1.go to https://github.com/miaow1988/SqueezeNet_v1.2 and download the 'symbol.json' and '.params' files (there is not a 'synset.txt' file! so don't use these lines, Hint: just comment these lines).

Install MXNet v1.5 (hint: create a new conda environmet with python 3, pip install mxnet==1.5.1) and follow the same steps of the lecture (part: Using pre-trained models as feature extractors). Find the flatten output layer and create a feature extractor (hint: It should be a numpy array of 1000 elements).

Download the dogs versus cats training folder from https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data (Remember the number of images is 12500 for each class).

Extract the array of features for different number of images (N: 10, 100, 500, 1000, also 5000 and 12500) and for each value train your favorite binary classifier (only one!!!) using GridSearch to optimize some hyperparameters. Consider to use https://notebooks.csc.fi if you have computational limitations.

Report the accuracy for each value of N and the computational time during the training step.

In [8]:
```
!pip install mxnet==1.5.1
```

```
Collecting mxnet==1.5.1
  Downloading mxnet-1.5.1-py2.py3-none-manylinux1_x86_64.whl (23.1 MB)
     |████████████████████████████████| 23.1 MB 42.9 MB/s eta 0:00:01  |████
████████                        | 8.7 MB 1.9 MB/s eta 0:00:081.9 MB/s eta 0:00:07/s
eta 0:00:06��████████████████                | 13.7 MB 1.9 MB/s eta 0:00:060:04.
5 MB 1.9 MB/s eta 0:00:03��██████████    | 20.4 MB 1.9 MB/s eta 0:00:02�██ | 2
2.8 MB 1.9 MB/s eta 0:00:01
Requirement already satisfied: numpy<2.0.0,>1.16.0 in /opt/conda/lib/python3.7
/site-packages (from mxnet==1.5.1) (1.18.2)
Requirement already satisfied: requests<3,>=2.20.0 in /opt/conda/lib/python3.7
/site-packages (from mxnet==1.5.1) (2.23.0)
Collecting graphviz<0.9.0,>=0.8.1
  Downloading graphviz-0.8.4-py2.py3-none-any.whl (16 kB)
Requirement already satisfied: idna<3,>=2.5 in /opt/conda/lib/python3.7/site-p
ackages (from requests<3,>=2.20.0->mxnet==1.5.1) (2.9)
Requirement already satisfied: chardet<4,>=3.0.2 in /opt/conda/lib/python3.7/s
ite-packages (from requests<3,>=2.20.0->mxnet==1.5.1) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.7/
site-packages (from requests<3,>=2.20.0->mxnet==1.5.1) (2019.11.28)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /opt
/conda/lib/python3.7/site-packages (from requests<3,>=2.20.0->mxnet==1.5.1) (1
.25.7)
Installing collected packages: graphviz, mxnet
  Attempting uninstall: graphviz
    Found existing installation: graphviz 0.13.2
    Uninstalling graphviz-0.13.2:
      Successfully uninstalled graphviz-0.13.2
Successfully installed graphviz-0.8.4 mxnet-1.5.1
```

In [5]:
```
import mxnet as mx

path = 'https://github.com/miaow1988/SqueezeNet_v1.2/'
[
    mx.test_utils.download(path +'/blob/master/model-symbol.json'),
    mx.test_utils.download(path +'blob/master/model-0000.params'),
]
```

Out[5]:
```
['model-symbol.json', 'model-0000.params']
```

In [8]:
```
!wget https://github.com/miaow1988/SqueezeNet_v1.2/raw/master/model-symbol.js
!wget https://github.com/miaow1988/SqueezeNet_v1.2/raw/master/model-0000.para
```

```
--2020-11-07 10:23:47--  https://github.com/miaow1988/SqueezeNet_v1.2/raw/mast
er/model-symbol.json
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/miaow1988/SqueezeNet_v1.2/master/m
odel-symbol.json [following]
--2020-11-07 10:23:47--  https://raw.githubusercontent.com/miaow1988/SqueezeNe
t_v1.2/master/model-symbol.json
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.84.
133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.84
.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24187 (24K) [text/plain]
Saving to: 'model-symbol.json'

model-symbol.json   100%[===================>]  23.62K  --.-KB/s    in 0.01s

2020-11-07 10:23:47 (1.57 MB/s) - 'model-symbol.json' saved [24187/24187]

--2020-11-07 10:23:48--  https://github.com/miaow1988/SqueezeNet_v1.2/raw/mast
er/model-0000.params
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/miaow1988/SqueezeNet_v1.2/master/m
odel-0000.params [following]
--2020-11-07 10:23:49--  https://raw.githubusercontent.com/miaow1988/SqueezeNe
t_v1.2/master/model-0000.params
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.84.
133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.84
.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4945034 (4.7M) [application/octet-stream]
Saving to: 'model-0000.params'

model-0000.params   100%[===================>]   4.72M  12.8MB/s    in 0.4s

2020-11-07 10:23:50 (12.8 MB/s) - 'model-0000.params' saved [4945034/4945034]
```

In [10]:

```
! pip install opencv-python-headless
! pip install --user kaggle --upgrade
```

```
Collecting opencv-python-headless
  Downloading opencv_python_headless-4.4.0.46-cp37-cp37m-manylinux2014_x86_64.
whl (36.7 MB)
     |████████████████████████████████| 36.7 MB 42.8 MB/s eta 0:00:01MB 2.1 M
B/s eta 0:00:13                    | 13.6 MB 2.1 MB/s eta 0:00:12ﾀﾀ████████
| 17.5 MB 2.1 MB/s eta 0:00:10████████████████                | 18.9 MB 2.1 MB
```

```
/s eta 0:00:09████                | 21.1 MB 2.1 MB/s eta 0:00:08▒████████
| 23.7 MB 42.8 MB/s eta 0:00:01101███████████       | 28.9 MB 42.8 MB/s eta 0:
00:01 | 30.8 MB 42.8 MB/s eta 0:00:01|████████████████████████████    | 33.1
MB 42.8 MB/s eta 0:00:01
Requirement already satisfied: numpy>=1.14.5 in /opt/conda/lib/python3.7/site-
packages (from opencv-python-headless) (1.18.2)
Installing collected packages: opencv-python-headless
Successfully installed opencv-python-headless-4.4.0.46
Collecting kaggle
  Downloading kaggle-1.5.9.tar.gz (58 kB)
     |████████████████████████████████| 58 kB 1.5 MB/s eta 0:00:011
Requirement already satisfied, skipping upgrade: six>=1.10 in /opt/conda/lib/p
ython3.7/site-packages (from kaggle) (1.14.0)
Requirement already satisfied, skipping upgrade: certifi in /opt/conda/lib/pyt
hon3.7/site-packages (from kaggle) (2019.11.28)
Requirement already satisfied, skipping upgrade: python-dateutil in /opt/conda
/lib/python3.7/site-packages (from kaggle) (2.8.1)
Requirement already satisfied, skipping upgrade: requests in /opt/conda/lib/py
thon3.7/site-packages (from kaggle) (2.23.0)
Requirement already satisfied, skipping upgrade: tqdm in /opt/conda/lib/python
3.7/site-packages (from kaggle) (4.43.0)
Collecting python-slugify
  Downloading python-slugify-4.0.1.tar.gz (11 kB)
Collecting slugify
  Downloading slugify-0.0.1.tar.gz (1.2 kB)
Requirement already satisfied, skipping upgrade: urllib3 in /opt/conda/lib/pyt
hon3.7/site-packages (from kaggle) (1.25.7)
Requirement already satisfied, skipping upgrade: idna<3,>=2.5 in /opt/conda/li
b/python3.7/site-packages (from requests->kaggle) (2.9)
Requirement already satisfied, skipping upgrade: chardet<4,>=3.0.2 in /opt/con
da/lib/python3.7/site-packages (from requests->kaggle) (3.0.4)
Collecting text-unidecode>=1.3
  Downloading text_unidecode-1.3-py2.py3-none-any.whl (78 kB)
     |████████████████████████████████| 78 kB 3.4 MB/s eta 0:00:011
Building wheels for collected packages: kaggle, python-slugify, slugify
  Building wheel for kaggle (setup.py) ... done
  Created wheel for kaggle: filename=kaggle-1.5.9-py3-none-any.whl size=73265
sha256=d2a32e42c1c9ad13d7feeaae1eac66f910874e3ef556195531f1285a0431c762
  Stored in directory: /home/jovyan/.cache/pip/wheels/09/25/76/1bbe8ad0c423e86
55942b6d3c781f58e0ea2791bf8ee8985b3
  Building wheel for python-slugify (setup.py) ... done
  Created wheel for python-slugify: filename=python_slugify-4.0.1-py2.py3-none
-any.whl size=6767 sha256=28a60d02dcab1924182c050daff80e8dba8e396d89a7d5361aec
9dd8f76d7fa5
  Stored in directory: /home/jovyan/.cache/pip/wheels/48/1b/6f/5c1cfab22eacbe0
095fc619786da6571b55253653c71324b5c
  Building wheel for slugify (setup.py) ... done
  Created wheel for slugify: filename=slugify-0.0.1-py3-none-any.whl size=1908
sha256=33bd827d718d4f5f03d9bf292f89e4e5a74fbaba47dc0df4fa6efb32f1503135
  Stored in directory: /home/jovyan/.cache/pip/wheels/d4/7b/0d/bd65011a1b44284
3bb4043e396f727ab0f1e76050355b9156a
Successfully built kaggle python-slugify slugify
Installing collected packages: text-unidecode, python-slugify, slugify, kaggle
```

```
  WARNING: The script slugify is installed in '/home/jovyan/.local/bin' which
is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this wa
rning, use --no-warn-script-location.
  WARNING: The script kaggle is installed in '/home/jovyan/.local/bin' which i
s not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this wa
rning, use --no-warn-script-location.
Successfully installed kaggle-1.5.9 python-slugify-4.0.1 slugify-0.0.1 text-un
idecode-1.3
```

In [15]:
```
# if "opencv-python" not in pkgs:
#     !pip install opencv-python
# if "opencv-contrib-python" not in pkgs:
#     !pip install opencv-contrib-python
```

