In [ ]:
```
from time import time
import scipy.io as sio
import numpy as np
import json
from math import sqrt
from Saliency import Saliency
from SSA import SSA_H_Plus
from PIL import Image
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import os
from termcolor import colored
from keras.preprocessing import image
from keras.applications.vgg16 import VGG16,preprocess_input
import keras.preprocessing as preprocessing
from keras.models import Model
from keras.utils import load_img
import numpy as np
import tensorflow as tf
import saliency.core as saliency
from IPython.display import HTML, display
import tabulate
```

In [ ]:
```
class FeatureExtractor:
    def __init__(self) :
        base_model=VGG16(weights='vgg16_weights.h5')
        self.model=Model(inputs=base_model.input, outputs=base_model.get_
    def extract(self,img_path):
        img = load_img(img_path, target_size=(224, 224))
        img=img.convert('RGB')
        x=preprocessing.image.image_utils.img_to_array(img)
        x=np.expand_dims(x,axis=0)
        x=preprocess_input(x)
        feature=self.model.predict(x)[0]
        return feature/np.linalg.norm(feature)
    def extract_by_array(self,img_array):
        x=np.expand_dims(img_array,axis=0)
        x=preprocess_input(x)
        feature=self.model.predict(x)[0]
        return feature/np.linalg.norm(feature)
```

In [ ]:
```
class Saliency:
    def __init__(self) :
        m = VGG16(weights='vgg16_weights.h5')
        conv_layer = m.get_layer('block5_conv3')
        self.model = Model([m.inputs], [conv_layer.output, m.output])

        self.class_idx_str = 'class_idx_str'
    def LoadImage(self,file_path):
        im = Image.open(file_path)
        im = im.resize((224,224))
        im = np.asarray(im)
        return im
    def PreprocessImage(self,im):
        im = preprocess_input(im)
        return im
    def call_model_function(self,images, call_model_args=None, expected_k
        target_class_idx =  call_model_args[self.class_idx_str]
        images = tf.convert_to_tensor(images)
        with tf.GradientTape() as tape:
            if expected_keys==[saliency.base.INPUT_OUTPUT_GRADIENTS]:
                tape.watch(images)
                _, output_layer = self.model(images)
                output_layer = output_layer[:,target_class_idx]
                gradients = np.array(tape.gradient(output_layer, images))
                return {saliency.base.INPUT_OUTPUT_GRADIENTS: gradients}
            else:
                conv_layer, output_layer = self.model(images)
                gradients = np.array(tape.gradient(output_layer, conv_lay
                return {saliency.base.CONVOLUTION_LAYER_VALUES: conv_laye
                        saliency.base.CONVOLUTION_OUTPUT_GRADIENTS: gradi

    def GetSaliency(self,input,trs):
        im_orig = self.LoadImage(input)
        im = self.PreprocessImage(im_orig)
        _, predictions = self.model(np.array([im]))
        prediction_class = np.argmax(predictions[0])
        call_model_args = {self.class_idx_str: prediction_class}
        xrai_object = saliency.XRAI()
        xrai_attributions = xrai_object.GetMask(im, self.call_model_funct
        mask = xrai_attributions >= np.percentile(xrai_attributions, trs)
        im_mask = np.array(im_orig)
        im_mask[~mask] = 0
        return im_mask
```

In [ ]:
```
def getFolderAndFileName(imgPath):
    img_path_str=str(imgPath)
    img_path_splitted= img_path_str.split("/")
    return [img_path_splitted[len(img_path_splitted)-2],img_path_splitted
```

In [ ]:
```
img_path_dogs=[str(x) for x in sio.loadmat('cached-data/new/dogs/img_path
SSA_features=[x for x in sio.loadmat('cached-data/new/dogs/ssa_features_d
img_names=[getFolderAndFileName(x)[1] for x in img_path_dogs]
img_folders=[getFolderAndFileName(x)[0] for x in img_path_dogs]
all_results=[]
```

In [ ]:
```
fe=FeatureExtractor()
slc=Saliency()
```

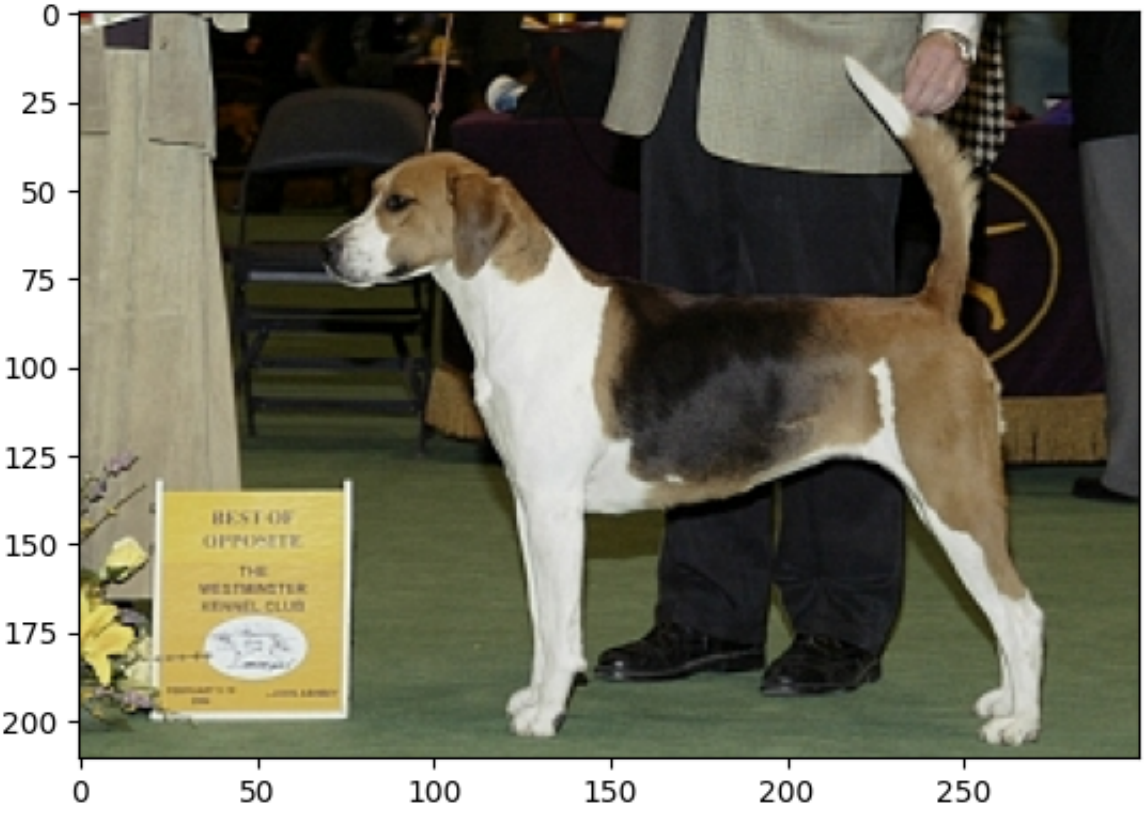
In [ ]:
```
# split_point=int(len(img_names)*.1)
split_point=10
count_of_results=5
training, test = img_names[split_point:], img_names[:split_point]
```

Non Saliency part:

In [ ]:
```
count_of_true=0
count_of_false=0
count_of_all=0
AllData=[['input','output1','output2','output3','output4','output5']]
for input in test:
    index_of_test=img_names.index(input)
    input_folder=img_folders[index_of_test]
    thisrec=[input_folder]
    print('input:')
    print(input_folder)
    img = mpimg.imread(img_path_dogs[index_of_test].replace(' ',''))
    imgplot = plt.imshow(img)
    plt.show()
    input_ssa=SSA_features[index_of_test]
    score_of_trained=[]
    for trained in training:
        index_of_trained=img_names.index(trained)
        trained_ssa=SSA_features[index_of_trained]
        distance= sqrt( sum(np.multiply(trained_ssa - input_ssa,trained_s
        trained_folder=img_folders[index_of_trained]
        score_of_trained.append([distance,trained,trained_folder])
    score_of_trained.sort()
    score_of_trained=score_of_trained[:count_of_results]
    count_of_this_true=0
    print('output:')
    for res in score_of_trained:
        if(res[2]==input_folder):
            print(colored(res[2],'green'))
            count_of_true=count_of_true+1
        else:
            print(colored(res[2],'red'))
            count_of_false=count_of_false+1

        thisrec.append(res[2])
        count_of_all=count_of_all+1
        img=mpimg.imread(os.getcwd()+'/data-set/dogs/images/Images/'+res[
        imgplot=plt.imshow(img)
        plt.show()
    print('--------')
    AllData.append(thisrec)
AllData.append([count_of_all,0,0,0,0,count_of_true])
print('Count Of True: '+str(count_of_true))
print('Count Of False: '+str(count_of_false))
print('Count Of All: '+str(count_of_all))
table = tabulate.tabulate(AllData, tablefmt='html')
table
```

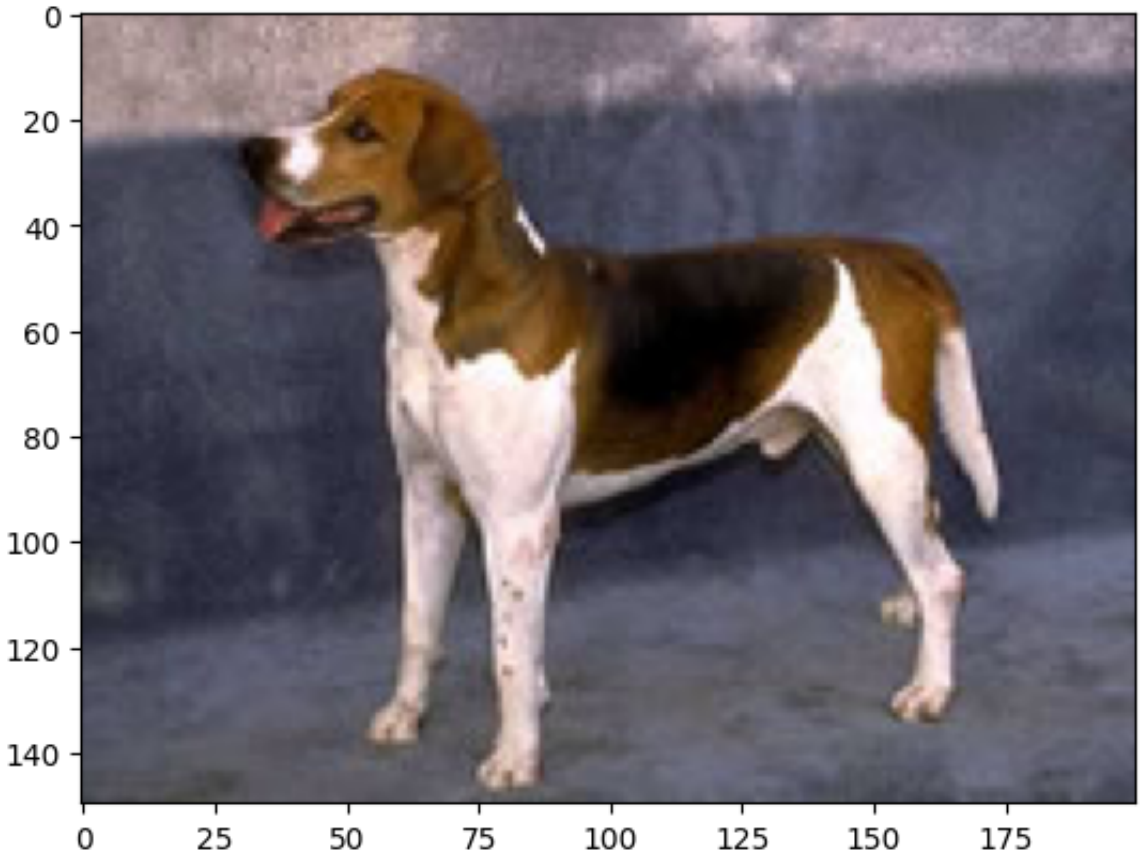
```
input:
n02089973-English_foxhound
```

output:
n02089973-English_foxhound



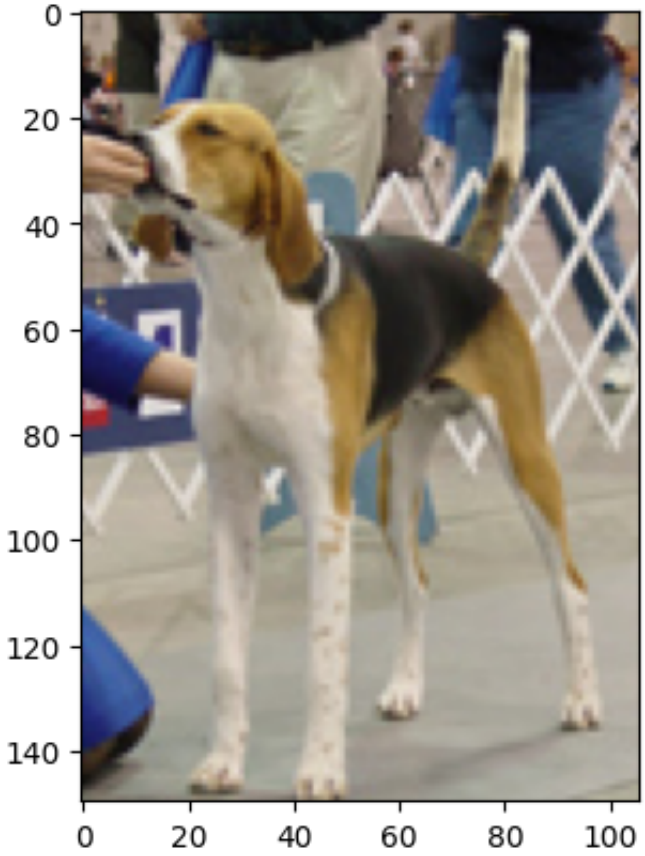
n02089867-Walker_hound

n02089973-English_foxhound



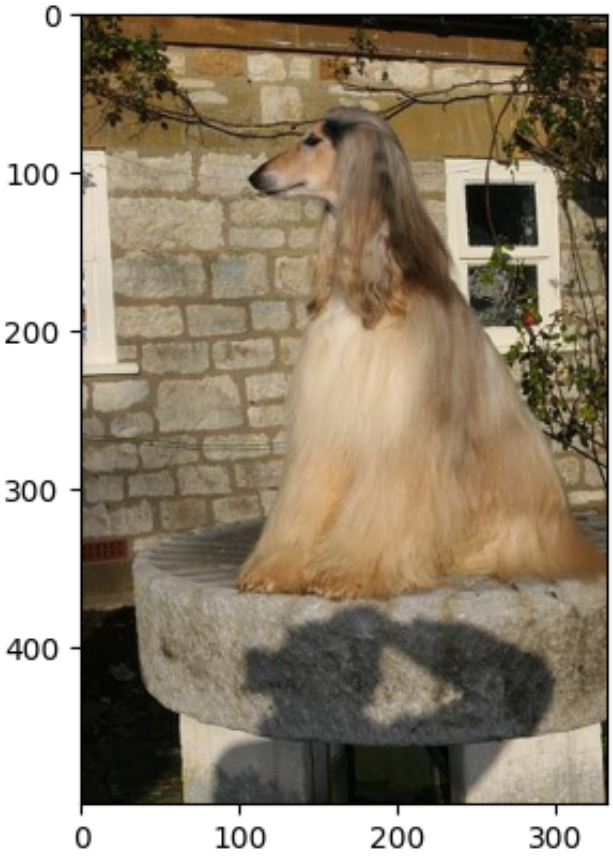
n02089973-English_foxhound

n02089973-English_foxhound

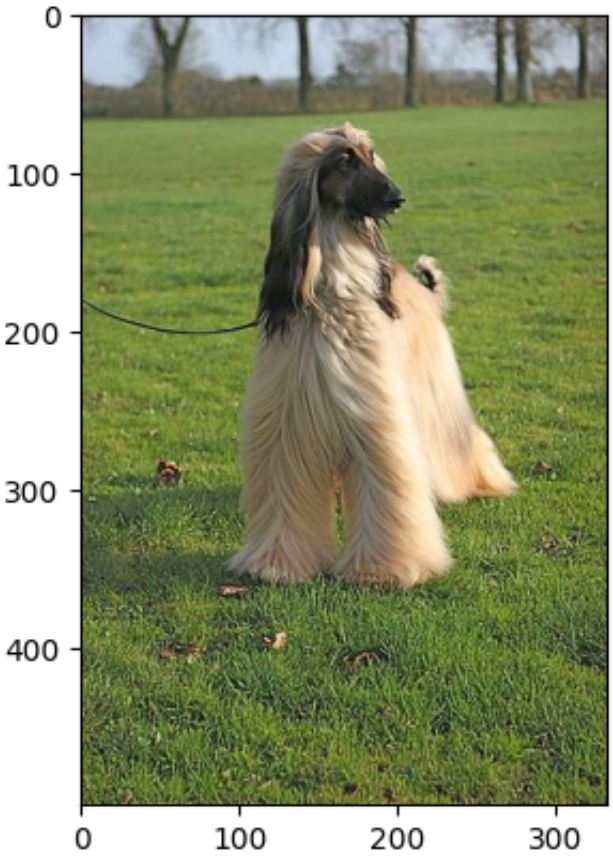


---

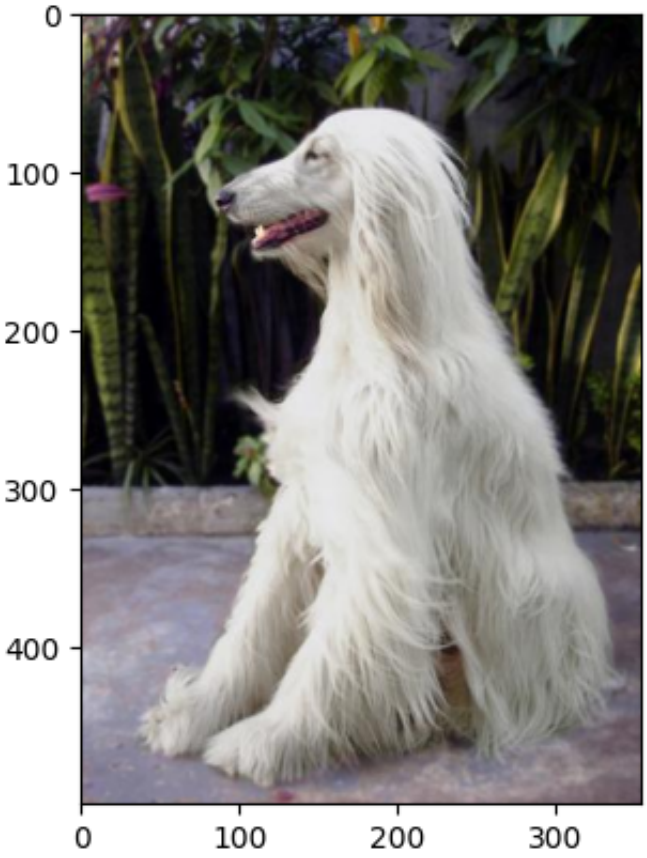
input:
n02088094-Afghan_hound



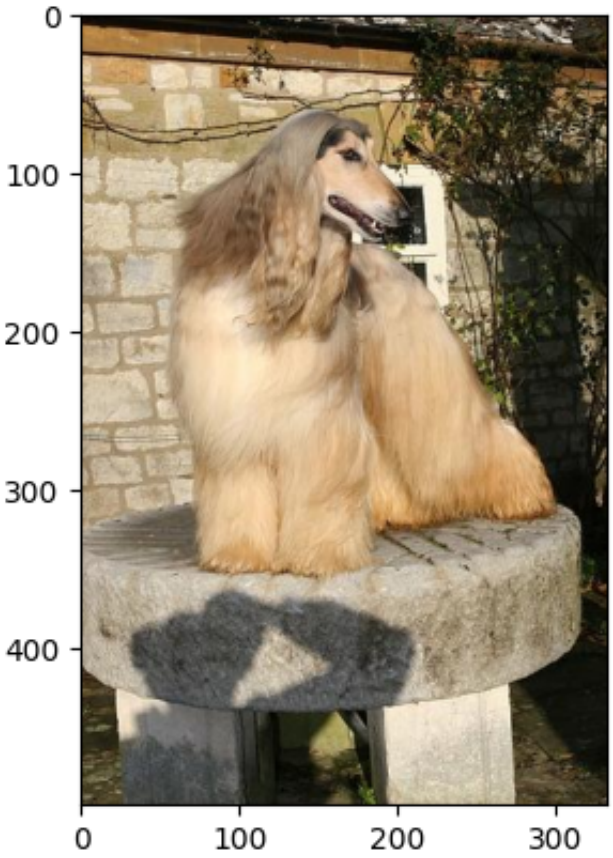
output:
n02088094-Afghan_hound



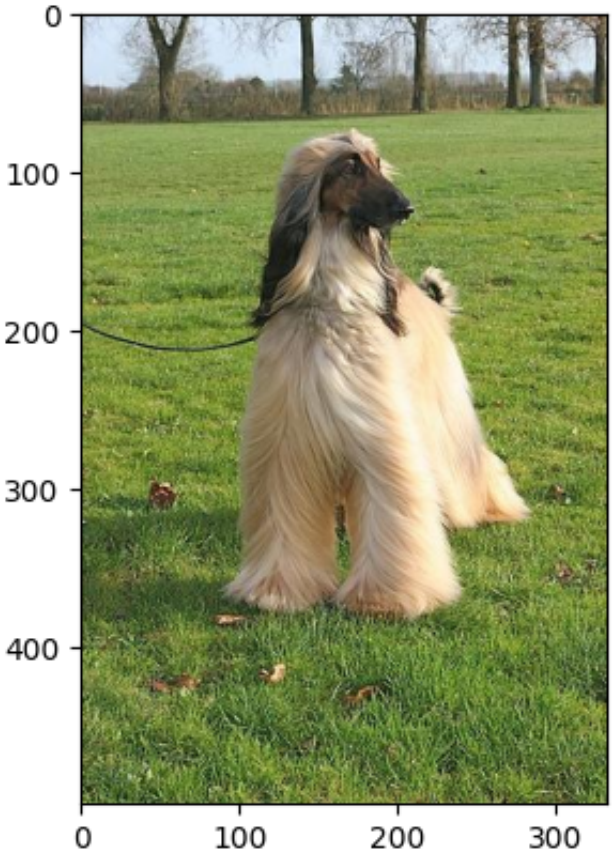
n02088094-Afghan_hound



n02088094-Afghan_hound

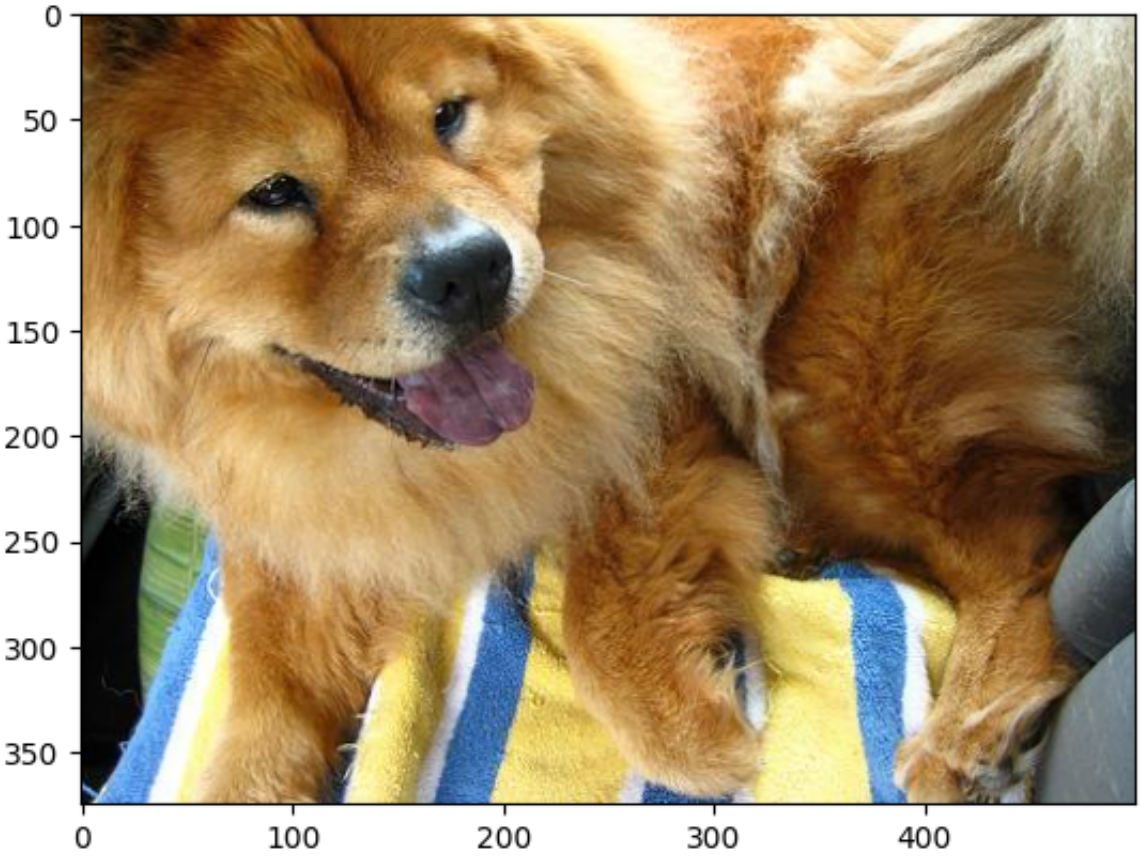


n02088094-Afghan_hound



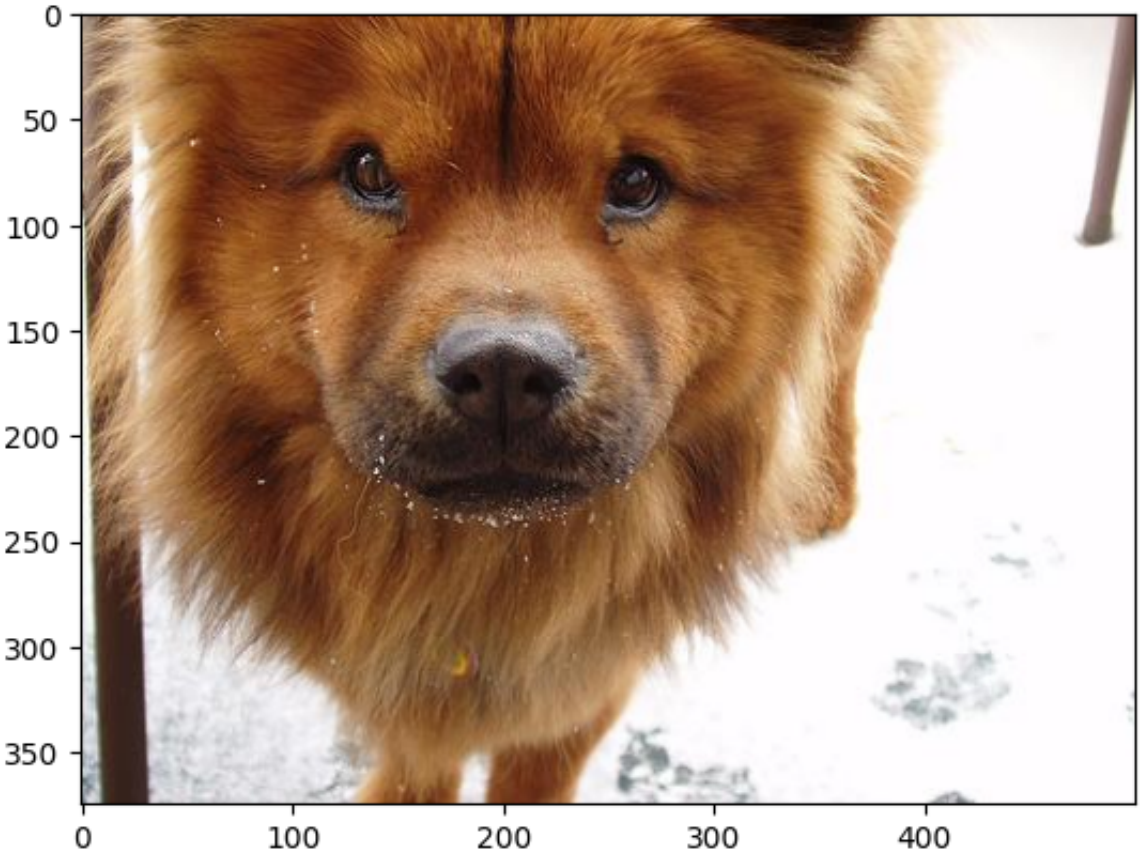
n02088094-Afghan_hound


```
--------
input:
n02112137-chow
```

