Assignment0 / {Flag}

Graphics Programming
Tristan Goodell

Input Images

image1.png

“Banff National Park”
512x341

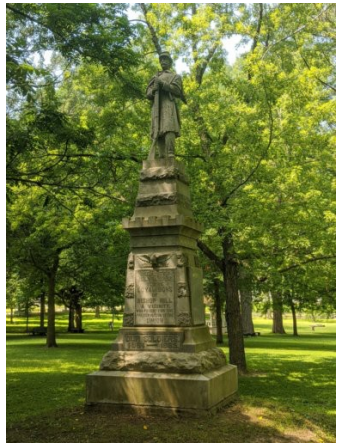
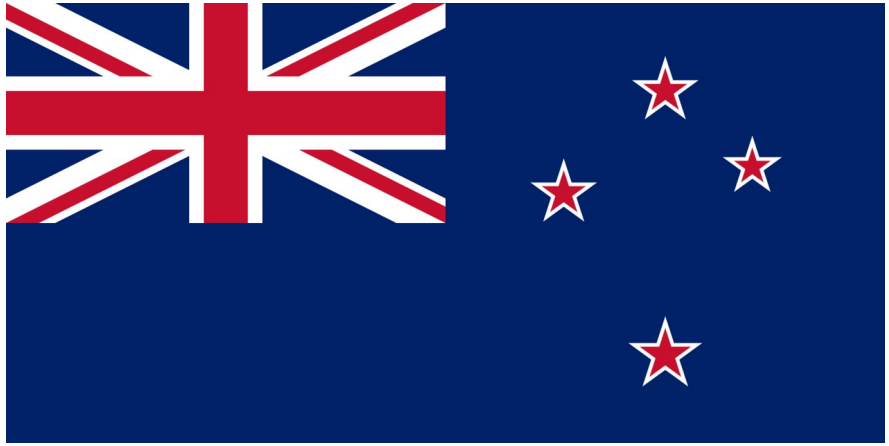


image2.png

“Bishop Hill”
384x512



flag.png

“Flag of New Zealand”
1600x800

Colors / {Pic_1a}

- *Task:* Switch the red and green channels of image1.

image1.png

pic_1_a.png

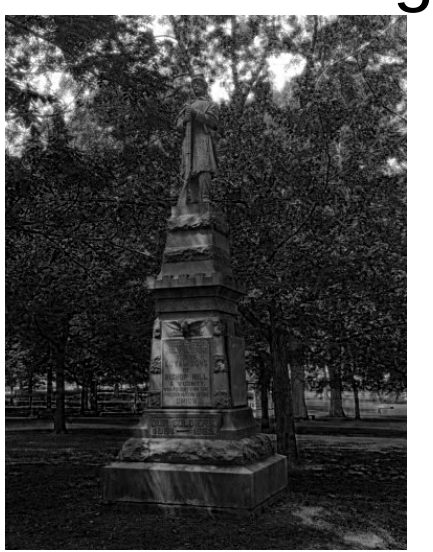
Colors / {Pic_1a} / [code]

```
1 # Switch red and green channels in image1
2 image1=cv2.imread("input/image1.png")
3 red=image1[:,:,2]
4 green=image1[:,:,1]
5 image1_gr_switch=image1[:,:,:]
6 image1_gr_switch[:,:,1]=red
7 image1_gr_switch[:,:,2]=green
8 cv2.imwrite("output/pic_1_a.png",image1_gr_switch)
```

Colors / {Pic_1b}

- *Task:* Extract the blue channel from image2.

image2.png



pic_1_b.png

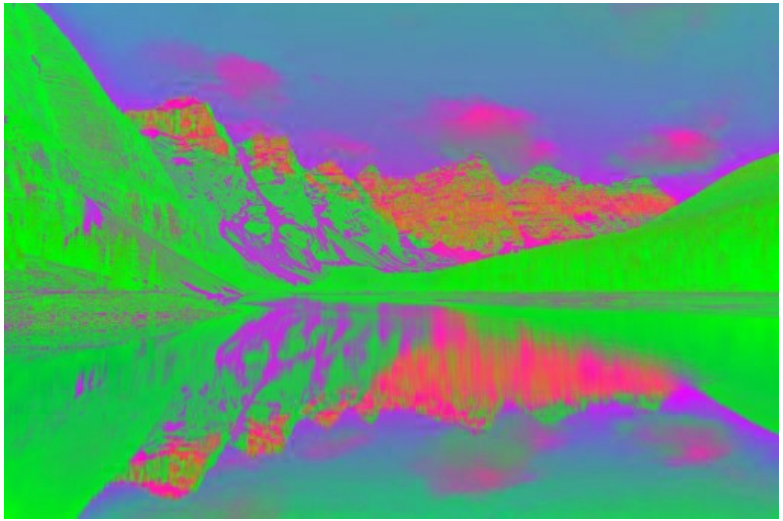
Colors / {Pic_1b} / [code]

```
1  # Extract blue channel to become grayscale in image2
2  image2=cv2.imread("input/image2.png")
3  blue=image2[:,:,0]
4  cv2.imwrite("output/pic_1_b.png",blue)
```