Upload your https://github.com/Kaggle/kaggle-api#api-credentials

In [11]:
```
mkdir ~/.kaggle
```

```
mkdir: cannot create directory ‘/home/jovyan/.kaggle’: File exists
```

In [12]:
```
! mv kaggle.json .kaggle/kaggle.json
```

In [13]:
```
! chmod 600 ~/.kaggle/kaggle.json
```

In [17]:
```
! ls -l ~/.kaggle
```

```
total 4
-rw-------. 1 jovyan root 63 Nov  7 10:19 kaggle.json
```

In [20]:
```
! ls -l ~/.local/bin
```

```
total 8
-rwxr-xr-x. 1 jovyan root 217 Nov  7 10:24 kaggle
-rwxr-xr-x. 1 jovyan root 333 Nov  7 10:24 slugify
```

In [19]:
```
!pip show kaggle
```

```
Name: kaggle
Version: 1.5.9
Summary: Kaggle API
Home-page: https://github.com/Kaggle/kaggle-api
Author: Kaggle
Author-email: support@kaggle.com
License: Apache 2.0
Location: /home/jovyan/.local/lib/python3.7/site-packages
Requires: tqdm, certifi, requests, urllib3, six, slugify, python-slugify, pyth
on-dateutil
Required-by:
```

In [21]:
```
! find / -name kaggle 2>/dev/null
```

```
/home/jovyan/.local/lib/python3.7/site-packages/kaggle
/home/jovyan/.local/bin/kaggle
```

In [24]:
```
# !kaggle competitions download -c dogs-vs-cats-redux-kernels-edition
```

In [1]:
```
! unzip dogs-vs-cats-redux-kernels-edition.zip
```

```
Archive:  dogs-vs-cats-redux-kernels-edition.zip
   creating: dogs-vs-cats-redux-kernels-edition/
  inflating: dogs-vs-cats-redux-kernels-edition/test.zip
   creating: __MACOSX/
   creating: __MACOSX/dogs-vs-cats-redux-kernels-edition/
  inflating: __MACOSX/dogs-vs-cats-redux-kernels-edition/._test.zip
  inflating: dogs-vs-cats-redux-kernels-edition/train.zip
  inflating: __MACOSX/dogs-vs-cats-redux-kernels-edition/._train.zip
  inflating: dogs-vs-cats-redux-kernels-edition/sample_submission.csv
  inflating: __MACOSX/dogs-vs-cats-redux-kernels-edition/._sample_submission.c
sv
  inflating: __MACOSX/._dogs-vs-cats-redux-kernels-edition
```

In [ ]:
```
! unzip ~/dogs-vs-cats-redux-kernels-edition/train.zip
```

In [9]:
```
import mxnet as mx
import numpy as np
from mxnet import nd,autograd
from mxnet import gluon

%matplotlib inline
import matplotlib.pyplot as plt
import cv2

# define a simple data batch
from collections import namedtuple
Batch = namedtuple('Batch', ['data'])
```

In [11]:
```
context=mx.cpu()
```

In [13]:
```
mod = mx.mod.Module(symbol=sym, context=mx.cpu(), label_names=None)
mod.bind(for_training=False, data_shapes=[('data', (1,3,224,224))],
         label_shapes=mod._label_shapes)
mod.set_params(arg_params, aux_params, allow_missing=True)
# with open('pretrained/synset.txt', 'r') as f:
#     labels = [l.rstrip() for l in f]
```

In [100…
```
sym, arg_params, aux_params = mx.model.load_checkpoint('model', 0)
```

In [17]:
```
def get_image(url, show=False):
    if url.startswith('http'):
        # download and show the image
        fname = mx.test_utils.download(url)
    else:
        fname = url
    img = cv2.cvtColor(cv2.imread(fname), cv2.COLOR_BGR2RGB)
    if img is None:
         return None
    if show:
         plt.imshow(img)
         plt.axis('off')
    # convert into format (batch, RGB, width, height)
    img = cv2.resize(img, (224, 224))
    img = np.swapaxes(img, 0, 2)
    img = np.swapaxes(img, 1, 2)
    img = img[np.newaxis, :]
    return img

def predict(url):
    img = get_image(url, show=True)
    # compute the predict probabilities
    mod.forward(Batch([mx.nd.array(img)]))
    prob = mod.get_outputs()[0].asnumpy()
    # print the top-5
    prob = np.squeeze(prob)
    a = np.argsort(prob)[::-1]
    for i in a[0:5]:
        print('probability=%f, class=%s' %(prob[i], labels[i]))
```

In [48]:
```
# list the last 10 layers
all_layers = sym.get_internals()
all_layers.list_outputs()[-3:]
```

Out[48]:
```
['flatten0_output', 'softmax_label', 'softmax_output']
```

In [14]:
```
fe_sym = all_layers['flatten0_output']
fe_mod = mx.mod.Module(symbol=fe_sym, context=mx.cpu(), label_names=None)
fe_mod.bind(for_training=False, data_shapes=[('data', (1,3,224,224))])
fe_mod.set_params(arg_params, aux_params)
```

In [15]:
```
def get_features(img):
    fe_mod.forward(Batch([mx.nd.array(img)]))
    features = fe_mod.get_outputs()[0].asnumpy()
    return features
```

