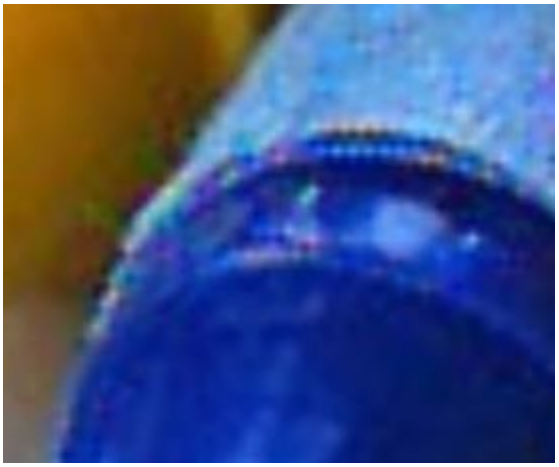
Name : Haoyuan Zhong

Part-1 : Linear Interpolation

1) Insert your linear interpolated test image(hope.jpg) here:

2) Display the map/plot of all the 3 training images here:

3) Post close-up of any artifacts you came across.



4) Average_per_pixel error and Max_pixel_error for each of 3 training images :

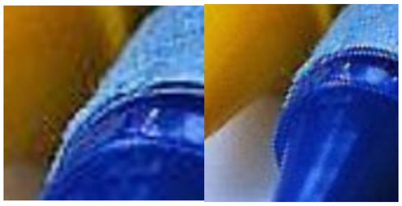
Image	Average_per_pixel_error	Max_pixel_error
Crayons	143.5	53633
Tony	23.7	9640
Iceberg	105.2	30186

Part-2 : Freeman Method

5) Insert your Freeman Method test image(hope.jpg) here:

6) Display the map/plot of all the 3 training images here:

7) Post close-up of any artifacts you came across.

