공학석사 학위논문

딥러닝 기반 응급 기관 내 삽관 영상의 해부학적 구조물 분할에 관한 연구

A Study on the Segmentation of Anatomical Structure in Emergent Endotracheal Intubation Using Deep-Learning Algorithm

2023년 2월

서울대학교 대학원

협동과정 바이오엔지니어링전공

최 승 재

딥러닝 기반 응급 기관 내 삽관 영상의 해부학적 구조물 분할에 관한 연구

지도 교수 Sungwan Kim

이 논문을 공학석사 학위논문으로 제출함
2022년 12월

서울대학교 대학원
협동과정 바이오엔지니어링전공
최 승 재

최승재의 공학석사 학위논문을 인준함
2023년 1월

위 원 장 이 정 찬 (인)

부위원장 Sungwan Kim (인)

위 원 조 민 우 (인)

초 록

본 연구는 응급실에서 비디오 후두경을 이용해 촬영된 기관 내 삽관 영상을 이용하여 구강 내 구조물인 성대, 후두 덮개, 연골, 그리고 혀를 딥러닝을 이용해 segmentation하는 연구이다.

본 연구는 Mask R-CNN, DeepLabv3+, 그리고 U-Net 모델을 이용하여 segmentation을 진행했다. Mask R-CNN 모델은 추론을 통해 각 구조물에 대해 여러 개의 마스크가 생성될 수 있다. 본 연구에서 segmentation을 진행하는 구조물의 레이블링은 구조물별로 하나의 마스크에 표현되었다. 따라서 모델의 결과로 나오는 각 구조물에 대한 여러 개의 마스크를 각각 하나의 마스크로 만드는 과정이 필요하다.

본 논문은 Mask R-CNN 모델의 결과로 각 구조물 별로 나오는 여러 개의 마스크를 각각 하나의 마스크로 만드는 과정을 진행했다. 이후 DeepLabv3+, U-Net 모델과 같은 평가 방식을 이용해 모델의 성능을 검증하였다. 성능 검증을 위한 평가 지표로 dice similarity coefficient, detection모델 평가에 사용되는 방법을 도입하였고 frames per second도 평가 지표로 사용하였다.

본 연구에서 사용된 응급 상황에서 촬영된 기관 내 삽관 동영상은 실제 기관 내 삽관이 이뤄지는 환경을 반영하기 위해 기도 주변의 이물, 모션 블러, 그리고 빛 반사가 존재하는 이미지를 포함하여 연구가 진행되었다.

본 연구를 통해 실제 상황을 반영한 데이터는 딥러닝을 통해 구조물의 segmentation이 이뤄질 수 있음을 확인하였고, 실시간으로 활용이 가능할 수 있는 모델을 확인할 수 있었다. 응급 상황에서 촬영된 데이터를 이용한 첫 연구로 본 연구에서 개발된 알고리즘은 실제 촬영된 영상에 적용해 구조물이 segmentation된 영상을 얻을 수 있다. 이렇게

얻어진 영상을 통해 경험이 적은 의료 종사자들이 구강 내 구조물에 대한 이해를 높일 수 있을 것으로 생각되며, 원격 기관 내 삽관 보조 시스템 구축 및 기관 내 삽관 자동화 시스템 개발의 초석으로 사용될 수 있을 것이라 생각된다.

주요어 : 딥러닝, 기관 내 삽관, 영상 분할, 영상 처리
학 번 : 2021-27700

목 차

표 목차

그림 목차

제 1 장 서 론

제 1 절 연구의 배경 및 동향

기관 내 삽관은 중환자실, 응급실, 병동 그리고 수술실에서 심폐소생술, 의식 상실, 그리고 호흡부전과 같은 응급 상황에서 기도를 확보하기 위해 시행되는 핵심 기술 [1]로 mask ventilation이 어려울 때 또는 기계를 이용한 prolonged mechanical ventilation이 필요할 때 기도를 확보 및 보호하기 위해 필요하다 [2]. 하지만 기관 내 삽관 시행 시 첫 시도에 삽관을 성공하지 못하는 경우 환기가 충분하게 이루어지지 않을 수 있으며, 기도에 외상이 발생할 수 있다 [1, 3]. 비디오 후두경을 이용해 기관 내 삽관이 이뤄지는 경우 후두를 잘 보이게 해주어 기관 내 삽관을 용이하게 하여 삽관 성공률을 높임으로써 환자 안전에 기여할 수 있다 [4, 5].

