

제공되는 데이터 분할 과제

컴퓨터비전응용
송실대 컴퓨터학부
장준영

- 개발 환경 : python 언어 사용, visual_studio
- 코드 내용 설명

○ DICOM 이미지 로드와 gray_scale

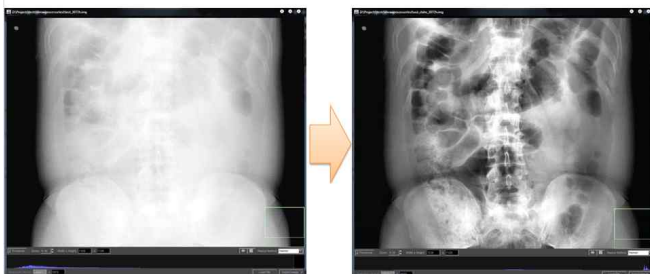
이진화를 하기 위한 gray_scale이 적용된 이미지로 만들기 위해
명암대비 스트레칭을 한다.

$$P(x, y) = \frac{P(x, y) - \min}{\max - \min} \times 255$$

○ 일부 이미지에 대한 히스토그램 평활화

응용 사례 : CLAHE 적용 결과

- X-ray Image



○ 이진화 임계값 설정

Otsu Thresholding : $\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$ 을 응용한

새로운 분할 지점 계산식 도입 (보정값)

`ret_final = 0.1 * ret_avg + 0.9 * ret_now`

○ 열기 연산

kernel(11 x 11) 에 대해 침식, 팽창 순서대로 1번씩 진행

○ 윤곽선 검출 -> Suzuki & Abe의 알고리즘

○ 마스크 영역 추출 & 원본 이미지에 마스크 영역으로 overlay

(1) 열기 연산까지 마친 binary_img에 대해서

행과 열이 같은 mask_img 생성 (값들은 0으로 초기화)

(2) 윤곽선을 이루는 좌표의 배열인 contours의 모든 것을

(255)값으로 채워 넣는다. 그리고 내부도 모두 꽉 채운다.

(3) 알아보기 쉽게 만들어진 mask_img에 대해

비트 반전 시킨 값을 mask에 저장한다.

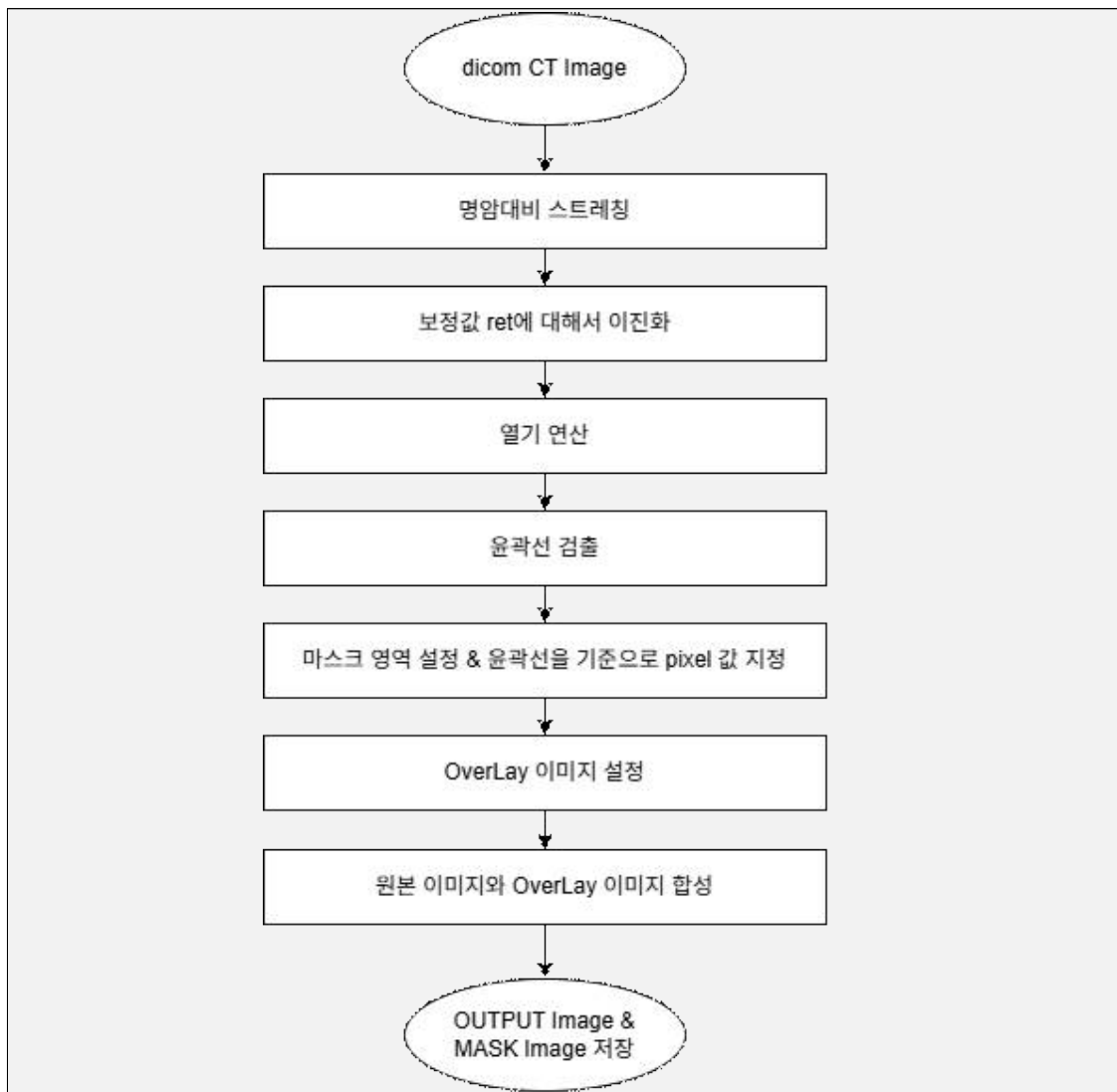
(4) 그리고 mask의 numpy 배열값이 0인 것들에 대해 빨간색으로 칠하고,

