

Output prediction for dog prediction:

Cat_or_dog_image_classifier.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

CNN-cat-and-dog

dog-cat-full-dataset

data

test

cats

dogs

dog.0.jpg

dog.10004.jpg

dog.10014.jpg

dog.10019.jpg

dog.10025.jpg

dog.10026.jpg

dog.10027.jpg

dog.10028.jpg

dog.10029.jpg

dog.10033.jpg

dog.1004.jpg

dog.10040.jpg

dog.10046.jpg

dog.10048.jpg

dog.10049.jpg

+ Code + Text

Part 4 - making a single prediction

0s

```
import numpy as np
from keras.preprocessing import image

test_image = tf.keras.utils.load_img('/content/dog-cat-full-dataset/data/test/dogs/dog.10028.jpg',
                                     target_size = (64, 64))
#test_image = tf.keras.utils.load_img(test_image)
test_image = np.expand_dims(test_image, axis = 0)
result = cnn.predict(test_image)
training_set.class_indices
if result[0][0] == 1:
    prediction = 'dog'
else:
    prediction = 'cat'
```

1/1 [=====] - 0s 20ms/step

0s

[21] print(prediction)

dog

2s

```
%shell
jupyter nbconvert --to html /content/Cat_or_dog_image_classifier.ipynb
```

[NbConvertApp] Converting notebook /content/Cat_or_dog_image_classifier.ipynb to html
[NbConvertApp] Writing 622211 bytes to /content/cat_or_dog_image_classifier.html

1s completed at 4:30 PM

Output for cat prediction:

Cat_or_dog_image_classifier.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

...

CNN-cat-and-dog

dog-cat-full-dataset

data

test

cats

cat.0.jpg

cat.10003.jpg

cat.10004.jpg

cat.10008.jpg

cat.10013.jpg

cat.10014.jpg

cat.1002.jpg

cat.10028.jpg

cat.10031.jpg

cat.10034.jpg

cat.10047.jpg

cat.10050.jpg

cat.10060.jpg

cat.10061.jpg

cat.10066.jpg

cat.10079.jpg

Disk

50.03 GB available

Part 4 - Making a single prediction

[24] import numpy as np

from keras.preprocessing import image

test_image = tf.keras.utils.load_img('/content/dog-cat-full-dataset/data/test/cats/cat.10028.jpg',

target_size = (64, 64))

#test_image = tf.keras.utils.load_img(test_image)

test_image = np.expand_dims(test_image, axis = 0)

result = cnn.predict(test_image)

training_set.class_indices

if result[0][0] == 1:

prediction = 'dog'

else:

prediction = 'cat'

1/1 [=====] - 0s 19ms/step

print(prediction)

cat

[23] %shell

jupyter nbconvert --to html /content/Cat_or_dog_image_classifier.ipynb

[NbConvertApp] Converting notebook /content/Cat_or_dog_image_classifier.ipynb to html

[NbConvertApp] Writing 622211 bytes to /content/Cat_or_dog_image_classifier.html

0s completed at 4:48 PM