

# Aneurysm Visualization Module

NEURO세미나  
AI연구2팀인턴 정수미

2024.01.30

[ D E E P  
N O I D ]

## 목차 | CONTENTS

I Summary

II Method

III Result

# | Summary

## ■ Aneurysm Visualization Module

### ■ Summary

- ✓ Input : numpy array(.npz )
- ✓ Output : 2D image(.jpeg, 3D-like)
- ✓ (1) View point adjustment
- (2) Aneurysm Slice Extraction & Concatenation

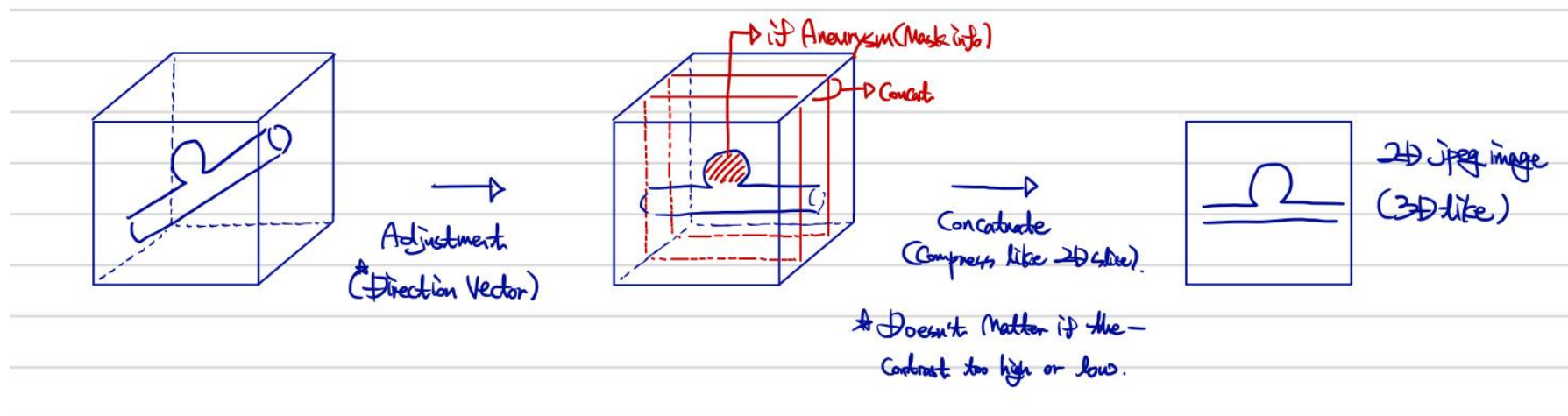


Fig 1. Flowchart

## ■ Aneurysm Visualization Module

### ■ Dataset

✓ Data: npz format, 하나의 파일은 환자 한 명 (Patient-wise)

- DICOM - (1)Vessel Segmentation (2) Patch Extraction → patch → .npz
- center: 각 패치의 중심점
- direction: 각 패치의 방향벡터 ( $[x_0, y_0, z_0]$ ,  $[x_1, y_1, z_1]$ ,  $[x_2, y_2, z_2]$ )
  - $x_0, y_0, z_0$ : camera location
  - $x_1, y_1, z_1$ : direction /  $x_2, y_2, z_2$ : angle
- data: 패치 데이터 (Vessel Segmentation output)
- mask: 패치 데이터 (Aneurysm Mask; GT)
- Intersection\_over\_aneurysm: IoU

```
1. center: (n, 3)          | centre point (vessel)
2. direction: (n, 9)       | direction vectors (x0 ~ x2, y0 ~ y2, z0 ~ z2)
3. data: (n, 32, 32, 32)  | inputdata patch
4. mask: (n, 32, 32, 32)  | mask value (for segmentation)
5. intersection_over_aneurysm: (n, 1) | IoU(ratio, float)
```

Fig 1. dataset

## ■ Aneurysm Visualization Module

### ■ Dataset

- ✓ All abnormal, Single Aneurysm, 2~3 patients의 Patch 1개 선택 (Maximum IoU value)
  - HHHHH\_CA\_0000{XX} → 22번 (199), 34번 (742), 62번 (252)
    - Patch Selection : IoU=1인 patch의 연속된 index들 중에서 중앙값을 선택

