Project_Team1

May 16, 2018

Team Members: Anurag Marwah (Net ID: am8482)

1 Initialize

```
In [1]: import keras
```

```
/Applications/anaconda/lib/python3.6/site-packages/h5py/__init__.py:34: FutureWarning: Convers
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
```

```
In [2]: from scipy import misc
        import time
        import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
        %matplotlib inline
```

```
In [3]: from keras.applications.vgg16 import VGG16
        from keras.preprocessing import image
        from keras.applications.vgg16 import decode_predictions, preprocess_input
```

2 Download Model

```
In [4]: start_time = time.time()
        # Download pre-trained convolutional neural net
        model = VGG16(weights='imagenet')
        elapsed_time = time.time() - start_time
        print('Time to download: {0:.2f} seconds'.format(elapsed_time))
```

```
Time to download: 3.49 seconds
```

```
In [5]: model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
```

```
=================================================================
input_1 (InputLayer)         (None, 224, 224, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
fc1 (Dense)                  (None, 4096)              102764544
_________________________________________________________________
fc2 (Dense)                  (None, 4096)              16781312
_________________________________________________________________
predictions (Dense)          (None, 1000)              4097000
=================================================================
Total params: 138,357,544
```

```
Trainable params: 138,357,544
Non-trainable params: 0
_________________________________________________________________
```

```
In [6]: model.input_shape
```

```
Out[6]: (None, 224, 224, 3)
```

3 Load Test Data

```
In [7]: batch_size = 10
        nrow = model.input_shape[1]
        ncol = model.input_shape[2]
        nchan = model.input_shape[3]
        image_parts = 9
        new_batch_shape = (image_parts,nrow,ncol,nchan)
```

3.1 Load Test Images

```
In [8]: batch_size = 10
        class_name = 'testimage'
        batch_shape1 = (batch_size,nrow,ncol,nchan)
        batch_shape2 = (batch_size,2*nrow,2*ncol,nchan)
        batch_shape3 = (batch_size,2*nrow,ncol,nchan)
        batch_shape4 = (batch_size,nrow,2*ncol,nchan)
        x = np.zeros(batch_shape1)
        x2 = np.zeros(batch_shape2)
        x3 = np.zeros(batch_shape3)
        x4 = np.zeros(batch_shape4)
        for i in range(batch_size):
            fn = '{0:s}/{1:s}_{2:1d}.jpg'.format(class_name,class_name, i)
            x[i,:,:,:] = image.load_img(fn, target_size=(nrow, ncol))
            x2[i,:,:,:] = image.load_img(fn, target_size=(2*nrow, 2*ncol))
            x3[i,:,:,:] = image.load_img(fn, target_size=(2*nrow, ncol))
            x4[i,:,:,:] = image.load_img(fn, target_size=(nrow, 2*ncol))
```

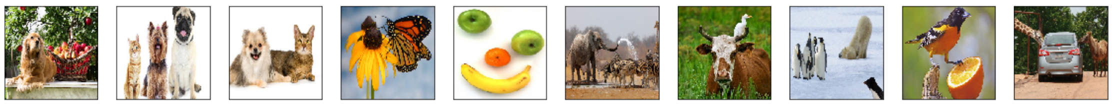
3.2 Display Test Images

```
In [9]: # Display the image
        def disp_image(im):
            if (len(im.shape) == 2):
                # Gray scale image
                plt.imshow(im, cmap='gray')
            else:
                # Color image.
                im1 = (im-np.min(im))/(np.max(im)-np.min(im))*255
                im1 = im1.astype(np.uint8)
```

```
            plt.imshow(im1)

        # Remove axis ticks
        plt.xticks([])
        plt.yticks([])
```

```
In [10]: plt.figure(figsize=(20,20))
         nplot = 10
         for i in range(nplot):
             plt.subplot(1,nplot,i+1)
             disp_image(x[i,:,:,:])
```



