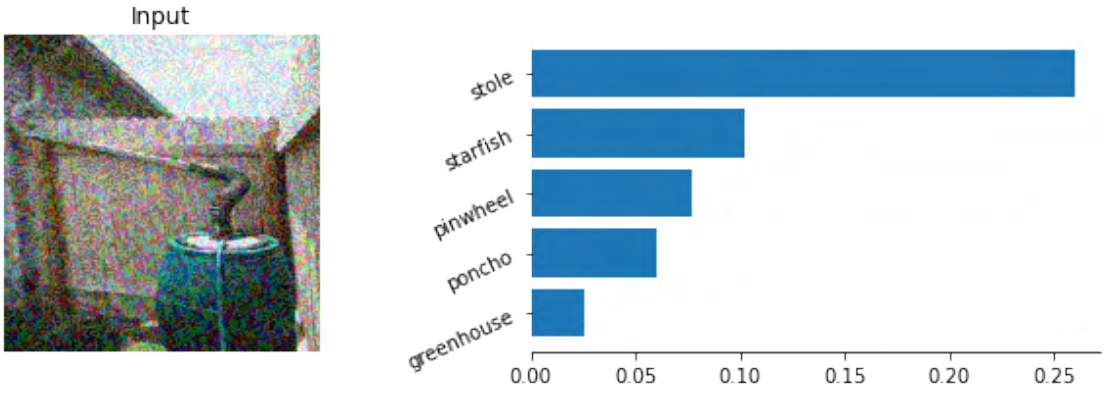

```
purse
1/1 [==============================] - 0s 49ms/step
```



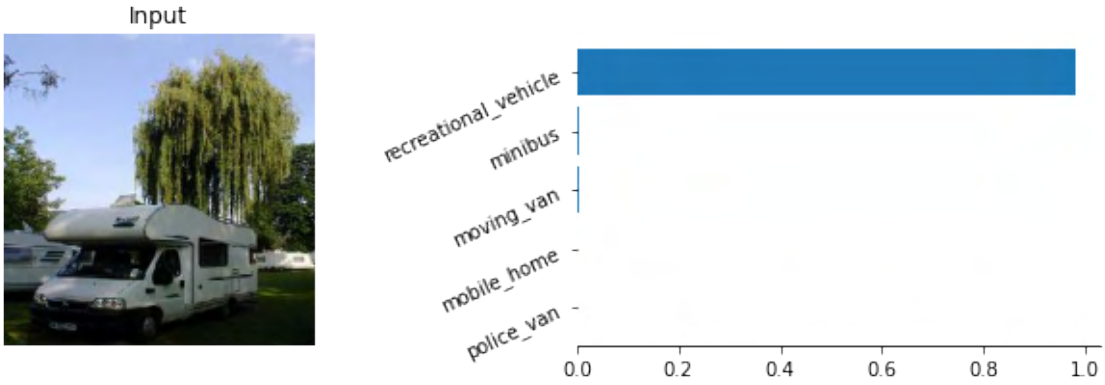

```
bubble

Loading Next Image
\n04049303.JPEG
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 43ms/step
```



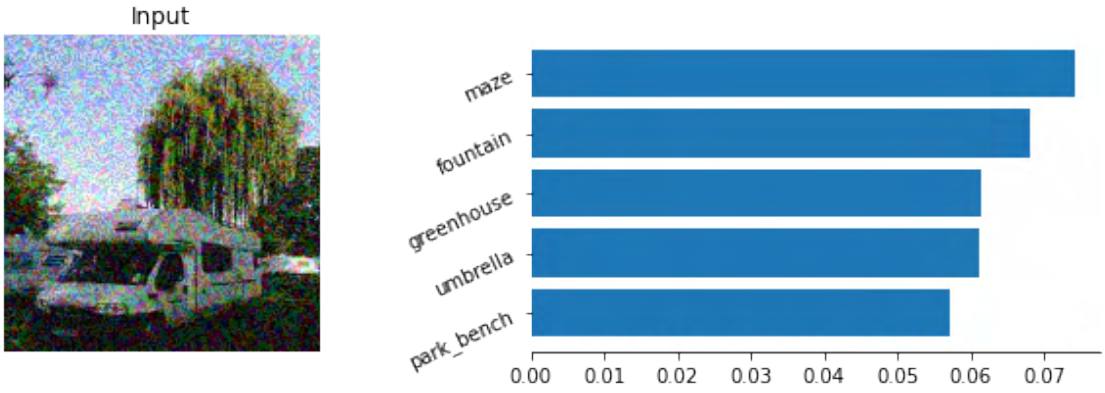

```
rain_barrel
1/1 [==============================] - 0s 34ms/step
```



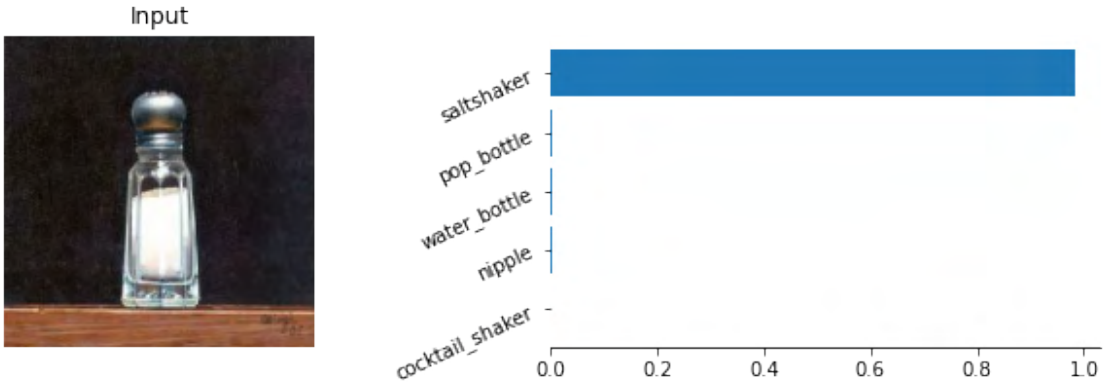

```
stole

Loading Next Image
\n04065272.JPEG
1/1 [==============================] - 0s 61ms/step
1/1 [==============================] - 0s 58ms/step
1/1 [==============================] - 0s 44ms/step
```



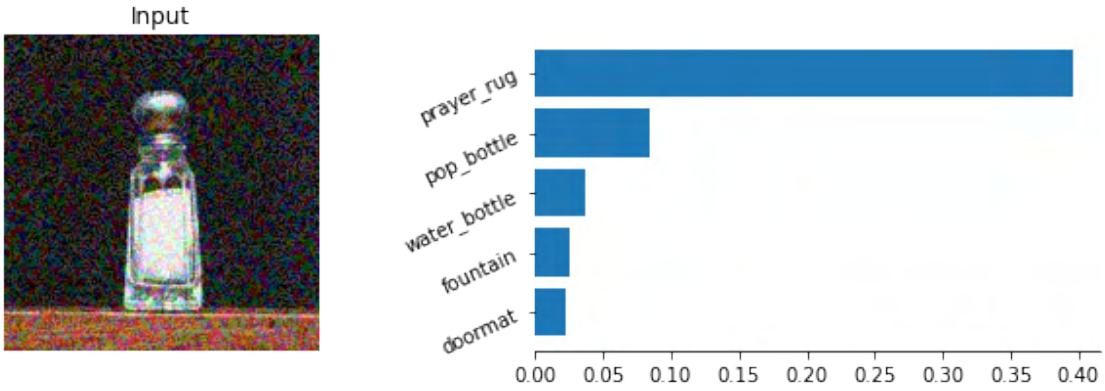

```
recreational_vehicle
1/1 [==============================] - 0s 57ms/step
```



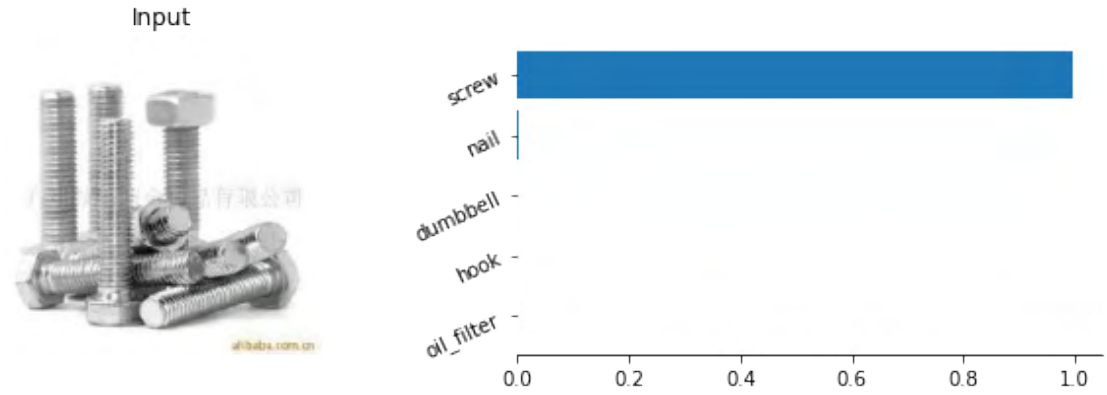

```
maze

Loading Next Image
\n04131690.JPEG
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 40ms/step
```



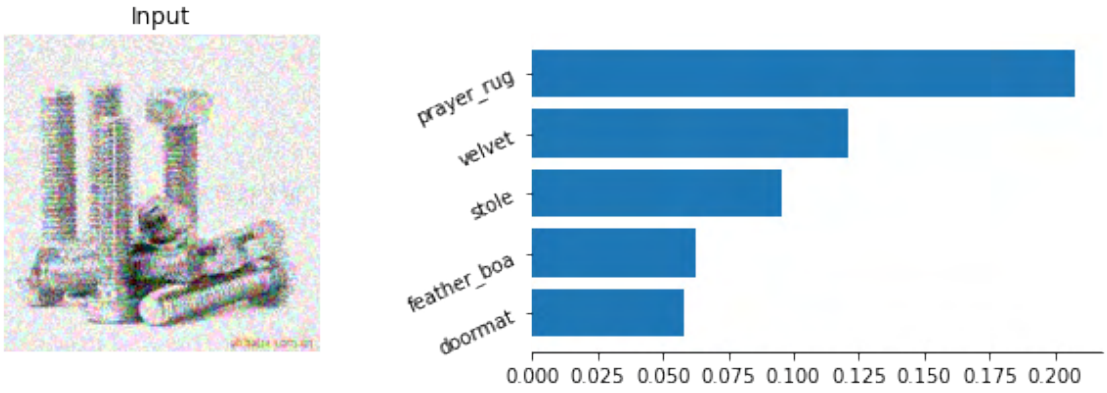

```
saltshaker
1/1 [==============================] - 0s 47ms/step
```