이를 overlay에 저장한다.

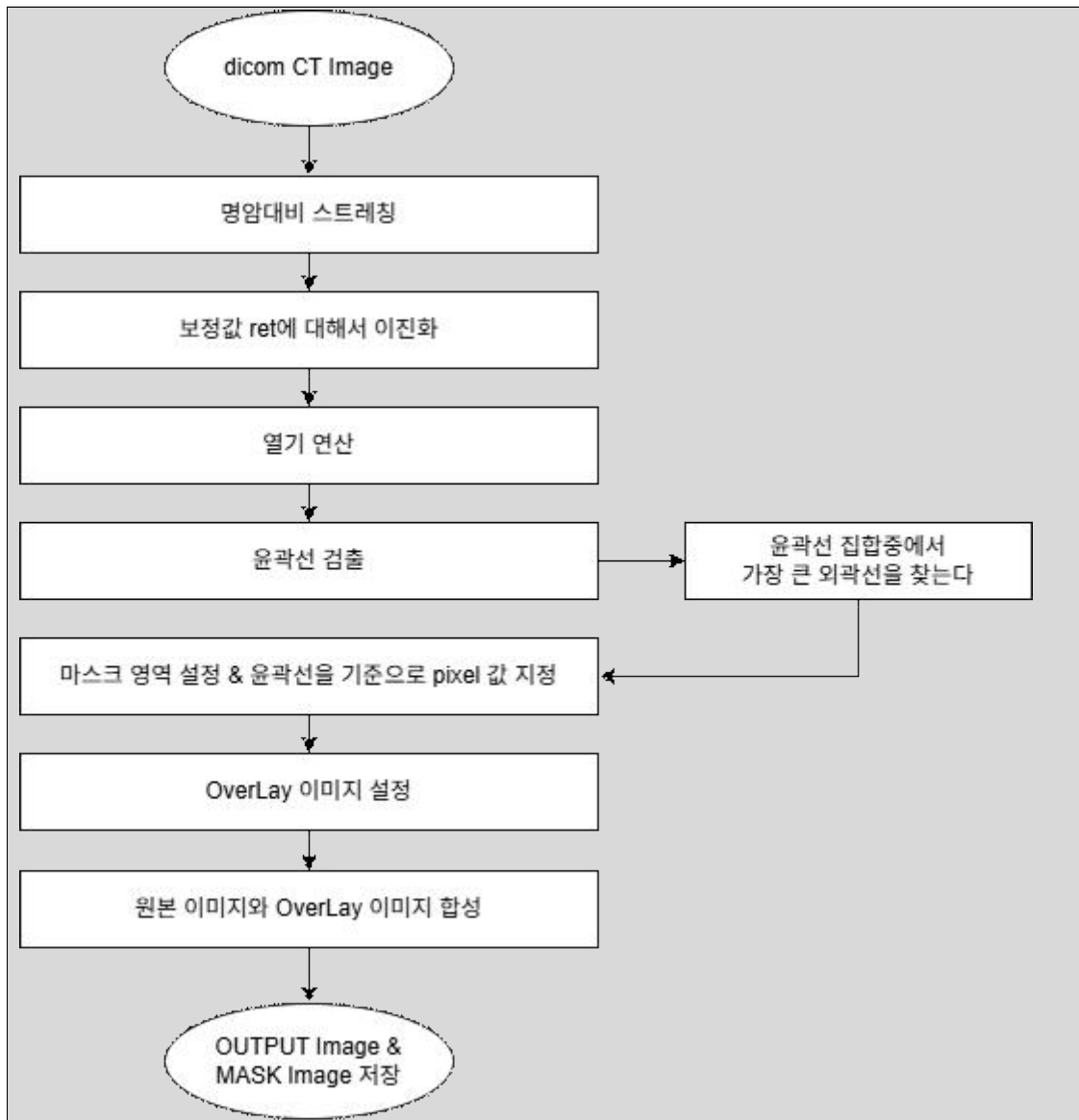
(5) 원본 이미지와 overlay를 합산한다.

