Available: rgb2gray, imshow, fftshift, np.matmul, imread

Eye detection -> fft?

Signals and Systems

Prof. Hae-Gon Jeon

Programing Assignment

(1) 2D Discrete Fourier Transform

(2) Azimuthal Averaging

(3) High Pass Filtering

(4) 2D Discrete Inverse Fourier Transform

(5) Report

Specific Objectives:

- Detect the Fake images in frequency domain
- Understanding one of concepts of computer vision techniques, based on Signals and Systems knowledge
- Practice your programing skill

Evaluation

Mid term exam (25%)

Final term exam (35%)

Homework (10%) : Upload answers with your derivations for problems in the textbook in GEL site.

Programing project (20%) : Any programing language is available. Upload your report and source codes in GEL site.

Class participation (10%)

Image Generation

: Real Image

: Generated Image

- **Super Resolution**

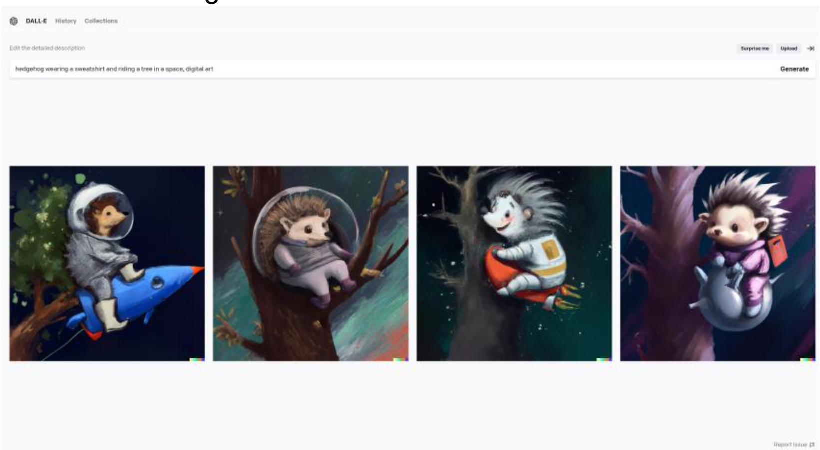
- **Interpolation: Generate interpolated images between two real images**

Image from: Axel Sauer et al, StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets, SIGGRAPH 2022

Application of Image Generation

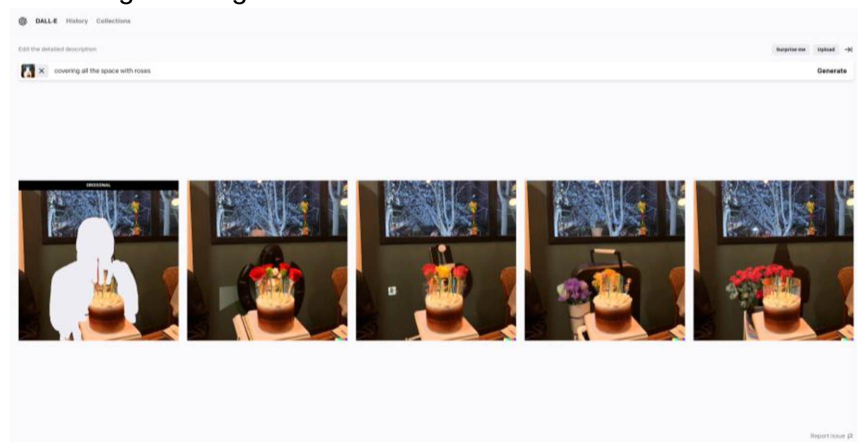
- **Text to Image and Image Editing**

➢ Text to Image



Generate images followed by a command

➢ Image Editing



Remove the certain part and composite the removed part with the harmonized image

- **Image Synthesis**

➢ Fashion Show

Synthesize a face to the different models

➢ Media Contents

Synthesize a virtual face to real human

Captured Image from: https://labs.openai.com/, https://www.youtube.com/watch?v=tW3l2jb0wBc
Image from: https://www.balenciaga.com

Fake Image Detection

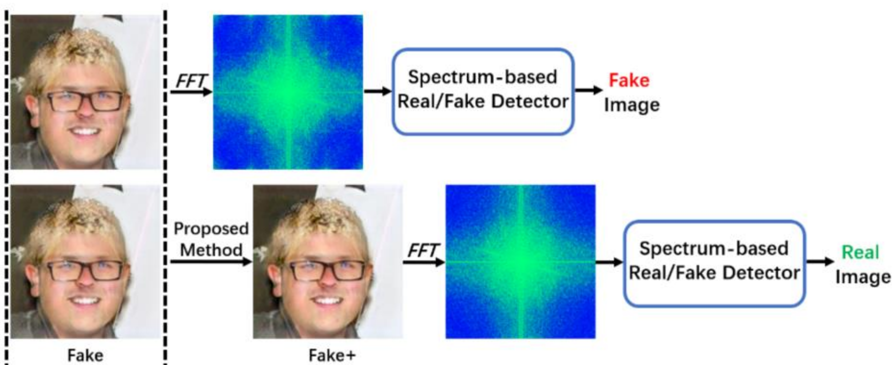
- **DeepFake: Deep Learning + Fake**

Commercial with an unauthorized personal image

Fake news

- **Fake Image Detection**

➢ Detect the fake image



Detect the fake image in frequency domain with Fourier Transform

➢ Detect the spliced parts from a forgery

Photoshopped image

The white part is the Spliced

Captured Image from: https://newsone.com/playlist/trump-arrest-fake-photos/item/1, https://www.youtube.com/watch?v=UuszlOBKkrM
Reference Pape: Chengdong Dong et al., Think Twice Before Detecting GAN-generated Fake Images from their Spectral Domain Imprints, CVPR 2022
Haiwei Wu et al., Robust Image Forgery Detection over Online Social Network Shared Images, CVPR 2022