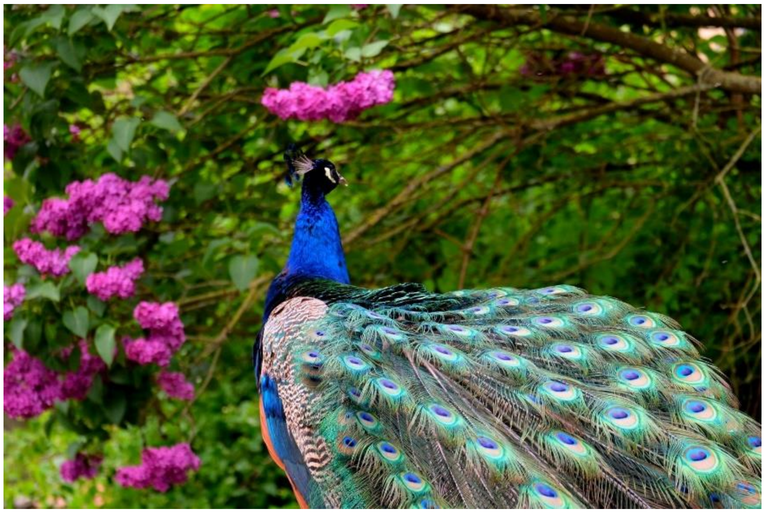


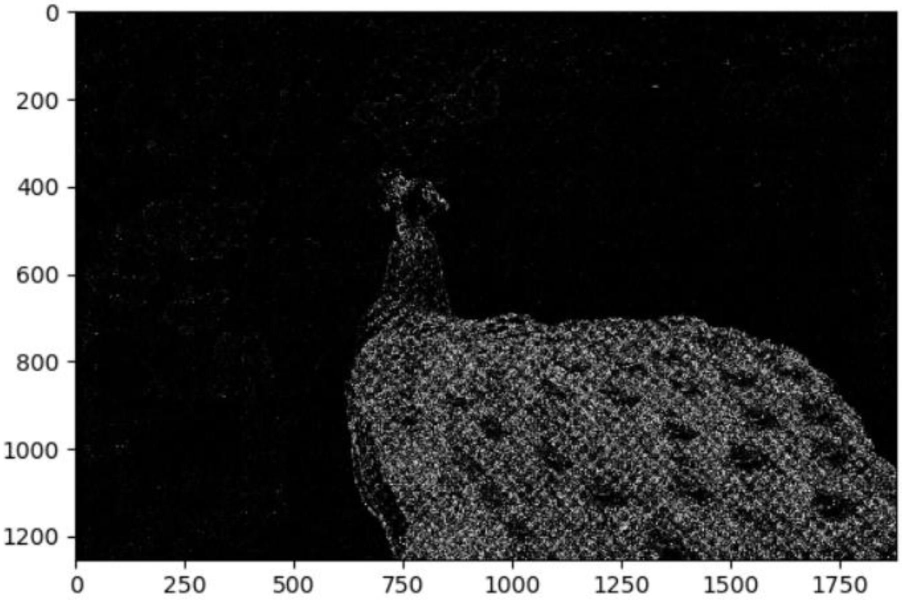
8) Average_per_pixel error and Max_pixel_error for each of 3 training images :

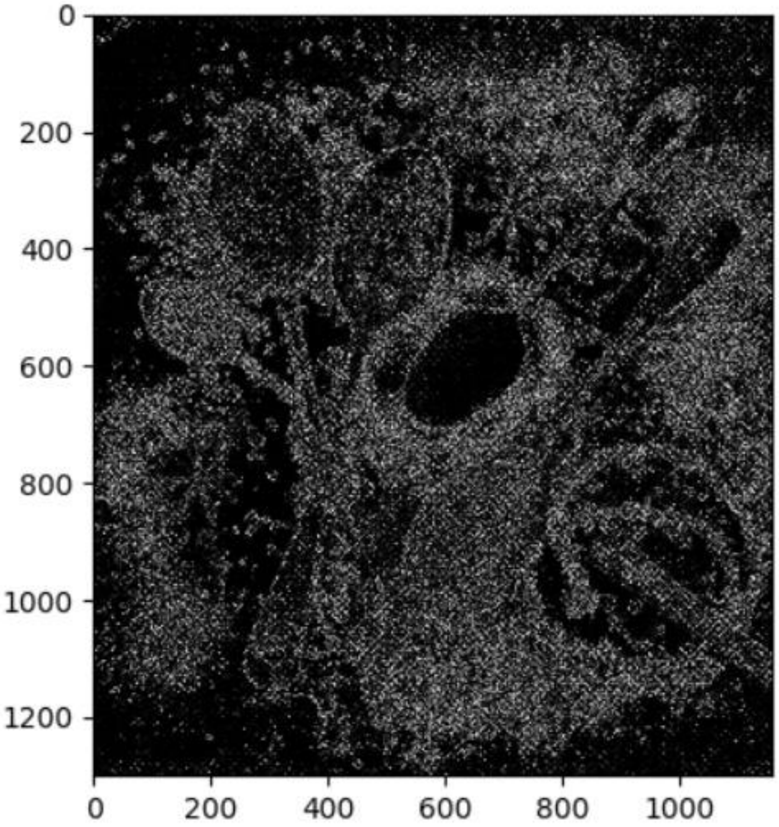
Image	Average_per_pixel_error	Max_pixel_error
Crayons	100.5	48173
Tony	15.6	10753
Iceberg	66.5	34075

Part-3 : Images of your choice

1) Post 2 images your choice here and the corresponding error maps of your outputs with the Freeman method.

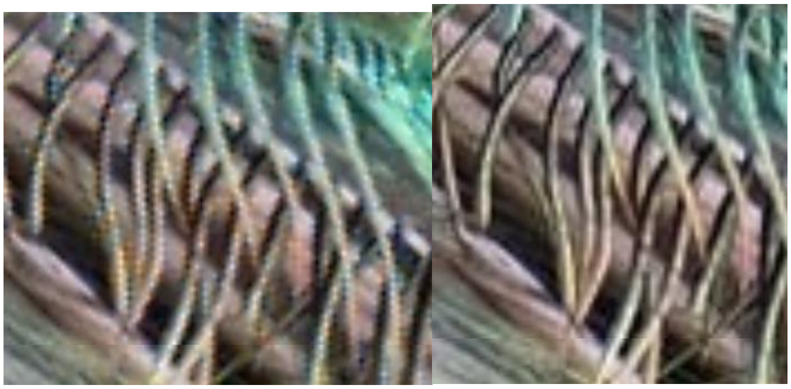



0
200
400
600
800
1000
1200
0
250
500
750
1000
1250
1500
1750



2) Any image that breaks the method and why do you think so?