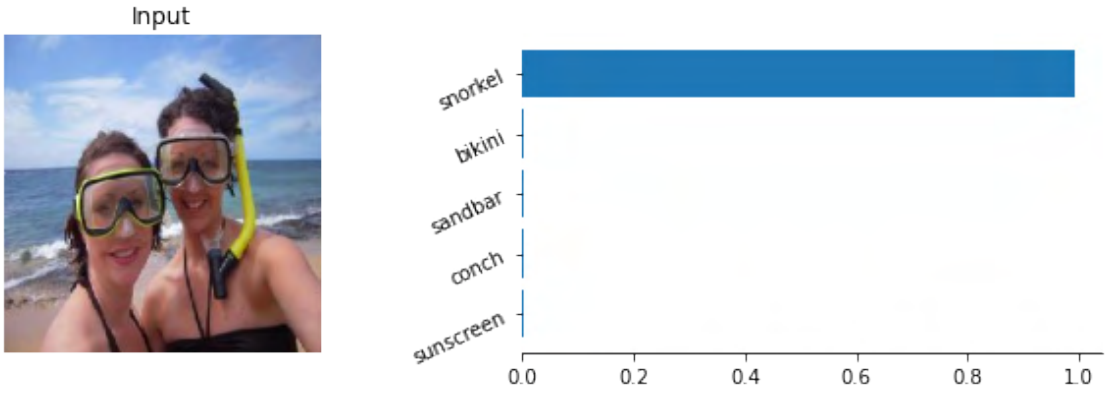

```
prayer_rug

Loading Next Image
\n04153751.JPEG
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 58ms/step
1/1 [==============================] - 0s 42ms/step
```



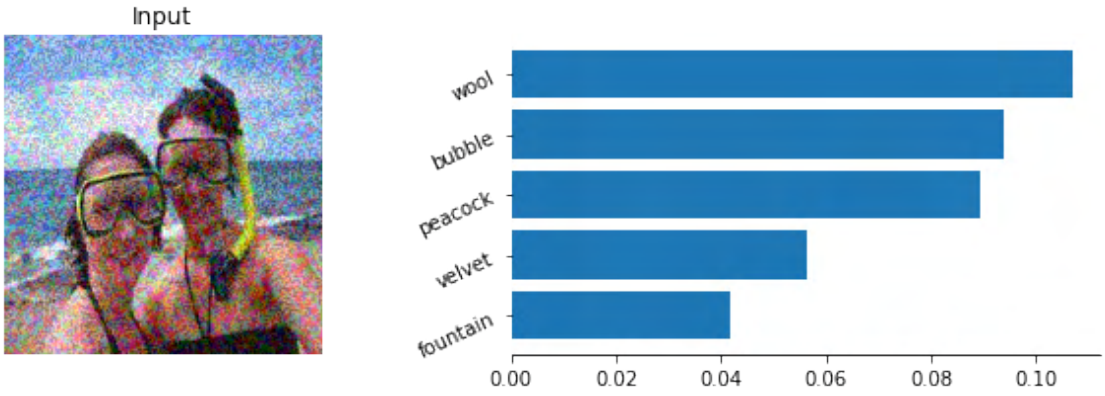

```
screw
1/1 [==============================] - 0s 38ms/step
```



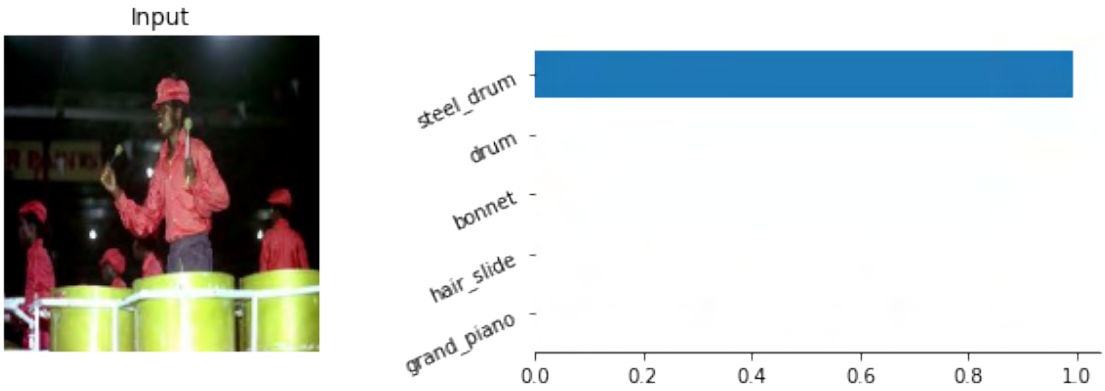

```
prayer_rug

Loading Next Image
\n04251144.JPEG
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 44ms/step
```



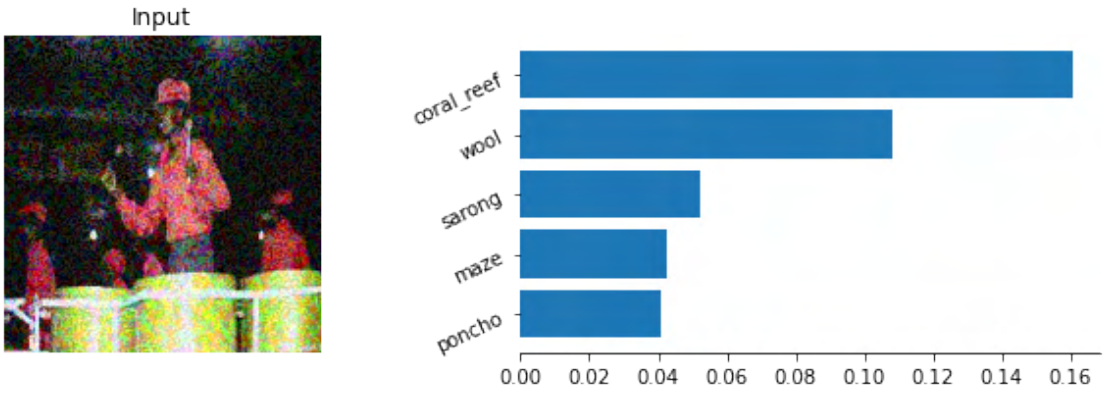

```
snorkel
1/1 [==============================] - 0s 55ms/step
```



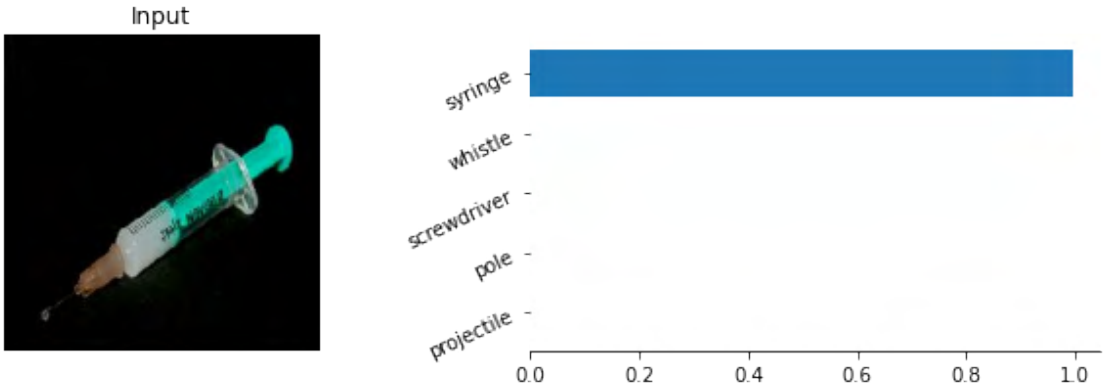

```
wool

Loading Next Image
\n04311174.JPEG
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 49ms/step
```



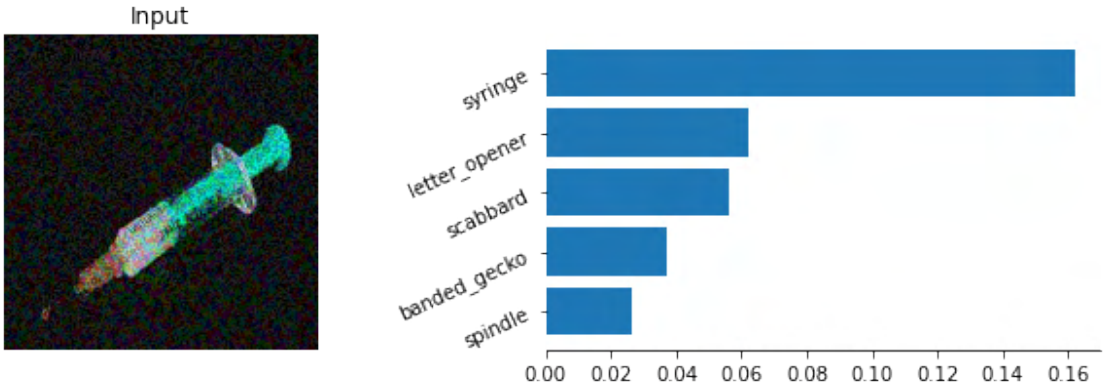

```
steel_drum
1/1 [==============================] - 0s 35ms/step
```