딥러닝을 이용한 기관 내 삽관과 관련된 연구들로 구조물의 segmentation을 시도한 연구들이 있다. 컴퓨터 보조 진단 시스템을 위한 첫 단계로 U-Net [6] 기반의 모델을 이용해 후두 이미지를 이용한 glottal area를 segmentation 하고자 하는 연구가 있었다 [7]. transformer [8, 9]와 convolutional neural network (CNN)를 결합한 모델을 이용해 후두경 이미지에서 구조물을 segmentation하려는 연구가 있었으며 [10], Mask R-CNN [11] 모델로 이비인후과 의사들의 진료 시 필요한 정보를 제공하기 위해 vocal fold와 glottal region을 segmentation하는 연구를 진행하였다 [12].

기관 내 삽관 데이터를 이용한 연구로 segmentation뿐만 아니라

classification 및 detection과 관련된 연구도 진행되었다. 진단에 도움을 주기 위해 CNN 기반의 classification 모델을 이용해 후두 종양을 분류하는 연구가 진행되기도 하였고 [13], YOLO 기반의 모델을 이용하여 비디오 후두경 데이터에서 laryngeal squamous cell carcinoma을 detection하는 연구가 진행되기도 했고 [14], YOLOv3 [15]를 이용하여 후두 덮개, 성대 주름, 기관 내 튜브들을 detection하는 모델을 개발하기도 하였다 [16].

제 2 절 연구의 목적

본 연구에서는 응급실에서 비디오 후두경을 이용해 기관 내 삽관을 시행하면서 촬영된 동영상에서 구조물을 segmentation하는 알고리즘을 개발하였다. 사람이 기관 내 삽관 술기를 시행하는 경우 구강 내의 구조물을 정확하게 파악 및 인지하는 것부터 시작한다. 이에 기관 내 삽관 자동화 로봇과 같은 자동화 시스템을 위해서는 먼저 구조물을 정확하게 인식할 수 있어야 한다. 자동화 시스템을 개발하기 위해서는 우선적으로 실제 상황에서 촬영된 데이터에서 구조물을 인식하여 segmentation할 수 있어야 한다.

기관 내 삽관은 전통적으로 직접 후두경을 이용하여 시행되어 왔다. 하지만 직접 후두경은 술자의 숙련도나 환자의 신체 상태에 따라서 성대가 보이지 않는 경우가 발생하여 삽관이 실패할 수도 있다. 첫 삽관이 실패하여 반복적으로 삽관을 시도할 경우 흡인 위험성 증가, 환기 부족으로 인한 산소포화도 저하, 기도 외상의 가능성이 증가하여 환자의 치명률이 증가하게 된다. 반면, 비디오 후두경을 이용하여 삽관을 시행할 경우 비디오 후두경의 모니터에 구강 내 구조물을

비춰주어 술자의 삽관 숙련도나 환자의 신체상태에 영향을 적게 받아 삽관의 성공률이 직접 후두경보다 증가하게 된다.

후두경 이미지에 딥러닝을 적용해 segmentation과 classification을 진행하는 것으로 진단에 도움을 준다는 연구가 있다 [12, 13, 17]. 이러한 연구를 기반으로 응급 상황에서 비디오 후두경을 통해 촬영된 영상에서 구조물에 대한 segmentation을 진행하는 것으로 기관 내 삽관의 성공률을 더 높일 수 있을 것이다.