Colors / {Pic_1c}

- *Task:* Invert the green channel in image1.

image1.png



pic_1_c.png

Colors / {Pic_1c} / [code]

```
1  # Invert the green channel of image1 without touching the other channels
2  image1=cv2.imread("input/image1.png")
3  red=image1[:,:,2]
4  blue=image1[:,:,0]
5  green=255-image1[:,:,1]
6  image1_gi_switch=image1[:,:,:]
7  image1_gi_switch[:,:,2]=red
8  image1_gi_switch[:,:,0]=blue
9  image1_gi_switch[:,:,1]=green
10 cv2.imwrite("output/pic_1_c.png",image1_gi_switch)
```

Colors / {Pic_1d}

- *Task:* Add 100 to every color value in image2.

image2.png

pic_1_d.png

Colors / {Pic_1d} / [code]

```
1  # Add 100 to each channel of image2
2  image2=cv2.imread("input/image2.png")
3  image2[image2>255-100]=255
4  image2[image2<255]+=100
5  cv2.imwrite("output/pic_1_d.png",image2)
```

Colors / {Pic_1d} / [commentary]

- Does uint8 allow adding 100 to each color intensity value?
  - Yes, it allows us to do this but it causes overflow. We resolve this by automatically setting everything 155 or higher to 255 and adding 100 to every color value that is below 155.

Copy & Paste / {Pic_2a}

- *Task:* Set the green to 255 in the middle 100px of img2.

image2.png

pic_2_a.png

Copy & Paste / {Pic_2a} / [code]

```
1  # Select middle 100x100 region and max out green channel for image2
2  image2=cv2.imread("input/image2.png")
3  h,w=image2.shape[:2]
4  red=image2[:,:,2]
5  blue=image2[:,:,0]
6  nimage2=image2
7  nimage2[h*103//256:h*153//256,w*71//192:w*121//192,1]=255
8  nimage2[:,:,0]=blue
9  nimage2[:,:,2]=red
10 cv2.imwrite(“output/pic_2_a.png”,nimage2)
```

Copy & Paste / {Pic_2b}

- *Task:* Paste the middle 100px of img2 onto img1.

image1.png

image2.png

pic_2_b.png

Copy & Paste / {Pic_2b} / [code]

```
1  # Paste center 100x100 of image2 onto image1
2  image1=cv2.imread("input/image1.png")
3  image2=cv2.imread("input/image2.png")
4  paste=image2[h//2-50:h//2+50,w//2-50:w//2+50]
5
6  h,w=image1.shape[:2]
7  image1[h//2-50:h//2+50,w//2-50:w//2+50]=paste
8  cv2.imwrite("output/pic_2_b.png",image1)
```

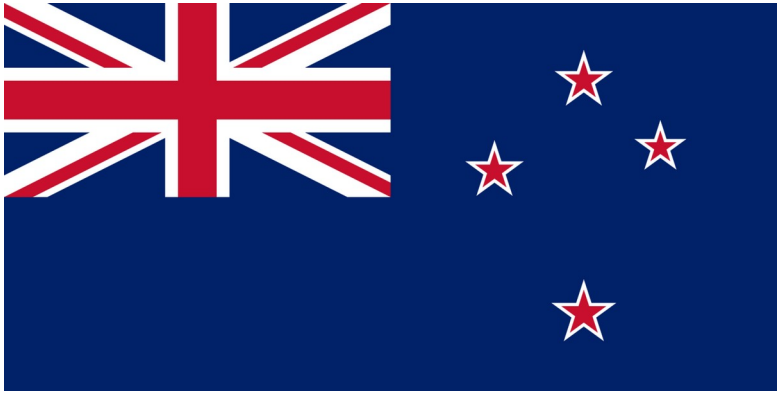
Stats / {image1}

- *Pixels:* 523,776
- *Min Intensity Value:* 0
- *Max Intensity Value:* 255
- *Stdev Intensity:* 58.833
- *Mean Intensity:* 105.064

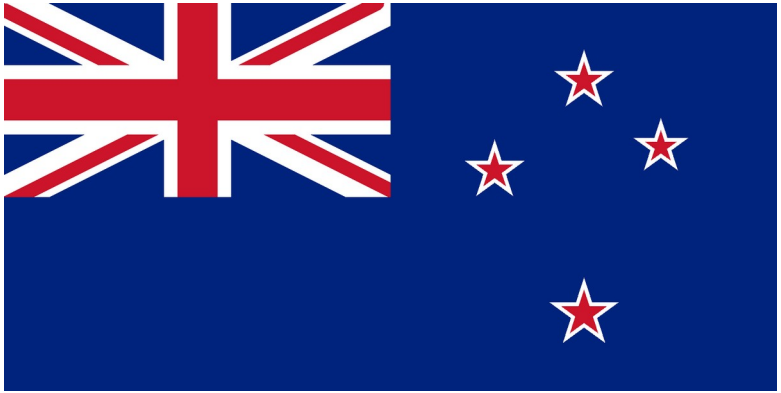
- These numbers are expected; the photo uses the full color intensity spectrum, and the mean & standard deviation are similar to other landscape photos.
- In this case, the standard deviation is the contrast of the photo.

Flag / {Pic_4a}

- *Task:* Programmatically recreate *flag.png*.



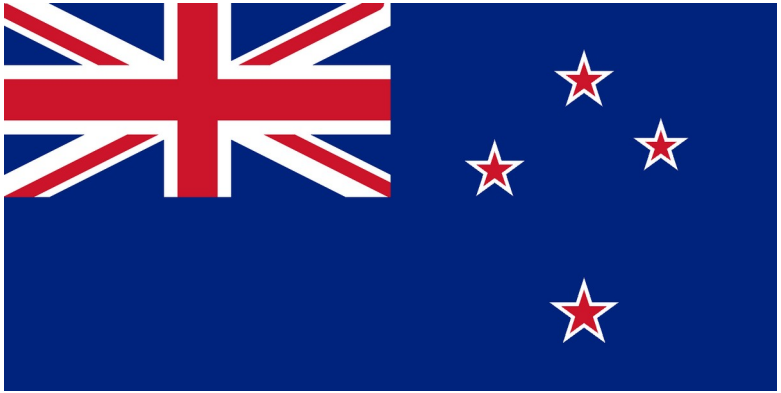
flag.png



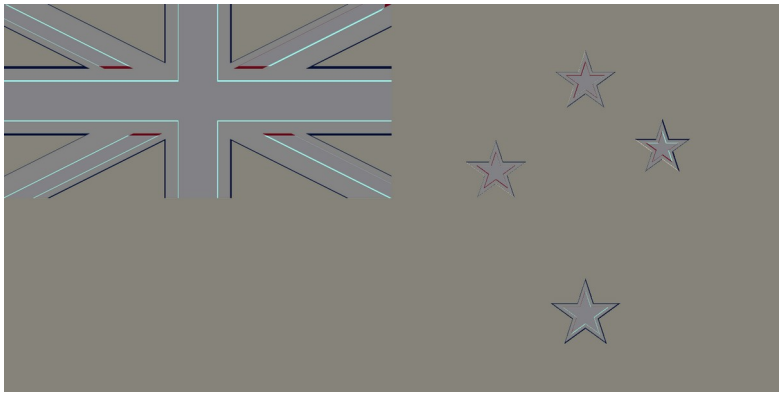
pic_4_a.png

Flag / {Pic_4b}

- *Task:* Normalize the difference between the two.



pic_4_a.png



pic_4_b.png

Flag / {Commentary}

- Where do the differences come from?
  - flag.png and pic_4_a.png have slightly different colors. This is mostly due to contradicting reports on the official colors of New Zealand. In the end, Catalina Blue, White, and Philippine Red were used.
  - There are some weird artifacts in the Union Jack caused by the diagonal lines.
  - Finally, there are a few instances where star size is off by a few pixels.