

Prácticas de Visión por Computador

Fundamentos de Python para manipulación de imágenes

Pablo Mesejo

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial



UNIVERSIDAD
DE GRANADA



Lectura de Imágenes

```
filename = "images/orapple.jpg"
```

```
import matplotlib.image as mpimg  
image1 = mpimg.imread(filename)
```

```
from PIL import Image  
image2 = Image.open(filename)
```

```
import cv2 as cv  
image3 = cv.imread(filename)
```



Python Imaging Library (PIL)



Lectura de Imágenes (y 2)

```
image1.shape
```

```
Out[1]: (535, 500, 3)
```



```
image2.shape
```

```
AttributeError: 'JpegImageFile' object has no attribute  
    'shape'
```

```
import numpy as np
```

```
im = np.asarray(image2)
```

```
im.shape
```

```
Out[1]: (535, 500, 3)
```

Python Imaging Library (PIL)

```
image3.shape
```

```
Out[1]: (535, 500, 3)
```



Visualización de Imágenes

```
from IPython.display import Image  
Image(filename)
```

IP[y]: IPython
Interactive Computing

```
import matplotlib.pyplot as plt  
plt.imshow(image1)
```

matplotlib

```
image2.show()
```

Python Imaging Library (PIL)

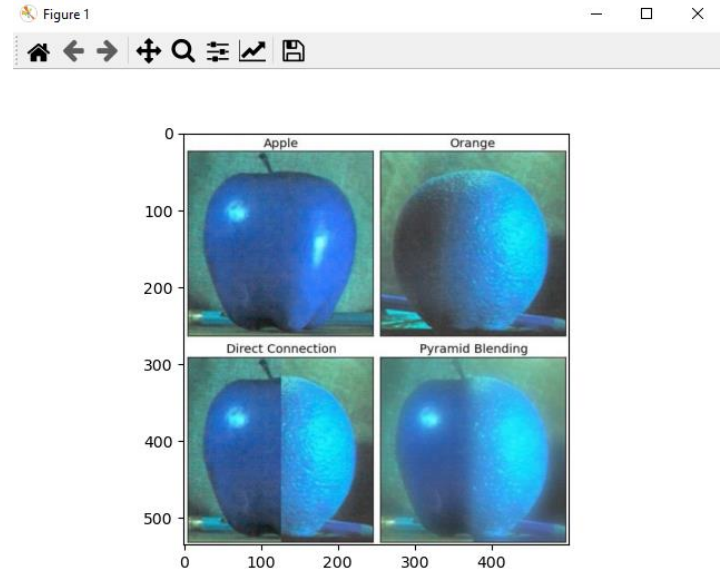
```
cv.imshow('Titulo Imagen', image3)
```

 OpenCV

Ojo con los detalles

OpenCV almacena las imágenes en BGR en lugar de RGB. Si leemos con *OpenCV* y visualizamos con *matplotlib* debemos tener cuidado.

```
filename = "images/orapple.jpg"
import cv2 as cv
image = cv.imread(filename)
import matplotlib.pyplot as plt
plt.imshow(image)
```

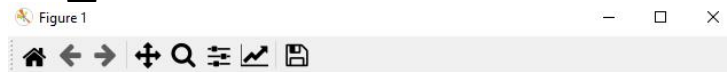


Ojo con los detalles

Debemos reordenar los canales para mostrarlos correctamente

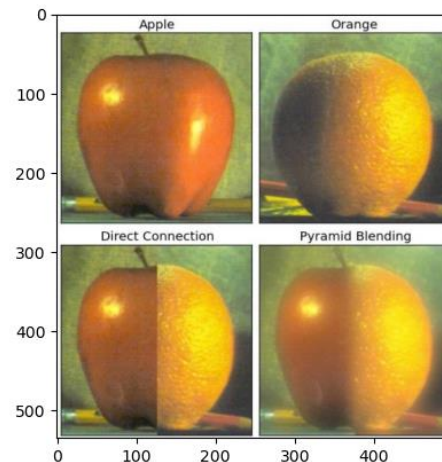
Una alternativa es:

```
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
```



