

U

LISBOA

UNIVERSIDADE
DE LISBOA

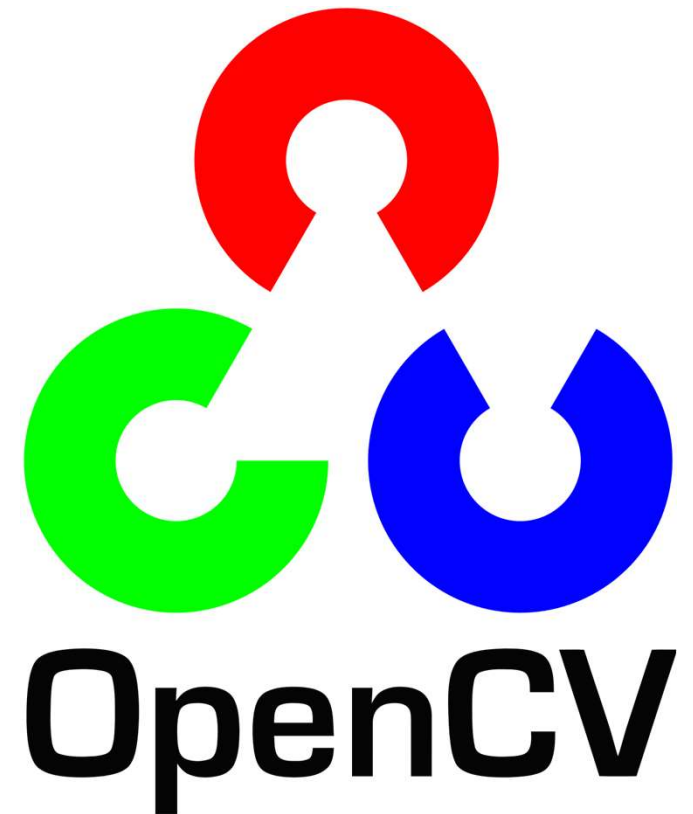
Carlos J. Costa

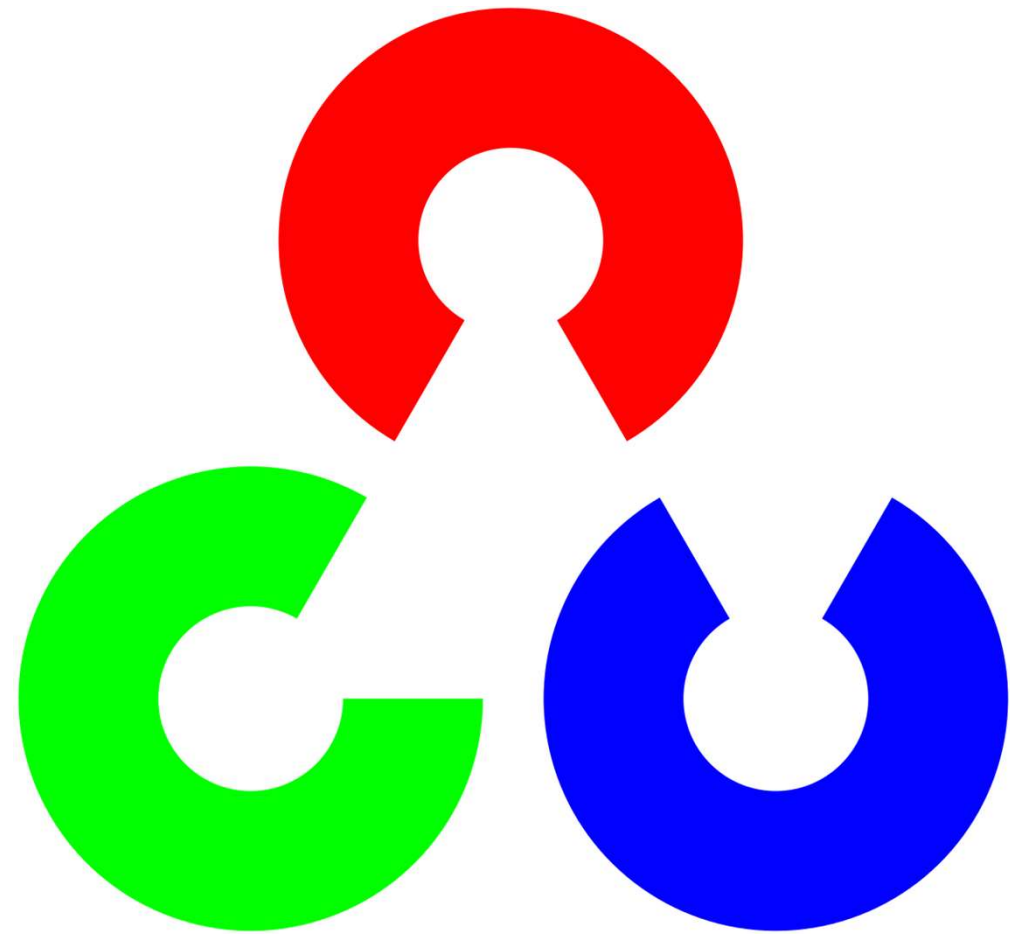
OPEN CV

2021

Agenda

- Open CV
- Read and save
- Image resize
- Split and Merge images
- Add Images
- Rotate
- ROI





