

Trần Ngọc Đoàn - 19146175 - PROJECT AI

Github: <https://github.com/DoanAI/PROJECT.git>

```
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
import numpy as np

from google.colab import drive
drive.mount('/content/drive/')

 Mounted at /content/drive/

import os
train_image_files_path = "/content/drive/MyDrive/AI/Project/training set"
valid_image_files_path = "/content/drive/MyDrive/AI/Project/test set"

from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_data_gen = ImageDataGenerator(rescale=1/255)
validation_data_gen = ImageDataGenerator(rescale=1/255)

train_generator = train_data_gen.flow_from_directory(train_image_files_path, target_size=(150, 150))
validation_generator = validation_data_gen.flow_from_directory(valid_image_files_path, target_size=(150, 150))

Found 2900 images belonging to 29 classes.
Found 580 images belonging to 29 classes.

#MoHinh
from keras.models import Sequential
from keras.layers import Dense, Dropout, Conv2D, MaxPooling2D, Flatten

model=Sequential()

model.add(Conv2D(32,(3,3), activation='relu',input_shape=(200,200,3)))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(64,(3,3), activation='relu'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(128,(3,3), activation='relu'))
model.add(MaxPooling2D(2,2))
model.add(Flatten())
model.add(Dense(512, activation=tf.nn.relu))
model.add(Dense(29, activation=tf.nn.softmax))
model.summary()

Model: "sequential"
```

Layer (type)	Output Shape	Param #
--------------	--------------	---------

```
=====
conv2d (Conv2D)           (None, 198, 198, 32)    896
max_pooling2d (MaxPooling2D (None, 99, 99, 32)    0
)
conv2d_1 (Conv2D)          (None, 97, 97, 64)    18496
max_pooling2d_1 (MaxPooling  (None, 48, 48, 64)    0
2D)
conv2d_2 (Conv2D)          (None, 46, 46, 128)   73856
max_pooling2d_2 (MaxPooling  (None, 23, 23, 128)   0
2D)
flatten (Flatten)          (None, 67712)        0
dense (Dense)              (None, 512)          34669056
dense_1 (Dense)             (None, 29)          14877
=====
Total params: 34,777,181
Trainable params: 34,777,181
Non-trainable params: 0
```

---

```
from tensorflow.keras.optimizers import Adam
model.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=[accuracy])

EPOCHS=100
history=model.fit(train_generator, steps_per_epoch=3, epochs=EPOCHS, verbose=1,
                   validation_data = validation_generator, validation_steps=3)

3/3 [=====] - 3s 896ms/step - loss: 0.9490 - acc: 0.7188
Epoch 73/100
3/3 [=====] - 2s 889ms/step - loss: 0.8009 - acc: 0.7917
Epoch 74/100
3/3 [=====] - 5s 2s/step - loss: 1.1155 - acc: 0.6771
Epoch 75/100
3/3 [=====] - 2s 960ms/step - loss: 0.9264 - acc: 0.7396
Epoch 76/100
3/3 [=====] - 3s 1s/step - loss: 0.9064 - acc: 0.7396
Epoch 77/100
3/3 [=====] - 6s 3s/step - loss: 1.0168 - acc: 0.6771
Epoch 78/100
3/3 [=====] - 1s 316ms/step - loss: 1.0140 - acc: 0.7812
Epoch 79/100
3/3 [=====] - 1s 303ms/step - loss: 0.8553 - acc: 0.7708
Epoch 80/100
3/3 [=====] - 2s 1s/step - loss: 0.8026 - acc: 0.7708
Epoch 81/100
3/3 [=====] - 3s 1s/step - loss: 0.7121 - acc: 0.7812
Epoch 82/100
3/3 [=====] - 2s 860ms/step - loss: 0.8729 - acc: 0.7188
Epoch 83/100
3/3 [=====] - 2s 579ms/step - loss: 0.6625 - acc: 0.8333
Epoch 84/100
```

```
-r--. . . --  
3/3 [=====] - 2s 861ms/step - loss: 0.6713 - acc: 0.8125  
Epoch 85/100  
3/3 [=====] - 2s 553ms/step - loss: 0.7657 - acc: 0.8125  
Epoch 86/100  
3/3 [=====] - 1s 301ms/step - loss: 0.5957 - acc: 0.8021  
Epoch 87/100  
3/3 [=====] - 3s 1s/step - loss: 0.7824 - acc: 0.7708 -  
Epoch 88/100  
3/3 [=====] - 2s 1s/step - loss: 0.6365 - acc: 0.7917 -  
Epoch 89/100  
3/3 [=====] - 2s 898ms/step - loss: 0.4714 - acc: 0.8646  
Epoch 90/100  
3/3 [=====] - 1s 583ms/step - loss: 0.7146 - acc: 0.7812  
Epoch 91/100  
3/3 [=====] - 1s 593ms/step - loss: 0.6439 - acc: 0.8021  
Epoch 92/100  
3/3 [=====] - 1s 277ms/step - loss: 0.7266 - acc: 0.8125  
Epoch 93/100  
3/3 [=====] - 2s 830ms/step - loss: 0.7496 - acc: 0.7708  
Epoch 94/100  
3/3 [=====] - 1s 293ms/step - loss: 0.7284 - acc: 0.8125  
Epoch 95/100  
3/3 [=====] - 1s 553ms/step - loss: 0.5647 - acc: 0.8646  
Epoch 96/100  
3/3 [=====] - 2s 811ms/step - loss: 0.6287 - acc: 0.8646  
Epoch 97/100  
3/3 [=====] - 2s 818ms/step - loss: 0.3517 - acc: 0.9167  
Epoch 98/100  
3/3 [=====] - 1s 294ms/step - loss: 0.4469 - acc: 0.8750  
Epoch 99/100  
3/3 [=====] - 2s 566ms/step - loss: 0.4459 - acc: 0.9062  
Epoch 100/100  
3/3 [=====] - 1s 304ms/step - loss: 0.5602 - acc: 0.8646
```

