Python 3 Capstone Assignment

The Project

1. This is a project with minimal scaffolding. Expect to use the the discussion forums to gain insights! It's not cheating to ask others for opinions or perspectives!
2. Be inquisitive, try out new things.
3. Use the previous modules for insights into how to complete the functions! You'll have to combine Pillow, OpenCV, and Pytesseract
4. There are hints provided in Coursera, feel free to explore the hints if needed. Each hint provide progressively more details on how to solve the issue. This project is intended to be comprehensive and difficult if you do it without the hints.

The Assignment

Take a ZIP file) of images and process them, using a library built into python that you need to learn how to use. A ZIP file takes several different files and compresses them, thus saving space, into one single file. The files in the ZIP file we provide are newspaper images (like you saw in week 3). Your task is to write python code which allows one to search through the images looking for the occurrences of keywords and faces. E.g. if you search for "pizza" it will return a contact sheet of all of the faces which were located on the newspaper page which mentions "pizza". This will test your ability to learn a new (library), your ability to use OpenCV to detect faces, your ability to use tesseract to do optical character recognition, and your ability to use PIL to composite images together into contact sheets.

Each page of the newspapers is saved as a single PNG image in a file called images.zip. These newspapers are in english, and contain a variety of stories, advertisements and images. Note: This file is fairly large (~200 MB) and may take some time to work with, I would encourage you to use small_img.zip for testing.

Here's an example of the output expected. Using the small_img.zip file, if I search for the string "Christopher" I should see the following image:

If I were to use the images.zip file and search for "Mark" I should see the following image (note that there are times when there are no faces on a page, but a word is found!):

Note: That big file can take some time to process - for me it took nearly ten minutes! Use the small one for testing.

Let's Start Coding

In [1]:

```
import zipfile
import os
from PIL import Image
import pytesseract
import cv2 as cv
import numpy as np
```

Loading OpenCV Classifier for Face Detection

In [2]:

```
face_cascade = cv.CascadeClassifier('readonly/haarcascade_frontalface_default.xml')
```

Function to unZip a Zipfle into a giving directory

In [3]:

```
def unzipfiles(aZipFile,aDir):
    #Extract files from zipfile local path
    # Hint 1
```

```
    aZipRef = zipfile.ZipFile(aZipFile, 'r')
    aZipRef.extractall(aDir)
    aZipRef.close()
```

Unziping 'small_img.zip' file into 'small_img' directory

In [4]:

```
#Extracting images in small_img.zip file
imageDirectory = 'small_img'
imZipFile = 'readonly/small_img.zip'
unzipfiles(imZipFile, imageDirectory)
```

Creating a global Dictionary to store filenames, images and faces

In [5]:

```
# This will be a dictionary of lists indexed by filename
imagesDir = {}
# [filename][0] : a PIL Image File
# [filename][1] : the Text detecte in the image
# [filename][2] : faces in the image, top and bottom coordinates
```

A Function to load to load images from a directory and return a list with filenames loaded

In [6]:

```
def loadImages(aDirectory):
    # Hint 2
    aList = []
    aList = os.listdir(aDirectory)
    aList.sort()
    for aFile in aList:
        imagesDir[aFile] = [Image.open(aDirectory + '/' + aFile)]
    return aList
```

Let's update the 'filename_list' that needs to be updated

In [7]:

```
filename_list = [] # list of filenames
filename_list = loadImages(imageDirectory)
```

Let's search for a word in text and images if word is found

Only if the word is found, the image is analyzed to find faces, faces are 100x100 pixels and contact sheet row max images is 5

In [8]:

```
def searchWord(aWord):
    for fName in filename_list:
        aImg = imagesDir[fName][0]
        imagesDir[fName].append(pytesseract.image_to_string(aImg).replace('-\n',''))
        # Hint 3
        if aWord in imagesDir[fName][1]:
            print('Results found in file',fName)
            try:
                faces = (face_cascade.detectMultiScale(np.array(aImg),1.28,6)).tolist()
                imagesDir[fName].append(faces)
                facesInfile = []
                for x,y,w,h in imagesDir[fName][2]:
                    facesInfile.append(aImg.crop((x,y,x+w,y+h)))
                contact_sheet = Image.new(aImg.mode, (500,100*int(np.ceil(len(facesInfile)/5))))
                x = 0
                y = 0
```

