

## Pseudo Color

Input:

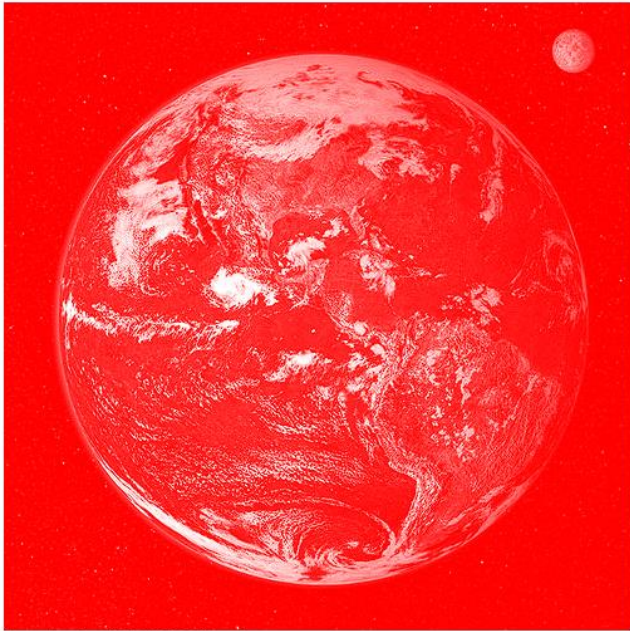
```
✓ [5] 1 img = io.imread("https://images2.pics4learning.com/catalog/e/earth_from_space.jpg") # use the above image or any of your preferred image.
3s 2 #img = io.imread("https://images2.pics4learning.com/catalog/p/prari007.jpg")
3 #image_rgb = cv.cvtColor(img, cv.COLOR_BGR2RGB)
4 img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
5
6 #img = cv.resize(img, (0,0), fx=0.7, fy=0.7)
7 # Check width, height, and dimension
8 cv2_imshow(img)
9 print(img.shape)
10 print("Grayscale")
11
```



(500, 500)

Example code:

```
[9] 1 # Prepare for 3 channels
    2 threeChannel = np.ones((img.shape[0],img.shape[1],3), dtype=np.uint8) # prepare the storage with the same size as the original image
    3
    4 threeChannel[:, :,0] *= img[:, :] # B
    5 threeChannel[:, :,1] *= img[:, :] # G
    6 threeChannel[:, :,2] *= 255 # R
    7
    8 # image_oneChannel = np.array(oneChannel, np.uint8)
    9 cv2_imshow(threeChannel)
   10 print("Pseudo color", threeChannel.shape)
   11
```



Pseudo color (500, 500, 3)

หลังจากนำเข้าไฟล์รูปและแปลงให้เป็น Greyscale เหลือ 1 channel แล้ว จาก code ตัวอย่าง จะเป็นการนำภาพให้กลับมาเป็น 3 channel RGB แล้วทำการเปลี่ยนค่าสีแต่ละ channel ดัง code บรรทัดที่ 4-5 จะเป็นการทำให้ค่า channel Blue, Green มีค่า intensity ตามรูป Greyscale แต่บรรทัดที่ 6 จะเปลี่ยนค่า channel Red คุณด้วย 255 ซึ่งเป็นค่าสูงสุด จึงทำให้รูปเป็นสีแดง

```
✓ 1s 1 # Prepare for 3 channels
2 threeChannel = np.ones((img.shape[0],img.shape[1],3), dtype=np.uint8) # prepare the storage with the same size as the original image
3
4 threeChannel[:, :, 0] *= 100 # B
5 threeChannel[:, :, 1] *= img[:, :] # G
6 threeChannel[:, :, 2] *= img[:, :] # R
7
8 # image_oneChannel = np.array(oneChannel, np.uint8)
9 cv2_imshow(threeChannel)
10 print("Pseudo color", threeChannel.shape)
11
```



Pseudo color (500, 500, 3)

