

Process and Display Images Using Python Imaging Library

```
In [1]: import numpy as np
```

```
In [2]: """
    Every image is converted to pixels, range of a pixel is 0,255
    Image is represented in the form of arrays
"""
arr1 = np.ones((5,5),dtype = int)
arr1
```

```
Out[2]: array([[1, 1, 1, 1, 1],
               [1, 1, 1, 1, 1],
               [1, 1, 1, 1, 1],
               [1, 1, 1, 1, 1],
               [1, 1, 1, 1, 1]])
```

```
In [3]: arr1 * 255
```

```
Out[3]: array([[255, 255, 255, 255, 255],
               [255, 255, 255, 255, 255],
               [255, 255, 255, 255, 255],
               [255, 255, 255, 255, 255],
               [255, 255, 255, 255, 255]])
```

```
In [4]: import matplotlib.pyplot as plt
# matplotlib is used for visualization
```

```
In [5]: %matplotlib inline
```

```
In [6]: from PIL import Image # PIL - Python Imaging Library
```

```
In [28]: macaw = Image.open(r'████████\Macaw.jpeg')
```

```
In [29]: type(macaw)
```

```
Out[29]: PIL.JpegImagePlugin.JpegImageFile
```

In [30]: macaw

Out[30]:



Image credits - <https://www.pexels.com/photo/photo-of-yellow-and-blue-macaw-with-one-wing-open-perched-on-a-wooden-stick-2317904/> (<https://www.pexels.com/photo/photo-of-yellow-and-blue-macaw-with-one-wing-open-perched-on-a-wooden-stick-2317904/>), *Ilo Frey*

In [10]: `macaw_arr = np.asarray(macaw)`
`macaw_arr`

Out[10]: `array([[[27, 16, 22],
[27, 16, 22],
[27, 16, 22],
...,
[27, 16, 22],
[27, 16, 22],
[27, 16, 22]],
[[27, 16, 22],
[27, 16, 22],
[27, 16, 22],
...,
[27, 16, 22],
[27, 16, 22],
[27, 16, 22]],
[[27, 16, 22],
[27, 16, 22],
[27, 16, 22],
...,
[27, 16, 22],
[27, 16, 22],
[27, 16, 22]],
...,
[[32, 21, 27],
[32, 21, 27],
[32, 21, 27],
...,
[27, 16, 22],
[27, 16, 22],
[27, 16, 22]],
[[32, 21, 27],
[32, 21, 27],
[32, 21, 27],
...,
[27, 16, 22],
[27, 16, 22],
[27, 16, 22]],
[[33, 22, 28],
[33, 22, 28],
[33, 22, 28],
...,
[27, 16, 22],
[27, 16, 22],
[27, 16, 22]]], dtype=uint8)`

In [11]: `type(macaw_arr)`

Out[11]: `numpy.ndarray`

```
In [12]: plt.imshow(macaw_arr)
# plt.show(array_name) based on the new versions
```

```
Out[12]: <matplotlib.image.AxesImage at 0x2c040f68190>
```



```
In [31]: macaw_arr.shape
# pixel rows, columns, channels (red, green, blue)
```

