

Reading, writing and displaying images with OpenCV

Let's start by importing the OpenCV library

```
In [1]: # Press CTRL + ENTER to run this line
# You should see an * between the [ ] on the left
# OpenCV takes a couple seconds to import the first time

import cv2
```

```
In [2]: # Now Let's import numpy
# We use as np, so that everything we call on numpy, we can type np instead
# It's short and looks neater

import numpy as np
```

```
In [3]: # We don't need to do this again, but it's a good habit
import cv2

# Load an image using 'imread' specifying the path to image
input=cv2.imread(r"D:\Data Science with AI\face\open cv with haar cascade\peace")
```

```
In [4]: # Our file 'input.jpg' is now loaded and stored in python
# as a variable we named 'image'

# To display our image variable, we use 'imshow'
# The first parameter will be title shown on image window
# The second parameter is the image variable
cv2.imshow('Test my image',input)

# 'waitKey' allows us to input information when a image window is open
# By leaving it blank it just waits for anykey to be pressed before
# continuing. By placing numbers (except 0), we can specify a delay for
# how long you keep the window open (time is in milliseconds here)
cv2.waitKey()

# This closes all open windows
# Failure to place this will cause your program to hang
cv2.destroyAllWindows()
```

lets take a closer look at how images are stored

```
In [5]: #import numpy
import numpy as np
```

```
In [6]: print(input.shape)
```

```
(408, 612, 3)
```

shape gives the dimensions of the image array

the 2d dimensions are 830 pixels in high by 1245 pixels wide. the '3L' means that there are 3 other components (RGB) that make up this image

```
In [7]: # Lets print each dimensions of the image
```

```
print('Height of image:' ,int(input.shape[0]),'pixels')  
print('Width of image:' , int(input.shape[1]), 'pixels')
```

```
Height of image: 408 pixels
```

```
Width of image: 612 pixels
```

how do we save images we edit in opencv?

```
In [8]: # simply use 'imwrite' specifying the file name and the image to be saved
```

```
cv2.imwrite('output.jpg', input)  
cv2.imwrite('output.png', input)
```

```
Out[8]: True
```

```
In [ ]:
```