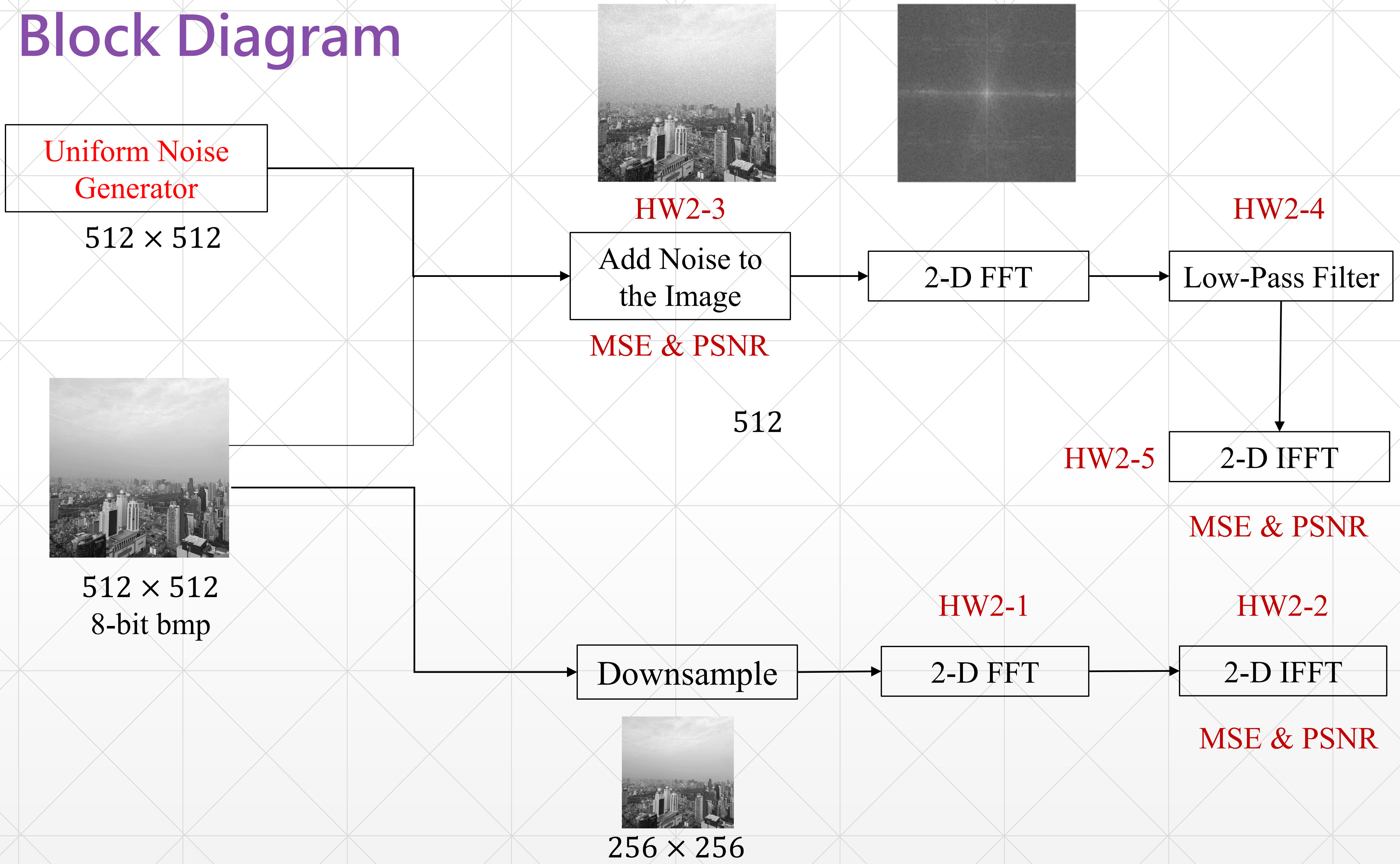
Digital Image Processing
(CO6041)
HOMEWORK#2

TA：Yi-Hao Lin（林邑豪)
Adviser：Chih-Wei Tang（唐之瑋)

Visual Communications Lab
Department of Communication Engineering
National Central University

Nov 22, 2021

Block Diagram



Radix-2 Fast Fourier Transform(1/4)