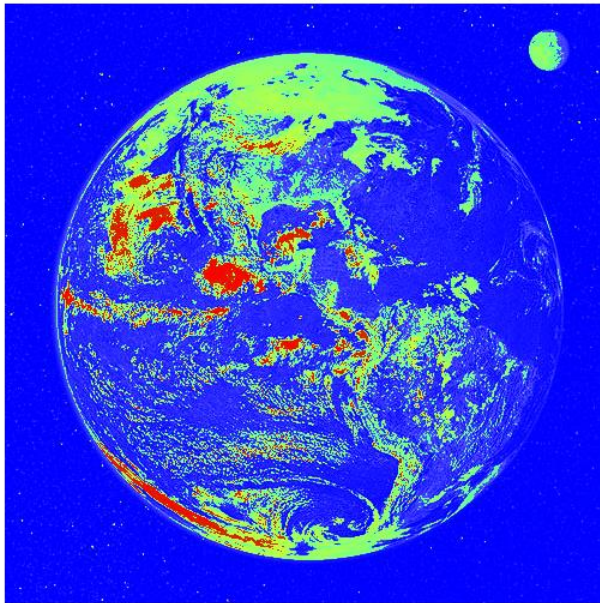
เมื่อทดลองปรับเปลี่ยนค่าที่บรรทัดที่ 4-6 ปรับให้ค่า Blue เป็นค่าคงที่ ในที่นี้เป็นค่า 100 แล้วให้ Green และ Red เป็นค่าตาม Greyscale จะเห็นว่าส่วนที่เป็นสีขาวของรูป Greyscale จะเป็นสีเหลืองจางๆ ซึ่งเป็นส่วนผสมของ Green และ Red ผสมกับ Blue ที่มีค่าเพียง 100 สีจึงไม่สว่างจนเป็นสีขาว แต่เห็นเป็นสีเหลืองเพราะค่าของ Green, Red มีค่าสูง (หรือเป็น 255) ตามค่า intensity ของรูปตั้งต้น

แต่ในส่วนที่ Greyscale เป็นพื้นหลังสีดำ ในภาพสีนี้จะกลายเป็นสีที่ค่อนข้างไปทาง Blue เนื่องจาก Green, Red มีค่าต่ำ (หรือเป็น 0) ตามรูป Greyscale แต่ Blue ที่เป็นค่าคงที่จึงมีค่ามากกว่าสี Green, Red ทำให้ pixel นั้นออกมาเป็นสี Blue

```

1 # Prepare for 3 channels
2 threeChannel = np.ones((img.shape[0],img.shape[1],3), dtype=np.uint8) # prepare the storage with the same size as the original image
3
4 for i in range(img.shape[0]):
5     for j in range(img.shape[1]):
6         intensity = img[i, j]
7         if int(intensity) in range(-1, 86):
8             threeChannel[i,j,0] *= 255 # B
9             threeChannel[i,j,1] *= img[i,j] # G
10            threeChannel[i,j,2] *= img[i,j] # R
11        elif int(intensity) in range(86, 200):
12            threeChannel[i,j,0] *= 255 - img[i,j] # B
13            threeChannel[i,j,1] *= 255 # G
14            threeChannel[i,j,2] *= img[i,j] # R
15        else:
16            threeChannel[i,j,0] *= 0 # B
17            threeChannel[i,j,1] *= 255 - img[i,j] # G
18            threeChannel[i,j,2] *= img[i,j] # R
19
20 cv2_imshow(threeChannel)
21 print("Pseudo color", threeChannel.shape)
22

