**막장 불연속면 인식 및 후처리**

**2024.12.00**

**한국철도기술연구원**

**첨단물류시스템연구실**

**박준영**

본 보고서에 관한 모든 권한은 한국철도기술연구원의

"첨단물류시스템연구실"에 귀속된다.

**목 차**

1. **서론 0**

**1.1 연구 배경 0**

**1.2 선행 연구 0**

**1.2.1 영상처리 기반 불연속면 구분 0**

**1.2.1 Segment 기반 불연속면 구분 0**

**1.2.1 3D 포인트 클라우드 기반 불연속면 구분 0**

**1.2.4 SAM-Adapter 0**

1. **본론 0**

**2.1 막장 불연속면 인식 0**

**2.2 이미지 보정 0**

**3. 성능검증 0**

**4. 결론**

**5. 참조문헌**

**1. 서 론**

**1.1 연구 배경**

최근 교통 인프라의 확장과 철도 교통 수요의 증가로 인해 터널시공이 증가하고 있다. 터널 시공에서 터널의 굴착을 위해 굴착면을 천공하고, 천공 홀에 할암봉을 삽입한 뒤, 할암봉의 팽창을 유도해 암반을 파쇄하는 방법이 사용되고 있다. 그러나 이러한 방식에서 다수의 할암봉을 삽입함에 따라 작업자의 안전 문제와 작업 피로도가 증가하는 문제가 있다. 이에 굴착면을 천공하고 할암봉을 삽입하는 과정을 자동화할 필요가 있다.

할암, 천공 시스템의 자동화를 위해서는 굴착면에 천공 홀을 형성할 때, 최적의 천공 위치를 자동으로 탐색하는 기술이 필수적이다. 이를 위해 먼저 굴착면에 나타나는 불연속면을 정확하게 인식해야 한다. 불연속면을 인식함으로써 굴착면의 구조적 취약점을 사전에 파악하여 굴착 과정에서 발생할 수 있는 붕괴, 균열, 변형과 같은 안전 사고를 방지하고 최적의 굴착 계획 수립할 수 있다. 또한 불연속면들을 피해서 불연속면들 사이의 공간에 위치한 굴착면에 천공이 이루어져 안전하고 효율적인 굴착을 할 수 있다.

본 연구에서는 딥러닝 기술을 통해 굴착면에 형성된 불연속면을 자동으로 인식하는 시스템을 제안한다. 구체적으로, 불연속면 예측 모델을 이용하여 터널의 굴착면을 촬영한 영상에서 세그먼트를 분리하고 이를 불연속면으로서 결정한다. 이를 위해, SAM-Adapter을 이용하여 불연속면을 예측한 뒤 이미지를 보정하는 과정을 통해 막장 불연속면의 인식 정확도를 높이는데 중점을 둔다.

본 연구의 배경으로는 다음과 같은 필요성이 존재한다. 첫째, 제안된 방법을 통해 인식된 불연속면을 바탕으로 굴착면 내 최적의 천공위치를 선정함으로써 천공, 할암 시스템 자동화에 기여할 수 있다. 둘째, 작업자의 경험적 판단이 아닌 과학적 분석을 통해 불연속면을 구분함으로써 더 일관성 있는 작업을 가능하게 한다. 셋째, 이미지에 포함된 왜곡과 노이즈는 불연속면의 위치와 모양을 왜곡시키고 배경과 불연속면 간의 경계의 정확한 인식을 방해하므로 이미지 데이터의 보정이 필요하다. 이러한 필요성에 따라, 본 연구에서는 SAM-Adapter를 기반으로 불연속면을 인식하고 PCA 기반 후처리를 통해 이미지를 보정하였다.

해당 연구에서는 먼저 SAM-Adapter에서 추출된 이미지의 연결 성분이 끊어진 문제를 해결하기 위한 과정을 상세히 설명하고, 이러한 문제를 해결하기 위해 적용된 PCA 기반의 접근 방식을 제시한다. 마지막으로, 본 연구의 결과를 바탕으로 끊어진 연결 성분 문제를 해결하기 위한 방법론의 효용성과 한계점을 논의하며, 향후 연구 방향에 대해 논의한다.

**1.2 선행 연구**

**1.2.1 영상처리 기반 불연속면 구분**

기존에는 막장 불연속면 구분을 위해 지질 조사를 수행하고 기술자들이 직접 불연속면을 구분하였다. 하지만 불연속면 구분 과정에서 개인의 경험과 해석에 따라 결과가 달라질 수 있으며 이는 신뢰성과 객관성을 저하시키게 된다. 또한 지질 조사에 많은 시간과 노력이 소요되는 문제점이 있다. 이에 따라 영상처리 기술을 이용해 불연속면을 구분하는 연구가 이루어졌다. 디지털 카메라를 이용해 터널 굴착면 이미지를 수집하고 이를 영상처리 기술로 분석함으로써 기존의 수작업 방식을 보완하고 효율성을 높이는 접근 방식이 제안되었다. 하지만 이러한 연구들에서 막장 불연속면 구분에 사용된 영상처리 기법은 엣지 검출기를 기반으로 한 방법이므로 이로 인한 한계점이 존재한다. 엣지 검출기는 이미지에서 픽셀값의 변화율을 이용해 경계를 탐지하는 방식으로 작동한다. 하지만 막장면처럼 오랜 시간 여러 암석이 뒤섞인 환경에서는 픽셀값의 변화가 명확하지 않고 불규칙하게 나타나는 경우가 많아 엣지 검출기로 불연속면을 구분하는데 한계가 있다.

**1.2.2 Segment 기반 불연속면 구분**

기존의 영상처리를 이용한 불연속면 구분의 한계점을 극복하기 위해 딥러닝을 통한 막장 불연속면 구분 연구가 수행되었다. 그 중 인공 신경망을 활용한 기술의 발달로 막장 이미지를 분석하고 불연속면의 추출 및 평가를 자동화하는 접근법이 제안되었다. 이러한 연구들은 기존의 전통적인 엣지 검출기법에 비해 픽셀 수준이 아닌 다층적 특성을 분석함으로써 불연속면을 Segment할 수 있다는 장점이 있다. 이는 이미지 내 구조적 특징과 의미론적 관계를 파악하여 불연속면의 위치와 형상을 분리하고 분석할 수 있음을 의미한다. 그러나 기존의 딥러닝 기반 연구들은 막장 이미지를 여러 구역으로 분할한 뒤 각 구역 내의 모든 균열이나 절리선을 감지 한다. 이러한 접근법은 구역 간의 연속성을 놓치는 경우가 생겨 정확한 경계 추출이 어렵고, 모든 균열과 절리선 추출 시 실제 굴착에 필요하지 않은 정보까지 포함되어, 발파 위치를 결정할 때 혼란을 준다는 한계가 있다.

**1.2.3 3D 포인트 클라우드 기반 불연속면 구분**

이와 같은 한계점을 극복하기 위해 최근에는 LiDAR(Light Detection and Ranging)및 사진측량 기법을 활용하여 3D 이미지를 생성하고 이를 바탕으로 3D 포인트 클라우드를 이용한 불연속면 인식 연구가 진행되었다. 이러한 접근법은 불연속면의 정확한 위치와 방향을 파악할 수 있으며 기존 2D이미지 분석에 비해 3차원 좌표 반사도 등 더 많은 정보를 제공한다는 장점이 있다. 그러나 3D 이미지를 생성하기 위한 이미지 수집 단계에서 많은 시간이 소요되고 이후 3D 포인트 클라우드를 처리하고 분석하는 과정에서 대용량 데이터를 다루어야 한다는 한계가 있다. 포인트 클라우드는 각 점 마다 3차원 좌표와 추가적인 속성을 포함하기 때문에 데이터 크기가 방대하며 이를 효과적으로 처리하기 위해서는 고성능 계산 자원과 메모리가 요구된다. 이러한 복잡한 연산 과정은 처리 속도를 저하시켜 실시간으로 막장면을 분석하기에는 적합하지 않다.

**1.2.4 SAM-Adapter**

이러한 문제를 해결하기 위해 본 연구에서는 이미지 분할 모델인 SAM-Adapter을 사용해 막장 불연속면을 인식하고 후처리를 통해 막장 불연속면을 구분하는 기술을 제안한다. 모델의 선정 근거와 자세한 설명은 다음과 같다. SAM-Adapter 모델은 종래에 알려진 SAM(Segment Anything Mode)을 기반으로 개량된 학습 모델로서, 기존의 SAM 모델에 어댑터가 추가된 구조를 갖고 있으며 상세 구성은 그림 1과 같다. 기존의 SAM을 세그먼테이션 식별을 위한 백본으로서 사용하는데, 복수의 이미지 인코더들 사이에 어댑터 모듈이 추가된 구조를 갖는다. SAM의 마스크 디코더의 경우, 사전 훈련된 SAM의 가중치로 초기화되며, 어댑터와 같이 훈련된다.