In [18]:
```
img = get_image('https://icatcare.org/app/uploads/2018/07/Thinking-of-getting
features = get_features(img)
print("{}\n shape: {}".format(features,features.shape))
```

```
[[ 2.731811    3.9008982   4.8866773   6.902369    5.587853    5.5976596
   3.6299744   5.6956005   5.783274    5.9859447  10.328437    7.249119
  10.089799   11.63578     5.579315   10.29896     9.659985    9.824672
   5.997461    9.134588   10.0658655  10.166415    9.8080845   6.289982
  13.240795    1.9723082   7.6495514   3.712172    2.7841299   6.2934337
   6.9453053   8.631689    7.29645     6.801901    6.52334     5.698734
   8.282388    3.830843   10.181621    8.7683935  12.962755    9.04188
   9.2452     10.77944    11.305762    5.7258263  11.200751   10.964636
   3.4149942   5.8307867   7.5116644   9.0251      3.9738111   3.9767187
   8.057208    4.70411     8.180446    4.212362    7.947       5.900941
   7.274863    9.651376    6.8402615   3.0084639   4.359813    4.1929684
   8.628105    3.9853938   5.851279    7.3413982   7.4339814   6.6312613
   8.691614    8.439708    8.843549    9.059312    9.449075   12.678203
  11.697756    7.8836474   7.3444357   6.451958    6.8108234  11.757459
  10.983695    7.6246867   5.50387     7.03473    12.915622    8.201405
   3.994502    5.4100966   6.856489    6.924605    9.808682    6.971078
   4.193388    3.5059521   1.6024159   4.7404647   3.3018126   2.9084258
   2.415486    5.3541465   8.748269    5.6931176   5.7850704   6.1290493
   5.1263065   5.997225    7.454259    6.640497   11.303702    7.120902
   9.347335    8.703756    4.8624525  12.27395     6.1802354   4.480561
   5.6459785   1.0452362   6.122116    5.4823976  11.299996    4.1871934
   7.719489    2.937037    4.467442    4.9877334   5.112029    6.401343
   9.139223    8.80435     6.144146    2.6501632   4.6860394   3.7774022
  10.005826    8.510313    7.547855    7.3255305   7.330723    3.8226004
   4.816145    6.8668685   4.516519    7.42678     6.5116243   3.608323
   7.267048   10.171126    7.746412    5.281473    8.630499    5.626668
   6.502434   13.557568    9.849358    4.837753    4.3339353   3.7696319
   5.360929    2.2633295   2.9870687   2.431563    3.9928882   2.2981818
   3.342405    6.635318    2.1925945   8.712073    4.962669    8.373492
   4.807753    2.6049435   4.978068    3.2130506   7.113287    1.5388217
   2.2567575   3.4583302   2.7490807   4.909944    4.181       3.6504633
   5.1258836   7.6492605   5.345396    4.6102114   3.7957761   2.6940234
   6.086538    5.387833    1.9049265   8.116254    7.423186    3.3709123
```

```
 5.39928     6.287825    3.9034204   6.0577965   4.336672    7.3916006
 5.458247    3.5410912   0.81163764  4.913134    3.077363    2.524287
 3.4502792   5.504821    5.5890636   3.1113174   1.8214495   4.9293275
 4.6584773   5.1635823   3.283465    3.5414634   2.3487635   1.1612074
 4.398783    7.2059193   7.16017     5.134449    3.4099345   7.284354
 1.0771716   3.487802    7.6779857   6.426784    6.5381465   2.5813906
 4.531569    5.8846455   6.2358975   7.270987    3.3144867   5.6216626
 4.9576645   2.6886454   7.497139    3.8266113   2.013793    6.8338447
 5.250487    2.5354967  10.352083    6.9917502   9.562755    5.32888
 7.01524    12.10749     7.255442    2.042023    2.8452964   4.24453
 5.7447877   5.4434137   1.5472516   3.399714    6.343063    9.781341
 8.963383    3.3579893   1.9317482   1.5271794   5.028176    7.896083
 7.771656   10.337724    9.5677395   7.921704    7.9268093   5.369312
 6.6163335  10.761533   11.807285    9.075544   11.988953   22.69674
22.162317   12.080123   12.826468   21.835646   14.643052   20.231905
13.029122   13.020738   10.732139   10.547261   15.8279295   7.679766
 1.9486965   1.9702134   3.064889    3.9313185   8.431654    9.492553
 6.441333    7.102499    7.8435125   9.124731    6.701487    3.3743453
 7.6716223   9.54777     8.999782   10.56783     9.4135     11.694572
 8.665095    8.841963    5.558522    8.441072    9.33401     8.717615
10.550846    8.072866    7.4475694   5.9206643  12.910718    5.2324095
10.362946    6.9798365  11.14673    10.812778    6.1894913   4.9107876
15.443866   17.415867   11.733549   11.387103    3.4203026  11.02024
 5.4453588   3.4969053   6.37595     6.3474665  13.639744    7.9808803
 6.675927    6.944559    7.7240233   5.366458    4.750808    6.613134
 4.9604473   6.077957    9.4021015   7.6230927  10.184332   10.919796
 4.308359    9.011373    9.799604    9.725924    5.6023226   5.693448
 7.512158    4.707306    7.7173443   7.0380173   2.5001922   8.207552
 7.1257854   7.286926    4.864601    4.420124    7.9287887   7.4102335
 3.3168476   4.6729074   6.184408    4.9319315   2.6935422   7.1012454
 2.4658332   2.8927302   5.3334613   2.9348176   4.585414    7.929239
 6.5128427   4.916825    4.062633    6.6393585   2.137959    9.514823
 7.6458163  10.8959055   5.6789308   5.738284    8.675852    7.4208884
 4.5315175   9.101155    9.213087   11.059609    8.1790495   9.600116
 8.79123     2.672641    5.024305    2.4073465  12.150895    0.76296645
 0.5038572   7.698404    4.465545    6.7847085  10.827217    4.0900283
 6.0966697   8.6208      7.5405455   3.555274   13.943614    9.350677
 7.003285   11.929672    6.121218    3.8996964   5.9602203   1.7010777
 7.0568337   4.2559648   8.233021    7.9206133   8.374679    7.7616096
10.31823     6.185758   14.058908   12.32454     4.4500713   3.797952
11.232954   10.459605    9.267935   13.071583    5.9936566   9.338697
 3.3234684   7.087867   10.936118    4.7873693   8.6668625   4.6345
 5.067448   13.847438   11.273699   13.505603   11.774431   11.961157
10.948317   17.55803     7.588044   10.792003    3.4737253  13.225761
 4.9713287  14.716065   10.018598    3.4421518   3.917945    4.786414
 3.2410662   8.168964    9.910036    4.873431    3.3967483  16.29944
10.226325    5.2889414   3.4686158   8.962409   13.654783    7.991584
 4.363832    8.174877    5.6081524   6.8352046   3.5675802   7.301462
 5.1853724  10.017093   11.806421    9.212146    8.442925    5.801412
 9.667005   12.208715   11.963174    7.345216    9.081496    9.427848
 8.753899    7.291614    1.8946741  11.63785     9.241548   13.928189
13.061649   15.367339   12.914506   12.344784   11.604443    8.127105
 2.5657384   2.1852183  15.490842   10.774004   16.341034   11.205921
```

```
 9.197769    5.786353    4.4183006   9.60548     8.044159   11.22243
 5.2172      5.9751215  10.967845    3.1460898   5.2085066   9.072665
 7.3603706   7.6866255   7.0325317   3.9155037  10.085673    6.1941175
13.6523285   6.0107265   2.6337411   4.906931    7.181141   14.790271
 2.850151    7.7269645  13.043631   15.613863    2.5837495   9.394104
 8.667604    0.41363364  4.146783    9.6787195   9.279223    8.79654
16.404696   11.51936     5.368806    1.9849385   7.5876045   6.9396825
14.072977    4.3137946   4.2473345   2.52747    14.339635   14.996346
 5.089149    2.257803   10.512023    7.163758    6.479707    0.69810337
 8.151603    2.1268332  19.057432    0.35773617  7.0367436   0.763118
 3.461458    6.594456   13.109665    7.9887624   4.0685062   7.1986885
 8.8045025   3.918889   11.875958   12.563931    1.2866048  10.684031
10.414188    9.318345    6.433751   10.80721    13.9467     13.667537
10.968392    0.314351    8.297129    7.201408    5.26485     9.962434
12.999093    6.0167007   6.755423    4.2900414  12.476637    9.821192
13.725554    3.7854536  12.910077    1.9558831   8.380435    8.893559
 2.7058535   5.4100466   7.5020165   5.542321    9.119853    9.737722
13.1177     10.940123   11.175706    1.4065359  14.934692   15.472114
11.472243    2.4852593  14.423504    2.9746687   2.8564596  11.771427
 6.131822   11.445189    9.001334    9.262844    2.1180599   9.955629
 7.700889   13.58174     8.407601    9.292692    3.1940012   3.2324188
 5.7446823  17.093952    9.24341     3.5197124   3.9177275   9.475568
13.095527    5.0687804  10.414358   10.499538    5.890953    8.429687
 3.6683652   9.282003    1.9323317   4.94462     8.410937   10.613594
 3.3637269   2.1542635   9.364938    9.538077   11.225166    2.4386787
 9.07326     6.725077    5.5647864   7.6732163   6.2793345   3.7686021
 1.9405173  10.818339    9.915367    1.2044474   5.1299224  12.860835
 4.752086   11.391398   13.650657   10.387296   14.147362   12.050395
 6.5631723   2.067395    5.991474    8.105744    6.441248    8.340168
 3.8919115   4.535644   12.865545    5.725673    3.525283    9.760836
15.003663    7.715211    7.301467    6.431038   14.705736    3.2747548
 8.713833    5.6867237   8.02887     3.5077865   5.1369505   7.5494084
12.989616    8.225568   13.512066   16.237343    8.576901    4.9031467
10.67287     7.3488264  11.329115    1.726067    4.0114665  11.25949
 9.169683   11.62729    15.071682    9.140865    1.1424178  17.970129
 9.407948    2.7270193  18.140476    9.003847    2.5072138  10.93585
 4.9892464  11.738513    0.5201955   7.8873      4.248439   11.841048
11.20146     7.455632    7.2535      8.160788    7.776449    4.217444
 5.5322924   7.02042     4.0031557  10.046072    8.734596   15.27913
 9.0532      2.3913183   5.7322516  18.22416     7.513456    3.9214144
 5.300208    3.04205     6.3857603   8.329536   11.639784   17.700708
 8.512333    8.652157    7.3959546   7.1831307   5.430406   13.718815
 6.624551   15.252481   14.832652    6.638097   11.76788    16.180166
 8.354548    7.0313063  11.555988   13.561804    7.1899123   3.1991363
 3.8116345   2.8905377  11.651768   15.387637   14.0408535   6.565447
13.462717   12.679193   10.703341   11.839082    6.4391747   7.2293053
 6.2900877  10.939683   17.566399    7.92463     5.329709    7.7819667
 3.584652   13.153734    7.8821006   4.1144996   1.4778733   0.96194965
11.641569    6.847817   10.172745    4.8699865  10.389147    8.457949
 6.292947   11.046793    4.5261507   9.97533     2.824687    8.58576
 7.7183886   3.741003    9.822101    7.828369    2.1353517   4.3781743
 6.6460557   9.450028    8.374486    1.8673849   6.9086323   7.694237
10.442545    1.6138116   4.992464    5.0941353   5.5209217   3.50121
```

```
 11.521552   13.694515    4.702955    6.5822797  11.461238    2.6599429
 11.959796   11.3552      5.2679377   3.2046185  10.093634   12.032435
 10.127258    2.407112    6.0026207  11.668374    5.411328    9.999649
  9.526369    2.4523501   7.007791   16.546972    1.5456226   8.56724
  3.0186112  13.096735    8.174237   12.758465    9.848465    9.257078
  0.96111006  5.5220346   2.5690815   2.9280348  10.095376    8.211961
  3.4089372   2.830946    7.2162166   9.619767    0.22439507 10.54825
 10.733223    7.3180695  11.090987   11.247994    6.3090415   7.3976064
  6.7536345  17.710531    8.317595   11.241352    7.7849245  11.963486
  3.5570066   8.755916    5.8202972   6.244552    9.4146185   8.909975
 11.538079    6.4253035  17.255009   13.751663   11.622289   15.469001
  4.237758    7.736162   10.365318   13.24044    18.205471   14.377344
  9.415742   10.363938    3.6586409   9.332913    9.287965    9.225867
 10.726653    2.3643985   3.2395904   2.773296    7.177313   10.607796
 13.280321    7.7800007   3.320212   10.017647    6.189191    6.967745
  5.764682    8.264951    2.065886    9.877997    9.344261   11.44254
  5.910216    4.108808    5.1819353   2.406356    4.4888806   4.482684
  3.3455486  12.267346    6.304007    5.7975836   5.489004    2.9057488
  8.029712    7.311815    5.331028    6.780764    7.2698026   7.03161
  6.9609914   8.205074    6.2120485   9.704169    4.3887997   9.308927
  5.334109    6.4561872   5.7876287   4.3589535   7.1413755   4.1641693
  8.003898    8.619182    2.5742774   2.4626925   3.8087509   5.987408
 10.418253    9.594042   18.342413   11.671421    2.637681    9.242509
  2.5423734   6.47289     5.5148525   5.139831    1.7740538   3.9243221
  5.780046    5.5538406   2.6732502   7.5682645   8.417677    2.763147
 10.0679455   9.356203    5.7668076  13.894629    7.128599    8.015945
  4.268324    7.279767    3.9210467   4.57002     6.2065      5.7564244
  7.1734467   3.3807797  11.519496   16.3085    ]]
shape: (1, 1000)
```

In [19]:

```
# from https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data
from os import listdir
from os.path import isfile, join
import os


mypath = join(os.getcwd(),'train')

cats_imgs = [join(mypath,f) for f in listdir(mypath) if f.startswith('cat')]
dogs_imgs = [join(mypath,f) for f in listdir(mypath) if f.startswith('dog')]
```

In [20]:

```
print("cats: {} and dogs: {}".format(len(cats_imgs),len(dogs_imgs)))
```

```
cats: 12500 and dogs: 12500
```

In [22]:

```
Nmax = 100 #(N: 10, 100, 500, 1000, also 5000 and 12500)
cats_features = [get_features(get_image(img)).ravel() for img in cats_imgs[:Nm
dogs_features = [get_features(get_image(img)).ravel() for img in dogs_imgs[:Nm
```

In [24]:
```
Y_cats = np.array(Nmax * [1])
Y_dogs = np.array(Nmax * [0])
```

In [25]:
```
X_cvd = np.vstack([cats_features,dogs_features])
Y_cvd = np.vstack([Y_cats,Y_dogs]).ravel()
```

