

Nivell 3

Manipulació d'imatges amb Matplotlib.

Carregareu qualsevol imatge (jpg, png ..) amb Matplotlib. adoneu-vos que les imatges RGB (Red, Green, Blue) són realment només amples × alçades × 3 matrius (tres canals Vermell, Verd i Blau), una per cada color de nombres enters int8,

manipuleu aquests bytes i torneu a utilitzar Matplotlib per desar la imatge modificada un cop hàgiu acabat.

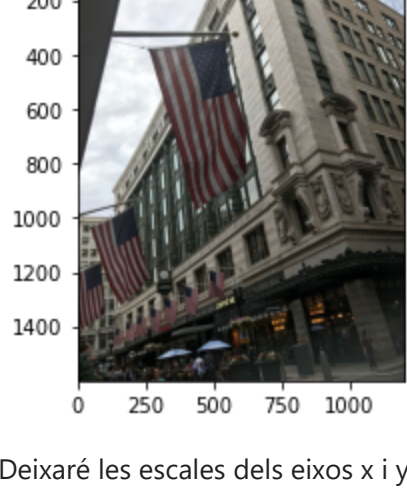
Ajuda:Importeu, import matplotlib.image as mpimg. estudieu el metodde mpimg.imread()

Comencem! després d'importar les llibreries necessàries, llegim i mostrem la imatge seleccionada.

```
In [19]: %matplotlib inline
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

imatge= plt.imread ('img3.jpg')
plt.imshow(imatge)
```

Out[19]: <matplotlib.image.AxesImage at 0x1e3cbdbd7f0>



Deixaré les escales dels eixos x i y , tot i q les podria eliminar amb el mètode plt.axis('off'). Les imatges es guarden com a arrays, en aquest cas, un 3D array, ja q és en color. Cada dimensió correspon a un color Red [0] Green [1] Blue[2]

```
In [20]: print(type(imatge))

print(imatge.shape)
```

<class 'numpy.ndarray'>
(1600, 1200, 3)

si fem un print image, podem veure l'array.

```
In [21]: print(imatge)

[[[ 58  58  56]
 [ 58  58  56]
 [ 58  58  56]
 ...
 [194 210 226]
 [174 190 206]
 [174 190 206]]

 [[ 58  58  56]
 [ 58  58  56]
 [ 58  58  56]
 ...
 [166 182 198]
 [175 191 207]
 [192 208 224]]

 [[ 58  58  56]
 [ 58  58  56]
 [ 58  58  56]
 ...
 [181 197 213]
 [188 204 220]
 [174 190 206]]

 ...

 [[ 13  18  12]
 [ 12  17  11]
 [ 10  15  9]
 ...
 [ 66  68  63]
 [ 66  68  63]
 [ 66  68  63]]

 [[ 21  26  20]
 [ 19  24  18]
 [ 16  21  15]
 ...
 [ 66  68  63]
 [ 66  68  63]
 [ 66  68  63]]

 [[ 27  32  26]
 [ 25  30  24]
 [ 21  26  20]
 ...
 [ 66  68  63]
 [ 66  68  63]
 [ 66  68  63]]]
```

Exercici 8

Mostreu-me a veure que passa quan eliminem el canal G Verd o B Blau.

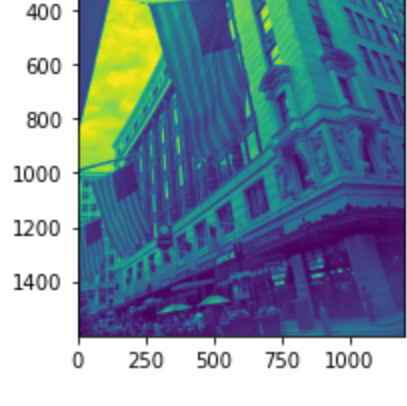
Mostreu-me a veure què passa quan eliminem el canal G Verd o B Blau. Hauries d'utilitzar la indexació per seleccionar el canal que voleu anul·lar.

Utilitzar el mètode, mpimg.imsave () de la llibreria importada, per guardar les imatges modificades i que haureu de pujar al vostre repositori a github.

Eliminaré un dels colors, convertint el numpy array en dos dimensions enlloc de tres. no importa quin eliminem, el resultat és el mateix ja q matplotlib afegeix un colormap de manera automàtica.

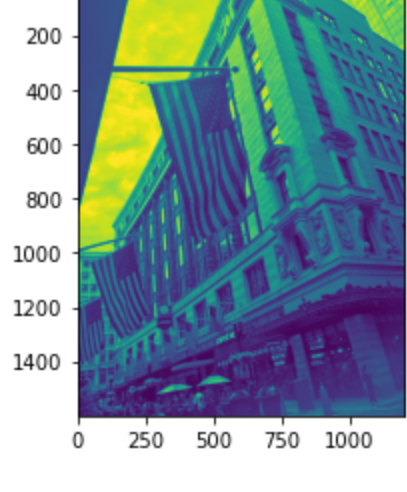
```
In [22]: imatge= plt.imread ('img3.jpg')
imatge0= imatge[:, :, 0]
plt.imshow(imatge0)
```