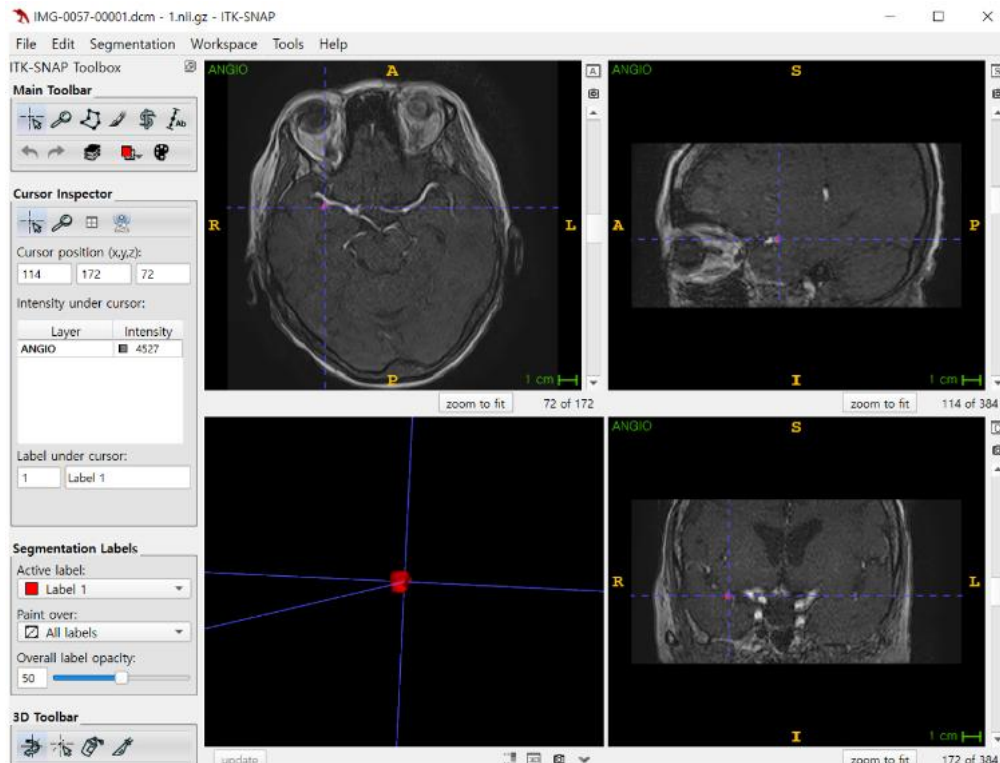


Fig 1. HHHHH\_CA\_000062.npz

```
Keys in the npz file: ['center', 'direction', 'data', 'mask',
Shape of center: (348, 3)
Shape of direction: (348, 9)
Shape of data: (348, 32, 32, 32)
Shape of mask: (348, 32, 32, 32)
Shape of intersection_over_aneurysm: (348, 1)
```

Fig 1. data shape

```
Selected indices: [231, 232, 233, 234, 235, 236, 237, 238, 239,
240, 241, 251, 252, 253, 254, 255, 256, 257,
258, 259, 260, 261, 262, 263, 264]
```

Fig 2. Selected indices

```
direction : [ 1.00000000e+00  1.18533117e-14 -2.13162821e-14
2.18574607e-14. -4.77036791e-02  9.98861531e-01
1.08229520e-14 -9.98861531e-01. -4.77036791e-02]
center : [-38.4984 -36.82375418 -27.1373649 ]
intersection_over_aneurysm : [1.]
```

Fig 3. Selected patch Info



# Method

## ■ Aneurysm Visualization Module

- Viewpoint adjustment : 단계별로 진행
  - ✓ (1단계) Optimized Vessel View : Z축을 기준으로 90도 회전
  - ✓ (2단계) 3D Aneurysm Centroid
  - ✓ (3단계) Patch Rotation Method : 방향벡터에 수직 & aneurysm의 중심을 지나는 벡터를 포함하는 평면 회전

| Library                 | Function         |
|-------------------------|------------------|
| numpy                   |                  |
| matplotlib              |                  |
| scipy.spatial.transform | rotation         |
| scipy.ndimage           | affine_transform |
|                         | rotate           |
| Numpy.linalg            | norm             |