In [26]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_cvd, Y_cvd, random_stat
```

In [27]:
```
from sklearn.linear_model import LogisticRegression

lg = LogisticRegression().fit(X_train, y_train)

print("Test set score: {:.2f}".format(lg.score(X_test, y_test)))
```

```
Test set score: 0.88
/opt/conda/lib/python3.7/site-packages/sklearn/linear_model/_logistic.py:940:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regress
ion
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

In [30]:
```
img = get_image('https://s3.amazonaws.com/cdn-origin-etr.akc.org/wp-content/u
features = get_features(img)
print("{}\n shape: {}".format(features,features.shape))
```

```
[[ 3.8680282   3.0770736   3.8007517   3.9388447   4.4794497   2.513034
   1.0257548   3.3912466   4.1260457   1.1125107   3.0114415   2.2824717
   3.308097    3.6617973   2.2812922   1.8932449   4.174476    3.710502
   0.45999247  4.1961174   1.2289764   9.014702    6.0741653   5.186479
   4.293989    3.2768373   4.330217    3.7503705   6.209451    4.7506633
   3.809687    8.039387    5.815378    4.479296    4.644457    2.287041
   4.105688    3.3356824   8.155916    4.587722    4.465425    3.855896
   4.043045    5.1172795   4.4702134   3.342398    5.4953494   6.7198224
   3.84865     1.182495    2.0083091   9.924515    2.7754364   2.005786
   3.0824454   1.4339454   2.6161735   1.7753203   1.2789447   2.4843435
   1.5804738   6.9447136   4.049789    2.9182622   1.8067827   1.929847
   2.0441859   2.8852274   2.3605494   5.1194777   2.5484276   5.3415647
   3.779574    3.719067    4.224366    4.0597167   2.472064    2.177067
   6.892461    4.2934427   4.110542    5.4240932   6.0283465   4.938808
   1.0311867   1.9734994   4.873224   10.826043    6.586989   10.874734
```

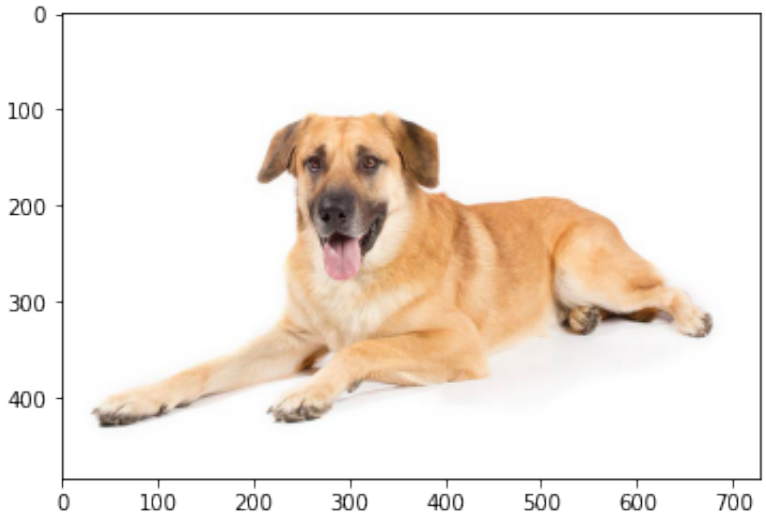
```
 3.7613354   2.369898    2.6786683   1.7927071   7.2485766   1.7895155
 2.0912905   4.3573813   1.0156956   4.274913    1.7799224   8.096338
 3.7537284   7.285345    7.295822    5.0679016   2.7826424   6.1655016
 2.4368806   1.3257196   4.5575867   6.473884   11.895296    6.8320866
 5.361       2.6531858   4.564212    5.8689027   1.9365345   2.6203759
 5.183484    1.7165029   3.0182095   2.274215    4.6858997   5.2946324
 7.652743    1.6650032   1.8852196   4.6898293   2.1779253   1.8760023
 1.9162517   3.1303358   2.3637066   0.35587963  2.2771838   1.5375512
 1.5735669   2.5764523   2.037687    1.8024894   0.71565324  1.8617755
 3.788113    2.082352    3.8433325   1.4228038   3.1118274   7.0973954
 5.8102417  12.48751     9.515371    8.887295   10.349273    8.000686
 7.0232496   8.454158   10.69126    22.593256   20.629393   21.447111
18.516518   24.88076    16.95143    18.494385   20.20789    19.739748
21.395742   17.001986   20.08869    16.340874   23.145485   12.96442
13.869895   13.64452    21.876612    9.410792   21.389843   17.367453
18.362257   16.216228   11.91341    10.4619875  14.590394   12.6465645
 8.882405    7.0168157  11.795401   13.819958   14.217406   11.200841
11.822543    8.74059    11.354834   11.930998   11.440539   15.740785
12.229618   10.81119    13.474916    7.0924706  15.383324    8.773028
10.668766   15.059237   12.254301   23.055195   25.893064   17.828959
16.51152    19.407652   15.712897   12.098238   12.826345   11.783195
15.069426   10.8568325  13.442257   14.364362   13.296346    6.623045
17.27046     9.067432    7.9476194  16.236116   12.765402   12.864075
 5.9975576  10.899133    8.6172495  12.071175   13.07105     8.021412
16.05708    14.175797   15.534326    9.862089   17.731947   10.223112
11.533164   12.68937    20.79065    24.117907   14.96606    14.423234
26.991064   19.944319   14.93112    10.169757   10.967558   16.292212
 4.698208   13.856794   16.348349   13.423142   12.702295   17.278471
 7.61129     7.610845   11.838387    6.1189146  10.275008   11.141909
12.604921    7.161967    7.791848   14.553309    8.073789    9.755717
11.696058    7.3238277   6.3458424  12.541845    5.6082597   7.0530615
 7.7985682   4.5882974   2.4718292   7.9238615   2.8448157  10.304632
10.497861    5.761474    7.3295565   8.329276   13.054941    7.515375
 5.5412393   4.6372547   9.203834   16.512094   12.6724615   7.339274
 9.8838415   5.5945525  17.508465    3.043341    6.384654    8.147824
 2.899172    5.801489    5.7515035   4.71007     4.05036     2.2311087
 7.5179873   6.5069475   3.450876    5.650288    5.044524    7.6705995
 5.2298627   3.258254    5.1044135   3.3946898   5.014812    6.9599953
 3.4363325   2.3107822   2.1962838   1.4675207   1.6809529   1.153967
 2.3533711   1.9169884   2.5416303   3.2292058   3.1549604   3.1286445
 3.5050056   4.5812      4.082107    5.0206604   3.5471501   3.139664
 0.7460084   2.6777484   4.8386784   9.963821    6.791592    6.922235
 7.068977   10.095261    3.6340406  12.9250965   9.878363    7.8949637
 9.331418    8.254994    8.581553    7.0324945   9.6927      7.8517175
 9.84396     5.667616    5.898995    3.1890006   2.979977    3.8310874
 2.8480525   2.4231687   3.7270696   5.2704873   6.151808    7.5599146
 4.2671785   2.4458957   7.8349195   2.933027    3.9622145   6.083128
 6.6699862   5.0373363   3.8840804   4.243526    6.5641766   4.7457433
 2.9339612   8.081371    5.8358703   4.752475    5.7170324   3.7892807
 3.7321076  10.56513     8.151451    1.2368964   3.1142998   4.0869164
 5.3213515   7.4281797   3.529985    2.3502243   9.001321    4.9532614
 3.56115     3.601671    4.482304   10.766756    9.704199    4.901464
 7.957876    3.478146    4.8407044   5.4617915   5.372067    4.1909266
```