- The time complexity of the discrete Fourier transform is $\boldsymbol{O}(\boldsymbol{N}^2)$,which will cause computational burden when the sampling rate is large.
- Fast Fourier transform (FFT) algorithm can be used to decompose the calculation amount, then merge to reduce the time complexity to $\boldsymbol{O}(\boldsymbol{N}\log\boldsymbol{N})$.
- We split the N-point data sequence into two N/2-point data sequences $f_1(n)$ and $f_2(n)$, corresponding to the **even-numbered** and **odd-numbered** samples of $x(n)$, respectively, that is :

$$f_1(n) = x(2n)$$
$$f_2(n) = x(2n+1)\ , n = 0,1,\ldots,\frac{N}{2}-1$$

Fast Fourier Transform (FFT) - https://www.cmlab.csie.ntu.edu.tw/cml/dsp/training/coding/transform/fft.html
數位信號處理器_林顯易_第十一單元 快速傅立葉轉換_Part3 快速傅立葉演算法(上)- https://www.youtube.com/watch?v=IotFZRCSVP8

Radix-2 Fast Fourier Transform(2/4)

- To compute a discrete Fourier transform:

$$X(k) = \sum_{x=0}^{N-1} x(n)W_N^{kn}, W_N = e^{-j2\pi/N}\ for\ k = 0,1,2,\ldots,N-1$$

$$= \sum_{even} x(n)W_N^{kn} + \sum_{odd} x(n)W_N^{kn}$$

$$= \sum_{m=0}^{\frac{N}{2}-1} \boxed{x(2m)}\boxed{W_N^{2mk}} + \sum_{m=0}^{\frac{N}{2}-1} \boxed{x(2m+1)}\boxed{W_N^{(2m+1)k}}$$

$$= \sum_{m=0}^{\frac{N}{2}-1} \boxed{f_1(m)}\boxed{W_{\frac{N}{2}}^{mk}} + \sum_{m=0}^{\frac{N}{2}-1} \boxed{f_2(m)}\boxed{W_N^{2mk}W_N^{k}}$$

$$= F_1(k) + F_2(k)W_N^k$$

- where $F_1(k)$ and $F_2(k)$ are the N/2-point DFTs of the sequences $f_1(m)$ and $f_2(m)$, respectively.

Fast Fourier Transform (FFT) - https://www.cmlab.csie.ntu.edu.tw/cml/dsp/training/coding/transform/fft.html
數位信號處理器_林顯易_第十一單元 快速傅立葉轉換_Part3 快速傅立葉演算法(上)- https://www.youtube.com/watch?v=IotFZRCSVP8

Radix-2 Fast Fourier Transform(3/4)

- Since $F_1(k)$ and $F_2(k)$ are periodic, with period $N/2$, so

$$X\left(k+\frac{N}{2}\right)=F_1\left(k+\frac{N}{2}\right)+F_2\left(k+\frac{N}{2}\right)W_{\mathrm{N}}^{k+\frac{N}{2}}$$

$$=F_1(k)-W_{\mathrm{N}}^{k}F_2(k)$$

$$\because W_{\mathrm{N}}^{k+\frac{N}{2}}=e^{-j\frac{2\pi}{N}k}e^{-j\frac{2\pi}{N}\frac{N}{2}}=W_{\mathrm{N}}^{k}*(-1)$$