4 Process the Test Data and Classify

4.1 Break Image into Parts

```
In [11]: def breakimage(x,x2,x3,x4):
             t = np.zeros(new_batch_shape)
             t[0,:,:,:] = x[:,:,:]
             t[1,:,:,:] = x2[0:224,0:224,:] # Both-split 1
             t[2,:,:,:] = x2[0:224,224:448,:] # Both-split 2
             t[3,:,:,:] = x2[224:448,0:224,:] # Both-split 3
             t[4,:,:,:] = x2[224:448,224:448,:] # Both-split 4
             t[5,:,:,:] = x3[0:224,:,:] # Hori-split 1
             t[6,:,:,:] = x3[224:448,:,:] # Hori-split 2
             t[7,:,:,:] = x4[:,0:224,:] # Vert-split 1
             t[8,:,:,:] = x4[:,224:448,:] # Vert-split 2
             plt.figure(figsize=(20,20))

             # Display sub-images
             nplot = 9
             for i in range(nplot):
                 plt.subplot(1,nplot,i+1)
                 disp_image(t[i,:,:,:])

             return t
```

4.2 Classify Test Images

```
In [12]: def classifyimage(t):
             t2 = preprocess_input(t)
```

```
        preds = model.predict(t2)
        preds_decoded = decode_predictions(preds, top=3)
        ntop = 3
        res_dict = {}
        for i in range(ntop):
            class_name = []
            class_prob = []
            for j in range(9):
                class_name.append(preds_decoded[j][i][1])
                class_prob.append(preds_decoded[j][i][2])

            name_col = str('class %d' % i)
            prob_col = str('prob %d' % i)
            res_dict[name_col] = class_name
            res_dict[prob_col] = class_prob
        return res_dict
```

```
In [13]: def multiclassimages(x,x2,x3,x4,i):
             t2 = np.zeros(new_batch_shape)
             t2 = breakimage(x[i,:,:,:],x2[i,:,:,:],x3[i,:,:,:],x4[i,:,:,:])
             df = pd.DataFrame(data=classifyimage(t2))
             print(df)
```

```
In [14]: start_time = time.time()
         for i in range(batch_size):
             multiclassimages(x,x2,x3,x4,i)
         elapsed_time = time.time() - start_time
         print('Total Run-Time: {0:.2f} seconds'.format(elapsed_time))
```

```
           class 0          class 1         class 2    prob 0    prob 1  \
0   cocker_spaniel  golden_retriever    Afghan_hound  0.766667  0.057201
1   cocker_spaniel  golden_retriever  Sussex_spaniel  0.977880  0.007292
2              hip       pomegranate             fig  0.801726  0.104667
3 golden_retriever      Afghan_hound  cocker_spaniel  0.353880  0.171139
4  shopping_basket            hamper   shopping_cart  0.416912  0.186214
5             cock      picket_fence             hen  0.225569  0.205054
6     butcher_shop              cock            mask  0.052486  0.047130
7 golden_retriever    cocker_spaniel    Irish_setter  0.536958  0.401182
8      pomegranate   shopping_basket          orange  0.359964  0.276533

     prob 2
0  0.044101
1  0.004758
2  0.021298
3  0.165409
4  0.050461
5  0.139648
6  0.037781
```

```
7  0.009304
8  0.040549
             class 0              class 1          class 2    prob 0  \
0             briard           Great_Dane  German_shepherd  0.124858
1  Australian_terrier   Yorkshire_terrier    silky_terrier  0.651461
2        bull_mastiff                 pug    Saint_Bernard  0.604124
3          otterhound            Airedale     Irish_setter  0.800999
4              Saluki    golden_retriever  German_shepherd  0.231490
5        Afghan_hound              briard       Great_Dane  0.714725
6                 ear            fur_coat             corn  0.214995
7   Yorkshire_terrier  Australian_terrier    silky_terrier  0.576006