Fig 1. 사용한 라이브러리



## ■ Aneurysm Visualization Module

### ■ 3D patch Visualization in 2D slice

- ✓ Full Slice
- ✓ Overlay

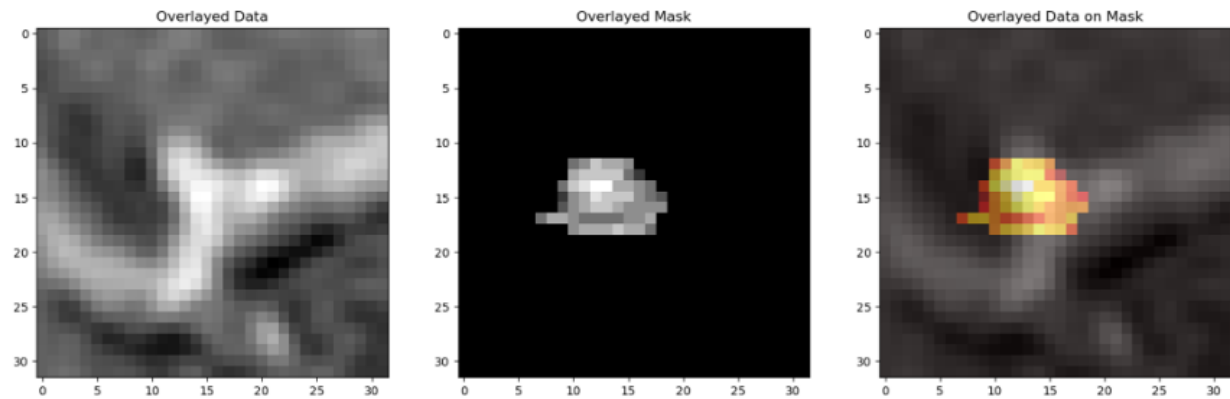


Fig 2. Overlay

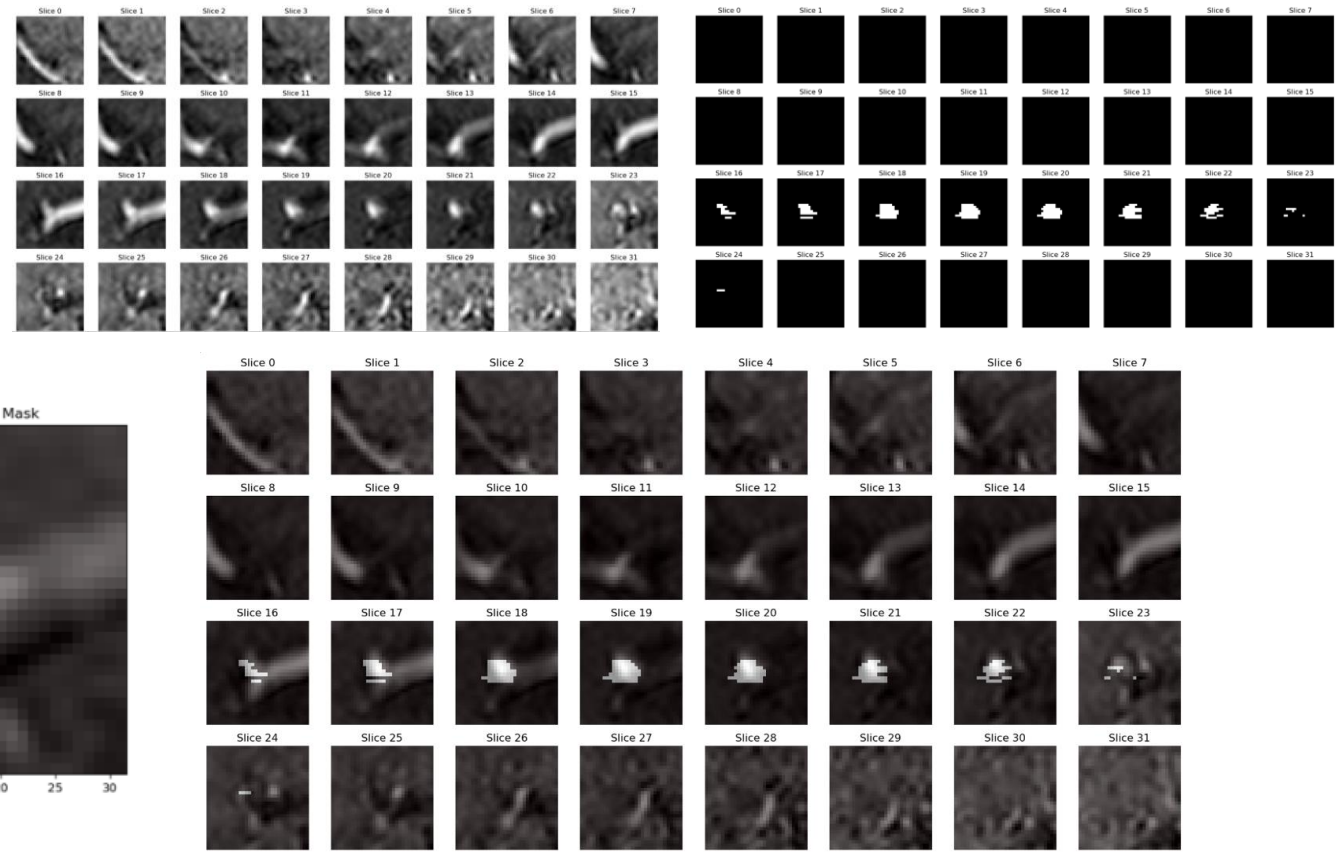


Fig 1. Full Slice

## ■ Aneurysm Visualization Module

### ■ Viewpoint adjustment : 단계별로 진행

#### ✓ (1단계) Optimized Vessel View

- 혈관의 주행방향대로 잘리는 patch와 목표결과(True Lateral View)를 고려하여 Z축을 기준으로 90도 회전 적용
- 방향벡터(direction)에도 동일한 회전 적용

```
direction: [ 1.00000000e+00  1.18533117e-14 -2.13162821e-14
            2.18574607e-14 -4.77036791e-02  9.98861531e-01
            1.08229520e-14 -9.98861531e-01 -4.77036791e-02]
rotated_direction: [ 1.00000000e+00  4.77036791e-02 -2.13162821e-14
                   2.18574607e-14  1.18503907e-14  9.98861531e-01
                   1.08229520e-14 -9.98861531e-01 -4.77036791e-02]
```