Fake Image Detection

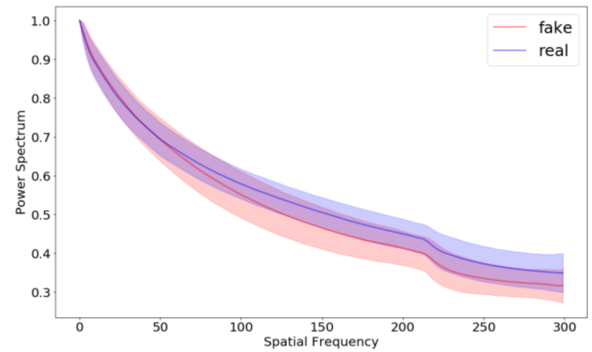
Detect the FAKE images!

4 images can be decided by 1D power Spectrum. The others can be decided by filtering and inverse Fourier transform.

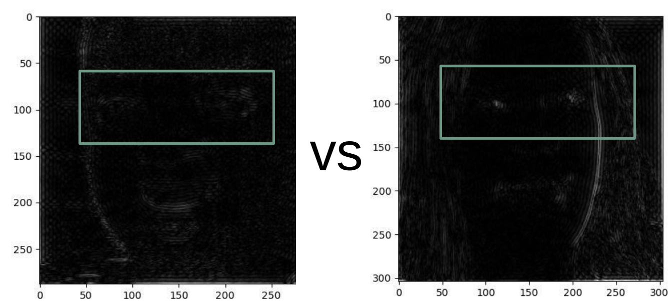
Dataset

OR

1D Power Spectrum



Filtering and IFT



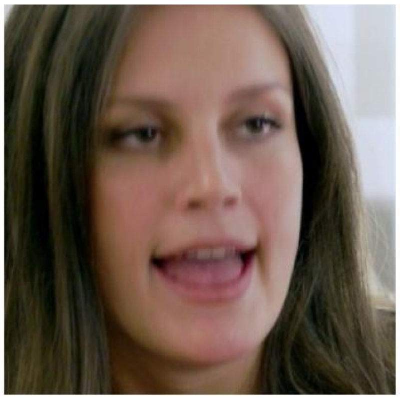
Choose the numbers of the fake images

Image from: Ricard et al., Watch your Up-Convolution: CNN Based Generative Deep Neural Networks are Failing to Reproduce Spectral Distributions, CVPR2020

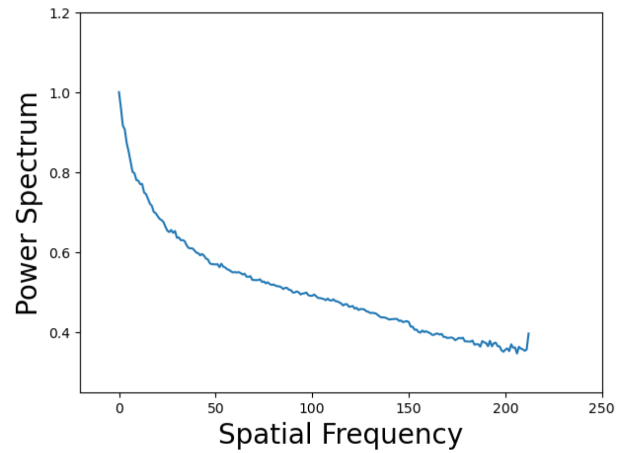
Task Pipeline

First, decide whether the image is fake or not using 1D power spectrum and then, decide whether fake or not using filtering and inverse Fourier transform.

0. input: image(RGB color)

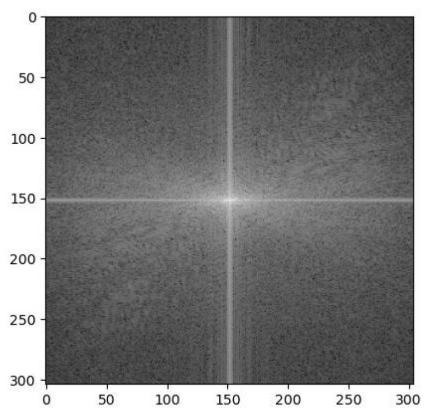


2. Azimuthal averaging and 1D power Spectrum

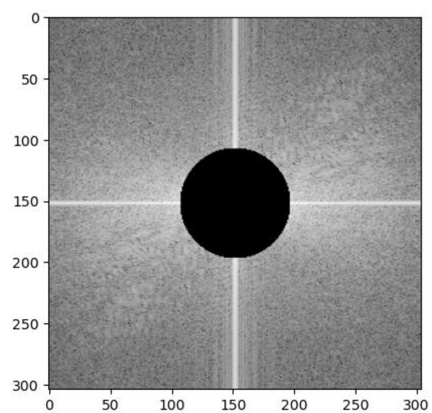


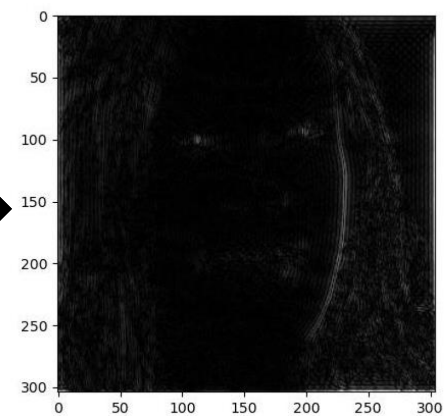
OR

1. 2D Discrete Fourier Transform



3. Filtering and 2D Inverse Discrete Fourier Transform(Gray scale)





Task1 - 2D Fourier Transform and 2D [4pts]