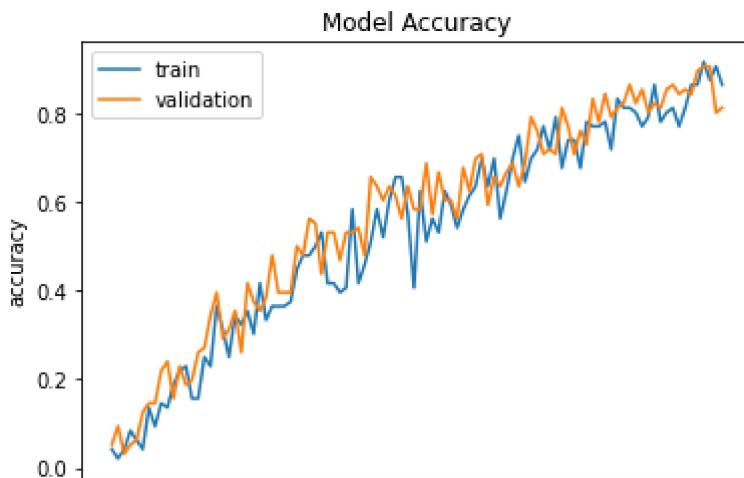


```
model.save('meo.h5')  
from tensorflow.keras.models import load_model  
model=load_model('meo.h5')  
  
score = model.evaluate(validation_generator,verbose=0)  
print('Sai số kiểm tra là: ',score[0])  
print('Độ chính xác kiểm tra là: ',score[1])  
  
Sai số kiểm tra là:  0.489311546087265  
Độ chính xác kiểm tra là:  0.867241382598877
```

```
plt.plot(history.history['acc'])  
plt.plot(history.history['val_acc'])  
plt.title('Model Accuracy')  
plt.ylabel('accuracy')  
plt.xlabel('epoch')  
plt.legend(['train','validation'], loc='upper-left')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: MatplotlibDeprecatio
    best
    upper right
    upper left
    lower left
    lower right
    right
    center left
    center right
    lower center
    upper center
    center
This will raise an exception in 3.3.
```

```
<matplotlib.legend.Legend at 0x7f836e2f9bd0>
```



```
from google.colab import files
from keras.preprocessing import image
%matplotlib inline
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
```

```
label=['Abyssinian','Bengal','Burmilla','Garfield','Gizmo','Grumpy','Havana Brown','Luna',
'Marie Coon','Meo Anh Long Dai','Meo Anh Long Ngan','Meo Ba Tu','Meo Rung Na-uy','M
'Meo Xiem','Mia','Molly','Munchkin','Nala','Ocicat','Oriental','Princess','Ragdoll'
'Scottish Fold','Simba','Singapura','Somali','Sophie','Sphynx']
```

```
uploaded=files.upload()
for fn in uploaded.keys():
    #predicting images
    path='/content/' + fn
    #In ảnh đọc được
    plt.imshow(mpimg.imread(path))
    img=image.load_img(path,target_size=(200,200))
    x=image.img_to_array(img)
    x=np.expand_dims(x,axis=0)
    images=np.vstack([x])
    y_predict = model.predict(images,batch_size=10)
    print(y_predict)
    print('Giá trị dự đoán: ', label[np.argmax(y_predict)])
    plt.show()
```