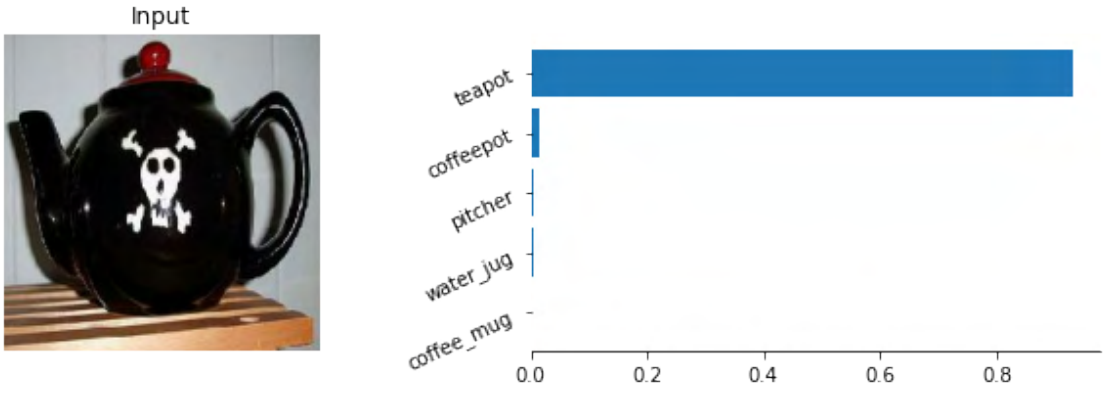

```
coral_reef

Loading Next Image
\n04376876.JPEG
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 61ms/step
```



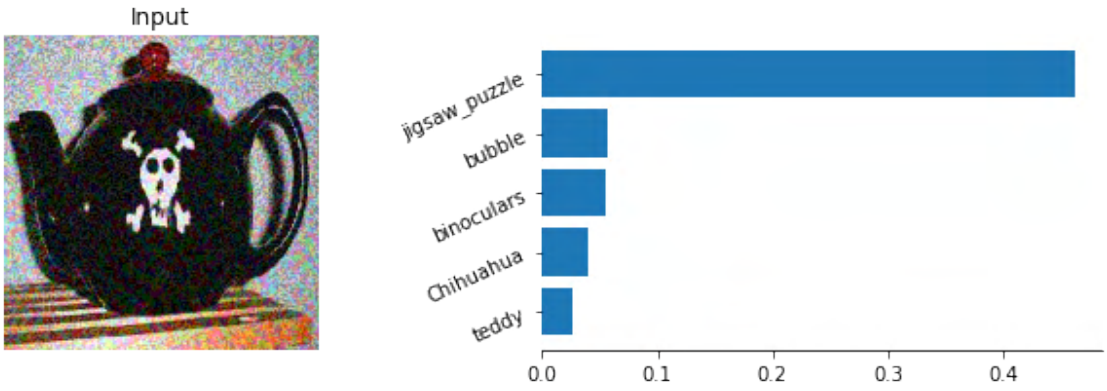

```
syringe
1/1 [==============================] - 0s 37ms/step
```



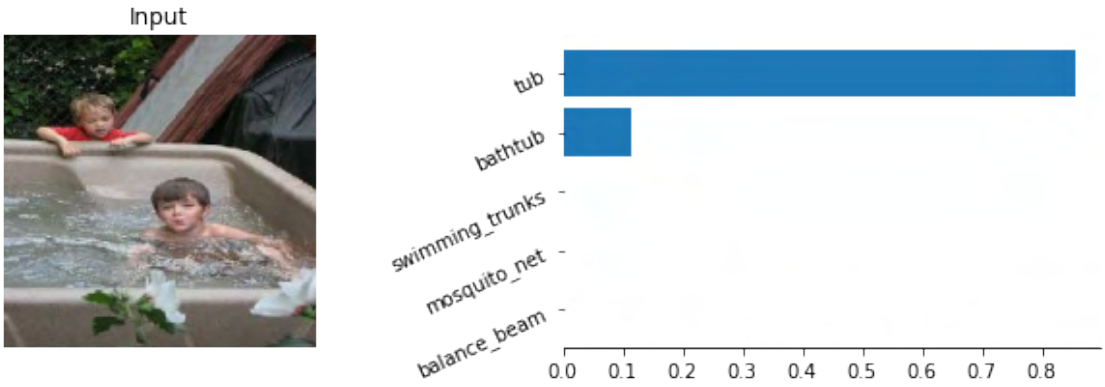

```
syringe

Loading Next Image
\n04398044.JPEG
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 37ms/step
```



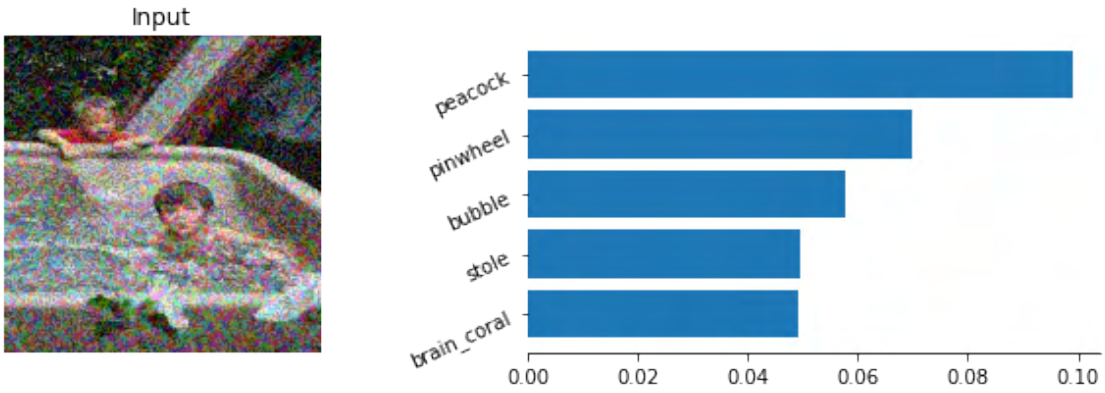

```
teapot
1/1 [==============================] - 0s 52ms/step
```



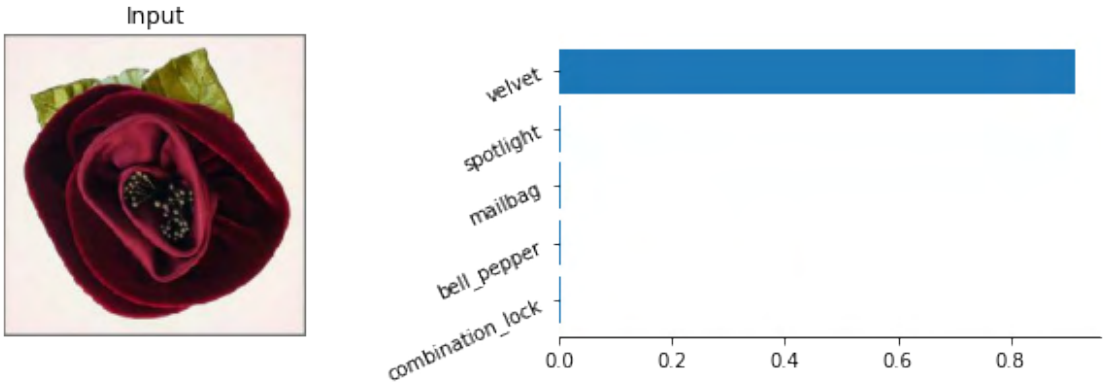

```
jigsaw_puzzle

Loading Next Image
\n04493381.JPEG
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 36ms/step
```



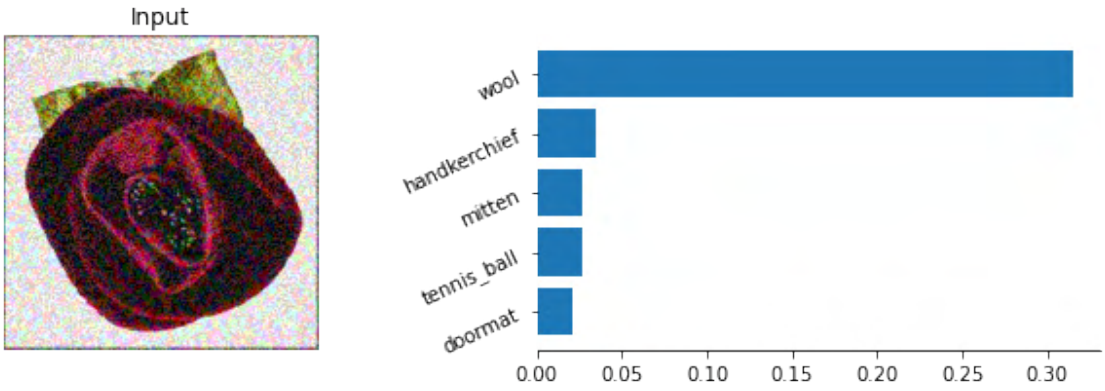

```
tub
1/1 [==============================] - 0s 53ms/step
```



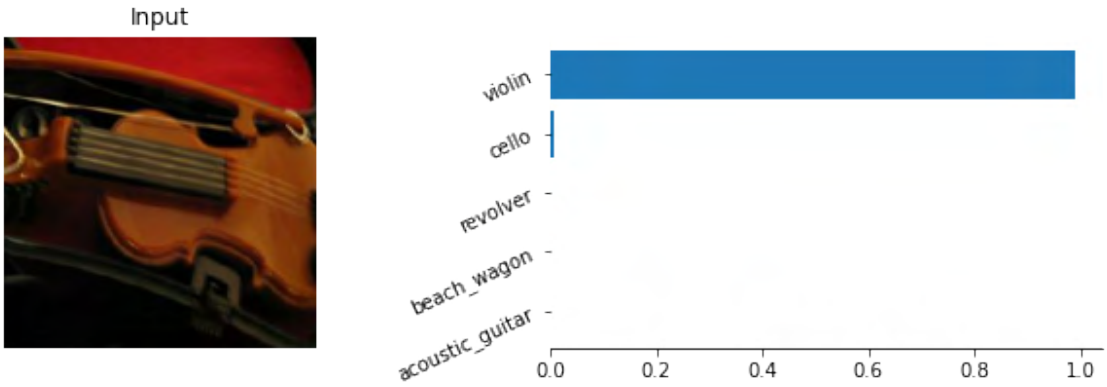

```
peacock

Loading Next Image
\n04525038.JPEG
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 36ms/step
```