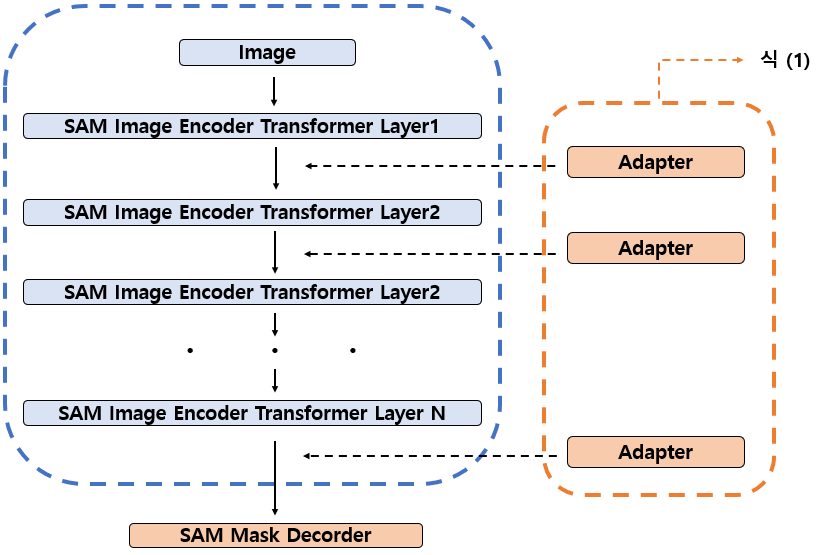


그림 1 SAM-Adapter 구조

이미지 데이터가 입력되면 기존 SAM 모델의 트랜스포머 레이어에서 이미지의 기본적인 특징이 추출된다. 이 단계에서는 SAM모델이 학습한 범용적인 이미지 특징들이 처리된다. Adapter는 태스크 특정 지식(task-specific knowledge)을 추가 학습한 이후 각 트랜스포머 레이어에 위치한 Adapter가 태스크 특정 지식을 트랜스포머 레이어에 주입한다. 이 프롬프트는 트랜스포머 레이어가 기존에 처리한 범용적인 특징과 결합되어 추가 학습된 작업에 맞춰 정교한 세그먼테이션은 가능하게 한다. 또한, 각 Adapter 모듈의 상세 구조를 살펴보면, 태스크 특정 지식 Fi를 인코더에 추가하기 위하여, Adapter 모듈을 사용한다. 각 어댑터는 수식 1과 같이 Fi를 입력으로 수신하고, 프롬프트 Pi를 출력한다.

EMB00000fa80136 수식 1

이때, EMB00000fa80137은 태스크 정보를 학습하는 역할을 수행하는 것으로, 각 어댑터마다 태스크 특정 프롬프트를 생성하게 하는 리니어 레이어이고, EMB00000fa8014d은 트랜스포머 레이어와 차원을 맞게 하도록, 업샘플링을 수행하는 리니어 레이어이다.

SAM-Adpeter 모델은 SAM과 비교하여 객체와 배경을 보다 정확하게 분리할 수 있는 것으로 알려져 있다. 따라서, 본 기술에 적용하면, 굴곡으로 인한 그림자 및 불연속면을 구분하고, 미세한 불연속면 추출할 수 있게된다. 이와 같은 SAM-Adapter 모델은 앞서 소개한 논문을 통해 공개된 것으로, 상세한 구성은 해당 논문을 참조하기로 한다. 특히, 태스크 특정 지식(task-specific knowledge) Fi 에 불연속면 추출에 대한 정보를 제공하면, 태스크에 적합한 미세조정이 가능하게 된다. 실제 실험에서는 370장의 이미지를 9:1 비율로 나누어 학습 이미지 338장, 검증 이미지 32장을 이용하였다.

**2. 본 론**

**2.1 불연속면 인식**

**2.1.1 데이터셋 구축**

학습 데이터셋 구축을 위해 한국도로공사의 페이스 맵핑(face mapping) 자료를 활용하였으며, 페이스 맵핑은 막장의 지질 구조, 암반 상태, 불연속면, 지하수 유무를 조사하여 기록한 자료이다. 두 번째로 한국철도시설공단의 터널 굴착 검측서 자료를 활용하였으며, 터널 굴착 과정에서 수행된 굴착 단면 검사, 지보재 검사, 사용된 굴착 공법과 진행 상태등을 기록한 문서이다. 본 연구에서는 페이스 맵핑과 터널 굴착 검측서에서 막장 이미지만을 추출하여 사용하였다. 이후 추출한 이미지에 불연속면을 레이블링하는 과정을 거친다. 그림 2는 레이블링 과정의 예시로 수집한 막장 이미지에 불연속면을 나타내는 경계를 레이블링한 결과를 보여준다. 각 원들은 불연속면을 포함한 구역을 나타낸다.

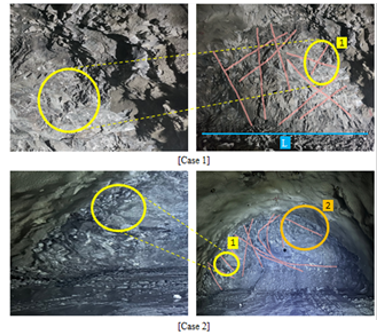


그림 2 레이블링 예시

막장 이미지에서 불연속면에 해당하는 부분이 많아 이를 모두 레이블링할 경우, 과도한 레이블링으로 인해 분석이 복잡해지고 학습 및 훈련 시간이 증가하여 모델이 불연속면을 예측하는데 걸리는 시간이 증가하는 문제가 발생할 수 있다. 이에 길이에 따른 우선순위를 부여하여 레이블링 진행하였으며 길이의 기준은 다음과 같다. 이미지의 밑면 길이를 L이라고 할 때 최소 4/L의 길이를 가지는 불연속면을 레이블링 하여 이미지당 레이블링 개수를 15개 이내로 제한하였다.

**2.1.1 인식 결과**

본 연구에서 SAM-Adapter는 레이블링한 이미지를 바탕으로 불연속면을 예측한다. SAM-Adpter의 학습에는 총 370장의 이미지가 사용되었으며 이 중 338장은 학습용, 32장은 검증용으로 9:1의 비율로 나누어 활용하였다. 그림 3은 불연속면을 인식한 예시를 나타내며 이를 통해 불연속면을 인식함을 확인할 수 있었다.

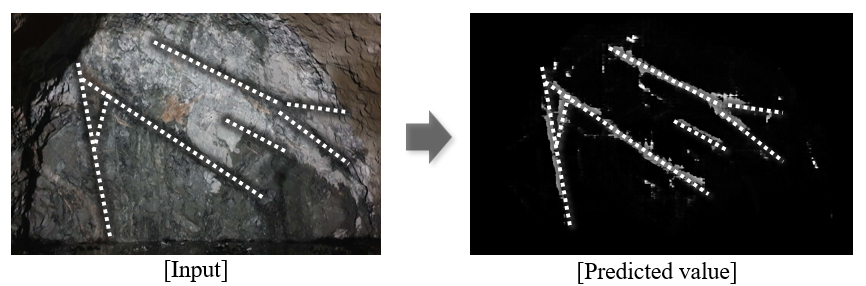


그림 3 불연속면 인식 예시

**2.2 이미지 보정**

**2.2.1 이미지 보정 알고리즘**

불연속면 이미지는 불연속면의 특성상 여러 객체들이 서로 연결된 것이나 암석 파편이나 흙더미 등의 이유로 분리된 것으로 촬영되었거나, 촬영 시 조명으로 인해 생기는 그림자 등 여러 가지 상황에 의해 불연속면이 분리된 상태로 보일 수 있다. 후처리 단계에서는 예측한 불연속면 간의 위치와 객체의 주요 방향을 나타낸 각도성분을 바탕으로, 분리된 상태로 예측된 불연속면을 하나의 연속적인 불연속면으로 복원한다. 이를 통해 데이터 수집 과정에서 생기는 불완전한 이미지 정보로 인해 불연속면이 끊어져 있거나 일부가 누락된 상태로 검출된 경우에도 후처리 단계를 통해 구조가 복원되고 누락된 불연속면을 나타나게 한다. 후처리 단계로는 그림 6과 같으며 이미지 보정 과정의 전체 코드의 경로는 C:\Users\User\imgetest\post-processing.ipynb 파일에 저장되어 있다.

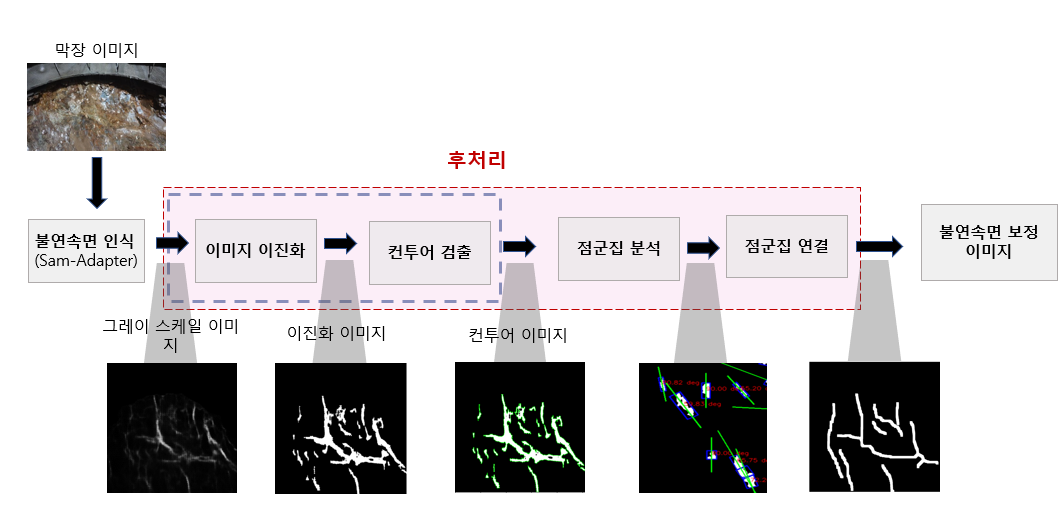
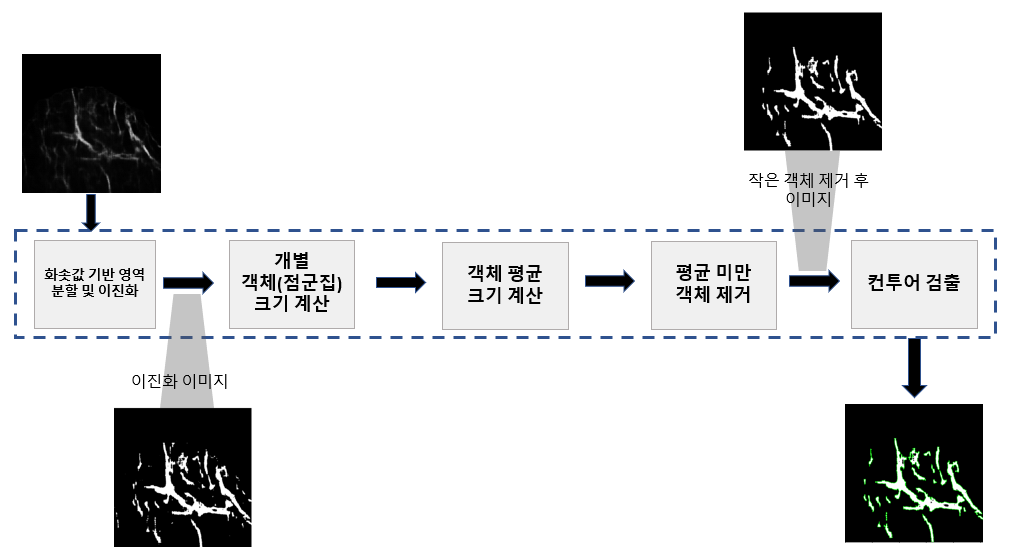