Chọn tệp 29 tệp

- **Abyssinian.jpg**(image/jpeg) - 7555 bytes, last modified: 5/6/2022 - 100% done
  - **Bengal.jpg**(image/jpeg) - 10469 bytes, last modified: 5/6/2022 - 100% done
  - **Burmilla.jpg**(image/jpeg) - 4718 bytes, last modified: 5/6/2022 - 100% done
  - **Garfield.jpg**(image/jpeg) - 6052 bytes, last modified: 4/6/2022 - 100% done
  - **Gizmo.jpg**(image/jpeg) - 7540 bytes, last modified: 4/6/2022 - 100% done
  - **Grumpy.jpg**(image/jpeg) - 7581 bytes, last modified: 4/6/2022 - 100% done
  - **Havana Brown.jpg**(image/jpeg) - 6023 bytes, last modified: 5/6/2022 - 100% done
  - **Luna.jpg**(image/jpeg) - 9292 bytes, last modified: 4/6/2022 - 100% done
  - **Manie Coon.jpg**(image/jpeg) - 13797 bytes, last modified: 5/6/2022 - 100% done
  - **Meo Anh Long Dai.jpg**(image/jpeg) - 6216 bytes, last modified: 4/6/2022 - 100% done
  - **Meo Anh Long Ngan.jpg**(image/jpeg) - 5273 bytes, last modified: 4/6/2022 - 100% done
  - **Meo Ba Tu.jpg**(image/jpeg) - 4619 bytes, last modified: 4/6/2022 - 100% done
  - **Meo Rung Na-uy.jpg**(image/jpeg) - 4643 bytes, last modified: 4/6/2022 - 100% done
  - **Meo Tam The.jpg**(image/jpeg) - 6312 bytes, last modified: 4/6/2022 - 100% done
  - **Meo Xiem.jpg**(image/jpeg) - 48826 bytes, last modified: 4/6/2022 - 100% done
  - **Mia.jpg**(image/jpeg) - 6325 bytes, last modified: 4/6/2022 - 100% done
  - **Molly.jpg**(image/jpeg) - 9465 bytes, last modified: 4/6/2022 - 100% done
  - **Munchkin.jpg**(image/jpeg) - 4184 bytes, last modified: 4/6/2022 - 100% done
  - **Nala.jpg**(image/jpeg) - 7942 bytes, last modified: 4/6/2022 - 100% done
  - **Ocicat.jpg**(image/jpeg) - 3977 bytes, last modified: 5/6/2022 - 100% done
  - **Oriental.jpg**(image/jpeg) - 6641 bytes, last modified: 5/6/2022 - 100% done
  - **Princess.jpg**(image/jpeg) - 4152 bytes, last modified: 4/6/2022 - 100% done
  - **Ragdoll.jpg**(image/jpeg) - 5415 bytes, last modified: 5/6/2022 - 100% done
  - **Scottist Fold.jpg**(image/jpeg) - 5866 bytes, last modified: 5/6/2022 - 100% done
  - **Simba.jpg**(image/jpeg) - 5736 bytes, last modified: 1/6/2022 - 100% done
  - **Singapura.jpg**(image/jpeg) - 5036 bytes, last modified: 5/6/2022 - 100% done
  - **Somali.jpg**(image/jpeg) - 5126 bytes, last modified: 5/6/2022 - 100% done
  - **Sophie.jpg**(image/jpeg) - 8883 bytes, last modified: 4/6/2022 - 100% done
  - **Sphynx.jpg**(image/jpeg) - 7692 bytes, last modified: 5/6/2022 - 100% done