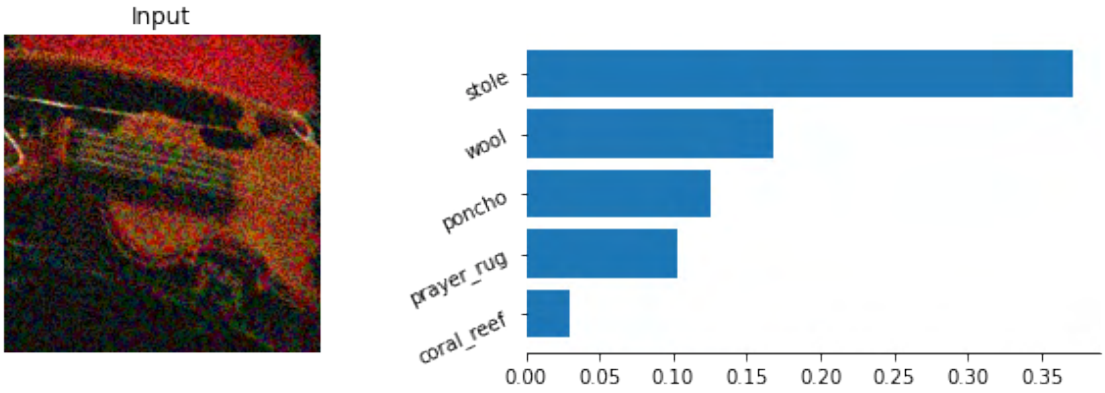

```
velvet
1/1 [==============================] - 0s 49ms/step
```



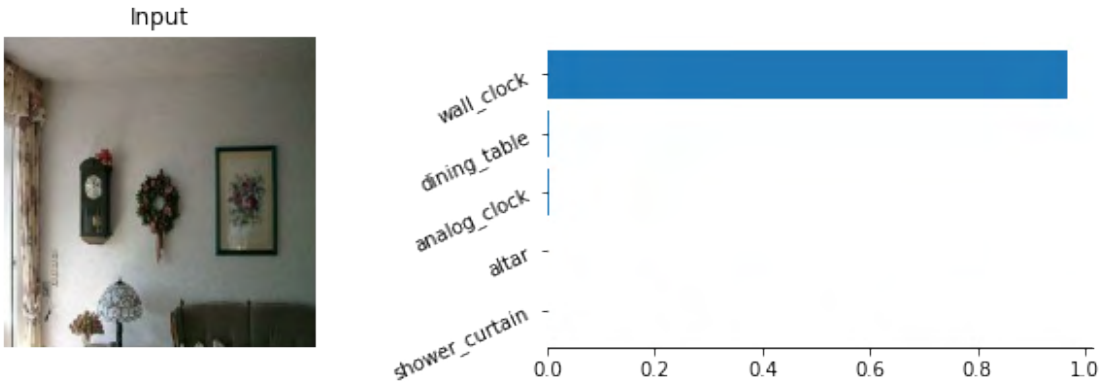

```
wool

Loading Next Image
\n04536866.JPEG
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 58ms/step
```



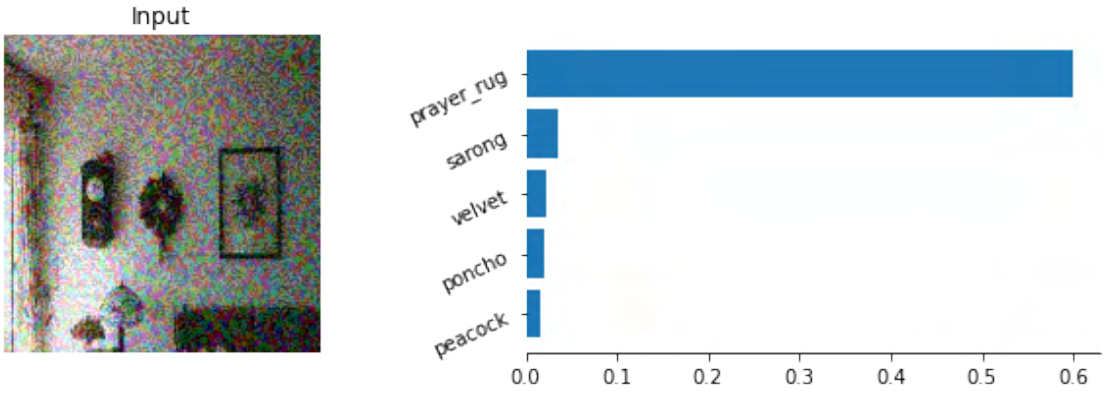

```
violin
1/1 [==============================] - 0s 51ms/step
```



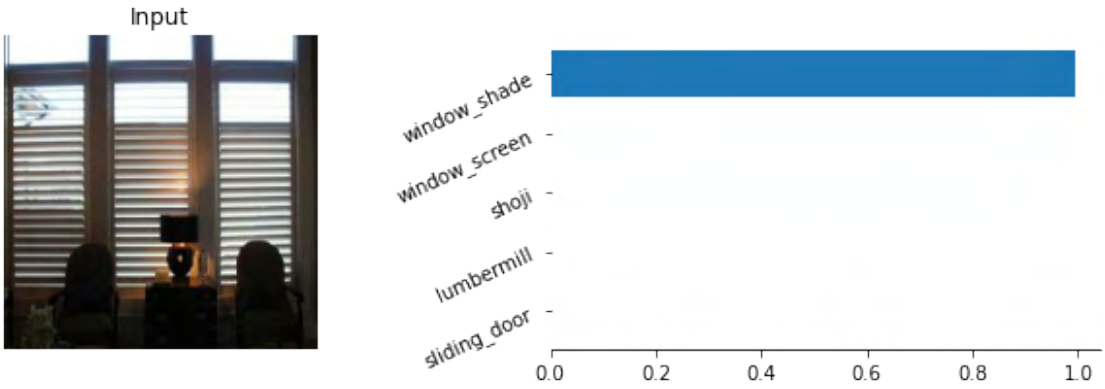

```
stole

Loading Next Image
\n04548280.JPEG
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 44ms/step
```



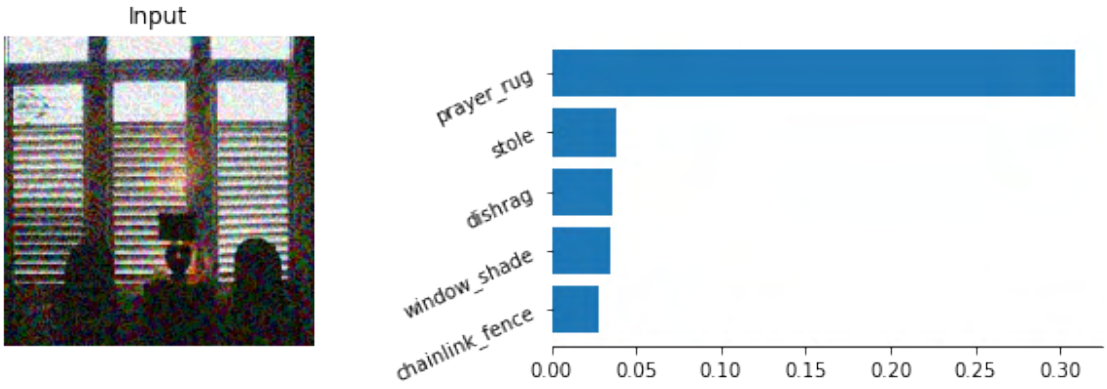

```
wall_clock
1/1 [==============================] - 0s 34ms/step
```



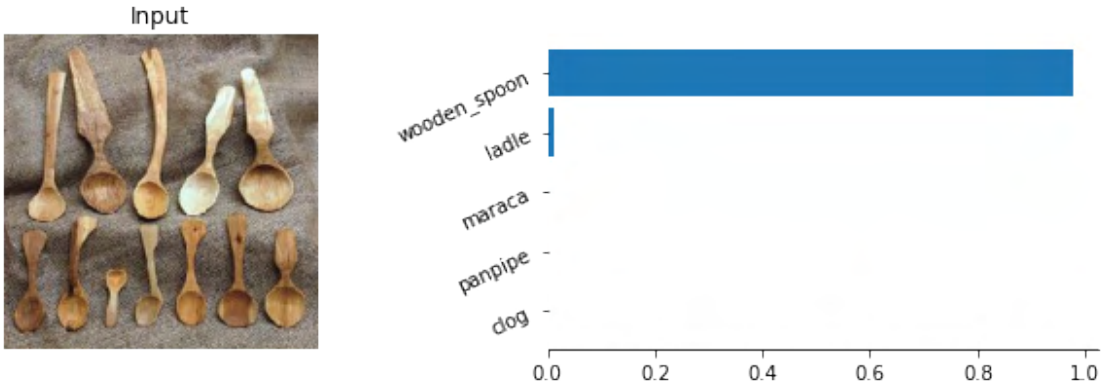

```
prayer_rug

Loading Next Image
\n04590129.JPEG
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 55ms/step
```



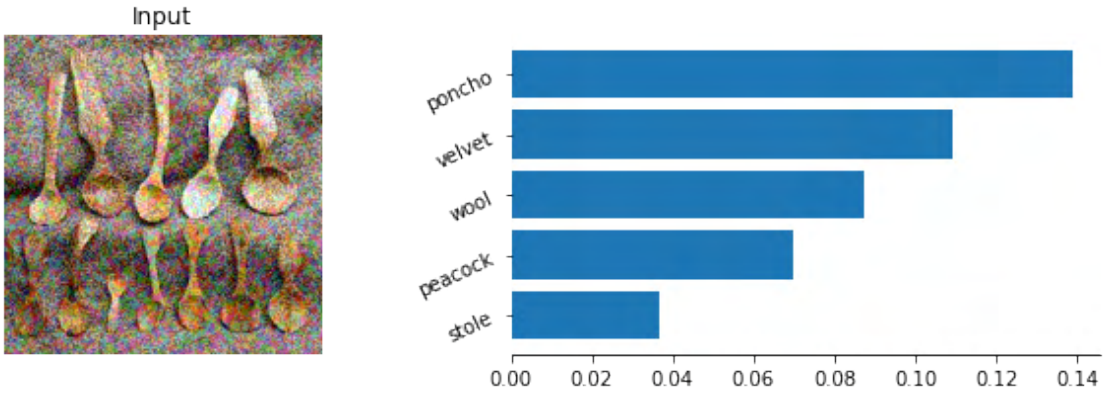

```
window_shade
1/1 [==============================] - 0s 57ms/step
```