본 연구를 통해 개발된 모델들의 성능 평가를 위해 dice similarity coefficient (DSC), detection 모델의 성능 평가에 사용되는 방법, 그리고 frames per second (FPS)를 사용하였다.

본 연구를 통해서 응급 상황에서 촬영된 기관 내 삽관 영상에서도 구조물을 segmentation할 수 있는 알고리즘의 개발이 본 연구의 목적이다.

제 2 장 데이터셋

제 1 절 영상 데이터 수집 및 구성

본 연구는 후향적 연구로 분당서울대학교병원 임상 시험 심의위원회(Institutional Review Board, IRB)의 승인을 받았다 (number B-2112-725-102). 연구 프로토콜은 1975년 헬싱키 선언 및 그 이후 개정된 윤리 가이드라인을 준수했다. 사전 동의는 분당서울대학교병원 임상 시험 심의위원회를 통해 면제되었다.

분당서울대학교병원 응급실에서 2020년 10월 10일부터 2021년 1월 2일까지 GlideScope Go (Verathon, US)의 비디오 후두경을 사용해 기관 내 삽관이 시행된 영상 중 54건을 사용하였다. 촬영된 영상은 초당 30프레임 AVI 확장자로 640 × 480 해상도로 촬영되었다.

제 2 절 영상 데이터 전처리

동영상에서 이미지를 추출하기 위한 기준을 정하기 위해 동영상에서 이미지를 추출하여 연구를 진행한 선행 논문들을 조사했다.

연구 [18]에서는 초당 15 프레임으로 저장된 동영상에서 초당 5 프레임으로 이미지를 추출하였고, 연구 [19]에서는 최대 30 프레임을 갖는 초음파 영상에서 초당 3 프레임으로 이미지를 추출하였고, 연구 [20]에서는 초당 30 프레임으로 녹화된 수술 동영상에서 수술 과정에 따라 14개의 phase로 나눴고 각 phase에 따라 랜덤하게 프레임 당 최소 0.3초 차이가 존재하도록 10 ~ 20 프레임의 이미지를 추출하였고,

연구 [21]에서는 동영상의 프레임 수는 나타나 있지 않았지만 3 프레임 당 1 프레임의 이미지를 추출하였고, 그리고 연구 [22]에서는 초당 30 프레임으로 저장된 대장 내시경 동영상에서 초당 1 프레임의 이미지를 추출하였다. 동영상을 이용해 진행된 연구에서 이미지를 추출할 때 초당 프레임 수에 대한 기준은 연구에 따라 상이하였다. 또한 실제 응급 상황에서 촬영된 기관 내 삽관 동영상을 이용한 연구는 본 연구가 처음이기 때문에 본 연구에서는 유사해 보이는 프레임의 정도 및 레이블링에 소요되는 시간을 임상의 3명이 고려해 논의하여 초당 2장 (15프레임 당 1장)으로 추출하기로 하였다.

VirtualDub [23] 프로그램을 사용하여 54건의 동영상에서 이미지를 추출하였다. 추출 시 소요되는 시간 및 추출된 데이터가 차지할 데이터의 크기를 고려했을 때, 동영상이 촬영된 전체 시간에서 이미지를 추출하는 것이 아닌 각 동영상마다 추출할 시간 간격을 설정하여 설정된 시간 간격에서 이미지 추출을 진행하였다.

이미지를 추출하기 위한 시간 간격의 설정은 기관 내 삽관이 처음 시도된 시점 (비디오 후두경이 환자의 구강 내부로 진입한 시점)에서 삽관이 종료된 시점 (동영상이 종료되었거나 비디오 후두경이 환자의 구강에서 완전히 벗어났다고 판단된 시점)까지로 설정하였다. 이는 그림 1에서 확인할 수 있다.

