

# OpenCV (Python)

---

# ¿Qué es OpenCV?

---

OpenCV es una librería open source para trabajar en visión artificial

Está escrita en C y C++ y está disponible para

- Windows, Linux y Mac OS

Librería diseñada para la eficiencia computacional y centrada en el procesamiento en tiempo real

# ¿Por qué la visión artificial es tan complicada?

---

Lo que nosotros vemos vs lo que ve la máquina



**But the camera sees this:**

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	74	65
20	41	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

# Principales funciones de entrada/salida

---

Las principales funciones de entrada y salida de la librería opencv son

- cv2.imread()
- cv2.imshow()
- cv2.imwrite()

También podemos utilizar otras librerías para la entrada/salida de datos

- Matplotlib
- ...

# Leer imágenes: imread

---

## Parámetros de entrada

- filename: nombre (con o sin ruta completa) del archivo de la imagen
- [flags]: especifican el tipo de “color” de la imagen

## Parámetros de salida

- retval: matriz correspondiente a la imagen leída

## CUIDADO:

- Cuando la imagen se lea en color, los canales vienen ordenados en B G R

# Leer imágenes: imread

Enumerator	
IMREAD_UNCHANGED Python: cv.IMREAD_UNCHANGED	If set, return the loaded image as is (with alpha channel, otherwise it gets cropped).
IMREAD_GRAYSCALE Python: cv.IMREAD_GRAYSCALE	If set, always convert image to the single channel grayscale image.
IMREAD_COLOR Python: cv.IMREAD_COLOR	If set, always convert image to the 3 channel BGR color image.
IMREAD_ANYDEPTH Python: cv.IMREAD_ANYDEPTH	If set, return 16-bit/32-bit image when the input has the corresponding depth, otherwise convert it to 8-bit.
IMREAD_ANYCOLOR Python: cv.IMREAD_ANYCOLOR	If set, the image is read in any possible color format.
IMREAD_LOAD_GDAL Python: cv.IMREAD_LOAD_GDAL	If set, use the gdal driver for loading the image.
IMREAD_REDUCED_GRAYSCALE_2 Python: cv.IMREAD_REDUCED_GRAYSCALE_2	If set, always convert image to the single channel grayscale image and the image size reduced 1/2.
IMREAD_REDUCED_COLOR_2 Python: cv.IMREAD_REDUCED_COLOR_2	If set, always convert image to the 3 channel BGR color image and the image size reduced 1/2.
IMREAD_REDUCED_GRAYSCALE_4 Python: cv.IMREAD_REDUCED_GRAYSCALE_4	If set, always convert image to the single channel grayscale image and the image size reduced 1/4.
IMREAD_REDUCED_COLOR_4 Python: cv.IMREAD_REDUCED_COLOR_4	If set, always convert image to the 3 channel BGR color image and the image size reduced 1/4.
IMREAD_REDUCED_GRAYSCALE_8 Python: cv.IMREAD_REDUCED_GRAYSCALE_8	If set, always convert image to the single channel grayscale image and the image size reduced 1/8.
IMREAD_REDUCED_COLOR_8 Python: cv.IMREAD_REDUCED_COLOR_8	If set, always convert image to the 3 channel BGR color image and the image size reduced 1/8.
IMREAD_IGNORE_ORIENTATION Python: cv.IMREAD_IGNORE_ORIENTATION	If set, do not rotate the image according to EXIF's orientation flag.

```
cv::IMREAD_UNCHANGED = -1,  
cv::IMREAD_GRAYSCALE = 0,  
cv::IMREAD_COLOR = 1,  
cv::IMREAD_ANYDEPTH = 2,  
cv::IMREAD_ANYCOLOR = 4,  
cv::IMREAD_LOAD_GDAL = 8,  
cv::IMREAD_REDUCED_GRAYSCALE_2 = 16,  
cv::IMREAD_REDUCED_COLOR_2 = 17,  
cv::IMREAD_REDUCED_GRAYSCALE_4 = 32,  
cv::IMREAD_REDUCED_COLOR_4 = 33,  
cv::IMREAD_REDUCED_GRAYSCALE_8 = 64,  
cv::IMREAD_REDUCED_COLOR_8 = 65,  
cv::IMREAD_IGNORE_ORIENTATION = 128
```

# Leer imágenes: imread

---

## Formatos aceptados

- Windows bitmaps (bmp)
- JPEG
- Png
- ...

# Mostrar imágenes: imshow

---

## Argumentos de entrada

- winname: string con el nombre de la ventana en la que se abre la imagen
- mat: matriz (imagen) que queremos mostrar

## Argumentos de salida

- None

## ¿Cómo se muestra?

- Si la imagen tiene formato sin signo 8 bits (uint8), la imagen se muestra tal cual
- Si la imagen está en formato coma flotante 32 bits (float32), los píxeles deben estar en [0,1] y serán multiplicados por 255 para mostrar
- Cuidado con el resto de combinaciones...

# Mostrar imágenes: imshow

---

## Importante

- La función imshow debe estar acompañada de la función waitKey(int), donde int es el número de milisegundos en los que se muestra la imagen
- Por defecto, pondremos siempre waitKey(0), lo cual hace que la imagen se muestre hasta que apretemos una tecla del teclado
- Para que no haya ningún fallo más, debemos cerrar todas las ventanas con la función destroyWindow() o destroAllWindows()

## Ejemplo

```
imshow(imagen)
```

```
waitKey(0)
```

```
destroyAllWindows()
```

# Guardar imágenes: imwrite

---

## Parámetros de entrada

- filename: nombre que queramos darle al archivo
- img: matriz que queremos almacenar
- [params]: lista de parámetros, valor

## Importante

- La imagen que vayamos a guardar debe estar en formato entero 8 bits sin signo (uint8)

# Otras funciones interesantes

---

## Conversión de espacios de color: CvtColor

- Parámetros de entrada:
  - src: imagen a transformar
  - dst: imagen destino
  - code: código de conversión de color
    - COLOR\_BGR2GRAY, COLOR\_RGB2GRAY, COLOR\_GRAY2BGR, COLOR\_GRAY2RGB, COLOR\_RGB2BGR, COLOR\_BGR2RGB
    - [https://docs.opencv.org/3.4.2/d7/d1b/group\\_\\_imgproc\\_\\_misc.html#ga4e0972be5de079fed4e3a10e24ef5ef0](https://docs.opencv.org/3.4.2/d7/d1b/group__imgproc__misc.html#ga4e0972be5de079fed4e3a10e24ef5ef0)

## Separación/Unión de componentes de color

- split()
- merge()