Saving Abyssinian.jpg to Abyssinian.jpg

Saving Bengal.jpg to Bengal.jpg

Saving Burmilla.jpg to Burmilla.jpg

Saving Garfield.jpg to Garfield.jpg

Saving Gizmo.jpg to Gizmo.jpg

Saving Grumpy.jpg to Grumpy.jpg

Saving Havana Brown.jpg to Havana

Saving Luna.jpg to Luna.jpg

Saving Manie Coon.jpg to Ma

Saving Meo Anh Long Dai.jpg to Meo Anh

Saving Meo Anh Long Ngan.jpg to Meo Anh Long Ngan.jpg

Saving Meo Ba Tu.jpg to Meo Ba Tu.jpg

Saving Meo Rung Na-uy.jpg to Meo Rung Na-uy.jpg

Saving Meo Tam The.jpg to Meo Tam The.jpg

Saving Meo Xiem.jpg to Meo Xiem.jpg

Saving Mia.jpg to Mia.jpg

Saving Molly.jpg to Molly.jpg

Saving Munchkin.jpg to Munchkin.jpg

Saving Nala.jpg to Nala.jpg

Saving Ocicat.jpg to Ocicat.jpg

Saving Oriental.jpg to Orient

Saving Princess.jpg to Princess.jpg

Saving Ragdoll.jpg to Ragdoll.jpg

Saving Scottist Fold.jpg to Scott

Saving Simba.jpg to Simba.jpg

Saving Singapura.jpg to Singa

Saving Somali.jpg to Somali.jpg

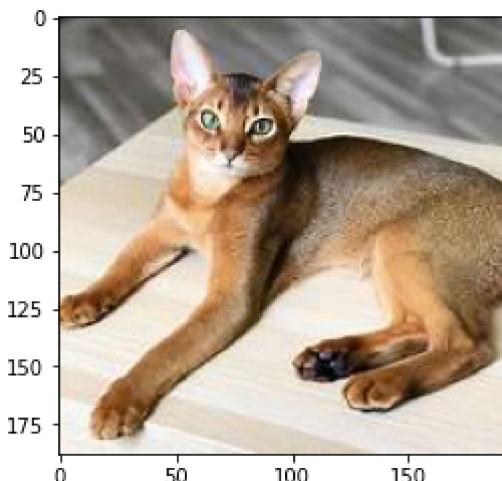
Saving Sophie.jpg to Sophie.jpg

Saving Sphynx.jpg to Sphynx.jpg

111 A. A. A. A. A. A. A. A. A.

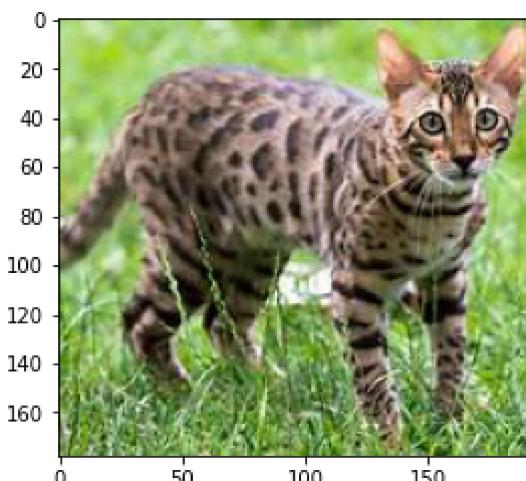
[0. 0. 0. 0. 0.]

Giá trị dự đoán: Abyssinian



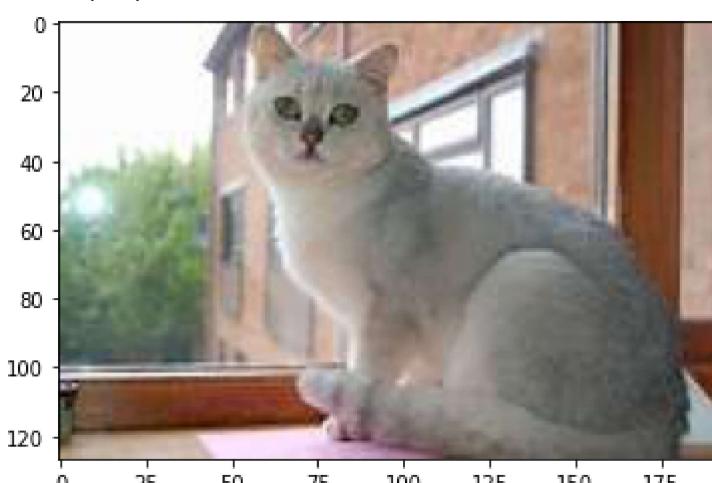
[[0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  
0. 0. 0. 0. 0.]]

Giá trị dự đoán: Bengal



[[0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  
0. 0. 0. 0. 0.]]

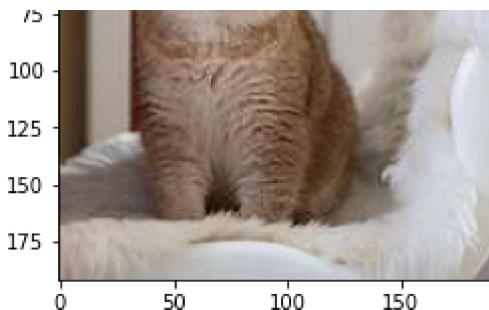
Giá trị dự đoán: Burmilla



[[0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  
0. 0. 0. 0. 0.]]

Giá trị dự đoán: Garfield





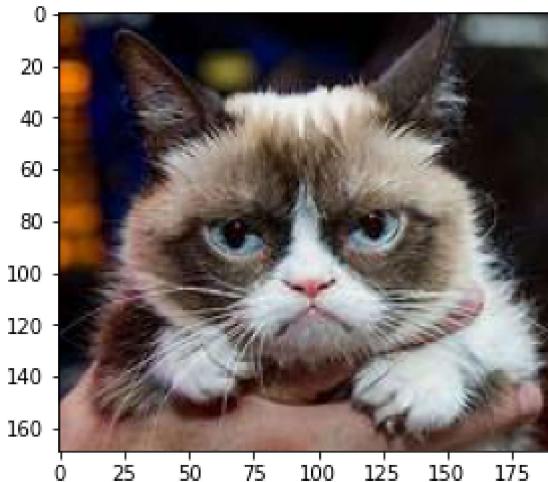
[[0. 0. 0. 0. 0. 1. 0.]]