그림 4 불연속면 자동 인식 알고리즘

후처리 단계는 다음과 같다. 첫째, 이미지 이진화 및 컨투어 검출 단계이다. 불연속면 예측 모델에서 출력된 그레이 스케일 이미지를 이진화 처리한 이후 컨투어를 검출하는 단계로, 컨투어 검출은 이미지에서 각 객체의 경계선을 추출하여 객체의 위치와 형태를 파악하는데 목적이 있다. 둘째, 컨투어 이미지로부터 점군집을 분석하여 연결 가능한 점군집을 찾는 단계이다. 점군집은 상하좌우 및 대각선을 포함한 8방향으로 연결되어 있는 픽셀들의 집합을 뜻하며 점군집의 특징을 추출하기 위해 주성분분석을 수행한 후 점군집 간의 유사도를 계산하여 간선을 생성한다. 셋째, 분석한 점군집을 바탕으로 점군집을 연결하는 단계이다. 간선 생성 이미지와 슈퍼픽셀화된 이미지를 이용하여 점군집을 연결하고 선의 두께를 조정하는 과정과 잔가지를 제거하는 과정을 포함한다. 슈퍼 픽셀화란 이미지 픽셀들에 대해 유사한 픽셀 값을 가지는 픽셀들을 묶어 하나의 픽셀로 만드는 과정이다. 각 단계의 자세한 설명은 2.2.2 ~ 2.2.4 절을 참고한다.

**2.2.2 이미지 이진화 및 컨투어 검출**

그림 5 이미지 이진화 및 컨투어 검출 과정을 도시화한 것이다. 컨투어 검출을 위해 이미지 이진화 이후 불연속면에 해당하는 객체만 남기기 위해 개별 객체를 구성하는 점군집의 크기가 평균 미만일 경우 제거 한 후 컨투어 검출을 수행한다.

 그림 5 후처리 상세 과정1

SAM-Adapter가 불연속면 예측을 수행할 때, 각각의 픽셀이 불연속면에 속하는지 여부를 개별적으로 판단하기 때문에 불연속면이 명확한 경우에는 밝게, 불명확한 경우에는 어둡게 나타난다. 어둡게 나타난 부분 중에서 불연속면에 해당하는 픽셀이 포함되어 있을 가능성이 있기 때문에, 이 정보를 손실하지 않기 위해 각 픽셀을 클러스터링 하여 밝은 영역, 중간 영역, 어두운 영역으로 분할한다. 밝은 영역은 불연속면에 해당하는 픽셀, 어두운 영역은 배경 픽셀, 중간 영역은 불연속면과 배경 사이에 위치한 불확실한 픽셀들로 해당 단계에서는 불연속면에 해당하는 픽셀로 가정하고 후속 처리 단계에서 불연속면인지 결정한다. 예시 이미지에서 각 영역의 중간값은 어두운 영역 4, 중간영역 40, 밝은영역 125로 나타났다. 그림 6은 클러스터링 과정과 이진화 과정의 코드이다.

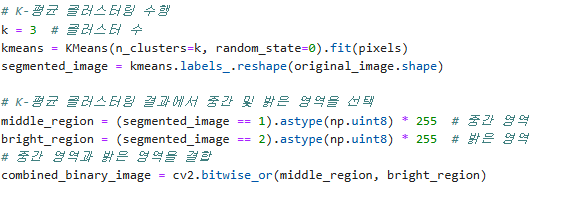


그림 6 클러스터링, 이진화 코드

클러스터링을 진행 할 때 sklearm.cluster에서 제공하는 K-평균 클러스터링을 사용하였으며 주요 파라미터로는 n\_clusters, random\_state이 있다. n\_clusters는 클러스터의 개수를 지정한다. random\_state는 초기 클러스터 중심을 고정하기 위한 파라미터로 동일한 데이터와 설정을 사용하더라도 실행할 때 마다 다른 결과가 나오는 것을 방지한다. K-평균 클러스터링은 초기 중심값을 무작위로 설정하기 때문에 random\_state 따라 결과가 달라질 수 있다. 본 연구에서는 이를 방지하고자 random\_stat값을 0으로 사용하였다. 0은 특별한 의미를 가지지 않지만 관례적으로 많이 사용되는 0을 선택하였다. 이후 중간 영역과 밝은 부분의 화소값을 255로 변경시키고 OR연산으로 두 영역을 결합시켜 이미지 이진화를 진행하였다.

다만, 이러한 이진화 처리에 의해 불연속면에 대한 정보뿐만 아니라, 노이즈 정보가 강조되었을 수도 있어 이러한 노이즈를 제거할 필요가 있다. 또한, 이진화 처리 이후에도 불연속면으로 예측된 객체들이 서로 분할된 상태로 배치되어 있음을 확인할 수 있다. 이에 본 기술에서는 불연속면으로 예측된 객체들 중 불연속면을 나타내는 객체들은 유지하고, 그렇지 않은 객체들은 제거하고자 한다. 이를 위해 이진화 이미지에 표시된 개별 백색 객체들의 크기를 모두 산출하고 이상치를 제거한 평균 크기 미만의 객체는 노이즈로 판단하여 제거하는 처리를 수행한다. 이상치 판단은 데이터 값이 평균으로부터 얼마나 떨어져 있는지 나타내는 Z-스코어를 사용하였으며 평균크기 미만 객체 제거 코드의 예시는 그림 7과 같다.



그림 7 평균크기 미만 객체 제거 코드

remove\_small\_components 함수는 이진화된 이미지에서 작은 연결 성분을 제거하여 주요 성분만 남기는 역할을 한다. 객체의 평균 크기를 구하기 위해 cv2.connectedComponents를 사용하여 각 객체에 고유한 label을 할당한다. 이후 반복문과np.sum을 활용하여 각 성분을 구성하는 픽셀의 수를 계산하고 이를component\_size리스트에 저장한다. 이상치를 제거를 위해 Z-스코어를 사용하며 Z-스코어는 각 성분의 크기가 평균에서 얼마나 떨어져 있는지 표준편차 단위로 측정한다. 이 값을 기준으로 설정된 범위에 속하지 않는 성분을 이상치로 간주하고 제거하였다. 예시 이미지에서 객체 제거 전에 추출된 컨투어는 91개로 노이즈를 포함한 작은 객체들도 포함된 개수이다. 임계값을 1또는 2로 설정 시 추출된 컨투어의 개수가 20개 동일 하였고 3으로 설정 시 14개로 줄어들어 일부 정상적인 객체까지 제거되었다. 본 연구에서는 정상적인 객체는 유지하면서 노이즈만 효과적으로 제거하기 위해 1과 2 사이값인 1.5를 임계값으로 사용하였다.

이후 이상치를 제거한 객체의 평균 크기를 filtered\_mean\_size에 저장하여 이 값보다 크기가 큰 객체일 경우에만 남겨 노이즈를 제거하였다. 노이즈를 제거한 이미지에 cv2.findContour를 사용하여 컨투어를 검출하며 코드는 그림 8과 같다.



그림 8 컨투어 검출 코드

cv2.findContours 함수는 OpenCV에서 이미지의 컨투어를 검출하는 함수로 매개변수는 다음과 같다. cv2.RETR\_EXTERNAL는 외각 윤곽선만 검출하는 모드이다. 다른 모드로는 cv2.RETR\_TREE,

cv2.RETR\_LIST가 있으며 각각 모든 윤곽선의 계층 구조를 탐색, 계층 구조 없이 모든 윤곽선을 탐색하는 모드이다. 여기서 계층 구조란 도넛 모양과 같이 한 컨투어 안에 다른 내부 컨투어가 있을 경우 이를 부모-자식관계로 저장하는 것을 말한다. cv2.CHAIN\_APPROX\_SIMPLE 윤곽선을 근사화 하는 방식으로 직선의 경우 시작점과 끝점처럼 선을 구성하는 최소한의 점만을 저장한다.

**2.2.3 점군집 분석**

점군집 분석 단계에서는 이전 과정에서 검출한 컨투어를 이용하여 점군집을 분석하며 이를 통해 각 객체가 불연속면을 구성하는 요소인지 판단한다. 그림 9는 후처리 과정에서 컨투어를 기반으로 객체를 분류하고 불연속면 후보군을 식별하는 단계를 도시화한 것이다.