OpenCV

1. Import Library

```
[2]: ▶ import cv2
```



2. Read, write and display images

```
[3]: ▶ # Read
my_image = cv2.imread('image01.png')
# Write
cv2.imwrite('image01a.png', my_image)
```

```
Out[3]: True
```

```
[4]: ▶ type(my_image)
```

```
Out[4]: numpy.ndarray
```

```
[5]: ▶ my_image.shape
```

```
Out[5]: (779, 1383, 3)
```

Read from URL



```
▶ import cv2
import urllib.request as req
import numpy as np

req = req.urlopen('https://carlosjcosta.files.wordpress.com/2015/02/cropped-fundo.jpg')
arr = np.asarray(bytearray(req.read()), dtype=np.uint8)
image_1 = cv2.imdecode(arr, -1)

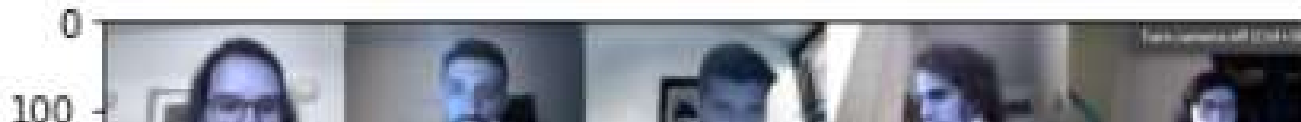
plt.imshow(image_1)
plt.show()
```





```
# Display image
cv2.imshow("my_image",my_image)
cv2.waitKey(0) # Wait for the user to press a key before continuing
cv2.destroyAllWindows()
```

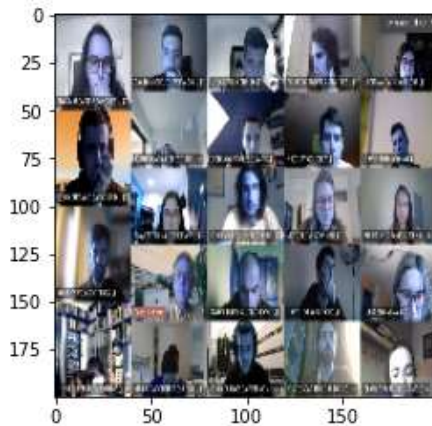
```
# Display images using matplotlib
import matplotlib.pyplot as plt
plt.imshow(my_image)
plt.show()
```



3. Resize image

```
[23]: ▶ # Resize image
# cv2.resize(src, dsize, fx, fy)
# dsize is a tuple
new_image = cv2.resize(my_image, (200,200))
```

```
[24]: ▶ plt.imshow(new_image)
plt.show()
```



```
[25]: ▶ # Resize image
new_image = cv2.resize(my_image, None, fx=0.7, fy=0.7)
```

```
[26]: ▶ # Scale up/expand both width and height by factor of 2
result_1 = cv2.resize(my_image, None, fx=2, fy=2, interpolation=cv2.INTER_CUBIC)

# Scale down/shrink both width and height by factor of 2
result_2 = cv2.resize(my_image, None, fx=2, fy=2, interpolation=cv2.INTER_AREA)
```



4. Conversion image

```
▶ # Converts an image from one color space to another.
# cv2.cvtColor(src, type)
# type:
# cv2.COLOR_BGR2GRAY
# cv2.COLOR_GRAY2BGR
# cv2.COLOR_BGR2RGB

type=cv2.COLOR_BGR2GRAY
new_image = cv2.cvtColor(my_image, type)
```

```
▶ # Display image again
cv2.imshow("new_image",new_image)
cv2.waitKey(0) # Wait for the user to press a key before continuing
cv2.destroyAllWindows()
```



Split and Merge Image Channels

```
8) import cv2
import numpy as np
img = cv2.imread('image01a.png')
b,g,r = cv2.split(img)
```

```
cv2.imshow("b_image",b)
cv2.waitKey(0)
```

8): -1

```
cv2.imshow("g_image",g)
cv2.waitKey(0)
```

9): -1

```
cv2.imshow("r_image",r)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

2): -1

```
imag=cv2.merge((b,r,g ))
```

```
cv2.imshow("merged_image",imag)
cv2.waitKey(0)
```



5. Adding Images

https://docs.opencv.org/master/dd/d4d/tutorial_js_image_arithmetics.html

https://docs.opencv.org/master/d0/d86/tutorial_py_image_arithmetics.html