Giá trị dự đoán: Gizmo



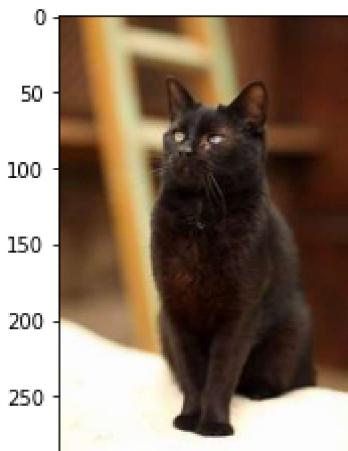
[[0. 0. 0. 0. 0. 1. 0.]]

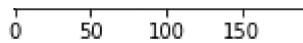
Giá trị dự đoán: Grumpy



[[0. 0. 0. 0. 0. 0. 1. 0.]]

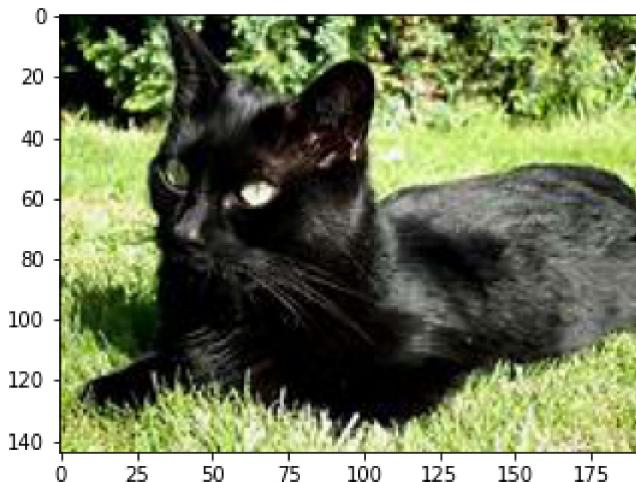
Giá trị dự đoán: Havana Brown





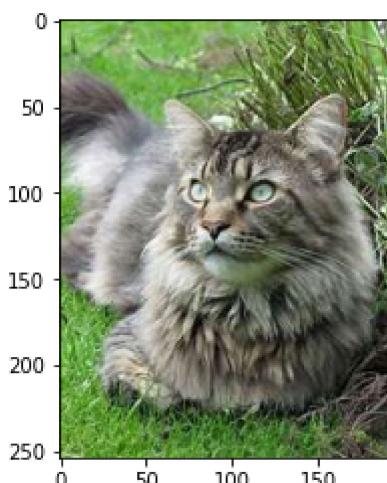
```
[[0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.000000e+00  
0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 9.973972e-25  
0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  
0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00]]
```

Giá trị dự đoán: Gizmo



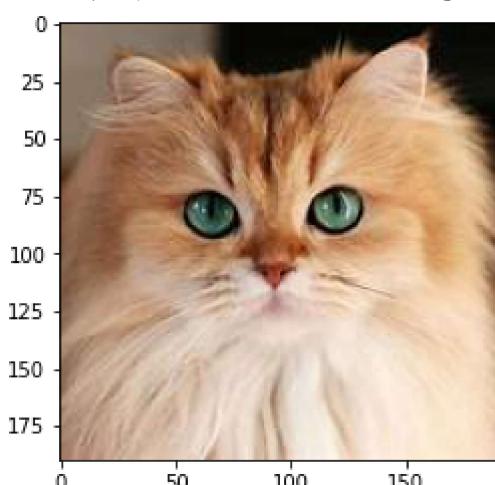
```
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  
0. 0. 0. 0. 0.]]
```

Giá trị dự đoán: Manie Coon



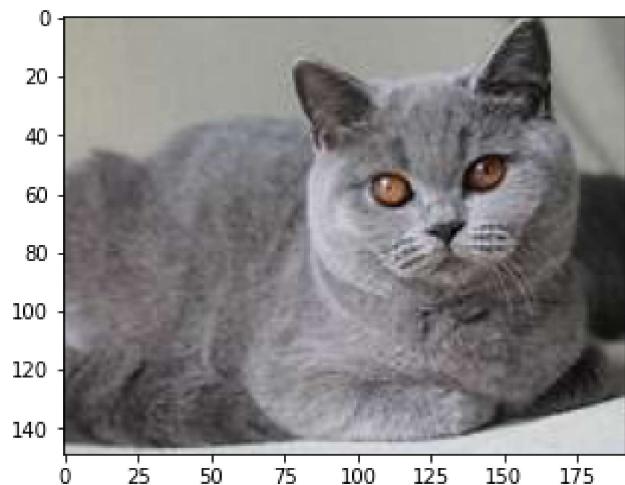
```
[[0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  
0. 0. 0. 0. 0.]]
```