Fig 2. direction, rotated\_direction

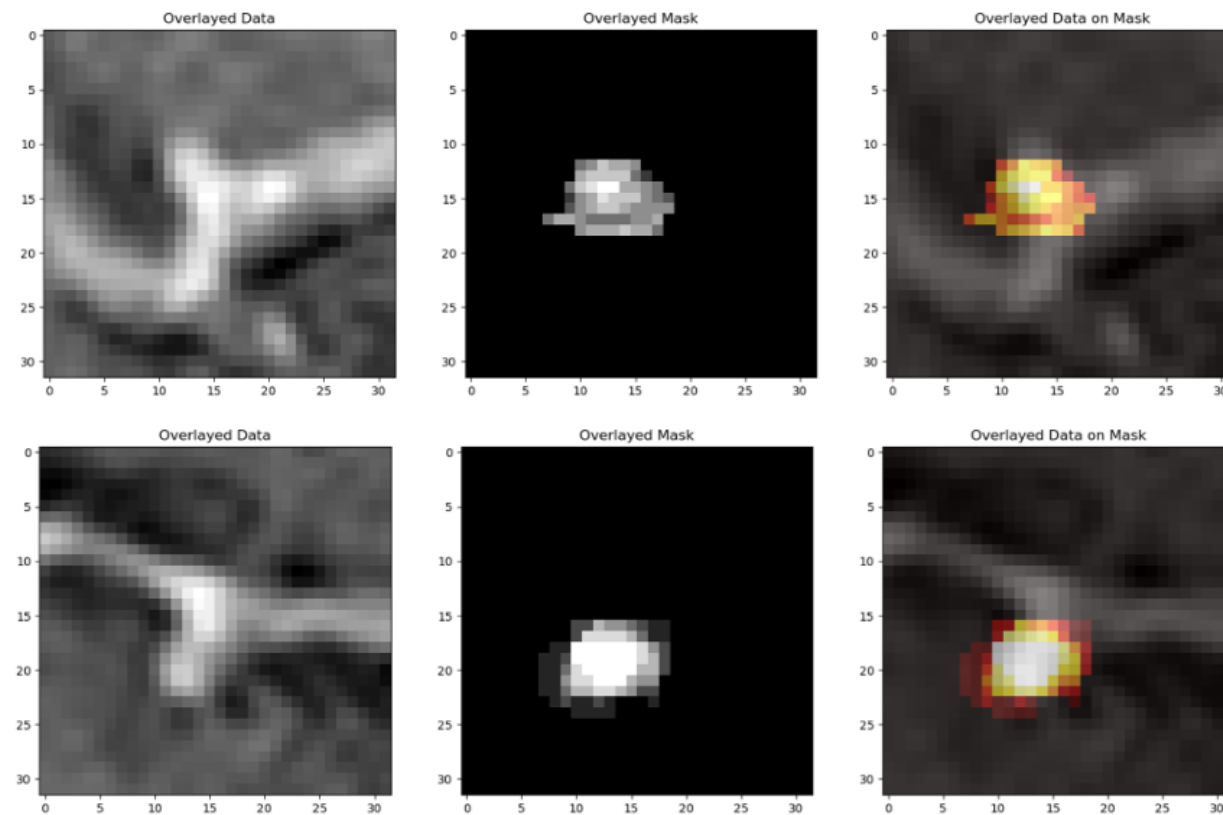


Fig 1. Optimized Vessel View Result

## ■ Aneurysm Visualization Module

- Viewpoint adjustment : 단계별로 진행

- ✓ (2단계) 3D Aneurysm Centroid

- 데이터셋의 center 값은 patch의 중심으로 Aneurysm의 중심과 대부분 일치 X
- Mask=1인 좌표들의 평균을 구해 3D Aneurysm의 Centroid

```
Aneurysm centroid: [15.81952663 19.40828402 12.93491124]
```

Fig 1. Aneurysm Centroid

## ■ Aneurysm Visualization Module

- Viewpoint adjustment : 단계별로 진행
  - ✓ (3단계) Patch Rotation Method
  - ✓ 방향벡터에 수직 & aneurysm의 중심을 지나는 벡터를 포함하는 평면 회전 (법선벡터가 z축과 평행)
    - 방향벡터에 수직 & aneurysm의 중심을 지나는 벡터 (perpendicular\_vector\_through\_centroid)

```
perpendicular_vector_through_centroid: [ 0.00000000e+00 -1.00000000e+00 -1.24344979e-14]
```

Fig 1. perpendicular\_vector\_through\_centroid

- '혈관의 방향벡터(Vessel\_direction)' 와 '방향벡터에 수직&aneurysm의 중심을 지나는 벡터'를 포함하는 평면(plane\_equation)
  - 해당 평면의 법선벡터(plane\_normal) Z축과 평행하도록 회전 적용

```
Plane normal vector: [-9.98861531e-01  5.93171297e-16 -4.77036791e-02]  
Plane equation coefficients: [-9.98861531e-01  5.93171297e-16 -4.77036791e-02  1.64185594e+01]
```

Fig 2. Plane\_equation, normal vector

## ■ Aneurysm Visualization Module

- Viewpoint adjustment : 단계별로 진행
  - ✓ (3단계) Patch Rotation Method
  - ✓ 방향벡터에 수직 & Aneurysm의 중심을 지나는 벡터를 포함하는 평면 회전 (법선벡터가 z축과 평행)

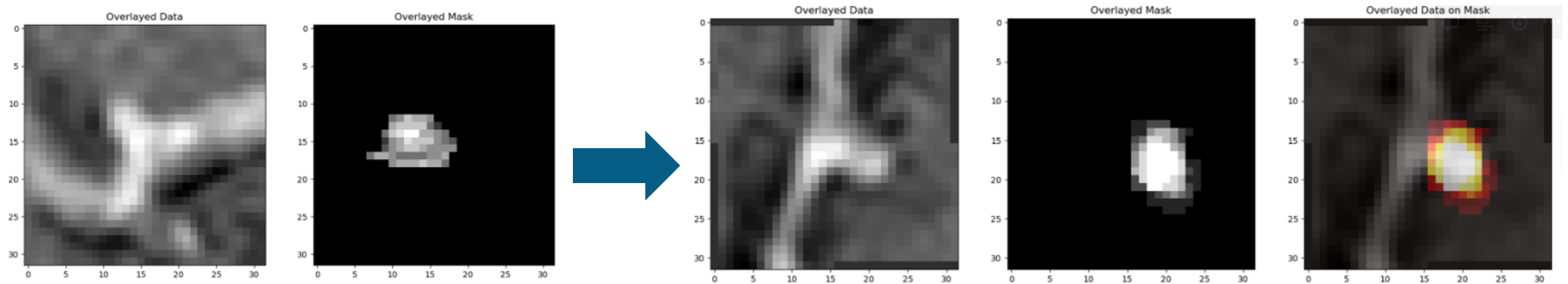


Fig 1. Viewpoint adjustment Result



Result

[ D E E P  
N O I D ]

## ■ Aneurysm Visualization Module

### ■ Viewpoint adjustment

#### ✓ Rotation 방식 적용 결과

- (한계) 예외사항에 대한 처리 부족

=> True Lateral이 아닌 경우, 최종 결과에서 추가적인 회전이 필요한 듯 보임

#### ✓ 해결방안

- (방법 1) True Lateral을 위한 최적의 회전각 산출 알고리즘
- (방법 2) 3D mask의 centroid를 지나면 혈관의 주행방향과 수직인 벡터와 평행하게 압축