3.9824352	7.826153	5.374219	6.6364446	5.2077975	5.048653
9.767099	5.474588	8.673956	5.8593984	8.8003645	8.977765
5.764562	5.8856754	10.115283	5.656527	6.3020754	1.7988725
10.137626	9.213063	8.932798	5.3274226	6.3451176	9.295259
4.603016	8.545669	14.786184	13.337121	3.5536613	1.9876574
4.2108045	2.117066	5.512703	2.2609928	2.7838316	6.528533
6.0566444	5.979617	5.3576627	6.6773214	6.8237157	1.7327584
6.0771494	7.8172703	11.391459	6.0309772	1.3873978	5.3197937
6.052633	8.771865	5.7783628	8.199217	2.2088525	5.900612
6.204755	10.817404	10.068063	8.559832	3.0130277	4.2749343
3.8140361	4.93711	6.4321914	5.121246	5.715035	7.9921656
7.1010275	3.2137325	6.8648405	9.041757	4.0067716	6.1250052
5.9150586	7.1603413	6.480427	1.0599625	3.0898623	3.4731574
5.867129	13.734294	6.9826245	3.2448647	6.381004	9.151717
5.8937654	4.2751555	10.6635475	5.400883	6.8975425	1.6018436
4.3811593	7.975232	4.3894663	10.228202	7.7783813	5.880664
6.7807274	4.118661	6.90561	9.982913	5.96773	2.2416847
1.548204	2.2985659	10.267278	6.053343	9.580989	10.09984
10.789435	6.78255	5.2336874	6.3093348	7.042093	3.0710175
10.939332	11.230664	6.038644	2.8797998	7.801449	7.3003435
6.91485	11.224192	7.3530545	7.907427	4.509783	5.1748543
12.468839	6.913947	2.0386205	12.003853	3.085519	11.60327
1.3448868	7.9728827	7.799876	11.132099	2.4971178	7.5451884
7.0501103	2.9947848	6.0505033	3.7928298	5.7613707	7.4438334
8.931353	6.358954	2.9246376	5.866925	3.8573735	4.7125697
9.554073	7.59521	5.839603	4.3058724	8.322736	6.196823
2.8935046	3.7616353	4.8635144	6.077796	9.921986	6.3661246
16.471178	5.241932	5.9486594	3.730866	3.8174648	1.1093864
1.3543245	7.933642	10.865643	5.02124	2.1542604	4.1125402
2.322716	7.7227135	10.350984	9.356061	2.951054	10.056329
5.118172	10.002452	7.786815	8.1611185	7.012722	9.483933
3.8071153	4.9451365	9.98679	7.114231	3.4956067	7.6764116
9.136812	8.403166	9.762652	7.0766582	6.5847936	7.2631617
10.181375	7.8843946	8.201239	0.9051311	9.523975	7.601859
4.7029924	7.6920447	7.8097067	7.2236757	9.573702	8.200586
8.244837	6.8137712	8.338899	6.05952	8.793468	8.873822
3.4639664	2.2926931	6.0000486	4.795587	1.4521159	10.090369
6.5389924	8.860401	7.4124527	9.4020815	3.9597404	6.983886
8.8215475	3.1968884	8.460633	6.128111	1.1649846	10.836473
7.646004	9.700603	9.831387	4.198988	4.0200505	7.5691924
4.557316	1.9791524	8.929832	6.945005	4.426864	2.422814
7.9006166	8.108625	2.6157663	2.4308028	5.033595	2.4171014
2.6559796	1.1304991	6.3882856	1.7867712	5.1571746	2.4655747
4.9779305	9.117288	3.5420427	6.5979075	3.8510625	2.9500475
9.604422	9.3112335	11.304276	4.434931	19.25583	6.695858
11.131171	9.075669	12.895	12.4662	3.0523033	7.029521
9.507388	1.3680283	6.813352	1.3718091	6.606264	7.6512923
9.496167	13.983543	5.5896544	4.2972054	1.8104817	4.544655
9.519791	12.296036	2.3290906	7.105483	12.78564	5.8050447
11.376695	1.9384876	2.4535577	5.3863897	1.5590639	7.798127
8.38577	9.016366	15.192917	10.590793	4.647105	6.8072195
8.513748	4.1790586	2.229322	4.4584036	2.4709072	10.045794
11.878091	4.518135	7.952676	3.2023196	2.7050774	7.8332987

```
  8.105803    1.7165117   8.605458    9.766088   11.855645   10.479746
  7.8026514   7.2911897   4.7548037   8.3493185   5.568733    7.7704573
  4.4971113   6.976409    9.309905    6.114211    8.529177    4.183968
  2.623794    5.056159    6.604316    7.487247    8.3725815   7.819886
  5.4270096   3.8981054   6.2843227  10.116517    7.06578     4.6473513
  5.0662637   5.378877    7.4909487   6.154413    4.636438    8.681302
  4.845439    7.805458    5.917714    6.4445877  12.458542    9.402724
  2.63179     7.6309524   8.150517    4.7716365   6.4135923   9.693477
  5.629182    7.3680015   9.480289    5.6531177   7.636259    4.543409
  2.8130362   2.5628386  10.775148    8.83947     9.697046    8.9554405
  5.122448    2.8479238   6.6367116  10.020451    5.7062306   5.554301
 10.600308   11.314349    5.623864    9.765764    9.736605    8.770255
  6.3605294   4.037713    4.383122    8.11133     3.4699535   5.010916
  9.313533   12.367925    6.7642603   3.3070233   7.72455     3.3671126
  2.723685   13.013277    4.3527923   8.93426     3.0037897   2.9002788
  9.682216    2.4089184  10.428298    5.4077883   3.6294236   0.82561
  8.775564    8.402793    8.439979    2.38039     5.6820807   4.6136727
  8.871776    3.9851723   9.375548    9.182731    3.058218    3.459215
  5.4962754   4.018234    8.839286    8.8314085   9.505224    2.3476098
  8.823815   12.072752    8.601356    7.9471993  11.292472   11.457238
  6.027743    7.071573    7.0377216   2.9373167   7.307343    8.48312
 19.111702    2.9167256   5.7694364   9.05401     7.0445848   5.576129
  2.3362818   8.641402    5.7351575   6.7036204   5.4995666   6.012248
  2.507907    6.9093833   5.001413    4.078641    4.557381    9.426562
  4.4181395   2.9529035   8.198942    2.3040478   2.800392    4.689731
 13.018832    2.6840668   3.003842    6.659847    4.55896     5.778887
 10.350336    5.926081    4.4054236  14.795793    1.2430875   7.5658617
  0.87765217  7.2287545   6.850049    4.951525    8.017592    7.024554
  5.953328    7.7224193   6.606814    6.7617393  11.722911    6.332674
  3.7297676   3.657504   12.947633   12.445496    3.9683583   4.0251756
  5.04758     7.196997    4.534546    5.3412795   9.387548   11.5132
  4.3527474   3.6554115   3.1876783   3.529498    8.873071    4.952435
  6.8690434   2.9971051   2.8173728   6.308332    1.3162148   7.098621
  5.2898364   3.9596841   1.9113549   1.9215572   7.324026    9.437474
  7.8335223   5.1925454   4.6072373   4.2747693   6.805999    3.2848585
  8.721183    7.721843    5.8097773   4.0527506   4.101082    5.9977326
  4.32023     3.37436     6.048842    5.917717    2.8842163   7.8604107
  3.3678865   4.4946995   3.9107037   4.710452    3.8326082   3.741211
  6.5966315   4.096897    7.036597    6.277786    7.1407843   5.955763
  7.9515915   7.523023    5.278676    2.6221385   3.9909554   8.753346
  6.462921    1.9766583   5.333998    3.7846956   2.9319718   6.691585
  6.369771    2.7630775   2.607686    3.20876     3.6895769   5.622818
  5.5627694   3.9689782   2.0877512   5.8020372   7.9343553   2.7381692
  2.6250763   3.7928612   1.1395389   5.831917    5.012505    4.5884542
  4.925548    5.024918    3.991863    6.383615    3.1053205   6.64485
  9.6873665   5.221062    5.8723288  13.390061  ]]
shape: (1, 1000)
```

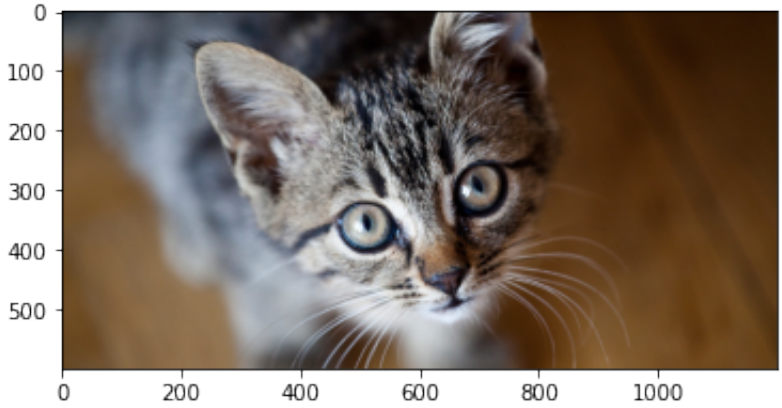
In [31]:

```
import matplotlib.image as mpimg

dog_test_path = join(os.getcwd(),'Chinook-On-White-03.jpg')
cat_test_path = join(os.getcwd(),'Thinking-of-getting-a-cat.png')

img = mpimg.imread(dog_test_path)
imgplot = plt.imshow(img)
plt.show()
```



In [32]:

```
img = mpimg.imread(cat_test_path)
imgplot = plt.imshow(img)
plt.show()
```



In [35]:
```
features_out = get_features(get_image(cat_test_path))

prob = lg.predict_proba(features_out)
pred = lg.predict(features_out)

if pred[0] == 1:
    fpred = 'cat'
else:
    fpred = 'dog'

print("prob: {} and prediction: {}".format(prob,fpred))
```

```
prob: [[7.82527617e-08 9.99999922e-01]] and prediction: cat
```

Repeat all previous steps using MobileNet V2 (https://github.com/KeyKy/mobilenet-mxnet). How the two networks compare?

In [36]:
```
!wget https://github.com/KeyKy/mobilenet-mxnet/raw/master/mobilenet_v2-symbol
!wget https://github.com/KeyKy/mobilenet-mxnet/raw/master/mobilenet_v2-0000.p
```

```
--2020-11-07 12:04:12--  https://github.com/KeyKy/mobilenet-mxnet/raw/master/m
obilenet_v2-symbol.json
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/KeyKy/mobilenet-mxnet/master/mobil
enet_v2-symbol.json [following]
--2020-11-07 12:04:13--  https://raw.githubusercontent.com/KeyKy/mobilenet-mxn
et/master/mobilenet_v2-symbol.json
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.84.
133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.84
.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 108557 (106K) [text/plain]
Saving to: 'mobilenet_v2-symbol.json'

mobilenet_v2-symbol 100%[===================>] 106.01K  --.-KB/s    in 0.05s

2020-11-07 12:04:13 (1.99 MB/s) - 'mobilenet_v2-symbol.json' saved [108557/108
557]

--2020-11-07 12:04:14--  https://github.com/KeyKy/mobilenet-mxnet/raw/master/m
obilenet_v2-0000.params
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/KeyKy/mobilenet-mxnet/master/mobil
enet_v2-0000.params [following]
--2020-11-07 12:04:14--  https://raw.githubusercontent.com/KeyKy/mobilenet-mxn
et/master/mobilenet_v2-0000.params
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.84.
133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.84
.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14178924 (14M) [application/octet-stream]
Saving to: 'mobilenet_v2-0000.params'

mobilenet_v2-0000.p 100%[===================>]  13.52M  34.5MB/s    in 0.4s

2020-11-07 12:04:16 (34.5 MB/s) - 'mobilenet_v2-0000.params' saved [14178924/1
4178924]
```

In [102…

```
# define a simple data batch
from collections import namedtuple
Batch = namedtuple('Batch', ['data'])
```