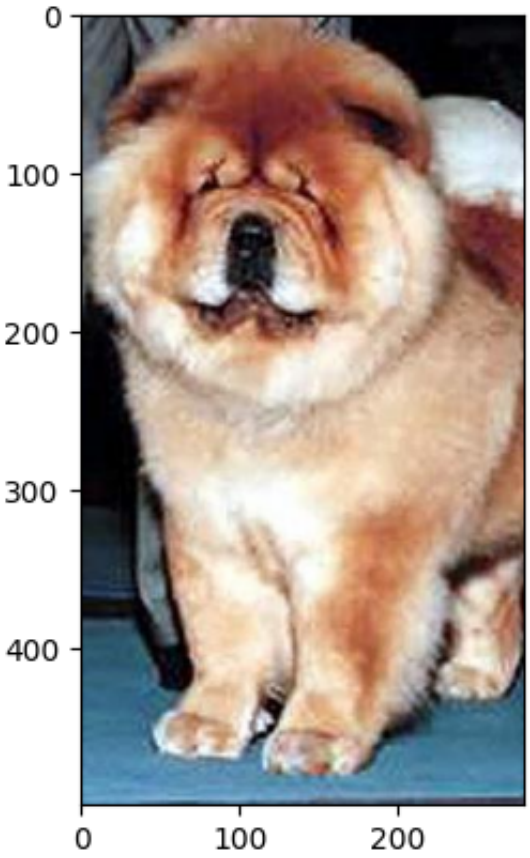


output:
n02112137-chow

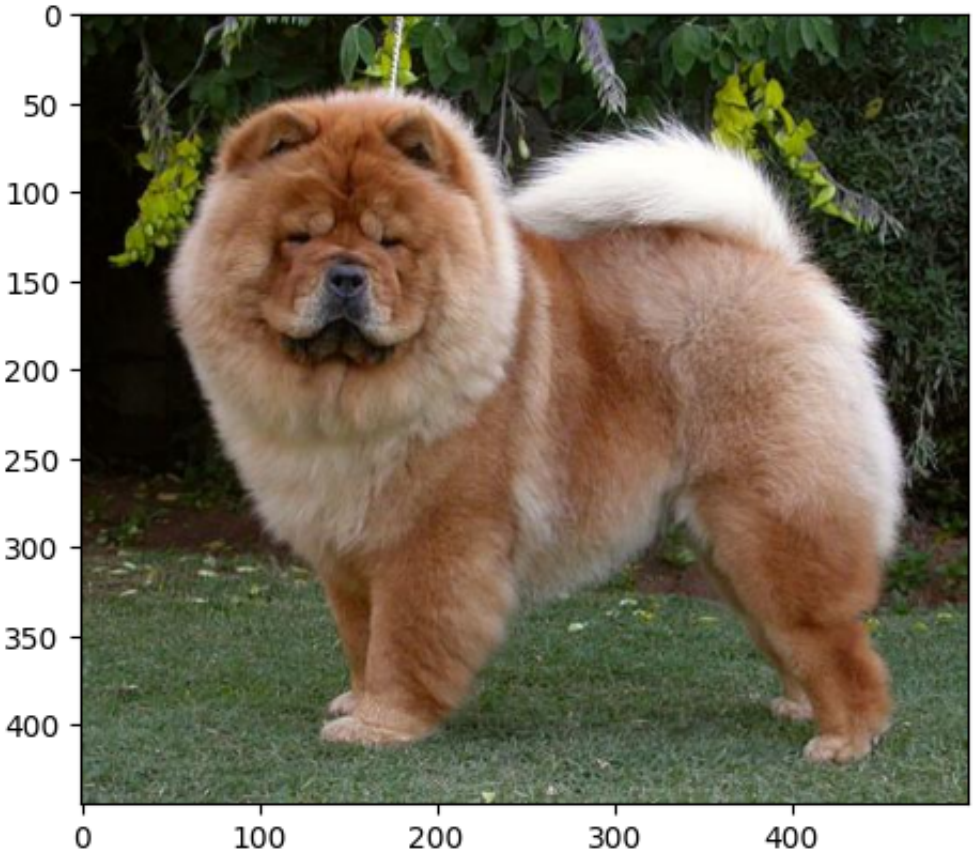
n02112137-chow



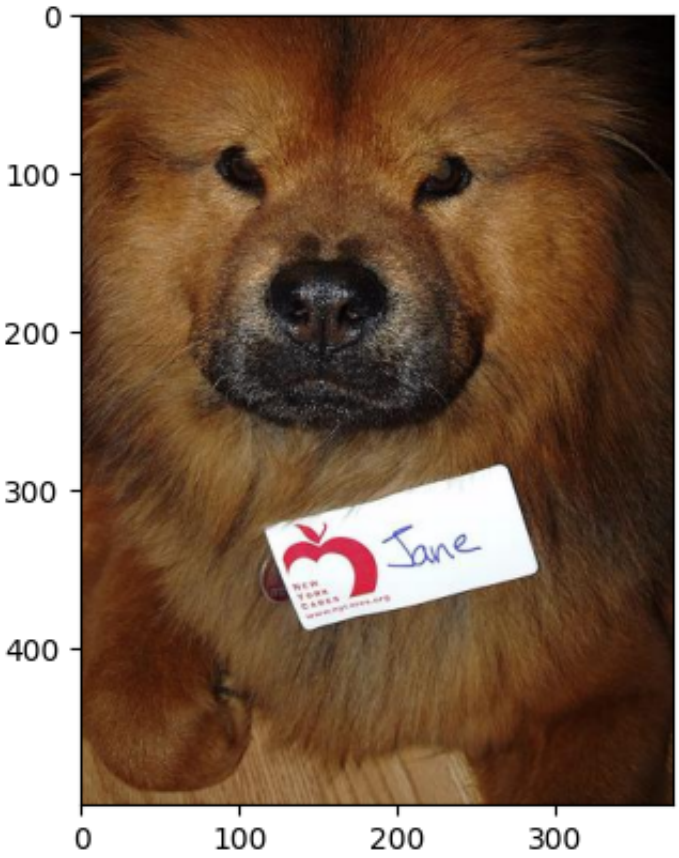
n02112137-chow



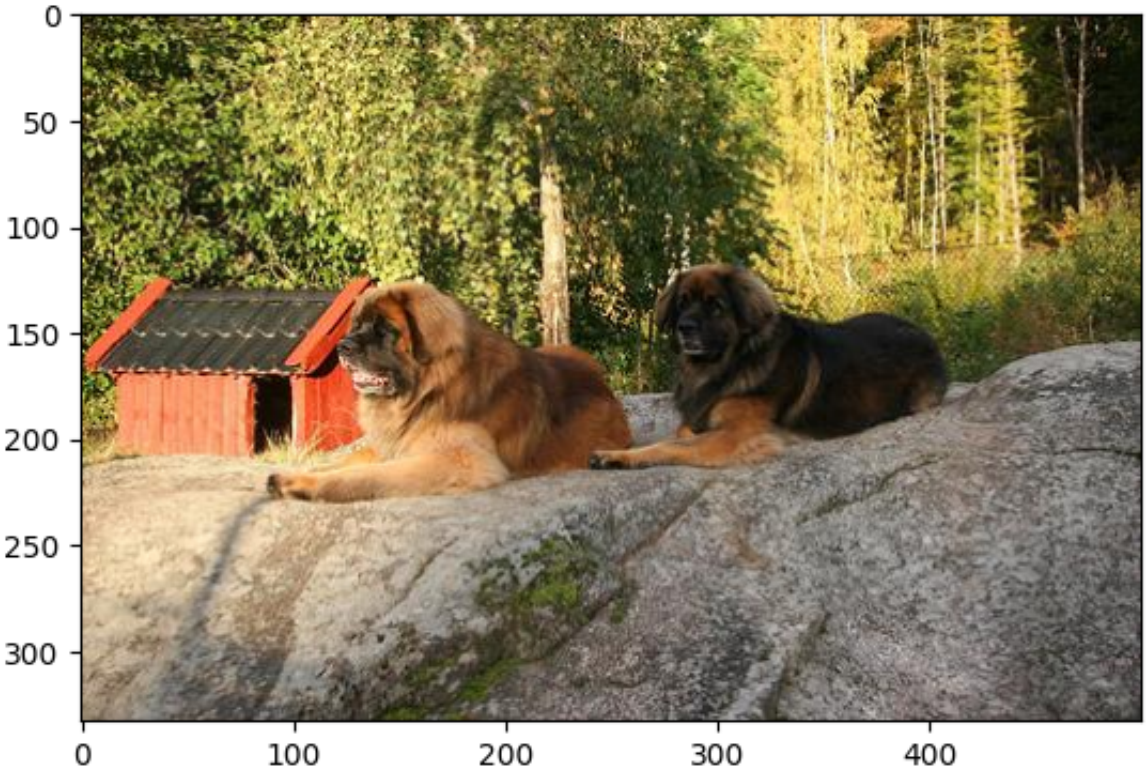
n02112137-chow



n02112137-chow

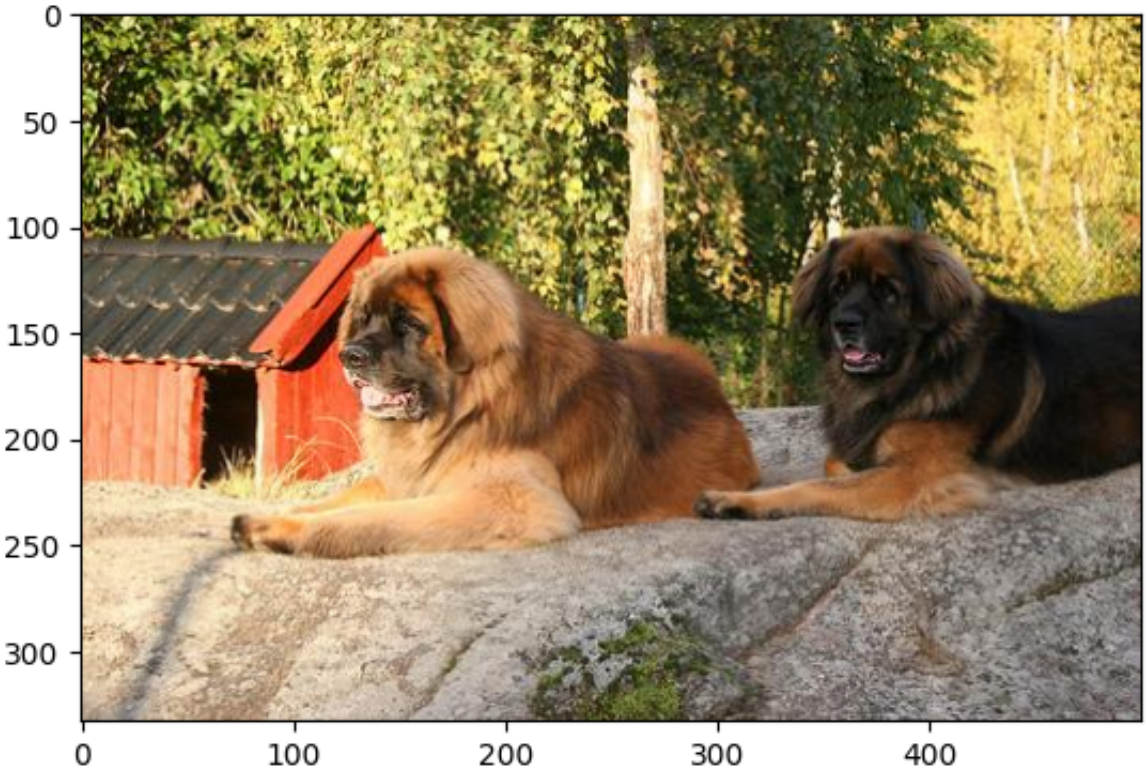



```
________
input:
n02111129-Leonberg
```