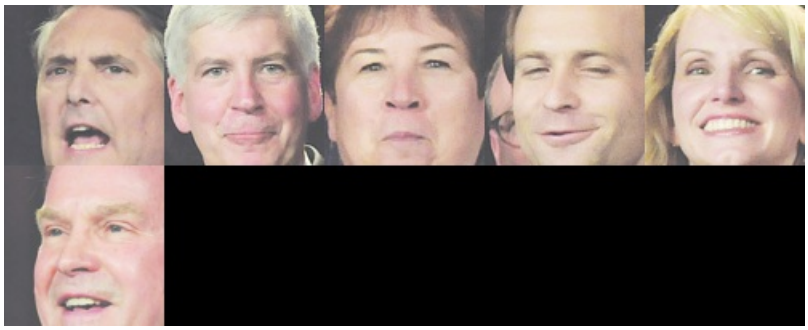
```
            y = 0
            for face in facesInfile:
                # Hint 4
                face.thumbnail((100,100))
                contact_sheet.paste(face, (x, y))
                if x+100 == contact_sheet.width:
                    x=0
                    y=y+100
                else:
                    x=x+100
            display(contact_sheet)
        except:
            print('But there were no faces in that file!')
```

Let's search for the name 'Christopher'

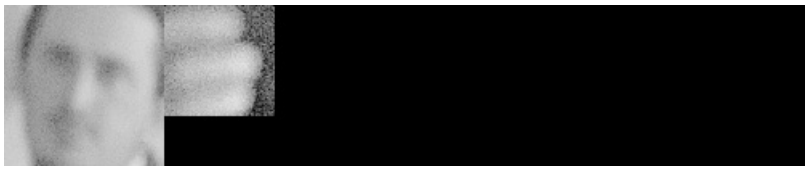
In [9]:

```
searchWord('Christopher')
```

```
Results found in file a-0.png
```



```
Results found in file a-3.png
```



Yes, first task done!

ANALYZING THE IMAGES IN IMAGES.ZIP FILE

Let's Unzip the files into images Folder

In [10]:

```
# Extracting images in images.zip file
imageDirectory = 'images'
imZipFile = 'readonly/images.zip'
unzipfiles(imZipFile, imageDirectory)
```

Let's get the new list of files to be analyzed

In [11]:

```
imagesDir = {} # dictionary of lists indexed with filenames
# [0] : PIL Image File
# [1] : Text in image
filename_list = [] # list of filenames
```

```
filename_list = loadImages(imageDirectory)
```

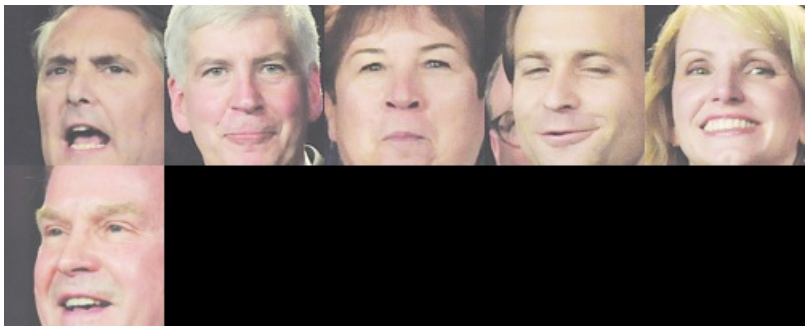
Let's search for the name 'Mark'

Notice how I found 1 more face than what the assignment required.

