

```
In [1]: from keras import backend as K
from keras.applications import imagenet_utils
import numpy as np
K.clear_session()
K.set_learning_phase(False)

from keras.applications import mobilenet as network
inputShape = (224, 224, 3)

print("[INFO] loading {}".format(network.__name__))
model = network.MobileNet(alpha=1.0,
                           weights="imagenet",
                           input_shape=inputShape)
```

```
/usr/local/lib/python2.7/dist-packages/h5py/__init__.py:36: FutureWarning:
Conversion of the second argument of issubdtype from `float` to `np.floati
ng` is deprecated. In future, it will be treated as `np.float64 == np.dtyp
e(float).type`.
```

```
from ._conv import register_converters as _register_converters
Using TensorFlow backend.
```

```
[INFO] loading keras.applications.mobilenet...
```

```

In [2]: from keras.preprocessing.image import img_to_array
        from keras.preprocessing.image import load_img
        import matplotlib.pyplot as plt
        %matplotlib inline

        # This is the image to be classified with above network
        image_file = './Pet_Dataset/production/Pet_00001.jpg'

        print("[INFO] loading and pre-processing image...")
        raw_image = load_img(image_file)
        plt.imshow(raw_image)
        plt.show()

        image = load_img(image_file, target_size=inputShape)
        image = img_to_array(image)
        image = np.expand_dims(image, axis=0)
        image = network.preprocess_input(image)

        print("[INFO] classifying image {} \n with {}".format(image_file,
                                                                network.__name__))

        preds = model.predict(image)
        P = imagenet_utils.decode_predictions(preds)

        results=[{'rank': i+1,
                  'score': prob*100.0,
                  'class': np.where(preds == prob)[1][0],
                  'id': imagenetID, 'label': label}
                 for (i, (imagenetID, label, prob)) in enumerate(P[0])]

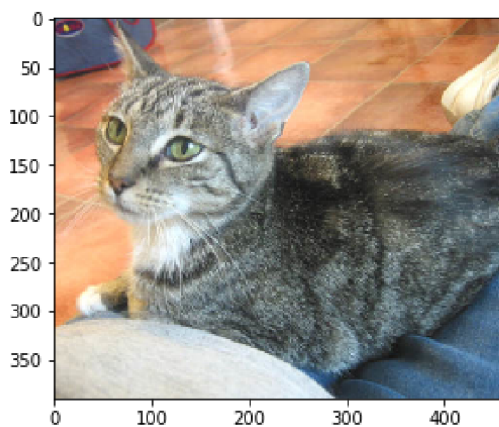
        print '-' * 60
        print "{0:>5} | {1:>6} | {2:>6} | {3:<12}| {4:<32}".format("Rank",
                                                                "Score",
                                                                "Class",
                                                                "ImageNet ID",
                                                                "Label")

        print '-' * 60
        for r in results:
            print "{rank:>5} | {score:>6.2f}% | {class:>6} | {id:<12}| {label:<48}".
                  format(**r)

        print '-' * 60

```

[INFO] loading and pre-processing image...



[INFO] classifying image ./Pet_Dataset/production/Pet_00001.jpg
with keras.applications.mobilenet...

Rank	Score	Class	ImageNet ID	Label
1	84.88%	281	n02123045	tabby
2	9.34%	282	n02123159	tiger_cat
3	5.23%	285	n02124075	Egyptian_cat
4	0.29%	287	n02127052	lynx
5	0.03%	330	n02325366	wood_rabbit