- 참고 자료 : 교재(명암대비 스트레칭, CLANE 알고리즘)

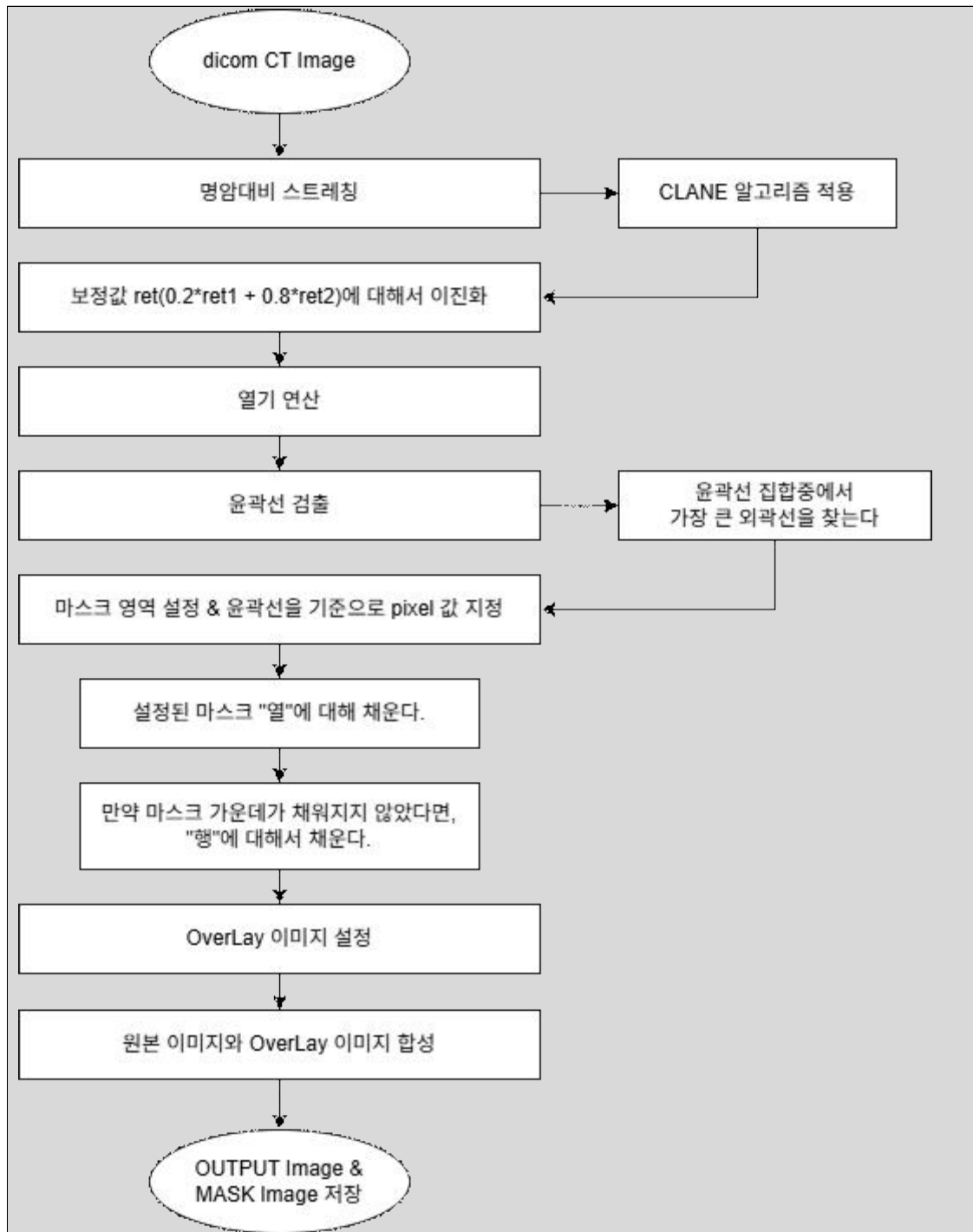
◇ Case 1



◇ Case 2, Case 3



◇ Case 4



특이한 case의 경우 (검출할 객체와 비슷한 밝기의 광원)
special_case.py를 실행시켜 시각적으로 광원을 block 처리해 mask를
추출한다. 100% 자동화할 수 없었다. 이러한 case 분포는 12/559, 약 2%다.

ex) Case 4/png/Case4_373.png