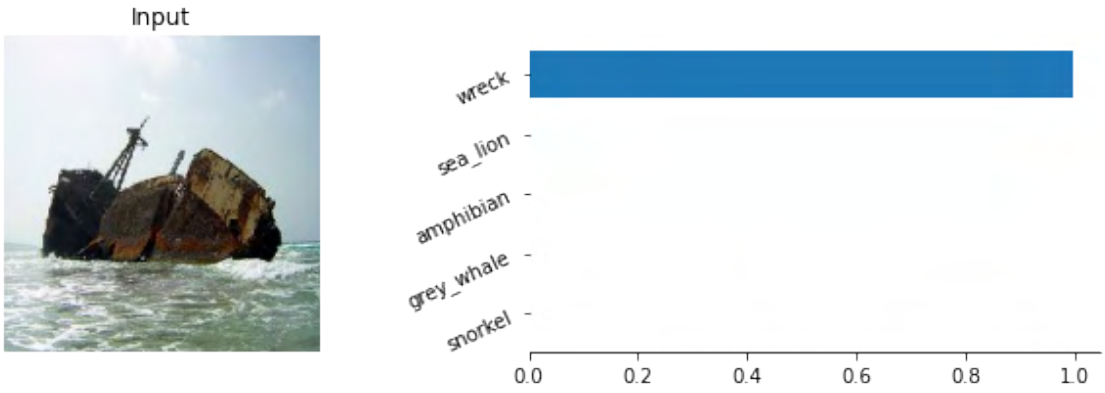

```
prayer_rug

Loading Next Image
\n04597913.JPEG
1/1 [==============================] - 0s 45ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 42ms/step
```



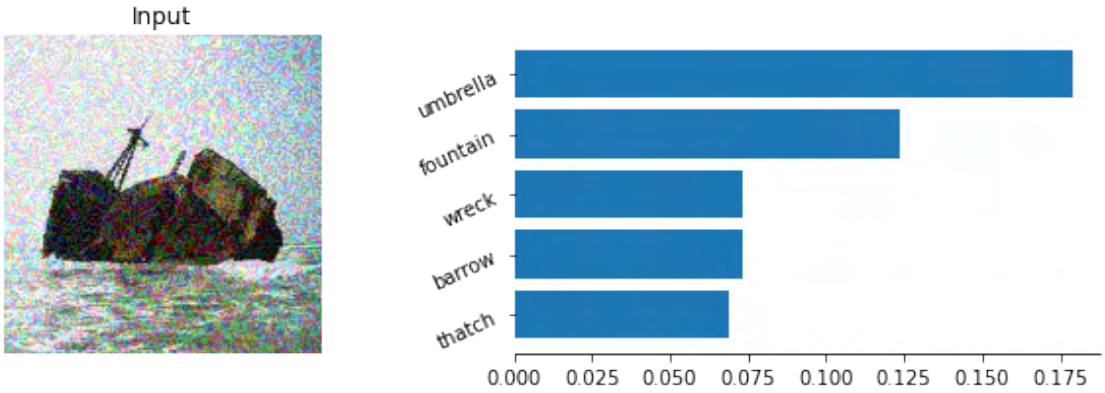

```
wooden_spoon
1/1 [==============================] - 0s 50ms/step
```



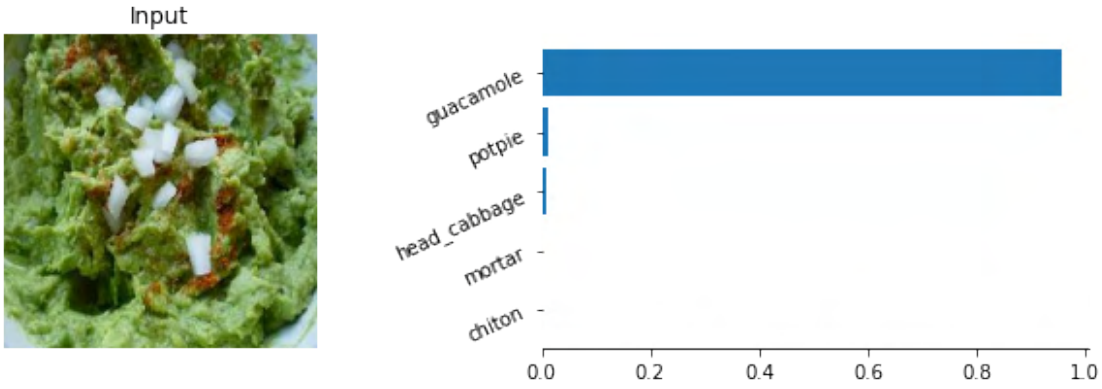

```
poncho

Loading Next Image
\n04606251.JPEG
1/1 [==============================] - 0s 58ms/step
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 54ms/step
```



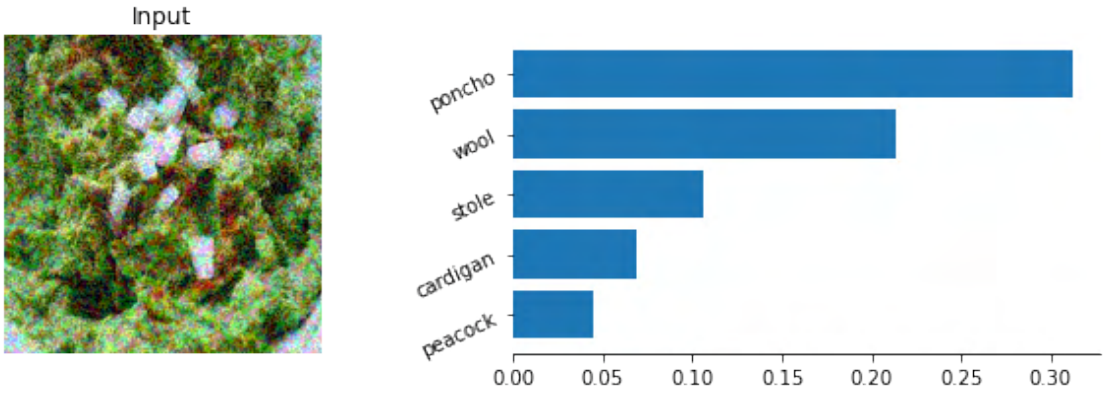

```
wreck
1/1 [==============================] - 0s 52ms/step
```



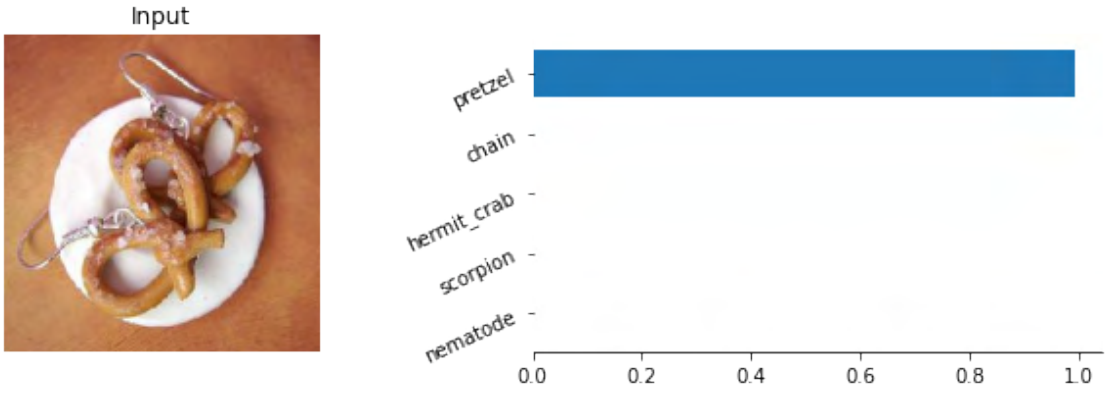

```
umbrella

Loading Next Image
\n07583066.JPEG
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 46ms/step
```



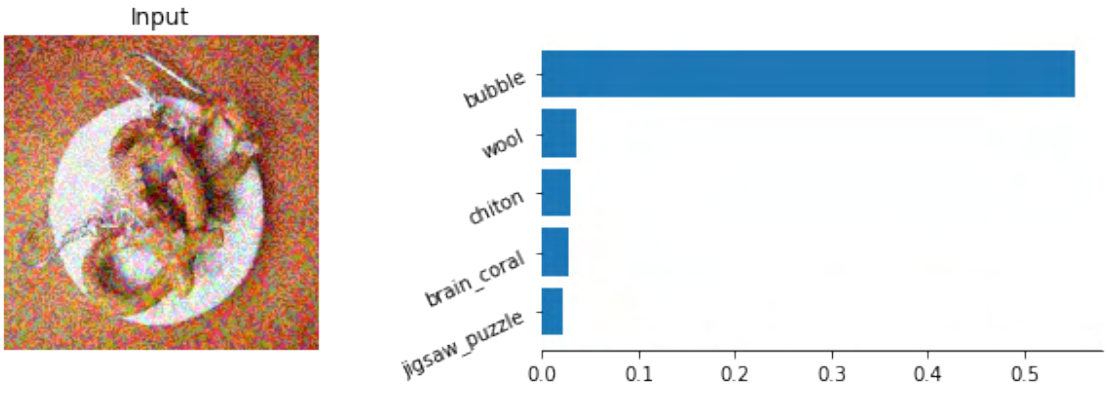

```
guacamole
1/1 [==============================] - 0s 37ms/step
```



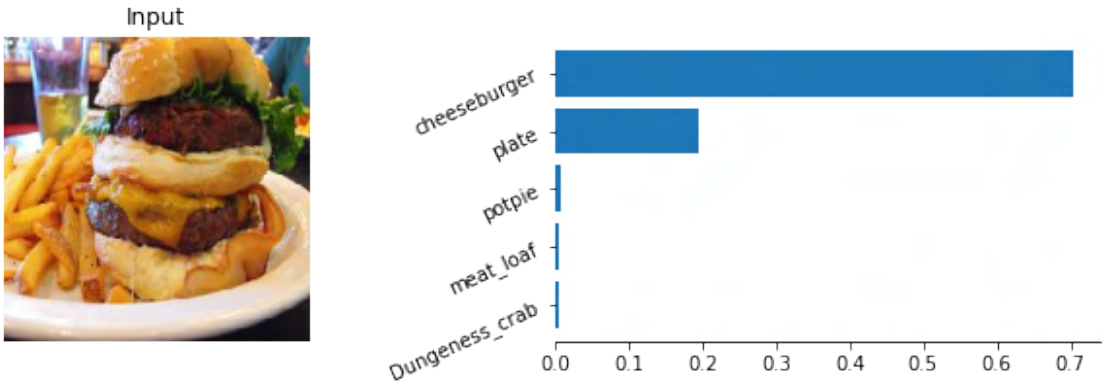

```
poncho

Loading Next Image
\n07695742.JPEG
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 38ms/step
```