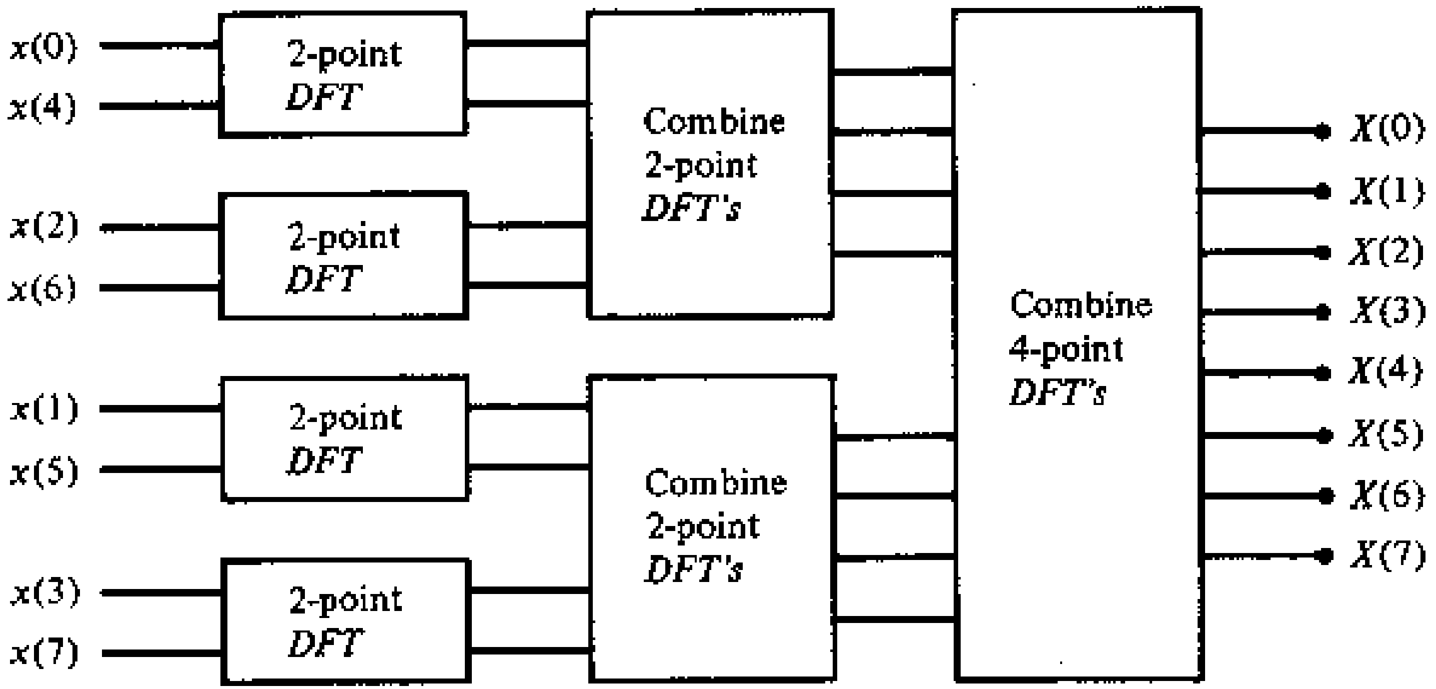
- Hence the equation may be expressed as

$$X(k)=F_1(k)+W_{\mathrm{N}}^{k}F_2(k)\,,k=0,1,\ldots,\frac{N}{2}-1$$

$$X\left(k+\frac{N}{2}\right)=F_1(k)-W_{\mathrm{N}}^{k}F_2(k),k=0,1,\ldots,\frac{N}{2}-1$$

Fast Fourier Transform (FFT) - https://www.cmlab.csie.ntu.edu.tw/cml/dsp/training/coding/transform/fft.html
數位信號處理器_林顯易_第十一單元 快速傅立葉轉換_Part3 快速傅立葉演算法(上)- https://www.youtube.com/watch?v=IotFZRCSVP8

Fast Fourier Transform(4/4)



Three stages in the computation of an N = 8-point DFT.

Figure TC.3.2-Fast Fourier Transform (FFT) - https://www.cmlab.csie.ntu.edu.tw/cml/dsp/training/coding/transform/fft.html

Generating Uniform Noise

- The general formula for the probability density function of the uniform distribution is:

$$f_X(x) = \begin{cases} \dfrac{1}{b-a}, for\ a \le x\ \le b \\ 0, otherwise \end{cases}$$

- We set $a\ =\ -20\ \&\ b = 20$
- After adding noise, **set grayscale values over 255 to 255, and to 0 for those are less than 0**.

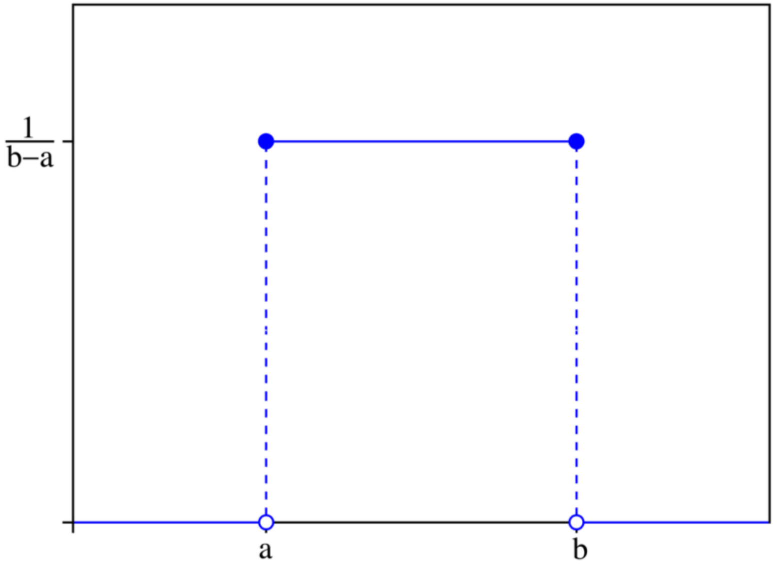
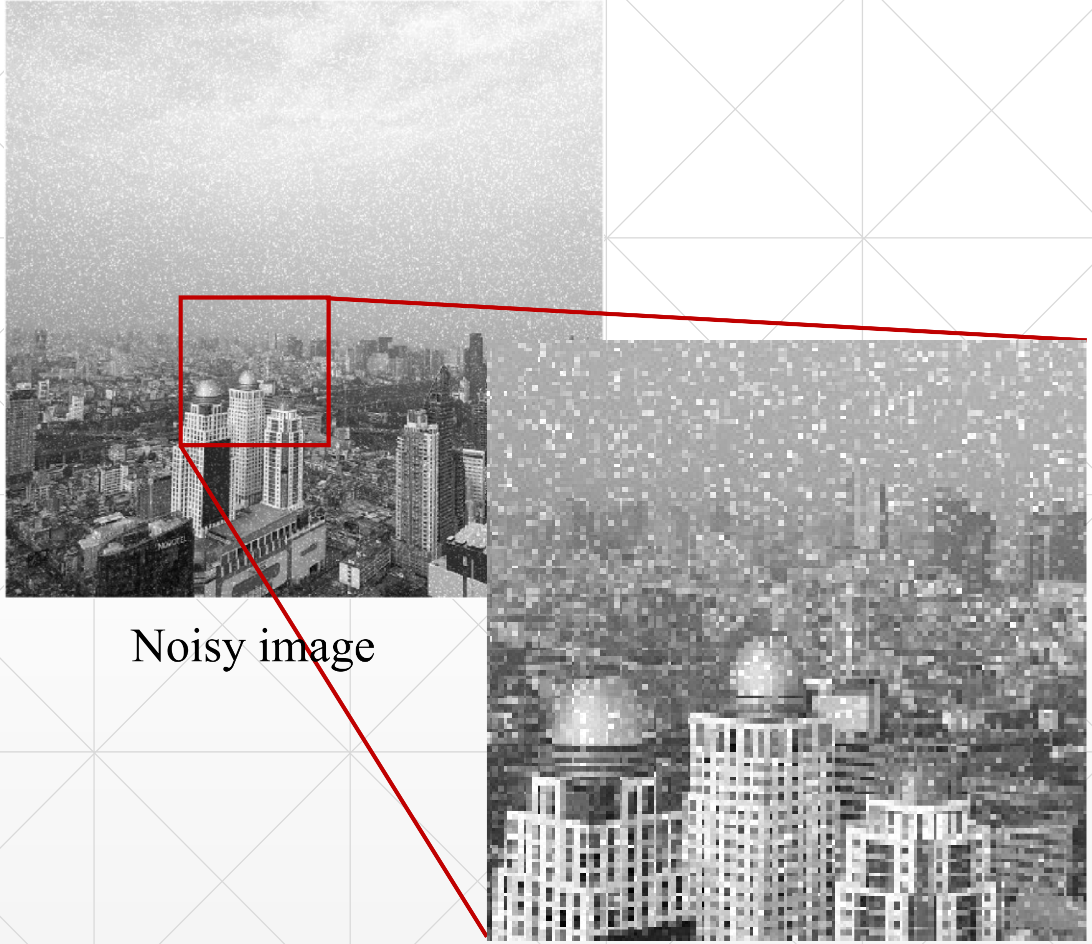


Figure: 連續型均勻分布 https://zh.wikipedia.org/wiki/%E9%80%A3%E7%BA%8C%E5%9E%8B%E5%9D%87%E5%8B%BB%E5%88%86%E5%B8%83

Adding Noise to Image

Original image



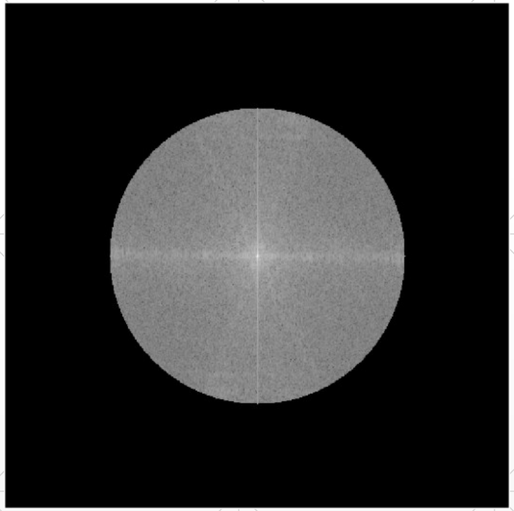
Noisy image

Low-Pass Filter

- A low-pass filter is a filter that passes signals with a frequency lower than a selected cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency.
- **Objective:** Achieve higher PSNR of the denoised image than the noisy image.

$$H(u,v)=\begin{cases}1 & ,if\ D(u,v)\le D_0\\ 0 & ,if\ D(u,v)>D_0\end{cases}$$

$$D(u,v)=\left[\left(u-\frac{M}{2}\right)^2+\left(v-\frac{N}{2}\right)^2\right]^{\frac{1}{2}}$$



Low-Pass Filter
with $D_0 = 150$ in the frequency domain

$D(u,v)$: image in the frequency domain.
$H(u,v)$: filter mask in the frequency domain.
(u,v): index of frequency domain.

Low-Pass Filter

Original image

Noisy image

Denoised image with $D_0 = 150$
(after IFFT)

Image Quality Metrics

- MSE (Mean Squared Error):

$$MSE = \frac{1}{Image\ Size} \sum_{i=1}^{Image\ Size} \left(Y_i - \hat{Y}_i\right)^2$$

- PSNR (Peak Signal-to-Noise Ratio):

$$PSNR = 10 \times \log\left(\frac{255^2}{MSE}\right)$$

$where$

Y_i: The $i\text{-}th$ pixel value of the original image

$\hat{Y}_i$: The $i\text{-}th$ pixel value of the image processed by IDCT

$Image\ Size$: Image length × Image width

Grading

- **Code & Demo (70%): Use the C/C++ only. Matlab or OpenCV is not allowed**
  - 2-D FFT (20%) (HW2-1)
  - 2-D IFFT & MSE, PSNR measuring (10% + 10%) (HW2-2)
  - Generating Uniform-noisy image & MSE, PSNR measuring (5% + 5%) (HW2-3)
  - Mask filtering (10%) (HW2-4)
  - 2-D IFFT & MSE, PSNR measuring (5% + 5%) (HW2-5)
- **Report (30%):**
  - Flowchart (10%)
  - Experiment results (10%)
  - Discussion and Analysis (10%)

Due Date & Demo Schedule

- **Demo Date**: Dec. 13 (Monday) or Dec. 14 (Tuesday)
- **Demo Time & Location**: 13:30 ~ 17:30 @TBD
- The demo schedule will be announced at the TA webpage.
- You should compress your entire project (including .c/.cpp, .exe file, etc.) and report (.pdf) as a .zip file and submit to New ee-class before Dec. 13, 13:00.
- No delay. (**If you have any special case, please inform us by sending an email early**.)

Note

- Do it yourself!
- You will get a zero when you delay or fail to operation in demo (code and demo part), but you can still get points in report part.
- Everyone will be asked a few questions and operations when you are in demo. (Do not call for help.)
- The TA will use another image to test your code.
- If you have a notebook, please bring your own notebook. Otherwise, some people may not be able to execute the code during the demo.
- Remote connection/control is not allowed.

The details will be announced on our course website:
https://sites.google.com/view/ncuvclab/home/course/fall-2021-ta-dip

REFERENCES

- Gonzalez, Rafael C., and Richard E. Woods, "Digital image processing," Prentice Hall, 2007.
- Test image "DIPpic1.bmp" download:

  https://drive.google.com/file/d/1zI86_Dat0xKyGKOTiknqeTlD5tVgdMGu/view?usp=sharing