True Lateral

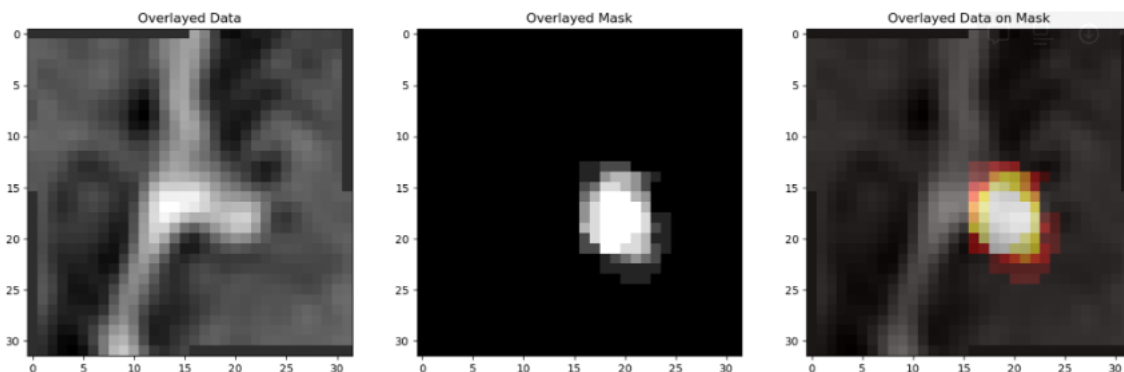


Fig 1. HHHHH\_CA\_000062 (252)

True Lateral X

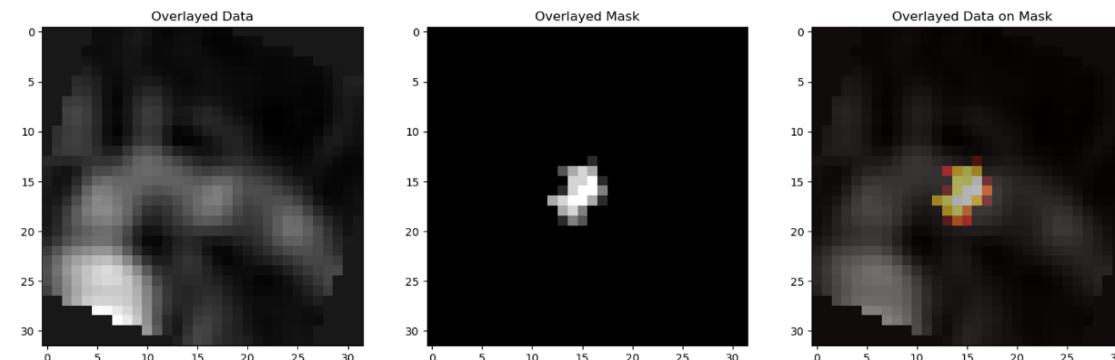


Fig 2. HHHHH\_CA\_000022 (199)

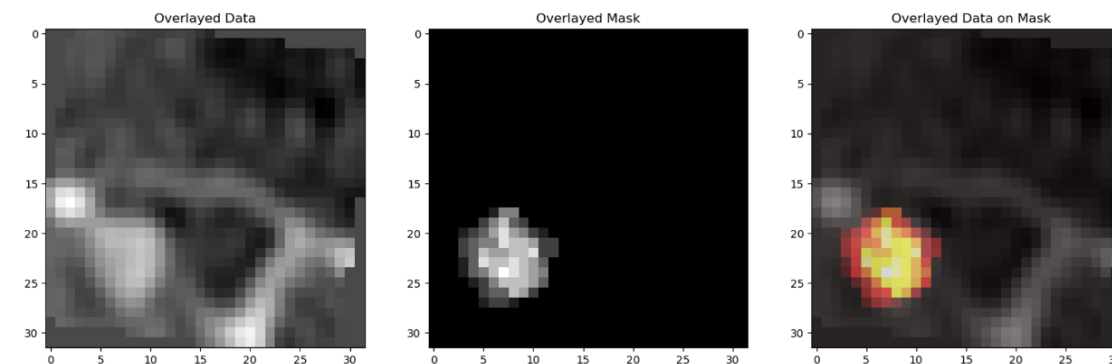


Fig 3. HHHHH\_CA\_000034 (742)

## ■ Aneurysm Visualization Module

### ■ Viewpoint adjustment 추가 진행사항

#### ✓ 해결방안

- (방법 2) 3D mask의 centroid를 지나면 혈관의 주행방향과 수직인 벡터와 평행하게 압축
  - 주어진 평면에 평행한 슬라이스를 추출하고 하나로 결합

