Assignment 02- Implement Image Processing Pipeline

This assignment is required to use of python, and the libraries you need to use are given in the *".py"* files.

Do not use any other libraries! You can use *numpy* and **opencv**(File reading only) in your filled code!

Try your best to implement a better visual-looking result with a faster speed and less memory consumption.

Your code should not run more than 30s.

Note: Do not copy any code from OpenISP/FastOpenISP, some guys did those before and finally get **0**.

Part 1 - Implement Bayer Domain Processor (40pts + 15pts extra)

Step 1. Dead Pixel Correction (10pts)

Implement

```
deadPixelCorrection()
```

Step 2.'Black Level Compensation' (5pts)\

Implement

```
blackLevelCompensation()
```

Step 3.'lens shading correction (Skipped)

Step 4. Anti Aliasing Filter (5pts)

Implement

```
antiAliasingFilter()
```

Step 5. Auto White Balance and Gain Control (10pts)

Implement

```
AWB()
```

Step 6. Chroma Noise Filtering (Extra 10pts)

Implement

```
ChromaNoiseFiltering()
```

Step 7. 'Color Filter Array Interpolation' Malvar (15pts)

Implement

```
CFA_Interpolation()
```

Part 2 - Implement YUV Domain Processor (45pts)

Implement

```
RGB2YUV()
UV2RGB()
```

Step 1 Edge Enhancement for Luma (15pts)

Implement

```
EdgeEnhancement()
```

Step 2 Brightness/Contrast Control (5pts)

Implement

```
BrightnessContrastControl()
```

Step 3 False Color Suppression (10pts)

Implement

```
FalseColorSuppression()
```

Step 4 Hue/Saturation control (10pts)

Implement

```
HueSaturationControl()
```

Step 5 Global Gamma Correction using BT.709 (5pts)

Implement

```
Gamma_BT709()
```

Part 3 - Going Above and Beyond (Optional 10pts)

Try your best to realize the best **visually satisfying results** you think. You can **add any steps you like** to enhance your final results. E.g., Local Tone Mapping, Better Auto White Balance Algorithm

Submission

To Grade submission for this homework. It would help if you tried committing and tagging your code as specified in the given code structure. We highly recommend that you finish the homework to prevent any issues when you start on the following projects for this class.

Your grades = correct code (70%) + runtime efficiency (10%)+good report (20%)

Code submission　(60%)

The code is required with a simple run, and then the TAs can see the results of each steps (name you saved results as *partxx-stepxx.jpg*)

Report （40%）

For each function you implemented, show the results crops , describe what you have done, and explain your results.

Compare your results with the Rawpy and try to improve your code performance.

Write your report using **Markdown** (like MarkText, a free markdown editor) or Latex and export it to PDF format. Markdown is easy to use, just learn to use.