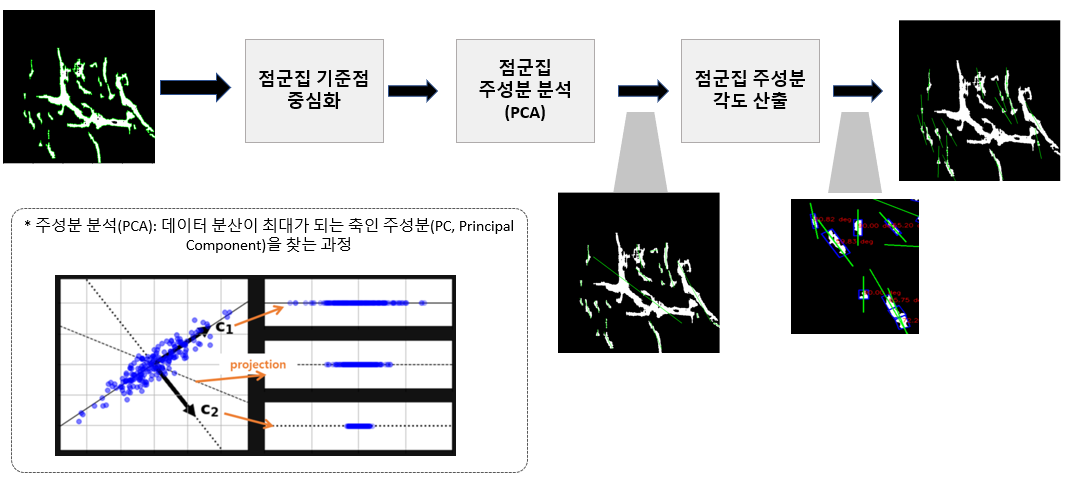


그림 9 후처리 상세 과정2

점군집의 연결을 위해 객체들은 본래 서로 연결된 것이나 분리된 상태로 간주한다. 연결이 이루어져야 할 후보군을 구성하는 각 픽셀들은 상하좌우 및 대각선을 포함한 8방향으로 연결되어 있다. 본 연구에서는 이러한 연결성을 기준으로 각 군집별 특징을 추출한다.

각 군집의 특징을 추출하기 위해 주성분 분석(PCA)을 수행하여 군집의 주요 축을 찾고 축의 각도를 계산하여 각 군집의 대표 특징으로서 각도 성분 추출한다. 주성분분석은 데이터 분산이 최대가 되는 축을 찾는 과정이다. 분산이 최대가 되는 축은 데이터의 변화량과 정보의 양을 가장 많이 포함하고 있어 데이터를 가장 잘 설명한다. 이를 각 군집에 적용시 각 군집의 주요 방향을 추출할 수 있다. 데이터 주성분분석의 수행 코드는 그림 10과 같다.

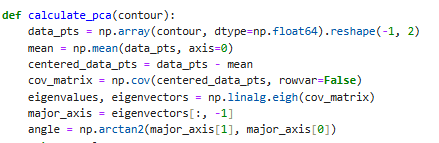


그림 10주성분분석 코드

데이터의 방향성을 반영하기 위해 각 군집의 중심 좌표값을 기준점으로 하여 주요 축을 찾는다. 이를 위해 변수 data\_pts에 컨투어의 경계선 좌표값들을 저장한다. 저장한 좌표값들을 np.mean을 사용해 컨투어 경계선 좌표값들의 중심점을 계산하고 각 경계선 좌표값에서 이 중심점(mean)을 빼주는 중심화 과정을 수행한다. 이후 중심화된 데이터를 기반으로 공분산 행렬을 계산한다. 공분산 행렬은 데이터의 분산과 변수 간 상관관계를 나타내며 이를 통해 데이터의 주요 분산 방향을 분석할 수 있다. 공분산 행렬은 np.cov를 사용하여 계산하고 np.linalg.eigh를 사용해 고유값과 고유벡터를 추출한다. 고유값은 데이터의 분산 크기를 나타내며, 고유벡터는 해당 분산 방향을 나타낸다. 마지막으로 주요 축의 방향을 각도로 나타내기 위해 고유벡터의 x,y성분을 사용하여 np.arctan2로 각도를 계산한다.

이와 같은 주성분 분석을 수행하면 각 군집을 구성하는 주요 방향을 확인할 수 있다. 하지만, 산출된 각 군집의 각도만을 이용하여 군집 연결 시 순차 연결이 되지 않아 중간 연결이 생략될 수 있고 각도는 유사하지만 방향성이 일치하지 않는 군집 간의 연결이 이루어진다.

이에 본 연구에서는 방향성이 일치하지 않는 연결을 방지하고자 각도와 거리를 기준으로 그룹을 나누고 군집의 노드화로 연결 순서를 결정하고 점군집 간의 유사도를 산출하여 군집을 연결하고자 한다. 그림 11은 이 과정을 도시화한 것이다.

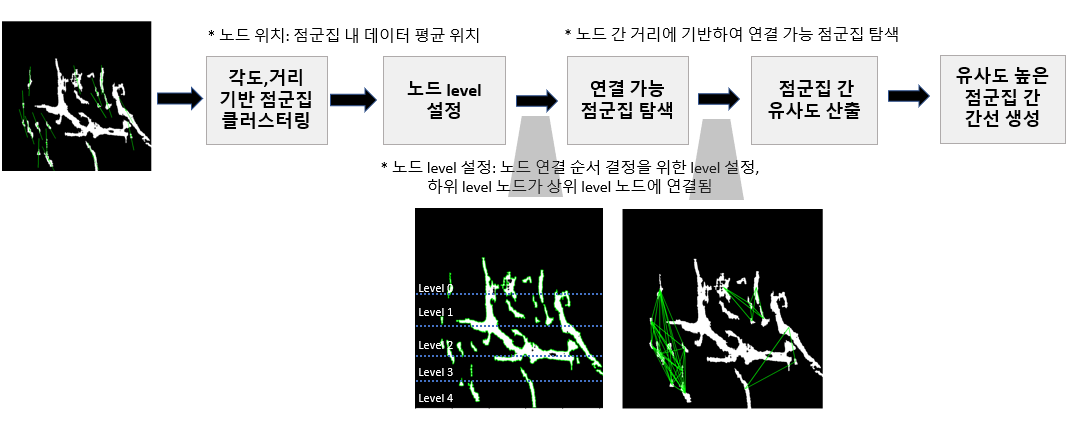


그림 11 후처리 상세 과정 3

각도는 유사하지만 방향성이 일치하지 않는 군집간의 연결을 방지하기 위해 각도성분을 기준으로 X축의 가까운 선분과 Y축에 가까운 선분으로 분류한다. 선분을 분류하기 위한 코드는 그림 12와 같다.

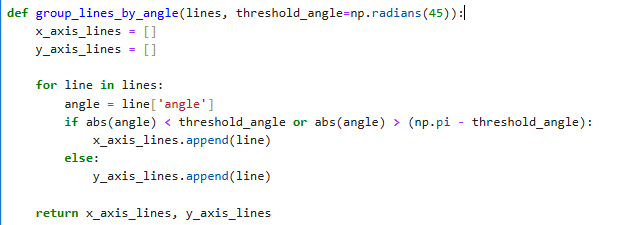


그림 12 각도 기반 선분 분류 코드

선분을 X축과 Y축에 가까운 선분으로 분류하기 위해 선분의 양 끝점 중 아래에 있는 점을 시작점으로 설정하고, 선분의 각도가 45°이하이거나 135°~180°사이에 해당하면 X축에 가까운 선분으로 간주하고 그렇지 않은 경우 Y축에 가까운 선분으로 간주하였다.

이후 멀리 떨어져 있는 군집의 잘못된 연결을 방지하기 위해 군집의 좌표값을 기준으로 그룹을 나누는 과정을 거친다. 그룹은 상단, 중단, 하단 또는 좌측, 중앙, 우측으로 분류하기 위해 이미지를 3등분하여 동일한 크기 3개의 그룹으로 나누었으며 코드는 그림 13과 같다.

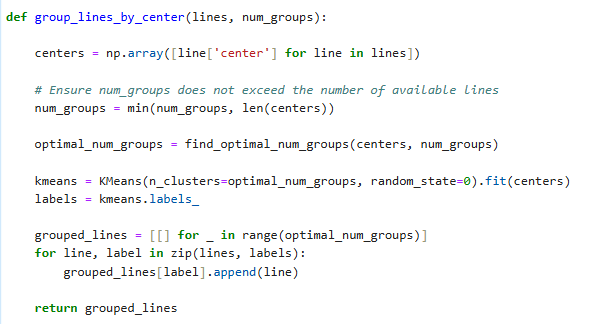


그림 13 좌표값 기반 선분 분류 코드

군집을 순차연결하기 위해 군집의 중심 좌표값을 노드로 설정하고 각 노드의 좌표값을 비교한다. 노드의 좌표값 차이가 미리 설정된 임계값 이상일 때 그림의 예시와 같이 새로운 LEVEL을 할당하는 방식으로 노드를 나누었다. 다양한 임계값을 실험해본 결과 10으로 설정시 노드가 지나치게 세분화 되어 불 필요한 레벨이 많이 생성되었다. 임계값을 30으로 설정시에는 서로 가까운 노드가 동일한 레벨로 묶이는 경우가 발생하여 임계값을 20으로 설정하였다. 이후 하위 LEVEL 노드를 상위 LEVEL 노드에 연결함으로써 물리적 인접성을 보장하면서 노드의 연결 순서를 결정하고 연결 가능한 점군집을 탐색하였다. 그림 14는 점군집 간의 유사도를 산출한 예시로 연결된 모든 노드 간의 유사도를 나타낸 것이다.