Giá trị dự đoán: Meo Anh Long Dai



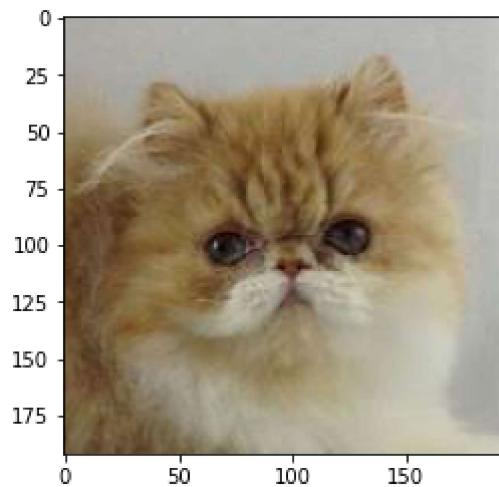
```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.  
0. 0. 0. 0. 0.]]
```

Giá trị dự đoán: Meo Anh Long Ngan



`[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]`

Giá trị dự đoán: Meo Ba Tu



`[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]`

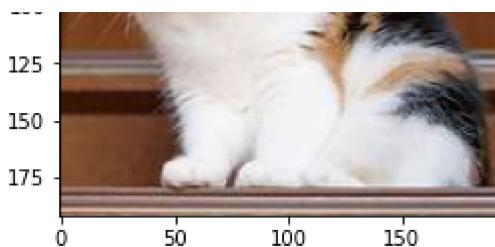
Giá trị dự đoán: Meo Rung Na-uy



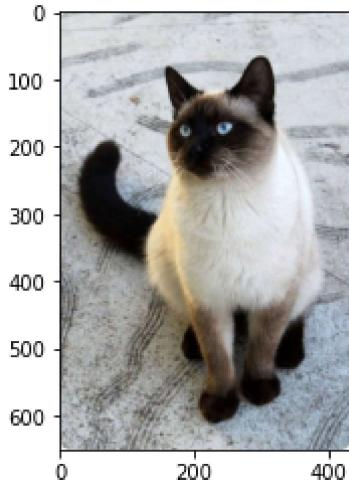
`[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]`