```



ดังนั้นเพื่อทำการลงสีลงใน pixel ที่ต้องการ จึงได้กำหนดว่า หาก intensity มีค่าช่วง 0-85 ให้ทำเป็นสีช่วง Blue, ค่าช่วง 86-200 ให้ทำเป็นสีช่วง Green, ค่าช่วงส่วนที่เหลือเป็น Red

code เริ่มทำงานด้วยการวิ่งไล่ทีละ pixel (loop for) เช็ค่า intensity greyscale ว่ามีค่าเท่าใด แล้วตั้งค่าเปลี่ยนสีแต่ละ channel ของ pixel นั้นให้เป็นไปตามที่ต้องการ และปรับค่าตัวเลขตามความเหมาะสมและความสวยงาม



```

✓ [40] 1 # Using OpenCV's applyColorMap
1s      2 color_map = cv.applyColorMap(img, cv.COLORMAP_JET)
        3
        4 res = np.hstack((color_map, threeChannel)) #stacking images side-by-side, for a clear comparison
        5 print("\t\t\t\t\tColormap Jet\t\t\t\t\tVS\t\t\t\t\tManual map color")
        6 cv2_imshow(res)

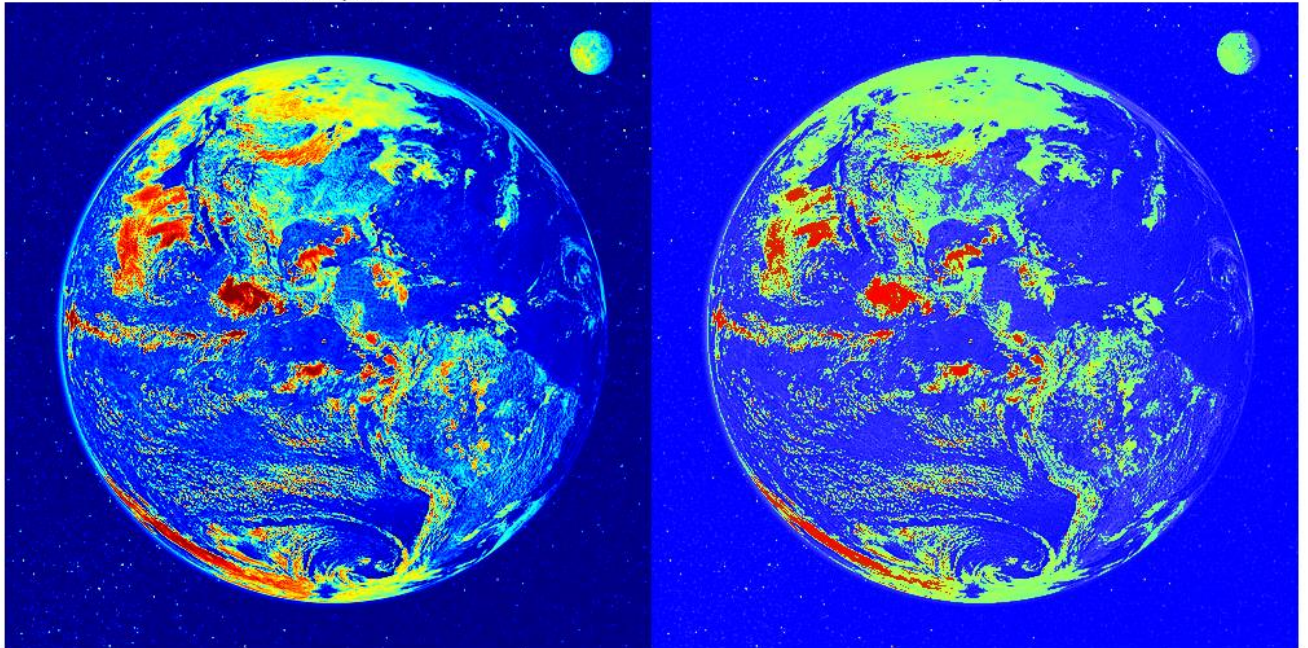
```



Colormap Jet

VS

Manual map color



OpenCV นั้นมี code สำเร็จแล้ว ชื่อว่า applyColorMap โดยสามารถเลือกชุดสีที่ต้องการ map ได้หลายชุดสี ในที่นี้คือ Colormap Jet ซึ่ง code manual map color ที่ใช้การไล่ปรับสีไปทีละ pixel ได้กำหนดช่วงค่าการปรับสีโดยมี Colormap Jet เป็นตัวอย่าง จึงได้ผลที่มีความใกล้เคียงดังภาพ