Otra alternativa es:

```
plt.imshow(image[:, :, ::-1])
```



Ojo con los detalles

cv2.imshow en Google Colab puede dar problemas



```
import cv2
```

```
img = cv2.imread('./drive/My Drive/Colab Notebooks/orapple.jpg')  
cv2.imshow('Imagen',img)
```



```
-----  
DisabledFunctionError                                Traceback (most recent call last)  
<ipython-input-2-ebc0249a3f51> in <module>()  
      2  
      3 img = cv2.imread('./drive/My Drive/Colab Notebooks/orapple.jpg')  
----> 4 cv2.imshow('Imagen',img)  
  
/usr/local/lib/python3.6/dist-packages/google/colab/_import_hooks/_cv2.py in wrapped(*args, **kwargs)  
     50 def wrapped(*args, **kwargs):  
     51     if not os.environ.get(env_var, False):  
--> 52         raise DisabledFunctionError(message, name or func.__name__)  
     53     return func(*args, **kwargs)  
     54
```

DisabledFunctionError: cv2.imshow() is disabled in Colab, because it causes Jupyter sessions to crash; see <https://github.com/jupyter/notebook/issues/3935>.

As a substitution, consider using

```
from google.colab.patches import cv2_imshow
```

Ojo con los detalles

cv2.imshow en Google Colab puede dar problemas

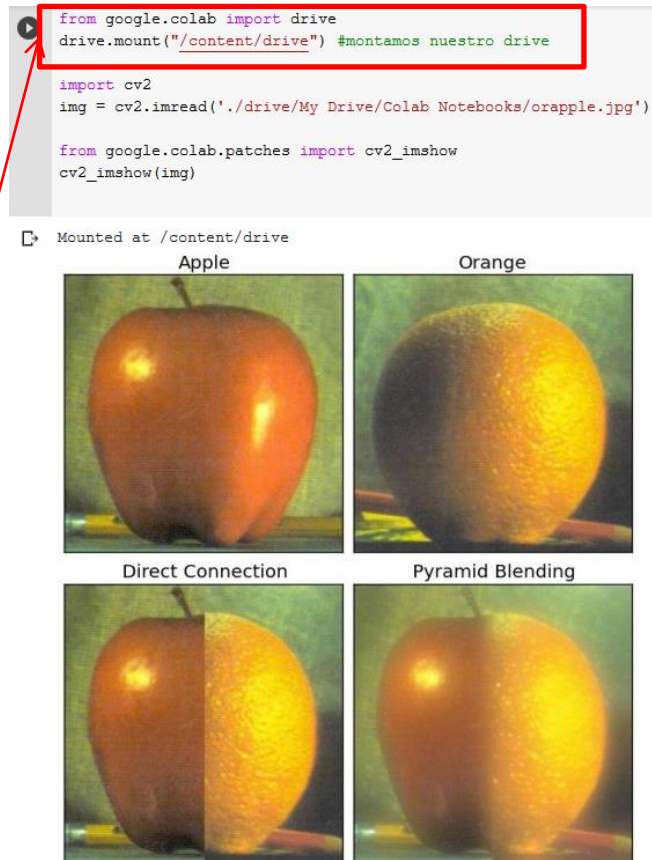
Una alternativa:

```
from google.colab.patches import cv2_imshow
cv2_imshow(img)
```

Otra alternativa:

```
import matplotlib.pyplot as plt
plt.imshow(img[:, :, ::-1])
```

Nota: acordaos de montar vuestro Drive en Colab para acceder a vuestros datos



Ojo con los detalles

Cuidado a la hora de visualizar matrices:

```
img = np.ones([500,500])*64  
cv.imshow("Escala de grises MAL",img)
```

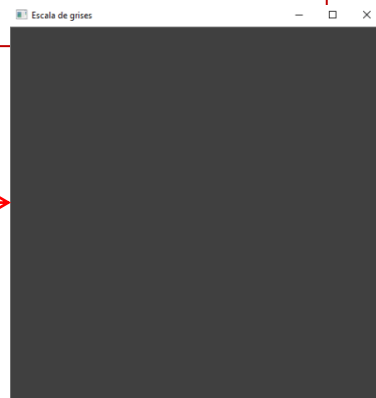
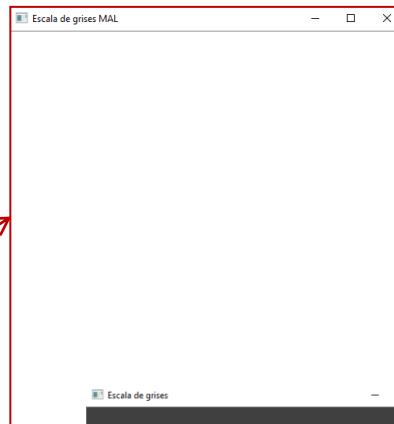
Es una matriz
de np.float64