일부 영상에서는 비디오 후두경이 환자의 구강에서 벗어나더라도 해당 동영상에서 다른 시간 축에서 기관 내 삽관을 다시 시도하는 경우도 존재했다. 이 경우 그림 2에서 같이 비디오 후두경이 구강에서 완전히 벗어났더라도 이를 따로 분류해서 이미지를 저장하지 않고 해당 동영상에서 삽관이 완전히 종료되었다고 판단된 시점 (후두경이 구강을 벗어나서 후두경이 구강을 다시 촬영하지 않거나 영상이 종료된 경우)을 종료 시점으로 정하여 추출을 진행했다.

위 과정을 통해 총 8,973장의 이미지가 추출되었다. 추출된 이미지에서 이미지가 깨진 경우 및 구강 외에서 촬영된 이미지는 1차적으로 제외되었다.

기관 내 삽관이 시작될 시점 기관 내 삽관이 종료된 시점

[그림 1] 영상에서 이미지를 추출하기 위한 구간 설정을 보여주는 예시 그림.

기관 내 삽관이 시작될 시점 구강 안쪽 촬영

구강에서 벗어남 다시 구강 안쪽 촬영

[그림 2] 기관 내 삽관이 여러 번 시행되는 경우에 대한 예시 이미지.

제 3 절 데이터 레이블링

1차 데이터 제외 과정을 통해 남은 이미지에서 성대 (vocal cord), 후두 덮개 (epiglottis), 연골 (corniculate cartilage), 그리고 혀 (tongue)에 대해 Labelme [24] 프로그램을 이용하여 3명의 임상의가 레이블링을 진행하였다.

본 논문에서 사용된 데이터를 이용해 레이블링이 진행되는 과정에서 1차 데이터 제외 과정에서 제외되지 않은 이미지와 non-informative한 이미지는 레이블링이 진행되지 않았다. 레이블링이 진행된 데이터는 resection surgery과 examination 같은 안정된 상황에서 촬영된 데이터가 아닌 응급 상황에서 촬영된 데이터로 기도 주변의 이물, 모션 블러, 빛 반사와 같은 다양한 변수가 존재하는 데이터로 이는 그림 3에서 확인할 수 있다. 응급 상황에서 존재할 수 있는 상황을 반영하기 위해 레이블링 할 해부학적 구조물이 다른 구조물이나 이물에 의해 가려지더라도 주변 구조물과의 관계 상 있어야 할 곳으로 인식되는 부위를 레이블링 하기로 하였다. 3명의 임상의가 54건을 나눠 레이블링을 진행하였다. 각 구조물에 대해 논의를 하여 아래의 원칙을 정하여 레이블링이 진행되었다.

- 성대
  - 성대가 명확하게 보이는 것을 기준으로 레이블링을 진행한다.
  - 성대를 레이블링 할 때 성대 안쪽의 trachea도 포함하여 레이블링을 진행한다.
- 후두 덮개

- 후두 덮개와 tongue base 사이의 경계가 애매하며, 후두 덮개는 카메라의 각도에 따라 다양한 형태로 보일 수 있다.
- 후두 덮개와 tongue base의 경계는 주변 구조물과의 관계를 통해 빛 반사에 의해 색이 변하는 지점이나 후두 덮개가 접히는 지점을 기준으로 레이블링을 한다.

● 연골
- 각도에 따라 다르게 보일 수 있으나 기본적으로 좌/우 corniculate/cuneiform cartilage까지 합하여 레이블링을 진행한다.
- Corniculate cartilage의 위/아래 경계는 빛의 조도가 변하는 지점으로 하기로 한다.

● 혀
- 혀의 앞면과 뒷면의 해부학적 특성이 다르나 하나의 레이블로 레이블링을 진행한다.
- 빛 반사, 이물에 관계없이 임상의가 혀로 인식할 수 있는 경우 레이블링을 진행한다.
- 후두 덮개와 tongue base가 맞닿는 부분은 약간의 여유를 두고 레이블링을 진행한다.

레이블링을 진행하는 과정에서 이견이 발생한 경우 이미지를 보면서 논의를 통해 의견을 통일하여 진행하였다. 레이블링이 완료된 데이터는 최종적으로 1명의 임상의가 검수를 진행하였다.