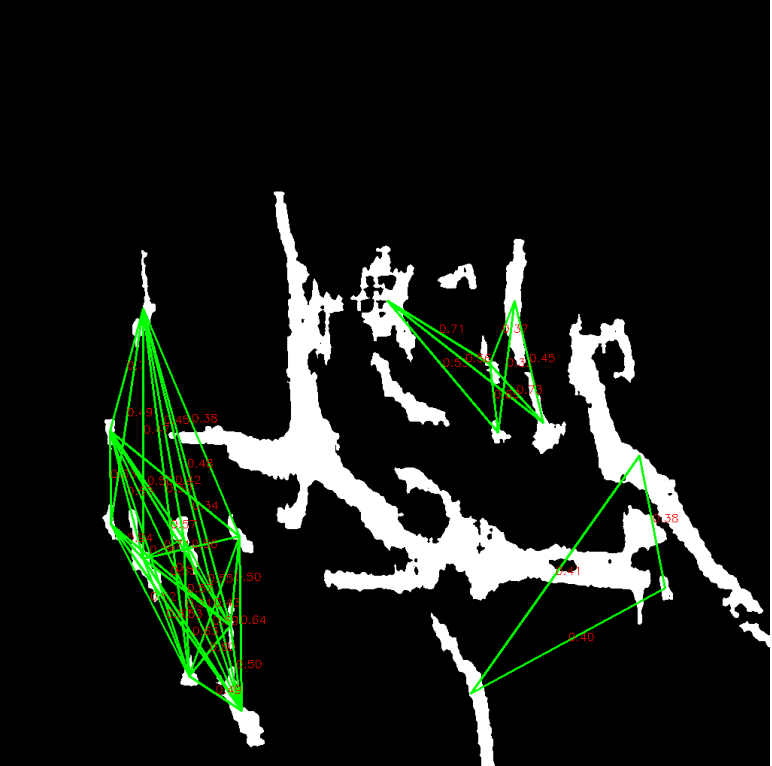


그림 14 유사도 계산 결과

유사도는 수식2 와 같이 주성분 분석으로 산출한 점군집의 각도 성분과 점군집의 중심점 좌표값을 하나의 척도로 변환하여 계산한다.

DRW00000fa80157 **수식2**

이때, 와 는 두 점군집의 주성분 각도이며 와 는 두 점군집의 중심점 좌표값이다. 이 둘의 차이를 각각 주성분 각도 차이의 최댓값을 의미하는 와 점군집 중심점 좌표값 차이의 최댓값을 의미하는 로 나누어 정규화한다. 이를 통해 점군집 간의 방향성과 위치적 특성에 기반한 점군집간 유사도를 산출할 수 있다. 그림 15는 수치로 표현된 점군집 간의 유사도가 0.8이상일 경우 간선을 생성한 이미지이다.

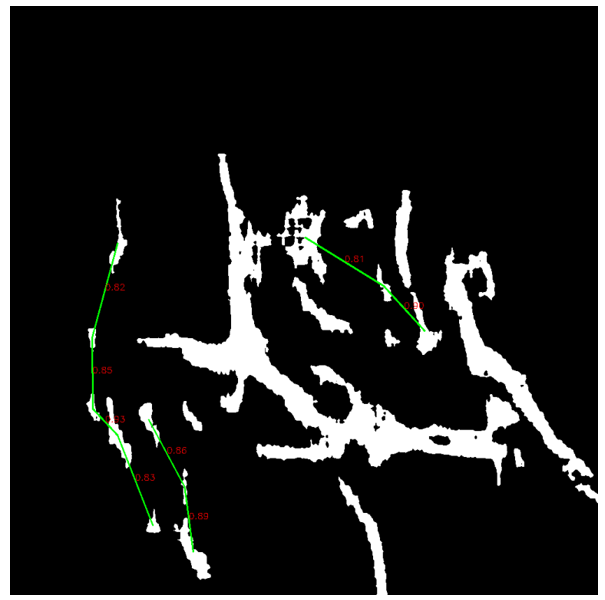


그림 15 간선 생성 이미지

하나의 점군집에 간선이 여러 개 생성될 경우 유사도가 제일 높은 간선만을 선택하여 점군집을 연결한다. 이때 간선은 점군집의 레벨에 따라 순차 연결되며 간선으로 연결된 점군집들은 하나의 불연속면으로 결정된다.

**2.2.4 점군집 연결**

생성된 간선을 이용하여 점군집을 연결하는 과정을 거친다. 먼저 후처리 과정을 거치지 않은 SAM-Adapter를 통해 예측된 이미지에 대해 슈퍼 픽셀화를 진행한다. 그림 16 은 슈퍼픽셀화된 이미지를 나타낸것으로 2.2.2절에서 최종 생성된 간선 이미지와 결합하여 점군집을 연결한다.

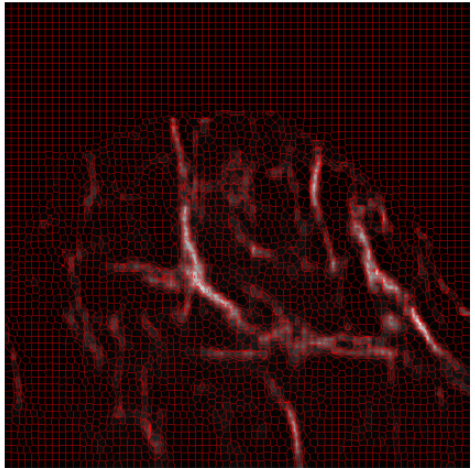


그림 16 슈퍼픽셀화 이미지

슈퍼 픽셀화는 이미지 픽셀들에 대해 유사한 픽셀 값을 가지는 픽셀들을 묶어 하나의 픽셀로 만드는 과정이다. 후처리 과정을 거친 이미지들은 이진화 되어 화소값이 0,1로만 구성되기 때문에 세부적인 경계 정보를 반영하기 어렵다. 따라서 그레이스케일로 구성된 후처리 이전의 예측 이미지를 사용하여 슈퍼 픽셀화를 진행한다. 그림 17은 슈퍼 픽셀화된 이미지와 간선이 생성된 이미지를 이용하여 점군집을 연결한 이미지를 나타낸 것이다.

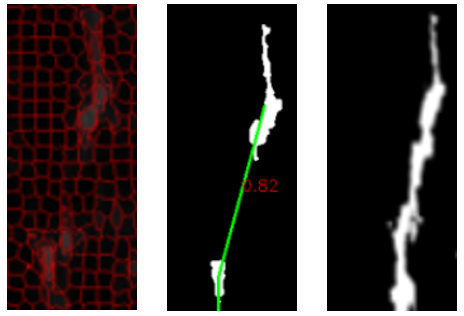


그림 17 점군집 연결 이미지

슈퍼 픽셀화된 이미지에서 간선이 지나는 경로에 해당하는 슈퍼 픽셀의 화솟값을 255로 설정하여 각 점군집을 연결하였다. 이를 통해 본래는 하나로 연결된 불연속면이었지만 추출 과정에서 끊어진 점군집들을 다시 연결함으로써 불연속면의 연속성을 회복하였다.

**2.2.3 두께 조정**

SAM-Adapter와 후처리를 통해 막장 불연속면을 예측한 이미지에서는 각각의 픽셀이 불연속면에 속하는지 여부를 개별적으로 판단하기 때문에 불연속면이 명확한 경우에는 두껍게, 불명확한 경우에는 얇게 나타난다. 따라서, GT이미지와 비교하기 위해선 예측 이미지의 두께를 일정하게 나타내는 작업이 이루어져야 한다. 예측 이미지의 두께 조정을 위해 그림 18에 도시된 바와 같이 불연속면을 예측한 이미지에 모폴로지 연산을 수행하는 단계와 골격화 하는 단계 그리고, 잔 가지를 제거하는 단계를 거쳐 최종적으로 두께를 조정한다.

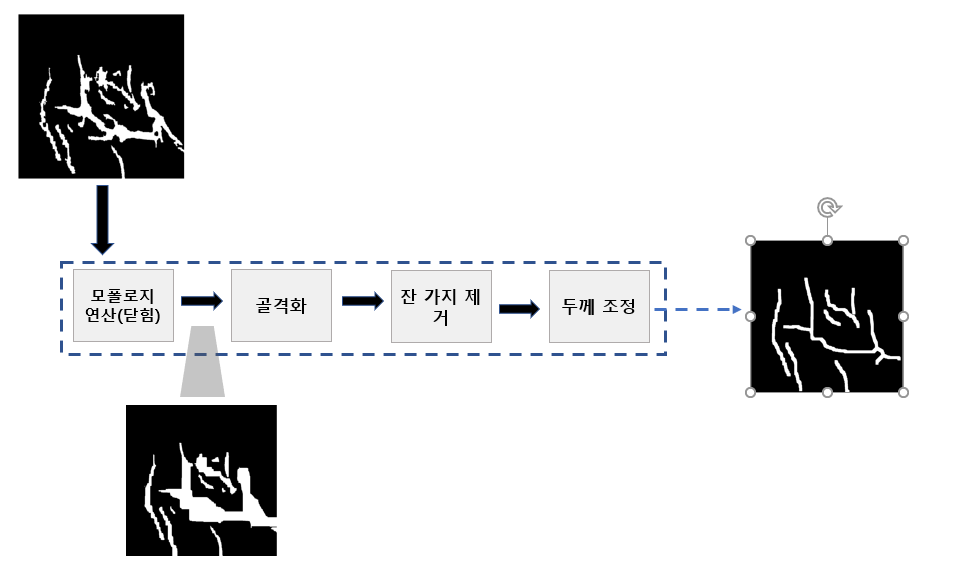


그림 18 두께 조정 흐름도

먼저, 불연속면을 예측 이미지에서 개별 선들에 대해 모폴로지 연산을 수행한다. 모폴로지 이진화된 이미지에서 객체의 모양을 다듬거나 노이즈를 제거하는데 사용되는 기법이다. 모폴로지 연산을 수행하지 않으면 불규칙한 경계선으로 인해 이후 단계인 골격화 단계에서 선들이 불필요하게 뻗어 나가거나 수 많은 가지가 발생하면 불연속면 인식에 방해가 된다 본 연구에서 사용한 모폴로지 연산의 코드는 그림 19와 같다.