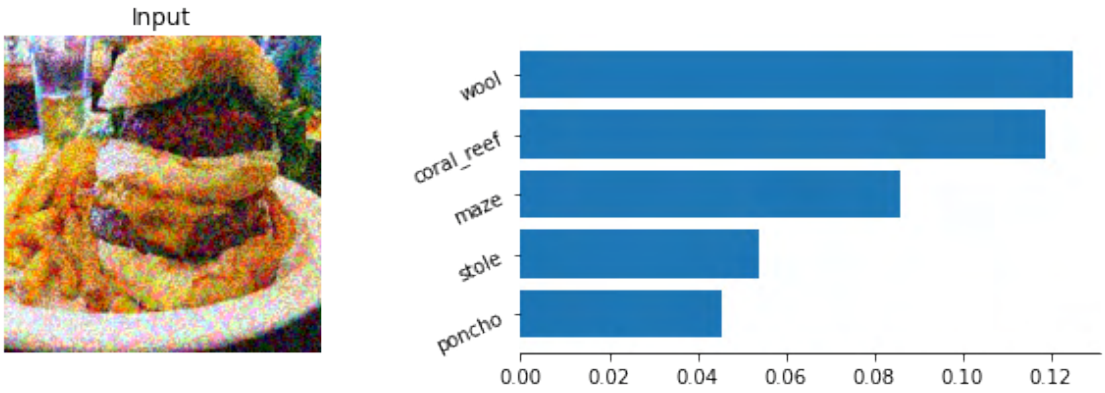

```
pretzel
1/1 [==============================] - 0s 50ms/step
```



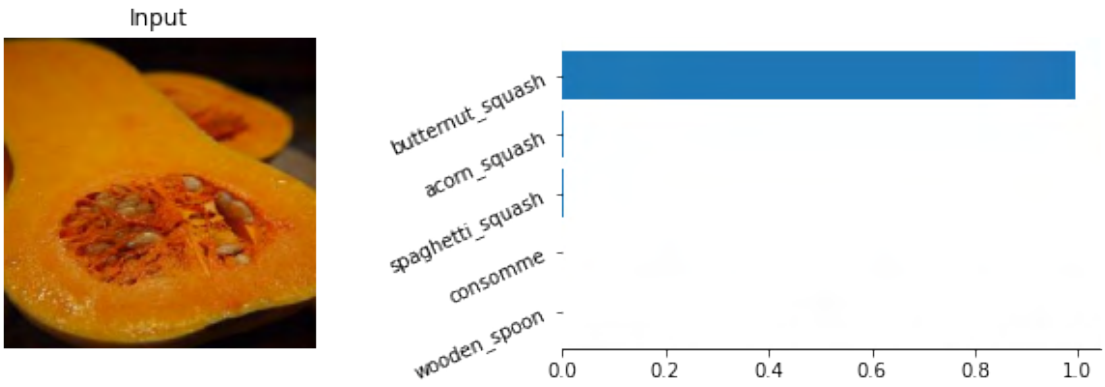

```
bubble

Loading Next Image
\n07697313.JPEG
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 53ms/step
```



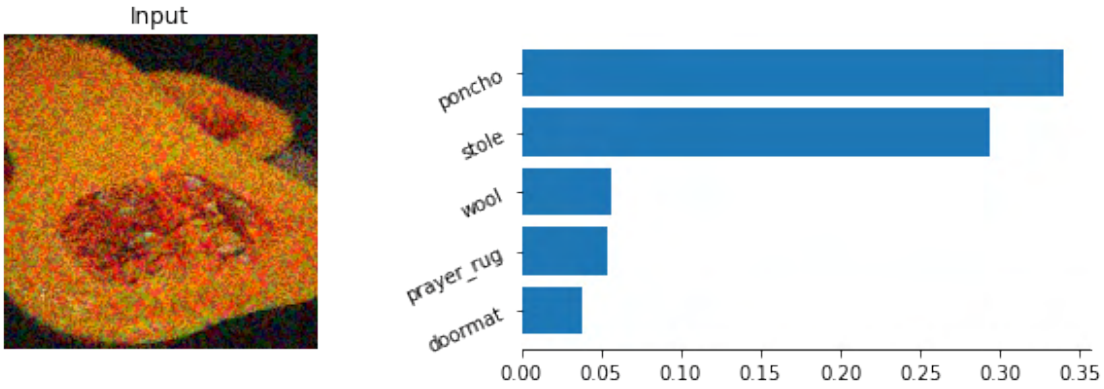

```
cheeseburger
1/1 [==============================] - 0s 48ms/step
```



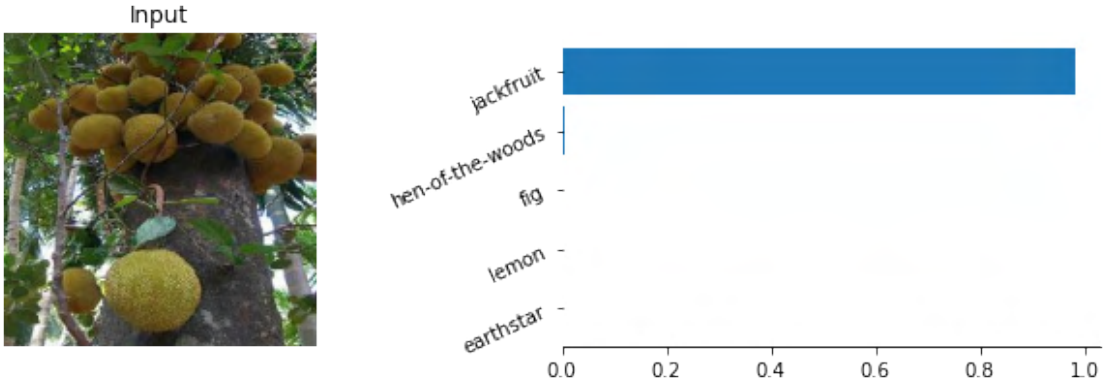

```
wool

Loading Next Image
\n07717556.JPEG
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 39ms/step
```



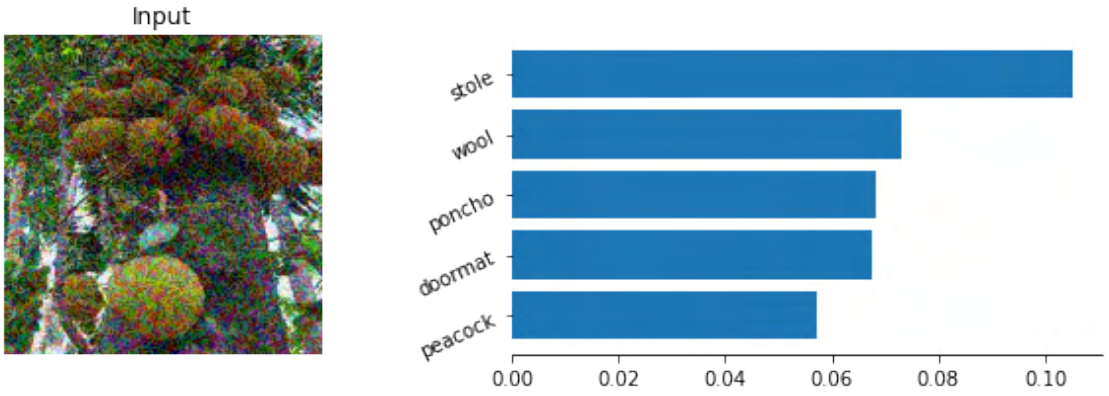

```
butternut_squash
1/1 [==============================] - 0s 48ms/step
```



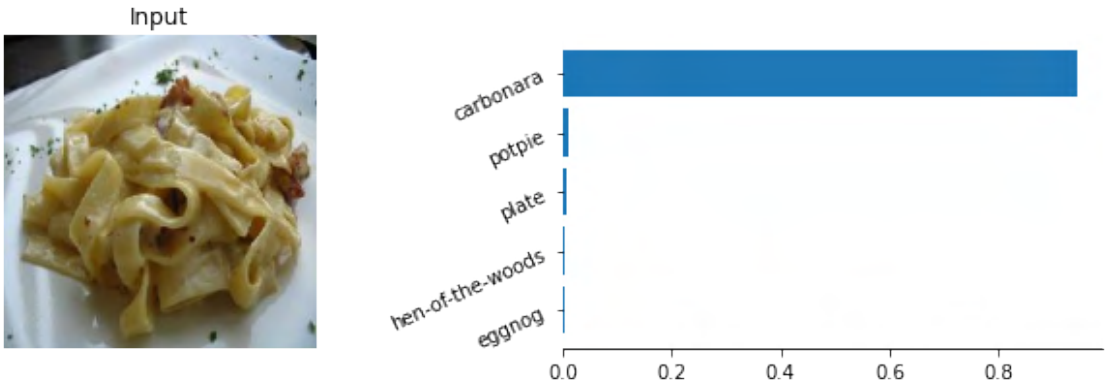

```
poncho

Loading Next Image
\n07754684.JPEG
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 38ms/step
```