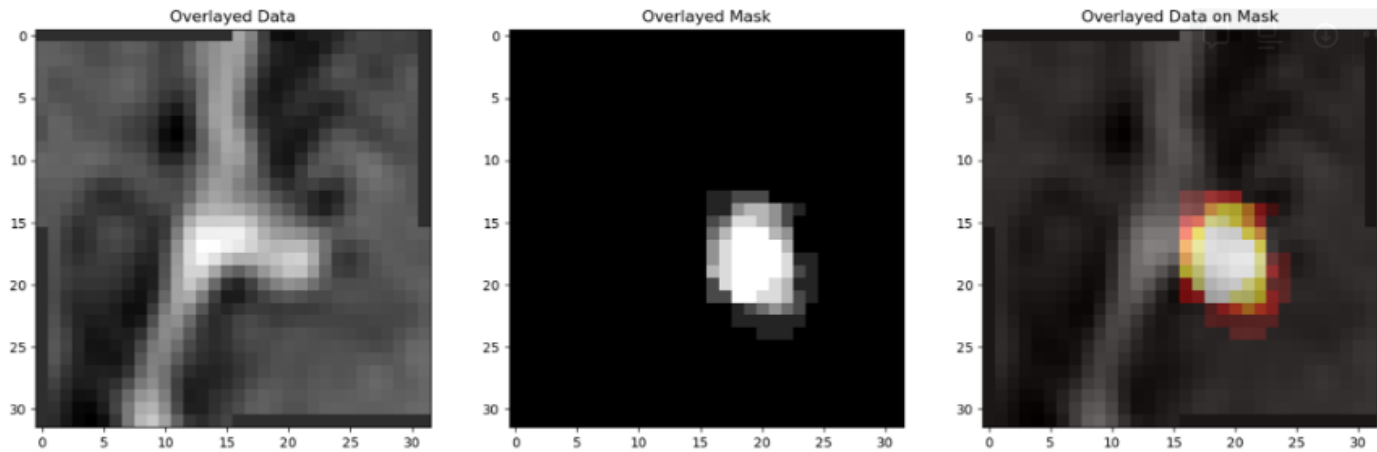
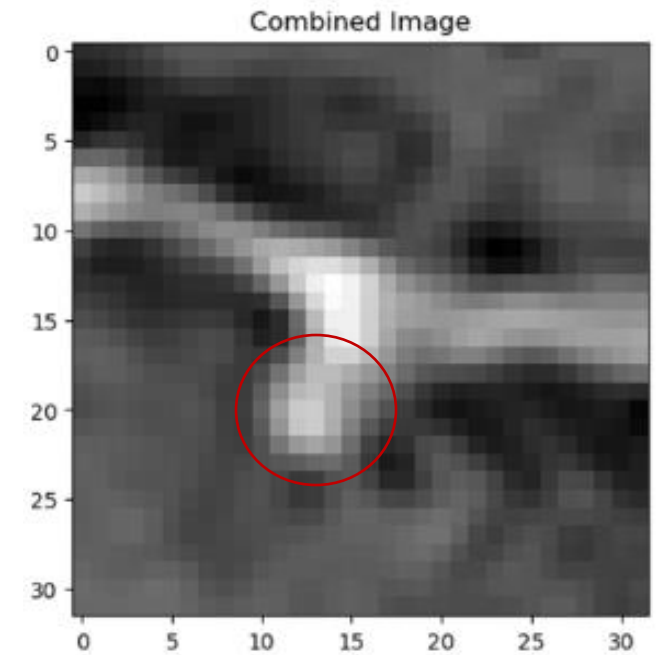


Fig 1. HHHHH\_CA\_000062 (252)





## ■ Aneurysm Visualization Module

### ■ Viewpoint adjustment 추가 진행사항

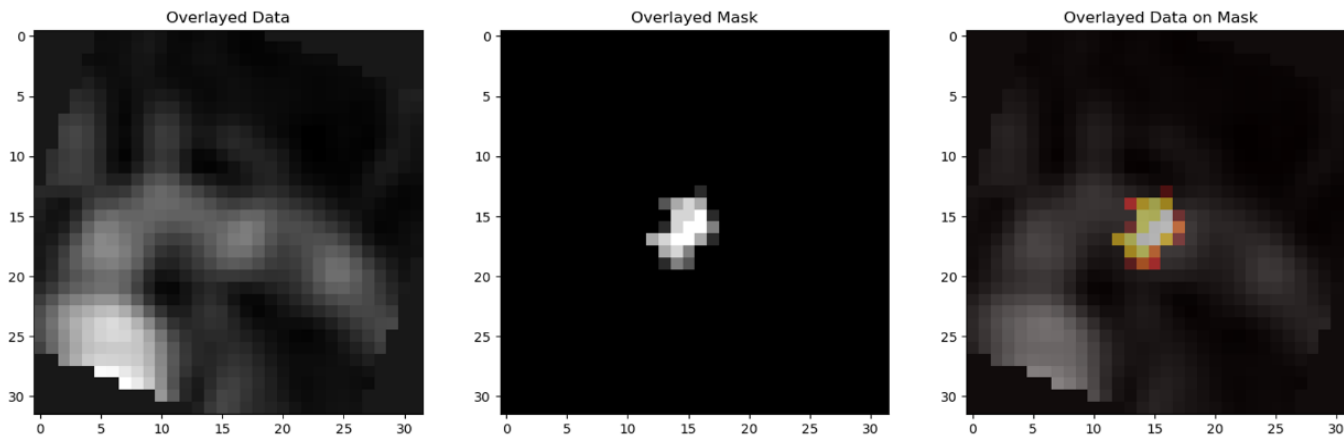


Fig 1. HHHHH\_CA\_000022 (199)