In [103…

```
context=mx.cpu()
```

In [104…

```
def get_image(url, show=False):
    if url.startswith('http'):
        # download and show the image
        fname = mx.test_utils.download(url)
    else:
        fname = url
    img = cv2.cvtColor(cv2.imread(fname), cv2.COLOR_BGR2RGB)
    if img is None:
     return None
    if show:
     plt.imshow(img)
     plt.axis('off')
    # convert into format (batch, RGB, width, height)
    img = cv2.resize(img, (224, 224))
    img = np.swapaxes(img, 0, 2)
    img = np.swapaxes(img, 1, 2)
    img = img[np.newaxis, :]
    return img

def predict(url):
    img = get_image(url, show=True)
    # compute the predict probabilities
    mod.forward(Batch([mx.nd.array(img)]))
    prob = mod.get_outputs()[0].asnumpy()
    # print the top-5
    prob = np.squeeze(prob)
    a = np.argsort(prob)[::-1]
    for i in a[0:5]:
        print('probability=%f, class=%s' %(prob[i], labels[i]))
```

In [105…

```
sym, arg_params, aux_params = mx.model.load_checkpoint('mobilenet_v2', 0)
```

In [106…

```
# list the last 10 layers
all_layers = sym.get_internals()
all_layers.list_outputs()[-5:]
```

Out[106… ['fc7_bias', 'fc7_output', 'fc7_flatten_output', 'prob_label', 'prob_output']

In [107…

```
fe_sym = all_layers['fc7_flatten_output']
fe_mod = mx.mod.Module(symbol=fe_sym, context=mx.cpu(), label_names=None)
fe_mod.bind(for_training=False, data_shapes=[('data', (1,3,224,224))])
fe_mod.set_params(arg_params, aux_params)
```

In [108…

```
def get_features(img):
    fe_mod.forward(Batch([mx.nd.array(img)]))
    features = fe_mod.get_outputs()[0].asnumpy()
    return features
```