Giá trị dự đoán: Meo Tam The

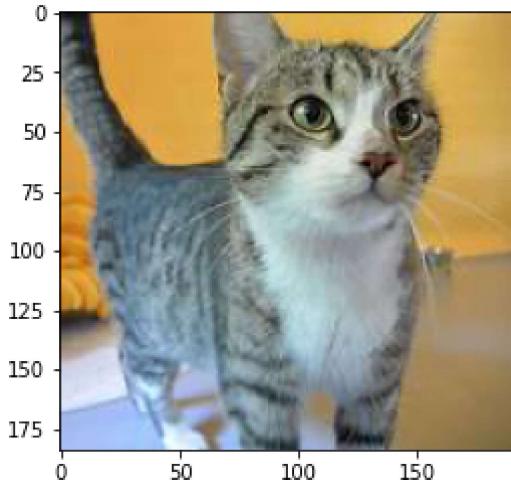




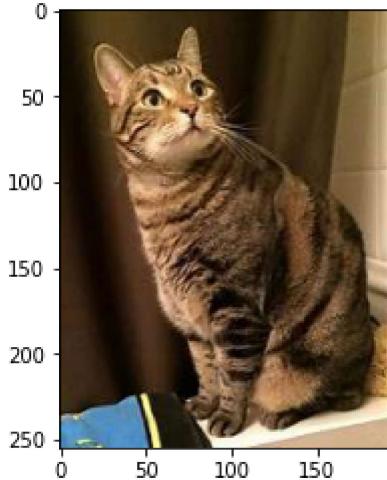
Giá trị dự đoán: Meo Xiem



Giá trị dự đoán: Munchkin

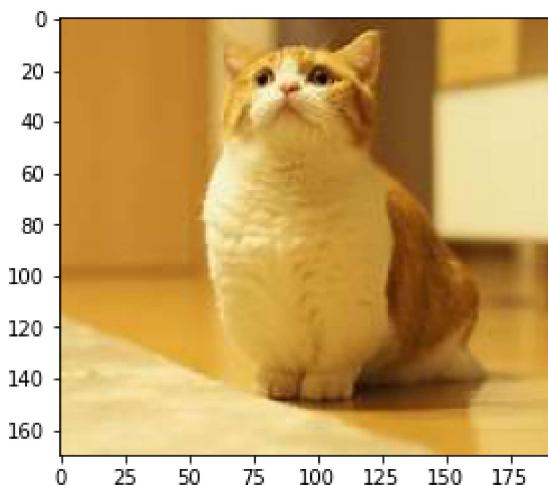


Giá trị dự đoán: Bengal



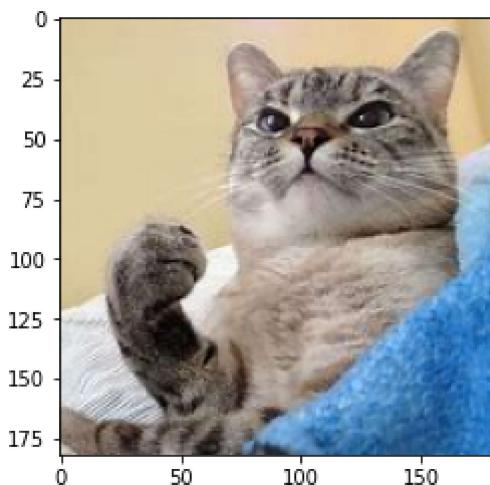
[0. 0.]

Giá trị dự đoán: Munchkin



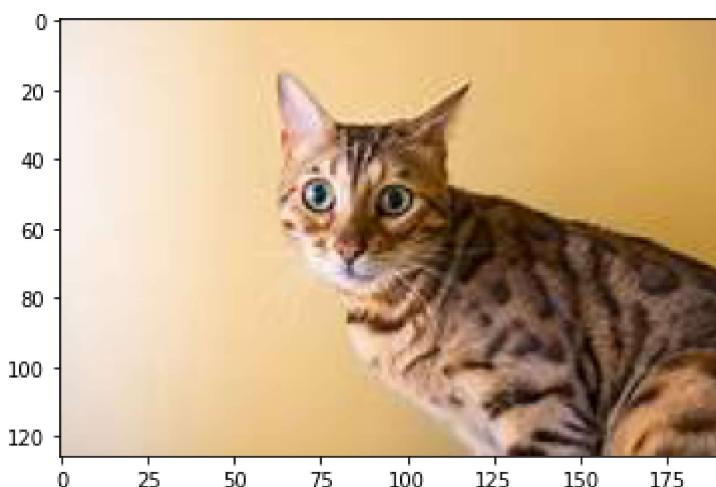
[0. 0.]

Giá trị dự đoán: Nala



[0. 0.]

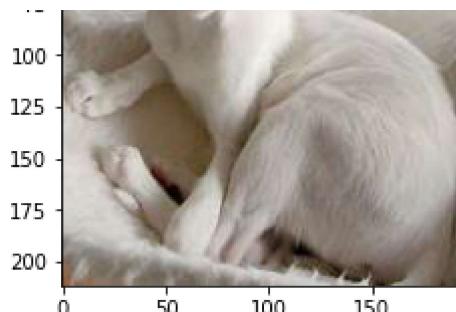
Giá trị dự đoán: Ocicat



[0. 0.]

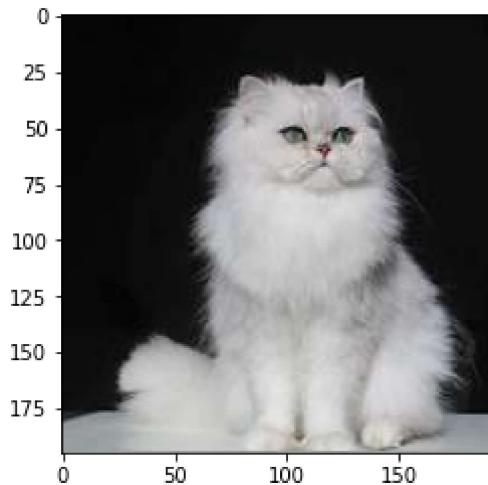
Giá trị dự đoán: Oriental





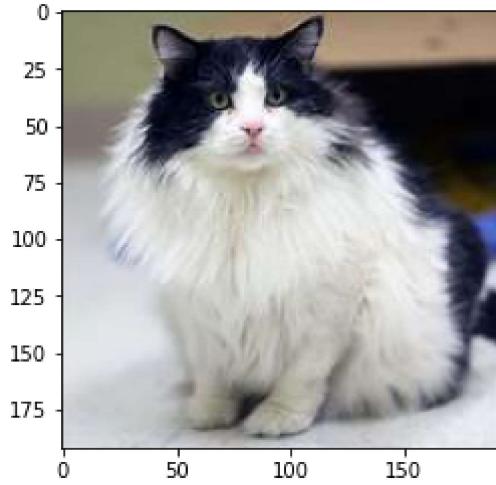
```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0.
 0. 0. 0. 0. 0.]]
```