```
▶ # Sum Images
# Sum the two image arrays for all channels
# Read in the two images

image_1 = cv2.imread('image01a.jpg')
image_2 = cv2.imread('image02a.jpg')

result = cv2.add(image_1, image_2)

cv2.imshow('result', result)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
▶ # Sum Images
image_1 = cv2.imread('image01a.jpg')
image_2 = cv2.imread('image02a.jpg')

result = cv2.addWeighted(image_1, 0.1, image_2, 0.9, 0)

cv2.imshow('result', result)
cv2.waitKey(0) # Wait for the user to press a key before continuing
cv2.destroyAllWindows()
```

6. Filtering Images

```
▶ # Filter by passing image through 3x3 averaging filter
# average_image
new_image= cv2.blur(image_1, (3,3))
```

```
▶ # Apply 3x3 gaussian filter on the original image
# gaussian_image
new_image = cv2.GaussianBlur(image_1, (3,3),0)
```

```
▶ # Apply 3x3 median filter on the original image
# median_image
new_image = cv2.medianBlur(image_1,3)
```

```
▶ import matplotlib.pyplot as plt

plt.imshow(new_image)
plt.show()
```



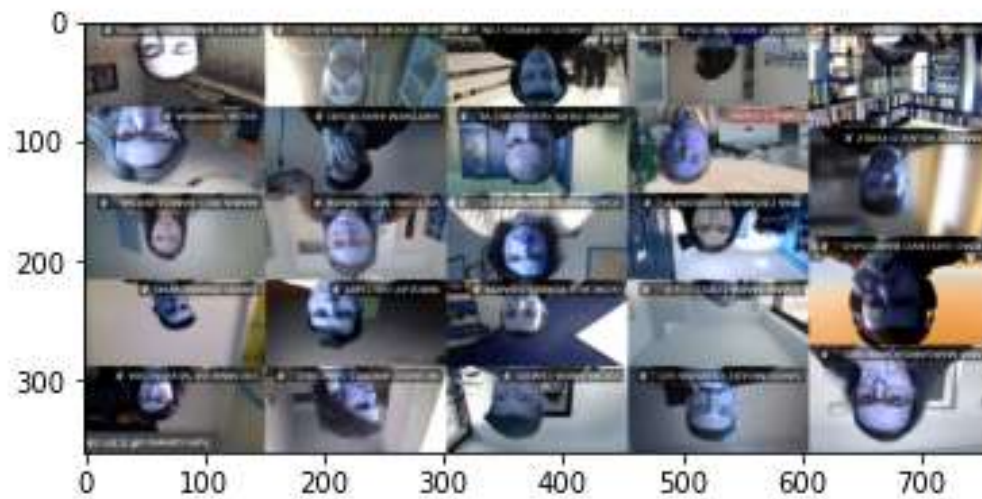
7. Rotate

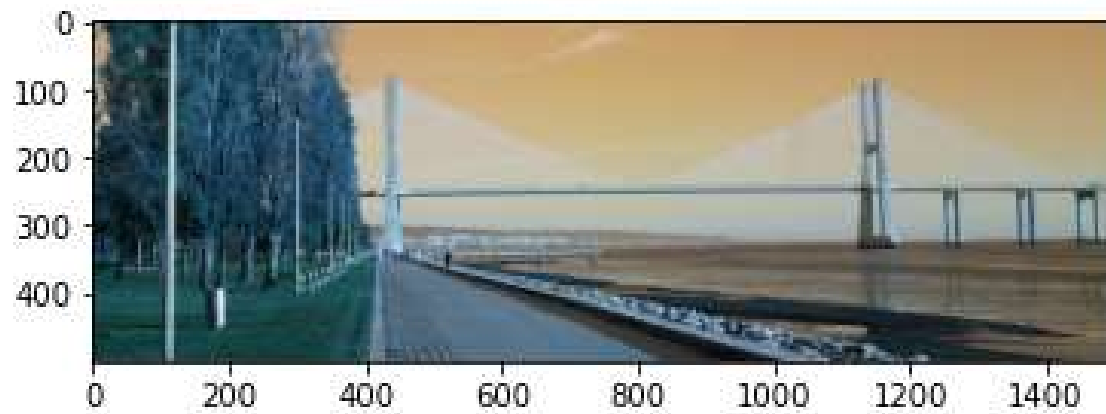
```
▶ image = cv2.imread('image01a.jpg',0)

# Rows and columns
rows, cols = image.shape

matrix = cv2.getRotationMatrix2D((cols/2,rows/2), 180, 1)
result = cv2.warpAffine(image_1,matrix,(cols,rows))

# Display resulting rotation
plt.imshow(result)
plt.show()
```





```
imag_ROI = image_1[300:400, 400:500]  
  
plt.imshow(imag_ROI)  
plt.show()
```

ROI



Conclusions

- Open CV
- Read and save
- Image resize
- Split and Merge images
- Add Images
- Rotate
- ROI

