

ESTIN

Basics of Image Processing

S4 (DS-AI)

2025-2026

Lab° 01

❖ Some Basics of OpenCV

While there are a number of good libraries (**Matplotlib**, **PIL**, ...etc), **OpenCV** is the most popular and documented library for handling images.

Reading, Writing, and displaying images with OpenCV

- ✓ Import the OpenCV library (`import cv2`).
- ✓ Load the image (flower.png) using `cv2.imread` specifying the path to image.
- ✓ Displaying this image using
 - `cv2.imshow` # to create GUI window to display an image on screen
 - `cv2.waitKey(0)` # to hold the window on screen
 - `cv2.destroyAllWindows()` # to remove the created GUI window
- ✓ Show the image type (`type(image)`), the image (`print(image)`), and the pixel [0,0] (`print(image[0,0])`)
- ✓ To know what is the array data type, use the dtype array property (`print(image.dtype)`)
- ✓ To get the dimension of the image, use the shape property (`print(image.shape)`)
- ✓ To save this image, use `'cv2.imwrite'` specifying the file name and the image
- ✓ Load the image in color (red_bird.png) using `'cv2.imread'`, and display it using `'cv2.imshow'`
- ✓ Show the pixel [0,0] (`print(image[0,0])`)
- ✓ Display this image using python plotting library **Matplotlib** (`plt.imshow`).
comment the results.
- ✓ Type this python code and tell what does it do

```
image_c = cv2.imread("...../red_bird.png", cv2.IMREAD_COLOR)
image_rgb = cv2.cvtColor(image_c, cv2.COLOR_BGR2RGB)
plt.imshow(image_rgb, cmap="gray")
```

```
plt.axis("off")
```

```
plt.show()
```

- ✓ Use `resize` to change the size of an image (`cv2.resize`)
- ✓ Use `cv2.split` function to split the image into each color channel. Display each channel.
- ✓ Re-make the original image using `cv2.merge` function.
- ✓ **Visualize the "Pure" Channels**

```
zeros=np.zeros(image_c.shape[0:2], dtype='uint8')
```

```
cv2.imshow('im_b',cv2.merge([b,zeros,zeros]))
```

```
cv2.imshow('im_g',cv2.merge([zeros,g,zeros]))
```

```
cv2.imshow('im_r',cv2.merge([zeros,zeros,r]))
```

```
cv2.waitKey()
```

```
cv2.destroyAllWindows()
```

Grayscale

Grayscale is process by which an image is converted from a full color to shades of grey.

- ✓ Use `cv2.cvtColor` to convert a color image to grayscale.

Histogram

- ✓ Calculate the histogram using `cv2.calcHist` and plot it using `plt.plot` (from matplotlib).
- ✓ What does the following code do

```
color=('b','g', 'r')
```

```
for i, col in enumerate(color):
```

```
    hist=cv2.calcHist([image_color], [i], None, [256], [0,256])
```

```
    plt.plot(hist, color=col)
```

To represent the image as surface using `view_surface.py` file, type the following code:

```
from view_surface import *
```

```
view_surface(' the path of the image ')
```