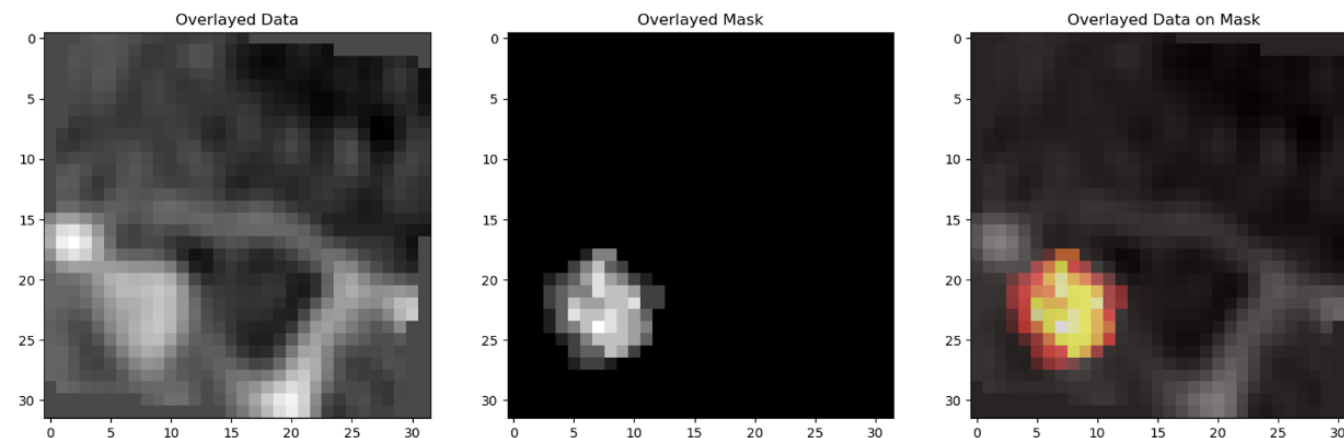
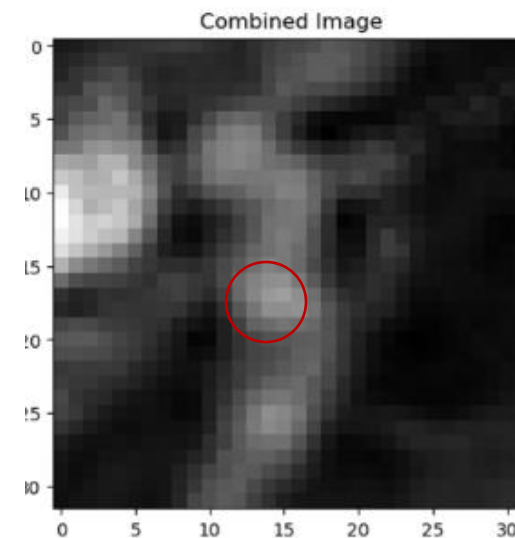
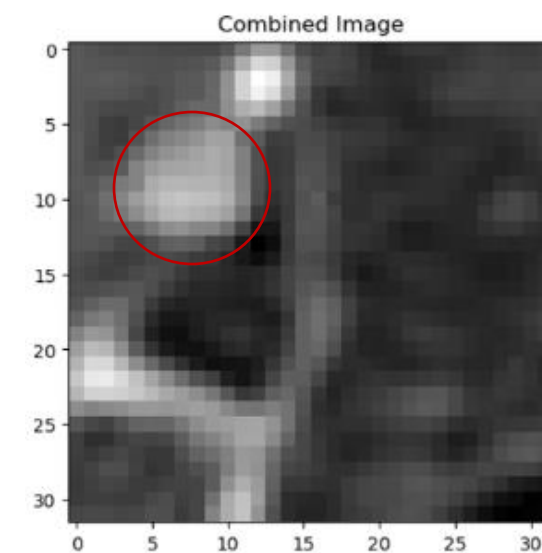


Fig 2. HHHHH\_CA\_000034 (742)



## ■ Aneurysm Visualization Module

### ■ Code Modularization

- ✓ .ipynb 파일을 .py 파일 형태로 변환 (클래스화)
- ✓ 코드 간결화
  - 불필요한 부분, 중복 제거

```

SliceVisualizer
└ show_slices

OverlayVisualizer
└ overlay_slices
└ visualize_overlay

OptimizedVesselView
└ rotate_data

AneurysmCentroid
└ calculate_centroid
└ show_slices_with_centroid

Rotation
└ calculate_rotation
└ _calculate_plane_normal
└ _calculate_rotation_angle
└ _apply_rotation
  
```

Fig 1. Aneurysm\_Visualization.py Structure

```

Aneurysm_Visualization.py
workspace > smjung > Aneurysm_Visualization.py > ...
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.spatial.transform import Rotation as R
4 from scipy.ndimage import affine_transform
5 from scipy.ndimage import rotate
6 from numpy.linalg import norm
7
8
9 class SliceVisualizer:
10     def __init__(self, data, mask):
11         self.data = data
12         self.mask = mask
13
14     def show_slices(self, with_overlay=False):
15         num_slices = self.data.shape[0]
16         cols = 8
17         rows = num_slices // cols + int(num_slices % cols != 0)
18
19         fig, axes = plt.subplots(rows, cols, figsize=(2 * cols, 2 * rows))
20         axes = axes.flatten()
21
22         for i in range(num_slices):
23             axes[i].imshow(self.data[i, :, :], cmap='gray', interpolation='none')
24             if with_overlay:
25                 axes[i].imshow(self.mask[i, :, :], cmap='hot', alpha=0.5, interpolation='none')
26             axes[i].set_title(f'Slice {i}')
27             axes[i].axis('off')
28
29         for i in range(num_slices, len(axes)):
30             axes[i].axis('off')
31
32         plt.tight_layout()
33         plt.show()
34
35 class OverlayVisualizer:
36     def __init__(self, data, mask):
37         self.data = data
38         self.mask = mask
39
40     def overlay_slices(self, process_all_slices=True, average_slices=False):
41         if process_all_slices:
42             overlaid_data = np.sum(self.data, axis=0)
43             overlaid_mask = np.sum(self.mask, axis=0)
44         else:
45             valid_slices = self.mask.any(axis=(1, 2))
46             overlaid_data = np.sum(self.data[valid_slices], axis=0)
47             overlaid_mask = np.sum(self.mask[valid_slices], axis=0)
48
49         if average_slices:
50             num_slices = len(self.data) if process_all_slices else valid_slices.sum()
51             overlaid_data /= num_slices
52             overlaid_mask /= num_slices
53
54         overlaid_data_on_mask = np.where(overlaid_mask > 0, overlaid_data, 0)
55
56         return overlaid_data_on_mask, overlaid_mask
  