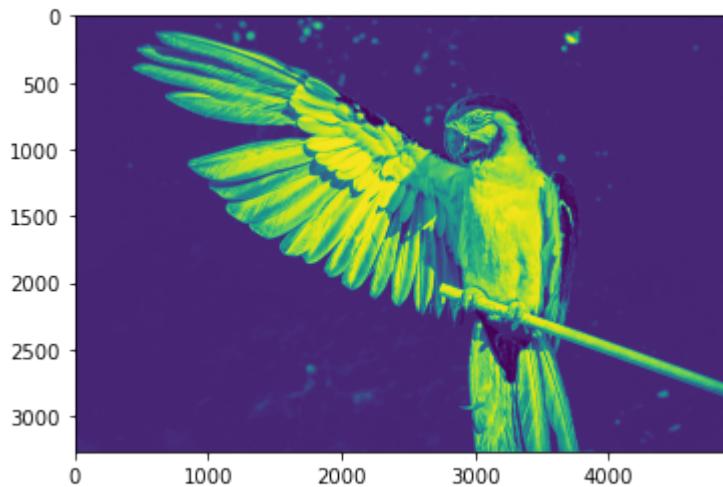
```
Out[31]: (3264, 4896, 3)
```

```
In [16]: macaw_red = macaw_arr.copy()  
macaw_red  
# Creating a copy of the original image
```

```
Out[16]: array([[[27, 16, 22],  
                  [27, 16, 22],  
                  [27, 16, 22],  
                  ...,  
                  [27, 16, 22],  
                  [27, 16, 22],  
                  [27, 16, 22]],  
  
                 [[[27, 16, 22],  
                  [27, 16, 22],  
                  [27, 16, 22],  
                  ...,  
                  [27, 16, 22],  
                  [27, 16, 22],  
                  [27, 16, 22]],  
  
                 [[[27, 16, 22],  
                  [27, 16, 22],  
                  [27, 16, 22],  
                  ...,  
                  [27, 16, 22],  
                  [27, 16, 22],  
                  [27, 16, 22]],  
  
                 ...,  
  
                 [[[32, 21, 27],  
                  [32, 21, 27],  
                  [32, 21, 27],  
                  ...,  
                  [27, 16, 22],  
                  [27, 16, 22],  
                  [27, 16, 22]],  
  
                 [[[32, 21, 27],  
                  [32, 21, 27],  
                  [32, 21, 27],  
                  ...,  
                  [27, 16, 22],  
                  [27, 16, 22],  
                  [27, 16, 22]],  
  
                 [[[33, 22, 28],  
                  [33, 22, 28],  
                  [33, 22, 28],  
                  ...,  
                  [27, 16, 22],  
                  [27, 16, 22],  
                  [27, 16, 22]]], dtype=uint8)
```

```
In [32]: plt.imshow(macaw_red[:, :, 0])
```

```
Out[32]: <matplotlib.image.AxesImage at 0x2c04dd11130>
```



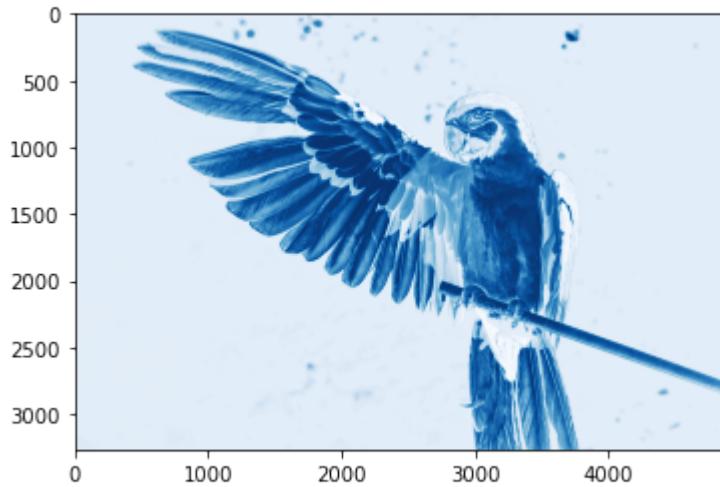
```
In [33]: plt.imshow(macaw_red[:, :, 0], cmap = 'gray')  
# Matplotlib documentation - Colormap reference
```

```
Out[33]: <matplotlib.image.AxesImage at 0x2c04e10e190>
```



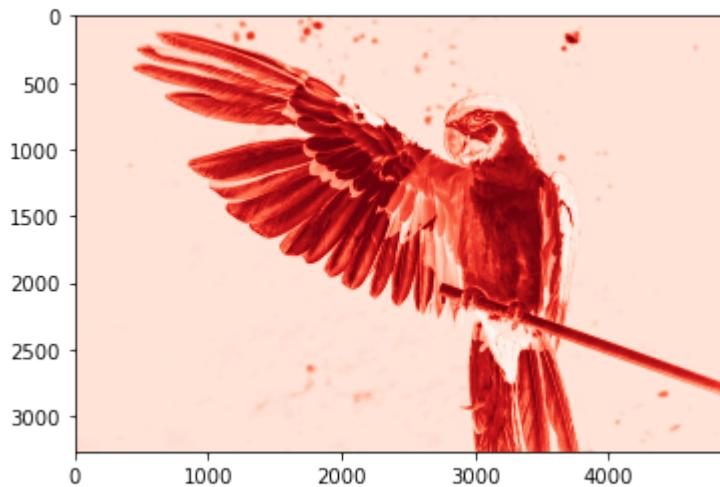
```
In [21]: plt.imshow(macaw_red[:, :, 0], cmap = 'Blues')
# Matplotlib documentation - Colormap reference
```

Out[21]: <matplotlib.image.AxesImage at 0x2c04dec03d0>



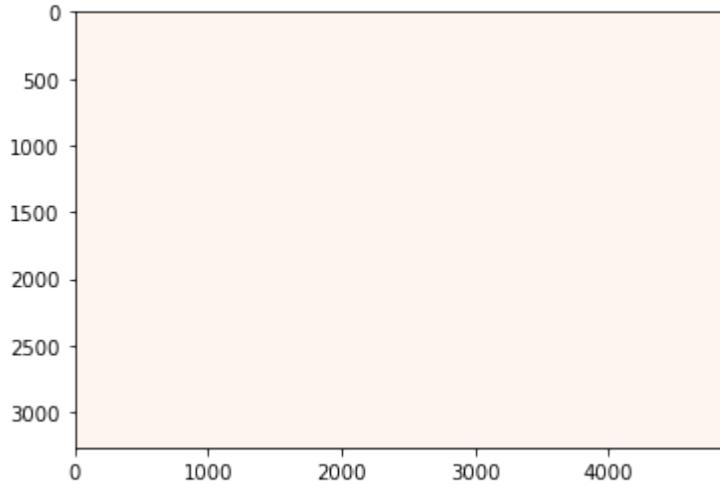
```
In [34]: plt.imshow(macaw_red[:, :, 0], cmap = 'Reds')
# Matplotlib documentation - Colormap reference
```

Out[34]: <matplotlib.image.AxesImage at 0x2c04e15e3a0>



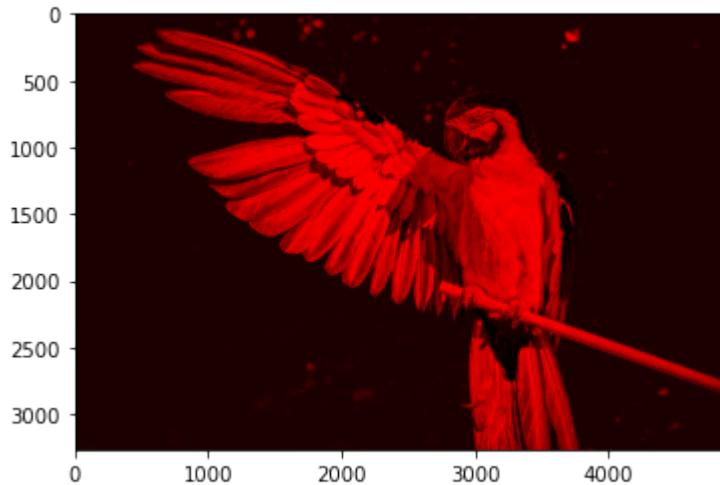
```
In [27]: plt.imshow(macaw_red[:, :, 1], cmap = 'Reds')
# Matplotlib documentation - Colormap reference
```

```
Out[27]: <matplotlib.image.AxesImage at 0x2c04e0a4610>
```



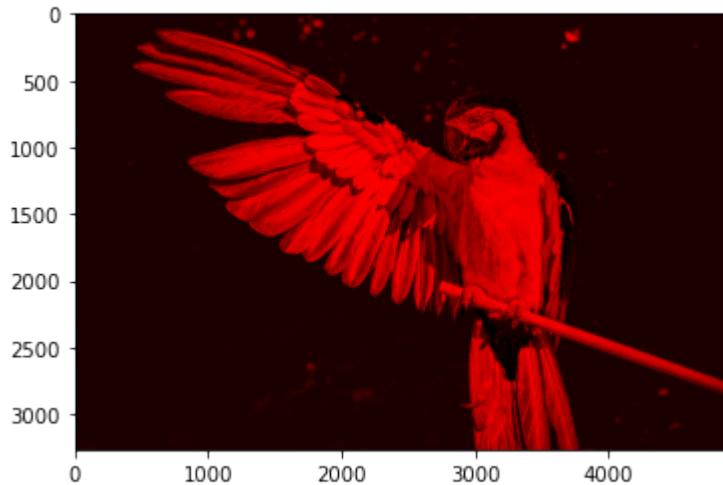
```
In [35]: macaw_red[:, :, 1] = 0
plt.imshow(macaw_red)
```

```
Out[35]: <matplotlib.image.AxesImage at 0x2c04e1c20d0>
```



```
In [36]: macaw_red[:, :, 2] = 0  
plt.imshow(macaw_red)
```

Out[36]: <matplotlib.image.AxesImage at 0x2c04e223130>



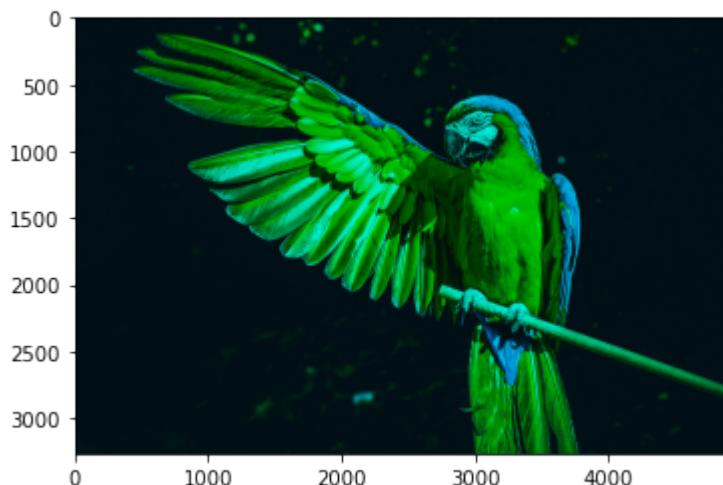
```
In [ ]: macaw_red[:, :, 2] = 0  
plt.imshow(macaw_red)
```

```
In [42]: macaw_image2 = macaw_arr.copy()
```

```
In [43]: macaw_image2[:, :, 0] = 0
```

```
In [44]: plt.imshow(macaw_image2)
```

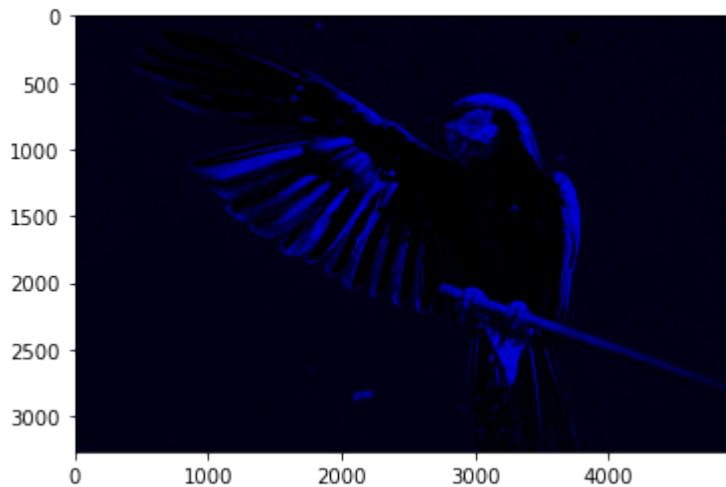
Out[44]: <matplotlib.image.AxesImage at 0x2c04e2d0a90>



```
In [45]: macaw_image2[:, :, 1] = 0
```

```
In [46]: plt.imshow(macaw_image2)
```

```
Out[46]: <matplotlib.image.AxesImage at 0x2c04e30c1f0>
```



```
In [48]: macaw_image3 = macaw_arr.copy()
```

```
In [49]: macaw_image3[:,0,:] = 0
```

```
In [50]: macaw_image3[:,1,:] = 0
```

```
In [51]: plt.imshow(macaw_image3)
```

```
Out[51]: <matplotlib.image.AxesImage at 0x2c05f6a8850>
```



Summary

```
import numpy as np
```

Resources

- Matplotlib documentation - Colormap reference
- Image credits - <https://www.pexels.com/photo/photo-of-yellow-and-blue-macaw-with-one-wing-open-perched-on-a-wooden-stick-2317904/>
[\(https://www.pexels.com/photo/photo-of-yellow-and-blue-macaw-with-one-wing-open-perched-on-a-wooden-stick-2317904/\)](https://www.pexels.com/photo/photo-of-yellow-and-blue-macaw-with-one-wing-open-perched-on-a-wooden-stick-2317904/), Ilo Frey
- OpenCV documentation, other computer vision libraries - yolov8, yolov9 (to be used later)