



FACULTAD DE INGENIERÍA  
UNIVERSIDAD DE BUENOS AIRES

67.61 - FUNDAMENTOS MATEMÁTICOS DE LA VISIÓN ROBÓTICA

## **La vida en colores**

CAMILA SERRA (97422)

MARIANO EZEQUIEL WILLINER (83469)

2 de mayo de 2021

## Introducción

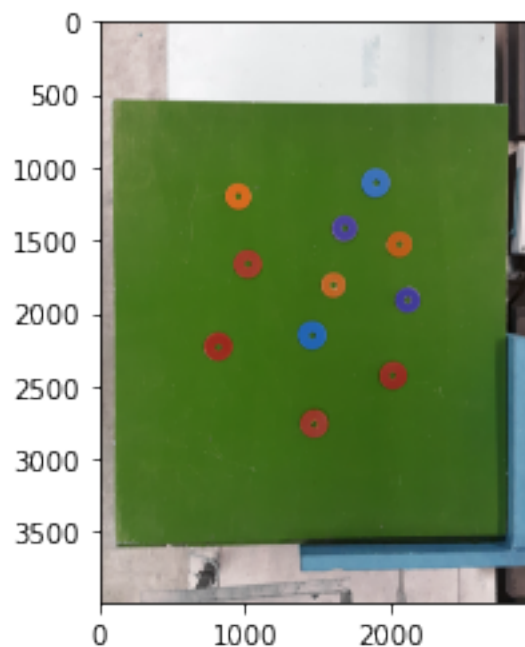
El objetivo de este trabajo es generar máscaras para recuperar cada grupo de colores (azules, rojas) sobre una imagen suministrada de arandelas. Bonus: permitir discriminar colores con sus distintas tonalidades (azul y celeste, rojo y naranja).

## Desarrollo

Modificamos la imagen para llevarla a HSV. Luego, definimos los 4 colores que queremos separar (azul y celeste, rojo y naranja). Utilizamos la función `inRange`<sup>1</sup> a la cual le pasamos la imagen, el rango inferior y el rango superior de color. Esto devuelve un vector binario mostrando si ese pixel está dentro el rango. Luego, utilizando la función `bitwise_and`<sup>2</sup> generamos una imagen que contenga solo esos pixeles.

```
[1]: %matplotlib inline
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
```

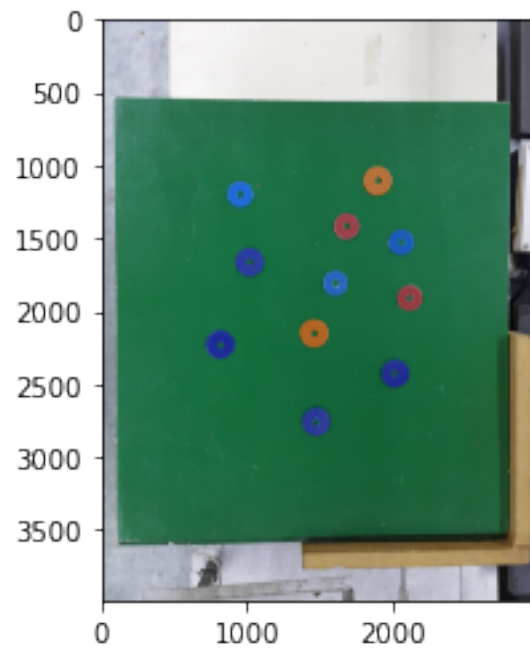
```
[2]: original_image = cv.imread('washers.jpg')
plt.imshow(original_image)
plt.show()
```



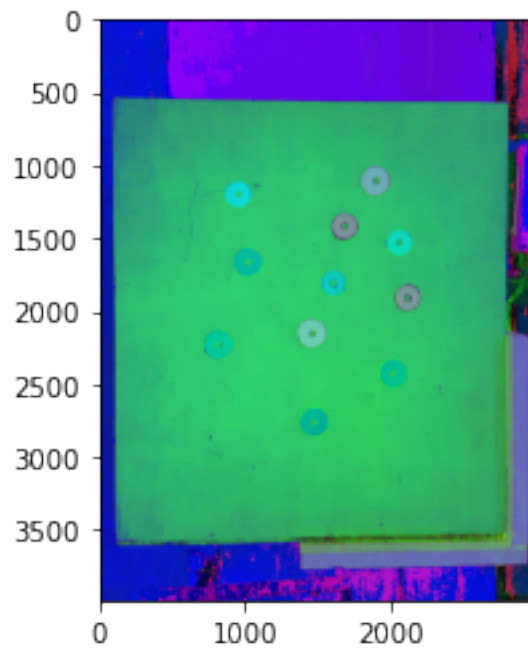
<sup>1</sup>[https://docs.opencv.org/3.4/d2/de8/group\\_\\_core\\_\\_array.html#ga48af0ab51e36436c5d04340e036ce981](https://docs.opencv.org/3.4/d2/de8/group__core__array.html#ga48af0ab51e36436c5d04340e036ce981)

<sup>2</sup>[https://docs.opencv.org/master/d2/de8/group\\_\\_core\\_\\_array.html#ga60b4d04b251ba5eb1392c34425497e14](https://docs.opencv.org/master/d2/de8/group__core__array.html#ga60b4d04b251ba5eb1392c34425497e14)

```
[3]: rgb_image = cv.cvtColor(original_image, cv.COLOR_BGR2RGB)
plt.imshow(rgb_image)
plt.show()
```



```
[4]: hsv_image = cv.cvtColor(rgb_image, cv.COLOR_BGR2HSV)
plt.imshow(hsv_image)
plt.show()
```

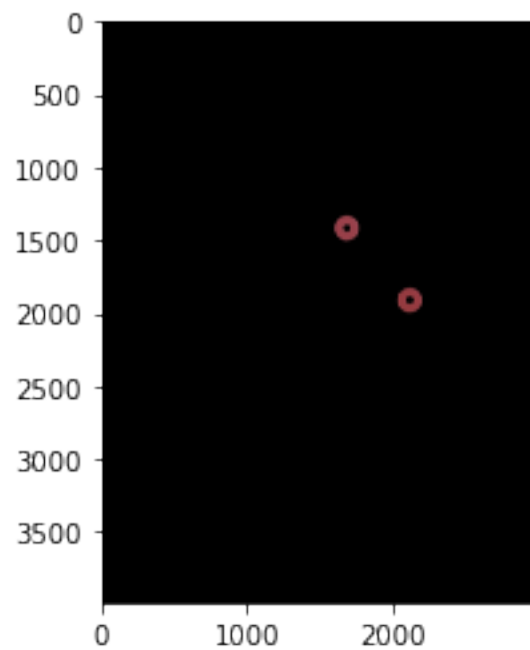


```
[5]: colors = {
    "red": {
        "lower_bound": (119,100,100),
        "upper_bound": (125,255,255),
    },
    "blue": {
        "lower_bound": (2,100,100),
        "upper_bound": (8,255,255),
    },
    "orange": {
        "lower_bound": (106,100,100),
        "upper_bound": (108,255,255),
    },
    "light_blue": {
        "lower_bound": (12,100,100),
        "upper_bound": (15,255,255),
    }
}
```

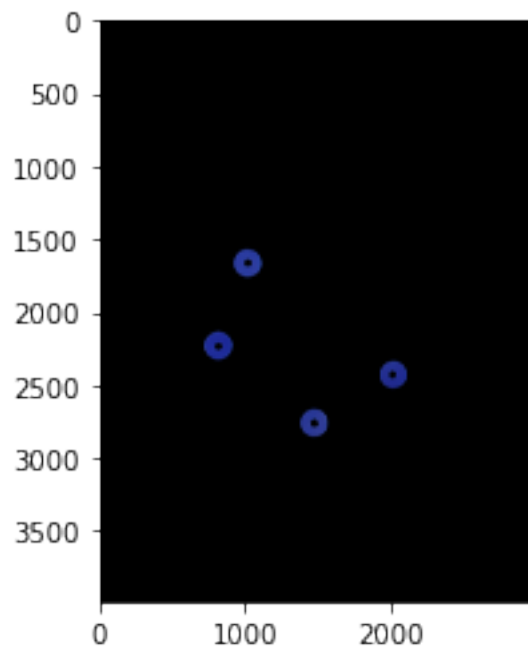
```
[6]: for key in colors:
      mask = cv.inRange(hsv_image, colors[key]["lower_bound"],
      ↪ colors[key]["upper_bound"])
      masked_image = cv.bitwise_and(rgb_image, rgb_image, mask=mask)

      print(key)
      plt.figure(key)
      plt.imshow(masked_image)
      plt.show()
```

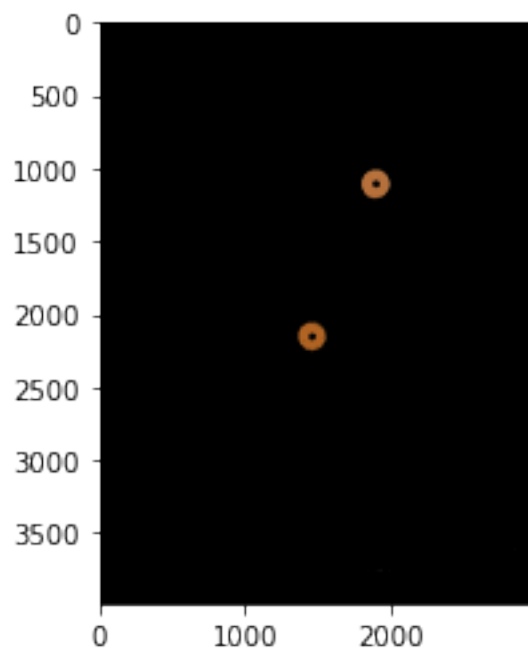
red



blue



orange



light\_blue

