Submission Guideline

Outline

- Homework file format
  - Programming language
  - README
  - Image I/O
- Submission requirement
  - Source Code
  - Report

Homework File Format

Programming Language

- Python
  - numpy
  - matplotlib
  - OpenCV
  - Pillow
- Matlab
  - Image processing toolbox
- C/C++
  - OpenCV
    - Installation guideline (`apt install libopencv-lib`)
    - Using OpenCV with gcc and CMake :

Only for Image I/O and plotting

README

- Named README
  - README.sh for python
  - README.m for matlab
  - README.sh for c/cpp
- The file should include the following information:
  - Homework number
  - Your name
  - Your student ID #
  - Your email address

Python Example

- README.sh

```
# DIP Homework Assignment #1
# Name: William Watt
# ID #: x12345678
# email: wwatt@csie.ntu.edu.tw
python hw1.py --input lena.jpg --output hw1_result.jpg
```

- Run your code
  - `sh README.sh`
- For Windows users
  - WSL
  - bash in powershell

MATLAB Example

- README.m

```
% DIP Homework Assignment #1
% Name: William Watt
% ID #: X12345678
% email: wwatt@csie.ntu.edu.tw
hw1(input_path, output_path)
```

- Run your code

```
>>README
```

C/Cpp Example

- README
  - Please refer to OpenCV webpage.

```
# DIP Homework Assignment #1
# Name: William Watt
# ID : X12345678
# email: wwatt@csie.ntu.edu.tw
cmake
make
./hw1 input_image output_path
```

- Remember to link other libs you used
- Run your code

```
sh README.sh
```

Image I/O (Python example)

- read raw

```
img = np.fromfile('sample.raw', dtype='uint8')
```

- write raw

```
img.tofile("result.raw")
```

- write jpg

```
cv2.imwrite("result.jpg", img)
```

- read jpg
  - grayscale image

```
img = cv2.imread("sample.jpg", cv2.IMREAD_GRAYSCALE) # 1 channel
img = cv2.imread("sample.jpg") # 3 channel
```

  - color image

```
img = cv2.imread("sample.jpg") # 3 channel
```

Image I/O (MATLAB raw example)

- Read raw file

```
fid=fopen(image_name,'rb');
pixel=fread(fid,inf, 'uchar');
fclose(fid);
```

- Write raw file

```
fid = fopen('test.raw', 'wb');
fwrite(fid, pixel, 'uchar');
fclose(fid);
```

Image I/O (MATLAB jpg example)

- Grayscale image

```
function hw1(image_name, output_name)
    img = imread(image_name); % # of channel: 3
    img = rgb2gray(img); % # of channel: 1

    % do your algorithm here

    imwrite(img, output_name);
end
```

- Color image

```
function hw1(image_name, output_name)
    img = imread(image_name); % # of channel: 3

    % do your algorithm here

    imwrite(img, output_name);
end
```

Image I/O (C/Cpp raw example)

- Grayscale image

```c
int main(){
    FILE *file;
    unsigned char image_data[SIZE][SIZE];
    file = fopen("sample.raw", "rb");
    fread(image_data, sizeof(unsigned char), SIZE*SIZE, file);
    fclose(file);

    // do some image processing task...

    file = fopen("result.raw", "wb");
    fwrite(image_data, sizeof(unsigned char), SIZE*SIZE, file);
    fclose(file);
    return 0;
}
```

- Color image

```c
int main(){
    FILE *file;
    unsigned char image_data[3][SIZE][SIZE];
    file = fopen("sample.raw", "rb");
    fread(image_data, sizeof(unsigned char), SIZE* SIZE * 3, file);
    fclose(file);

    // do some image processing task...

    file = fopen("result.raw", "wb");
    fwrite(image_data, sizeof(unsigned char), SIZE* SIZE * 3, file);
    fclose(file);
    return 0;
}
```

Image I/O (C/Cpp jpg example)

- Grayscale image

```cpp
#include <stdio.h>
#include <opencv2/opencv.hpp>
using namespace cv;
int main(){
    Mat img;
    img = imread("sample.jpg", IMREAD_GRAYSCALE); //1 channel
    img = imread("sample.jpg"); //3 channel

    //do some image processing task...

    imwrite("result.jpg", img);
    return 0;
}
```

- Color image

```cpp
#include <stdio.h>
#include <opencv2/opencv.hpp>
using namespace cv;
int main(){
    Mat img;
    img = imread("sample.jpg"); //3 channel

    //do some image processing task...

    imwrite("result.jpg", img);
    return 0;
}
```

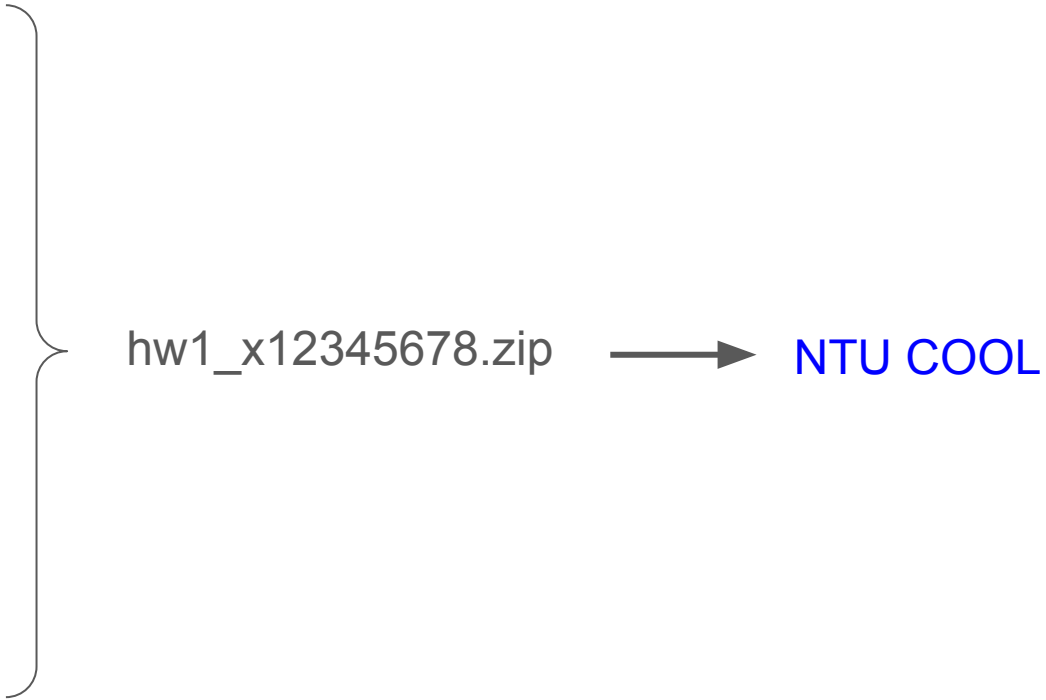
Submission Requirement

1. Source Code

+- hw1_x12345678/

    +- README.sh

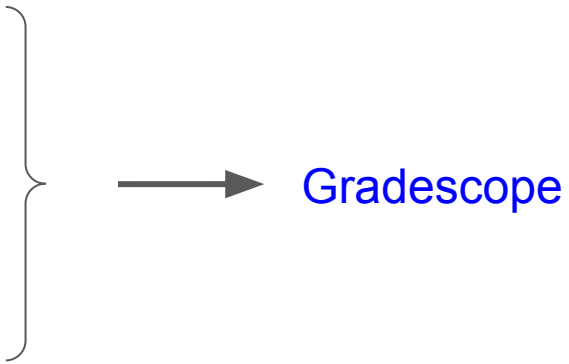
    +- mycode.cpp



2. Report

Every problems should contain:

1. Your motivation and approach (include parameters)
2. Original images
3. Output images
4. Discussion of results



Gradescope

https://www.gradescope.com/

1. Click Sign Up and choose sign up as a student

2. Enter Course Entry Code (ER6V2V)

3. Enter your name, school email, and student ID.

4. Then the system will send an email to your address to set up your password.

Grading Policy

- Program 30%
  - Format 10/30
  - Execution
  - Algorithm 20/30
- Report 70%
  - motivation and approach 20/70
  - performance of results 20/70
  - discussion 30/70

Remark

- Unix-Based environment is recommended
- If you use Windows system
  - Windows Subsystem for Linux (WSL)
  - bash in powershell
    - Anaconda
  - CSIE workstation
    - You may need X-server to show Image
- Compress the folder by ZIP only
- If you have any question, feel free to post on NTUCOOL
- TA hour
  - Tue. 14:00~16:00
  - Fri. 10:00~12:00

Homework 1

Problem 1: Warm-up

1. Color image -> Gray image
   a. Can’t use functions to convert to gray image directly
   b. You can use those functions for checking
2. Basic manipulations
   a. horizontal flipping

Problem 2: Image Enhancement

1. Brightness adjustment
2. Histogram Modification
   a. Global histogram equalization
   b. Local histogram equalization
3. Contrast Manipulation
   a. Log transform
   b. Inverse-log transform
   c. Power-law transform

Problem 3: Noise Removal

- Noise removal
  - Design a method to remove Gaussian noise
  - Design the other method to remove salt-and-papper noise
  - Compute PSNR of these two results