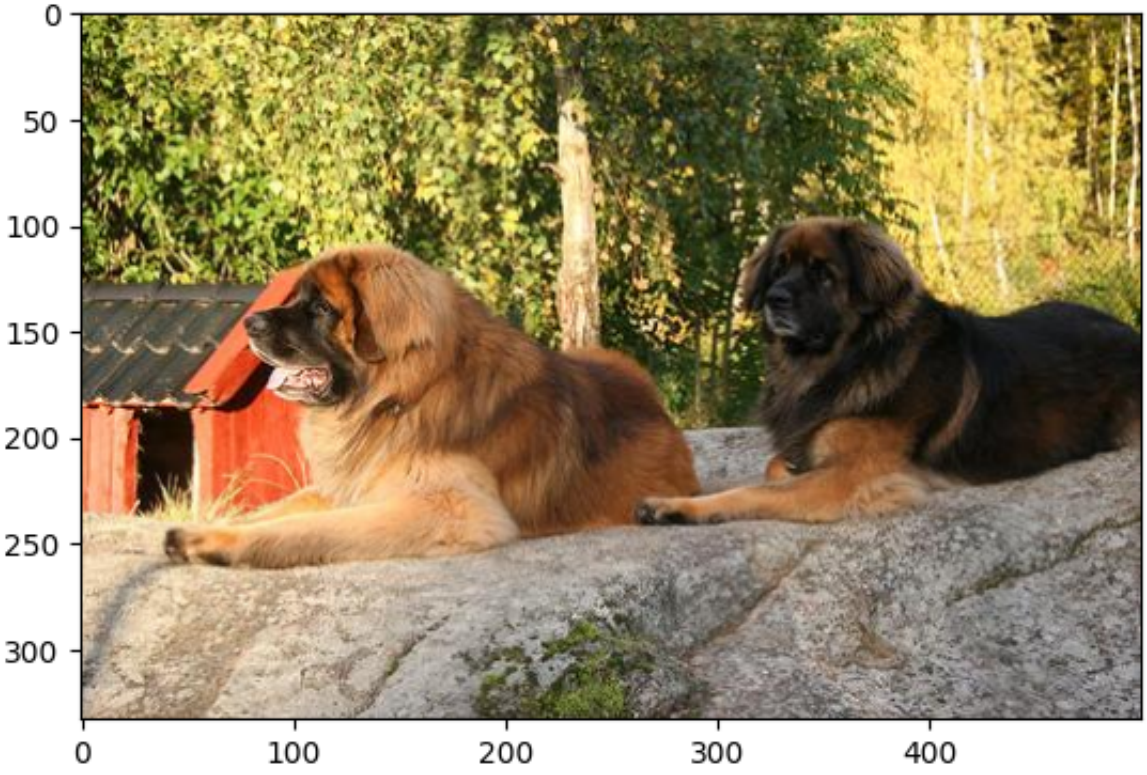


output:
n02111129-Leonberg

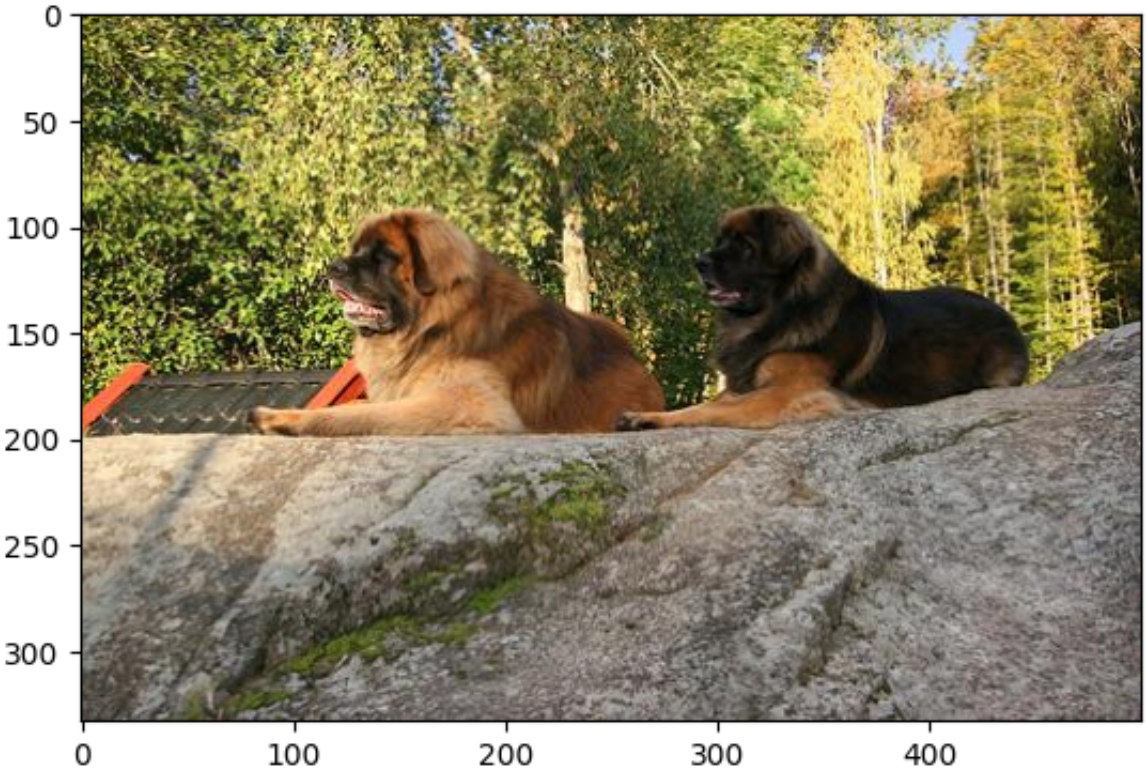
n02111129-Leonberg



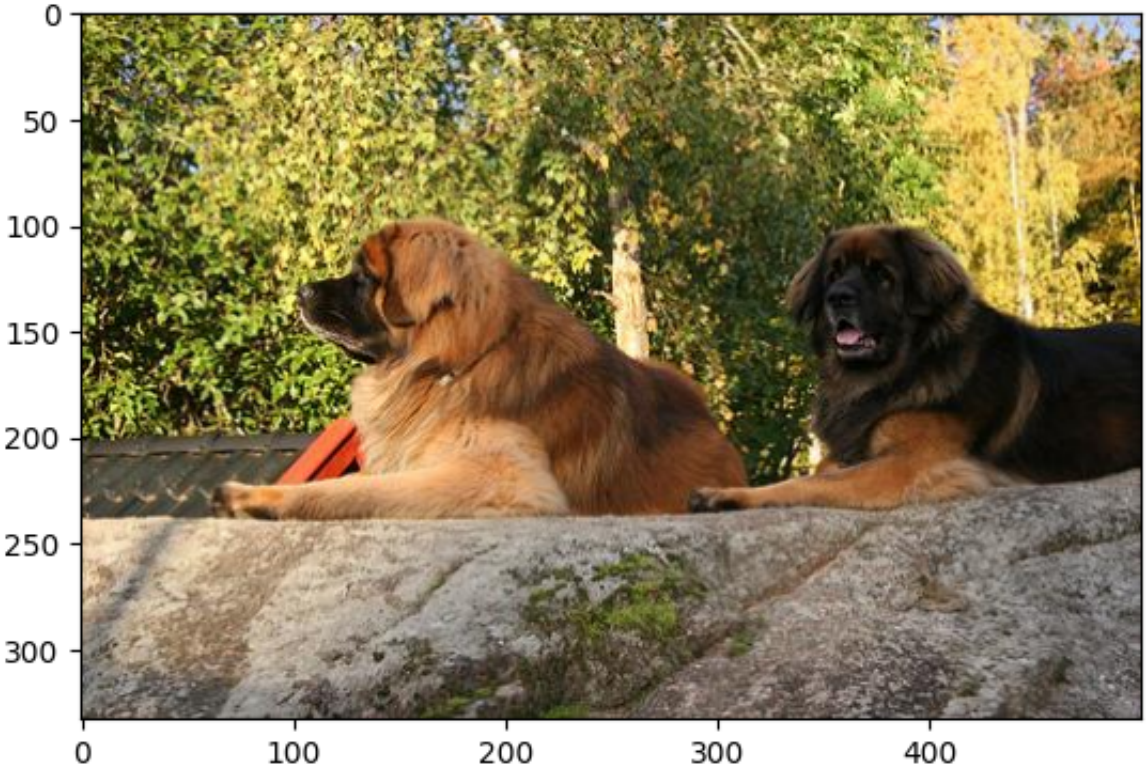
n02111129-Leonberg



n02111129-Leonberg



n02111129-Leonberg



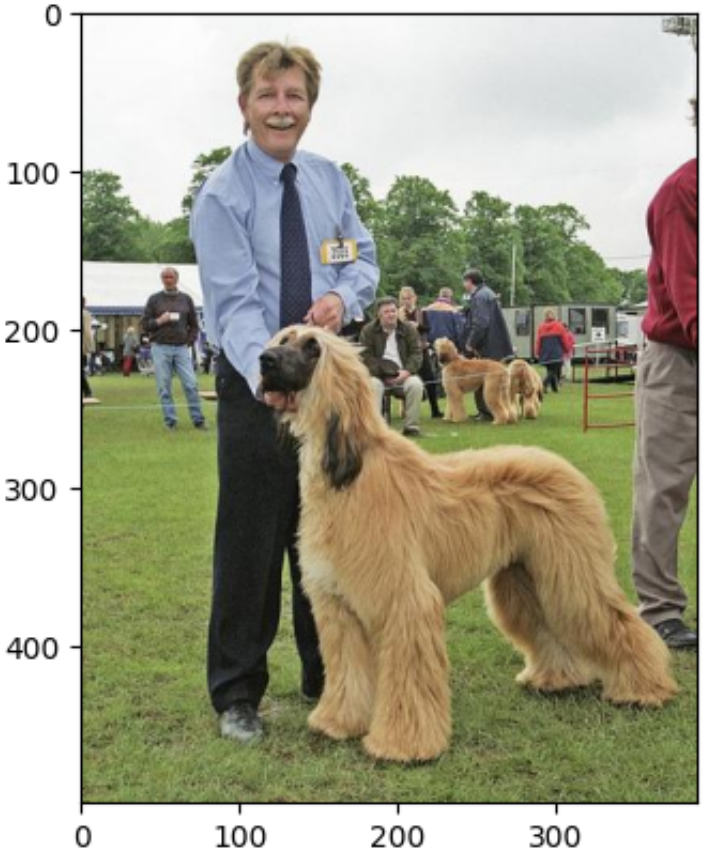
---

input:
n02088094-Afghan_hound

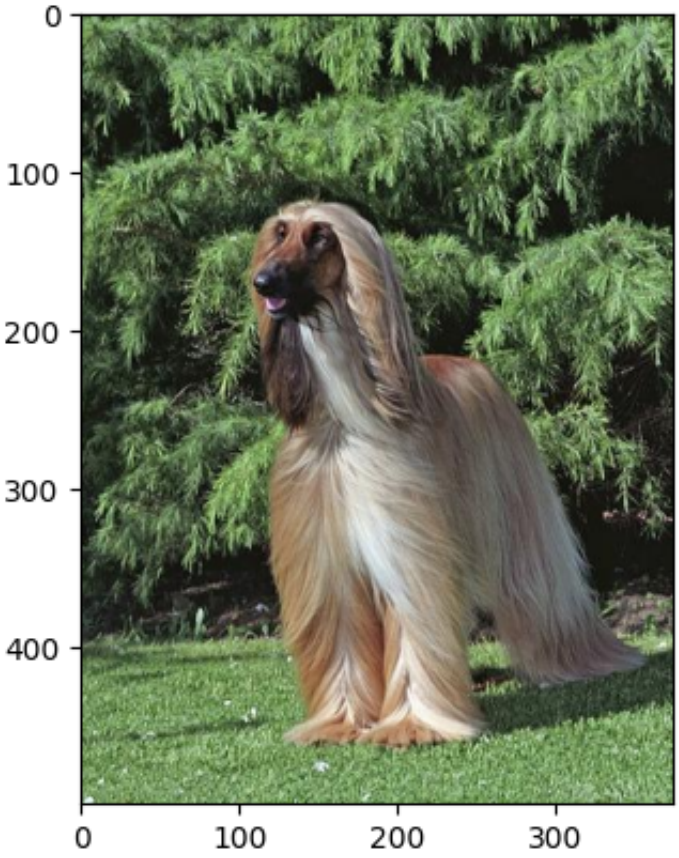
output:
n02088094-Afghan_hound

n02088094-Afghan_hound

n02088094-Afghan_hound



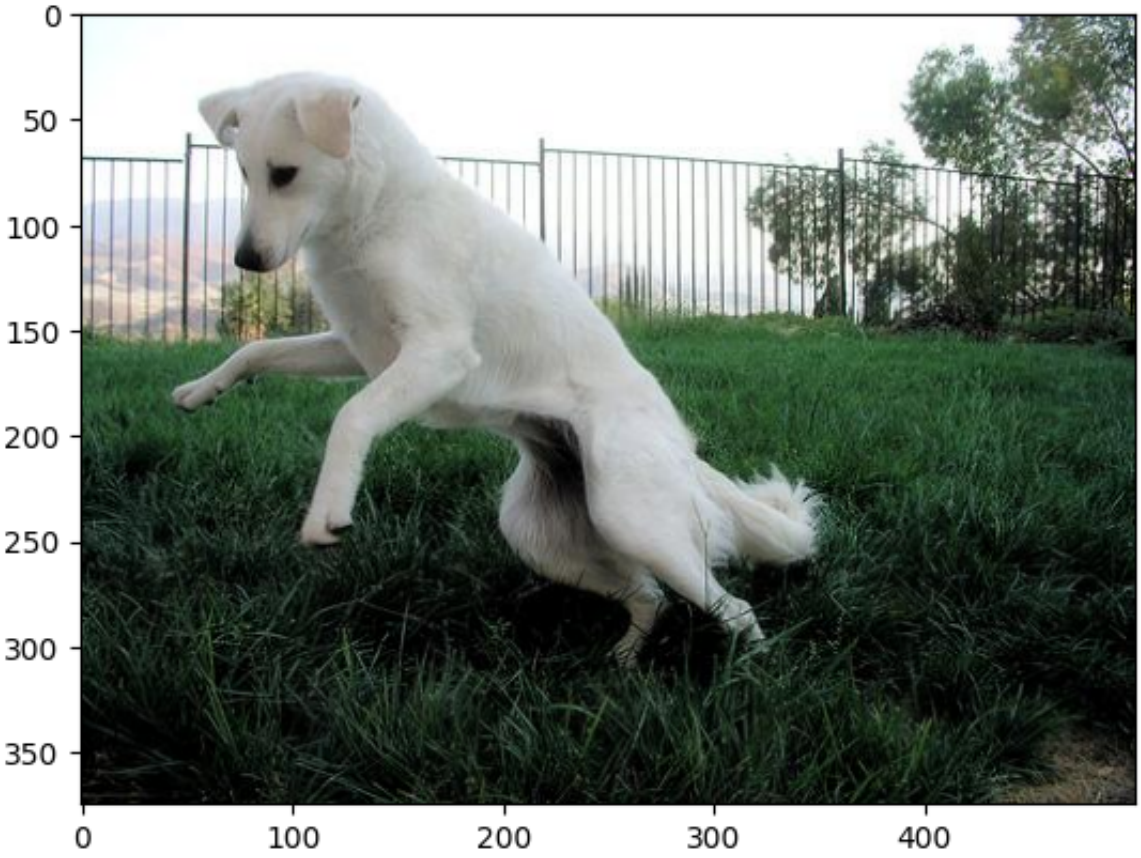
n02088094-Afghan_hound



n02097474-Tibetan_terrier

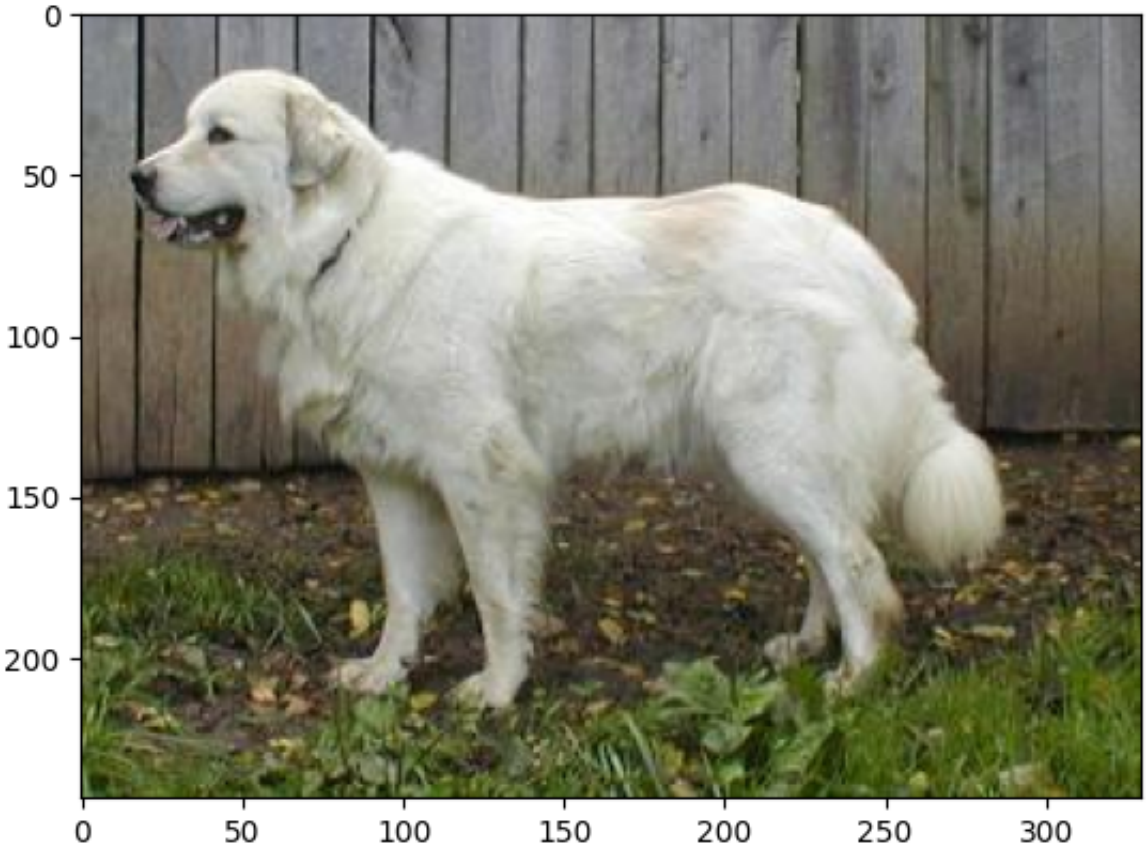
---

input:
n02104029-kuvasz

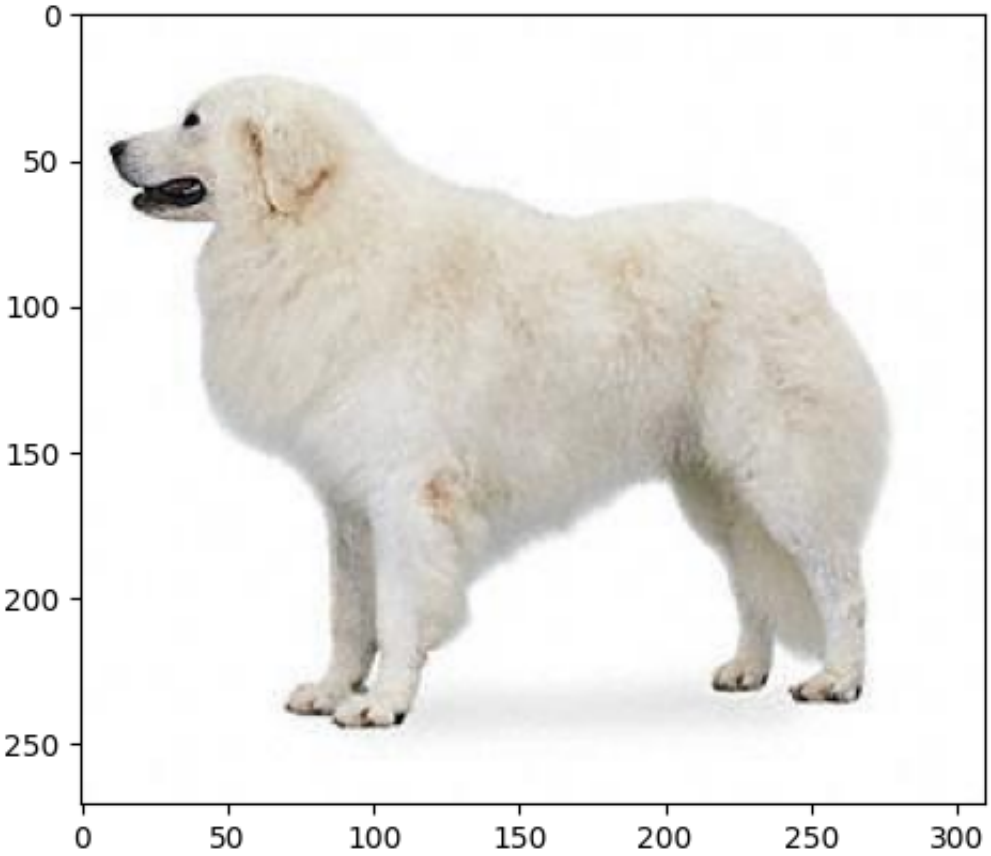


output:
n02104029-kuvasz

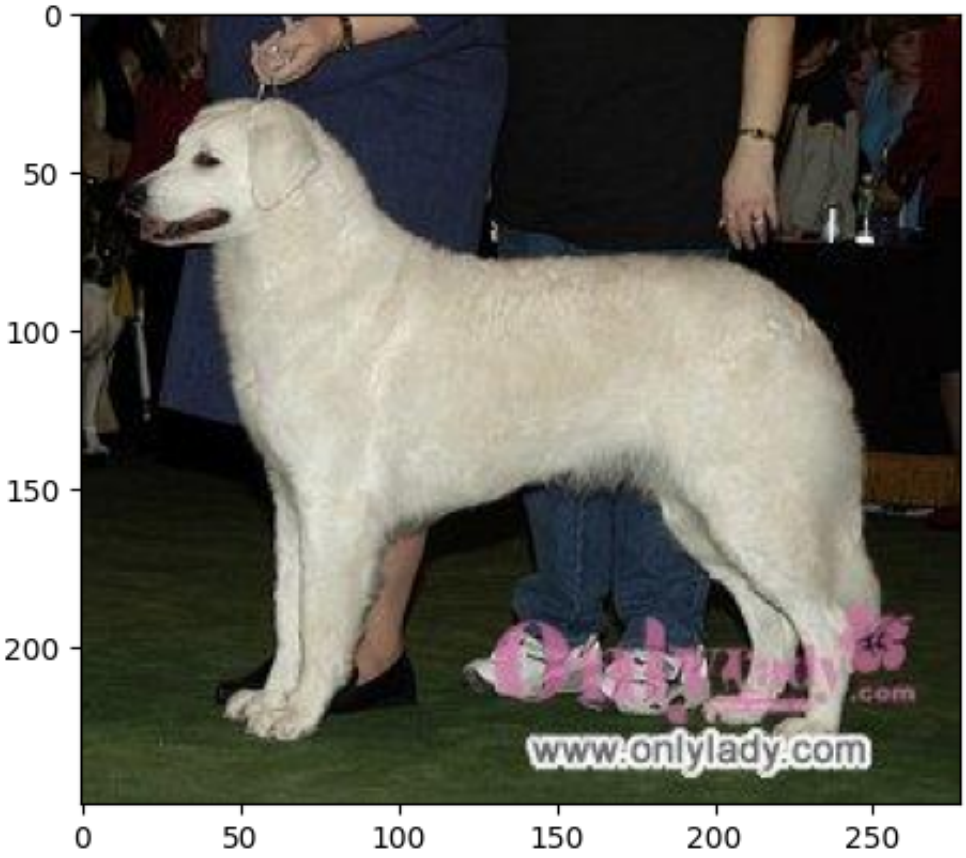
n02111500-Great_Pyrenees



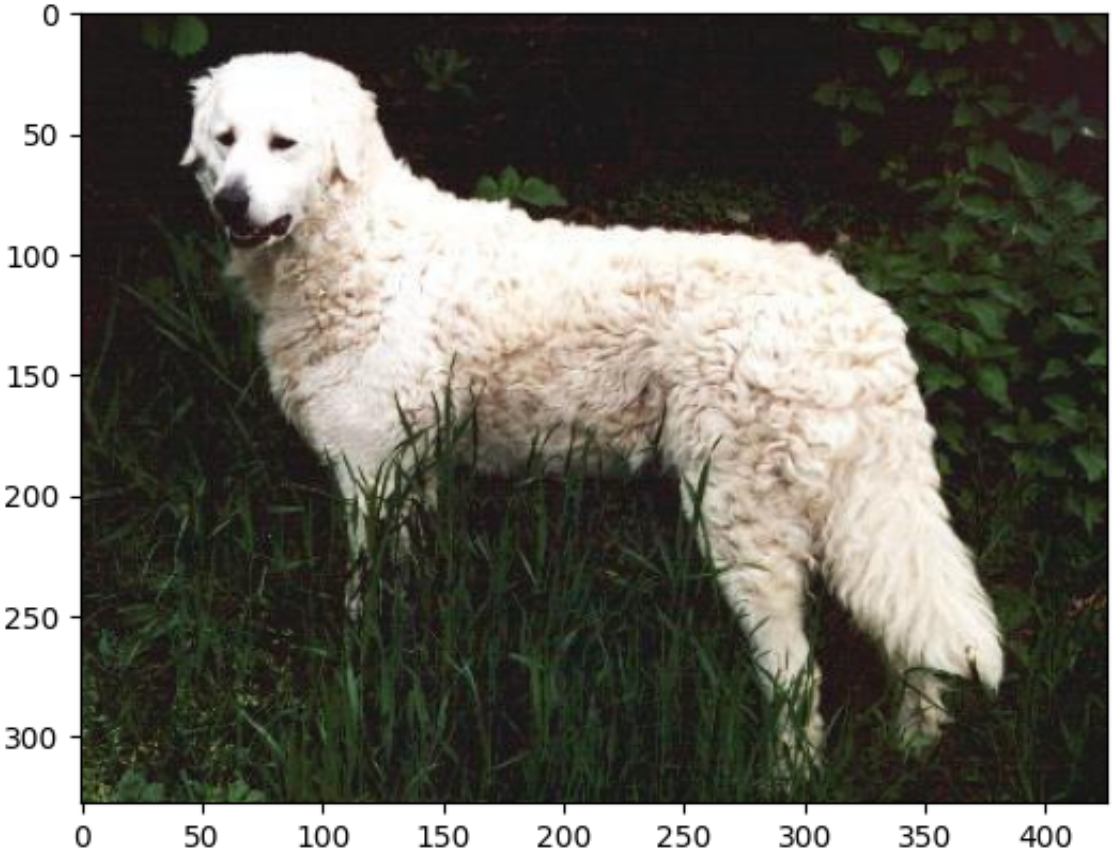
n02104029-kuvasz



n02104029-kuvasz



n02104029-kuvasz

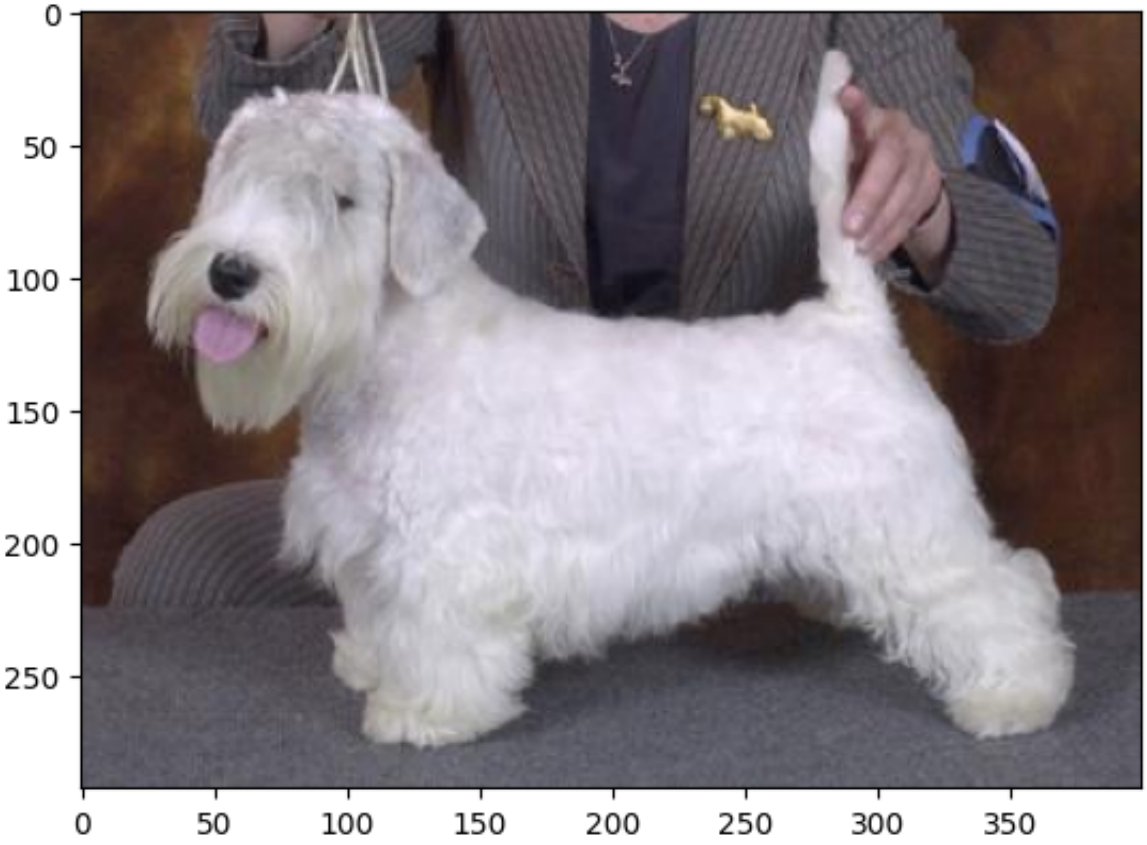



```
________
input:
n02095889-Sealyham_terrier
```