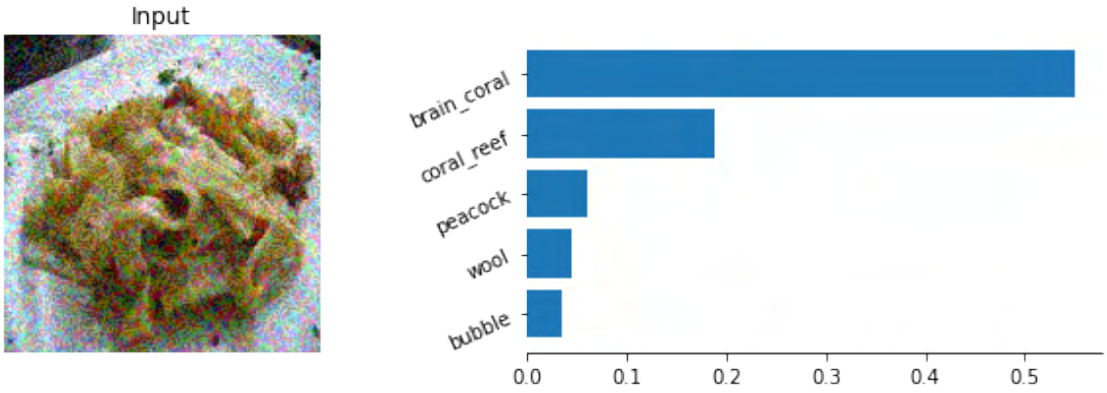

```
jackfruit
1/1 [==============================] - 0s 48ms/step
```



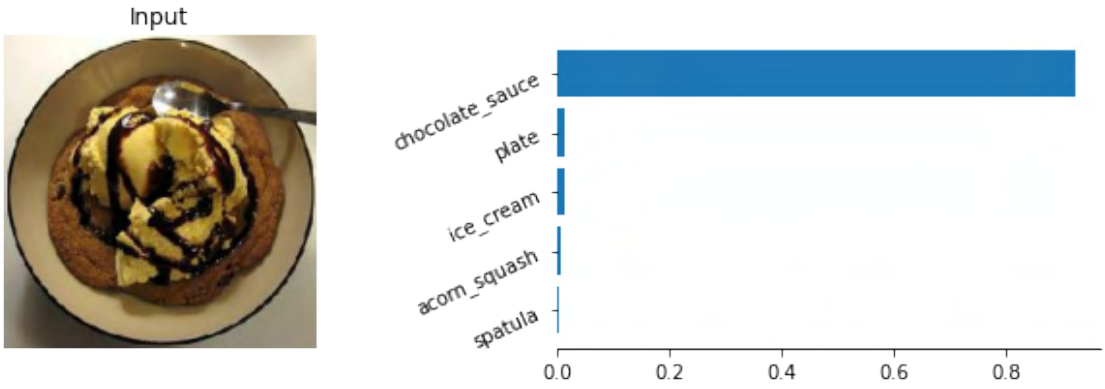

```
stole

Loading Next Image
\n07831146.JPEG
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 56ms/step
```



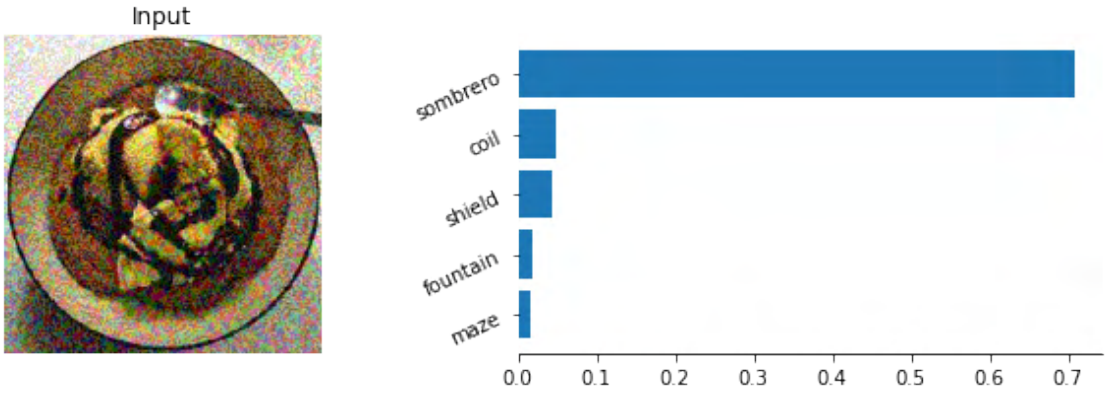

```
carbonara
1/1 [==============================] - 0s 51ms/step
```



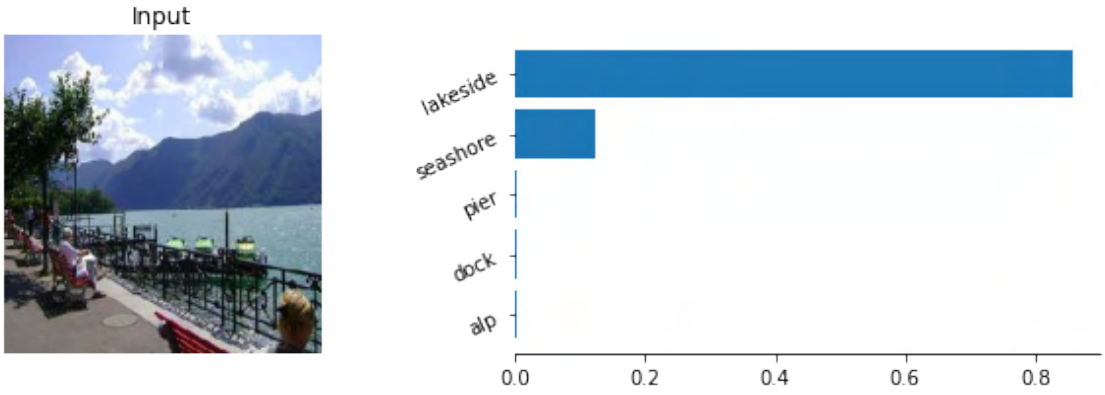

```
brain_coral

Loading Next Image
\n07836838.JPEG
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 47ms/step
```



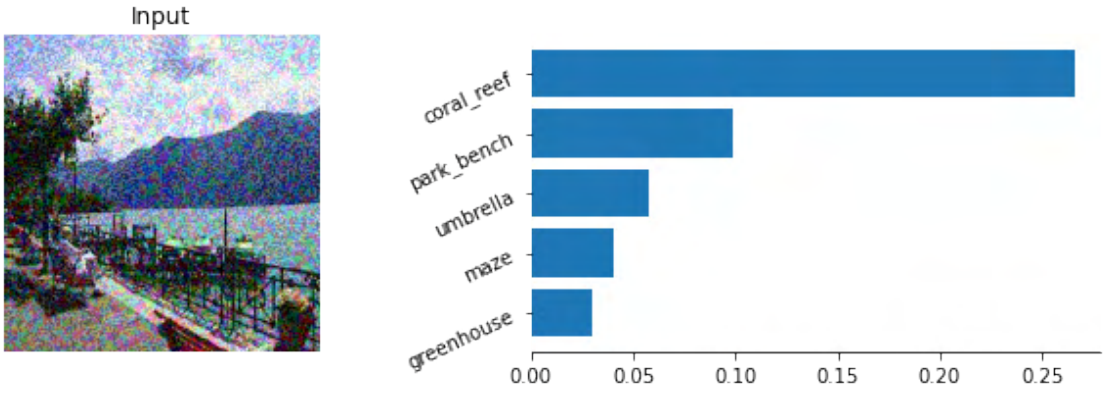

```
chocolate_sauce
1/1 [==============================] - 0s 70ms/step
```



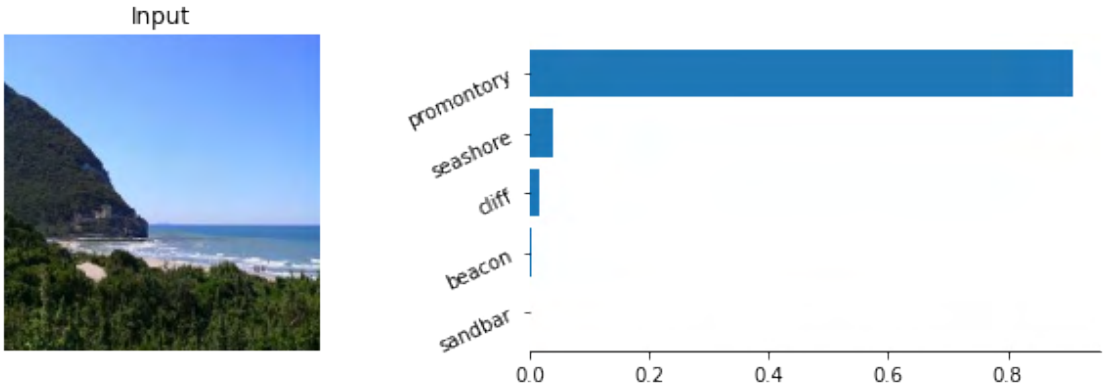

```
sombrero

Loading Next Image
\n09332890.JPEG
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 50ms/step
```



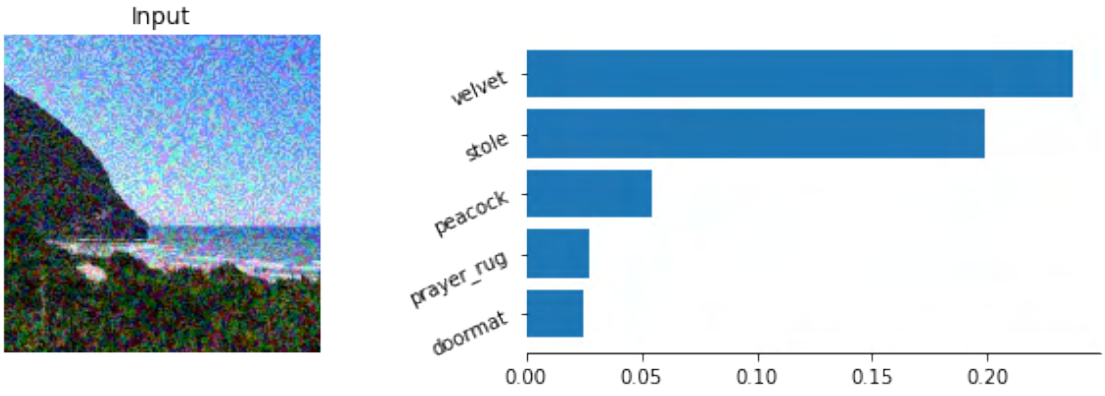

```
lakeside
1/1 [==============================] - 0s 55ms/step
```