그림 19 모폴로지 연산 코드

OpenCV의 cv2.morphologyEx를 사용하여 모폴로지 연산을 수행하였으며 함수 사용 시 커널의 크기, 연산의 종류, 반복 횟수를 지정해야 한다. 커널은 모폴로지 연산에서 사용되는 구조 요소로 연산을 적용할 때 기준이 되는 필터이다. 커널의 크기는 연산 강도에 영향을 미치며, 값이 클수록 더 강한 연산 효과를 얻을 수 있다. 연산의 종류로는 침식, 팽창, 열기, 닫힘 연산이 있다. 침식은 객체의 경계선을 축소하여 노이즈를 제거하고, 팽창은 객체의 경계선을 확장해 구멍을 메우거나 끊어진 경계선을 연결하며, 열기는 침식 후 팽창을 적용해 배경 노이즈를 제거하고, 닫힘은 팽창 후 침식을 적용해 끊어진 부분을 연결하거나 내부 구멍을 메운다. 본 연구에서는 선 내부에 있는 내부 구멍을 메꾸고 경계선 사이를 연결하기 위해 침식연산을 수행 하였다.

이후, 모폴로지 연산을 완료한 이미지에 대해 골격화를 진행한다. 골격화는 불연속면을 이루는 경계선의 중심축만을 남기고 나머지 픽셀을 제거하는 과정으로 선의 가장 중요한 주축을 추출하는 방법으로 코드의 예시는 그림 20과 같다.



그림 20 골격화 코드

Python의 이미지 처리 라이브러리인 scikit-image에서 제공하는 골격화함수 skeletonize를 사용하였으며 그림21과 같이 골격화를 진행하면 불연속면의 가장 기본적인 구조만이 남게 된다.

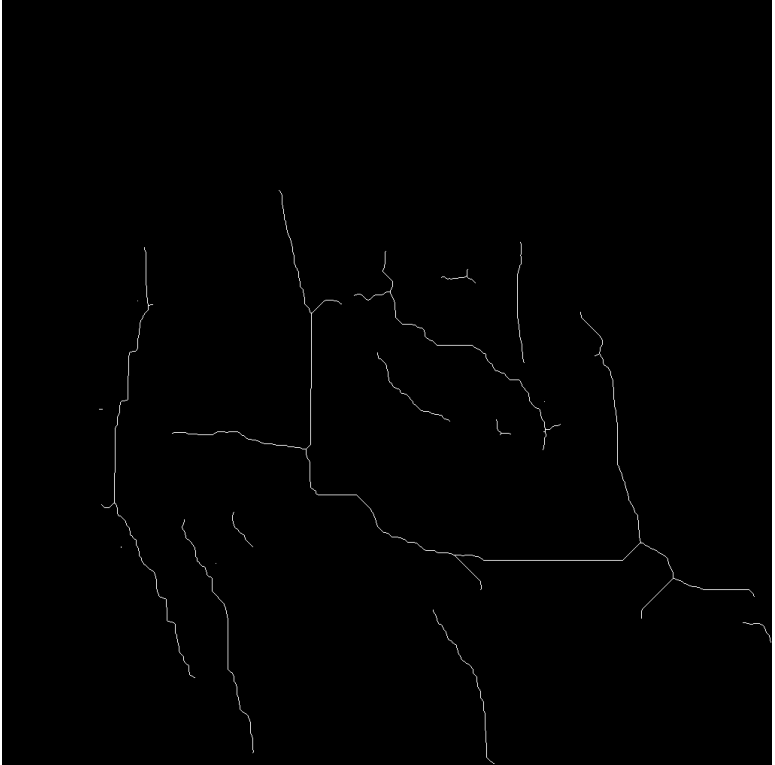


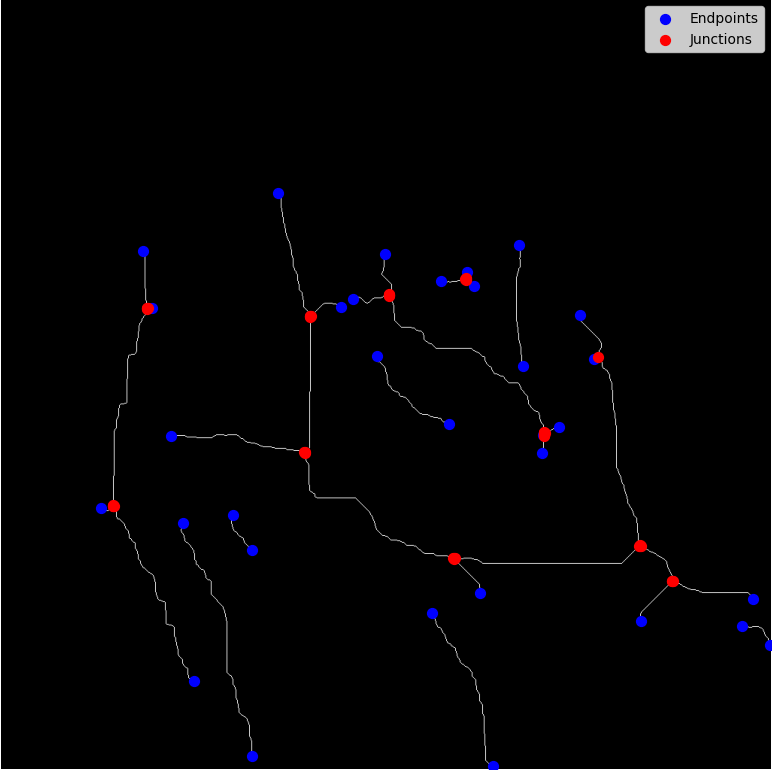
그림 21 골격화 결과

그림 19 골격화 결과

골격화를 거친 이미지에서 불연속면의 주요 축이 남게 되지만, 주요 축이 아닌 다른 방향으로 뻗어 나간 잔가지들이 남아 있다. 이러한 선들은 불연속면의 주요 구조를 왜곡할 수 있어 다른 방향으로 뻗어 나간 선들을 제거하는 과정을 거친다.

이를 위해, 골격화 과정을 거쳐 주요 축만 남긴 이미지에서 그림 22와 같이 교차점과 끝점을 찾는다. 교차점과 끝점을 찾는다. 끝점과 교차점을 찾기 위해 각 픽셀들을 상하좌우 및 대각선을 포함한 8방향에 이웃한 픽셀이 있는지 확인한 뒤, 이웃한 픽셀이 하나일 경우는 끝점, 이웃한 픽셀이 세 개 이상이면 교차점으로 판단하였으며 코드의 예시는 그림 23과 같다.

그림 22 끝점 및 교차점 추출



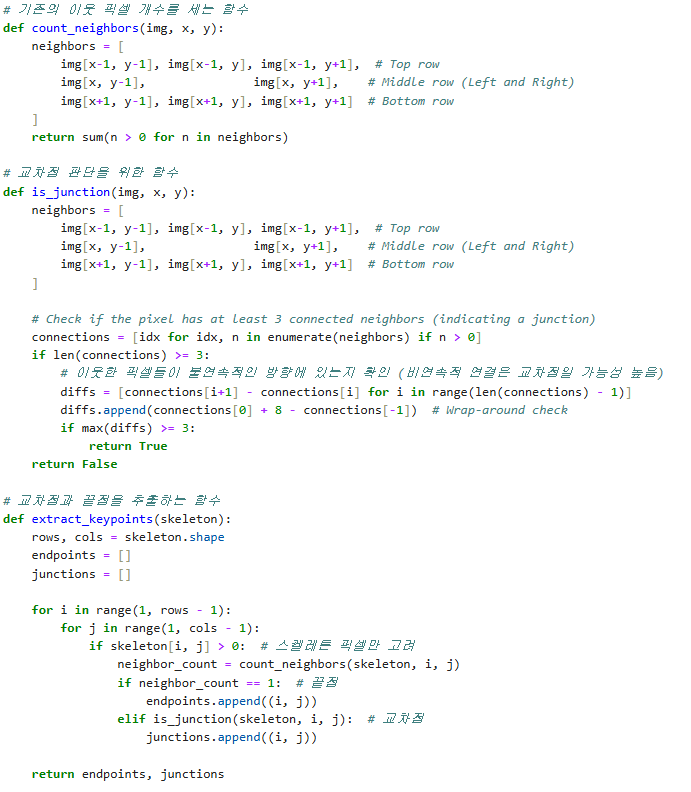


그림 23 교차점 및 끝점 추출 코드

Count\_neigbors함수는 주어진 픽셀의 8방향 이웃 픽셀중 화솟값이 0이아닌 이웃 픽셀을 리스트로 저장하여 sum함수를 이용해 반환한다. is\_junction함수는 주어진 픽셀이 교차점인지 판단하는 함수로 extract\_ketpoint함수에서 이웃 픽셀의 수를 계산하기 위해 사용한다. is\_junction 함수는 주어진 픽셀이 교차점인지 판단하는 함수로, 8방향 이웃 픽셀 중 연결된 픽셀의 불연속성을 계산하여 교차점 여부를 확인한다. 이 함수는 extract\_keypoints 함수에서 교차점인지 판단하는 데 사용된다. extract\_keypoints 함수는 스켈레톤 이미지에서 끝점(이웃이 1개인 픽셀)과 교차점(불연속 연결이 3개 이상인 픽셀)을 탐지하여 각각 리스트로 저장한 뒤 반환한다. 이후 잔가지 제거를 위해 교차점과 끝점의 길이를 계산하여 미리 설정된 임계값 이하일 경우 픽셀의 화솟값을 0으로 바꾸어 불연속면을 이루는 선에서 제외하였다.