output:
n02095889-Sealyham_terrier

n02095889-Sealyham_terrier

n02095889-Sealyham_terrier

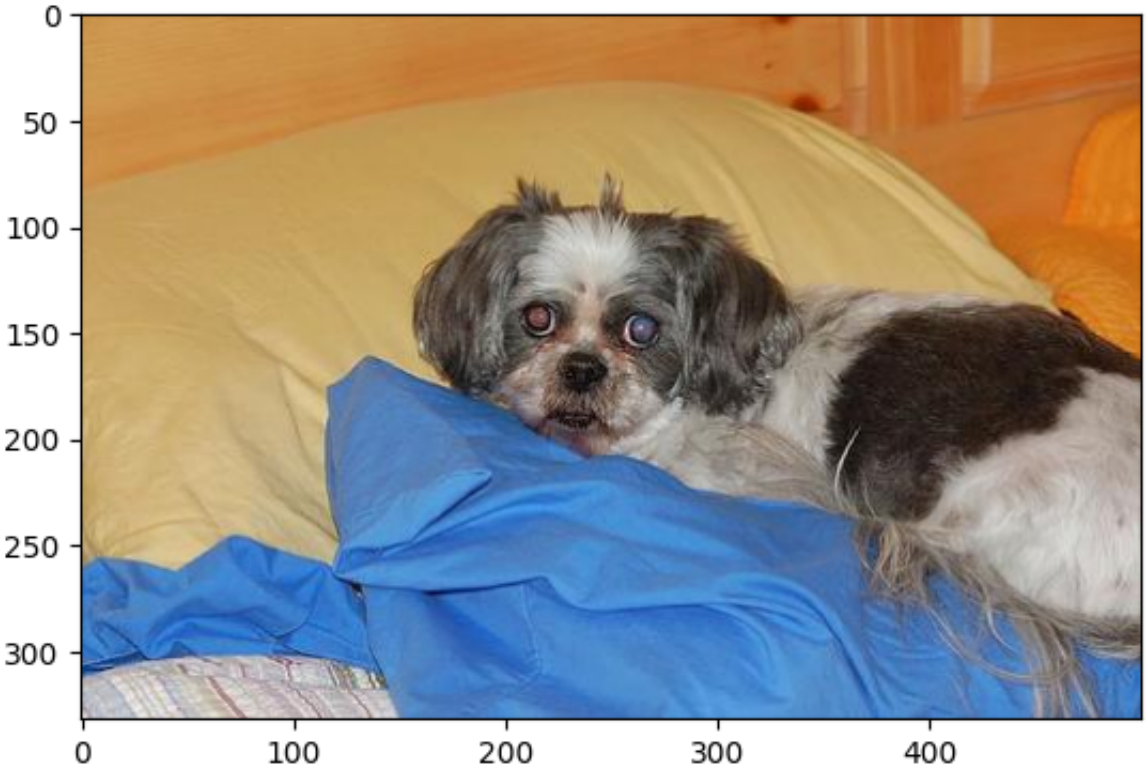


n02095889-Sealyham_terrier

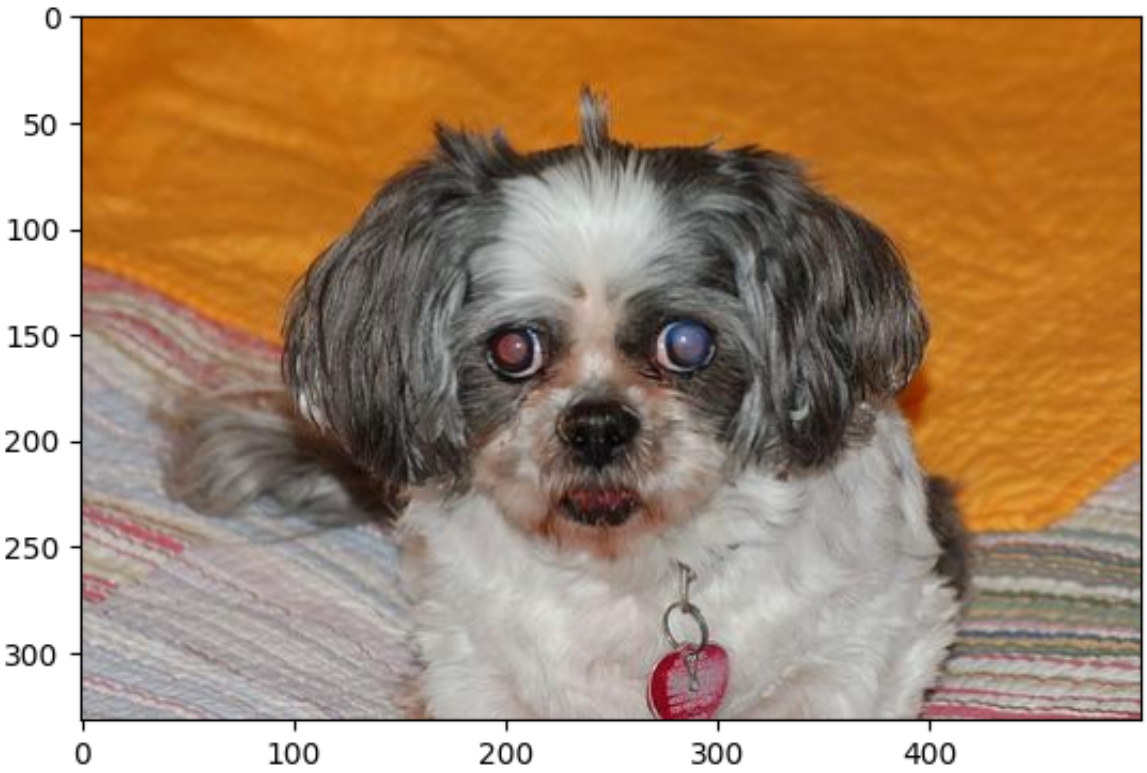
n02095889-Sealyham_terrier

---

input:
n02086240-Shih-Tzu

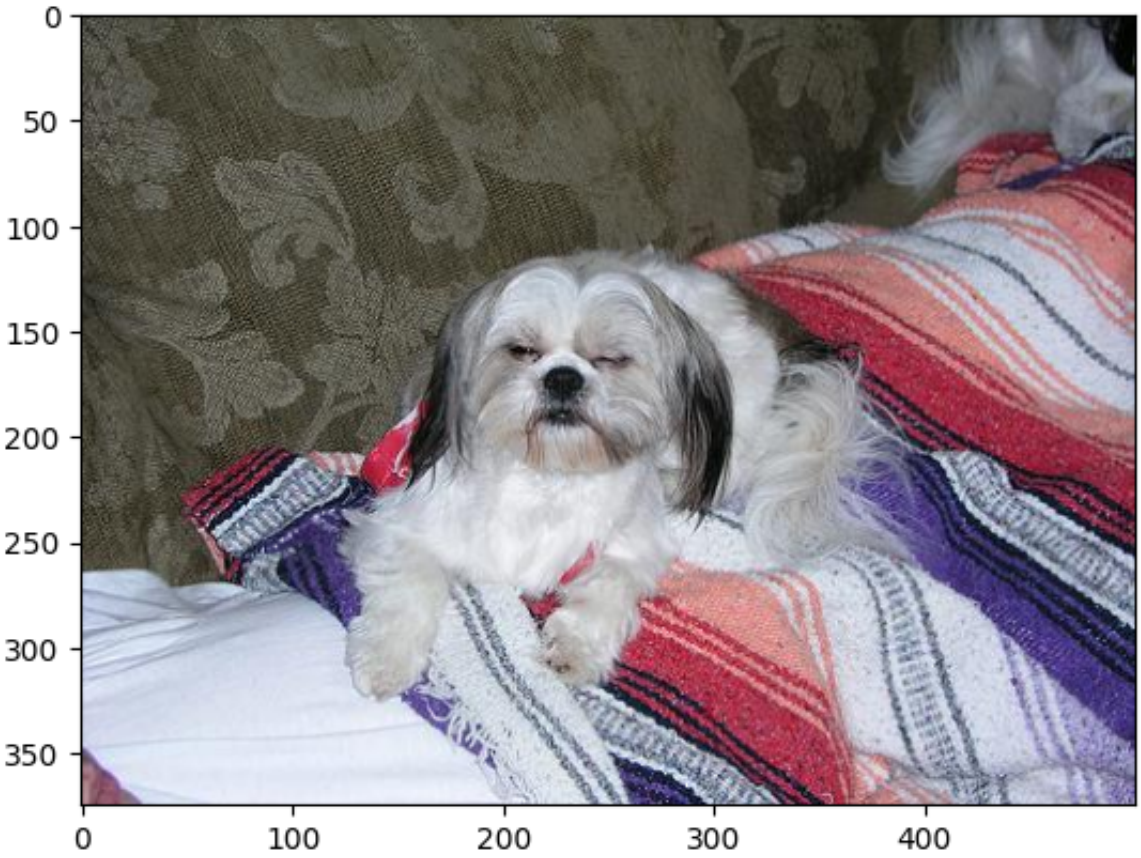


output:
n02086240-Shih-Tzu

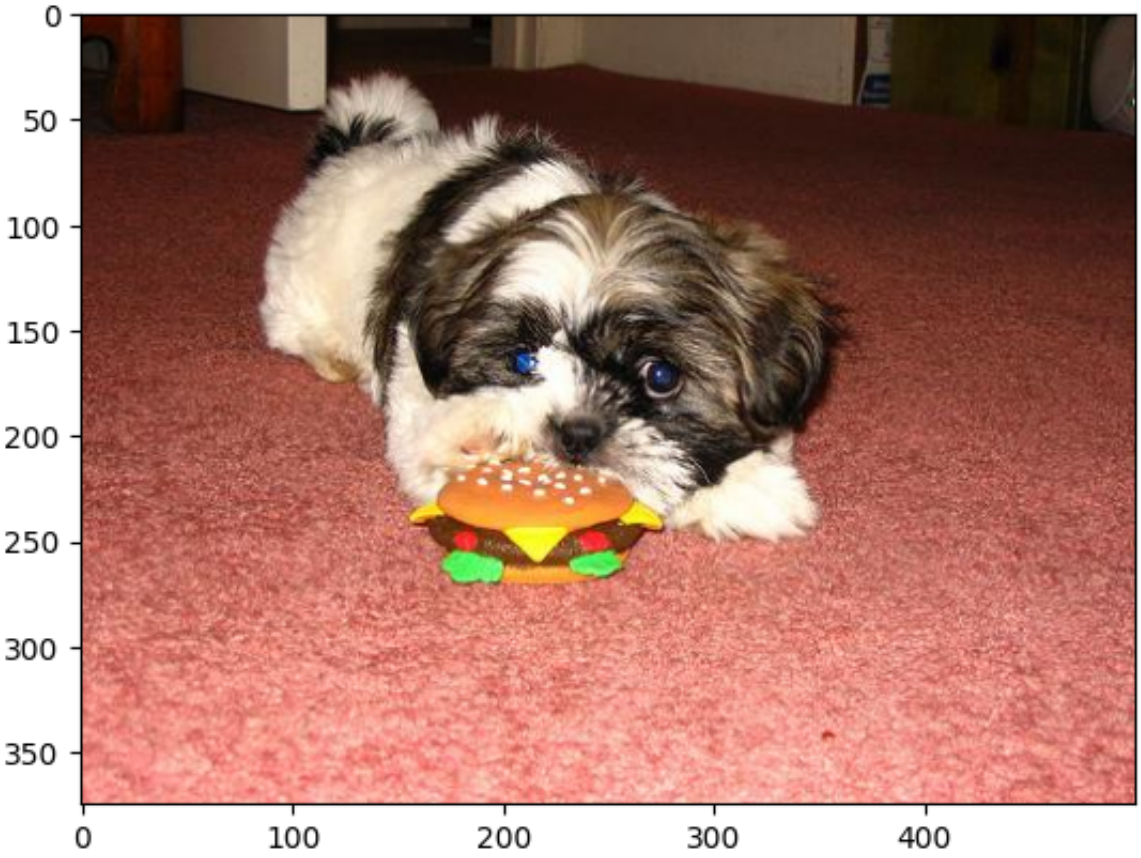


n02086240-Shih-Tzu

n02086240-Shih-Tzu



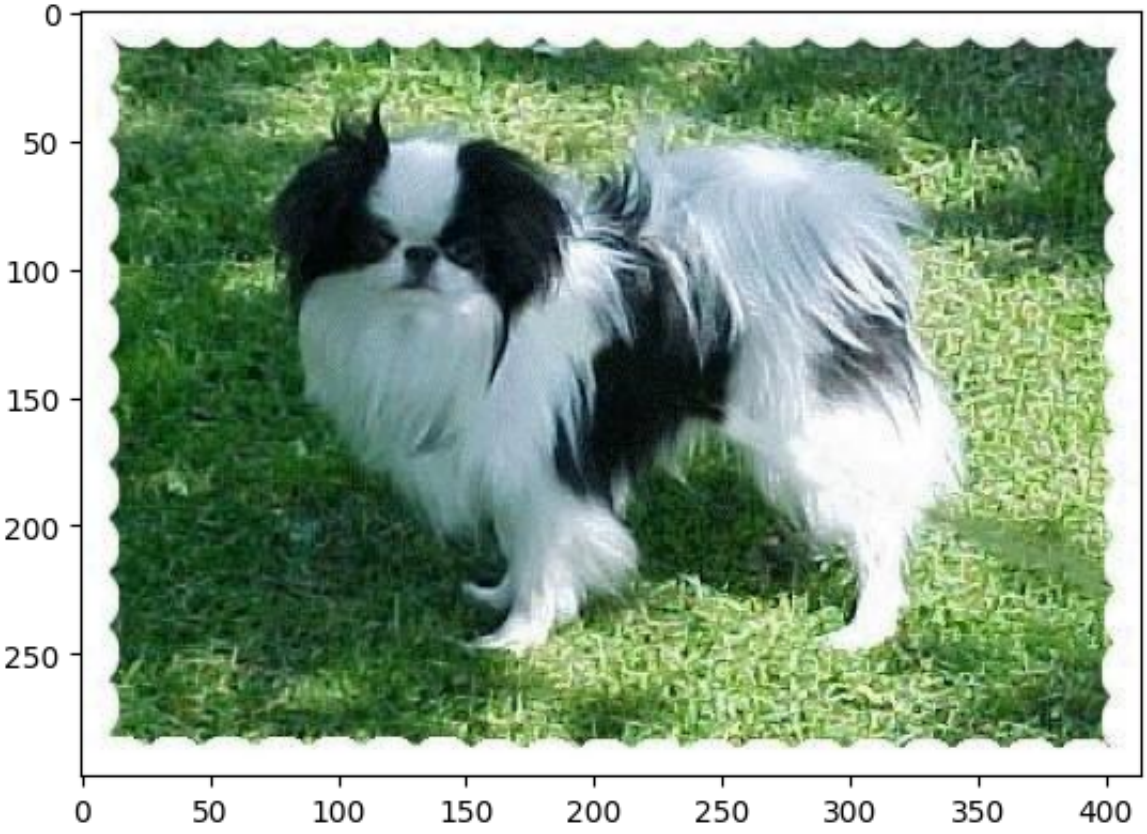
n02086240-Shih-Tzu



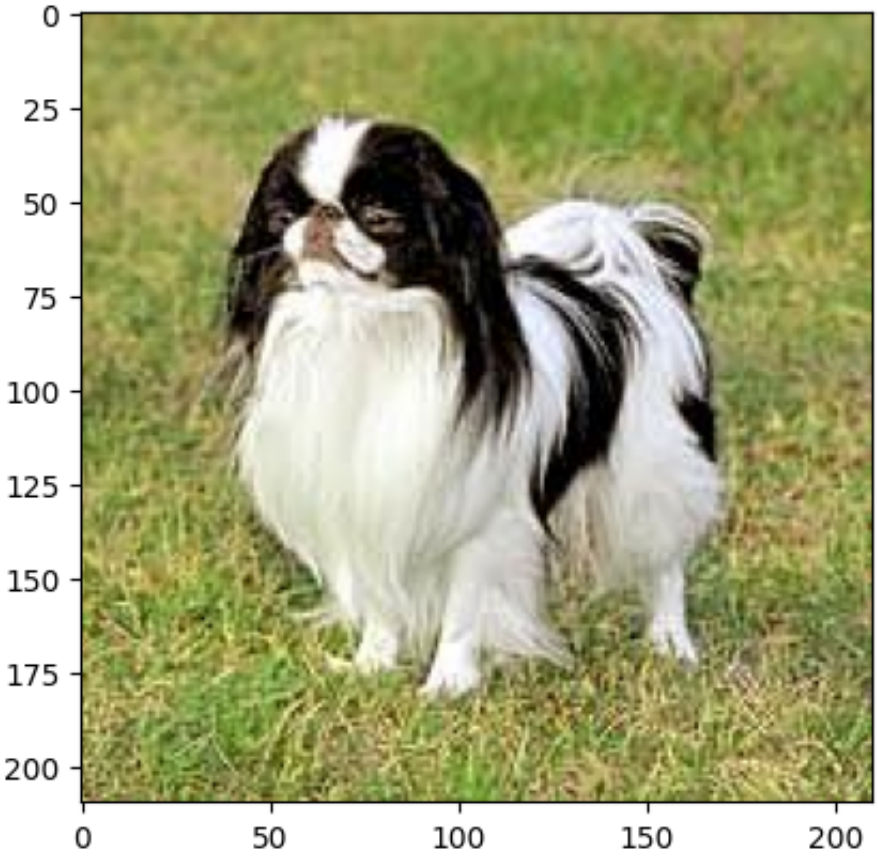
n02085782-Japanese_spaniel


```
________
input:
n02085782-Japanese_spaniel
```

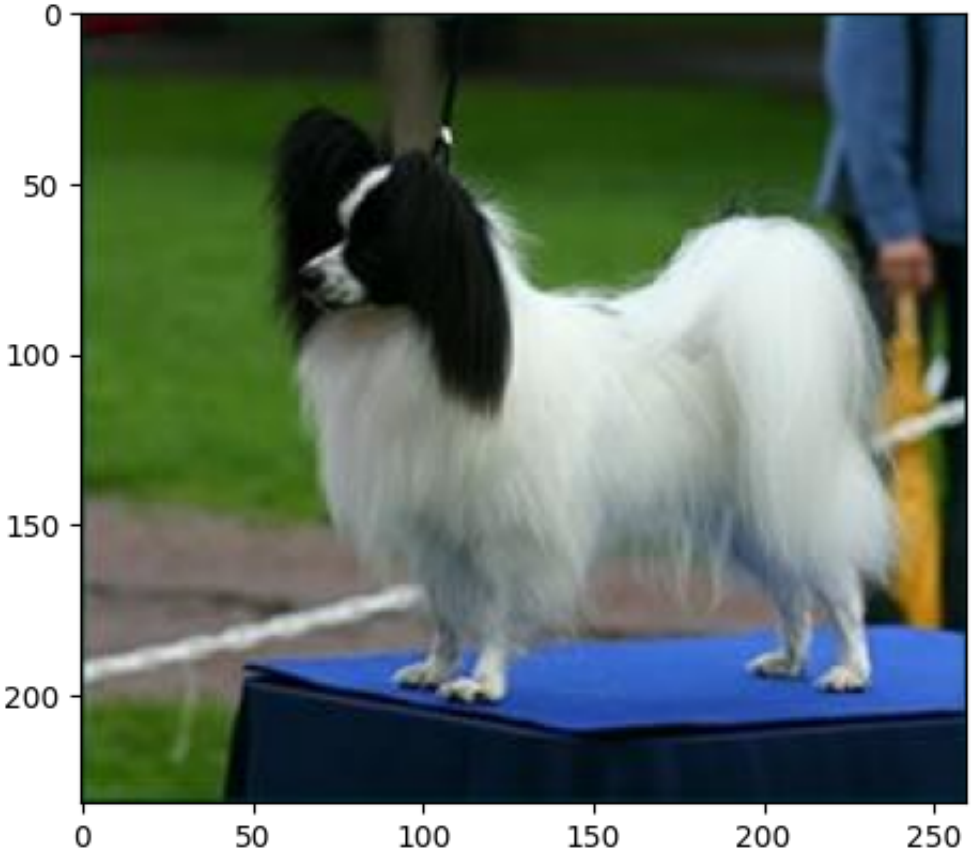
output:
n02085782-Japanese_spaniel



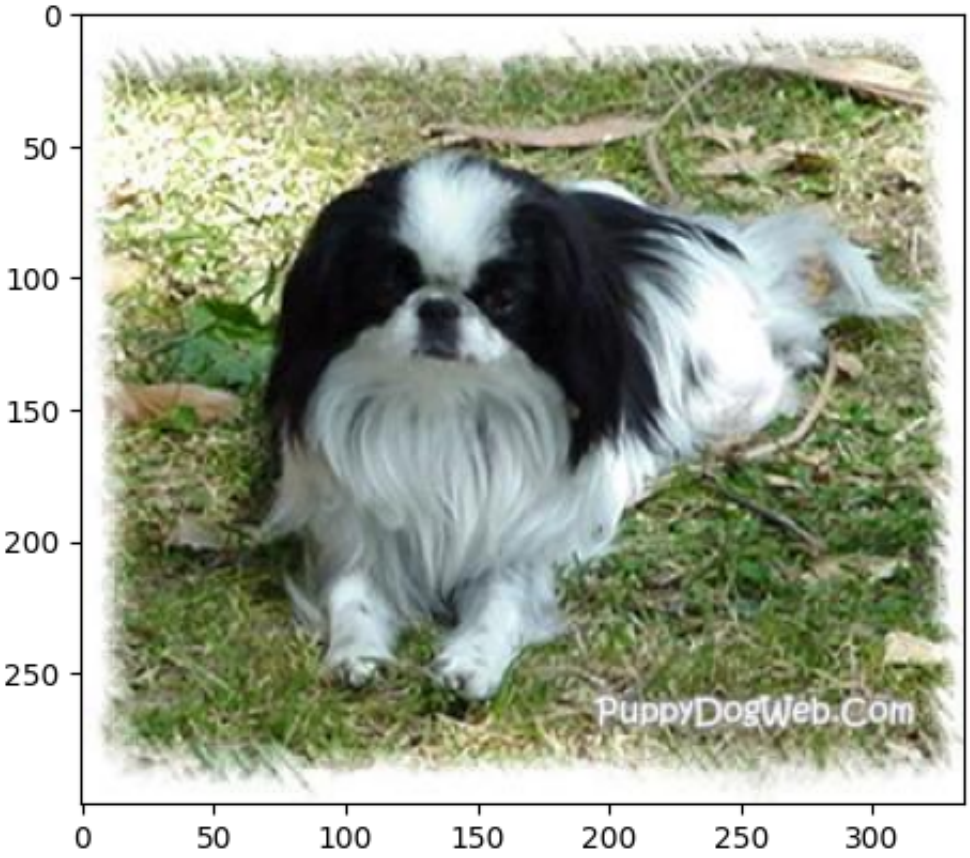
n02085782-Japanese_spaniel



n02086910-papillon



n02085782-Japanese_spaniel

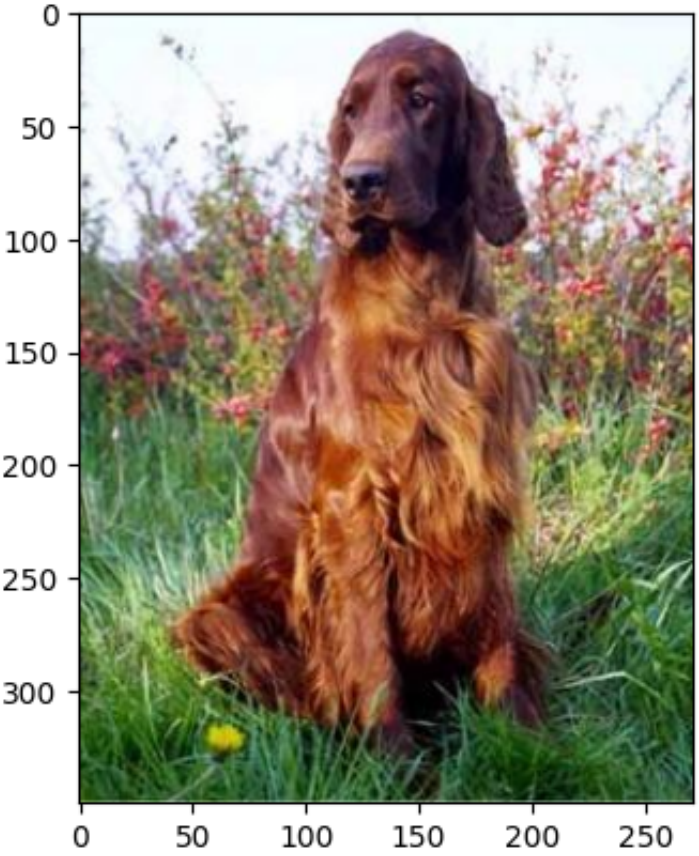


n02085782-Japanese_spaniel

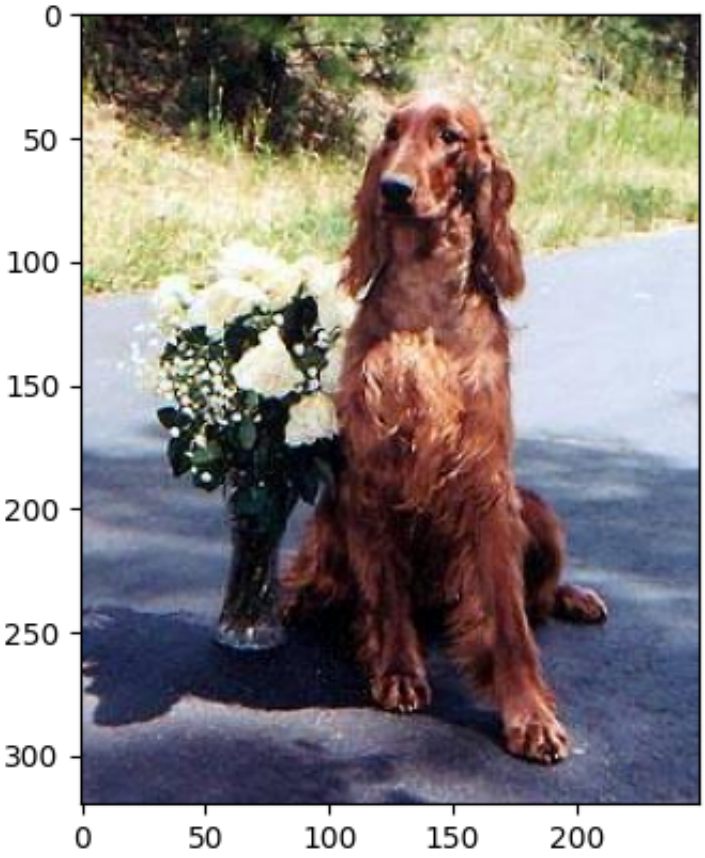
---

input:
n02100877-Irish_setter

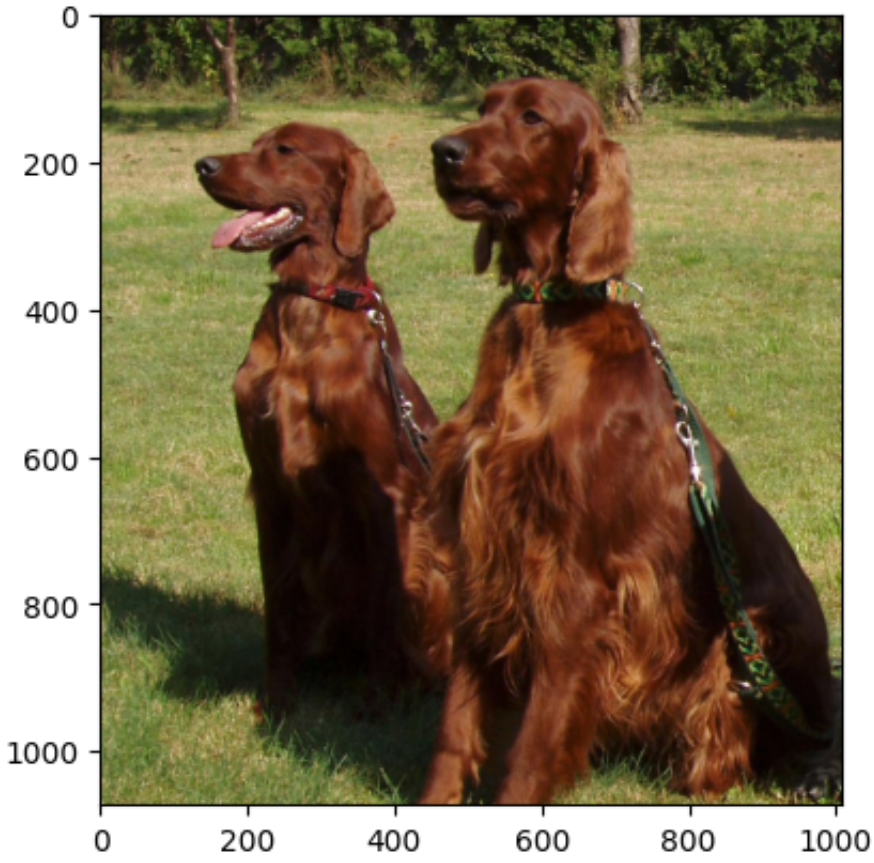
output:
n02100877-Irish_setter



n02100877-Irish_setter

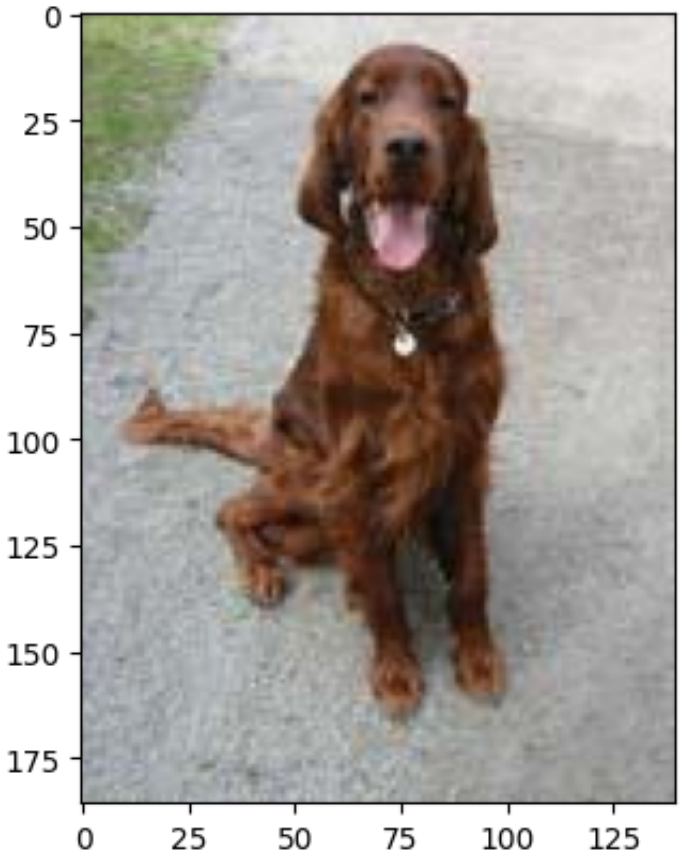


n02100877-Irish_setter



n02099267-flat-coated_retriever

n02100877-Irish_setter



```
--------
Count Of True: 44
Count Of False: 6
Count Of All: 50
```

