

## Exercício - OpenCV

Usando a imagem `boia.jpg`, execute os passos descritos abaixo. O objetivo final é encontrar uma forma de isolar a boia vermelha do plano de fundo (background)

1. Isole os canais vermelho e verde e salve as imagens com os nomes **1\_red.jpg** e **2\_green.jpg**, respectivamente.
2. Cria uma máscara binária correspondendo a área da boia, usando a técnica de *thresholding*. Lembrando: uma máscara binária é uma imagem de 1 canal. *Dica: use como entrada do threshold uma imagem representada pela razão entre os canais. Lembrando que o objeto é vermelho, portanto a máscara a ser calculada deve ser a máscara do canal vermelho.* Salve o resultado da máscara como **3\_red\_mask.jpg**.
3. Use uma operação morfológica para preencher os buracos da máscara. Para isso, crie um elemento estruturante e ache um bom tamanho para ele. Salve o resultado como **4\_closed.jpg**.
4. Encontre o contorno da boia usando a máscara. *Lembrando: usando o método de encontrar contornos do OpenCV, o resultado será um array de contornos, devido a imperfeição da imagem.* Você deve encontrar o contorno da boia. Após isso, desenhar o contorno da boia e salvar a imagem como **5\_contour.jpg**.
5. Encontre o *bounding box* da boia utilizando o contorno calculado previamente. Crie uma imagem representando a imagem restrita (*crop*) pela área do *bounding box*. Salve como **6\_crop.jpg**.
6. Faça o mesmo passo anterior para a imagem da máscara e salve como **7\_crop\_red\_mask.jpg**.
7. Utilizando a imagem da máscara “*croppada*”, inverta-a (salve como **8\_inv\_mask.jpg**). Crie uma imagem com background branco para a boia e converta a área da boia para preto, usando a máscara invertida. *Dica: use conjunção bit a bit entre o background branco, setando a máscara invertida como parâmetro mask.* Salve o resultado como **9\_white\_bg.jpg**.
8. Por fim, use a máscara calculada previamente para adicionar a boia no *background* branco. Salve como **10\_final.jpg**. O resultado deve ser semelhante a imagem abaixo:



9. Calcule os momentos da imagem final. *Lembrando que a imagem deve ter apenas 1 canal.* O resultado deve ser semelhante ao informado abaixo:

```
{'m00': 161873.0, 'm10': 22824133.0, 'm01': 46214518.0, 'm20': 4298568279.0, 'm11':  
6516261438.0, 'm02': 17607754038.0, 'm30': 910760417731.0, 'm21':  
1227237600530.0, 'm12': 2482698503358.0, 'm03': 7547144478988.0, 'mu20':  
1080359885.9901156, 'mu11': 2980.0552282333374, 'mu02': 4413572957.941412,  
'mu30': -793594.4438476562, 'mu21': 1451583.4290771484, 'mu12': -  
868607.8685302734, 'mu03': 17041139.087890625, 'nu20': 0.04123059661527771,  
'nu11': 1.1373011586220838e-07, 'nu02': 0.1684385440636819, 'nu30': -  
7.527704795489611e-08, 'nu21': 1.3769112957921543e-07, 'nu12': -  
8.23925075084138e-08, 'nu03': 1.6164511410962617e-06}
```

**O código, as imagens de cada passo e o resultado dos momentos são os artefatos do exercício que devem ser entregues.**