다음으로 골격화 이미지의 선들의 두께를 조절한다. 이를 위해 모폴로지의 팽창연산을 적용하였다. 팽창연산은 이미지에서 영역을 확장시키는 역할을 하며 이를 통해 얇은 골격화 선을 굵게 만들어주었다. 팽창 연산에서는 정사각형의 형태로 된 커널이 사용되며 커널의 크기에 따라 선의 두께가 조절되고 그림 24는 선의 두께가 조절된 이미지이다.

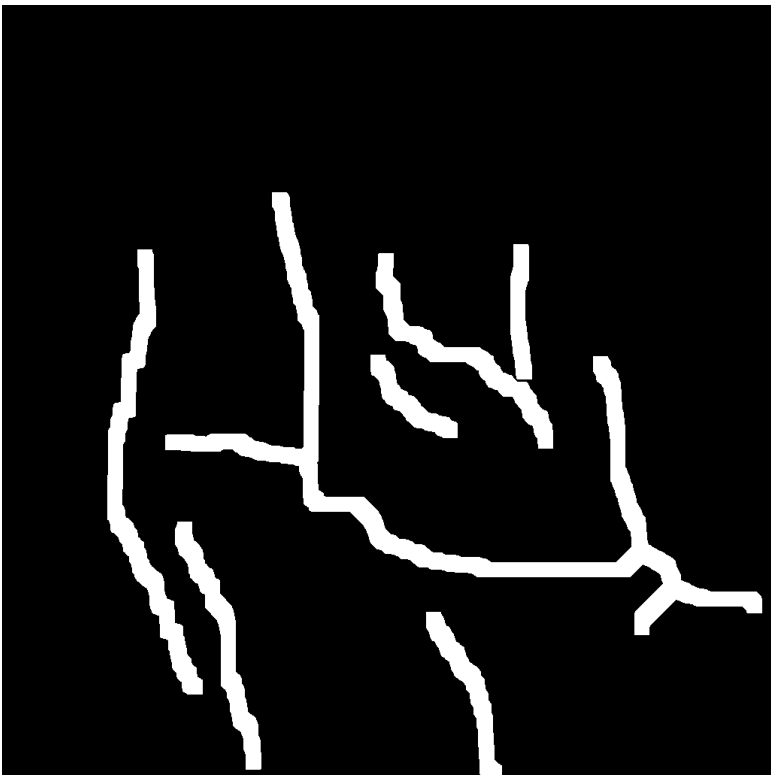


그림 24 두께 조절 이후 이미지

**3. 성능 검증**

이후, 두께조절을 완료한 이미지와 GT이미지 간의 비교를 위해서 GT이미지의 구역을 분할하였다. 한 구역당 2\*2 픽셀의 크기로 분할하였으며 이렇게 분할된 각 구역 안에 불연속면 이미지가 포함되어 있는지 여부를 기준으로 이진화 처리를 진행하였다.

즉, 불연속면이 포함된 구역은 1로 설정하고, 포함되지 않은 구역은 0 으로 설정하였다. 그림 25는 구역 설정의 예시를 도식화한 것이다.

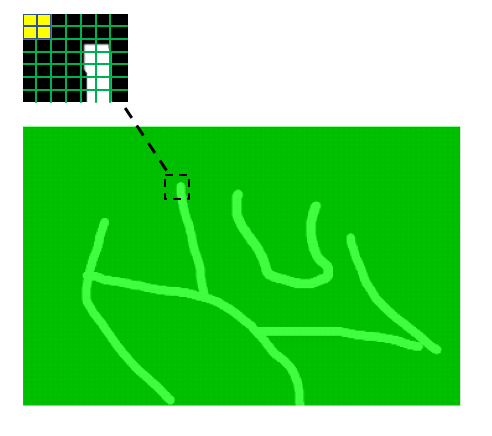


그림 25 구역설정 예시

그림의 예시와 같이 2\*2 픽셀의 크기로 구역을 나누고 이 구역내에 불연속면이 0.5이상 포함되어 있으면 1, 포함되지 않으면 0으로 설정하였다.

그림 7은 GT이미지와 예측 이미지를 분할한 후 분할한 구역의 값의 예시를 그림으로 나타낸 것이다. 그림을 보면 GT이미지와 예측 이미지 간의 구역 값이 차이가 나는 부분이 있다. 이를 통해 식(1)과 같이 예측 이미지의 성능을 평가 하였다.

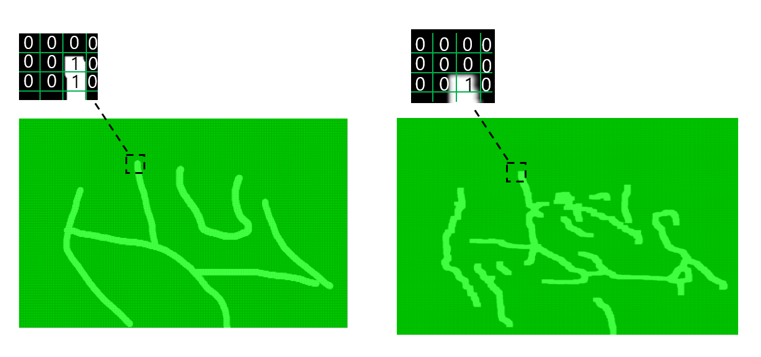
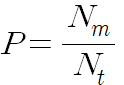


그림 26 구역 일치 예시

 식(1)

여기서 은 GT이미지와 예측 이미지 간 일치 구역의 수 는 모든 구역의 수이다. 위의 방법으로 검증 이미지 32장을 대상으로 성능평가 시 91.22%의 정확도를 보였다.

**4. 결 론**

본 연구에서는 막장 불연속면 구분을 위해 SAM-Adapter를 사용하여 불연속면을 예측하고, 후처리 과정을 통해 이미지를 보정하는 방법을 제안하였고 이를 검증하였다.

후처리 과정은 불연속면 예측 이미지에서 불필요한 노이즈를 제거하고, 추출된 점군집을 연결함으로써 불연속면의 구조를 복원하는 데 중점을 두었다. 예측된 이미지 포함된 작은 잡음이나 단절된 점군집은 후처리 과정을 통해 제거 및 연결되어, 불연속면이 본래의 연속적인 구조를 되찾을 수 있도록 하였다. 특히, 이 과정은 SAM-Adapter가 예측한 불연속면 경계를 보다 명확하게 하여, 불연속면 인식의 정확도와 신뢰성을 높이는데 기여하였다.

결과적으로 점군집을 연결함으로써 불연속면의 연속성을 일정 부분 복원할 수 있었으나, 완전한 연결이 이루어지지 않았고, 일부는 부분적으로만 연결되었다. 또한, GT이미지와 예측 이미지 간의 차이가 여전히 존재하여, 여러 개선이 필요한 점들이 남아 있다. 예를 들어 점군집 간의 잘못된 연결이나 예측 이미지의 경계선이 일정하지 않는 문제가 있었다. 이러한 문제들은 앞으로의 연구에서 해결해야 할 과제로 남아 있으며 보다 정교한 후처리 과정과 예측 모델의 개선이 필요할 것으로 보인다. 또한 성능 검증 방법에서 이미지를 구성하는 전체 픽셀 중 불연속면이 차지하는 비율이 적어 예측 이미지와 GT이미지간의 차이가 크더라도 성능 평가 결과가 높게 나오는 문제점이 있었다. 이 역시 향후 연구에서 개선해야할 중요한 과제로 남아 있다. 보다 정확한 성능 평가를 위해, 불연속면이 있는 부분과 없는 부분을 구분하여 평가하는 방식이나 새로운 성능 지표 도입이 필요해 보인다.

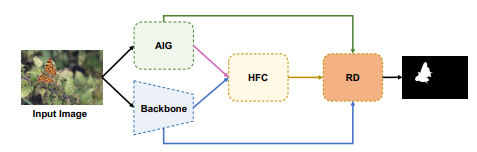
SAM-Adapter의 학습을 위한 하드웨어 및 소프트웨어는 OS Ubuntu 20.04, GPU NVIDIA RTXA6000, CUDA 12.4, Python 3.8을 사용하였다.

**4. 추가 모델 조사**

막장면은 오랜 시간 여러 암석이 뒤섞인 복잡한 환경으로 배경과 객체 간의 차이가 명확하지 않다. 이는 배경과 객체 간의 차이가 작은 위장객체를 탐지하는 것과 상당히 유사하다. 이에 따라 위장객체 탐지 모델에 막장 불연속면에 적용하는 방법을 찾아보고자 한다.

4.1) AIGNet(Additional Information Generation)

AIGNet은 이미지 내에 존재하는 다양한 추가 정보를 활용하지 못하는 기존 모델들의 한계를 극복하기 위해 설계된 모델로 상세 구성은 그림과 같다.



이미지가 입력되면 AIG모듈과 Backbone에 각각 입력되어 특징이 추출된다. Backbone에서는 추상적이고 전역적인 저해상도 특징부터 세부적이고 지역적인 고해상도 특징 같은 이미지의 범용적인 특징들이 추출되며, AIG모듈에서는 주파수, 경계, 질감, 엣지 총 네 가지의 특징 정보를 별도로 추출하여 추가 정보의 학습이 이루어진다. HFC모듈에서는 Backbone의 범용적인 특징과 AIG모듈에서 생성된 추가 정보를 결합하여 초기 예측이 이루어지고 초기 예측한 세그먼테이션 맵 을RD모듈로 전달한다. RD모듈은 수식 1과 같이 3번의 보정 과정을 통해 정보의 정밀도를 높이며, 이를 기반으로 최종 객체 탐지 결과를 생성한다.