8                 pug   Brabancon_griffon    silky_terrier  0.345285

     prob 1    prob 2
0  0.104123  0.085896
1  0.209994  0.055904
2  0.197983  0.035861
3  0.046910  0.023467
4  0.225457  0.080568
5  0.094506  0.054567
6  0.121811  0.064957
7  0.137183  0.103789
8  0.085346  0.069883
       class 0            class 1             class 2    prob 0    prob 1  \
0   Pomeranian        Persian_cat               tabby  0.139771  0.137405
1   Pomeranian           Pekinese           Chihuahua  0.757135  0.150206
2  Egyptian_cat             tabby           tiger_cat  0.676974  0.157537
3        collie  Shetland_sheepdog            keeshond  0.712127  0.244613
4         tabby       Egyptian_cat           tiger_cat  0.592123  0.212999
5        collie    German_shepherd  Australian_terrier  0.188641  0.087264
6      fur_coat             collie         Afghan_hound  0.349927  0.244921
7   Pomeranian           papillon           Chihuahua  0.901075  0.056454
8         tabby          tiger_cat        Egyptian_cat  0.672458  0.192942

     prob 2
0  0.115727
1  0.064621
2  0.099030
3  0.012101
4  0.068172
5  0.063730
6  0.182800
7  0.018398
8  0.106204
             class 0            class 1     class 2    prob 0    prob 1  \
0            monarch                bee     admiral  0.993158  0.001236
1  long-horned_beetle               bee      weevil  0.197424  0.166230
2            monarch  sulphur_butterfly     ringlet  0.999759  0.000139
```

```
3             macaw              daisy      sarong  0.464942  0.096552
4           monarch  sulphur_butterfly  table_lamp  0.996413  0.001137
5           monarch            admiral     limpkin  0.992350  0.001119
6             macaw            monarch         bee  0.643674  0.231550
7           monarch              daisy     admiral  0.185346  0.146710
8           monarch  sulphur_butterfly     ringlet  0.999782  0.000109

     prob 2
0  0.001193
1  0.124302
2  0.000062
3  0.047185
4  0.000358
5  0.000787
6  0.013176
7  0.134548
8  0.000063
          class 0   class 1   class 2    prob 0    prob 1    prob 2
0    Granny_Smith    banana    orange  0.394660  0.344958  0.126663
1    Granny_Smith    orange  cucumber  0.947602  0.021980  0.005418
2    Granny_Smith       fig    banana  0.977338  0.004895  0.002266
3  butternut_squash  banana    orange  0.316133  0.312474  0.140373
4          banana    orange     lemon  0.595190  0.138463  0.056897
5    Granny_Smith  cucumber    banana  0.997390  0.001146  0.000371
6          orange    banana     lemon  0.392543  0.264543  0.127899
7          banana  cucumber     plate  0.257921  0.111448  0.108399
8          banana  cucumber  zucchini  0.843559  0.084707  0.015285
          class 0           class 1          class 2    prob 0    prob 1  \
0           zebra     Arabian_camel           impala  0.998178  0.000801
1  Indian_elephant  African_elephant           tusker  0.472317  0.343056
2         ostrich        hartebeest           impala  0.123605  0.088232
3           zebra  African_elephant  Indian_elephant  0.551994  0.269120
4           zebra           gazelle           impala  0.998537  0.000663
5        fountain            geyser          vulture  0.659084  0.066207
6           zebra            impala       hartebeest  0.979836  0.010332
7  African_elephant          tusker  Indian_elephant  0.959733  0.022562
8           zebra           gazelle           impala  0.999862  0.000066

     prob 2
0  0.000368
1  0.170328
2  0.084587
3  0.066150
4  0.000587
5  0.043095
6  0.002884
7  0.017586
8  0.000057
```