In [109…

```
img = get_image('https://icatcare.org/app/uploads/2018/07/Thinking-of-getting
features = get_features(img)
print("{}\n shape: {}".format(features,features.shape))
```

```
[[-8.73116531e+01  2.83228577e+02 -3.59607025e+02 -4.03708984e+02
  -6.46255798e+02  1.48531494e+02 -3.47858490e+02  6.66299515e+01
   2.78768982e+02 -1.49736664e+02  8.38041992e+01 -1.37439499e+02
  -4.69580017e+02 -9.14800415e+01 -2.76609131e+02  1.35187820e+02
   1.80837677e+02 -3.41249298e+02  1.15875607e+01 -5.60398926e+02
  -2.54541214e+02  3.32528320e+02  2.52942486e+01  4.03285645e+02
   2.14639755e+02 -4.70253792e+01 -4.01036255e+02 -2.38460938e+02
  -5.36336288e+01 -6.14872131e+02 -3.40083069e+02 -2.73898651e+02
  -3.69430878e+02 -6.40303528e+02 -2.56124176e+02 -4.16391968e+02
  -1.71422821e+02 -2.47153595e+02 -1.99477921e+02 -1.43629562e+02
  -3.21645264e+02 -4.46027039e+02 -2.44809418e+02  1.03301361e+02
  -1.46011917e+02 -4.25061462e+02 -3.44877625e+02 -5.04795135e+02
  -1.92245743e+02 -4.46315369e+02 -7.29045776e+02  6.43142151e+02
  -5.43736694e+02 -3.97714417e+02 -5.26605774e+02 -7.36024475e+02
  -1.60982224e+02 -2.95890625e+02 -7.81803040e+02 -3.25579132e+02
  -8.40681839e+01 -6.43729919e+02  4.14718666e+01 -3.34889435e+02
  -5.07559692e+02 -1.62592194e+02 -1.88286270e+02 -6.47604553e+02
  -1.27652222e+02 -1.61137512e+02  1.22373108e+02 -4.67065430e+02
  -2.27845261e+02  4.93836746e+01 -8.06280289e+01  5.41381775e+02
  -2.29811203e+02 -4.32604919e+02  1.78805542e+02 -2.30924088e+02
   7.84939880e+02 -3.04035858e+02  2.51537018e+02  5.64096863e+02
  -1.44117874e+02 -2.62628815e+02  2.57276184e+02 -1.06053314e+02
   3.47322205e+02 -1.54493103e+02  5.06014801e+02  4.19897217e+02
   3.65872421e+01  3.79745605e+02 -3.07246521e+02 -9.59804459e+01
   3.53265350e+02 -1.74011169e+02  2.35754410e+02  1.05712608e+02
  -6.54890060e+01 -3.04422974e+02 -1.38272079e+02  2.41708221e+02
   1.80800858e+02 -2.11239624e+02 -2.68215027e+01 -2.94282776e+02
  -2.98940186e+02  2.11309464e+02 -7.89026062e+02 -1.89356308e+02
  -1.41223419e+02 -2.24167953e+02 -2.93801178e+02 -3.43777405e+02
  -4.11104340e+02 -1.95378922e+02 -2.99134857e+02 -3.35203796e+02
  -3.44139679e+02 -5.81374756e+02 -2.53917221e+02 -3.83287689e+02
  -2.50095322e+02 -3.55800873e+02  2.17763214e+02 -5.51281616e+02
   1.30699783e+02  2.94620914e+01 -1.50848785e+02 -4.85231018e+02
  -1.26667007e+02 -2.28896347e+02  1.63994827e+02  2.31013756e+01
  -1.94970383e+02  1.25436058e+02 -4.93182648e+02 -8.36208313e+02
  -4.18790253e+02 -5.33034363e+02 -4.73317169e+02  1.28507263e+02
  -1.32925348e+01  5.90635315e+02  8.77724380e+01 -3.58027588e+02
  -4.41136566e+02 -3.02285736e+02  1.07119179e+00 -1.44132538e+02
  -2.11550201e+02 -2.19061661e+02 -1.40891693e+02 -2.84773926e+02
   2.95348999e+02 -1.26902390e+02  3.59937531e+02  9.61616058e+01
   1.11995499e+02  1.98080063e+02  2.89108459e+02  3.19403076e+02
```

```
 1.86888733e+02  2.90249420e+02  2.05764603e+02  1.00953957e+02
 3.50883728e+02  1.05283470e+01  1.97404594e+01  7.93311920e+01
-2.65052643e+02 -1.91770187e+02 -1.89773346e+02  3.59652191e+02
-2.24482971e+02 -1.81828568e+02 -3.96077995e+01 -1.84517471e+02
-1.15133095e+02  3.68220353e+00  1.20558014e+01  1.42111145e+02
-3.45617767e+02  2.79823151e+01  2.16883789e+02  2.06653503e+02
 9.09597626e+01  8.25691071e+01  1.72576523e+02  1.05898094e+02
-1.28174362e+02  1.79173088e+01 -1.36777267e+02 -2.57285767e+02
-1.18767557e+01 -1.63106949e+02 -2.16043793e+02 -3.23243591e+02
 1.35109970e+02 -1.59201477e+02 -6.96820602e+01 -2.45047211e+02
-2.47837555e+02  1.39939133e+02  3.85478912e+02  1.06226418e+02
 3.16129951e+01  5.30940895e+01  7.00790253e+01 -2.39284424e+02
 8.53175278e+01 -1.02620743e+02  9.40465393e+01 -6.29597855e+01
 1.32124374e+02 -1.12506584e+02 -2.45347748e+02 -1.95705582e+02
 1.74687637e+02 -9.07406158e+01 -1.95923782e+02  4.59715271e+01
 6.33930635e+00  1.81140499e+01 -1.86845581e+02 -1.83902988e+01
 1.86407089e+02 -2.11985519e+02 -1.51796463e+02  8.58177643e+01
 1.02529945e+02 -1.67597890e+00  2.43248016e+02  1.17565674e+02
 1.19809334e+02  2.58186493e+01  2.07018555e+02  2.73701978e+00
 1.59103775e+02  4.15268066e+02  5.38940525e+00  1.58592697e+02
 1.07136545e+01 -4.27159180e+02  8.76968460e+01 -2.63751717e+01
 1.46347595e+02 -8.79527817e+01 -6.22740669e+01  1.83022308e+02
-1.76826965e+02  2.44402893e+02 -1.11497116e+02 -1.96212677e+02
 1.14402107e+02 -4.76016808e+01  7.86039257e+00  2.03656998e+02
-2.75657593e+02 -1.20230446e+02  1.41498230e+02  2.03809128e+02
 7.40632858e+01 -1.85474319e+02 -1.35940720e+02 -1.38541855e+02
-1.31060211e+02 -2.82232544e+02 -3.66773743e+02 -5.53908272e+01
 4.87712517e+01  3.23161224e+02 -1.40864548e+02 -7.35979843e+01
-1.75181015e+02  5.93632622e+01 -1.11508743e+02  3.40113564e+01
-2.11392860e+01  2.46271225e+02  2.70948456e+02  5.56281494e+02
 2.11420151e+02  1.56546616e+02 -3.02101946e+00  1.38711777e+02
 1.93668884e+02  1.10855705e+02  2.67667236e+02 -8.87146301e+01
 1.72356125e+02  3.59029938e+02  1.80479111e+02  9.39077225e+01
 2.44827881e+02  3.34930481e+02  3.73865143e+02  3.15318512e+02
-6.31149780e+02  2.39657242e+02 -2.90182312e+02 -7.63055384e-02
-2.27925262e+01 -1.94211426e+02 -1.63546173e+02 -2.89429779e+02
-4.79892883e+02 -2.10450806e+02  2.01957047e+02 -2.70558746e+02
-2.17545898e+02 -1.40745438e+02 -5.18035545e+01 -2.54324829e+02
 2.51760117e+02  1.22443413e+02  1.94074997e+02 -3.11680115e+02
-5.69095093e+02 -2.43075348e+02 -2.67301941e+02 -3.31525208e+02
 1.05294247e+01 -2.30267975e+02  3.60962311e+02 -4.43286377e+02
-5.63287781e+02 -6.30630005e+02 -8.98659897e+01  2.87421951e+01
-6.46856232e+01 -1.04104645e+02 -6.14435844e+01  1.10315269e+02
-3.39663940e+02  2.27345596e+02  2.67360809e+02 -3.88327751e+01
-1.40210892e+02  7.13622589e+01 -1.71835510e+02 -4.05042847e+02
-1.04456337e+02 -1.31367447e+02 -5.86992722e+01 -2.73099060e+02
-2.17096481e+01 -1.74208893e+02 -1.49035049e+02 -6.06719971e+02
-5.06569519e+02  5.03048859e+01 -1.83232971e+02 -2.73536926e+02
 4.23060112e+01  7.76638794e+01  6.58991699e+01 -1.44595032e+02
-4.31631565e+00 -3.02489567e+01 -2.78883667e+02 -1.96226517e+02
 8.43389511e+01  5.99944191e+01 -2.97409992e+01 -9.16810150e+01
 8.46862335e+01  1.65563400e+02  5.97852112e+02  5.63169250e+02
 4.08793427e+02  4.59753143e+02  5.49544983e+02  3.70307312e+02
```

```
 2.29478149e+02  8.46711731e+02  2.06450806e+01  2.77336853e+02
 3.49461731e+02  1.38802216e+02  3.13532227e+02  9.21686523e+02
 5.83529602e+02  5.86076546e+01  1.11749863e+02  3.99440125e+02
 8.52574921e+01 -1.56480026e+02  3.16373920e+01 -2.88674774e+02
 2.19867126e+02 -1.43932194e-01 -1.34706512e+02 -4.42049446e+01
-5.16368958e+02 -4.01468719e+02 -1.57929735e+01 -7.87631073e+01
-2.35952129e+01 -3.11173279e+02 -2.82017670e+02 -2.35689102e+02
 6.59882660e+01  5.76720352e+01 -3.30829071e+02  3.14159271e+02
 2.76695923e+02  1.42606125e+02 -2.82062439e+02  5.24797058e+01
 2.20414124e+02  5.19708748e+01  2.34885269e+02 -2.16377869e+02
-1.40891846e+02  1.41990891e+02  5.13254890e+01  4.14447968e+02
 1.22108025e+02  3.51192749e+02 -1.98787155e+02 -9.73346405e+01
-4.76022858e+02 -1.09480896e+02  2.73834717e+02  2.82844147e+02
 1.53335678e+02 -5.70043526e+01 -2.66108551e+02  5.58166046e+01
-2.03755936e+02 -6.96824646e+01 -1.64004478e+01  3.22296478e+02
-3.60644264e+01 -2.67300171e+02 -9.22670135e+01  4.36972046e+02
-1.66858795e+02 -2.35693100e+02  1.96473999e+02 -1.31496475e+02
-7.87603989e+01  3.91626167e+01  1.76897934e+02  3.29141785e+02
-8.90038605e+01  2.07620926e+02 -4.34666328e+01  1.73867477e+02
-1.18086487e+02 -5.46445312e+01 -2.51534561e+02 -2.72322357e+02
 4.25370544e+02 -2.18547531e+02  2.07546082e+02  3.09714489e+01
 2.16760147e+02 -1.92576950e+02  3.82441010e+02  3.20556580e+02
-1.70871490e+02  2.67490387e+01  2.33307617e+02 -2.59992340e+02
 2.69182800e+02  1.23228378e+02  1.15091011e+02 -1.23691216e+02
-2.41989250e+01  5.40787781e+02  1.13809113e+02  9.83203292e-01
-1.50544876e+02  1.33927124e+02  1.69518936e+02 -8.49747467e+01
-1.52804550e+02 -6.02077675e+00  4.61853912e+02  8.87366714e+01
 1.76044968e+02 -2.34203384e+02  1.39945694e+02 -3.51058228e+02
 2.71525940e+02  1.46038040e+02  1.27505188e+02  5.57647095e+02
-2.79661388e+01  2.78549561e+02  3.82018921e+02  1.01480608e+01
-2.05867020e+02 -1.14876167e+02 -2.10400986e+02  3.80276489e+02
 7.49855852e+00  4.38317261e+02  5.14286987e+02  2.58031445e+01
-2.55590546e+02 -8.32654724e+01 -1.93136658e+02  5.65613747e+01
-1.86433365e+02 -2.33817673e+02 -1.49063614e+02  2.22729584e+02
-8.95688629e+01  1.79967819e+02  3.85295776e+02  3.56979095e+02
 2.67097290e+02 -9.04998627e+01  8.04376297e+01  4.22617554e+02
 2.74027100e+02  3.17906708e+02  2.80316650e+02  6.92438660e+01
-1.62750076e+02  5.61574593e+01 -3.17479431e+02  1.98029957e+01
-1.67084000e+02 -1.13696587e+02  3.82618332e+01  1.64811386e+02
 4.53124268e+02 -1.85668350e+02  4.46261383e+02 -2.70037720e+02
 1.26649139e+02 -2.29229752e+02  2.39901901e+02  1.47904800e+02
-1.97947598e+01  1.38837387e+02  7.74572296e+01 -7.03947144e+01
 1.34667892e+02 -8.03233814e+00 -1.42542720e+00 -8.98536224e+01
-9.04409180e+01 -5.96350250e+01  3.39462646e+02  2.08550888e+02
-1.99777222e+02 -1.60533264e+02 -3.24461731e+02  3.01912479e+01
 3.62676331e+02  9.70280838e+01 -1.11973396e+02  6.08205627e+02
-1.31579498e+02  1.83665604e+02 -6.94832611e+01 -6.14844299e+02
 1.55449553e+01 -1.08869505e+01 -3.57214432e+01  5.25100708e+02
 4.08202171e+01  5.82328064e+02  2.10350739e+02  4.88474579e+01
-3.52919250e+02 -5.01865112e+02 -1.47032074e+02  2.58156128e+02
-1.19691162e+02  4.81695099e+02 -7.22056427e+01 -4.93056152e+02
 1.67156403e+02  4.17251923e+02  9.80329056e+01  2.93220520e+02
 4.14130157e+02  3.52702103e+01  3.97228508e+01 -4.75235672e+01
```

```
 1.94828506e+02  1.93870605e+02  4.57591492e+02  1.92034454e+02
-2.53381241e+02  3.41243713e+02  3.57524514e+00  2.26270723e+00
 2.85419128e+02 -4.48739563e+02 -8.62069931e+01 -1.52528091e+02
-1.15197495e+02  1.57195084e+02  7.28988495e+01 -9.24676285e+01
 2.13785431e+02 -3.09022430e+02  4.84462402e+02 -1.74942261e+02
-9.31685181e+01  3.87925079e+02  5.87947571e+02 -2.49364243e+02
 1.87842056e+02  1.50432999e+02  1.03977554e+02  4.09286560e+02
-1.69234695e+02 -1.34428589e+02  1.80724350e+02 -1.06841560e+02
 2.63050934e+02  3.62415802e+02 -1.37111374e+02  3.18874969e+02
-1.59398621e+02  2.68952423e+02 -1.17826796e+01 -1.02830635e+02
-1.26931770e+02 -4.01814331e+02  5.68547440e+01  8.36395447e+02
 1.14854912e+02  4.02169189e+02 -5.08348312e+01  2.65909607e+02
 1.60674606e+02  2.97231079e+02 -1.95677280e+01  3.79757721e+02
-2.22827255e+02  4.50861481e+02  2.20106903e+02 -3.19501892e+02
 4.72188759e+01 -2.61385174e+01  2.36070221e+02 -3.99794006e+01
-3.15240021e+02  1.44287094e+02 -1.09629555e+02  3.81374023e+02
 1.62381104e+02  2.59946564e+02  6.66903992e+01  2.09785034e+02
 3.40561493e+02 -1.06134476e+02  1.29394882e+02 -4.17344971e+02
 4.29281403e+02 -1.07562149e+02  2.77018005e+02 -1.73928970e+02
 1.34915848e+02  5.81688721e+02  7.21473217e+00 -1.17092752e+01
-4.79257843e+02  9.09435196e+01  6.48127319e+02  5.73984194e+00
 4.93044800e+02 -2.64046112e+02  3.99209442e+01  5.37747131e+02
 3.79929138e+02  1.36688400e+02  2.11437851e+02 -5.81138878e+01
 1.89908524e+02 -3.46671478e+02  1.12633751e+02  1.21254745e+02
 1.54147062e+01 -4.61321045e+02 -8.84481659e+01 -7.94591675e+01
-1.42018478e+02  3.58529480e+02 -3.16195404e+02  1.08165482e+02
 3.05151062e+01  8.56157532e+01 -3.07435913e+02 -4.65777191e+02
-2.58146400e+01 -1.81227142e+02 -2.01680145e+02  4.04095795e+02
-3.28097260e+02 -4.91827278e+01  3.13609985e+02  3.55171318e+01
 2.02527237e+02  3.08784882e+02  2.58618469e+02 -6.59095001e+01
 2.60918152e+02 -7.48920517e+01 -4.39153107e+02 -2.42540405e+02
 7.10074997e+01 -8.52475815e+01  4.98783478e+02  2.24005051e+01
 2.65708344e+02 -2.99967899e+01  1.77140076e+02 -3.67579132e+02
-2.54436996e+02 -4.21419403e+02  3.42242371e+02  1.08212296e+02
 4.45227661e+02  8.19966660e+01  1.43599564e+02  5.29647644e+02
 3.42527771e+02  3.34619476e+02 -2.31923676e+02  4.23526672e+02
-3.23448456e+02 -6.63926849e+01  6.59023361e+01 -1.39483429e+02
 4.85879913e+02 -2.97710754e+02  1.87722565e+02  1.97440567e+01
 6.48354919e+02  9.41612244e+01  6.38386917e+01 -2.26281021e+02
-3.35277710e+01  2.32568359e+02 -2.86349152e+02  3.18571045e+02
-1.99322174e+02  9.51764870e+00  2.17302078e+02 -4.23065369e+02
-5.32174133e+02  1.81637009e+02  2.53151810e+02  7.40239639e+01
 5.90716614e+02 -7.98938599e+01  1.46577103e+02 -8.70761719e+01
-8.39269485e+01  1.61078369e+02 -1.71059738e+02  3.12688141e+02
-2.07870483e+02  2.48158813e+02  5.27548767e+02 -1.98467846e+01
 2.60589752e+01 -1.18094940e+02  2.30247284e+02  3.73211853e+02
 4.23583893e+02 -2.45528885e+02  2.69668976e+02  2.19240860e+02
-5.17485161e+01 -2.22934128e+02  3.95064026e+02  4.75809570e+02
-8.24545898e+01 -1.25956383e+02  3.29276581e+02  3.47392822e+02
 2.66908051e+02  1.75846024e+01 -3.48423004e+02 -1.63219988e+00
-3.24979736e+02  2.98501343e+02  2.89077515e+02 -7.48175201e+01
-1.51922646e+01  2.19221100e+02 -4.47842743e+02  1.60354828e+02
-2.44623428e+02  6.56599976e+02 -7.44498367e+01 -1.64486038e+02
```

```
 -2.58506927e+02 -9.22066727e+01  1.17917816e+02 -3.45499611e+01
  1.31600449e+02  2.35300323e+02  8.36690292e+01  3.53352264e+02
 -4.15299683e+01  1.02926201e+02  1.53873016e+02  7.39747803e+02
 -1.41816010e+02  9.65118790e+01  7.36327667e+01  1.42206497e+02
 -5.24874695e+02  1.53826569e+02  1.10454340e+01 -2.21820114e+02
 -2.82684235e+02  5.46534485e+02  3.62710419e+01 -2.66107635e+02
  2.67081070e+00  1.00595284e+02  9.65550613e+01  2.14509995e+02
 -4.14353577e+02  2.35213608e+02  2.95628265e+02  5.02472412e+02
 -3.15958527e+02 -1.47173798e+02 -6.28636551e+01 -1.54001373e+02
  1.01229065e+02 -2.97219421e+02  9.64085815e+02  9.42380066e+01
 -8.30068207e+01  4.75489929e+02  1.73827225e+02 -1.36914474e+02
 -2.09416485e+01 -8.32644806e+01 -7.43884048e+01  2.20216293e+02
 -2.27273445e+01 -1.59196594e+02  2.25952625e+01  9.94817123e+01
 -3.05936310e+02  5.40522095e+02  2.87532196e+02 -1.29339813e+02
  3.08743835e+02  1.80305161e+02 -4.02803116e+01  9.69655838e+01
  2.50822433e+02  1.05025131e+02  4.55781860e+02 -3.93939575e+02
  2.14109207e+02  8.96003418e+01  1.68313095e+02  1.57637253e+02
 -2.43591400e+02 -1.18130875e+02 -5.90673981e+01  4.18340240e+02
  8.15387207e+02  6.24700661e+01  3.56615723e+02 -3.21553955e+01
 -1.96571388e+01  4.71809723e+02  7.49289932e+01  3.65674866e+02
  3.19777412e+01 -1.15966148e+01  3.60463440e+02  1.86177078e+02
 -1.46428818e+02  2.53124863e+02 -3.65473145e+02 -1.64330032e+02
  4.15236511e+02 -7.35705032e+01 -2.73659760e+02  7.94433899e+02
 -1.03539368e+02 -5.69932190e+02  6.27804852e+00 -1.13943436e+02
  1.19512863e+02  2.39206131e+02  3.46222801e+01  2.62871765e+02
  2.88392487e+02 -7.85758495e+00  4.39423492e+02 -3.80641670e+01
  2.41162399e+02  1.66575134e+02  3.69177094e+02 -1.81681274e+02
 -3.16552486e+01 -8.45768034e-01 -4.67535019e+01  9.68597260e+01
  4.81453918e+02  7.45757446e+01  7.45910568e+01  8.67211990e+01
  6.90543842e+00  1.30213791e+02 -1.52139572e+02 -5.82678223e+01
  4.49330383e+02  2.88994354e+02 -3.82640289e+02 -1.33419693e+02
 -2.09013794e+02  7.56396637e+01  1.46741959e+02 -2.90464668e+01
 -5.30035950e+02 -1.21185860e+02 -2.32903336e+02 -2.41526093e+02
 -1.83399673e+02 -4.87181030e+02 -5.19312622e+02 -1.94890350e+02
 -1.30098557e+02  2.05366882e+02  2.48577972e+02 -2.53232712e+02
  1.67371063e+01 -3.88452881e+02 -1.79087494e+02 -1.42386993e+02
  3.45091820e+01  1.33436264e+02  1.76280319e+02 -5.33443726e+02
  8.65622406e+01  6.50222244e+01  2.28327999e+01  1.19450218e+02
 -7.07207794e+01 -8.01078033e+01 -1.44556503e+02 -3.94198227e+01
 -1.81457840e+02  1.56317871e+02  2.20383347e+02 -3.01924103e+02
 -3.09929321e+02 -5.00823944e+02  4.46443253e+01 -2.43984314e+02
 -2.24061279e+01 -6.29932922e+02 -4.08174286e+02 -4.01408417e+02
 -2.69937988e+02  3.17821716e+02  9.24797745e+01  1.37188431e+02
 -1.95207840e+02 -6.18146362e+01 -4.06134827e+02  1.80980316e+02
 -3.21321167e+02 -5.16797295e+01 -9.13946152e+00  1.15476646e+02
 -4.39731049e+02 -1.79058517e+02  1.94967331e+02 -5.70143089e+01
  2.84124237e+02 -5.19332047e+01 -3.64969635e+02  1.75950211e+02
 -3.89129456e+02  3.36298065e+01 -2.52807846e+02 -3.09167664e+02
 -1.47562958e+02 -4.39925537e+02 -1.87984955e+02 -3.16677673e+02
 -9.73731308e+01 -4.17394531e+02  8.80572357e+01 -2.53235886e+02]]
shape: (1, 1000)
```