Out[ ]:

input	output1	output2	output3	output4	
n02089973-English_foxhound	n02089973-English_foxhound	n02089867-Walker_hound	n02089973-English_foxhound	n02089973-English_foxhound	E
n02088094-Afghan_hound	n02088094-Afghan_hound	n02088094-Afghan_hound	n02088094-Afghan_hound	n02088094-Afghan_hound	
n02112137-chow	n02112137-chow	n02112137-chow	n02112137-chow	n02112137-chow	
n02111129-Leonberg	n02111129-Leonberg	n02111129-Leonberg	n02111129-Leonberg	n02111129-Leonberg	
n02088094-Afghan_hound	n02088094-Afghan_hound	n02088094-Afghan_hound	n02088094-Afghan_hound	n02088094-Afghan_hound	
n02104029-kuvasz	n02104029-kuvasz	n02111500-Great_Pyrenees	n02104029-kuvasz	n02104029-kuvasz	
n02095889-Sealyham_terrier	n02095889-Sealyham_terrier	n02095889-Sealyham_terrier	n02095889-Sealyham_terrier	n02095889-Sealyham_terrier	
n02086240-Shih-Tzu	n02086240-Shih-Tzu	n02086240-Shih-Tzu	n02086240-Shih-Tzu	n02086240-Shih-Tzu	J
n02085782-Japanese_spaniel	n02085782-Japanese_spaniel	n02085782-Japanese_spaniel	n02086910-papillon	n02085782-Japanese_spaniel	J
n02100877-Irish_setter	n02100877-Irish_setter	n02100877-Irish_setter	n02100877-Irish_setter	n02099267-flat-coated_retriever	
50	0	0	0	0	

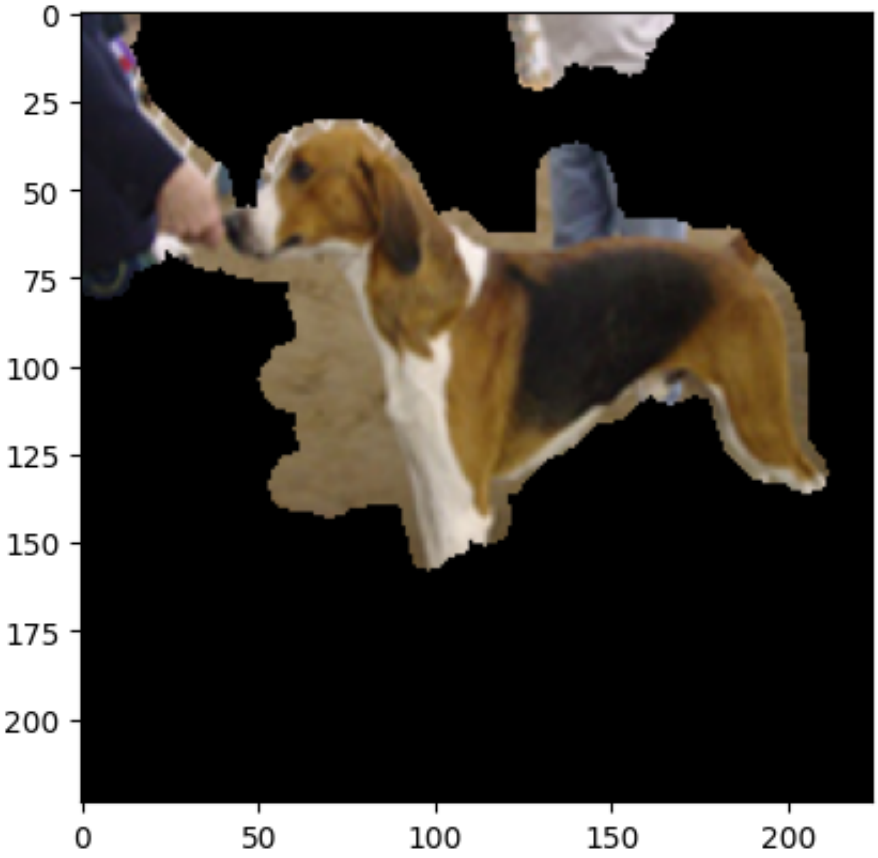
Saliency Part:

In [ ]:

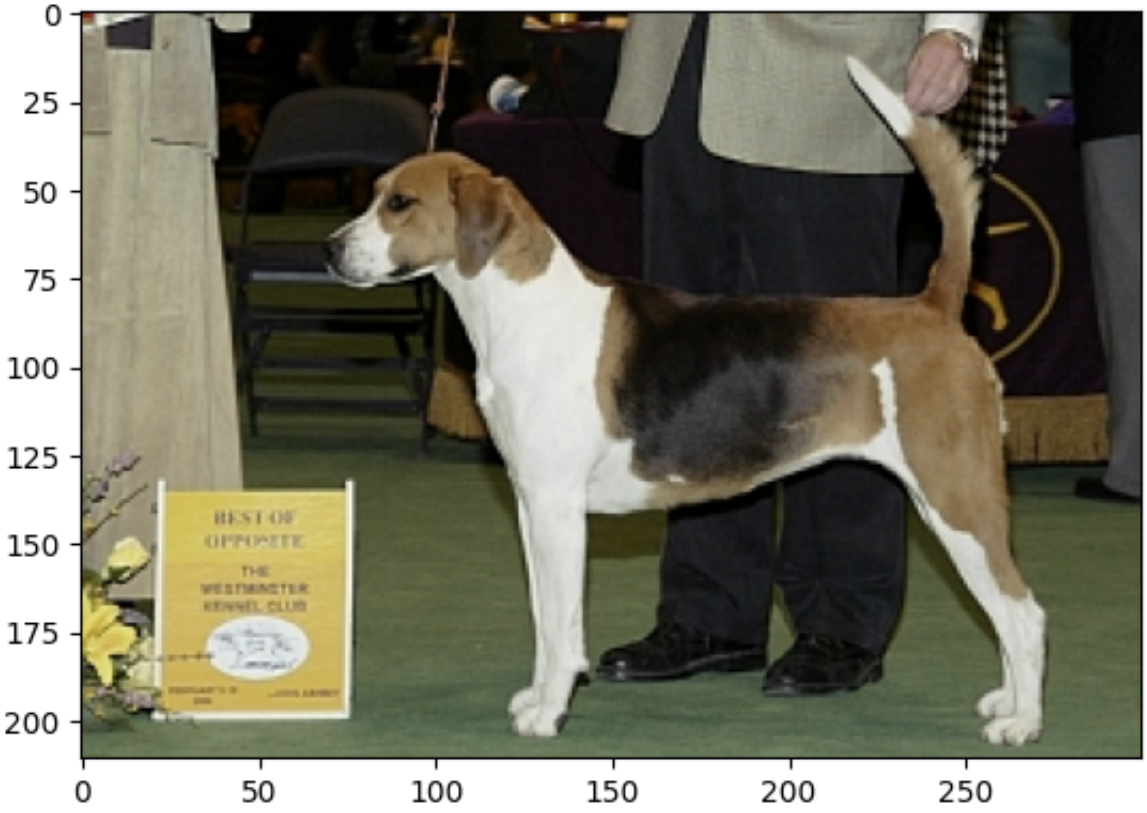
```
count_of_true=0
count_of_false=0
count_of_all=0
AllData=[['input','output1','output2','output3','output4','output5']]

for input in test:
    index_of_test=img_names.index(input)
    input_folder=img_folders[index_of_test]
    salient=slc.GetSaliency(('data-set/dogs/images/Images/'+input_folder+
    salient_feature=fe.extract_by_array(salient)
    input_ssa=SSA_H_Plus(salient_feature)
    thisrec=[input_folder]

    print('input:')
    print(input_folder)
    imgplot = plt.imshow(salient)
    plt.show()
    score_of_trained=[]
    for trained in training:
```

```
        index_of_trained=img_names.index(trained)
        trained_ssa=SSA_features[index_of_trained]
        distance= sqrt( sum(np.multiply(trained_ssa - input_ssa,trained_s
        trained_folder=img_folders[index_of_trained]
        score_of_trained.append([distance,trained,trained_folder])
    score_of_trained.sort()
    score_of_trained=score_of_trained[:count_of_results]
    new_rec={
        'input':input
        ,'class':input_folder,
        'output':[{'name':x[1],'class':x[2]} for x in score_of_trained]
    }
    print('output:')
    for res in score_of_trained:
        if(res[2]==input_folder):
            print(colored(res[2],'green'))
            count_of_true=count_of_true+1
        else:
            print(colored(res[2],'red'))
            count_of_false=count_of_false+1
        count_of_all=count_of_all+1
        thisrec.append(res[2])

        img=mpimg.imread(os.getcwd()+'/data-set/dogs/images/Images/'+res[
        imgplot=plt.imshow(img)
        plt.show()
        print('----')
    AllData.append(thisrec)
AllData.append([count_of_all,0,0,0,0,count_of_true])
print('Count Of True: '+str(count_of_true))
print('Count Of False: '+str(count_of_false))
print('Count Of All: '+str(count_of_all))
table = tabulate.tabulate(AllData, tablefmt='html')
table
```

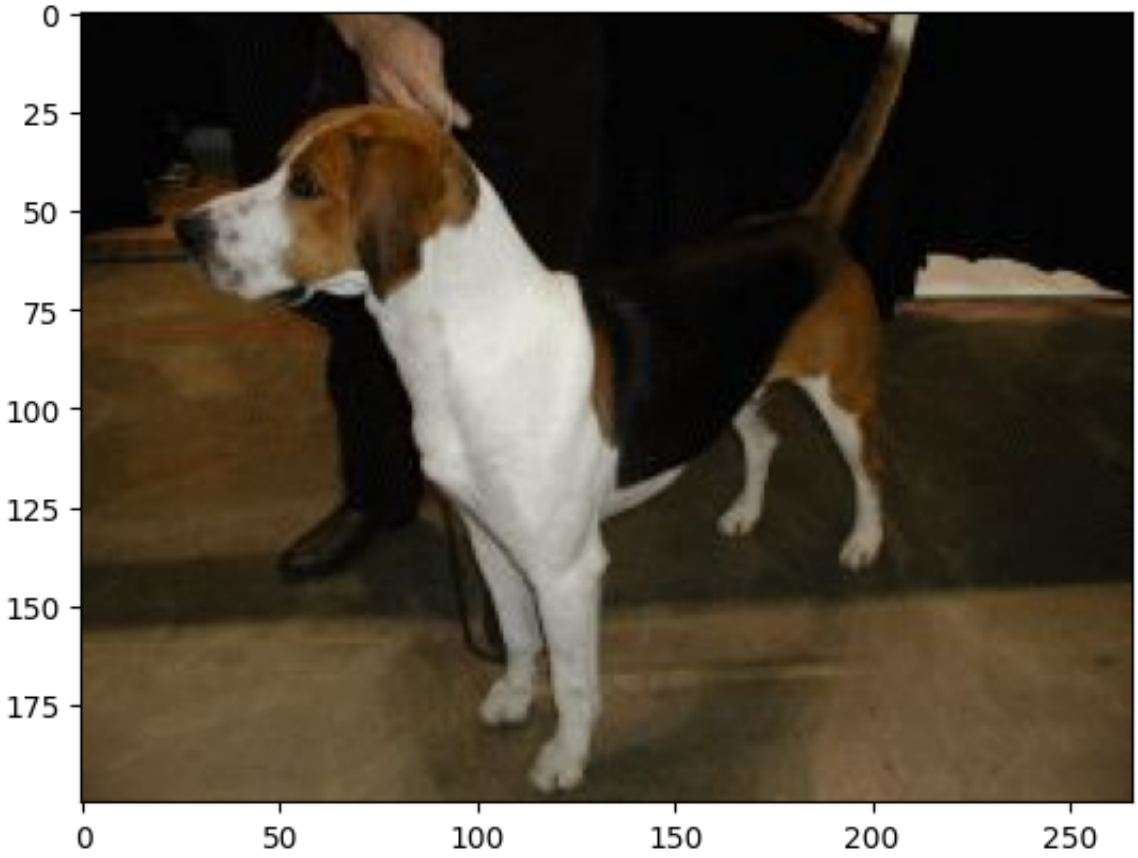
```
/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-p
ackages/saliency/core/xrai.py:126: FutureWarning: `selem` is a deprecate
d argument name for `dilation`. It will be removed in version 1.0. Pleas
e use `footprint` instead.
  masks = [dilation(mask, selem=selem) for mask in masks]
2022-10-25 11:52:58.007265: W tensorflow/core/platform/profile_utils/cpu
_utils.cc:128] Failed to get CPU frequency: 0 Hz
1/1 [==============================] - 0s 249ms/step
input:
n02089973-English_foxhound
```



output:
n02089973-English_foxhound



---
n02089973-English_foxhound



---

n02089973-English_foxhound

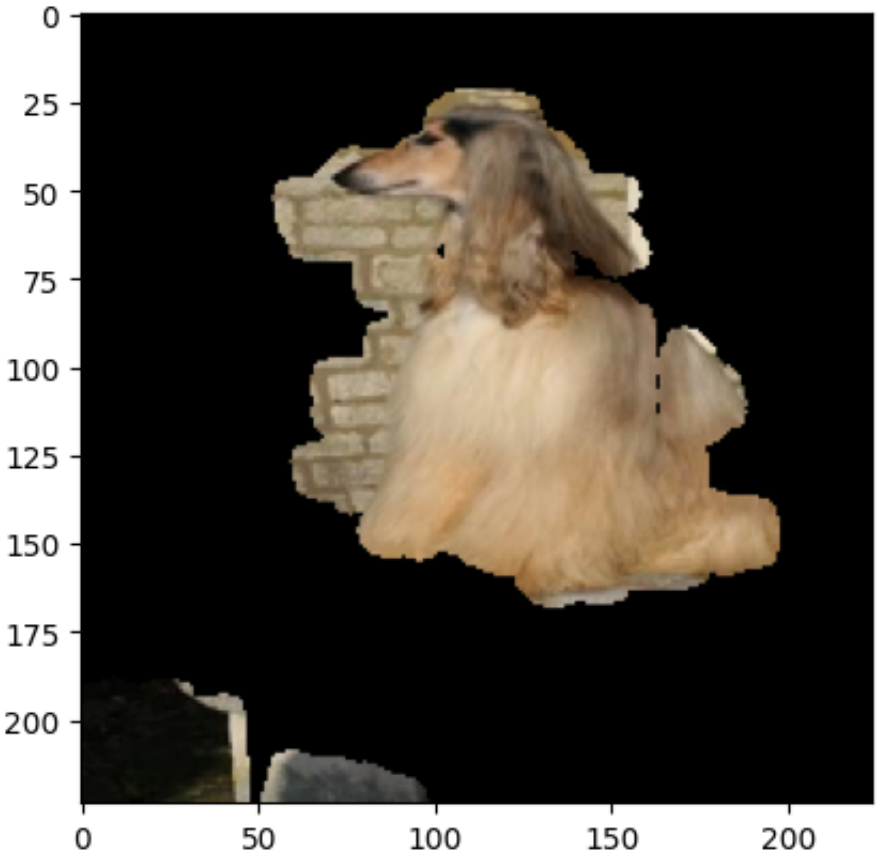
---

n02089867-Walker_hound

---

n02089973-English_foxhound

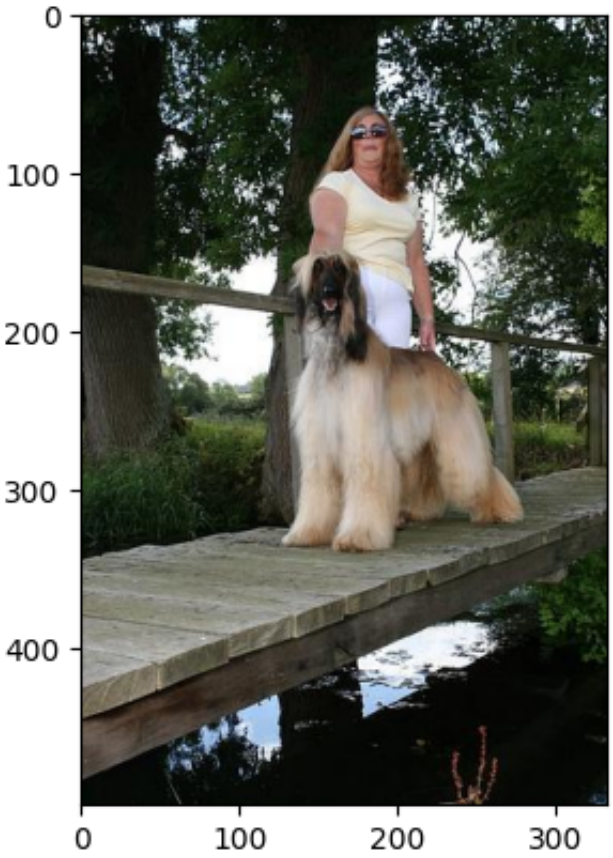
```
____
1/1 [==============================] - 0s 129ms/step
input:
n02088094-Afghan_hound
```




```
output:
n02112018-Pomeranian
```

