

Let's go through a simple skin cancer classification exercise

We are going to :

1. Install Clarifai if it is not already installed: We can install clarifai with pip but we first need to get a developer key from them . pip install clarifai --upgrade
2. Go to <https://clarifai.com/developer/login/> (<https://clarifai.com/developer/login/>) to create a developer account and get an API key then run:.
3. Run the cells in this notebook to follow along:

```
In [5]: cd C:/RSAVED/Scripts
```

```
C:\RSAVED\Scripts
```

```
In [4]: # Let' s import all our libraries (also in underlying code)
import os,sys,re
import numpy as np
import pandas as pd
import pickle
import shutil
import clarifai
import matplotlib
from clarifai.rest import ClarifaiApp
from clarifai.rest import Image as CImage
import regex as re
import matplotlib.pyplot as plt
import urllib
import urllib2
import requests

from clarifai.rest import ClarifaiApp
from clarifai.rest import Image as CImage
```

```
In [6]: # let' s now import our custom code and initiate a model
from DeepMole import DeepMole
%matplotlib inline
M= DeepMole()
```

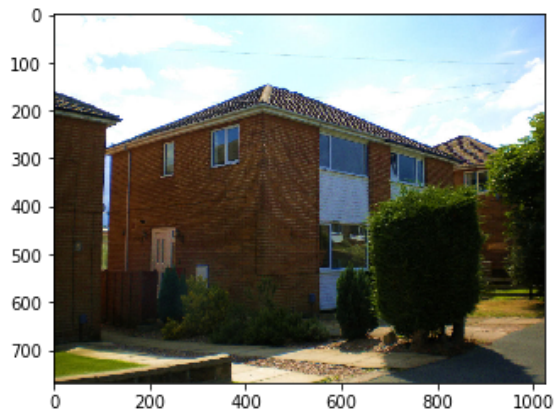
Let see how clever is clarifai's 'General model' :

To quickly assess clarifai let's classify 3 random images :

```
In [ ]: M.returntags(url='http://farm4.static.flickr.com/3075/32407
94796_0736730011.jpg')
```

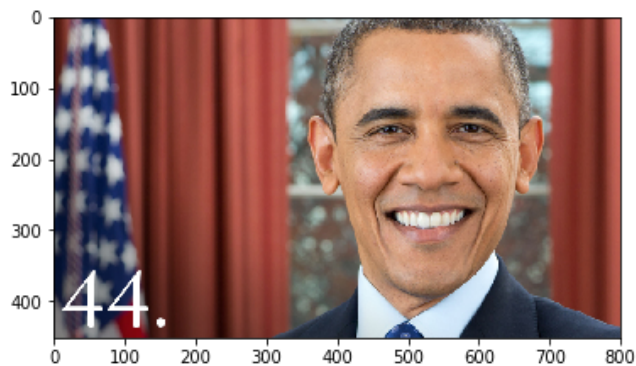
```
In [5]: M.returntags(url='https://i.ebayimg.com/00/s/NzY4WDEwMjQ=/z/poAAASwyWZZV44A/$_86.JPG')
```

```
home 0.9984274  
no person 0.9932965
```



```
In [6]: M.returntags(url="https://www.whitehouse.gov/sites/whitehouse.gov/files/images/first-family/44_barack_obama%5B1%5D.jpg")
```

```
people 0.98990667  
man 0.9887515
```



Let's add our custom skin images in order to retrain the model

In our case we will be using data from a skin cancer database

Clarifai's free account only let us upload 128 images so we will pull 64 images per class.

Each uploaded picture will be tagged with the correct label.

```
In [7]: M.writeimages(64)
```

['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010168.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0011067.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0011086.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010484.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0011177.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010467.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010919.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010585.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010263.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010399.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0009990.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010645.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010858.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010324.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0011065.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010513.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010206.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010183.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0011290.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010184.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010357.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0011046.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0009944.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010298.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010448.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0009978.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010920.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010658.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0011201.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0011309.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010337.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0010991.jpg
['Benign'] ['Malignant']
C:/data/CX0/ISIC-images/Benign/ISIC_0011248.jpg
['Benign'] ['Malignant']

Let's now train our custom model with the new added data :

This is where the transfer learning magic happens

1. First we will give a new name to this model
2. Then we will fine tune its weights
3. Last but not least we will use this new model on new data