Both images perform worse, especially in details.



Left is the recover image, which has many obvious blue spots compared to the right image. The reason is both these pictures has high contrast and more complex color variance. Apparently, just use the average of nearby pixels won't fit very well.

This algorithm can perform better with images which have large continuous area in similar color. That is to say, low contrast images. Moreover, it should perform better if green is dominant in an image because it has as twice sample rate as the red and blue pixels.

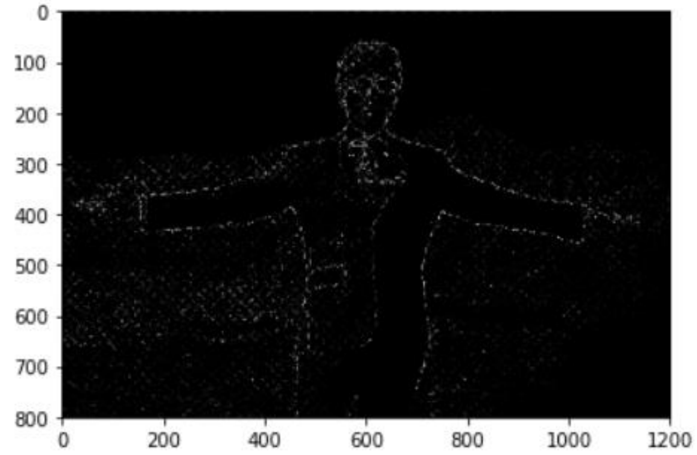
Part-4 : Bonus

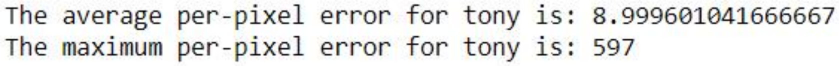
Post any extra credit details/images/references used here.

Instead use the algorithm described in the course website, which performs a medium average to all red and blue pixels, I tweak it a little bit.

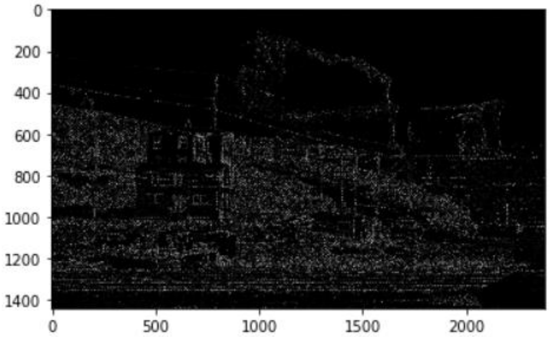
Notice that we already have the real value of some pixels' blue and red channels, we shouldn't change them using medium average. So I keep these original values, mask the output of medium average to extract the other values, and sum them up to construct the final solution.

Experiment shows that this approach achieves better average error and maximum error.

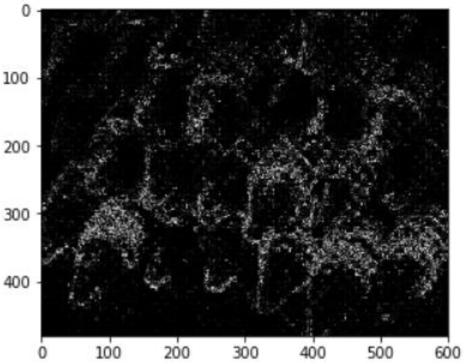




```
The average per-pixel error for tony is: 8.999601041666667
The maximum per-pixel error for tony is: 597
```



```
The average per-pixel error for iceberg is: 29.3947479359082
The maximum per-pixel error for iceberg is: 675
```



```
The average per-pixel error for crayons is: 32.299368055555554
The maximum per-pixel error for crayons is: 490
```