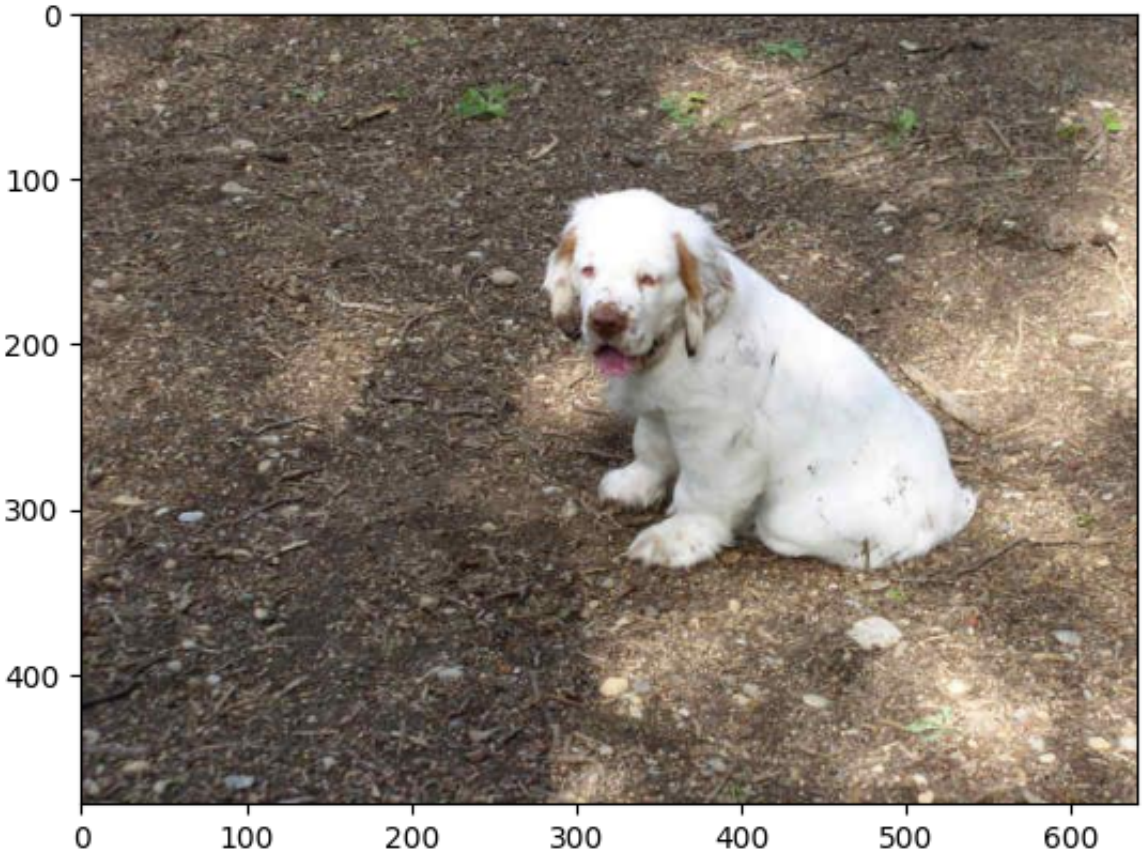
---

n02088094-Afghan_hound



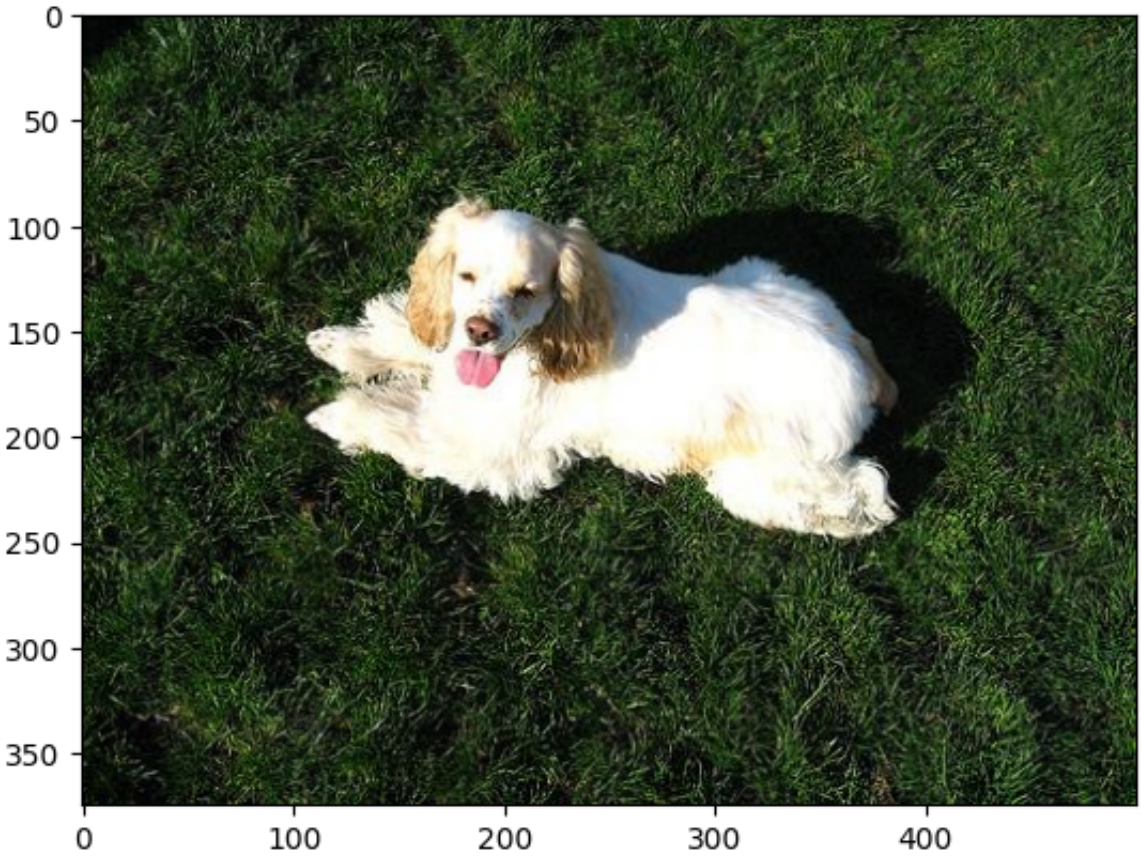
---

n02101556-clumber



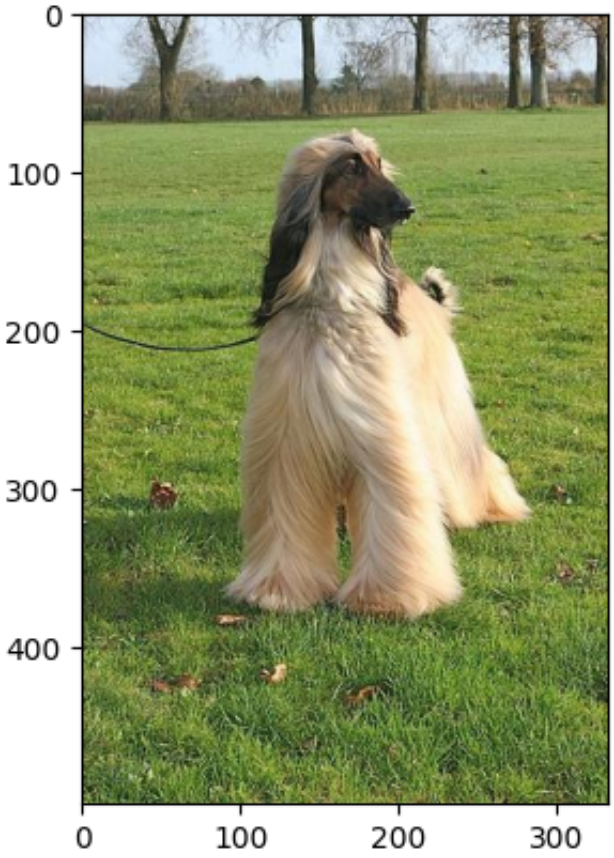
---

n02102318-cocker_spaniel

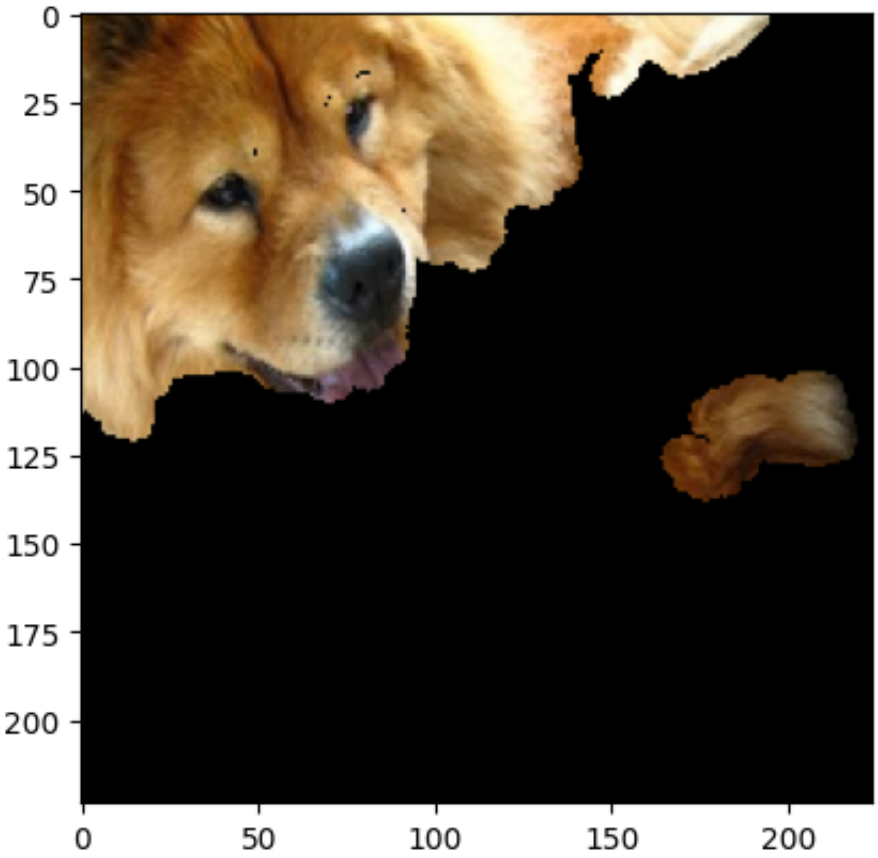


---

n02088094-Afghan_hound



```
____
1/1 [==============================] - 0s 133ms/step
input:
n02112137-chow
```



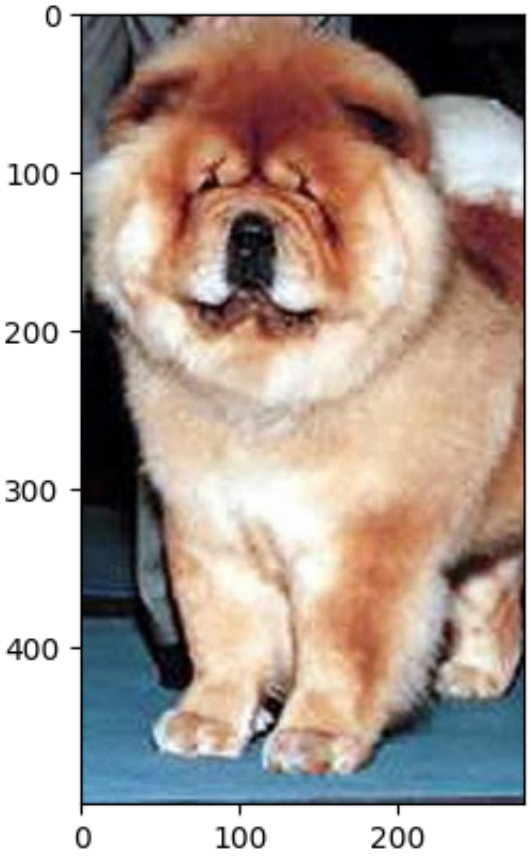

```
output:
n02112137-chow
```

---

n02112137-chow

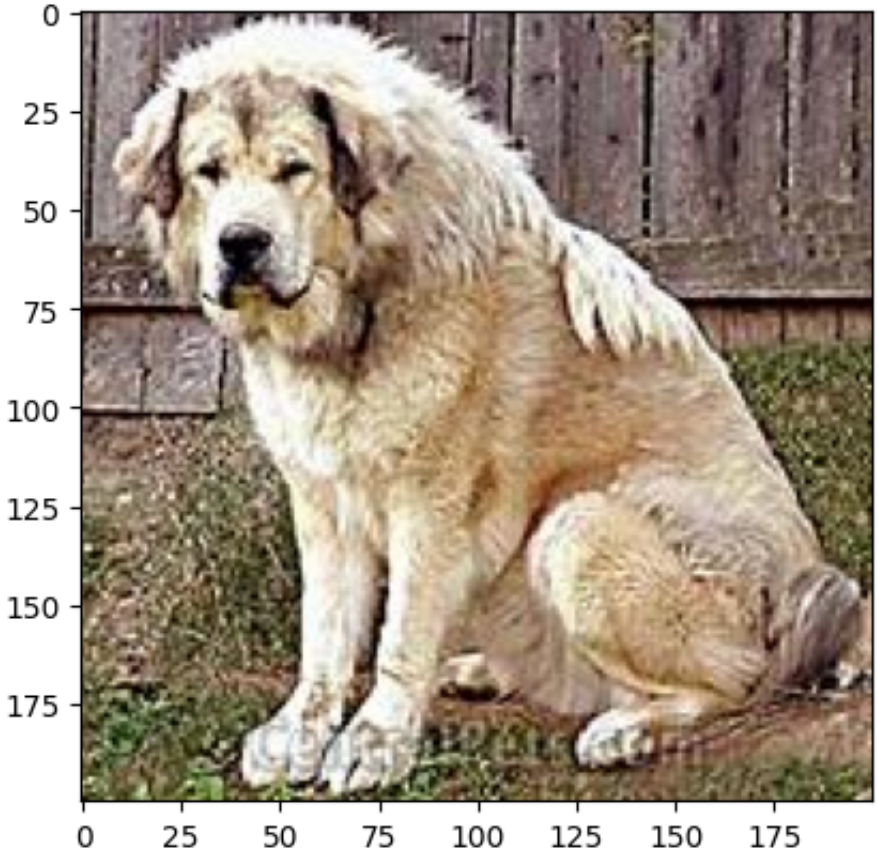
---

n02112137-chow



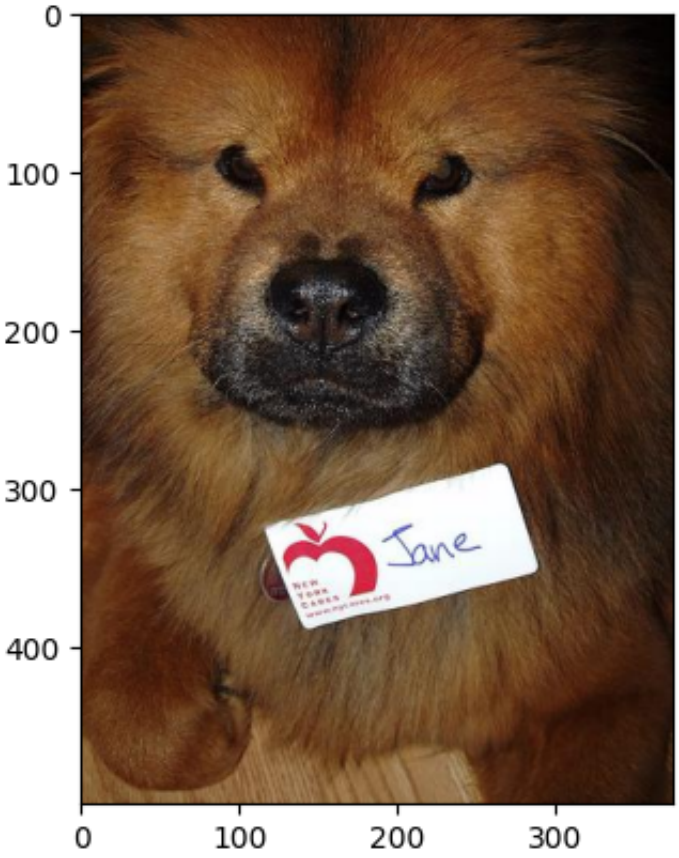
---

n02108551-Tibetan_mastiff

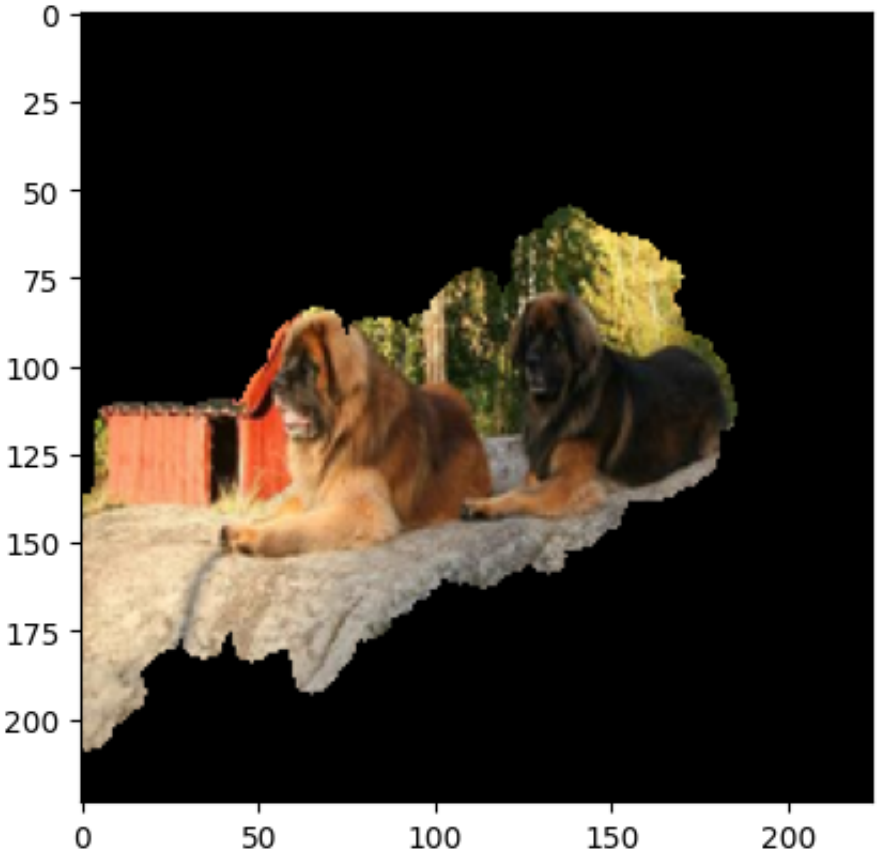


---

n02112137-chow

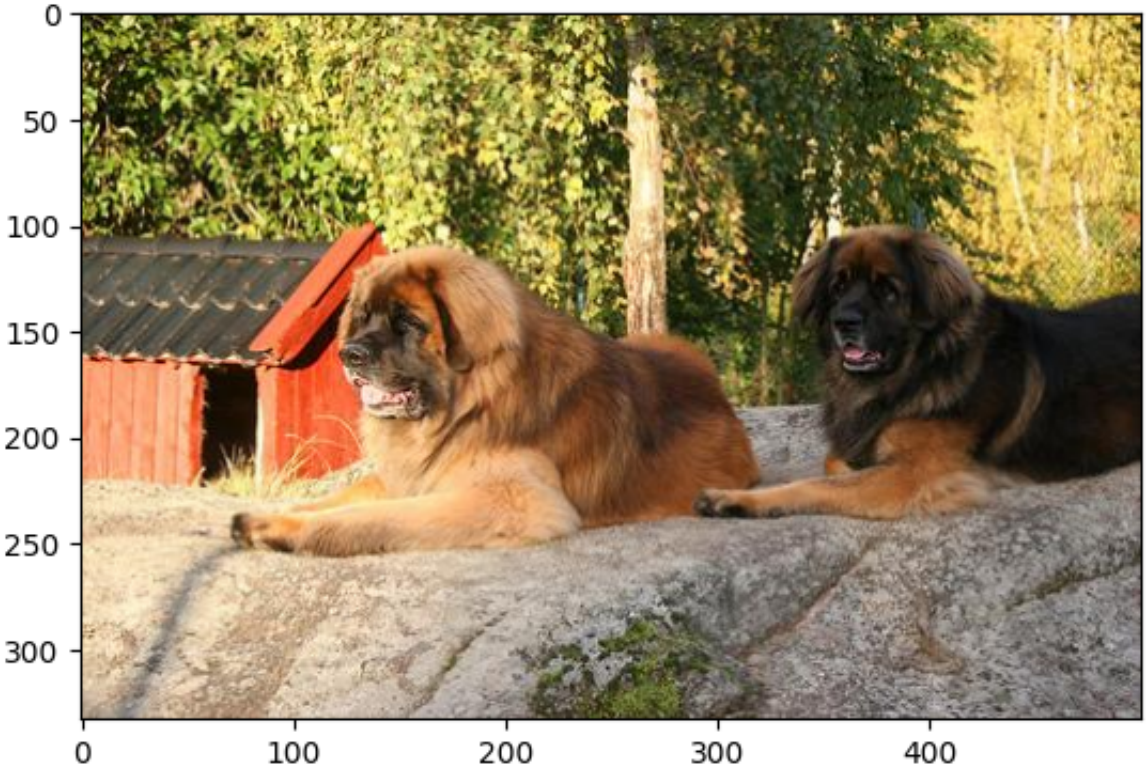



```
____
1/1 [==============================] - 0s 133ms/step
input:
n02111129-Leonberg
```

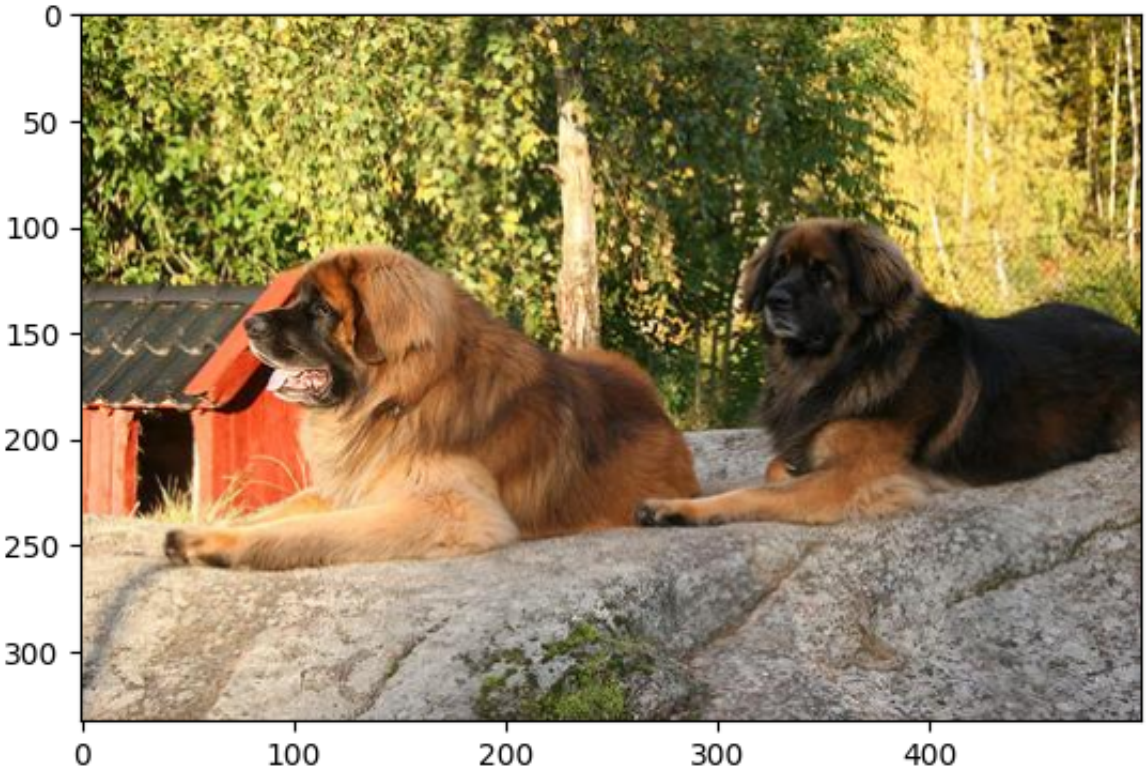


output:
n02111129-Leonberg

---
n02111129-Leonberg

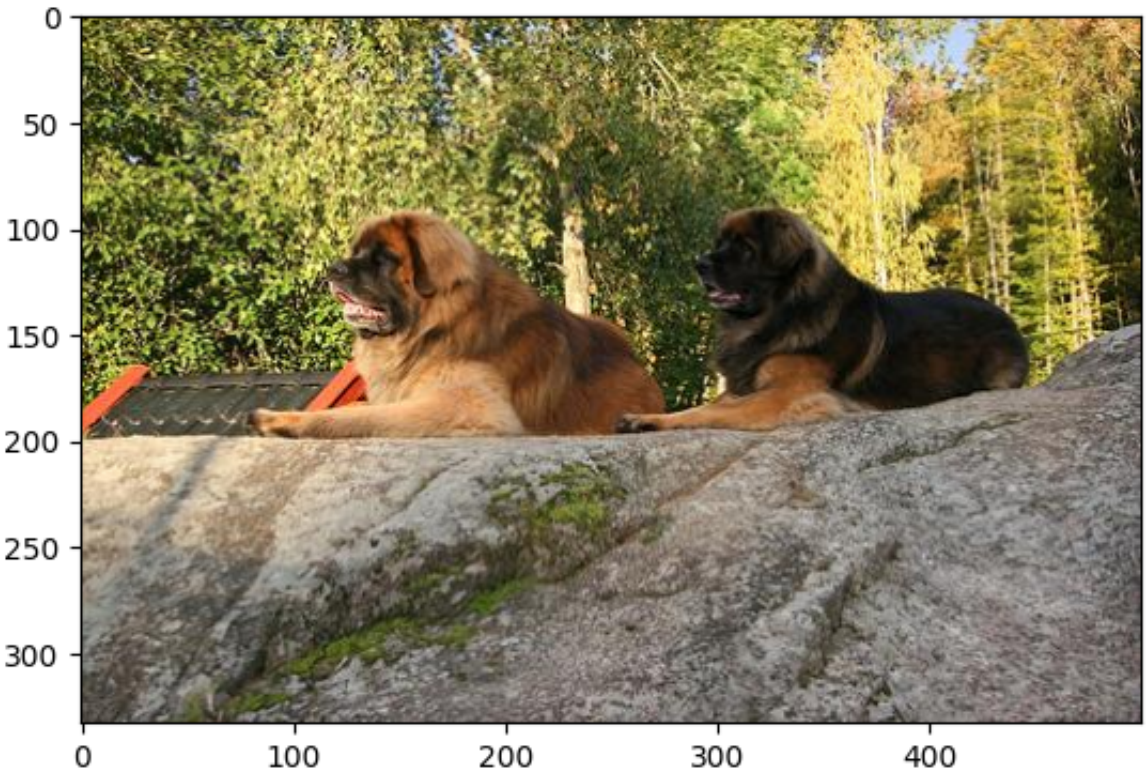


---
n02111129-Leonberg



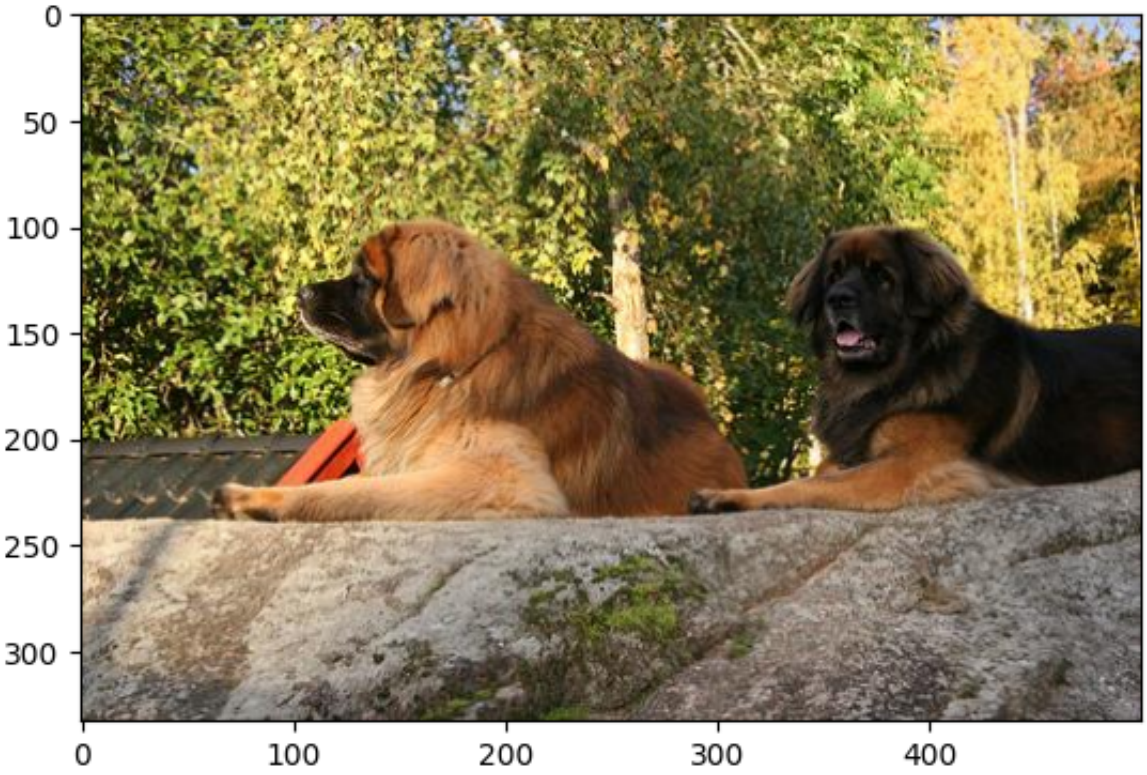
---

n02111129-Leonberg

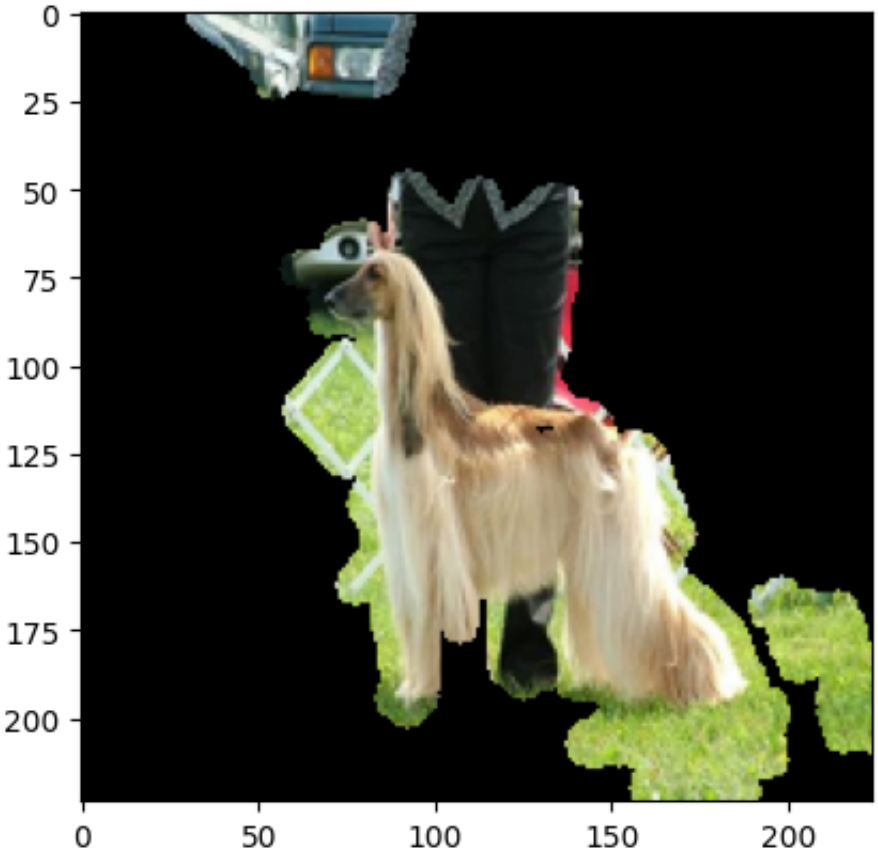


---

n02111129-Leonberg




```
____
1/1 [==============================] - 0s 134ms/step
input:
n02088094-Afghan_hound
```




```
output:
n02088094-Afghan_hound
```