Out[22]: <matplotlib.image.AxesImage at 0x1e3cc3aafd0>



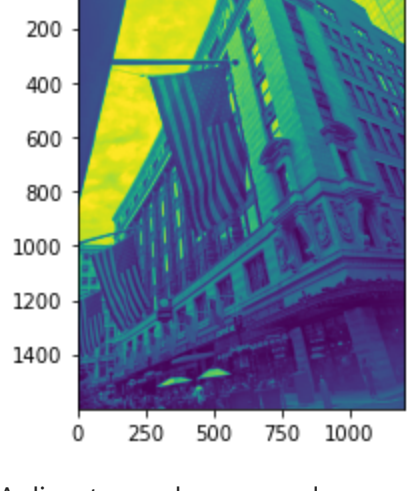
```
In [23]: imatge= plt.imread ('img3.jpg')
imatge0= imatge[:, :, 1]
plt.imshow(imatge0)
```

Out[23]: <matplotlib.image.AxesImage at 0x1e3cc5eff70>



```
In [24]: imatge= plt.imread ('img3.jpg')
imatge0= imatge[:, :, 2]
plt.imshow(imatge0)
```

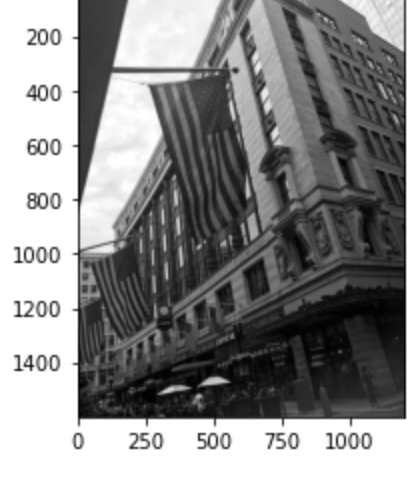
Out[24]: <matplotlib.image.AxesImage at 0x1e3cc648fa0>



Aplicant un colormap podem modificar la fotografia i convertir-la en escala de grisos, o aplicar-hi un altre efecte de color.

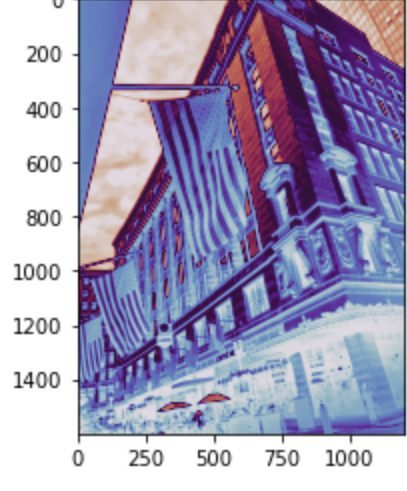
```
In [25]: plt.imshow(imatge0, cmap='gray')
```

Out[25]: <matplotlib.image.AxesImage at 0x1e3cc6a3d00>



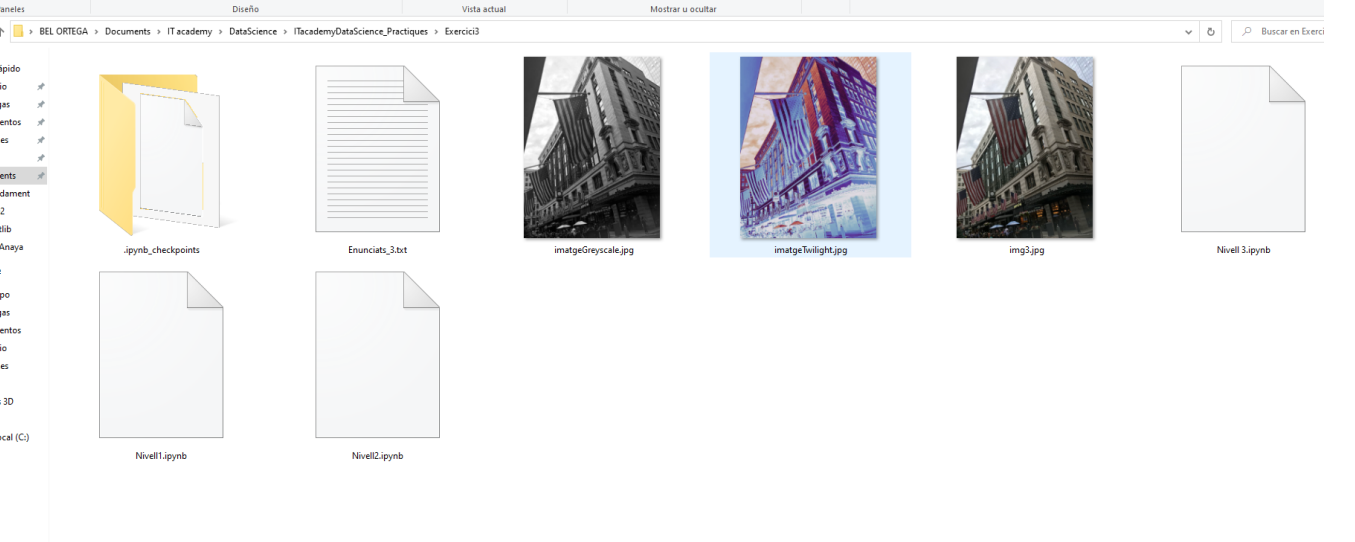
```
In [28]: plt.imshow(imatge0, cmap='twilight')
```

Out[28]: <matplotlib.image.AxesImage at 0x1e3cd4d2b80>



ara, finalment, guardarem les imatges, al directori on estem treballant, amb la funció plt.imsave()

```
In [27]: plt.imsave('imatgeGreyscale.jpg',imatge0, cmap='gray')
plt.imsave('imatgeTwilight.jpg',imatge0, cmap='twilight')
```



```
In [ ]:
```