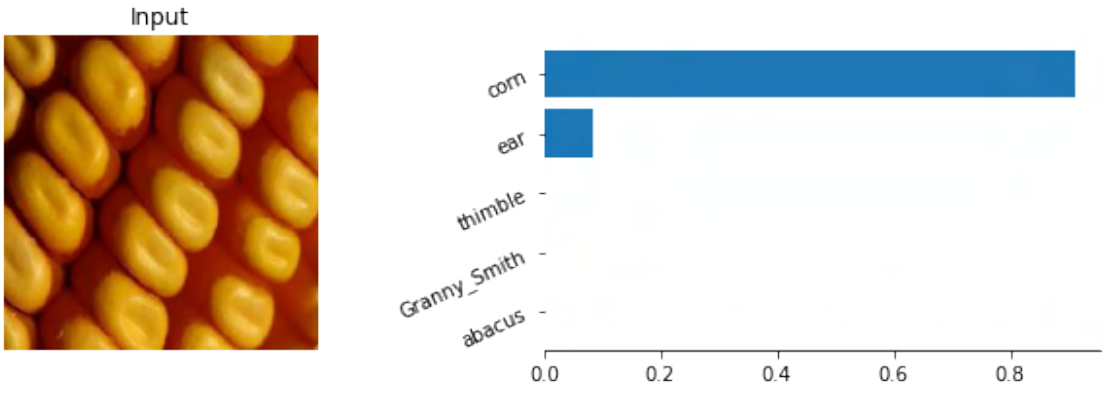

```
coral_reef

Loading Next Image
\n09399592.JPEG
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 51ms/step
```



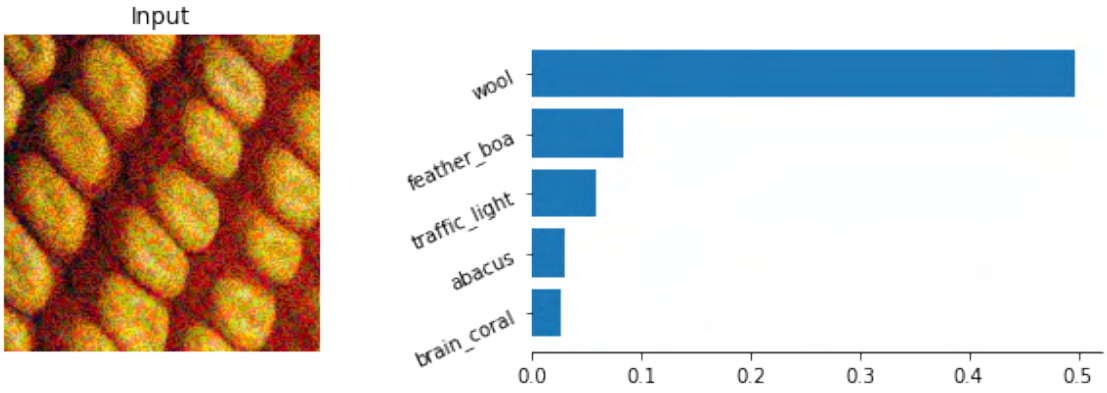

```
promontory
1/1 [==============================] - 0s 34ms/step
```



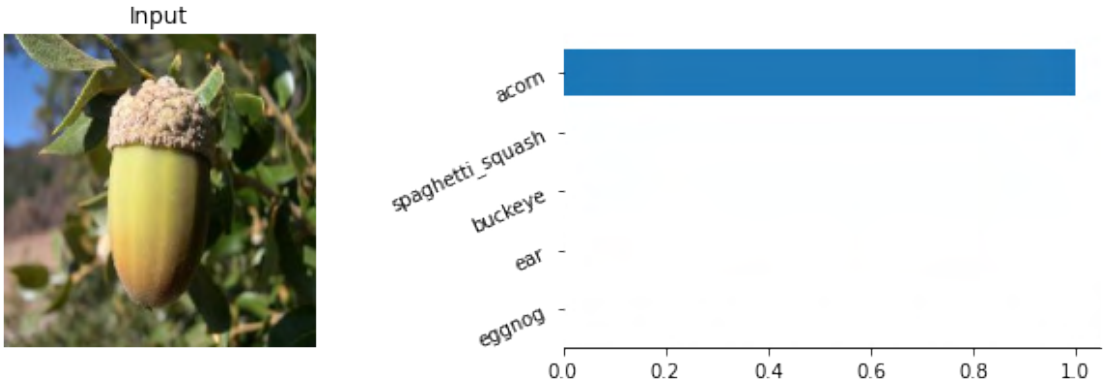

```
velvet

Loading Next Image
\n12144580.JPEG
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 57ms/step
```



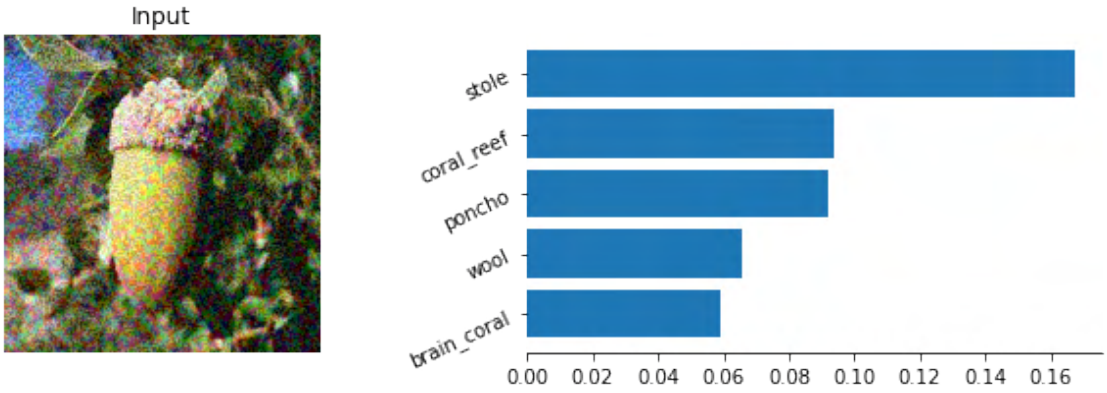

```
corn
1/1 [==============================] - 0s 55ms/step
```




```
wool

Loading Next Image
\n12267677.JPEG
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 52ms/step
```




```
acorn
1/1 [==============================] - 0s 52ms/step
```




```
stole

Loading Next Image
```

[90]:
```
print(output)
print(output2)
```

```
['goldfish', 'brambling', 'indigo_bunting', 'loggerhead', 'frilled_lizard',
'black_and_gold_garden_spider', 'African_grey', 'hornbill', 'wallaby', 'chiton',
'crane', 'limpkin', 'American_coot', 'dowitcher', 'killer_whale', 'Shih-Tzu',
'Staffordshire_bullterrier', 'Bedlington_terrier', 'Sealyham_terrier', 'flat-
coated_retriever', 'Appenzeller', 'Saint_Bernard', 'Siberian_husky',
'dalmatian', 'tiger', 'weevil', 'cricket', 'leafhopper', 'hare', 'hartebeest',
'polecat', 'otter', 'orangutan', 'proboscis_monkey', 'puffer', 'abaya',
'amphibian', 'analog_clock', 'backpack', 'balance_beam', 'barbell',
'barbershop', 'barometer', 'beacon', 'beaker', "carpenter's_kit",
'chainlink_fence', 'chiffonier', 'cliff_dwelling', 'coffee_mug', 'corkscrew',
'crib', 'dam', 'digital_watch', 'dining_table', 'dishwasher', 'dock',
'football_helmet', 'forklift', 'freight_car', 'French_horn', 'gasmask',
'guillotine', 'hand-held_computer', 'iPod', 'liner', 'marimba', 'modem',
'mousetrap', 'paddle', 'planetarium', 'police_van', 'punching_bag', 'purse',
'rain_barrel', 'recreational_vehicle', 'saltshaker', 'screw', 'snorkel',
'steel_drum', 'syringe', 'teapot', 'tub', 'velvet', 'violin', 'wall_clock',
'window_shade', 'wooden_spoon', 'wreck', 'guacamole', 'pretzel', 'cheeseburger',
'butternut_squash', 'jackfruit', 'carbonara', 'chocolate_sauce', 'lakeside',
'promontory', 'corn', 'acorn']
['stole', 'cockroach', 'stole', 'stole', 'stole', 'garden_spider', 'poncho',
'wool', 'stole', 'stole', 'stole', 'peacock', 'peacock', 'coral_reef',
'peacock', 'poncho', 'coral_reef', 'starfish', 'coral_reef', 'peacock',
'peacock', 'velvet', 'coral_reef', 'coral_reef', 'wool', 'weevil', 'poncho',
'prayer_rug', 'coral_reef', 'starfish', 'wool', 'stole', 'poncho', 'bubble',
'stole', 'jigsaw_puzzle', 'park_bench', 'wall_clock', 'stole', 'prayer_rug',
```

```
'greenhouse', 'pinwheel', 'stole', 'wool', 'prayer_rug', 'prayer_rug',
'peacock', 'stole', 'stole', 'park_bench', 'hook', 'poncho', 'coral_reef',
'prayer_rug', 'stole', 'bubble', 'jigsaw_puzzle', 'sock', 'chainlink_fence',
'prayer_rug', 'stole', 'bubble', 'prayer_rug', 'hand-held_computer', 'maze',
'wool', 'wool', 'prayer_rug', 'ant', 'poncho', 'prayer_rug', 'poncho',
'coral_reef', 'bubble', 'stole', 'maze', 'prayer_rug', 'prayer_rug', 'wool',
'coral_reef', 'syringe', 'jigsaw_puzzle', 'peacock', 'wool', 'stole',
'prayer_rug', 'prayer_rug', 'poncho', 'umbrella', 'poncho', 'bubble', 'wool',
'poncho', 'stole', 'brain_coral', 'sombrero', 'coral_reef', 'velvet', 'wool',
'stole']
```

[107]:
```
from sklearn.metrics import accuracy_score
acc = accuracy_score(output, output2, normalize=False)
print("The model was only accurate" , acc , "percent of the time when epsilon␣
 ↪was 0.01")
```

```
The model was only accurate 3 percent of the time when epsilon was 0.01
```

6 Congratulations!

You've come to the end of this assignment, and have seen a lot of the ways attack and fool an AI system. Here are the main points you should remember:

- It is very easy to fool a computer vision system if you know the model and its parameters.
- When designing an AI system, you need to think of adverse attacks againsts your system.

Congratulations on finishing this notebook!