In [110…

```
from os import listdir
from os.path import isfile, join
import os


mypath = join(os.getcwd(),'train')

cats_imgs = [join(mypath,f) for f in listdir(mypath) if f.startswith('cat')]
dogs_imgs = [join(mypath,f) for f in listdir(mypath) if f.startswith('dog')]
```

In [111…

```
print("cats: {} and dogs: {}".format(len(cats_imgs),len(dogs_imgs)))
```

```
cats: 12500 and dogs: 12500
```

In [112…

```
Nmax = 100
cats_features = [get_features(get_image(img)).ravel() for img in cats_imgs[:N
dogs_features = [get_features(get_image(img)).ravel() for img in dogs_imgs[:N
```

In [113…

```
Y_cats = np.array(Nmax * [1])
Y_dogs = np.array(Nmax * [0])
```

In [114…

```
X_cvd = np.vstack([cats_features,dogs_features])
Y_cvd = np.vstack([Y_cats,Y_dogs]).ravel()
```

In [115…

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_cvd, Y_cvd, random_stat
```

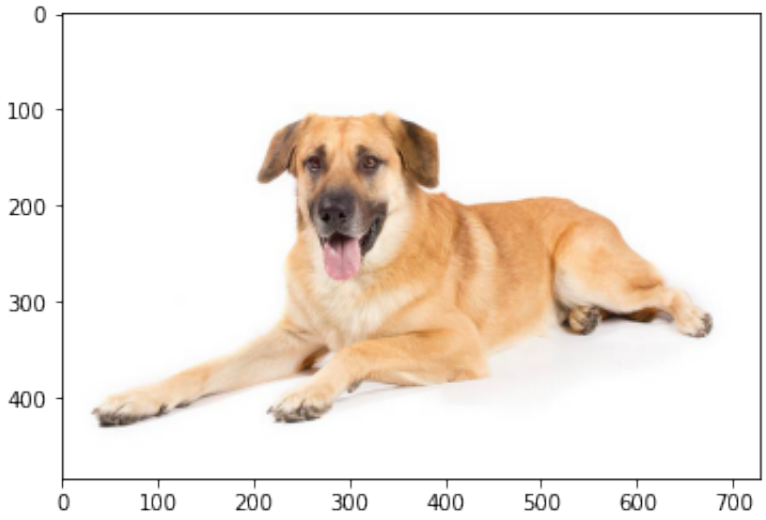
In [116…

```
from sklearn.linear_model import LogisticRegression

lg = LogisticRegression().fit(X_train, y_train)

print("Test set score: {:.2f}".format(lg.score(X_test, y_test)))
```

```
Test set score: 0.54
/opt/conda/lib/python3.7/site-packages/sklearn/linear_model/_logistic.py:940:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regress
ion
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

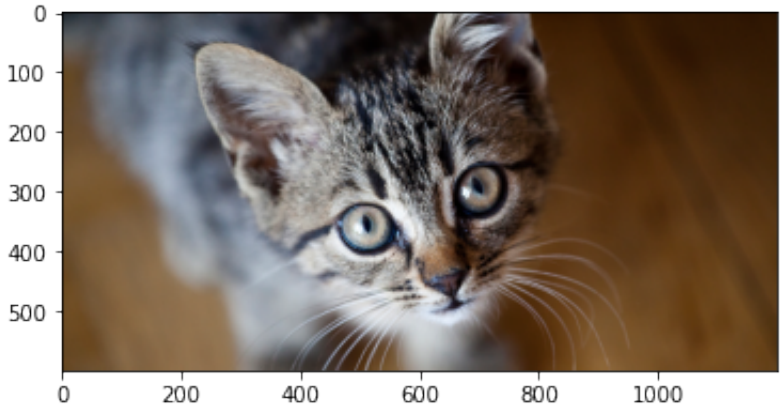
In [117…

```
import matplotlib.image as mpimg

dog_test_path = join(os.getcwd(),'Chinook-On-White-03.jpg')
cat_test_path = join(os.getcwd(),'Thinking-of-getting-a-cat.png')

img = mpimg.imread(dog_test_path)
imgplot = plt.imshow(img)
plt.show()
```



In [118…

```
img = mpimg.imread(cat_test_path)
imgplot = plt.imshow(img)
plt.show()
```



In [119…

```
features_out = get_features(get_image(cat_test_path))

prob = lg.predict_proba(features_out)
pred = lg.predict(features_out)

if pred[0] == 1:
    fpred = 'cat'
else:
    fpred = 'dog'

print("prob: {} and prediction: {}".format(prob,fpred))
```

```
prob: [[0.74842734 0.25157266]] and prediction: dog
```

Different networks apparently produced different results

In [ ]: