MLEAFIT CV

jero98772

Computer vision

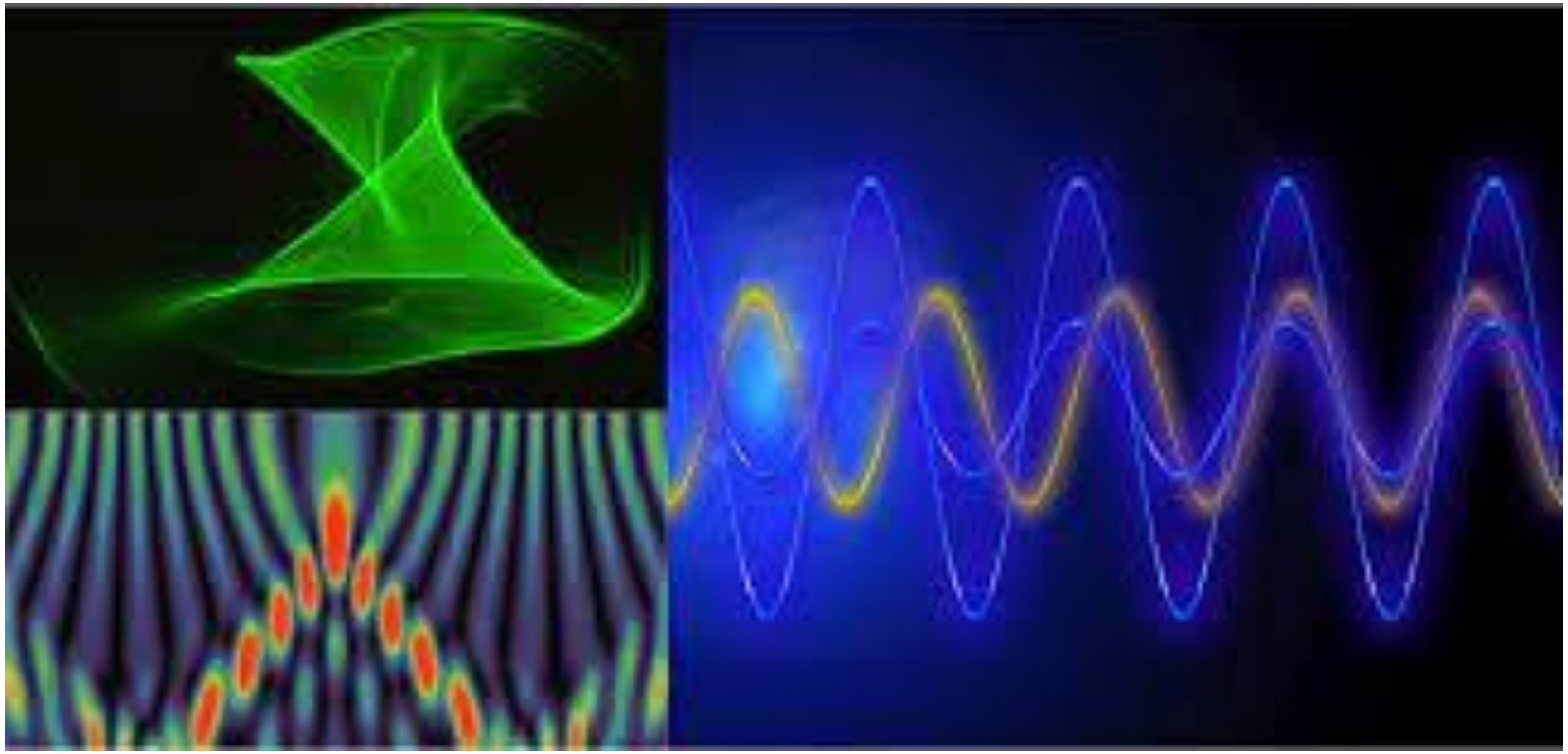
What is computer vison or (CV)

- Computer vision is a field that include methots for analyzing,proccesing ,acquiring and undestend images

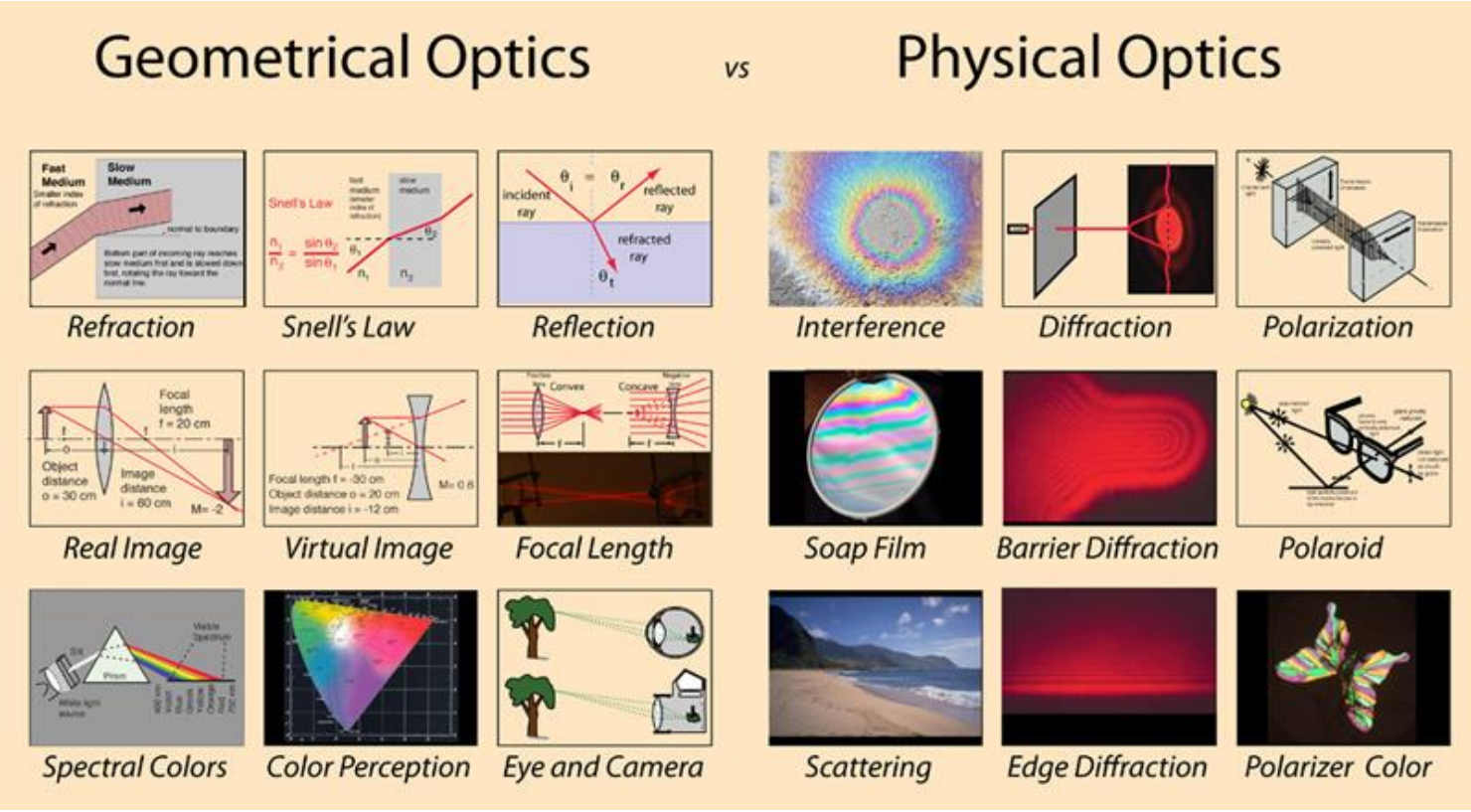
- We dont going to merge AI with CV (for today)

How we see

How physics see



vision for physics



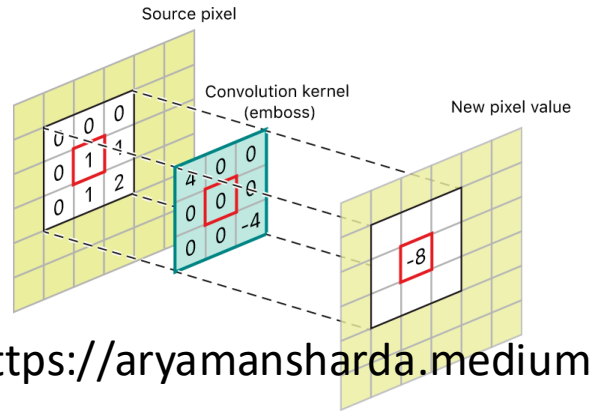
What is Computer vision

Blured


```
36          for x in -radius...radius {
37              for y in -radius...radius {
38                  let exponentNumerator = Double(-(x * x + y * y))
39                  let exponentDenominator = (2 * sigma * sigma)
40
41                  let eExpression = pow(M_E, exponentNumerator / exponentDenominator)
42                  let kernelValue = (eExpression / (2 * Double.pi * sigma * sigma))
43
44                  // We add radius to the indices to prevent out of bound issues because x
45                  kernel[x + radius][y + radius] = kernelValue
46                  sum += kernelValue
47              }
48          }
```

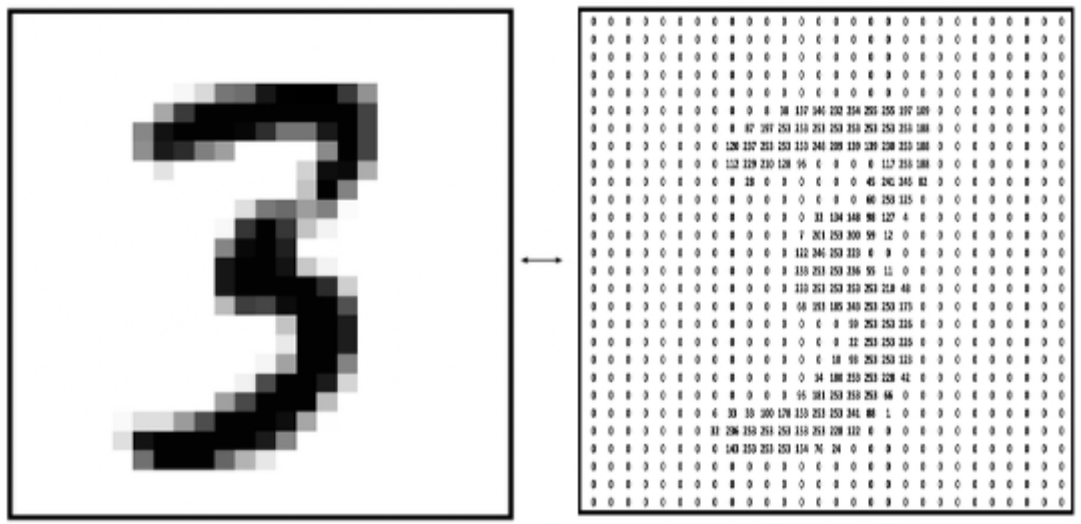
$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

```
redValue += Double(inputImage[x - kernelX, y - kernelY].red) * kernelValue
greenValue += Double(inputImage[x - kernelX, y - kernelY].green) * kernelValue
blueValue += Double(inputImage[x - kernelX, y - kernelY].blue) * kernelValue
```

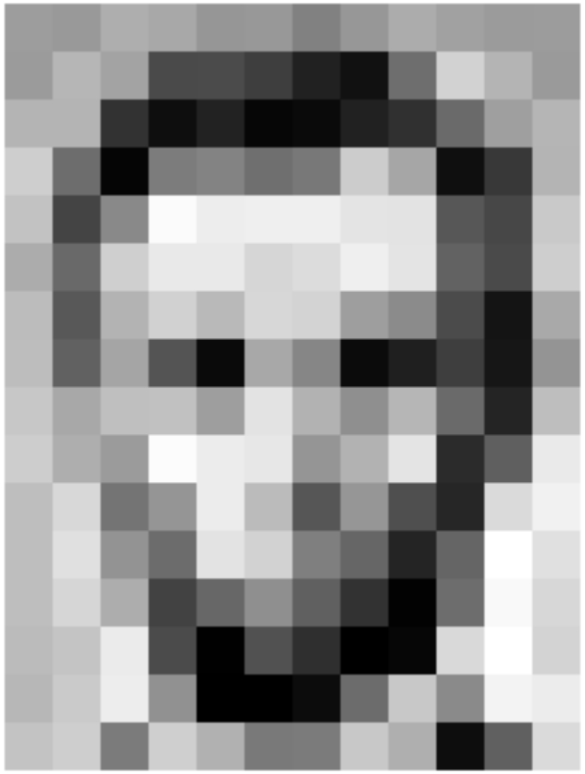


https://aryamansharda.medium.com/image-filters-gaussian-blur-eb36db6781b1

Matrix



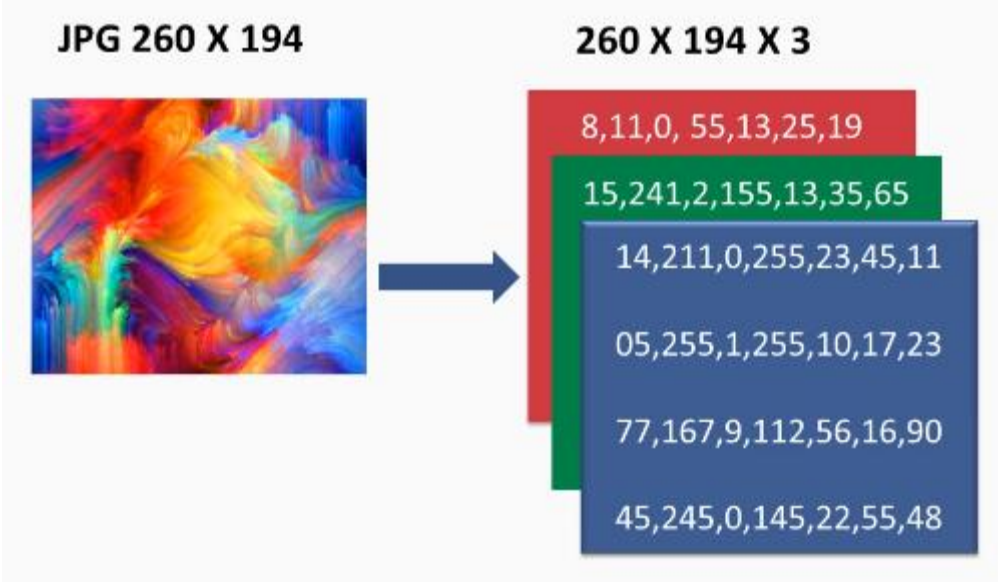
matrix



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

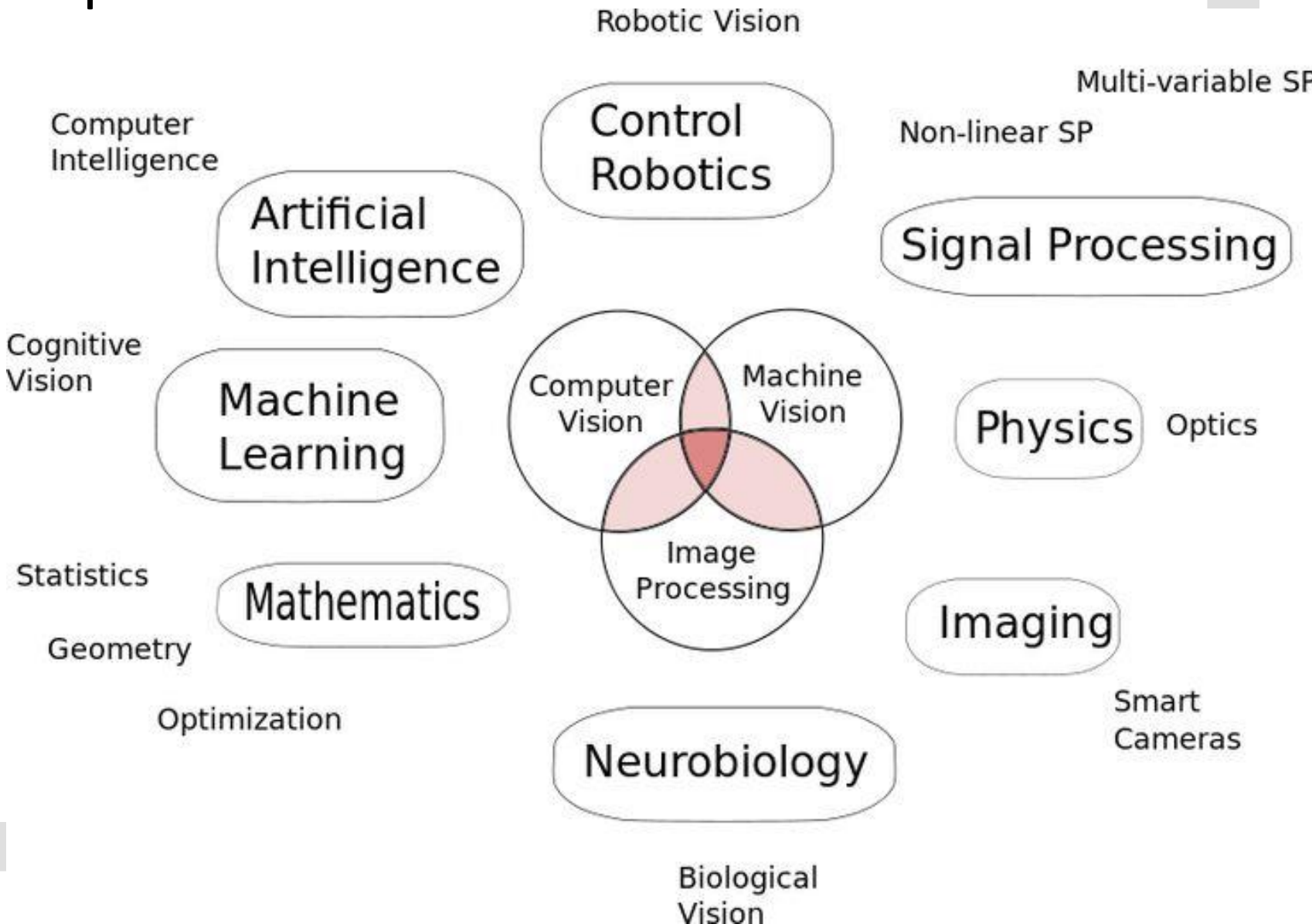
And more matrix



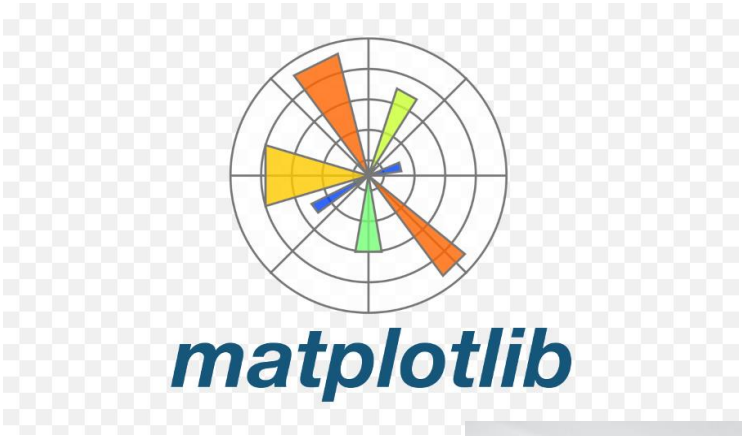
Why computer vision is not the usal

- It is many-to-one
- It is computacional intesive
- We dont understed the recognition problem

relation between computer vision and others fields



With wich Frameworks?



simple CV

- **Image Manipulation**: Pillow allows you to open, manipulate, and save many different image file formats. You can resize, rotate, crop, flip, and mirror images.
- **Image Filtering**: Apply various filters to images such as blur, contour, edge enhance, emboss, and sharpen.
- **Color Manipulation**: Adjust the color balance, brightness, contrast, saturation, and hue of images. You can also convert images between different color modes such as RGB, CMYK, grayscale, and others.
- **Text Overlays**: Add text to images, specifying font, size, color, and positioning.
- **Image Drawing**: Draw shapes such as lines, rectangles, ellipses, and polygons onto images.
- **Image Enhancements**: Apply enhancements like auto-contrast, auto-color, and auto-brightness adjustments to improve image quality.
- **Image Analysis**: Extract information from images such as histograms, basic statistics, and metadata.
- **Image Composition**: Combine multiple images into a single image, overlaying them in various ways.
- **Image Transformation**: Apply affine transformations like translation, rotation, scaling, and shearing to images.
- **Image Effects**: Apply artistic effects such as sepia tone, grayscale, and posterization.
- **Image Formats Conversion**: Convert images between different file formats.
- **Image Data Manipulation**: Access and manipulate individual pixels of an image.

SimpleCV

In computer vision, **blob detection** methods are aimed at detecting regions in a digital image that differ in properties, such as brightness or color, compared to surrounding regions. Informally, a blob is a region of an image in which some properties are constant or approximately constant; all the points in a blob can be considered in some sense to be similar to each other. The most common method for blob detection is convolution.

What i can do with it?

https://tutorial.simplecv.org/en/latest/index.html

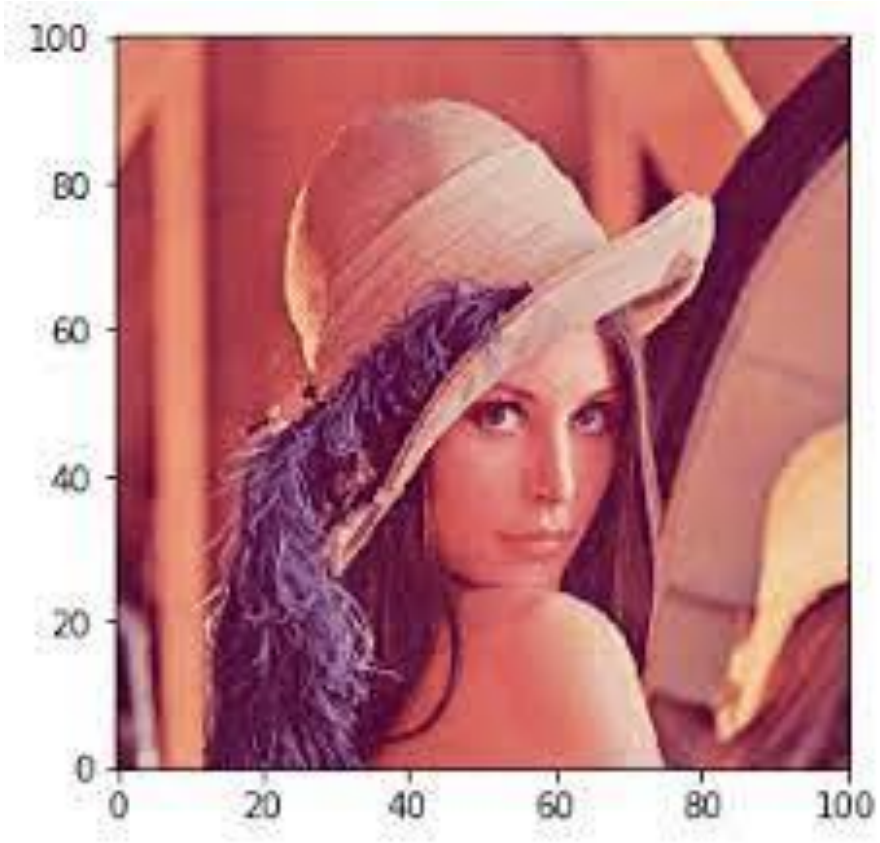
- **Loading and Saving Images**
- **Image Manipulation**
- Features are things you are looking for in the picture. They can be blobs, corners, lines
- Color Manipulation
- And more complex

pip install SimpleCV

```
import SimpleCV
cam = SimpleCV.Camera()

while True:
    img = cam.getImage()
    img.show()
```

matplotlib

- Plot image as a plot and interact with it



Numpy

- Make apply linear algebra technicques

```
>>>  a1D = np.array([1, 2, 3, 4])
>>> a2D = np.array([[1, 2], [3, 4]])
>>> a3D = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]]])
```

