

Homework1 ISP with Rawpy

Zhen Tong 120090694

Task 1 Bad Pixel Correction

Using the `rawpy` lib, the `postprocess` function will convert the `RAW` file to the `rgb` format with Dead Pixel Correction.

```
with rawpy.imread(input_file) as raw:  
    rgb = raw.postprocess(**params)  
  
    bgr = cv2.cvtColor(rgb, cv2.COLOR_RGB2BGR)  
    cv2.imwrite(output_file, bgr)
```

Run

```
python main.py ../data/set01/DSC00049.ARW ./data/task1/output.jpg --task-no 1
```

Output



Task 2 Black Level Compensation

User can set the parameter `user_black` to change the black level. However, the customized value is hard to define.

```
with rawpy.imread(input_file) as raw:
    params = {}
    if args.user_black is not None:
        params['user_black'] = args.user_black

    rgb = raw.postprocess(**params)

    bgr = cv2.cvtColor(rgb, cv2.COLOR_RGB2BGR)
    cv2.imwrite(output_file, bgr)
```

Run

```
python main.py ../data/set01/DSC00049.ARW ./data/task2/output.jpg --task-no 2 --
user-black 100
```

Output



Task 3 Auto White Balance and Gain Control

Turn on the `use_auto_wb` parameter, then the `postprocess` will automatically adjust the white balance.

```
params = {}
if args.use_auto_wb:
    params['use_auto_wb'] = args.use_auto_wb
rgb = raw.postprocess(**params)

bgr = cv2.cvtColor(rgb, cv2.COLOR_RGB2BGR)
cv2.imwrite(output_file, bgr)
```

Run

```
python main.py ../data/set01/DSC00049.ARW ./data/task3/output.jpg --task-no 3 --use-auto-wb
```

Output

Compared to the original image, the AWB one color is closer to the human eye.



Task 4 Test no_auto_scale

If user turn off the auto pixel value scale, the image can be darker or brighter.

```
with rawpy.imread(input_file) as raw:
    if args.no_auto_scale:
        params['no_auto_scale'] = args.no_auto_scale
        print("Enable no_auto_scale")

    rgb = raw.postprocess(**params)

    bgr = cv2.cvtColor(rgb, cv2.COLOR_RGB2BGR)
    cv2.imwrite(output_file, bgr)
```

Run

```
python main.py ../data/set01/DSC00049.ARW ./data/task4/output.jpg --task-no 4 --no-auto-scale
```

Output

In this image, the image is green. Because in the [Sony ARW](#), the green channel value is more concentrate in the higher part. Therefore without auto scale, it is more green.



Task 5 Test no_auto_bright

Similarly, if user turn off the auto bright, the pipeline won't go through the brightness control. It can be darker or lighter.

```
python main.py ../data/set01/DSC00049.ARW ./data/task4/output.jpg --task-no 4 --  
no-auto-scale
```

Output

In this image, the image is darker



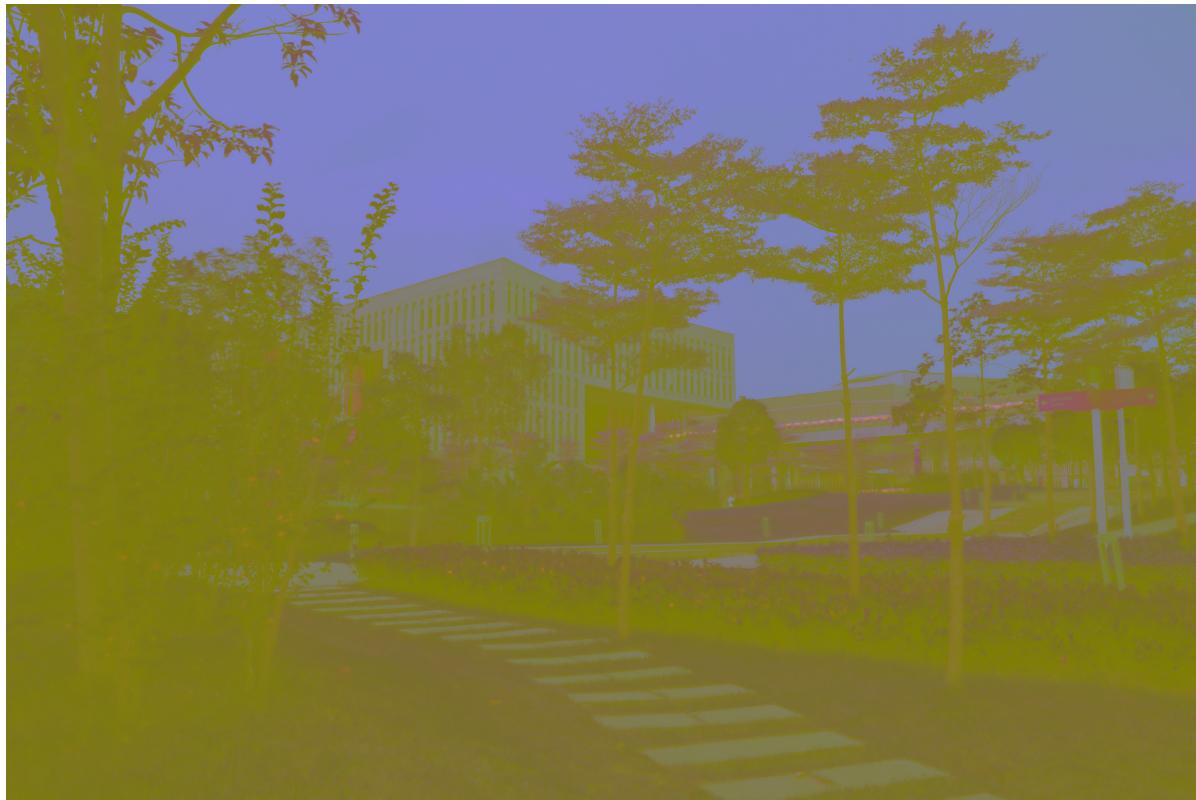
Task 6 Color Space Conversion

Color Space Conversion is either change from RGB to YUV, or from YUV to RGB.

```
python main.py ./data/set01/DSC00049.JPG ./data/task6/output1.jpg --task-no 6 --
RGB2YUV
```

Output

First we change the RGB to YUV, and plot the image in 3 channel. This is not a normal image because the color space is transformed. But latter we can convert it back to RGB. This image is just for an illustration



```
python main.py ./data/task6/output1.jpg ./data/task6/output2.jpg --task-no 6 --
YUV2RGB
```

Output

We can convert the above YUV image back to RGB with out damage.



Task 7 Test Gamma Correction

The Gamma Correction is a non-linear transform unlike above. Now we can let the darker part bright while the brighter part not increase its intensity too much with

$$\left(\frac{X}{255}\right)^\gamma \times 255$$

```
python main.py ../data/set01/DSC00049.ARW ./data/task7/output.jpg --task-no 7
```

Output

In the output image, we can see the grass is brighter, while the sky brightness changed little.



Run All

```
chmod +x run.sh  
./run.sh
```