Implementation details

for문 두번 대신 exp 계산후 matrix -> matrix multiplication

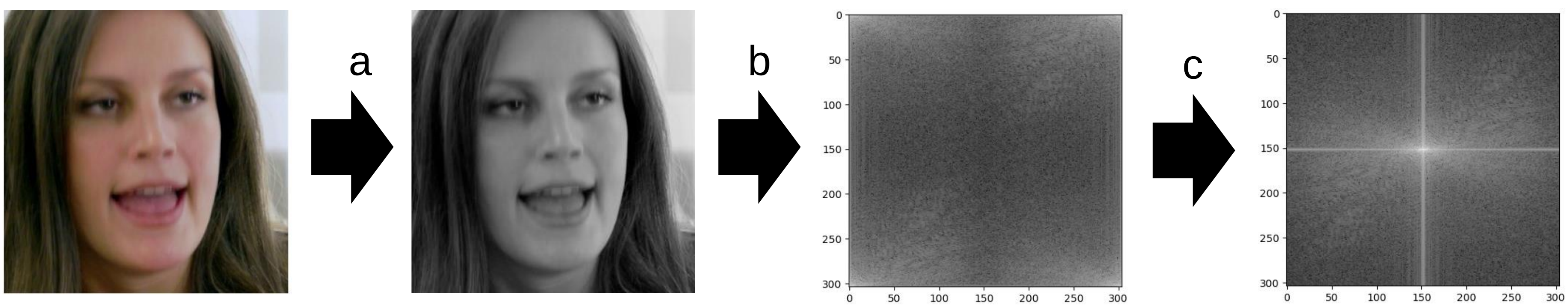
$$F(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \exp\left[-2\pi i \left(\frac{mu}{M} + \frac{nv}{N}\right)\right],$$

a * I * b = F(I)

$$u = 0, 1, \ldots, M-1, \qquad v = 0, 1, \ldots, N-1,$$

rgb2gray

1. *Implement 2D Fourier Transform Function*



a. *Load image with gray scale(You don't need to show)*
b. *Implement 2D Fourier transform with the same size of an image that you want to transform*
c. *Centerized the high frequency* Fftshift

Task2 – Azimuthal averaging [5pts]

0. *What is Azimuthal averaging?*
   a. *Azimuthal integration over radial frequencies ϕ*

$$AI(\omega_k) = \int_0^{2\pi} \left\|\mathcal{F}(I)(\omega_k \cdot \cos(\phi), \omega_k \cdot \sin(\phi))\right\|^2 d\phi \ \ for\ k \in \mathbb{Z}^+$$

$$where\ \mathcal{F}(I)(u,v)\ is\ the\ frequency\ domain\ of\ image\ I(x,y)$$

$$and\ w_k\ is\ the\ radius\ from\ the\ center\ of\ the\ frequency\ domain$$

   b. *Azimuthal averaging can make 2D power spectrum to the 1D power spectrum*

$$f(w_k) = \frac{AI(\omega_{k+1}) - AI(\omega_k)}{\pi(\omega_{k+1}^2 - \omega_k^2)}$$

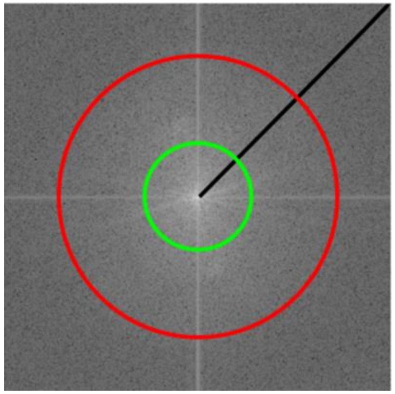
   c. *The thing what you do here is*

$$f(w_k) = \frac{(The\ sum\ of\ the\ frequencies\ between\ 2\ sequential\ radius)}{(The\ number\ of\ pixels\ between\ 2\ sequential\ radius)}$$

$$for\ k \in \{1, 2, \ldots, \lfloor \max(distance) \rfloor - 2\}$$

Recommend using round down the radius

※ Do not use the summation between radius 0 and radius 1.



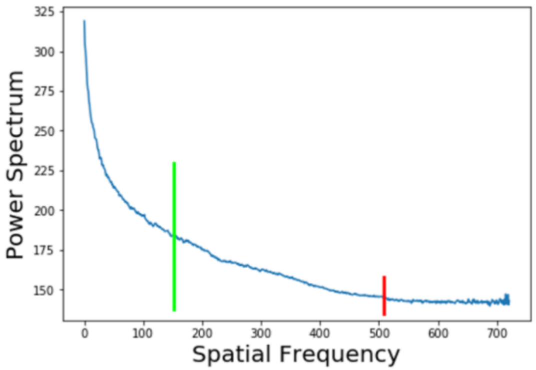
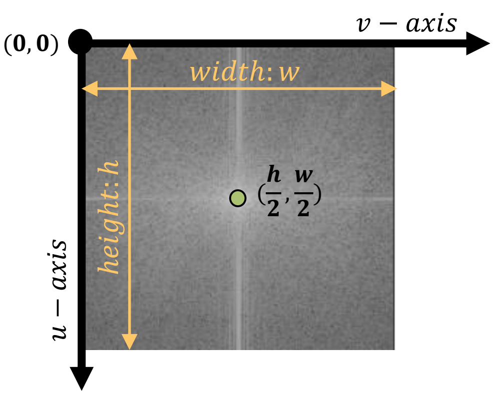


Image from: Ricard et al., Watch your Up-Convolution: CNN Based Generative Deep Neural Networks are Failing to Reproduce Spectral Distributions, CVPR2020

Task2 – Azimuthal averaging [5pts]

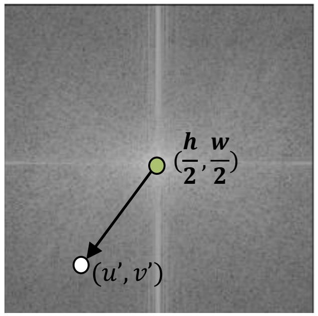
0. Overview

※ 'a'~'e' parts are the recommended programing not duty.



a. Calculate the center of the coordinate

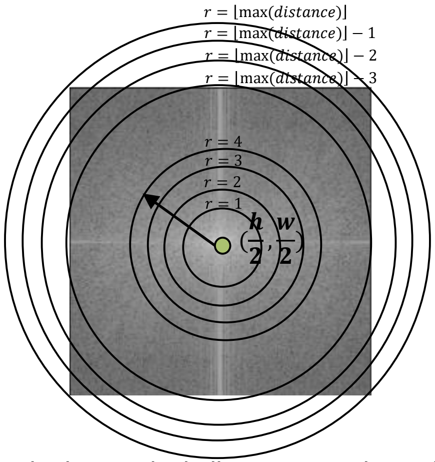
$$center: (\frac{h}{2}, \frac{w}{2})$$



b. Compute the distance(in the pixel coordinate) from the center

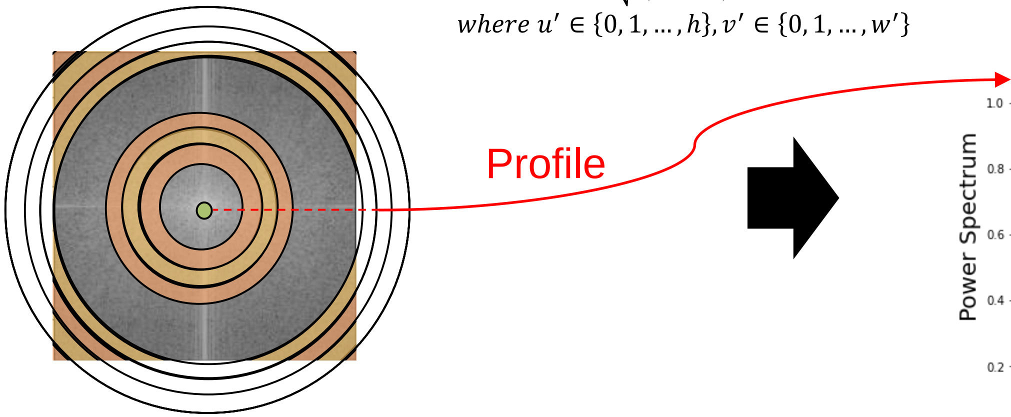
$$distance = \sqrt{\left(\frac{h}{2} - u'\right)^2 + \left(\frac{w}{2} - v'\right)^2}$$

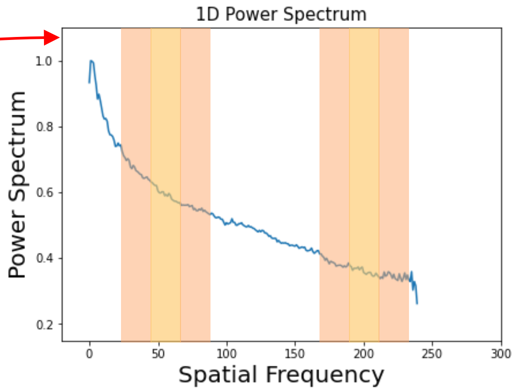
$$where\ u' \in \{0, 1, \ldots, h\}, v' \in \{0, 1, \ldots, w'\}$$



c. Take integer(>0) distances as the radius

$$r \in \{1, 2, \ldots \lfloor \max(distance) \rfloor\}$$





d. Add frequency power values(= the pixel values) between 2 circles(that have sequential radius): a donut in the figure

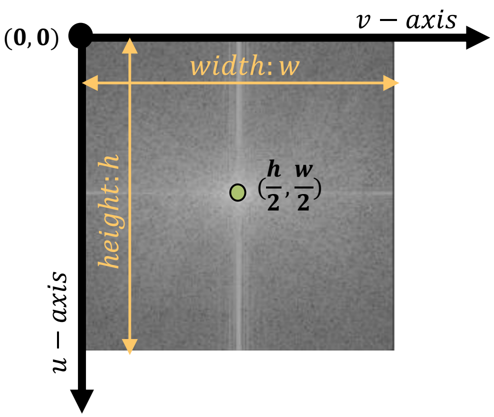
e. Average the frequency power values: Divide the value calculated from 'step1.d.' with the number of pixels in that area

f. Make 1D power spectrum

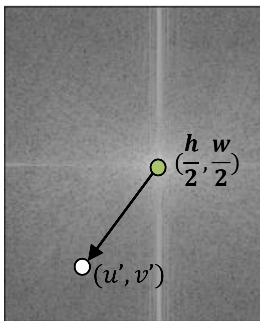
Task2 – Azimuthal averaging [5pts]

1. *Make azimuthal averaging*

※ *'a'~'e' parts are the recommended programing not duty.*



a. $center: (\frac{h}{2}, \frac{w}{2})$



b. $distance = \sqrt{\left(\frac{h}{2} - u'\right)^2 + \left(\frac{w}{2} - v'\right)^2}$

$where\ u' \in \{0, 1, \ldots, h\}, v' \in \{0, 1, \ldots, w'\}$

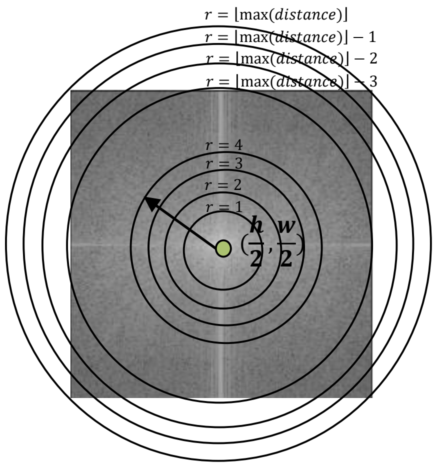
continue

a. *Calculate <u>the center</u> of the coordinate*
   a) *center of u-axis: half of the height*
   b) *center of v-axis: half of the width*

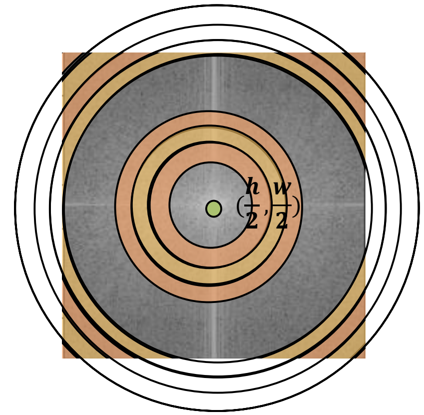
b. *Compute <u>the distance</u>(in the pixel coordinate) from the center to the other pixels*
   a) *save the distances using the list(or numpy(np) array)*
      e.g. distance = np.array([156.3, 155.1, …, 156.3])

Task2 – Azimuthal averaging [5pts]

1. Make azimuthal averaging



c. $r \in \{1, 2, \ldots \lfloor \max(distance) \rfloor\}$



d, e. *We need addition of frequency power values(= the pixel values) between 2 circles(that have sequential radius); a colored donut shape in the figure*

continue

c. Take integer(>0) distances as the <u>radius:</u>

- *a) Save sorted indicis of distances in the list(or numpy array)*
  - *e.g. distance_sorted_index = [200, 201, 203, …, 0]*
- *b) Type casting float or long type to integer*
  - *e.g. dinstance_int = [0, 0, 1, 1, 2, …, 156, 156, 156]*

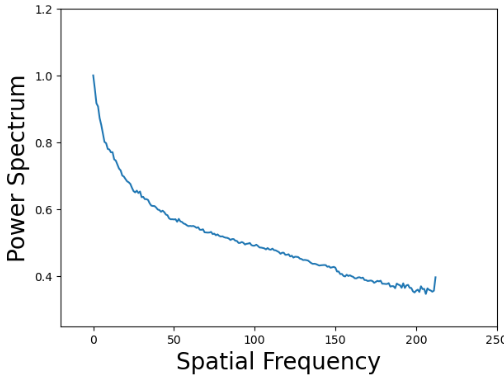
d. Make <u>cumulative summation of frequency power values</u> from integer type of distance 0 to integer type of max distance:

- *a) Make list of frequency power values same in the order of the sorted distance*
  - *e.g. freq_sorted = np.array([0, 1, 1, 2, 2, …, 249, 250])*
- *b) Add the frequency power values in cumulative way*
  - *e.g. cum_sum_freq = np.array([0, 1, 2, 4, 6, …, 100082, 100332])*

e. Average the frequency power values:

- *a) Take the cumulative summation values having same integer type of radius*
  - *e.g. cum_sum_same_int_rad = np.array([2, 6, …, 10082])*
- *b) Get the summation of frequency power values in the same integer type of radius*
  - *e.g. summation_b2w_2radius = cum_sum_same_int_rad[1:] - cum_sum_same_int_rad[:-1]*
- *c) Divide the each value with the number of pixels*

Task2 – Azimuthal averaging [5pts]



f

f. Make 1D power spectrum (Duty)

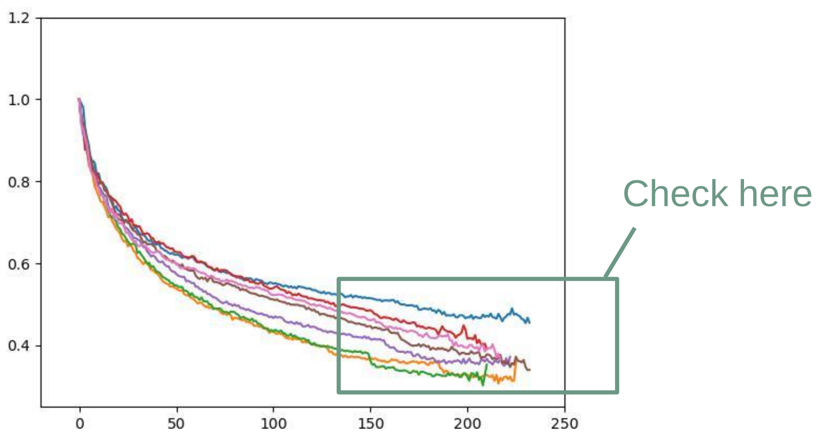
a) X-axis: spatial frequency

(a) Set the value from -20 to 250

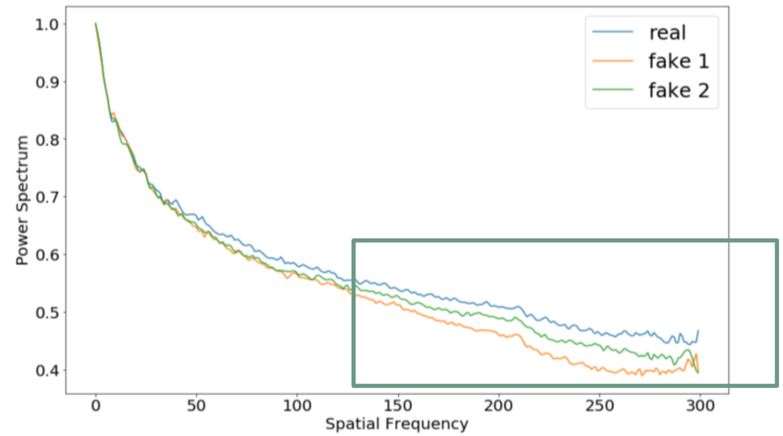
*b) Y-axis: average(that you get in the '*step e.c)*'). Divide all the average value with the max average value.*

(a) Set the value from 0.25 to 1.2

c) The figure at the bottom is the final result in this task. The two above lines are real and the two below are fake in the figure.



You can detect the fake images using 1D Power Spectrum



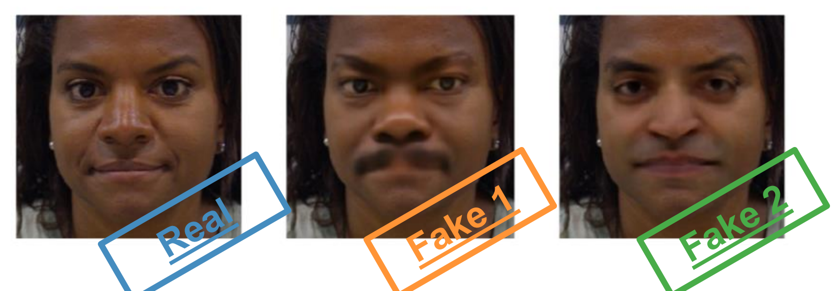
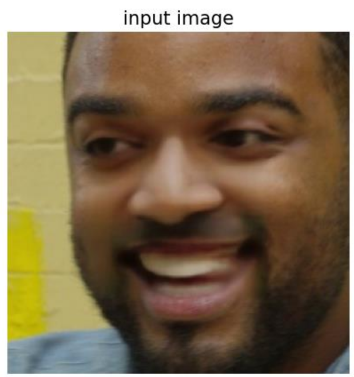


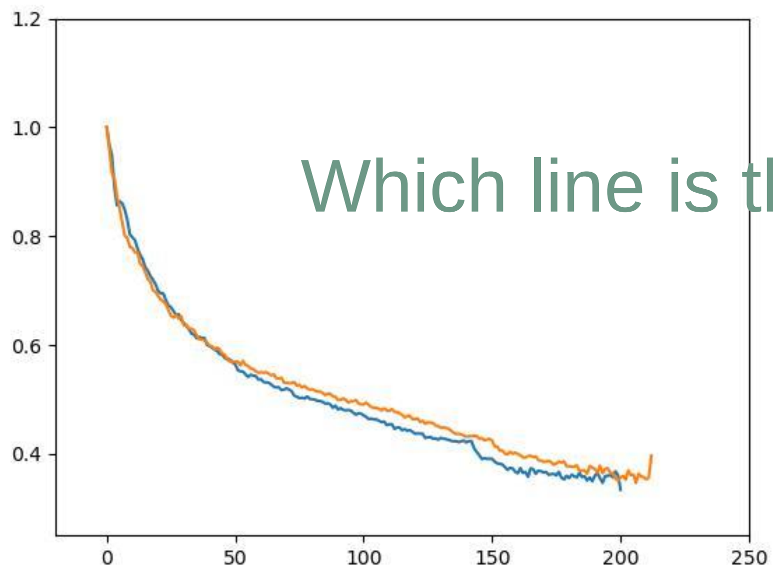
Image from: Ricard et al., Watch your Up-Convolution: CNN Based Generative Deep Neural Networks are Failing to Reproduce Spectral Distributions, CVPR2020

Task3 – High-pass filtering and 2D Inverse Fourier Transform [6pts]

Some images can't be decided whether fake or not, then use high-pass filtering.



vs



Which line is the fake image?

Task3 – High-pass filtering and 2D Inverse Fourier Transform [6pts]

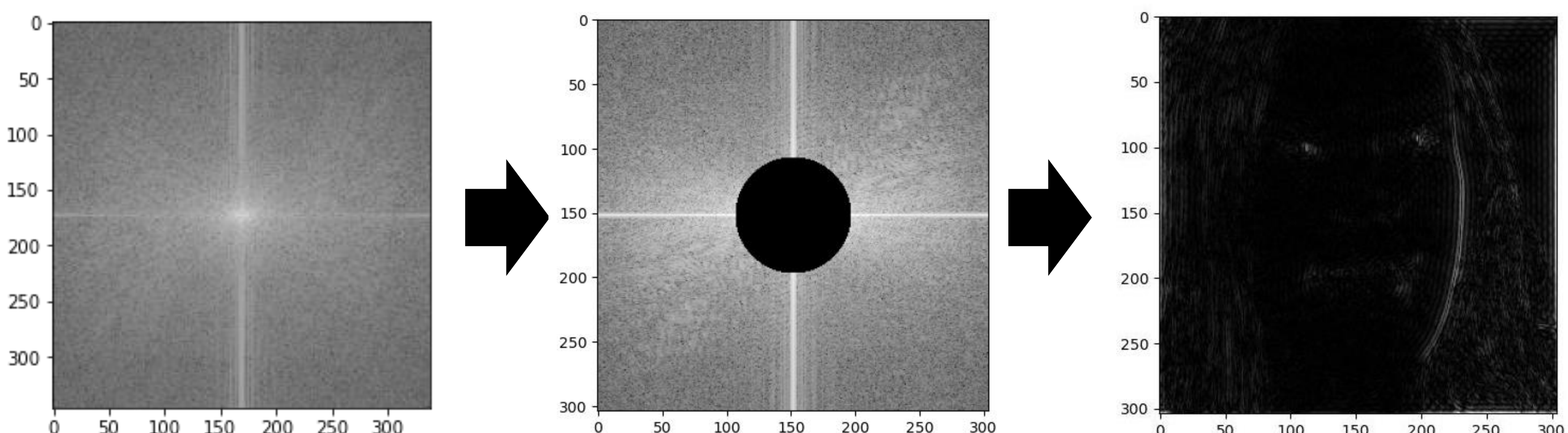
Implementation details

$$f(m,n) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \exp\left[2\pi i \left(\frac{mu}{M} + \frac{nv}{N}\right)\right] ,$$

$$m = 0, 1, \ldots, M-1 , \qquad n = 0, 1, \ldots, N-1 .$$

1. *High-Pass Filtering [2 pts]*
   1. *Make high pass filter and pass the frequency domain(gained from ‘task1’) through the filter*

   ※ recommend using radius 45 in pixel coordinate
2. *Implement 2D Inverse Fourier Transform Function [4 pts]*

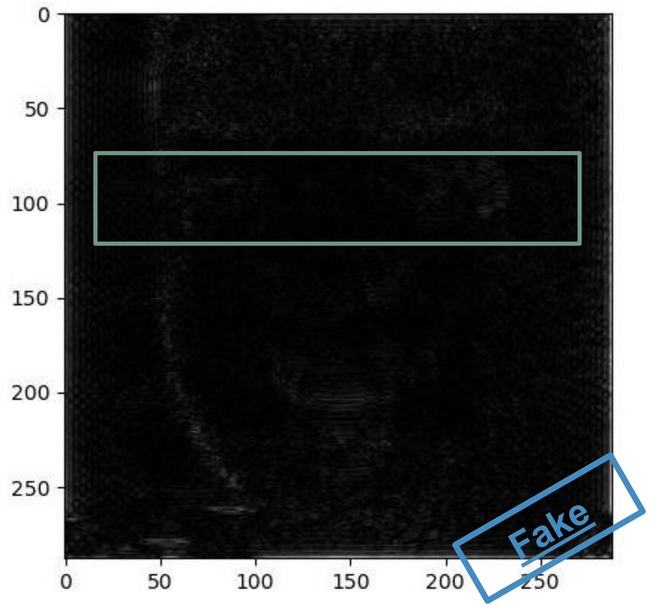


Highpass filter

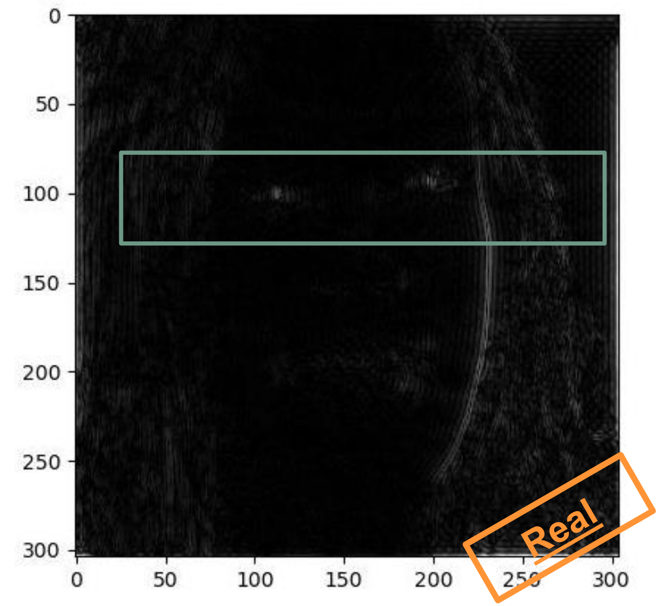
Task3 – High-pass filtering and 2D Inverse Fourier Transform [6pts]

3. *Compare the image with others. Especially focus on “Eyes”.*
   a. *The real image has high frequency in eyes.*

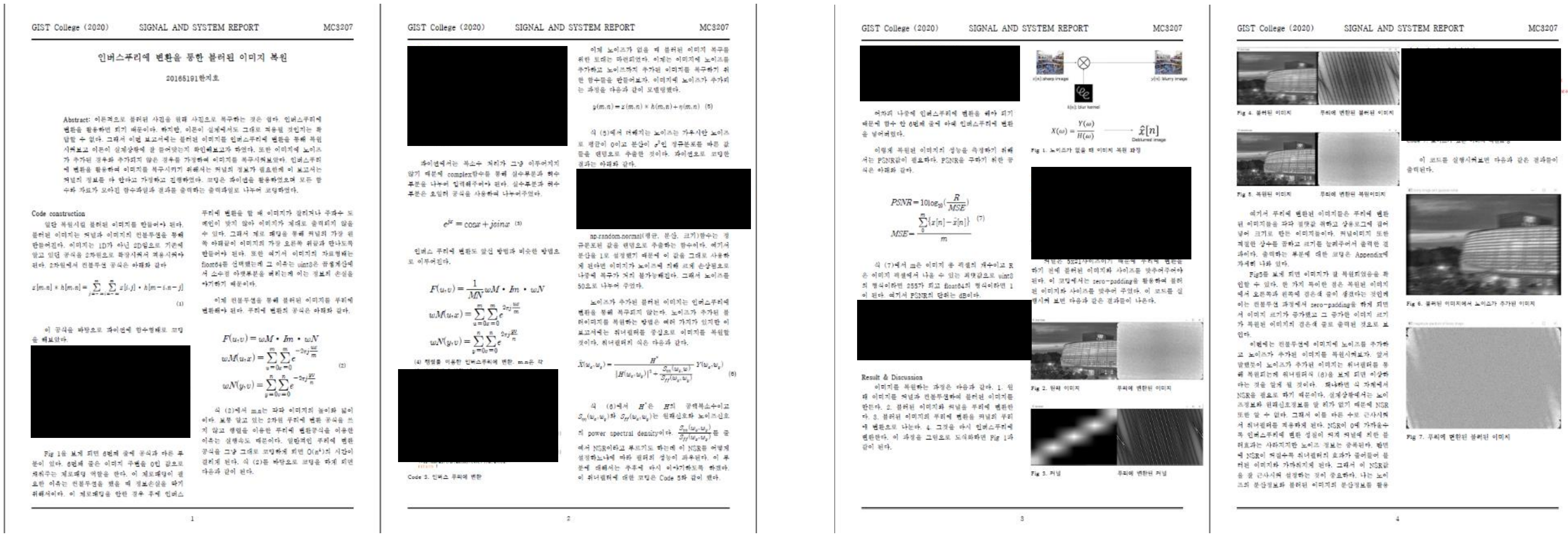
기준 (magnitude 등) 이상일 때 real

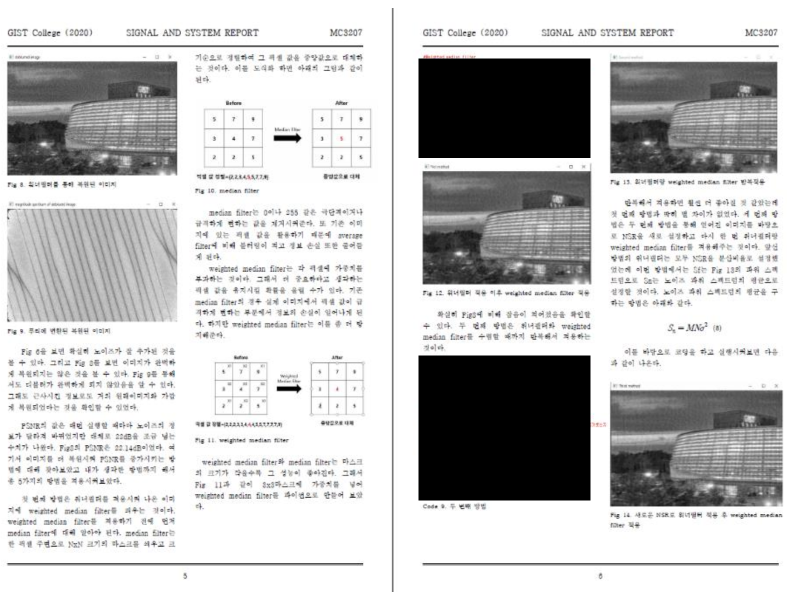


VS



Task4 - Write your report well [3pts]





Description of the background and your implementation with parts of source codes and images

Analysis your result based on your knowledge of your signals and systems.

For more scores [total 2pts]

Index	1D Power Spectrum	Filtering and 2D IFT	Conclusion
0	R	-	R(real)
1	F	R	F(fake)
2	F	-	F(fake)
3			
4			
5			
6			

1. Include the above format of a table in your report. I will check only a conclusion's column. You fill at least one blank space between '1D Power Spectrum' and 'Filtering and 2D IFT'. If you detect all the fake images, then you get more 1 point.
2. If you have to make better performances or a faster algorithm than that of professor's one, then you get more 1 point.

Deliverable and Cautions

Submit your source codes and a report including results in GEL site

You can use any programing languages

Due date: 11:59PM at 5/20 (Saturday)

NEVER use any function SUCH AS fft etc, except for data load and display. If you use any publicly available function, your score is zero!

No delay is allowed

Never copy source codes of your friends and online.
If your copy is observed, your credit for this course is definitely

F!!

Office hour

TA: Seonmi Park (Integrated, AI Graduate School)

Office hour: Mon/Tue 10:30~11:30

Place: Room 412 at EECS Building C

E-mail: bluesky1000@gm.gist.ac.kr

Phone: 062-715-6375

For more questions and appointments, please send me an e-mail.

Appendix - 2D Fourier and inverse Fourier Transform

Convolution

1D: $$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$$

2D: $$y[m,n] = x[m,n] * h[m,n] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i,j] \cdot h[m-i, n-j]$$

Fourier Transform

1D: $$F(k) \equiv \sum_{n=0}^{N-1} f(n)\, e^{\frac{-2\pi i k n}{N}}$$

2D: $$F(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \exp\left[-2\pi i \left(\frac{mu}{M} + \frac{nv}{N}\right)\right] ,$$

$$u = 0, 1, \ldots, M-1 , \qquad v = 0, 1, \ldots, N-1 ,$$

Inverse Fourier Transform

1D: $$f(n) \equiv \frac{1}{N} \sum_{k=0}^{N-1} F(k)\, e^{\frac{2\pi i k n}{N}}$$

2D: $$f(m,n) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \exp\left[2\pi i \left(\frac{mu}{M} + \frac{nv}{N}\right)\right] ,$$

$$m = 0, 1, \ldots, M-1 , \qquad n = 0, 1, \ldots, N-1 .$$