pip install numpy

Blobs

Open CV

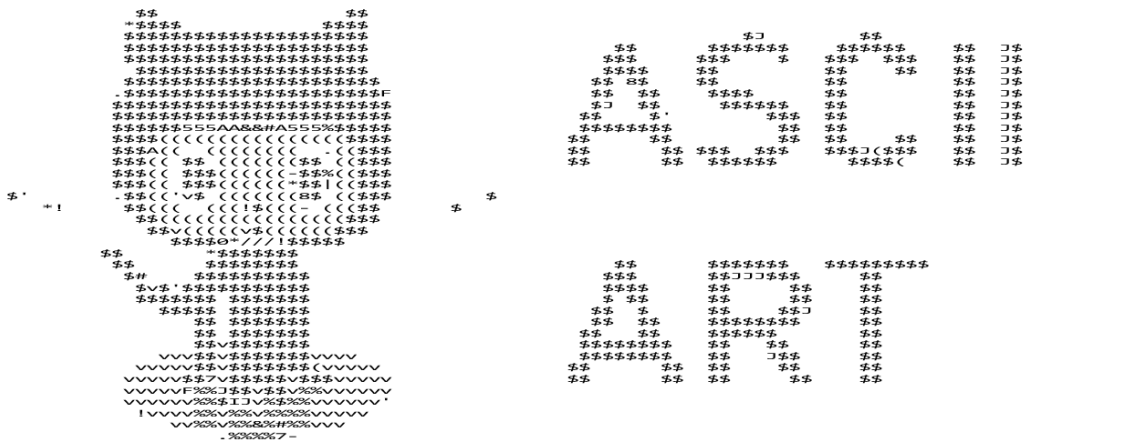
for video

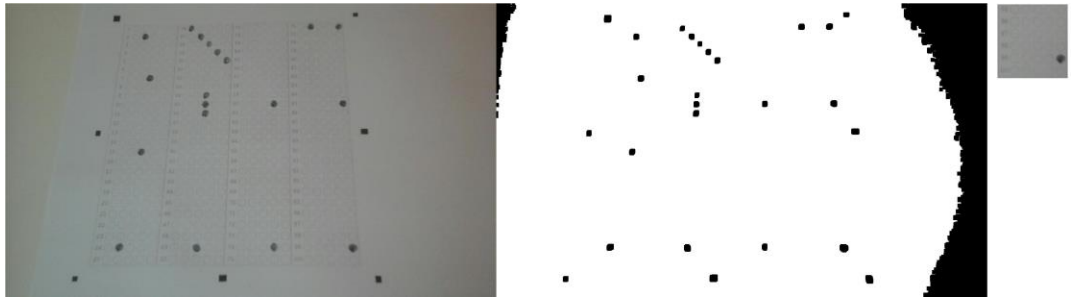
for image

for python

for c++

What you can do with cv?

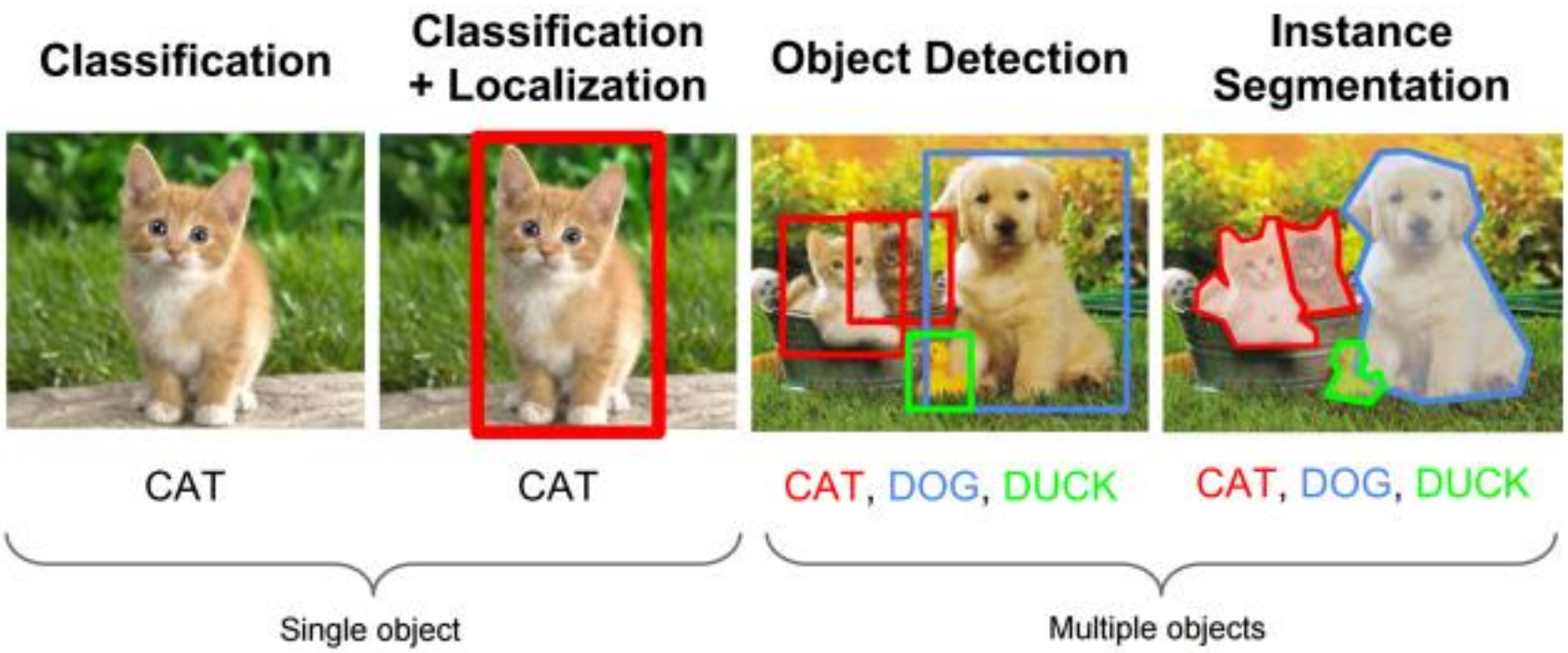




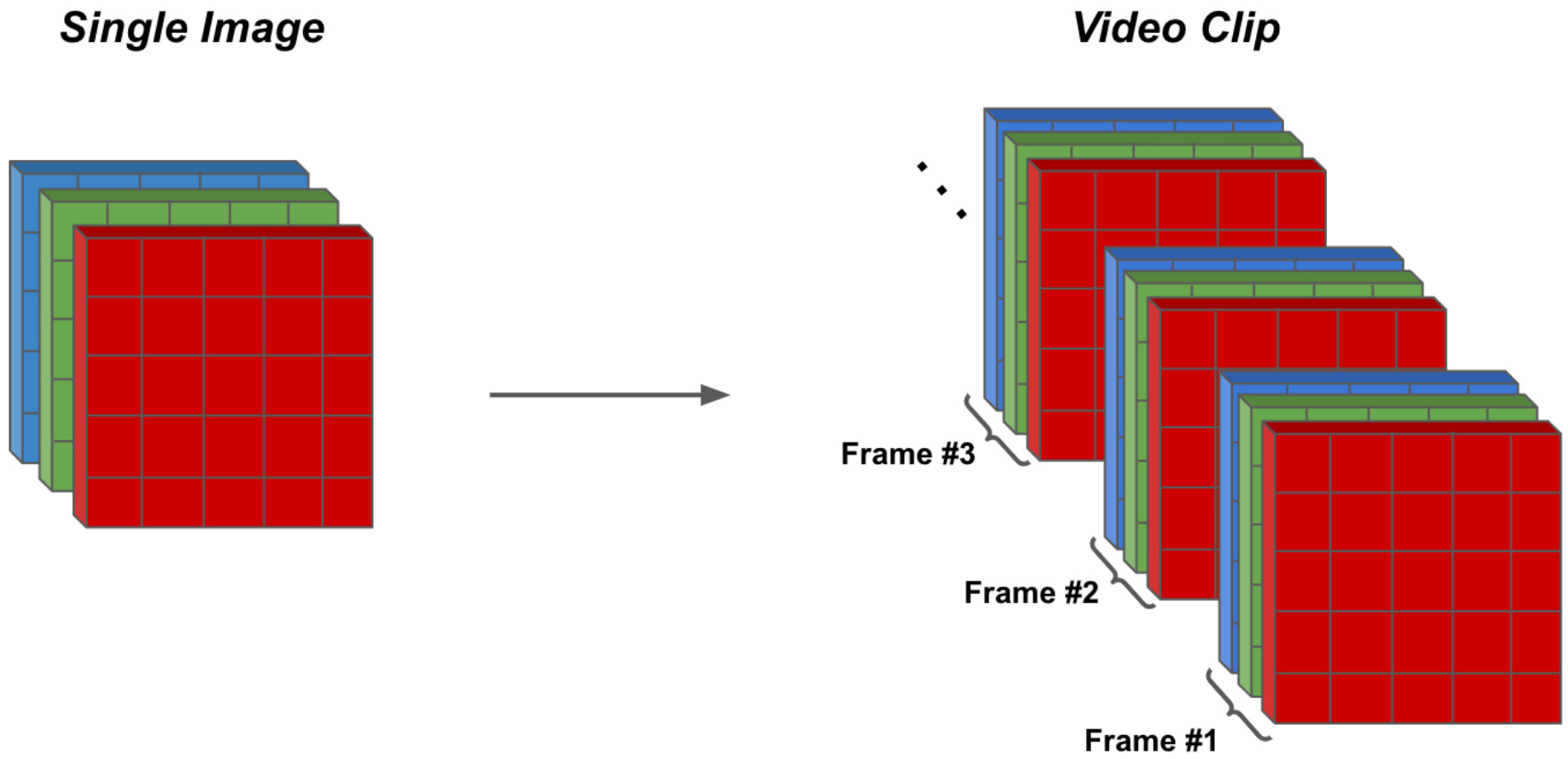
A list of algorithms

- **Negative**: Inverts the colors of an image by subtracting each color channel value from the maximum value (e.g., 255 for an 8-bit image).
- **Crop**: Selects a rectangular region of interest (ROI) from an image, discarding the rest.
- **Decrease Brightness**: Reduces the intensity of all pixels in an image, making it appear darker. This can be done by subtracting a constant value from each pixel.
- **Increase Brightness**: Increases the intensity of all pixels in an image, making it appear brighter. This can be done by adding a constant value to each pixel.
- **Grayscale**: Converts a color image into a grayscale image, where each pixel is represented by a single intensity value corresponding to its luminance.
- **Thresholding**: Converts an image into a binary image by assigning a binary value (typically 0 or 255) to each pixel based on whether its intensity value is above or below a specified threshold.
- **Edge Detection**: Identifies the edges in an image by detecting abrupt changes in intensity.
- **Blur/Smoothing**: Reduces noise and sharpness in an image by averaging the pixel values in a neighborhood around each pixel.
- **Histogram Equalization**: Enhances the contrast of an image by redistributing its intensity values to cover a wider range.
- **Resize/Scaling**: Changes the dimensions of an image by interpolating the pixel values to fit a new size.

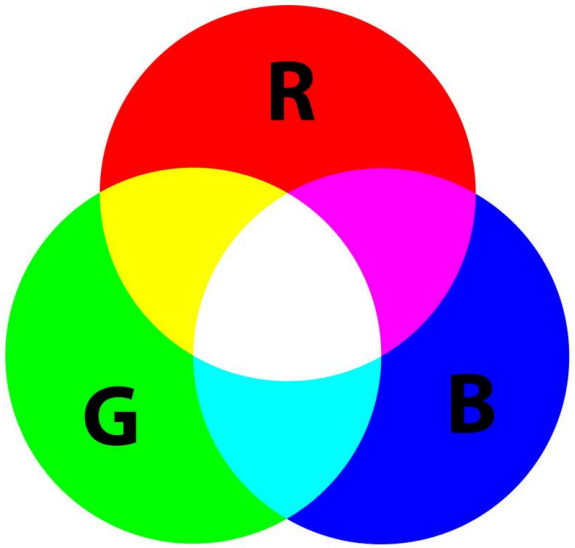
What we can do with CV and ML

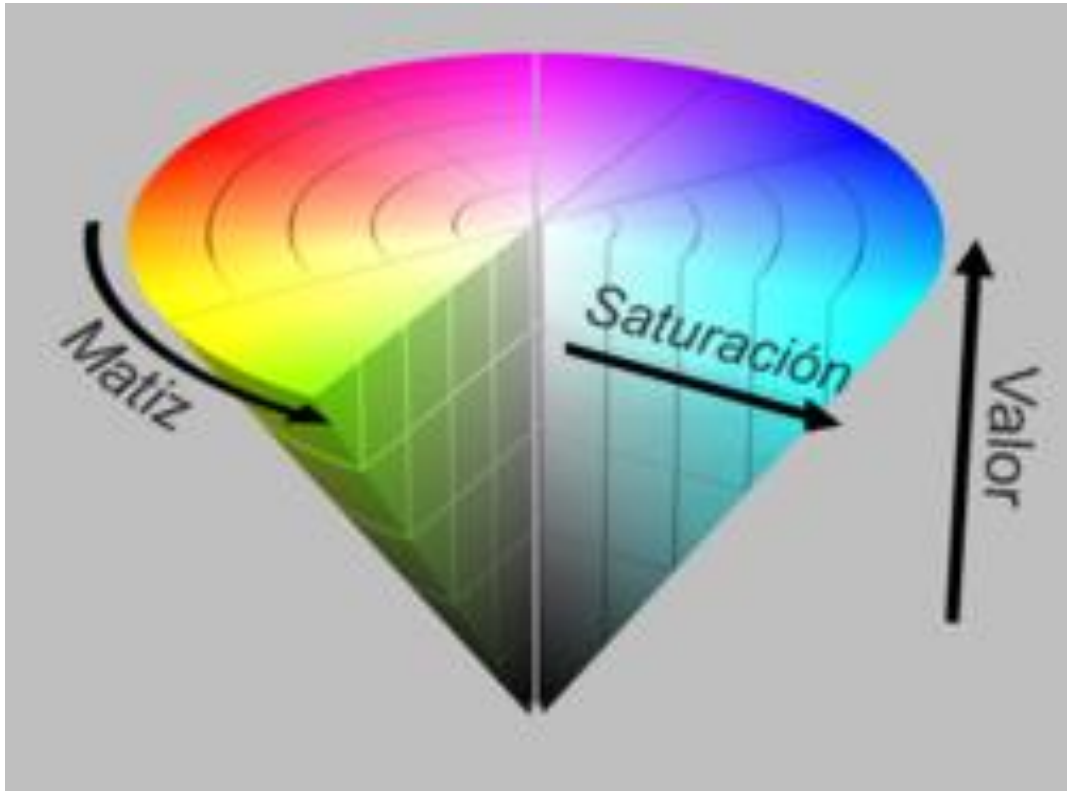


Anatomy of video and image



Colors format, rgb y hsv





How can i load a image in python?

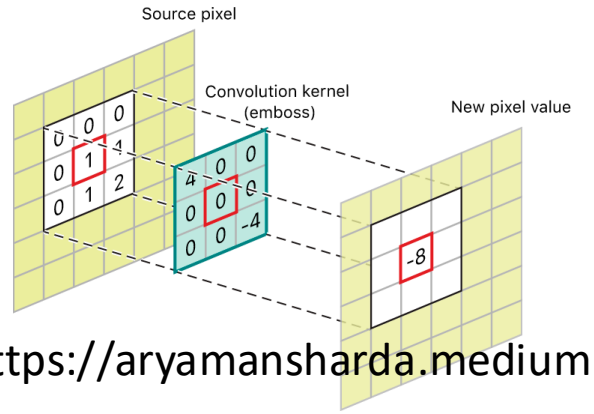
```
import cv2
import matplotlib.image as mpi
import matplotlib.pyplot as plt
import numpy as np

def img(file):
    datos = mpi.imread(file)
    return datos
```

Blured


```
36          for x in -radius...radius {
37              for y in -radius...radius {
38                  let exponentNumerator = Double(-(x * x + y * y))
39                  let exponentDenominator = (2 * sigma * sigma)
40
41                  let eExpression = pow(M_E, exponentNumerator / exponentDenominator)
42                  let kernelValue = (eExpression / (2 * Double.pi * sigma * sigma))
43
44                  // We add radius to the indices to prevent out of bound issues because x
45                  kernel[x + radius][y + radius] = kernelValue
46                  sum += kernelValue
47              }
48          }
```

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

```
redValue += Double(inputImage[x - kernelX, y - kernelY].red) * kernelValue
greenValue += Double(inputImage[x - kernelX, y - kernelY].green) * kernelValue
blueValue += Double(inputImage[x - kernelX, y - kernelY].blue) * kernelValue
```



https://aryamansharda.medium.com/image-filters-gaussian-blur-eb36db6781b1

Negative

```python
def negative(imagen):
        height = imagen.shape[0]
        width = imagen.shape[1]
        img = np.zeros((height,width,3),dtype=int)
        for i in range(height):
                for j in range(width):
                        img[i][j][0] = abs(imagen[i][j][0]-255)
                        img[i][j][1] = abs(imagen[i][j][1]-255)
                        img[i][j][2] = abs(imagen[i][j][2]-255)
        return img
```

Original image

Negative image

Mirrored

```
def mirror(imagen):
        height = imagen.shape[0]
        width = imagen.shape[1]
        img = np.zeros((height,width,3),dtype=int)
        for i in range(width):
                for j in range(height):

                        img[j][width-i-1] = imagen[j][i]
        return img
```

