

Submission Guideline

Outline

- Homework file format
 - Programming language
 - RUN
 - 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**

RUN

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

Python Example

- RUN.sh

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

- Run your code

- sh RUN.sh

MATLAB Example

- RUN.m

```
% DIP Homework Assignment #1  
% Name: William Watt  
% ID #: x12345678  
% email: wwatt@csie.ntu.edu.tw  
hw1(input_path, output_path)
```

- Run your code

```
>>RUN
```

C/Cpp Example

- RUN
 - 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 RUN.sh
```


Image I/O (Python example)

- read raw

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

- write raw

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

- write png

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

- read png

- grayscale image

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

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

- color image

```
img = cv2.imread("sample.png") #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 hwl(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 hwl(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

```
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

```
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

```
#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

```
#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/**

+ - RUN.sh

+ - mycode.py

+ - SampleImage/

+ - sample1.png

+ - sample2.png

.....

一定要有最外層 folder

hw1_x12345678.zip

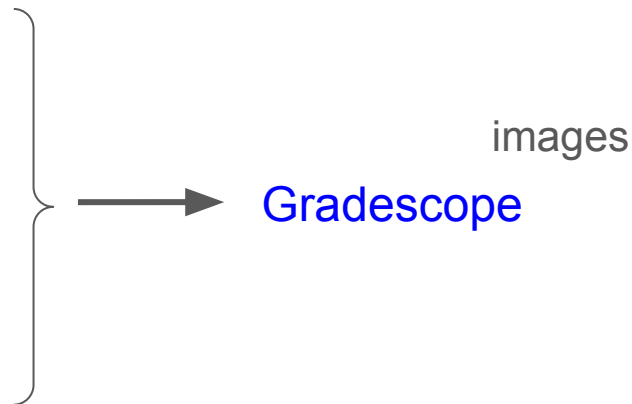


NTU COOL

2. Report

Every problems should contain:

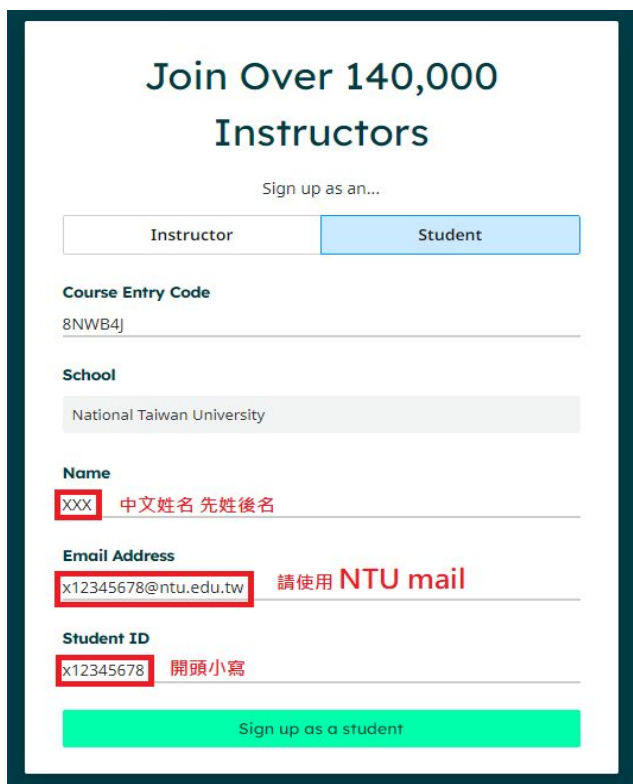
1. Original images
2. Output
3. Your **motivation and approach** (include parameters)
4. **Discussion** of results



Gradescope

<https://www.gradescope.com/>

1. Click Sign Up and choose sign up as a student
2. Enter Course Entry Code (**8NWB4J**)
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.



Join Over 140,000
Instructors

Sign up as an...

Course Entry Code
8NWB4J

School
National Taiwan University

Name
XXX 中文姓名 先姓後名

Email Address
x12345678@ntu.edu.tw 請使用 NTU mail

Student ID
x12345678 開頭小寫

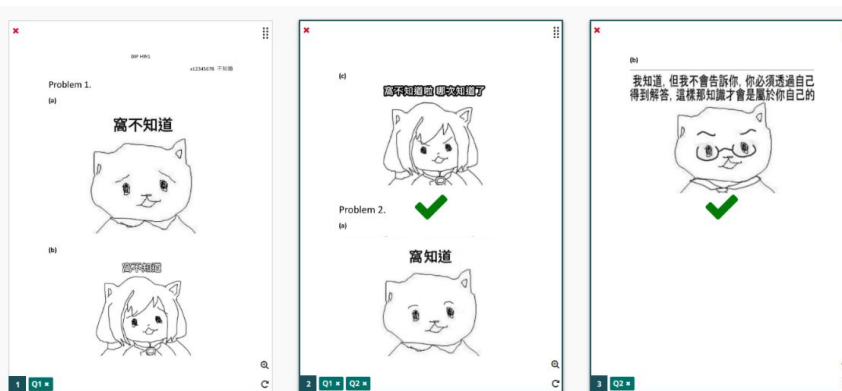
Gradescope

Select the correct pages

Question Outline

Select pages to assign to Question 2.

Title	Points
1 Problem 1	50.0 pts
P1 • P2 •	
2 Problem 2	50.0 pts
P2 • P3 •	



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

The regrade period is **3 days** after each homework's grade release

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 NTU COOL
- TA hour @532
 - Tue. 12:00~14:00
 - Thu. 13:00~15:00