Satura a blanco (es decir, a
1) todo valor mayor que 1

```
cv.imshow("Escala de grises",img/255)
```

Nota: o también podéis usar matplotlib

```
plt.imshow(img, cmap='gray', vmin=0, vmax=255)
```

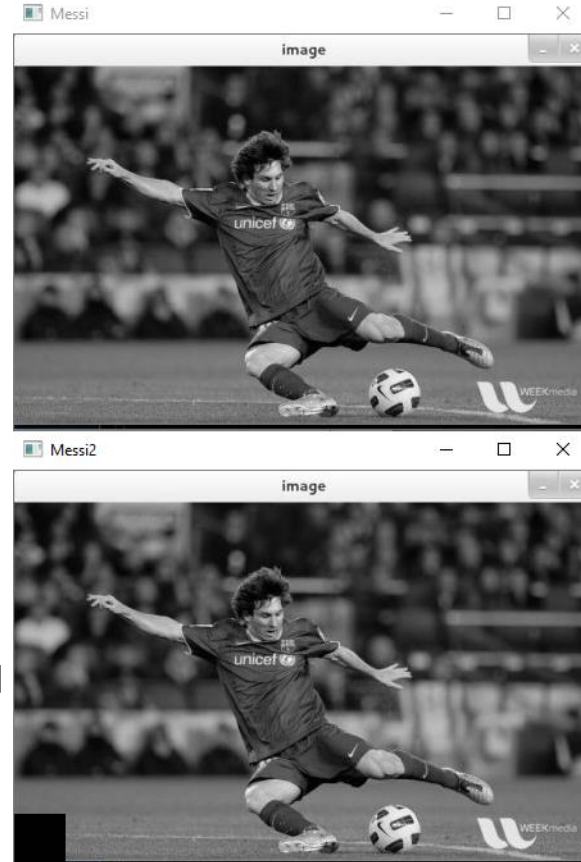


Ojo con los detalles

- Debemos tener cuidado...
 - ¿por qué el código siguiente no saca un cuadrado blanco?

```
import cv2
lado = 40
img = cv2.imread('images/messi.jpg')
img2 = img.copy()
img2[img2.shape[0]-lado:img2.shape[0],0:lado,:] = [1,1,1]
cv2.imshow('Messi', img)
cv2.imshow('Messi2', img2)
```

La esquina superior
izquierda es el (0,0)



Ojo con los detalles

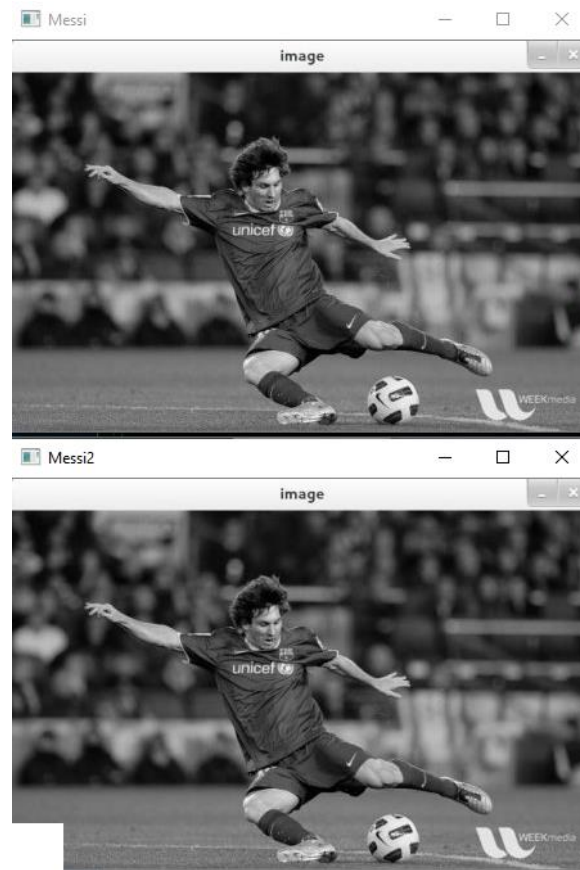
- Porque OpenCV lee la imagen como una matriz de enteros (y 1 es casi negro)

Nomb	Tipo	Tamaño	Valor
img	Array of uint8	(308, 450, 3)	[[[181 181 181]
img2	Array of uint8	(308, 450, 3)	[[[181 181 181]
lado	int	1	40