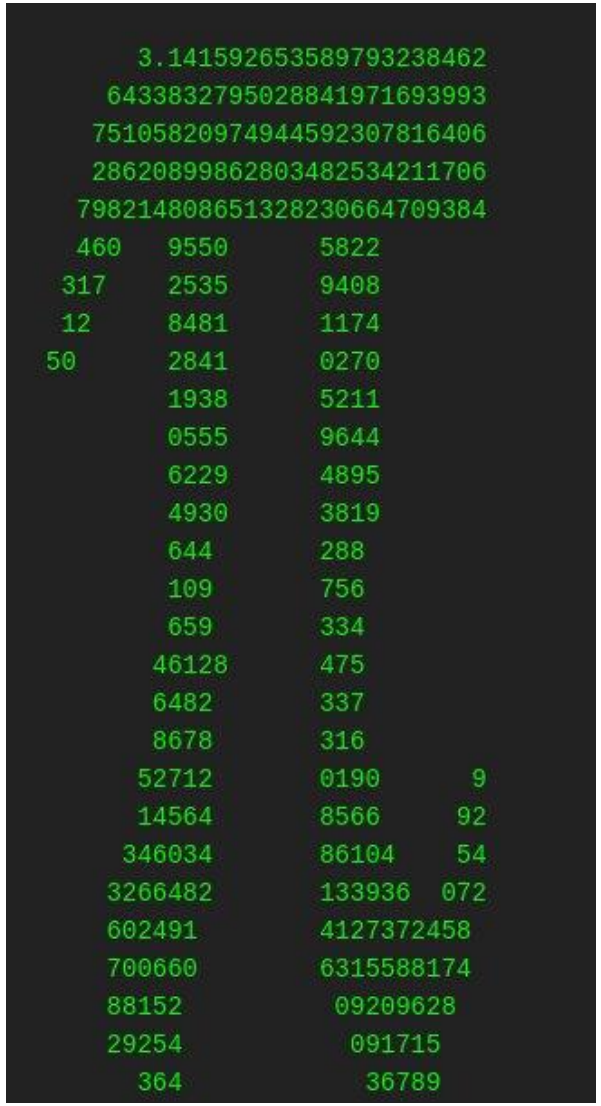
Rotate

```
def rotate90(imagen):
        height = imagen.shape[0]
        width = imagen.shape[1]
        img = np.zeros((alto,width,3),dtype=int)
        for i in range(height):
                for j in range(ancho):
                        img[j][i][0] = imagen[i][j][0]
                        img[j][i][1] = imagen[i][j][1]
                        img[j][i][2] = imagen[i][j][2]
        return img
```

How can i visualisate it?

```
def show(datos):
    f = plt.figure()
    f.add_subplot(1,2,1)
    plt.imshow(datos)
    plt.show()
```

Next meet workshop

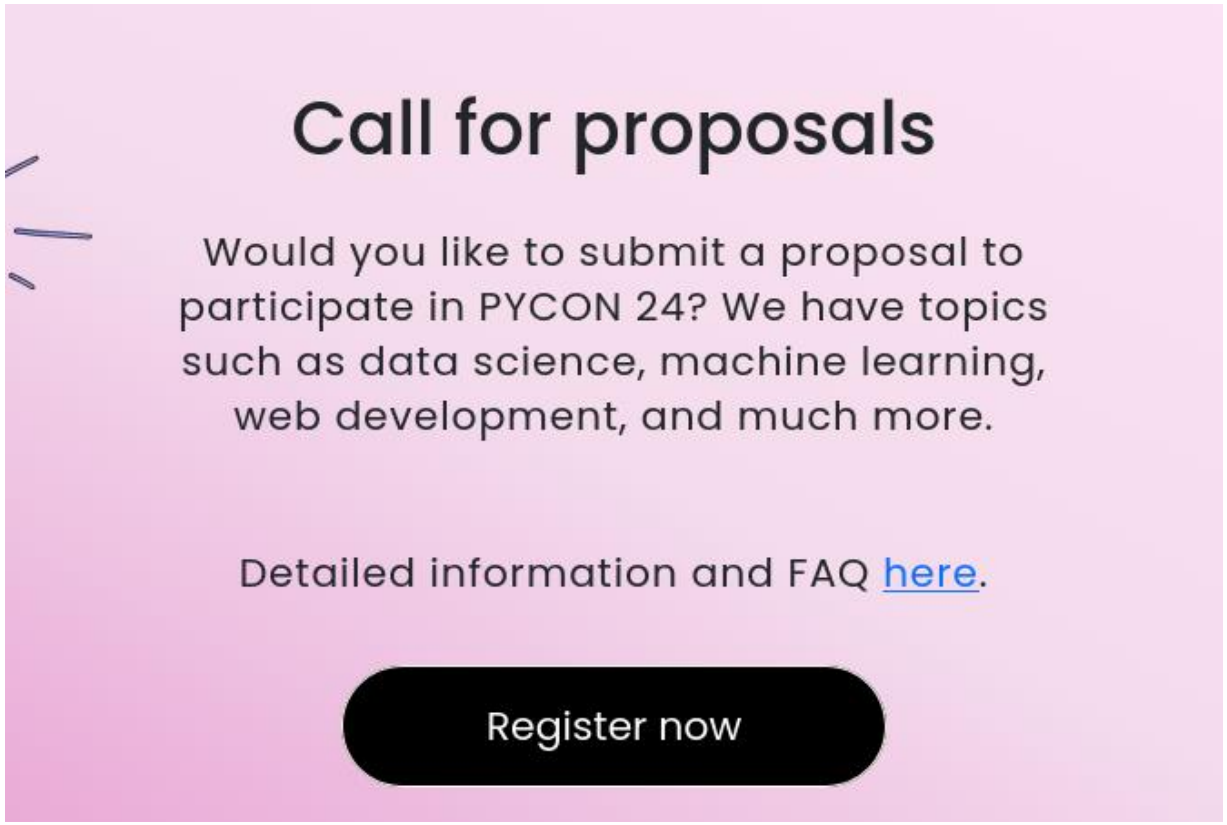


Thanks

- Question ?
  Suggestions?

proposal


Python
Medellín
BUSCAMOS
PONENTES PARA
MEETUPS!
¿Quieres compartir tus conocimientos
con la comunidad de Python Medellín?
¡Esta es tu oportunidad!


Call for proposals
Would you like to submit a proposal to participate in PYCON 24? We have topics such as data science, machine learning, web development, and much more.
Detailed information and FAQ here.
Register now