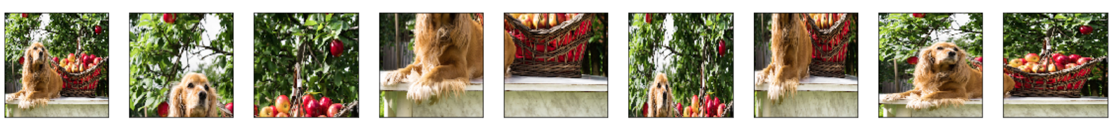
	class 0	class 1	class 2	prob 0	prob 1	prob 2
0	ox	water_buffalo	hartebeest	0.682823	0.080805	0.060105
1	warthog	bison	ox	0.211472	0.169261	0.138061
2	crane	bustard	ostrich	0.311691	0.204712	0.077140
3	vizsla	German_short-haired_pointer	dhole	0.426268	0.256385	0.084507
4	basset	ox	gazelle	0.264228	0.202332	0.089057
5	vulture	English_foxhound	crane	0.240324	0.144426	0.103000
6	vizsla	hartebeest	Rhodesian_ridgeback	0.513176	0.250404	0.047706
7	German_short-haired_pointer	vizsla	bolete	0.231073	0.153964	0.106356
8	bustard	water_ouzel	gazelle	0.379034	0.075323	0.066289

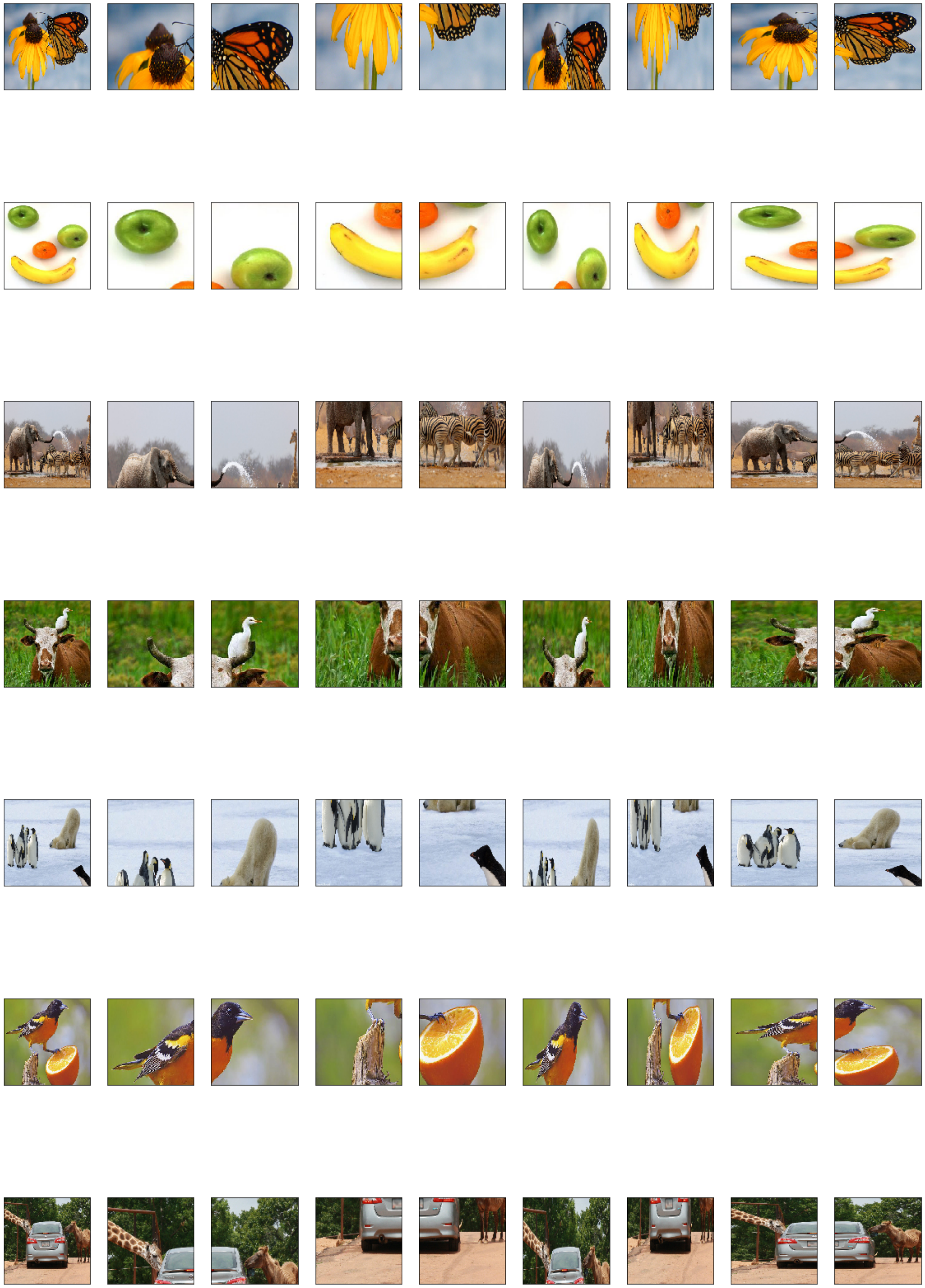
	class 0	class 1	class 2	prob 0	prob 1	prob 2
0	king_penguin	ice_bear	Siberian_husky	0.791439	0.189859	7.233626e-03
1	king_penguin	killer_whale	albatross	0.999991	0.000006	1.101749e-06
2	ice_bear	white_wolf	Arctic_fox	0.994363	0.004122	6.240166e-04
3	king_penguin	pelican	killer_whale	0.999997	0.000001	5.867903e-07
4	ice_bear	king_penguin	Arctic_fox	0.823620	0.100908	9.629729e-03
5	king_penguin	missile	picket_fence	0.611972	0.048322	4.469074e-02
6	king_penguin	dogsled	fur_coat	0.868800	0.049535	2.087563e-02
7	king_penguin	albatross	goose	0.999998	0.000002	1.356386e-07
8	ice_bear	Arctic_fox	white_wolf	0.998397	0.001123	2.344188e-04