Giá trị dự đoán: Princess



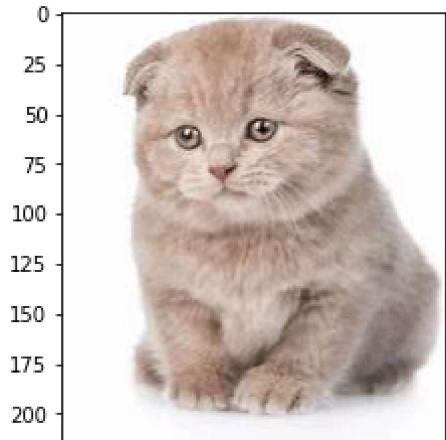
```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.
 0. 0. 0. 0. 0.]]
```

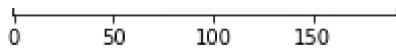
Giá trị dự đoán: Ragdoll



```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.
 0. 0. 0. 0. 0.]]
```

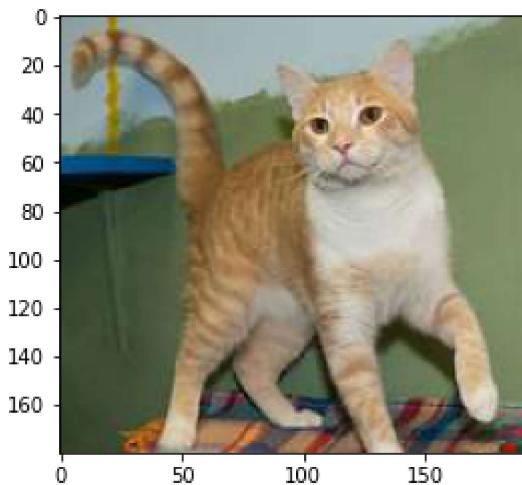
Giá trị dự đoán: Scottist Fold





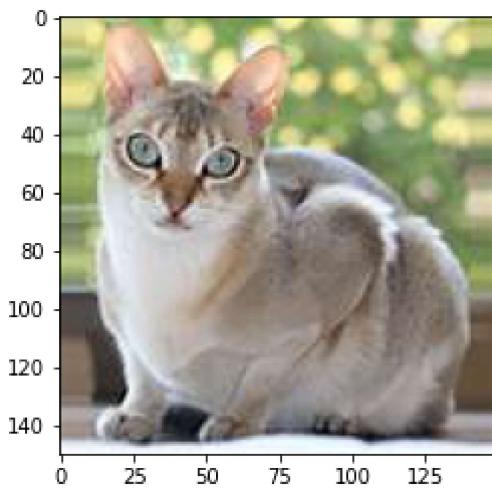
```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  
1. 0. 0. 0. 0.]]
```

Giá trị dự đoán: Simba



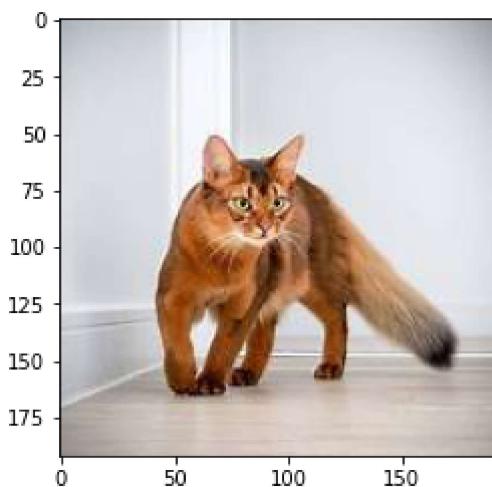
```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  
0. 1. 0. 0. 0.]]
```

Giá trị dự đoán: Singapura



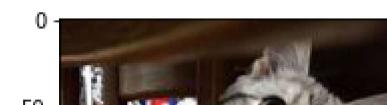
```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  
0. 0. 1. 0. 0.]]
```

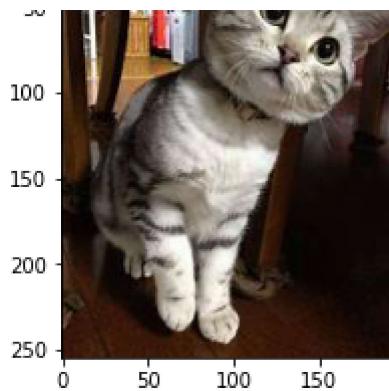
Giá trị dự đoán: Somali



```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  
0. 0. 0. 1. 0.]]
```

Giá trị dự đoán: Sophie





```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  
0. 0. 0. 0. 1.]]
```

Giá trị dự đoán: Sphynx