---

n02088094-Afghan_hound

---

n02088094-Afghan_hound

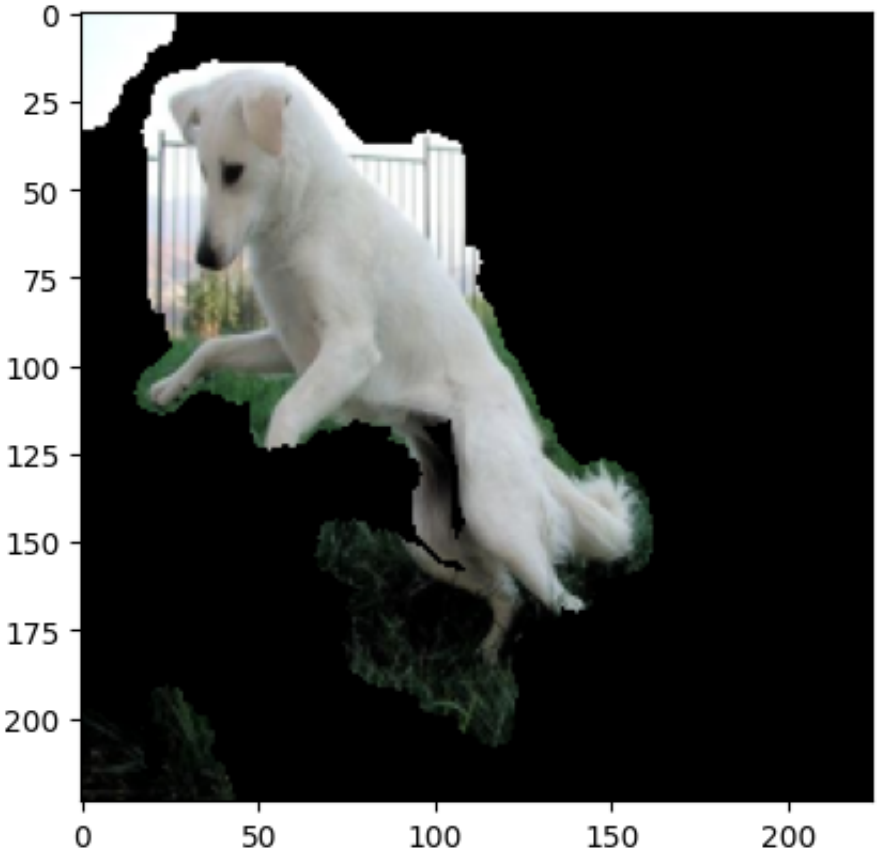
---

n02088094-Afghan_hound

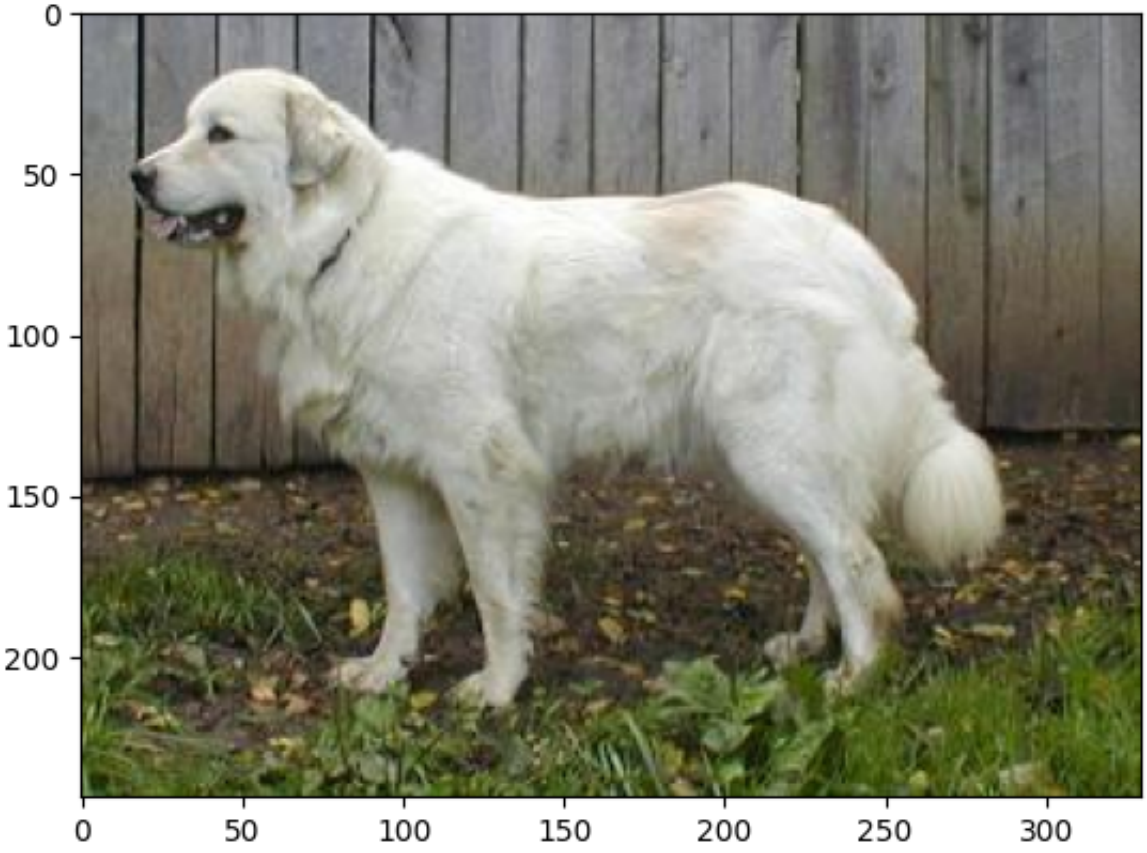
---

n02090622-borzoi


```
____
1/1 [==============================] - 0s 136ms/step
input:
n02104029-kuvasz
```




```
output:
n02111500-Great_Pyrenees
```

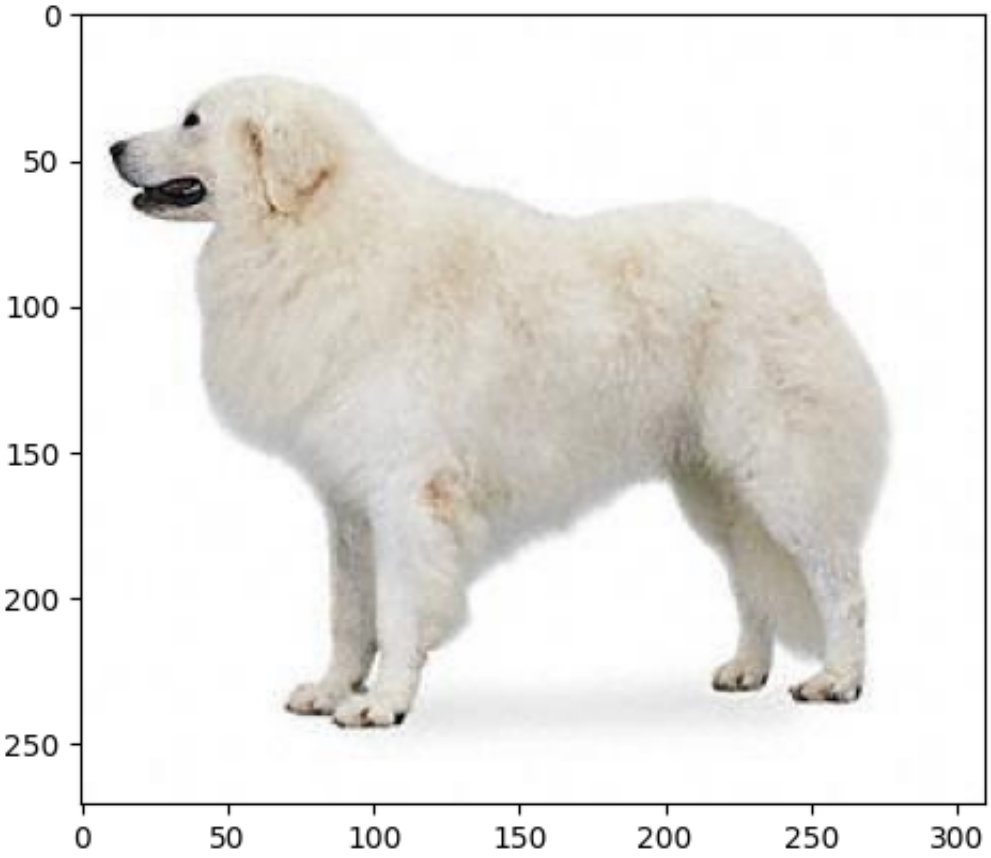


---

n02104029-kuvasz

---

n02104029-kuvasz

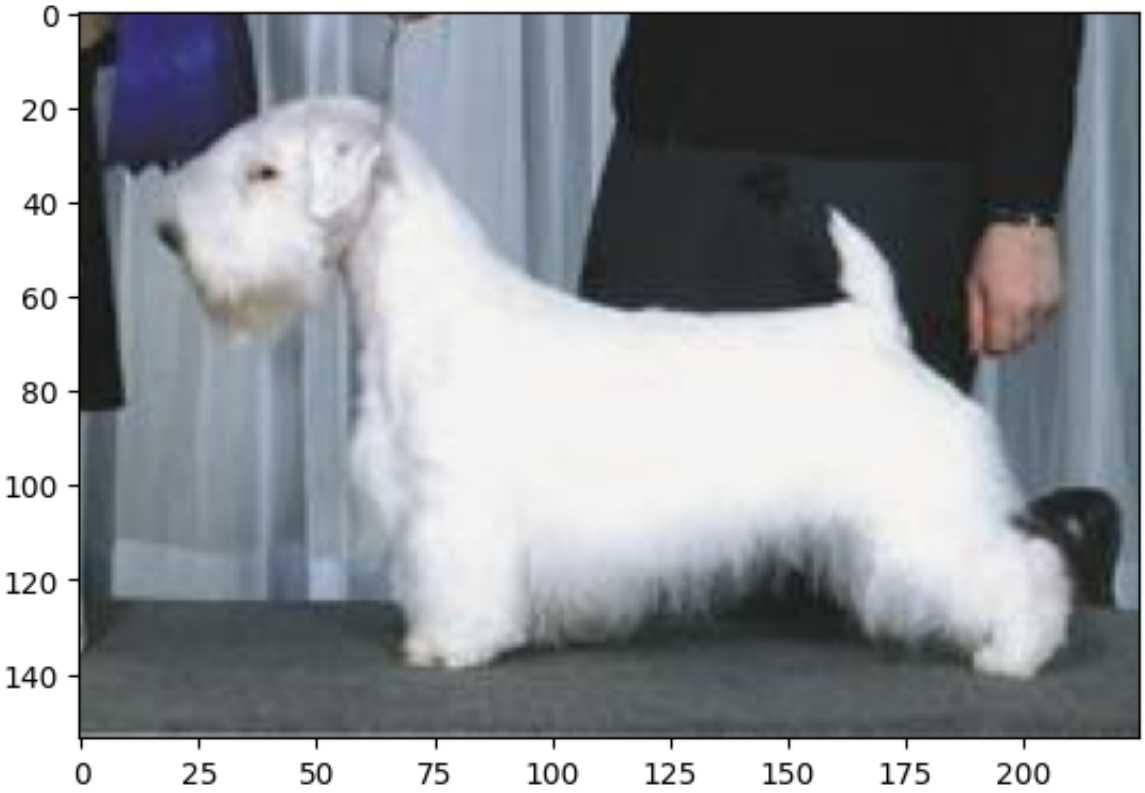


____

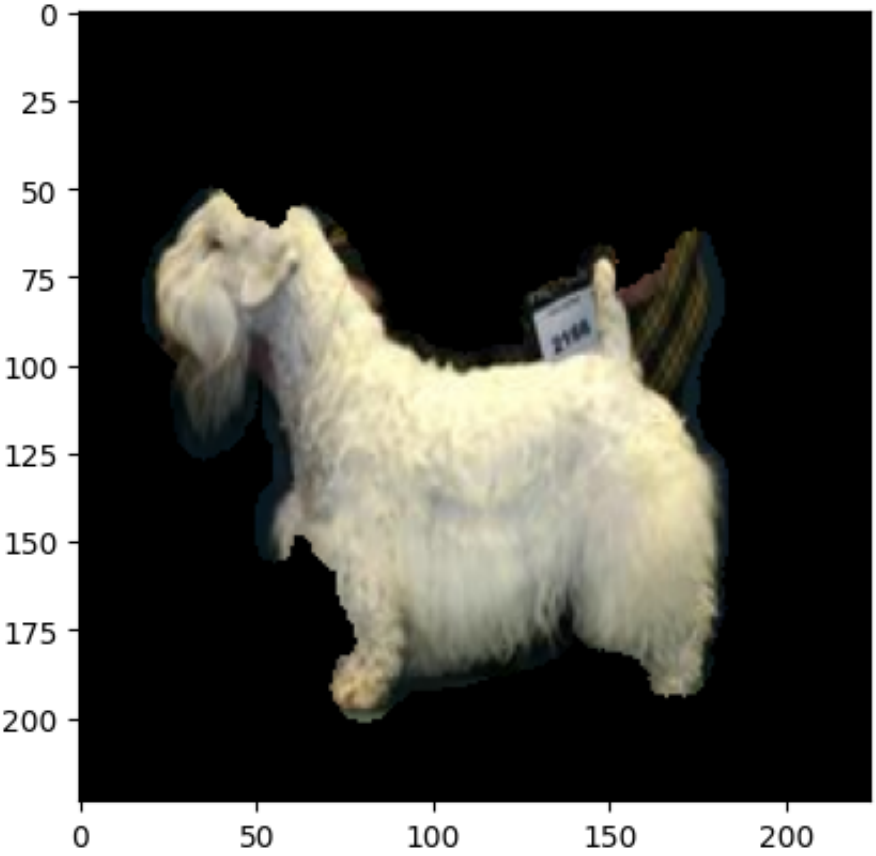
n02104029-kuvasz

____

n02095889-Sealyham_terrier




```
____
1/1 [==============================] - 0s 139ms/step
input:
n02095889-Sealyham_terrier
```



output:
n02095889-Sealyham_terrier

---

n02095889-Sealyham_terrier

---

n02095889-Sealyham_terrier

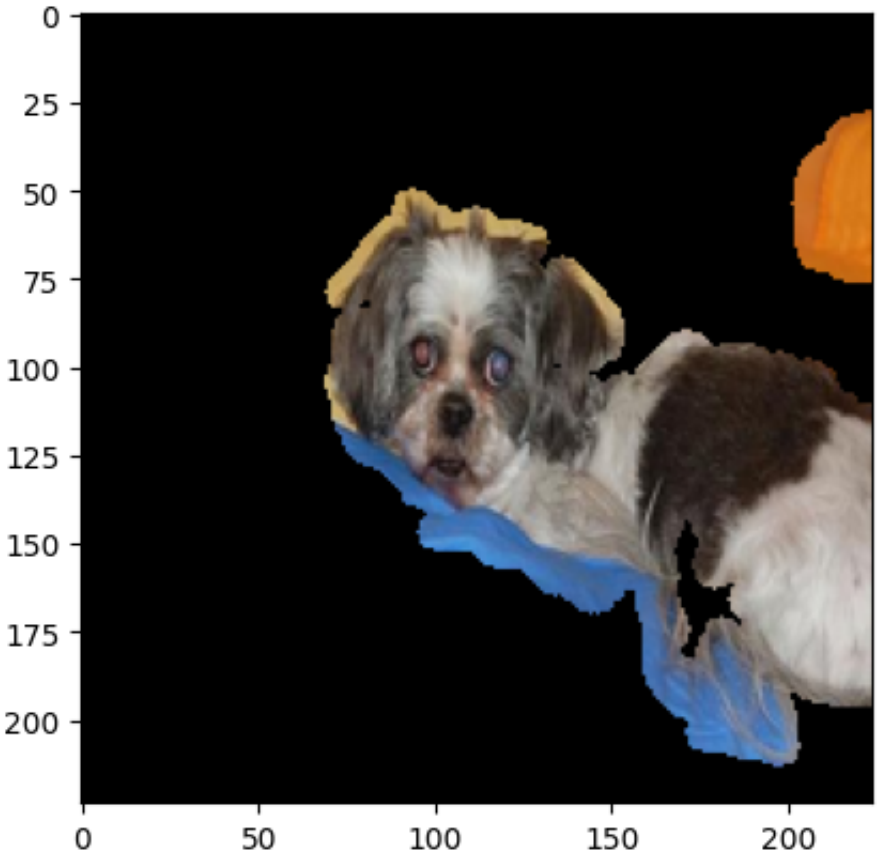
---

n02095889-Sealyham_terrier

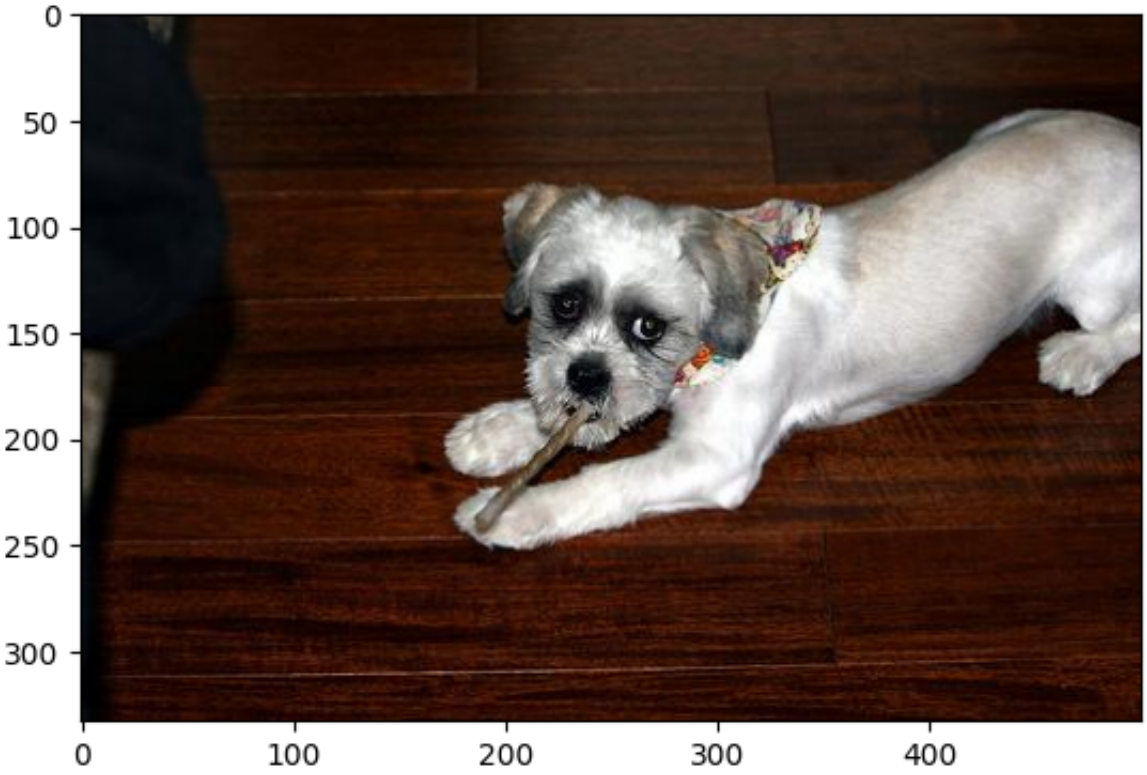
---

n02095889-Sealyham_terrier


```
____
1/1 [==============================] - 0s 137ms/step
input:
n02086240-Shih-Tzu
```




```
output:
n02086240-Shih-Tzu
```