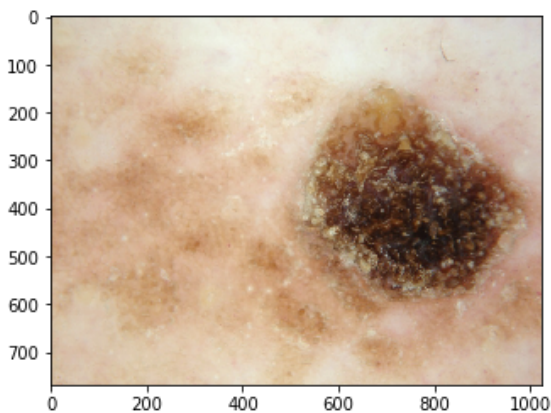
```
In [8]: name='cancer52'
```

```
In [9]: M.passimages(name)
```

```
done {'status': {'code': 10000, 'description': u'Ok'}, u
'model': {'name': u'cancer52', 'output_info': {'type_ext
': u'concept', 'message': u'Show output_info with: GET /mo
dels/{model_id}/output_info', 'type': u'concept', 'output
_config': {'closed_environment': False, 'concepts_mutuall
y_exclusive': False}}, u'created_at': u'2017-08-30T16:10:2
3.689742Z', u'app_id': u'dd68a42776274c1cb9ab40fa45f6b45d',
u'model_version': {'status': {'code': 21100, u'descriptio
n': u'Model trained successfully', u'created_at': u'2017-0
8-30T16:10:25.261532Z', u'id': u'a1b7a7ecbb7a42b5ac6138a951
eed351', u'total_input_count': 2739, u'active_concept_count
': 2}, u'id': u'cancer52'}}
```

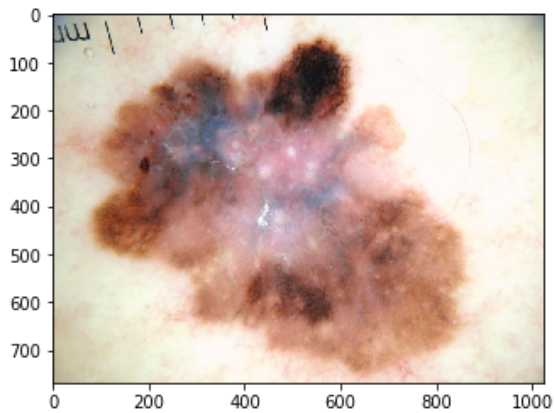
```
In [10]: M.check('C:/data/CX0/ISIC-images/Benign/ISIC_0009996.jpg', name)
```

```
Benign 0.8714169
Malignant 0.14289725
```



```
In [16]: M.check('C:/data/CX0/ISIC-images/Malignant/ISIC_0010392.jpg',name)
```

```
Malignant 0.99988186  
Benign 0.00010321584
```



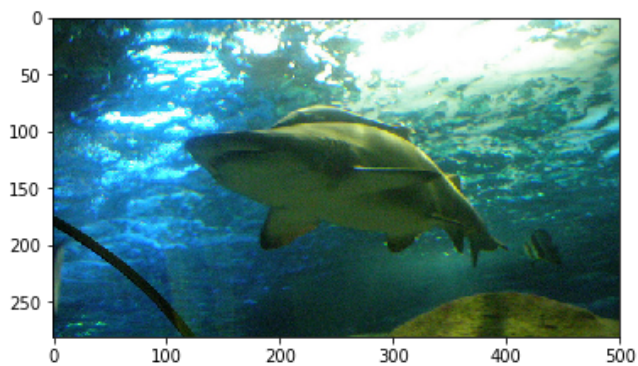
```
In [ ]:
```

Custom models are not perfect yet

Let 's see how our classifier is now doing on some general objects :

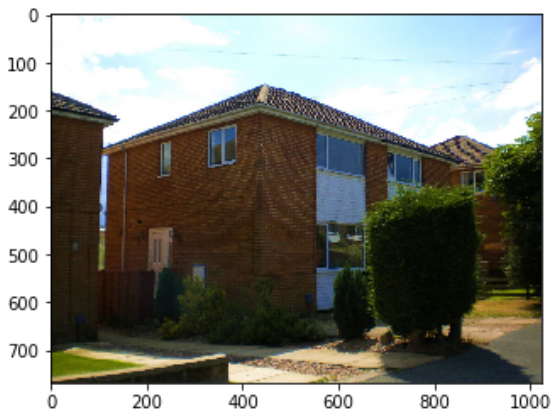
```
In [17]: M.returntags(url='http://farm4.static.flickr.com/3075/3240794796_0736730011.jpg')
```

```
Benign 0.0001892576  
Malignant 2.5283762e-09
```



```
In [18]: M.returntags(url='https://i.ebayimg.com/00/s/NzY4WDEwMjQ=/z/poAAAOSwyWZZV44A/$_86.JPG')
```

Malignant 0.9987207
Benign 4.447439e-07



Let's check what we use to get with general model :

Instead of using our custom model we will use the general embedding model proposed by Clarifai:

```
In [1]: M.check('C:/data/CX0/ISIC-images/Malignant/ISIC_0010392.jpg', "general-v1.3")
```

```
-----  
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-1-9f2bf1aefef6> in <module>()  
----> 1 M.check('C:/data/CX0/ISIC-images/Malignant/ISIC_0010392.jpg', "general-v1.3")  
NameError: name 'M' is not defined
```

That 's all for part one

Please send Q&A to: louis.vainqueur@bain.com

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []: