포트폴리오 설명

목차

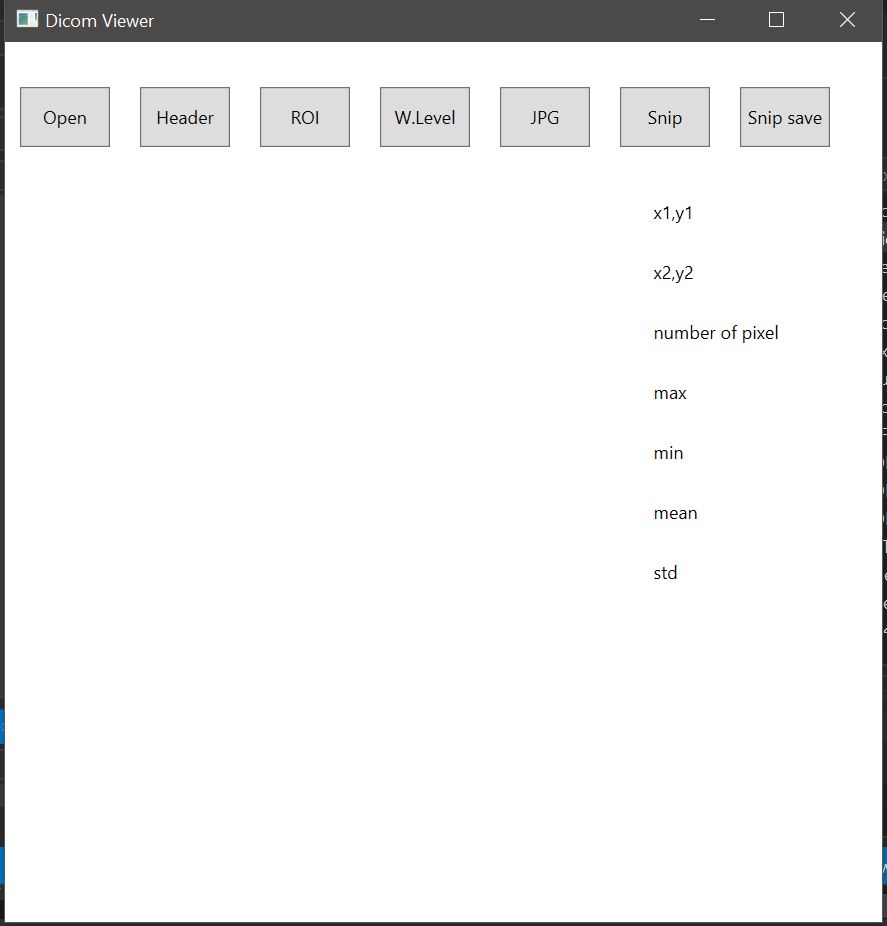
1. C#, WPF
   1. Dicom viewer
   2. Image processing
      1. RGB
      2. Edge detection
   3. Mvvm
2. Ai
3. 나머지
   1. Android
   2. GUI

Dicom viewer (C#, WPF)

1. Backgound Knowledge

* Dicom = 국제 표준 의료 영상 ( the international standard medical imaging information)
* Dicom viewer = software which will interpret the dicom information and display it as an image.
* 사용 언어 = C#

1. start UI

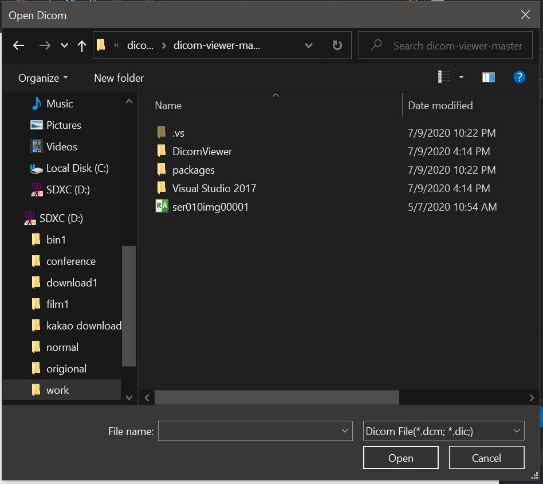
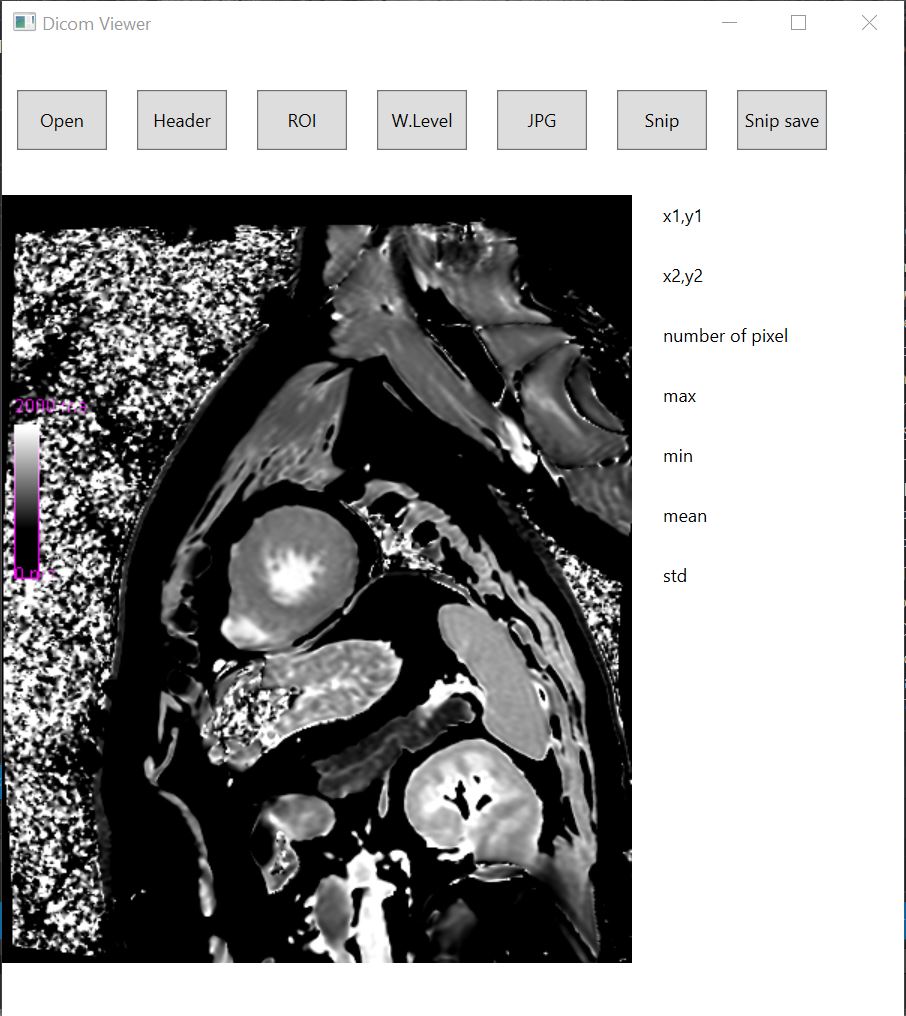


1. Button explaination

* open : display dicom medical image
* header : display dicom header
* ROI : drawing ellipse on the dicom image and measure every pixel in the ellipse
* W.Level : adjust contrast and brightness dicom image
* JPG : save dicom image into JPG file
* Snip : make snip image
* Snip save : save snip image into JPG file

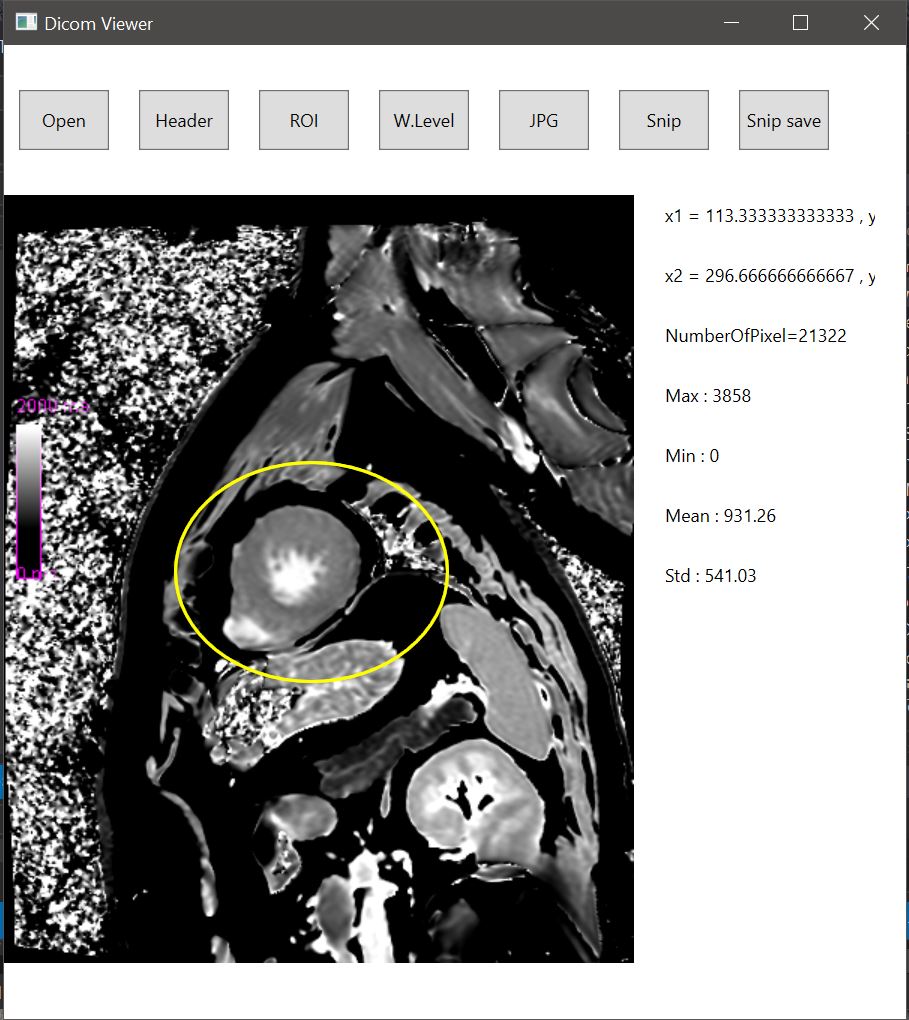
1. display dicom image

* click “ open button “ -> click “ ser010img00001 “

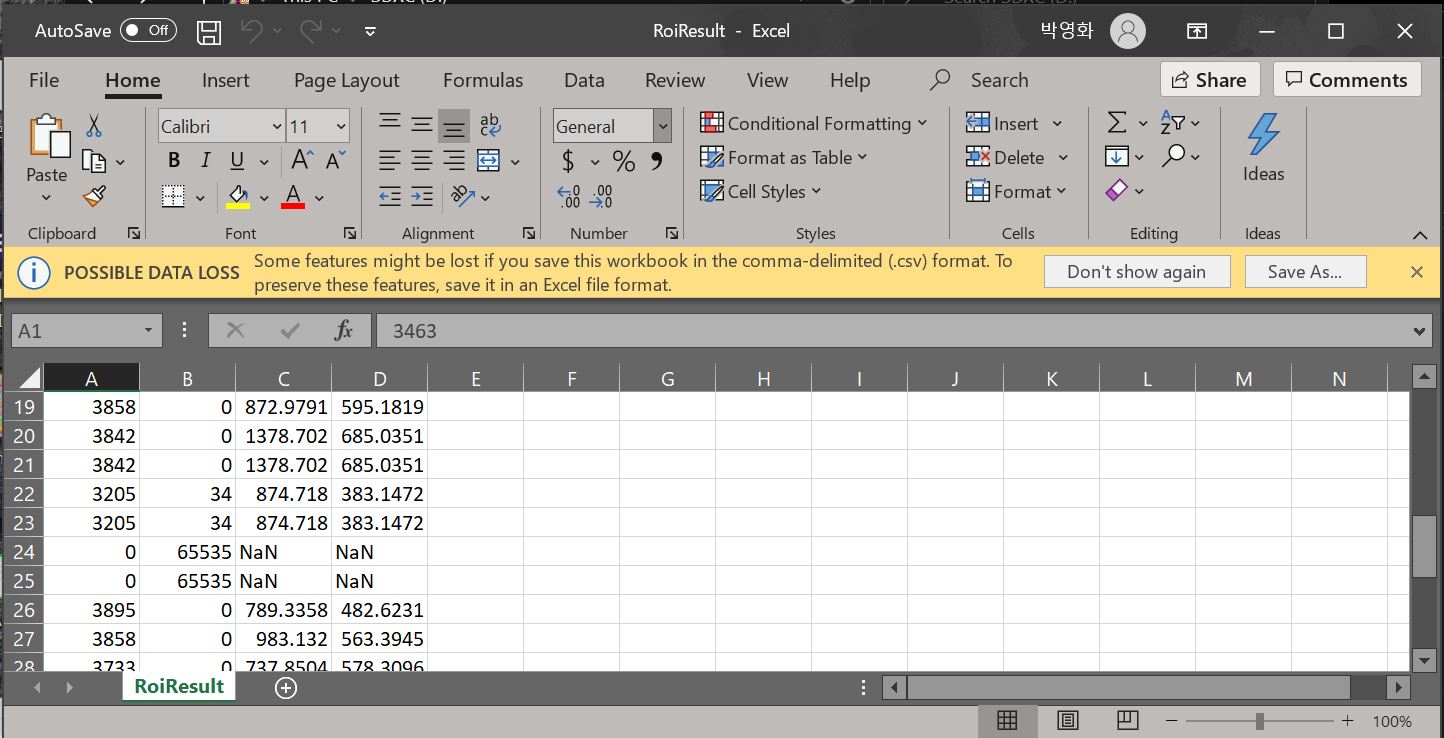


1. measure dicom

* measure every pixel in the ROI ellipse
* click “ ROI button “ -> mouse left click down on the image -> drag to right bottom side -> mouse left click up



* x1,y1 = first mouse click coordinate(첫번째 마우스 클릭 좌표)
* x2,y2 = second mouse click coordinate (두번쨰 마우스 클릭 좌표)
* Number of Pixel = number of pixel in the ellipse
* Max = the maximum pixel value in the ellipse
* Min = the minimum pixel value in the ellipse
* Mean = the mean value of every pixel in the ellipse (평균)
* Std = the standard deviation of every pixel in the ellipse (표준편차)
* make ROI measurement excel file in the D drive



1. display dicom header

* dicom header = the information of dicom
* click “ header button “ -> file -> load -> click “ 8786098\_POST ”

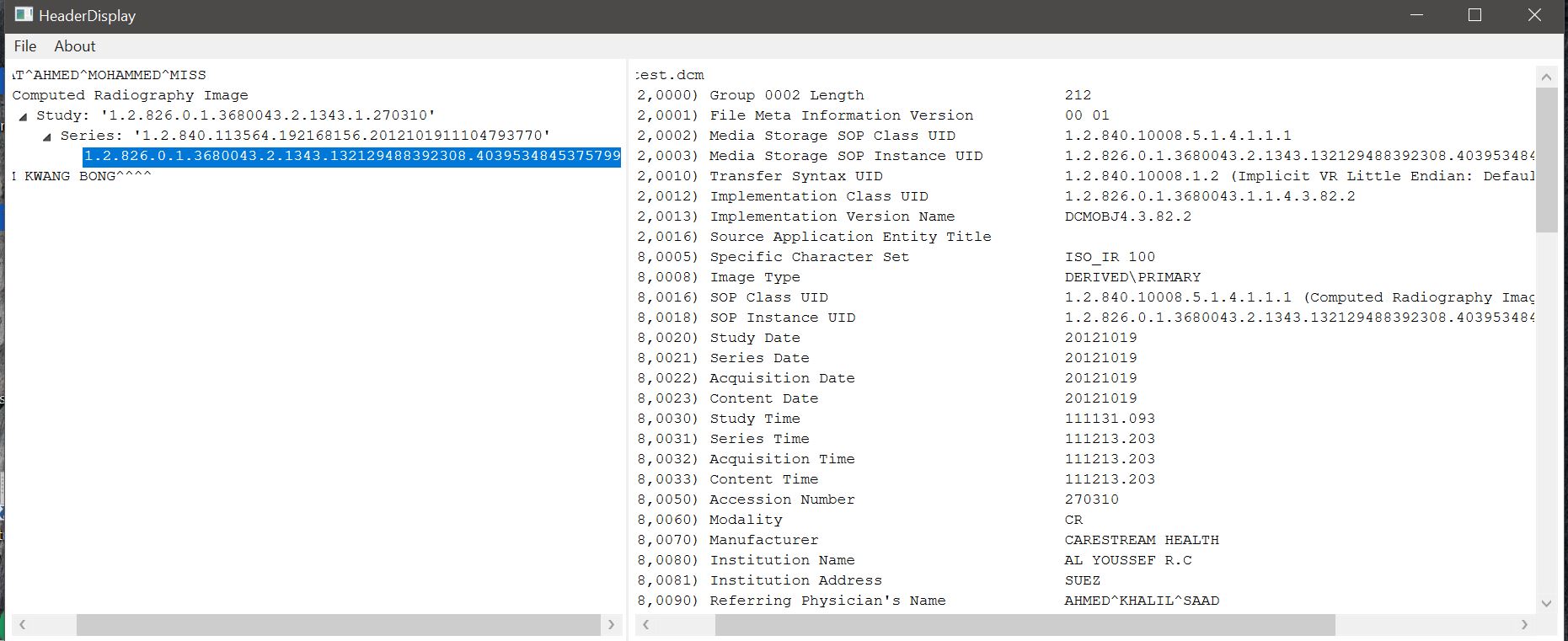
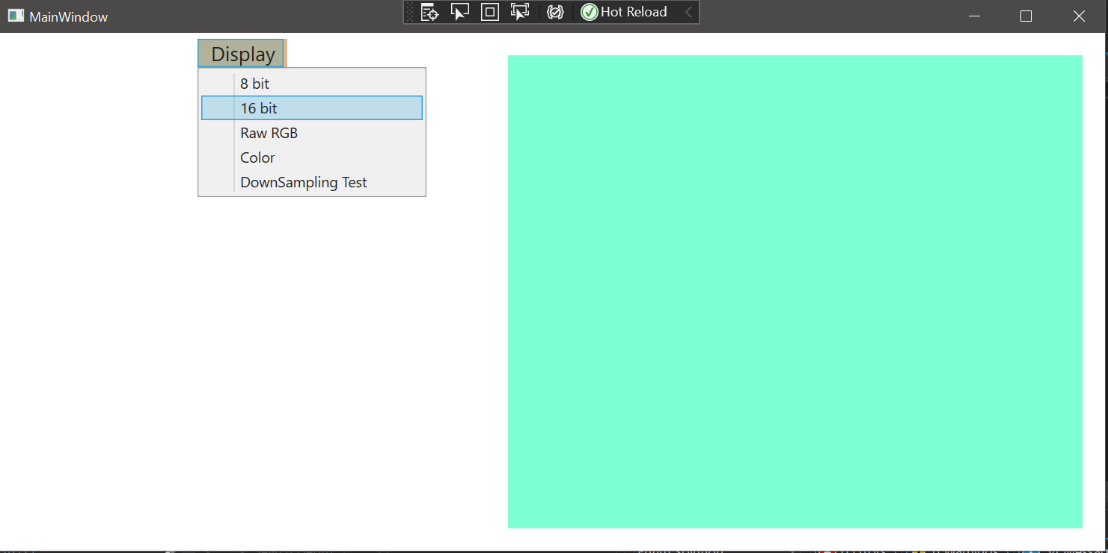


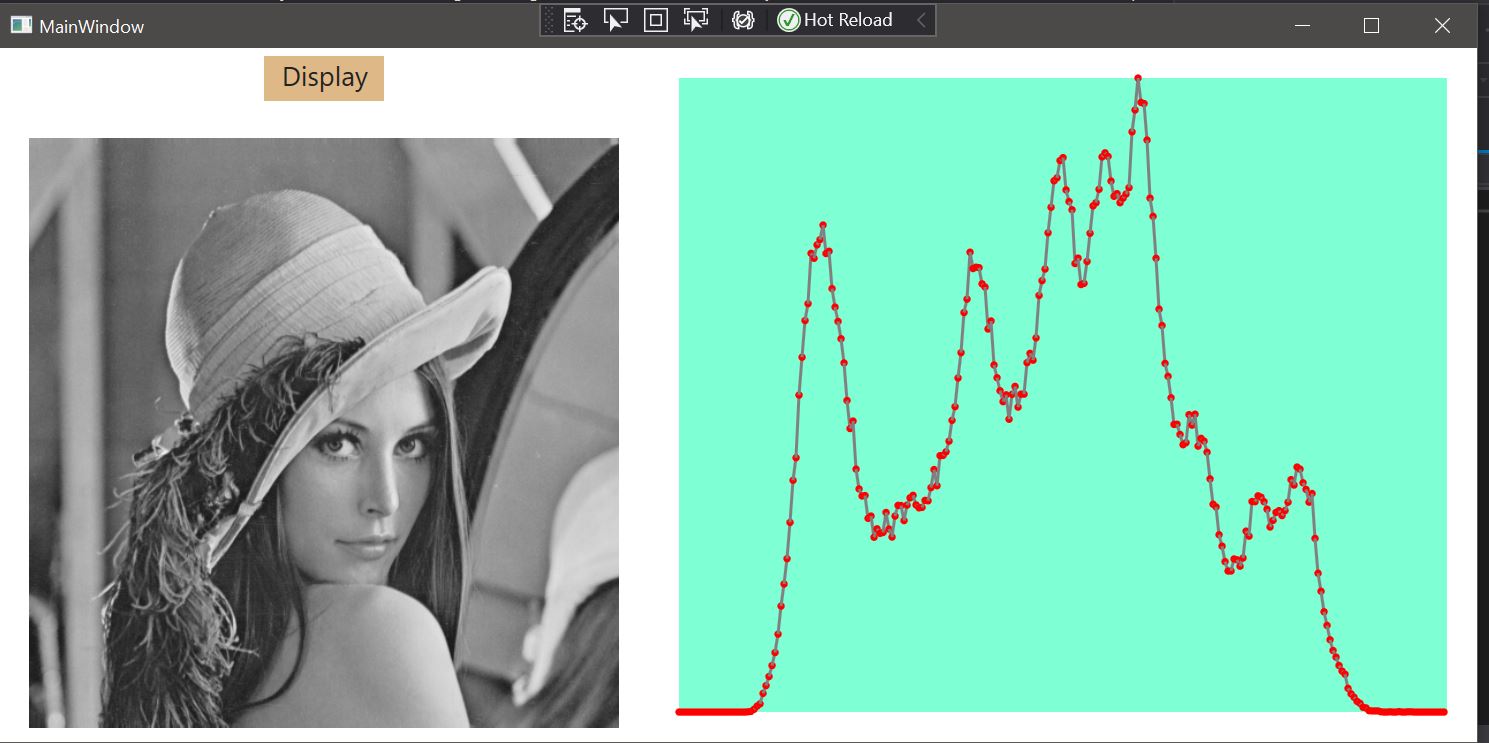
Image processing

RGB (C#, WPF )

1. 기능 : Image 에서 pixel value 에 대한 pixel 개수를 그래프로 나타냄
   1. X 축 : pixel value
   2. Y 축 : pixel 개수
2. 사용 언어 : C#, WPF
3. 실행순서
   1. 시작화면



* 1. Raw gray-scale graph display



* 1. RGB graph display

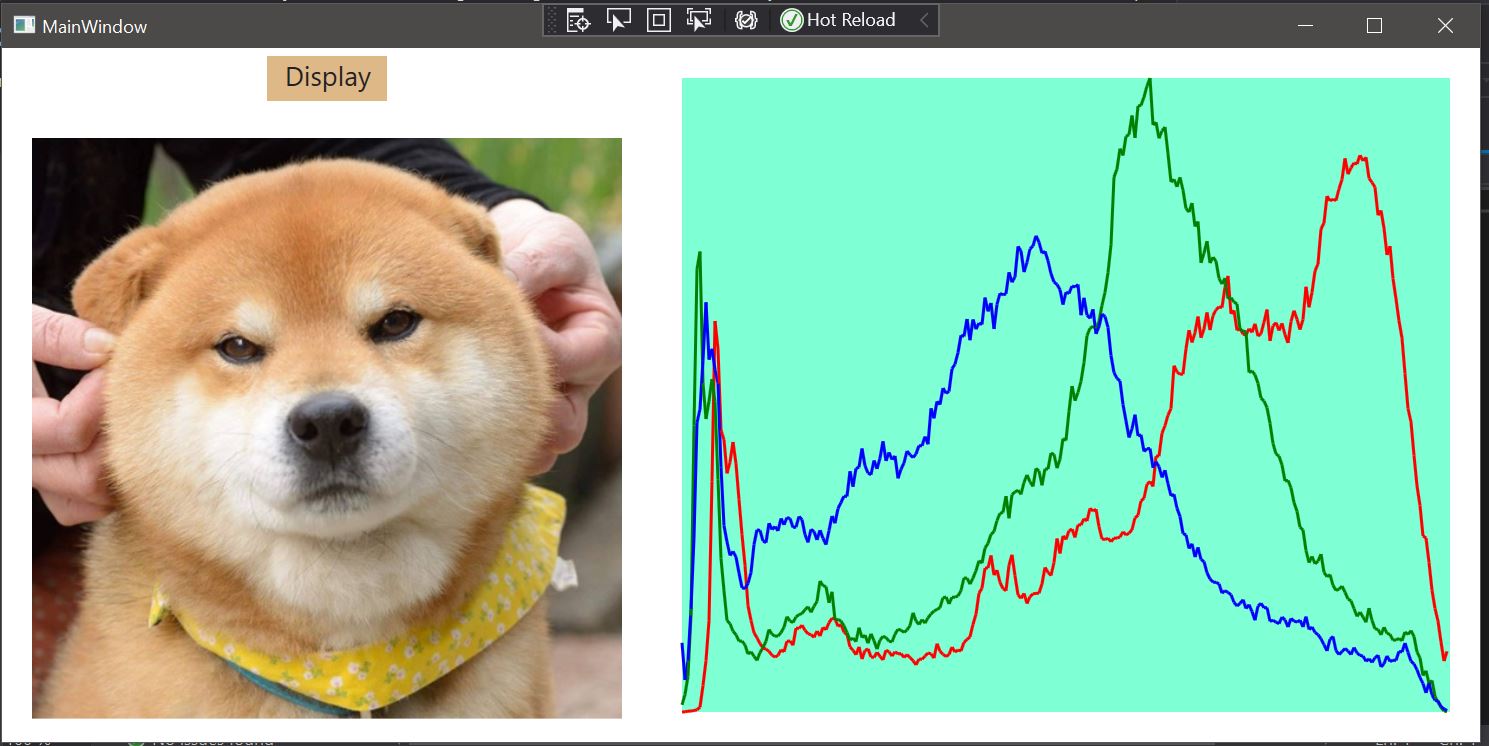


Image processing

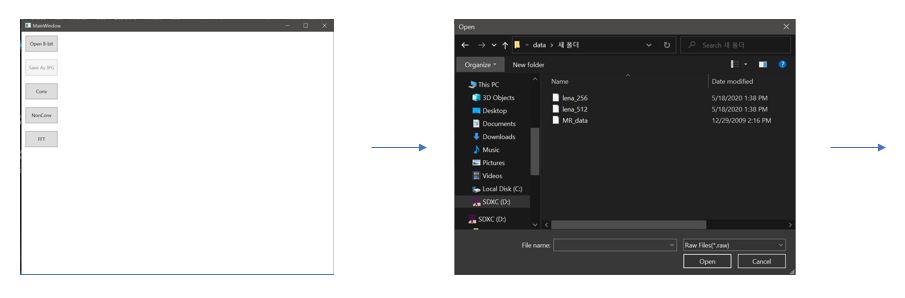
Edge detection (C#, WPF )

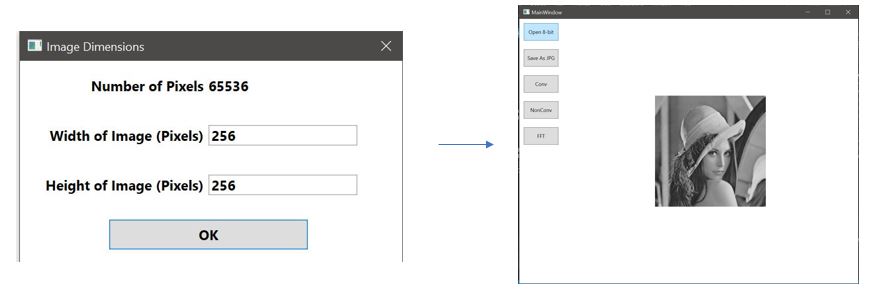
1. 설명

* Edge detection, MRI reconstruction, image processing 등의 기능이 있는 GUI program.

1. Display lena image

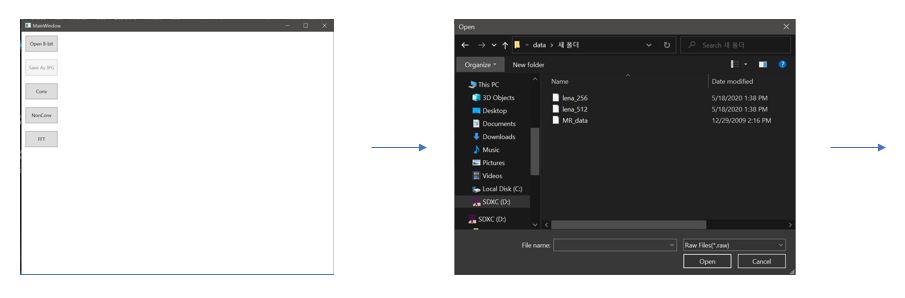
* Open 8-bit -> lena\_256 -> 0k

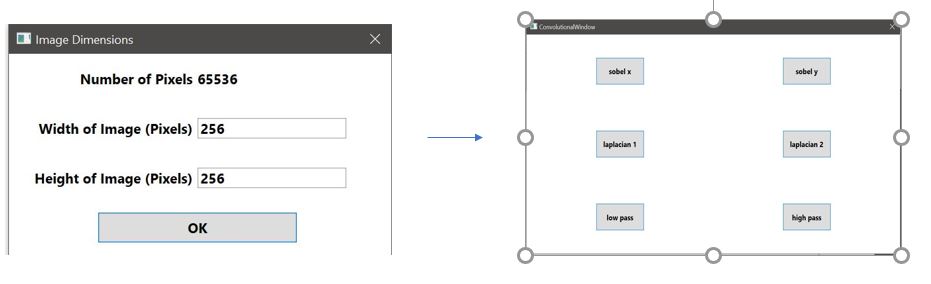




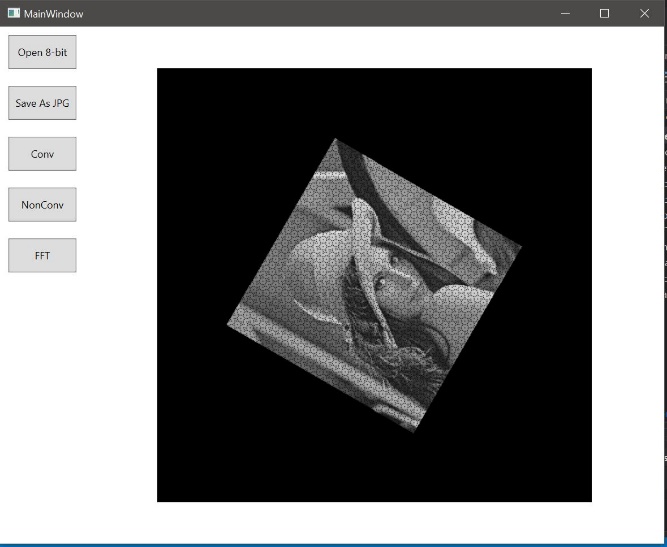
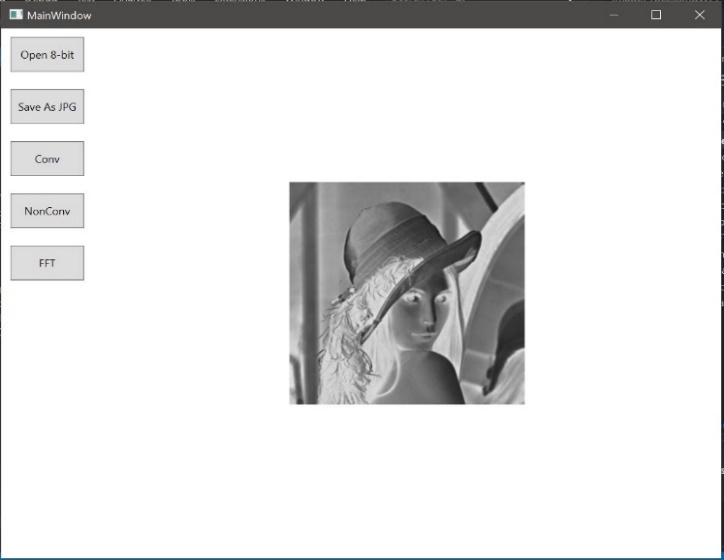
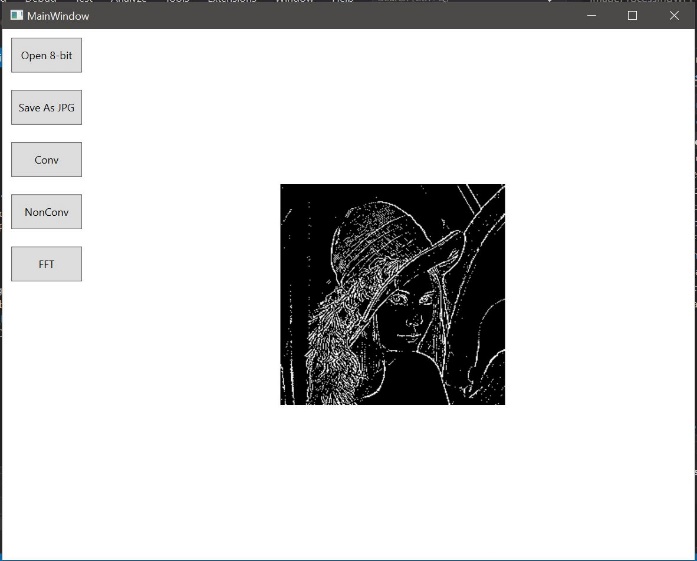
1. Edge detection

* Conv -> lena\_256 -> ok -> click something





* Image processing



Sobel y

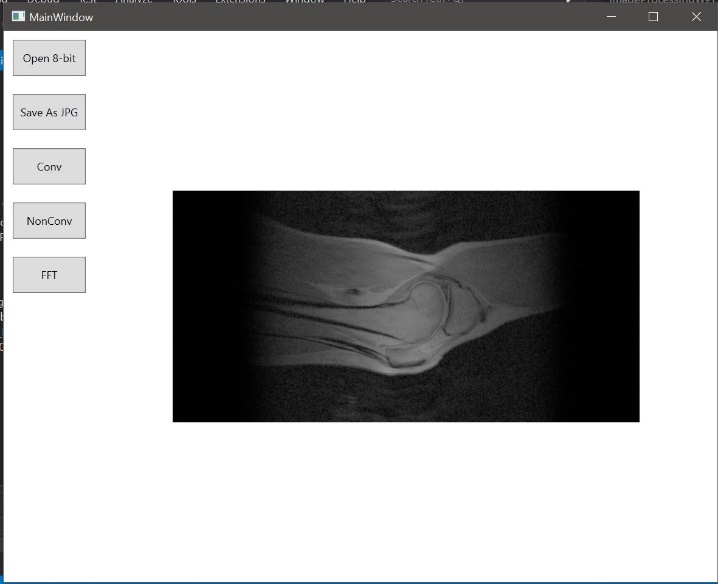
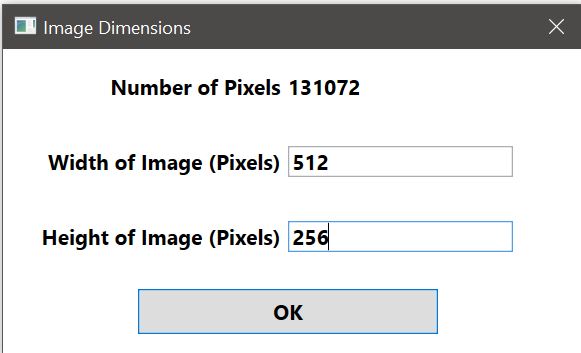
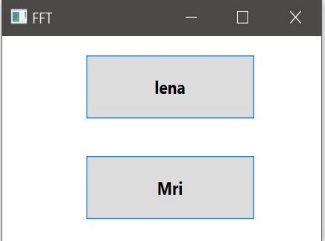
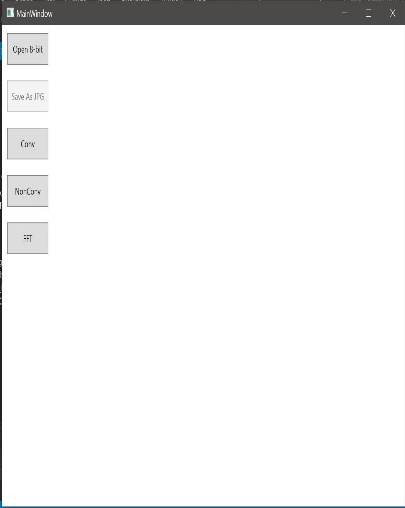
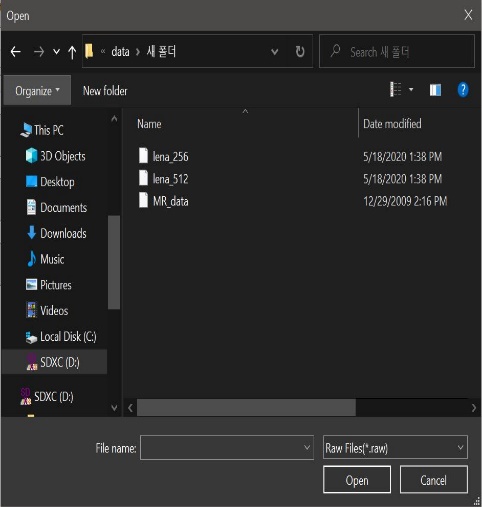
Laplacian 1

Inversion

Rotation

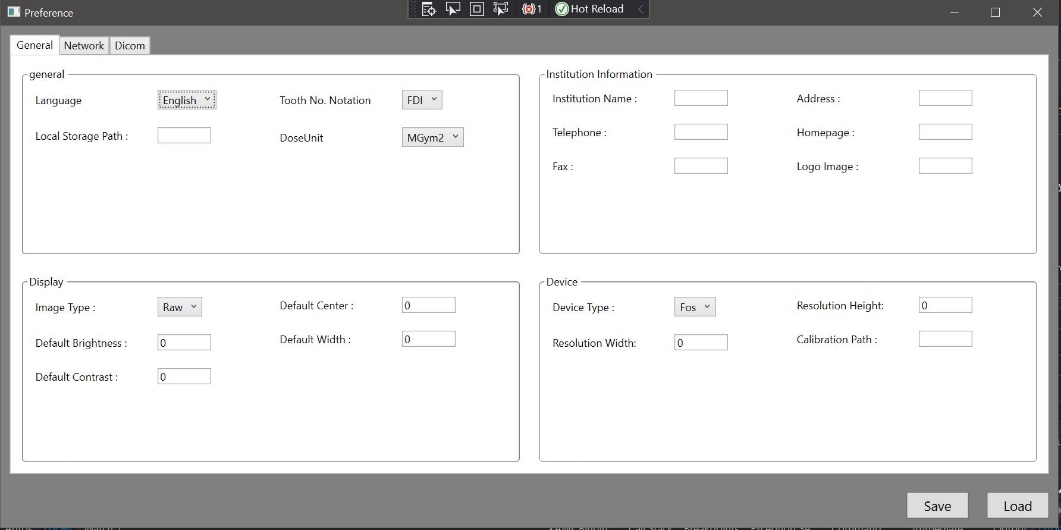
1. Display MRI

* FFT -> Mri -> MR\_data -> width of image (Pixels) = 512 , Height of Image (Pixels) = 256

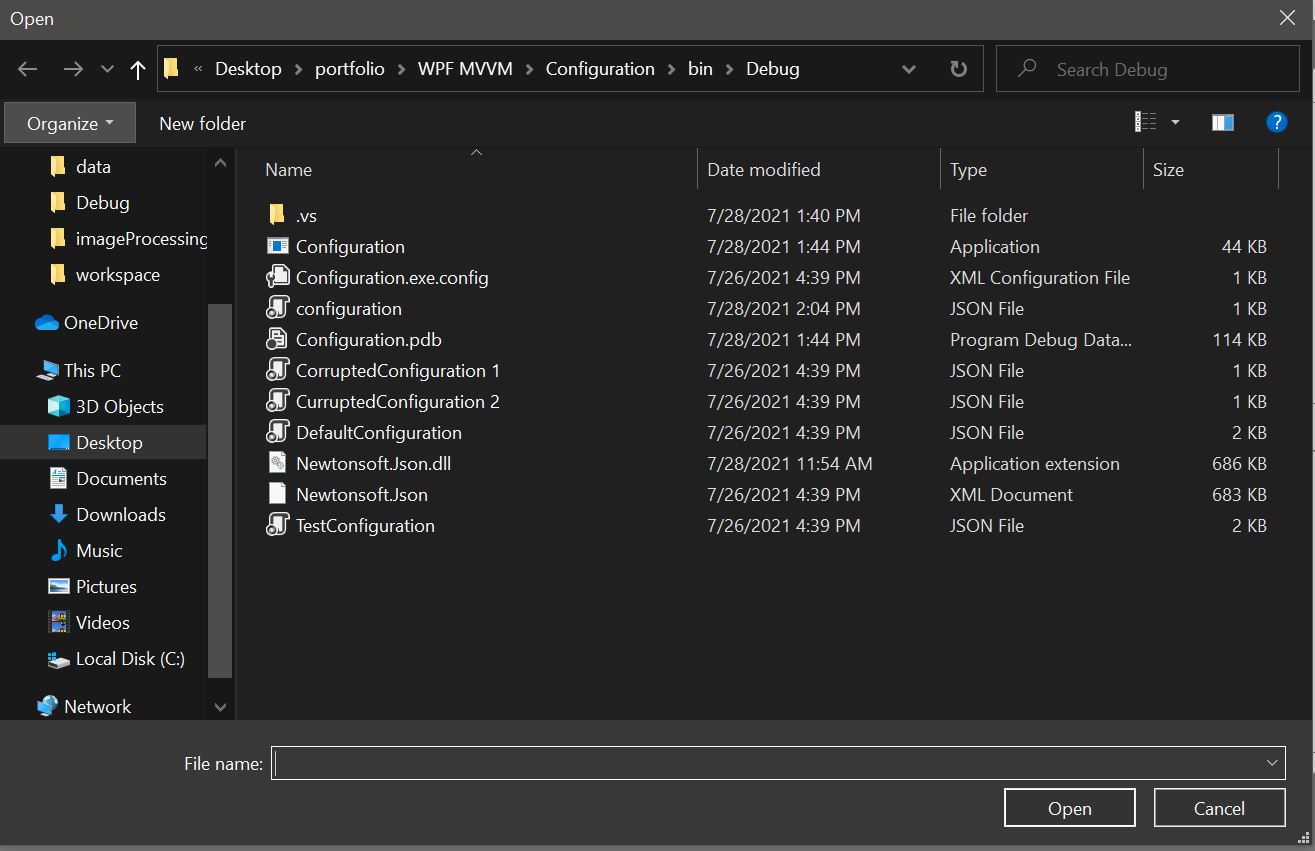


MVVM (C# , WPF)

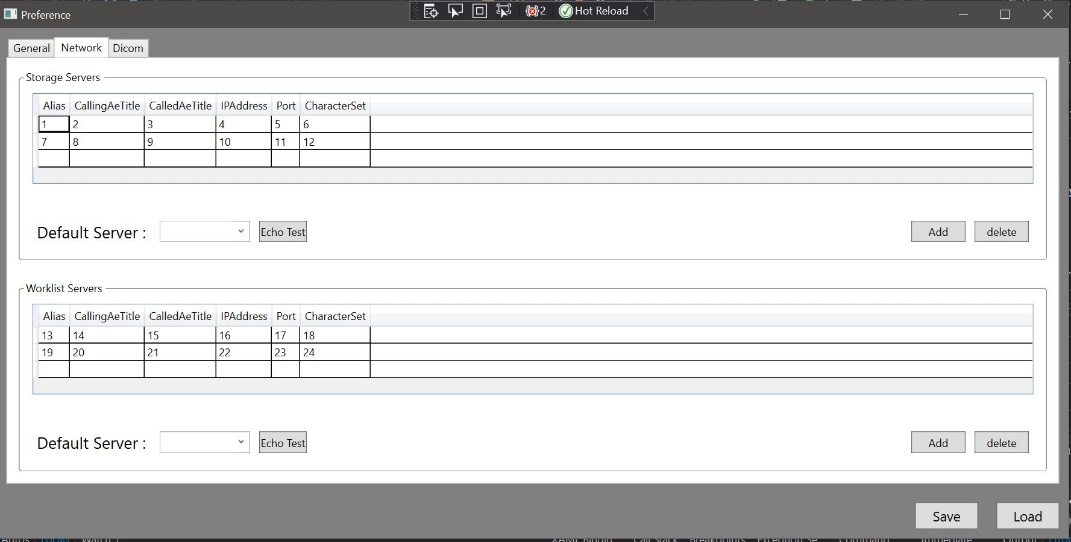
1. 설명
   1. 기능
      1. Save : 입력된 내용을 json 파일로 저장
      2. Load : json 파일 불러오기
   2. 특징 : MVVM 패턴에 맞게 구현함
2. 실행 순서
   1. 시작 화면

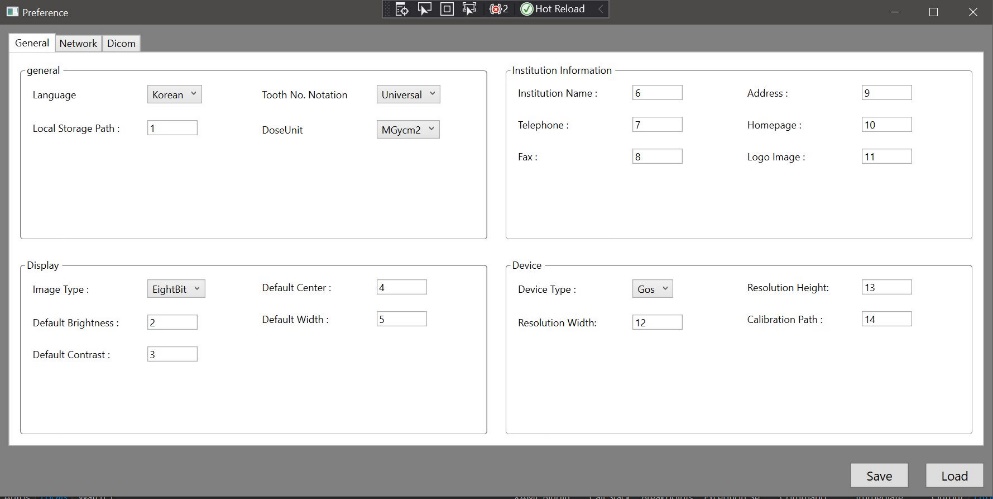


* 1. Load 버튼 클릭 🡪 json 파일 선택



* 1. 결과 확인





Deep learning (Python, keras )

: data 만 주어진 상황에서, 직접 deep learning regression model 찾고 코드 수정하여 개발 해보았습니다.

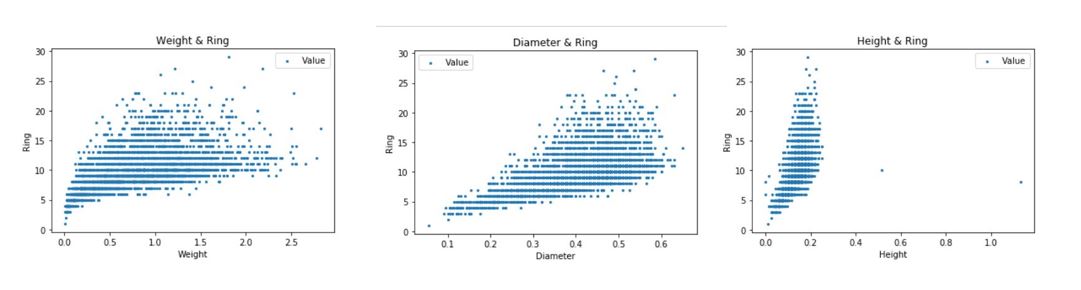
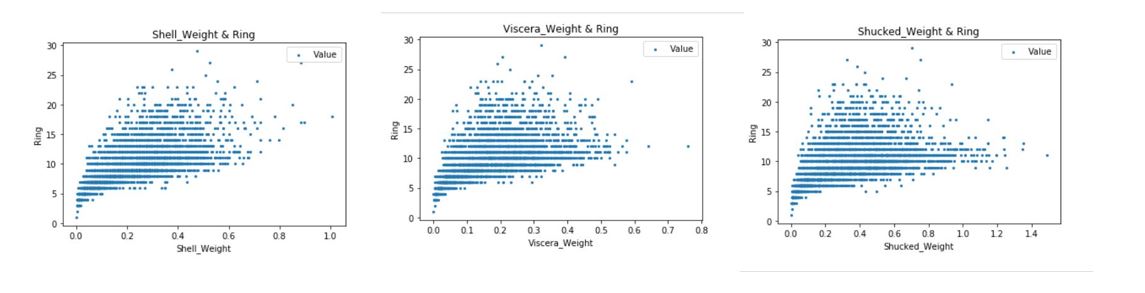
1. purpose

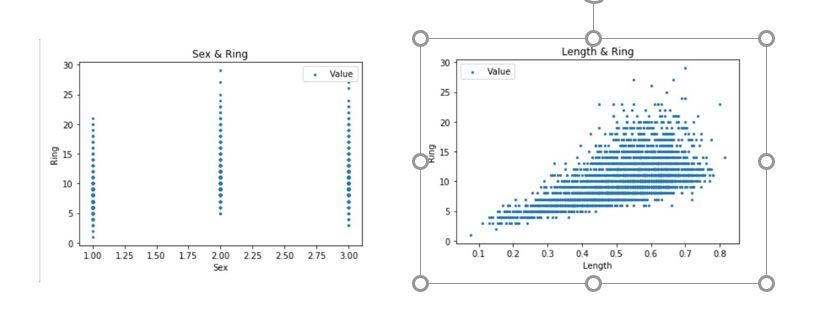
* 전복의 sex, Length, diameter, height, whole\_weight, shucked\_weight, viscera\_weight, shell\_weight 를 활용하여 Ring 의 개수를 예측하고 , ring 의 개수로 age를 확인한다.

Ex) ring 3 개 -> 3 years old , ring 5개 -> 5 years old

2. Data

* Data = ( sex, Length, diameter, height, whole\_weight, shucked\_weight, viscera\_weight, shell\_weight , ring )
* 전체 data 개수 = 4177개
* 성별은 수컷=3, 암컷 =2, 유성=1 로 치환
* Data 에서 단위는 표기 되어 있지 않음





3. Data classification

* Train data 개수 = 2506개(60%)
* Test data 개수= 836개(20%)
* Validation data 개수 = 835 개(20%)

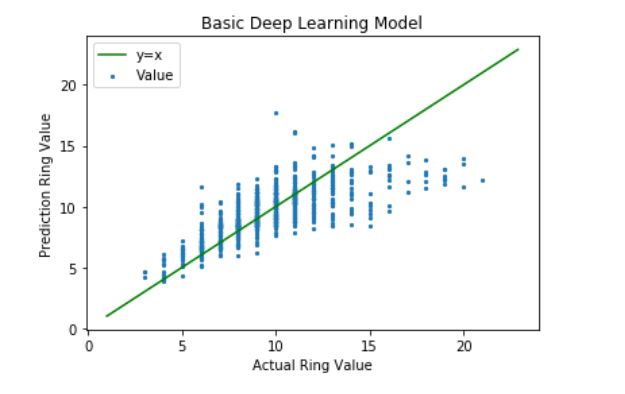
4. Training

* Optimizer = adam
* Loss function = mean squared error
* Epoch = 100
* Batch size = 1

5. Model comparison

|  |  |  |  |
| --- | --- | --- | --- |
|  | basic model | many layer model | many neuron model |
| Layer construction | (Input layer)  node 개수 = 64개  activation fuction = relu  (Hidden layer)  Layer 개수= 4개  node 개수 = 32개  activation fuction = relu  (Output layer)  node 개수 =1개 | (Input layer)  node 개수 = 64개  activation fuction = relu  (Hidden layer)  Layer 개수= 20개  node 개수 = 32개  activation fuction = relu  (Output layer)  node 개수 =1개 | (Input layer)  node 개수 = 128개  activation fuction = relu  (Hidden layer)  Layer 개수= 4개  node 개수 = 64개  activation fuction = relu  (Output layer)  node 개수 =1개 |
| performance | Rmse = 1.78  R2=0.56 | RMSE=2.00  R2=0.44 | RMSE = 2.16  R2 = 0.35 |

6. conclusion



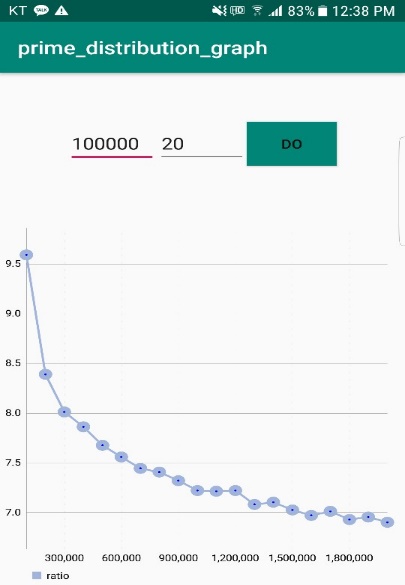
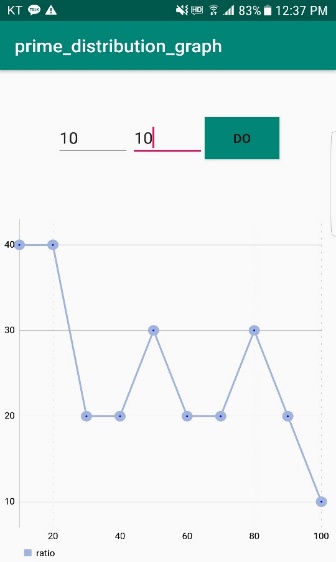
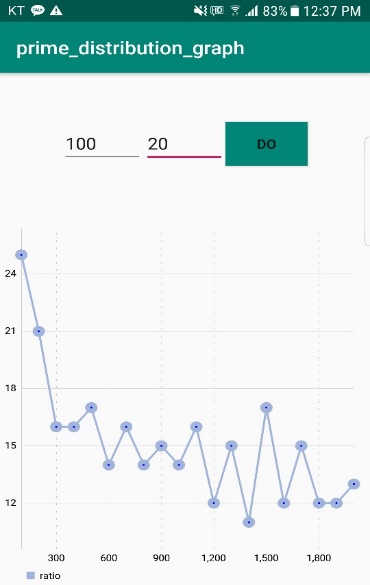
* Value 가y=x 에 근사 할수록 성능이 우수하다고 할 수 있다
* layer, neuron 를 증가 시킨다고 성능이 항상 개선 되지 않는다.

Android app (java, android)

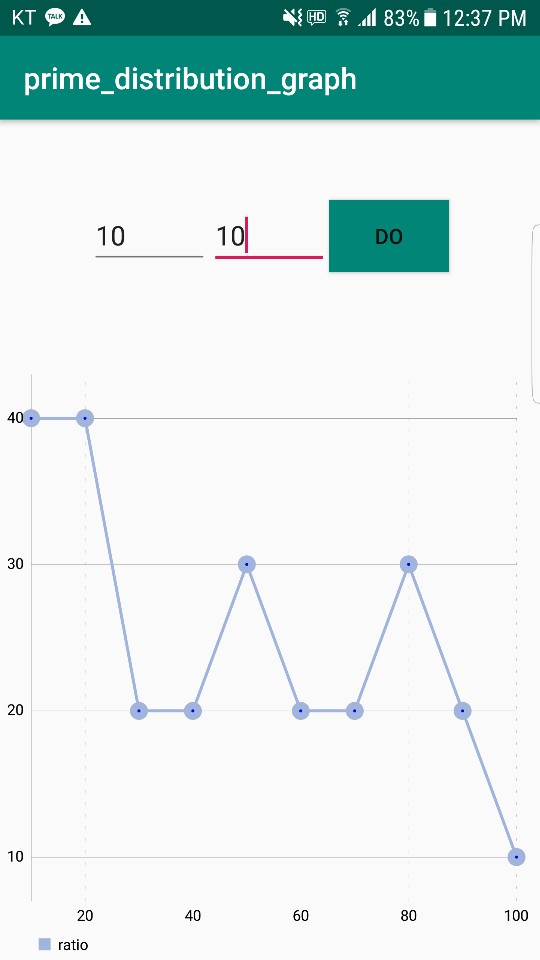
* 설명
* 구간 안에 있는 소수의 비율을 그래프로 그려주는 어플입니다.
* 소수 : 1과 자기 자신만으로 나누어 떨어지는 1보다 큰 양의 정수

Ex) 2,3,5,7,11,13,17 …

* + 사용 언어 = JAVA



* explaination
* 입력값 : 첫번째 입력값은 구간 길이이고 두번째 입력값은 그래프 점의 개수입니다.
* X 축 : 수의 범위
* Y 축 : 소수의 비율
* Example
* Input : 10,10
* 점 (10,40) 은 1부터 10 사이에 소수가 4 개 있다는 의미입니다.
* 점 (20,40) 은 11부터 20까지 소수가 4개 있다는 의미입니다.



모자이크 (Python)

1. Description

* 모자이크 그림을 만들수 있습니다.