```

Fig 2. Aneurysm\_Visualization.py

## ■ Aneurysm Visualization Module

### ■ Code Modularization

#### ✓ 개선점

- 각 클래스에서 공통으로 사용되는 메서드의 경우 클래스로 만들어놓고 이를 상속받아서 각 기능 클래스로 구현

| Class 이름            | Function 이름               | 역할  |
|---------------------|---------------------------|---|
| SliceVisualizer     | show_slices               | <ul style="list-style-type: none"> <li>- 모든 슬라이스 시각화</li> <li>- with_overlay=True : mask를 data위에 overlay하여 시각화</li> </ul>   |
| OverlayVisualizer   | overlay_slices            | <ul style="list-style-type: none"> <li>- 슬라이스를 합친 결과를 시각화</li> <li>- process_all_slices=True : 모든 슬라이스를 sum한 것을 시각화 (all/mask=1)</li> <li>- average_slices=True : 슬라이스를 합치고 평균을 낸 것을 시각화 (sum/average)</li> </ul> |
|                     | visualize_overlay         | <ul style="list-style-type: none"> <li>- overlay된 data 및 mask 시각화</li> <li>- overlay된 data, mask, data위의 mask를 함께 표시하는 이미지</li> </ul>   |
| OptimizedVesselView | rotate_data               | <ul style="list-style-type: none"> <li>- data, mask, direction을 z축을 기준으로 주어진 각도(90도)만큼 회전</li> </ul>  |
| AneurysmCentroid    | calculate_centroid        | <ul style="list-style-type: none"> <li>- rotate된 data와 mask를 사용하여 aneurysm의 centroid 계산</li> <li>- mask에서 1인 좌표들의 평균을 계산</li> </ul>   |
|                     | show_slices_with_centroid | <ul style="list-style-type: none"> <li>- 계산된 centroid를 슬라이스에 표시하여 시각화</li> </ul>  |
| Rotation            | calculate_rotation        | <ul style="list-style-type: none"> <li>- data를 회전시키기 위한 평면의 법선과 각도를 계산</li> </ul>   |
|                     | calculate_plane_normal    | <ul style="list-style-type: none"> <li>- 평면의 법선벡터 계산</li> </ul>   |
|                     | _calculate_rotation_angle | <ul style="list-style-type: none"> <li>- 회전해야 할 각도를 계산</li> </ul>   |
|                     | _apply_rotation           | <ul style="list-style-type: none"> <li>- 계산된 각도로 data와 mask를 회전</li> </ul>  |

Fig 1. Aneurysm\_Visualization.py

## ■ Aneurysm Visualization Module

### ■ Code Modularization : 추가 진행사항

#### ✓ 개선점

- 각 클래스에서 공통으로 사용되는 메서드의 경우 클래스로 만들어놓고 이를 상속받아서 각 기능 클래스로 구현

|                   |                   |
|-------------------|-------------------|
| VisualizerBase    | Show_slices       |
| SliceVisualizer   |                   |
| OverlayVisualizer | overlay_slices    |
|                   | visualize_overlay |

|                |                           |   |
|----------------|---------------------------|---|
| SliceExtractor | calculate_plane_normal    | - 평면의 법선벡터 계산                           |
|                | extract_plane_slice       | - 3D data에서 구한 평면에 해당하는 2D 슬라이스 추출      |
|                | create_slices             | - 여러 2D 슬라이스를 생성                        |
|                | combine_slices            | - 여러 2D 슬라이스를 결합                        |
|                | create_and_combine_slices | - 평면과 평행한 2D 슬라이스들을 추출하여 하나의 2D 이미지로 결합 |

| Class 이름            | Function 이름               | 역할  |
|---------------------|---------------------------|---|
| SliceVisualizer     | show_slices               | - 모든 슬라이스 시각화<br>- with_overlay=True : mask를 data위에 overlay하여 시각화   |
|                     | overlay_slices            | - 슬라이스를 합친 결과를 시각화<br>- process_all_slices=True : 모든 슬라이스를 sum한 것을 시각화 (all/mask=1)<br>- average_slices=True : 슬라이스를 합치고 평균을 낸 것을 시각화 (sum/average) |
|                     | visualize_overlay         | - overlay된 data 및 mask 시각화<br>- overlay된 data, mask, data위의 mask를 함께 표시하는 이미지   |
| OptimizedVesselView | rotate_data               | - data, mask, direction을 z축을 기준으로 주어진 각도(90도)만큼 회전  |
| AneurysmCentroid    | calculate_centroid        | - rotate된 data와 mask를 사용하여 aneurysm의 centroid 계산<br>- mask에서 1인 좌표들의 평균을 계산   |
|                     | show_slices_with_centroid | - 계산된 centroid를 슬라이스에 표시하여 시각화  |
| Rotation            | calculate_rotation        | - data를 회전시키기 위한 평면의 법선과 각도를 계산   |
|                     | calculate_plane_normal    | - 평면의 법선벡터 계산   |
|                     | _calculate_rotation_angle | - 회전해야 할 각도를 계산   |
|                     | _apply_rotation           | - 계산된 각도로 data와 mask를 회전  |

Fig 1. Aneurysm\_Visualization.py



감사합니다