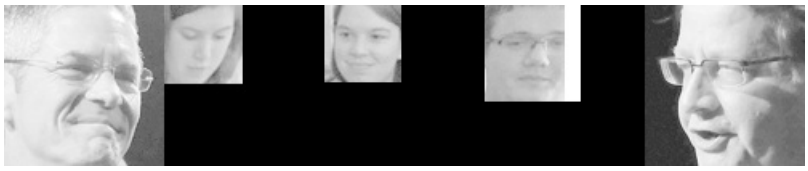
In [12]:

```
searchWord('Mark')
```

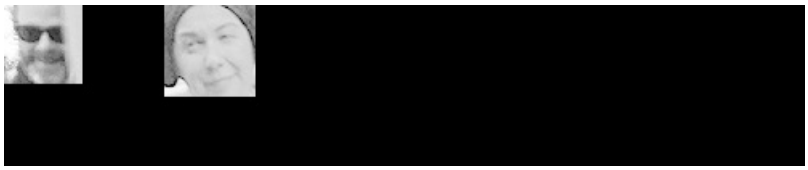
```
Results found in file a-0.png
```



```
Results found in file a-1.png
```

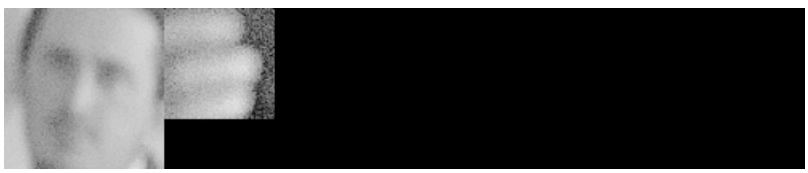


```
Results found in file a-10.png
But there were no faces in that file!
Results found in file a-13.png
```



```
Results found in file a-2.png
```

```
Results found in file a-3.png
```



```
Results found in file a-8.png
But there were no faces in that file!
```

You found one more face!

Yes I found one more face!

Now let's see if we can get even better results

By changing the paramaters of **'detectMultiScale'** we can find better settings for this excersice, but we need to have an application specification to understand which is the priority between speed and accuaracy. There are many faces that where not detected in the newspaper imgages, we can think speed is more important and having more false positives is something that can be handle in a second review or human review.

He I am going to analyze only the files that have the word **'Mark'** and try another setting to compare the results

Let's copy the searchWord function with a different name

I just to have a quick way to analyze again the faces in the images that have the word 'Mark'

In [14]:

```
def searchWord1(aWord):
    for fName in filename_list:
        aImg = imagesDir[fName][0]
        # imagesDir[fName].append(pytesseract.image_to_string(aImg).replace('-\n',''))
        imagesDir[fName].append(aWord)
        if aWord in imagesDir[fName][1]:
            print('Results found in file',fName)
            try:
                faces = (face_cascade.detectMultiScale(np.array(aImg),1.3,4)).tolist()
                imagesDir[fName].append(faces)
                facesInfile = []
                for x,y,w,h in imagesDir[fName][2]:
                    facesInfile.append(aImg.crop((x,y,x+w,y+h)))
                contact_sheet = Image.new(aImg.mode, (500,100*int(np.ceil(len(facesInfile)/5))))
                x = 0
                y = 0
                for face in facesInfile:
                    face.thumbnail((100,100))
                    contact_sheet.paste(face, (x, y))
                    if x+100 == contact_sheet.width:
                        x=0
                        y=y+100
                    else:
                        x=x+100
                display(contact_sheet)
            except:
                print('But there were no faces in that file!')
```

Lets get the files that have the word 'Mark' in the list to be analyzed

In [15]:

```
imagesDir= {} # dictionary of lists indexed with filenames
# [0] : PIL Image File
# [1] : Text in image
filename_list = [] # list of filenames
filename_list = loadImages(imageDirectory)
filename_list = filename_list [:8]
del filename_list[2]
del filename_list[2]
del filename_list[2]
```

In [16]:

```
print(filename_list)
```

```
['a-0.png', 'a-1.png', 'a-13.png', 'a-2.png', 'a-3.png']
```

In [17]:

```
searchWord1('Mark')
```

```
Results found in file a-0.png
```

```
Results found in file a-1.png
```

```
Results found in file a-13.png
```

```
Results found in file a-2.png
```

```
Results found in file a-3.png
```

Yes I found 3 more faces!