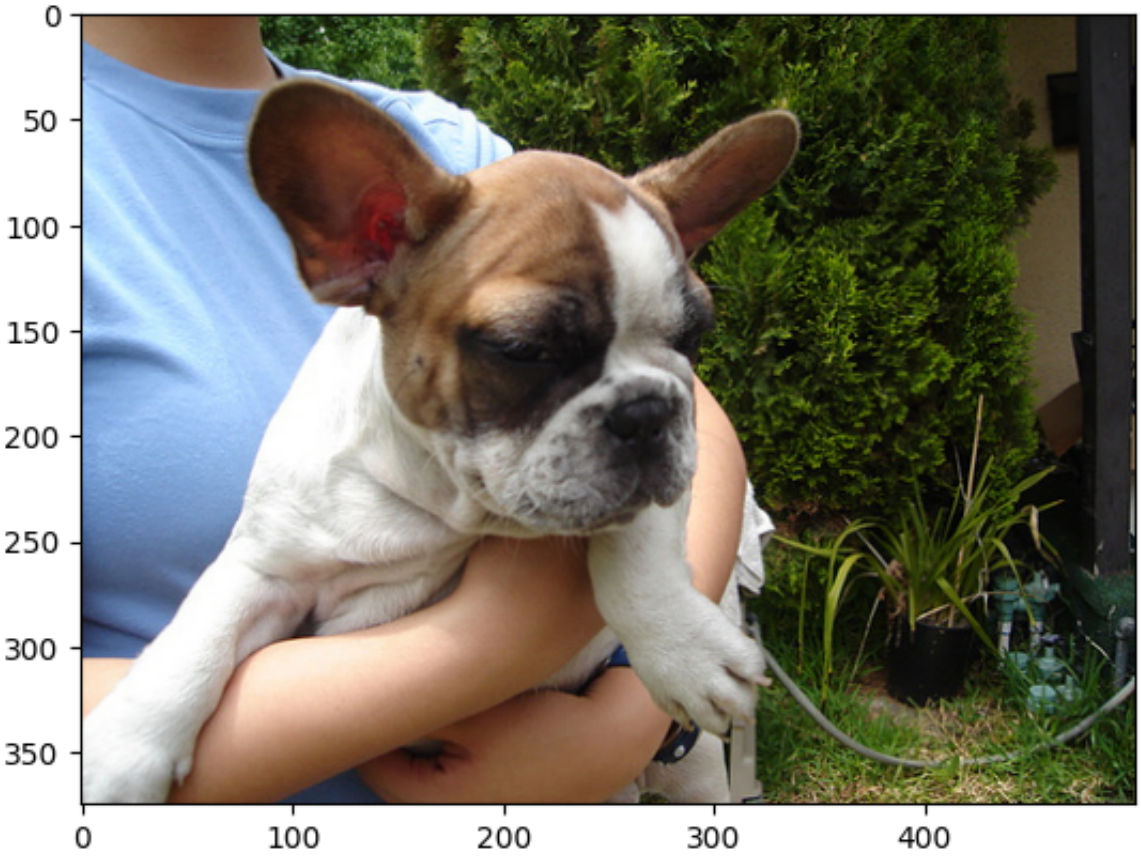


---

n02086240-Shih-Tzu

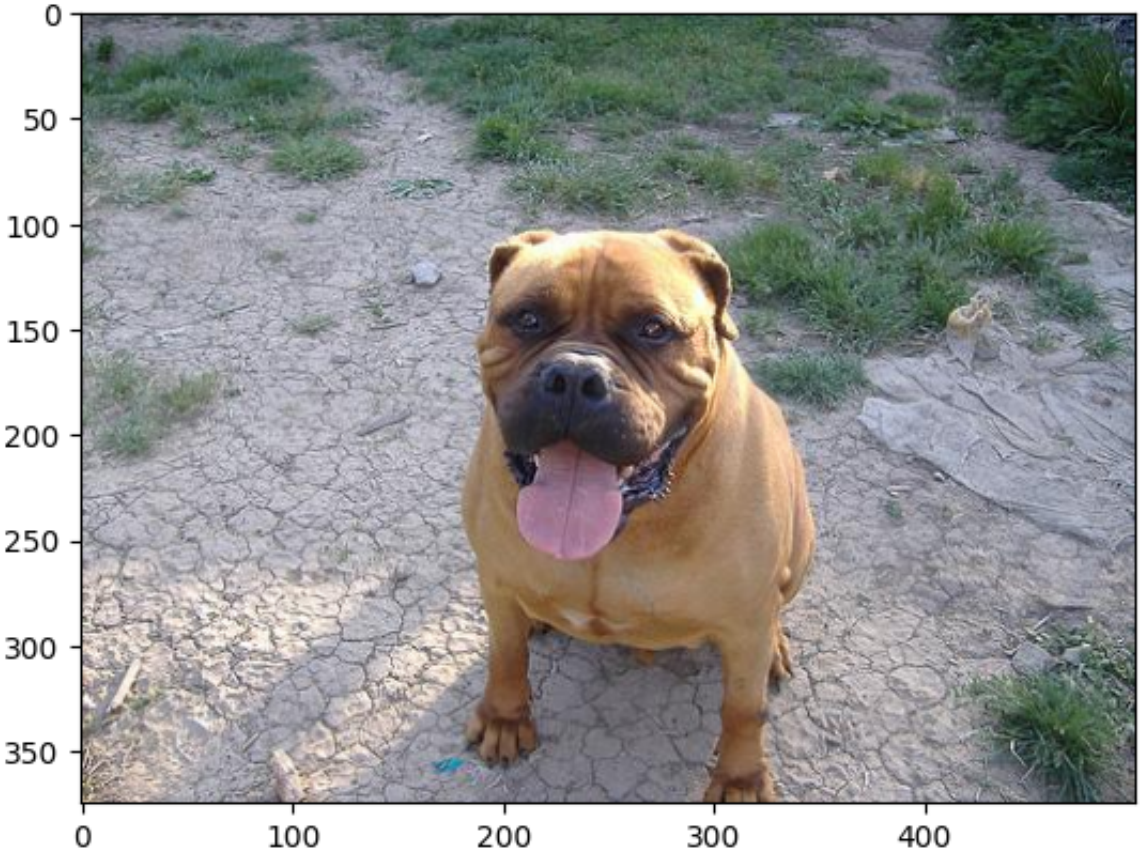
---

n02108915-French_bulldog



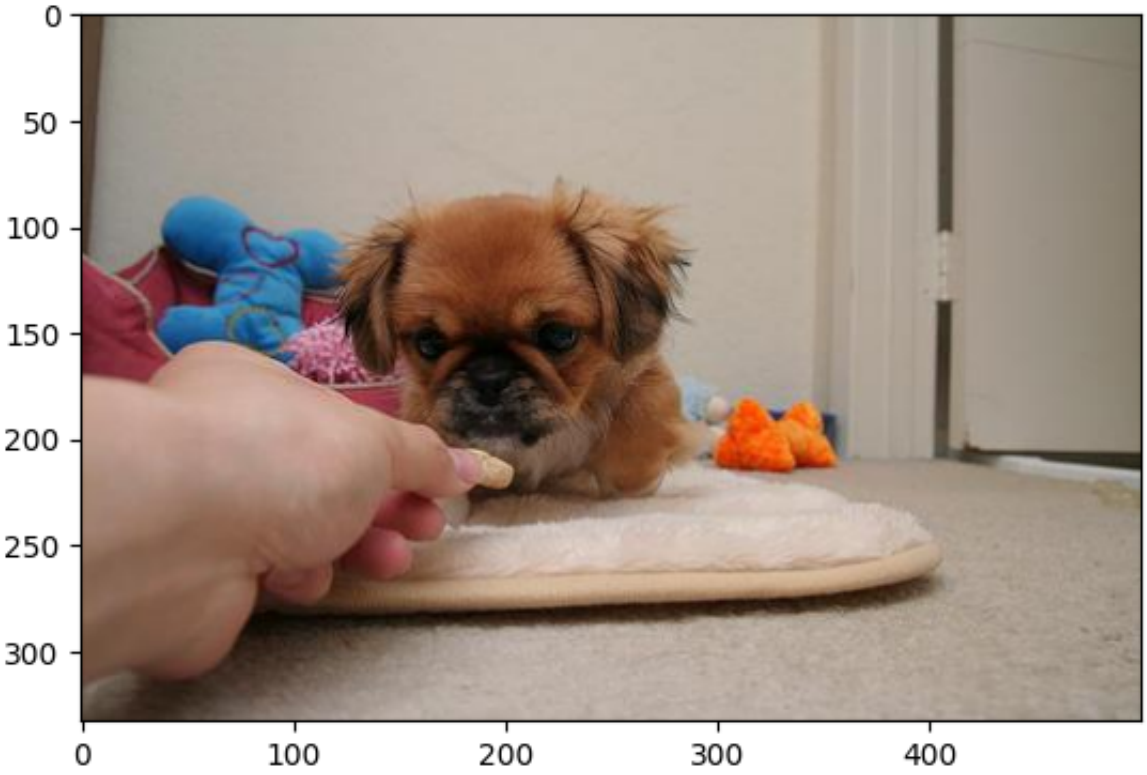
---

n02108422-bull_mastiff

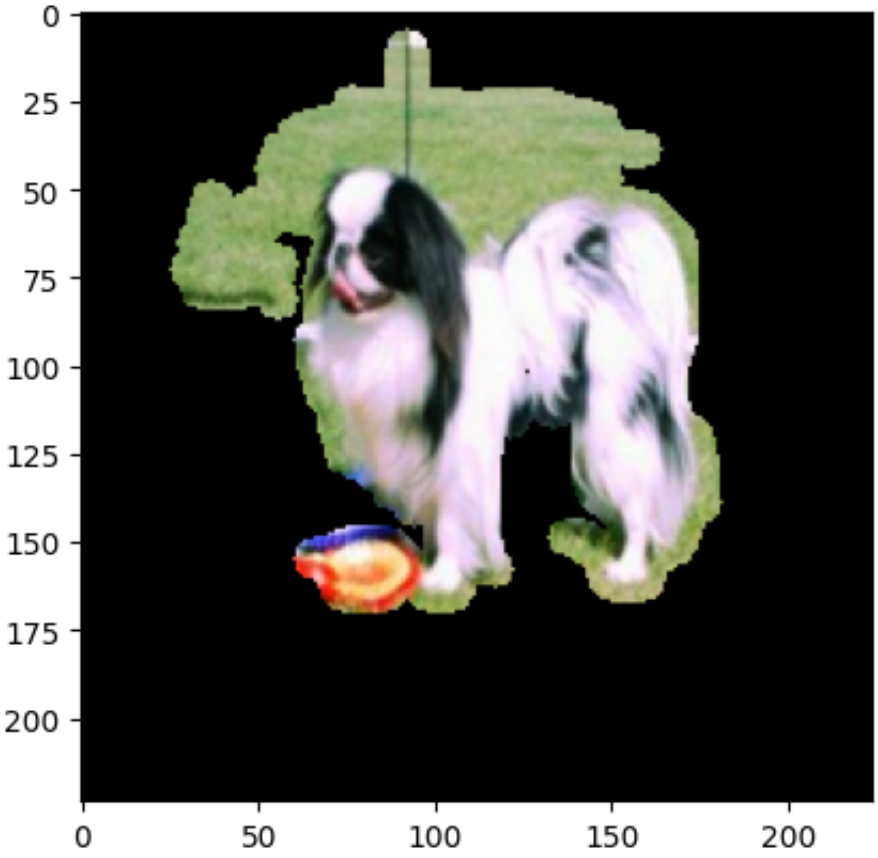


---

n02086079-Pekinese



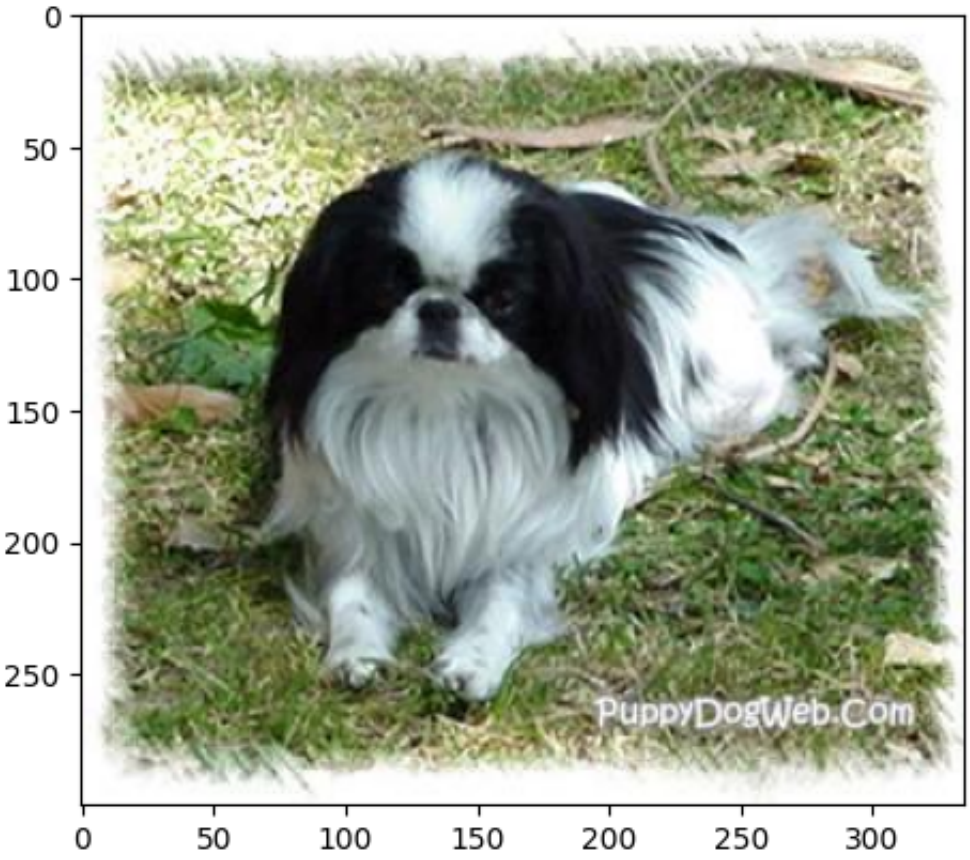

```
____
1/1 [==============================] - 0s 132ms/step
input:
n02085782-Japanese_spaniel
```




```
output:
n02085782-Japanese_spaniel
```

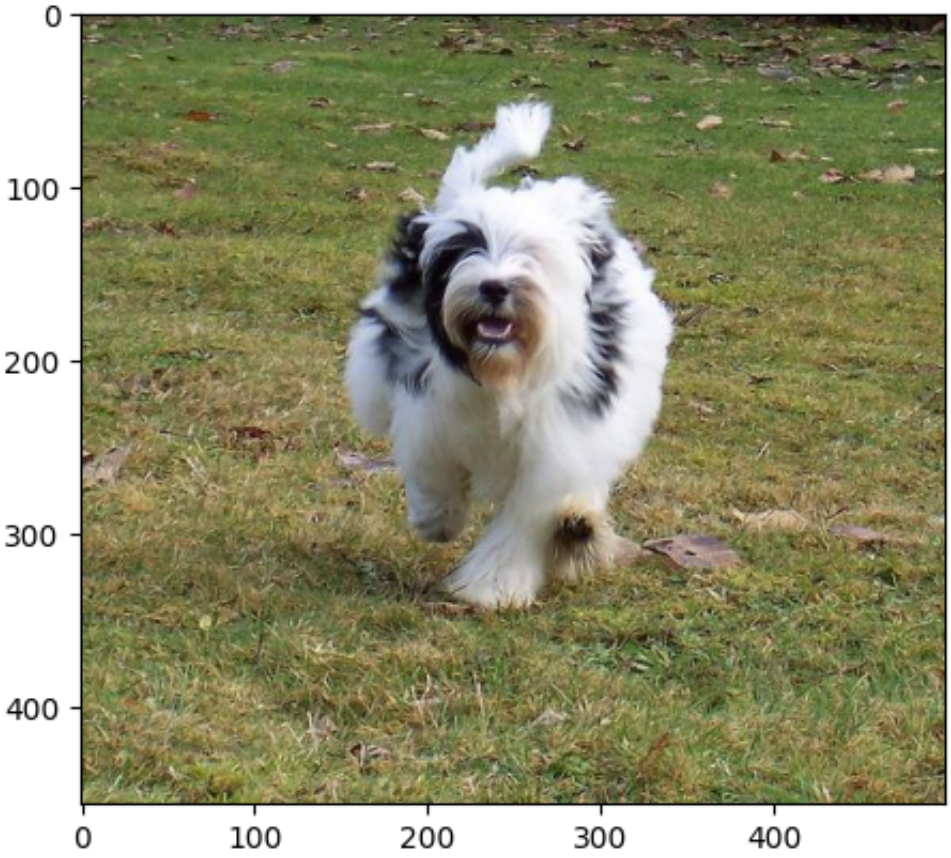
---

n02085782-Japanese_spaniel



---

n02097474-Tibetan_terrier

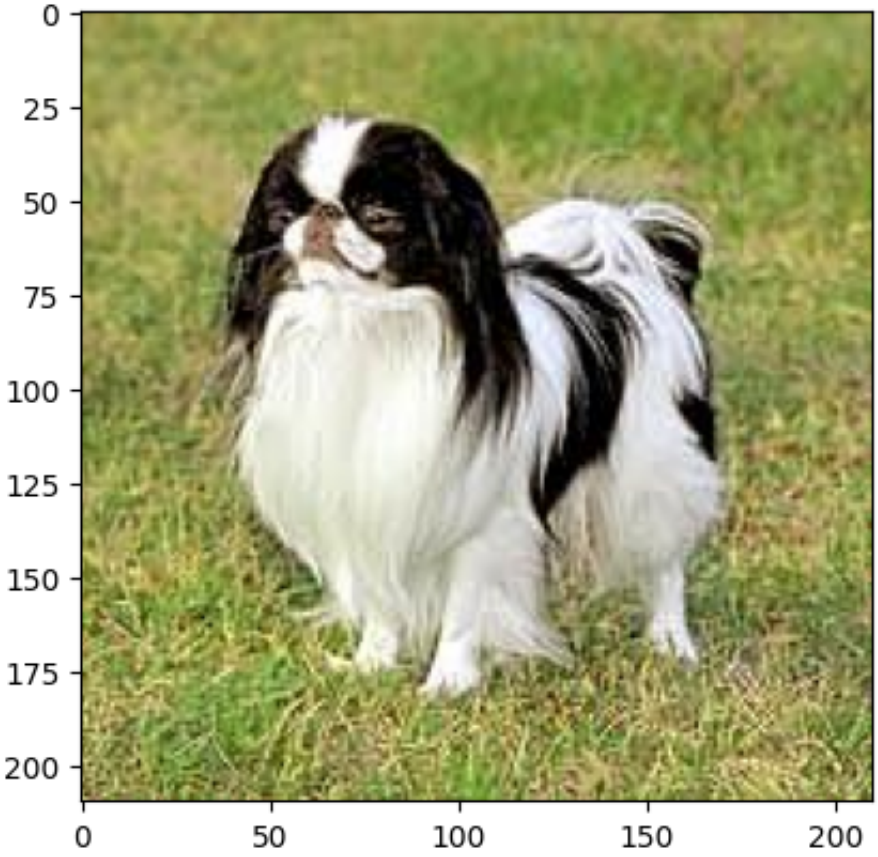


---

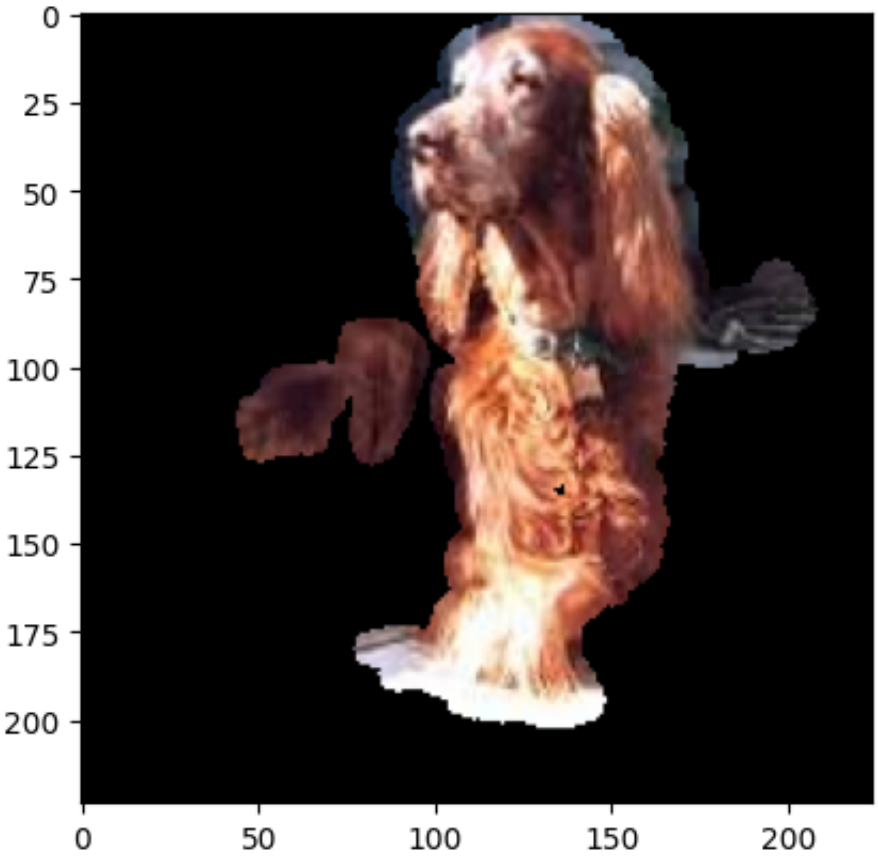
n02085782-Japanese_spaniel

---

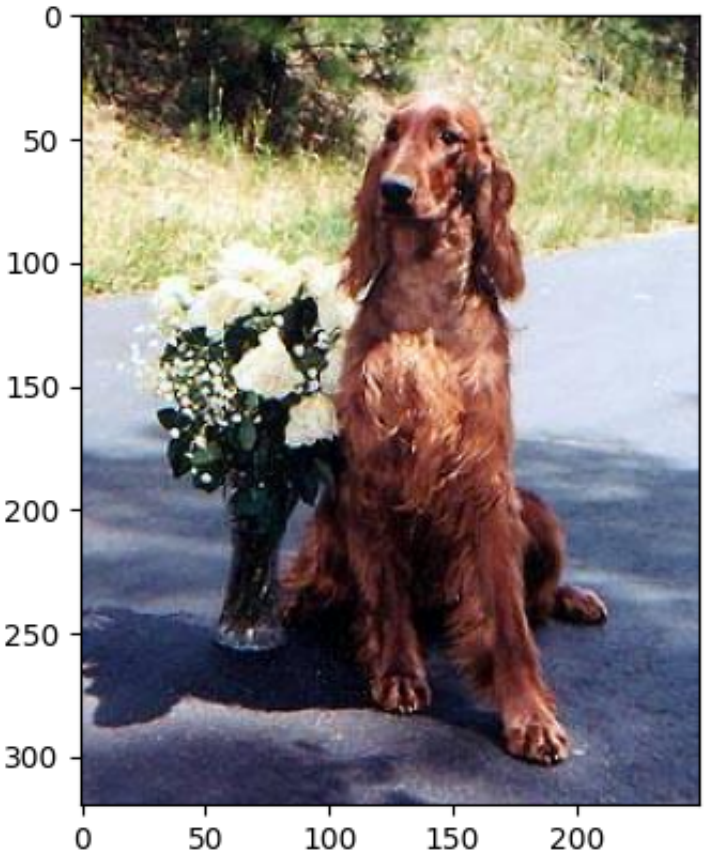
n02085782-Japanese_spaniel



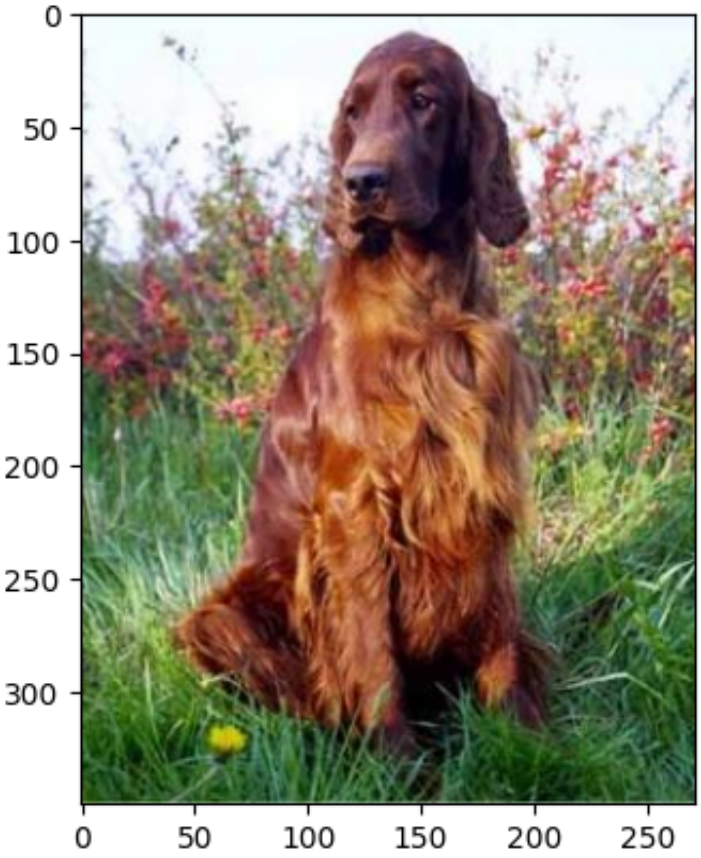

```
____
1/1 [==============================] - 0s 136ms/step
input:
n02100877-Irish_setter
```



output:
n02100877-Irish_setter

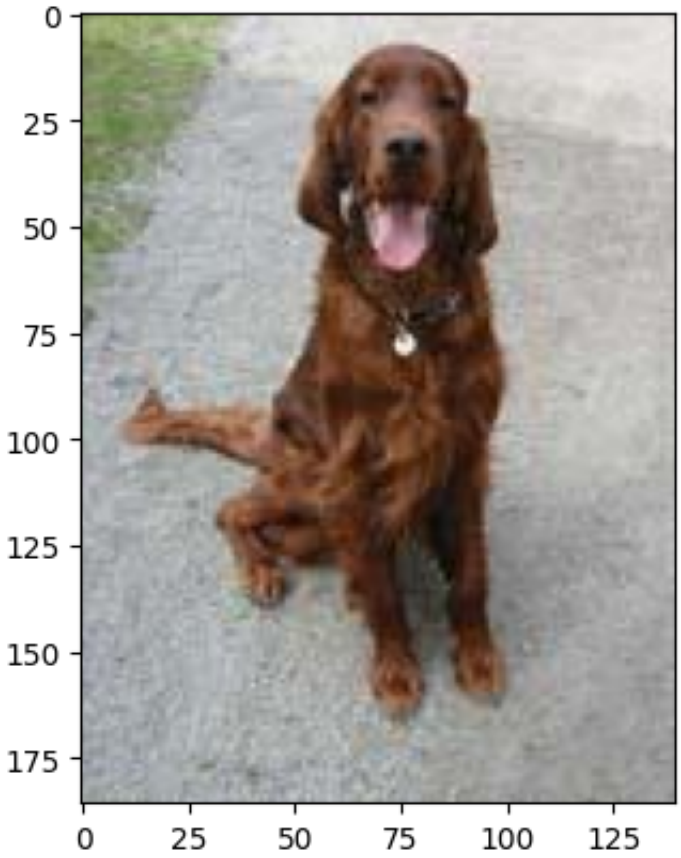


---
n02100877-Irish_setter



---

n02100877-Irish_setter



---

n02102480-Sussex_spaniel

---

n02102480-Sussex_spaniel

```
Count Of True: 36
Count Of False: 14
Count Of All: 50
```

Out[ ]:

input	output1	output2	output3	output4	
n02089973-English_foxhound	n02089973-English_foxhound	n02089973-English_foxhound	n02089973-English_foxhound	n02089867-Walker_hound	E
n02088094-Afghan_hound	n02112018-Pomeranian	n02088094-Afghan_hound	n02101556-clumber	n02102318-cocker_spaniel	
n02112137-chow	n02112137-chow	n02112137-chow	n02112137-chow	n02108551-Tibetan_mastiff	
n02111129-Leonberg	n02111129-Leonberg	n02111129-Leonberg	n02111129-Leonberg	n02111129-Leonberg	
n02088094-Afghan_hound	n02088094-Afghan_hound	n02088094-Afghan_hound	n02088094-Afghan_hound	n02088094-Afghan_hound	
n02104029-kuvasz	n02111500-Great_Pyrenees	n02104029-kuvasz	n02104029-kuvasz	n02104029-kuvasz	S
n02095889-Sealyham_terrier	n02095889-Sealyham_terrier	n02095889-Sealyham_terrier	n02095889-Sealyham_terrier	n02095889-Sealyham_terrier	S
n02086240-Shih-Tzu	n02086240-Shih-Tzu	n02086240-Shih-Tzu	n02108915-French_bulldog	n02108422-bull_mastiff	
n02085782-Japanese_spaniel	n02085782-Japanese_spaniel	n02085782-Japanese_spaniel	n02097474-Tibetan_terrier	n02085782-Japanese_spaniel	J
n02100877-Irish_setter	n02100877-Irish_setter	n02100877-Irish_setter	n02100877-Irish_setter	n02102480-Sussex_spaniel	
50	0	0	0	0	