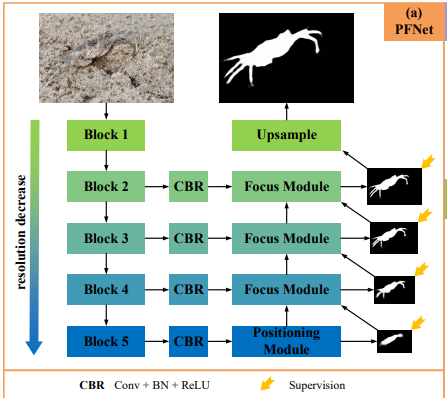
**수식1**

은 백본 네트워크에서 추출된 특징인 와 이전 단계의 세그먼테이션 예측 맵, 그리고 AIG모듈에서 생성된 추가 정보 를 입력으로 받아 새로운 세그먼테이션 예측 맵을 생성한다.

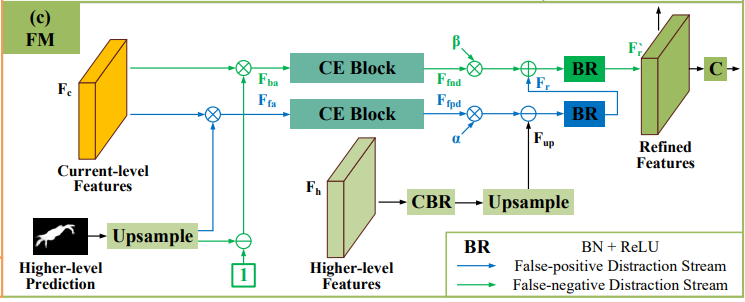
4.2) PFNet(Positioning and Focus Network)

PFNet은 객체와 배경 간의 모호한 경계를 효과적으로 처리하지 못하고 잘못된 예측을 효율적으로 제거하지 못하는 기존 방법의 한계를 극복하기 위해 설계된 모델로 자연에서의 포식 과정을 모방한 네트워크로서, 상세 구성은 그림 2와 같다.

PFNet은 위치 탐색(Positioning)과 집중(Focus)과정을 통해 위장된 객체를 분할하는 데 초점을 맞춘 모델이다.



이미지 데이터가 입력되면 Block들을 통과하게 되고 점진적으로 더 높은 수준의 정보를 생성한다. 각 블록을 통과할 때마다 Convolution, Batch Normalization, ReLU 로 구성된 CBR 연산이 적용되어, 이미지 내 패턴과 특징을 학습한다. 초기 블록은 주로 엣지, 텍스처와 같은 저수준의 특징을 추출하고, 블록을 거듭할수록 더 추상적이고 의미적인 고수준 특징이 생성된다. 마지막 블록에서 생성된 가장 높은 수준의 맵은 PM(Positioning Module)에 전달되고 PM은 이러한 고수준 특징과 채널 주의(Channel Attention)와 공간 주의(Spatial Attention)를 활용하여 대상 객체의 잠재적인 위치를 초기화 한다. PM에서 생성된 초기 위치 정보는 FM(Focus Module)으로 전달되어 더욱 정교한 객체 분할이 이루어진다. FM은 초기 위치 정보와 함께 현재 수준의 특징 맵과 더 높은 수준의 예측결과를 입력으로 받아 분산 탐지와 분산 제거를 수행하며 자세한 구조는 그림 3과 같다.

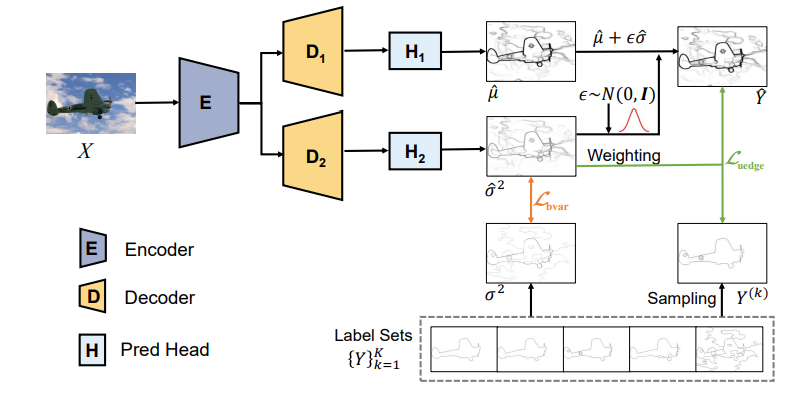


상위 수준 예측 정보를 통해 현재 특징 정보를 나타내는 에서 객체와 배경 관련 정보를 강조하여 CE(Context Exploration)Block에 전달한다. CE Block에서는 객체와 배경 정보의 탐지가 이루어지며 잘못 예측된 배경을 제거하고 누락 예측된 객체를 보완하며 을 생성한다. 이 과정을 반복적으로 수행하여 점진적인 정제를 통해 모호한 경계를 효과적으로 처리하고, 객체와 배경 간의 혼동을 줄인다.

막장 불연속면을 선이 아닌 면으로 인식하게 되면 경계의 모호성 때문에 불연속면이 끊어져 추출이 되거나 후처리 과정에서 예측된 불연속면이 변형 될 우려가 있어 막장 불연속면을 선으로 인식하는 모델을 찾아 적용해보고자 한다.

4.3) UAED(Uncertainty-aware Edge Detector)

UAED 모델은 사람마다 동일한 이미지에 대해 주관적인 관점으로 엣지를 인식하는 문제를 해결하기 위해 제안된 모델로서 상세 구성은 그림4와 같다.

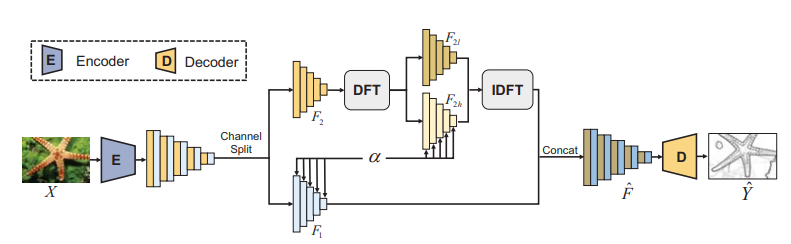


이미지가 입력되면 인코더를 통해 이미지의 특징 맵이 추출된다. 이후 두 개의 디코더에 전달되며, 첫 번째 디코더는 입력된 특징 맵을 기반으로 각 픽셀이 에지를 포함할 가능성을 예측하고 이를 평균값으로 나타낸다. 이 평균값은 해당 픽셀이 에지일 가능성을 나타내며, 값이 높을수록 에지일 가능성이 큰 픽셀임을 의미한다. 두 번째 디코더는 동일한 특징 맵을 기반으로, 다수의 GT이미지 간 해당 픽셀의 에지 여부에 대한 불확실성을 예측하며, 이를 분산값으로 나타낸다. 해당 논문에서의 불확실성은 다수의 GT이미지에서 서로 다르게 레이블링된 부분이다.

이후 첫 번째 디코더에서 예측한 평균값​과 두 번째 디코더에서 예측한 분산값을 이용해 각 픽셀에 대한 가우시안 분포를 생성한다. 생성된 가우시안 분포는 해당 픽셀이 에지일 가능성과 주석 간 불확실성을 모두 반영하며, 이를 통해 최종 예측 에지 맵을 샘플링한다. 샘플링된 예측값은 모델이 각 픽셀에서 에지를 판단한 결과를 나타내며, 후속 학습 단계에서 다수의 GT로부터 계산된 평균값과 분산값을 기준으로 손실을 계산하여 모델의 정확도를 높인다. 최종적으로 모델은 학습 과정에서 이러한 불확실성을 점진적으로 활용해 에지가 모호한 영역에서도 성능을 유지한다.

4.4) MUGE(Multiple Granularity Edge Detection)

MUGE 모델은 단일 수준의 예측 이미지만 생성하는 기존의 모델들과 달리, 에지의 주관성과 복잡성을 고려하여 다양한 수준의 예측 이미지를 생성하는 모델로서 상세 구성은 다음과 같다.



인코더를 통해 추출된 특징의 채널을 기준으로 절반씩 과 로 분리한다. 에서는 엣지로 예측된 픽셀의 수에 따라 α값이 0~1값으로 결정되며 이 값을 에 곱하여 에지의 세밀도를 조정한다. α값이 클수록 에지의 세부 디테일이 강조되고, 작을수록 단순한 형태의 에지가 표현된다. 에서는 특징을 주파수 도메인으로 변환하여 고주파 성분과 저주파 성분으로 분리한다. 고주파수 성분은 텍스처 및 디테일한 성분이 포함되고 저주파수 성분에는 객체의 전반적인 윤곽선을 나타낸다. 고주파수 성분에만 α값을 곱하여 텍스처와 디테일한 정보의 강도를 조정하는데 α값이 작으면 고주파 성분이 약화되어 디테일이 감소하고, α값이 크면 디테일이 강조된다. 이후, 조정된 고주파 성분과 저주파 성분을 다시 합치고, 역 주파수 변환(Inverse DFT)을 통해 공간 도메인으로 복원한다. 이렇게 조정된 과 채널 단위로 결합하여 디코더에 전달하고, 디코더는 이를 기반으로 다양한 수준의 에지 맵을 생성한다.

이를 막장 불연속면에 적용할 시 여러 암석이 뒤섞인 배경과 불연속면을 구분하는 부분 간의 주파수가 다른 것을 이용하면 불연속면을 구분하는 선을 추출할 수 있을 것으로 예상한다.