	class 0	class 1	class 2	prob 0	prob 1	prob 2
0	robin	worm_fence	brambling	0.877616	0.058950	0.034699
1	goldfinch	brambling	robin	0.465590	0.284076	0.177918
2	robin	brambling	magpie	0.832907	0.065516	0.063223
3	macaw	goldfinch	jacamar	0.232381	0.118159	0.096808
4	orange	lemon	ant	0.952402	0.043735	0.000901

```
5      robin   brambling  worm_fence  0.721108  0.146345  0.037095
6        ant      orange  leafhopper  0.174069  0.151354  0.108228
7  goldfinch   brambling   bee_eater  0.734095  0.087138  0.045009
8      robin      magpie      bulbul  0.984657  0.006216  0.002758
           class 0              class 1              class 2    prob 0  \
0    Arabian_camel                llama                hyena  0.453107
1         golfcart              minivan          beach_wagon  0.048044
2    Irish_terrier           brown_bear         Irish_setter  0.187772
3          minibus  recreational_vehicle              minivan  0.334459
4           baboon                crane                hyena  0.076962
5          minibus                  ear              minivan  0.026365
6          minibus            ambulance  recreational_vehicle  0.510054
7          minivan              ostrich                racer  0.061428
8  golden_retriever             red_wolf                llama  0.104572

     prob 1    prob 2
0  0.134783  0.074857
1  0.045082  0.037180
2  0.111535  0.058502
3  0.301103  0.075911
4  0.053645  0.053610
5  0.019694  0.019678
6  0.092927  0.076799
7  0.057906  0.048974
8  0.065259  0.062134
Total Run-Time: 55.49 seconds
```





In [ ]: