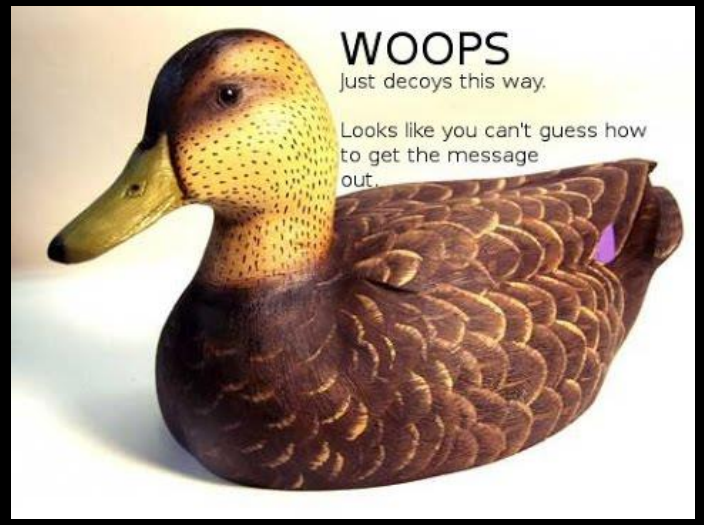
ASSIGNMENT 00 – FLAG

Noah Sherry

INPUT IMAGES

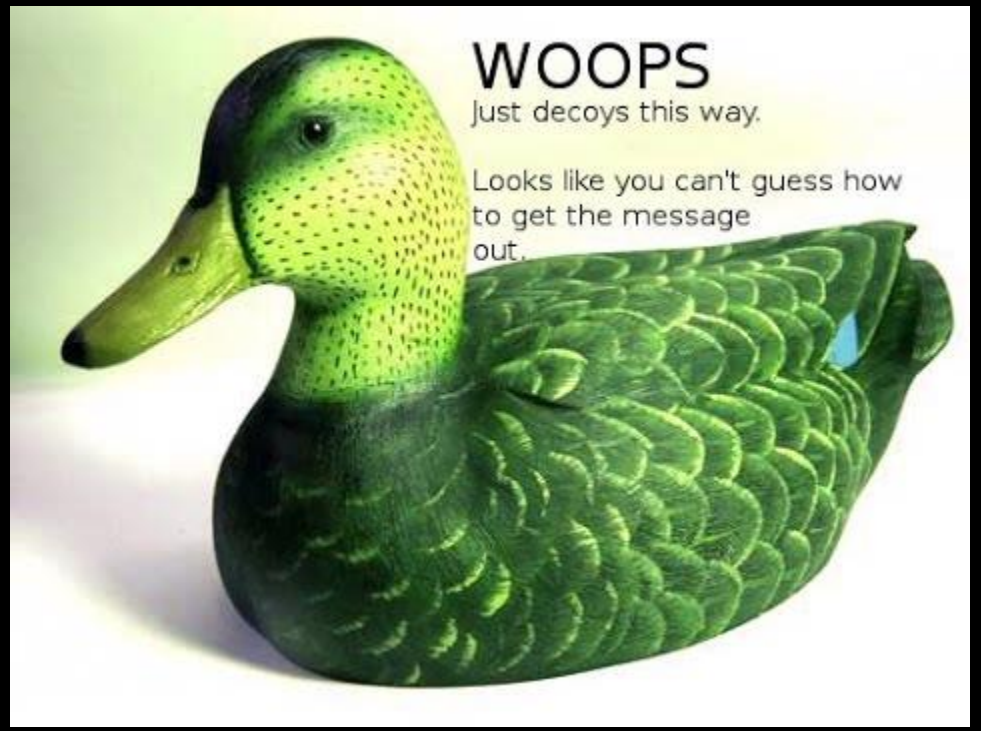


COLORS – A

- Swap Red/Green Channels

- Code:

```
def swapChannels(img,c1=0,c2=1):
    chan1=img[:,:,c1]
    chan2=img[:,:,c2]
    img_swap=img*1
    img_swap[:,:,c2]=chan1
    img_swap[:,:,c1]=chan2
```



COLORS - B

- Extract blue channel, save as monochrome
- Code:

```
def extractChannel(img,chan):
    ext=img[:,:,chan]
```

COLORS - C

- Invert green channel & preserve others
- Code:

```
def invertChannel(img,chan):
    inv = img*1
    inv[:,:,chan] = 255 - img[:,:,chan]
```

COLORS - D

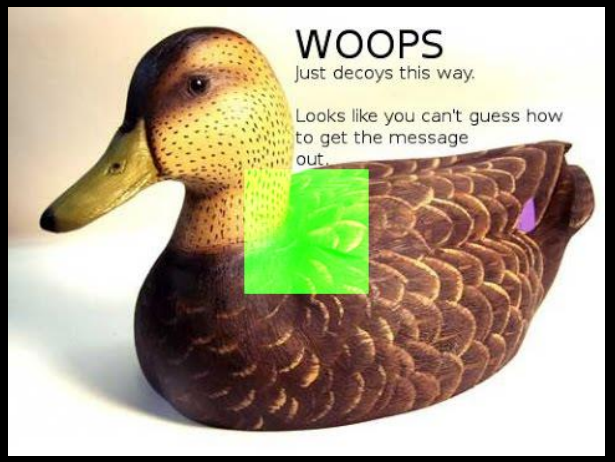
- Add 100 to every value
- Q1: Does uint8 allow overflow?
  - Yes, max integer for uint8 is 255, values greater will overflow

```
def addHundred(img):
    copy=img*1.0
    copy += 100
    copy[copy>255] = 255
```

COPY & PASTE - A

- Replace green channel with 255 on 100 x 100 center region
- Code:

```
def centerAdd(img,chan):
    cen = img*1
    h,w = img.shape[:2]
    cen[int(w/2)-50:int(w/2)+50,int(h/2)-50:int(h/2)+50,chan] = 255
```

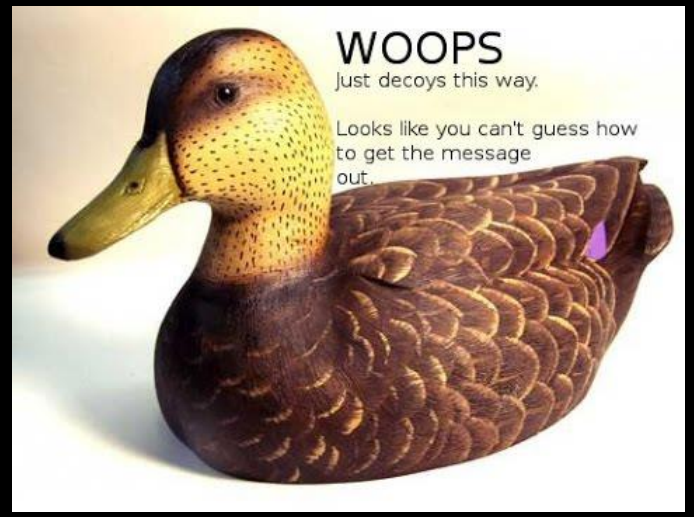


COPY & PASTE - B

- Paste centered 100 x 100 region of one image onto another
- Code:

```
copy1 = img1*1
    copy2 = img2*1
    w1,h1 = img1.shape[:2]
    w2, w2 = img2.shape[:2]
    copy2[int(w2 / 2) - 50:int(w2 / 2) + 50, int(h2 / 2) -
50:int(h2 / 2) + 50,:] = copy1[int(w1/2)-
50:int(w1/2)+50,int(h1/2)-50:int(h1/2)+50,:]
```

STATS – IMAGE 1



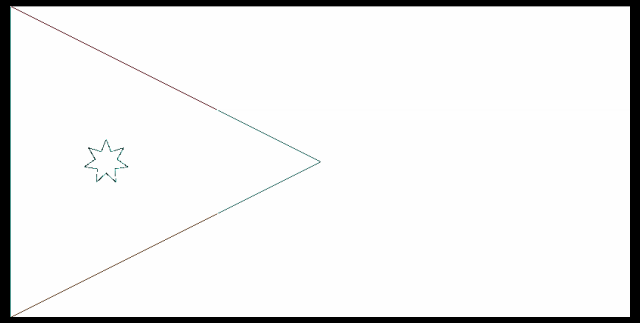
- Used Numpy functions
- Total Pixels = 172800
- Min Intensity Value = 0
- Max Intensity Value = 255
- Standard Deviation = 85.5666180393
- Mean Intensity Value = 168.845652006
- Q1: Numbers were expected. Both pure white and black in image
- Q2: Standard Deviation means how diverse are the intensities across the image

FLAG

- OG Flag:

- My Flag:

- Diff:



Lines around triangle and star, due to anti-aliasing in the original .png causing the pixel-perfect recreation to not match perfectly

ASSIGNMENT 00 – FLAG

Noah Sherry