- Una forma de solucionarlo:

```
import cv2
lado = 40
img = cv2.imread('images/messi.jpg')/255
img2 = img.copy()
img2[img2.shape[0]-lado:img2.shape[0],0:lado,:] =
    [1,1,1]
cv2.imshow('Messi', img)
cv2.imshow('Messi2', img2)
```

Filas de la imagen/matriz. Eje Y Columnas de la imagen/matriz. Eje X



Ojo con los detalles

- Otro ejemplo:
 - Leemos/construimos una imagen con tipo uint8 [0-255]
 - La modificamos con algunos píxeles float64 → la imagen pasa a ser de ese tipo
 - Automáticamente, al ser float, se espera que el rango de valores esté en [0-1]
 - Obtendremos el mensaje: `"WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers)."`
 - Y la imagen aparecerá saturada a blanco porque todos los valores superiores a 1 colapsarán en 1.

Ojo con los detalles

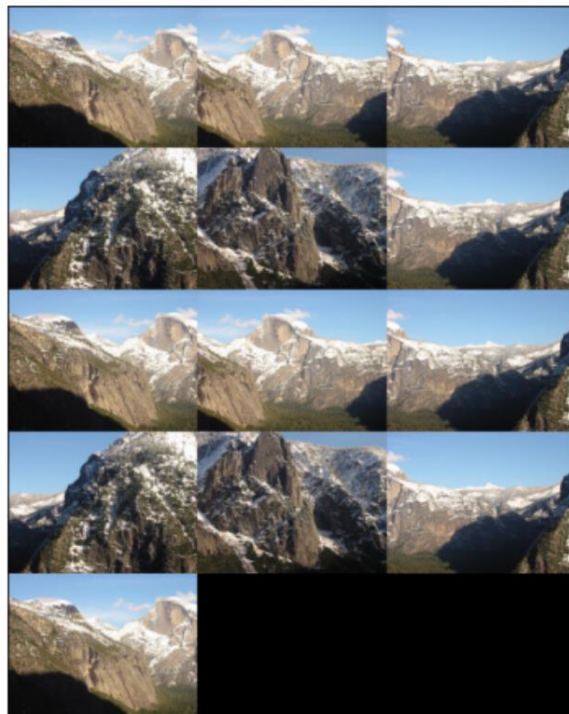
```
row=np.hstack((row,np.zeros(vim[0].shape)))
```

Yosemite



```
row=np.hstack((row,np.zeros(vim[0].shape,  
dtype=vim[0].dtype)))
```

Yosemite



Ojo con los detalles

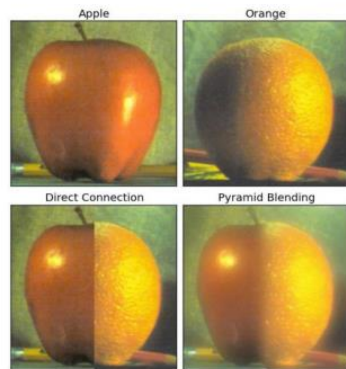
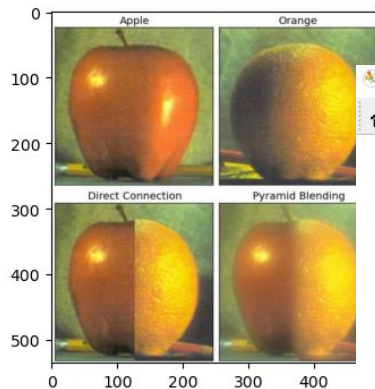
Si queréis mostrar varias figuras simultáneamente con *matplotlib*,
acordaos de usar `plt.figure()`, o se sobrescribirán

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
```

```
filename = "images/orapple.jpg"
image = mpimg.imread(filename)
```

```
plt.figure(1)
plt.imshow(image)
```

```
plt.figure(2)
plt.axis("off")
plt.imshow(image)
```



Enlaces de interés

- OpenCV-Python:
 - https://docs.opencv.org/4.6.0/d6/d00/tutorial_py_root.html
- Google Colab:
 - <https://colab.research.google.com/notebooks/intro.ipynb>
 - https://colab.research.google.com/notebooks/basic_features_overview.ipynb
 - https://colab.research.google.com/drive/1RWGmqoEQdeyh5Tss_oGtsXsFk8hbLGtWp
- Jupyter Notebook:
 - <https://www.dataquest.io/blog/jupyter-notebook-tutorial/>