[그림 3] 데이터셋 예시 이미지.

제 4 절 레이블링 데이터 처리

레이블링 된 데이터는 JSON 파일로 저장되었다. 저장된 JSON 파일을 Python으로 작성된 코드로 처리하여 NPY 파일을 얻는다. 이후 NPY 파일을 읽어 구조물이 한 이미지에 보일 수 있도록 그레이 스케일 PNG 파일로 저장했다. 그레이 스케일 이미지를 만들기 위해 각 구조물의 픽셀 값은 성대, 혀, 후두 덮개, 그리고 연골에 대해 60, 120, 180, 그리고 240 픽셀 값으로 각각 설정하여 시각화가 쉽도록 하였다. 이렇게 총 4,956장의 그레이 스케일 마스크 이미지가 얻어졌고 이 과정은 그림 4에서 확인할 수 있다.

[그림 4] 레이블링 된 데이터를 처리하여 그레이 스케일 마스크 이미지 파일을 만드는 과정.

제 5 절 데이터셋 구성

레이블링 된 데이터는 학습, 검증, 그리고 테스트 셋으로 나눠졌다. 각 영상마다 임상의가 주변 환경에 따라 CPR(cardiopulmonary resuscitation), VD (visual difficulty, VD), CPR&VD 그리고 Others로 영상을 레이블링 하였다. CPR은 심폐소생술이 이루어지는 상황에서 기관 내 삽관이 시행된 경우로 정의하였다. VD는 임상의가 판단하였을 때 구강 내에 토사물, 혈액, 점액, 이물질, 빛 반사 등으로 해부학적 구조물을 인지하는 데 방해가 되는 시야장애가 있는 경우로 정의하였다. CPR&VD는 CPR과 VD 모두 있는 경우로 정의하였으며 Others는 CPR과 VD 모두 없는 경우로 정의하였다. 데이터셋을 만들기 위해서 나눠진 영상 레이블이 각 셋에 적절한 비율로 들어 갈 수 있도록 고려하였고 하나의 케이스가 다른 셋으로 나눠지지 않도록 고려했다. 데이터는 랜덤하게 나뉘어졌으며, 각 영상의 레이블을 고려해 학습 셋,

검증 셋, 그리고 테스트 셋으로 분류하였으며 분류된 케이스 수의 비율이 6:2:2가 될 수 있도록 고려했다. 이를 통해 학습 셋 (32케이스, 2,888장), 검증 셋 (11케이스, 1,177장), 그리고 테스트 셋 (11케이스, 891장)으로 나눠졌다. 데이터셋에 대한 케이스의 분류, 케이스의 수 그리고 이미지 장 수는 표 1에서 확인할 수 있다.

[표 1] 기관 내 삽관 데이터셋의 구성.

	학습 셋	검증 셋	테스트 셋
CPR	4케이스	1케이스	1케이스
VD	9케이스	3케이스	3케이스
CPR&VD	11케이스	4케이스	4케이스
Others	8케이스	3케이스	3케이스
총 케이스	32케이스	11케이스	11케이스
총 이미지 수	2,888장	1,177장	891장

제 3 장 방 법

제 1 절 모델 선정

레이블링이 완료된 데이터를 사용해 이미지에서 성대, 후두 덮개, 연골, 그리고 혀의 segmentation을 진행하기 위해 Mask R-CNN [11], DeepLabv3+ [25], 그리고 U-Net [6]을 사용했다.

DeepLabv3+와 U-Net은 semantic segmentation 모델로 동일한 클래스에 대해 1개의 마스크로 표현된다. 이는 하나의 이미지에 여러 동일한 물체가 존재하더라도 모두 하나의 객체로 표현됨을 의미한다. Mask R-CNN 모델은 instance segmentation 모델로 동일한 클래스에 대해 개별적인 마스크로 표현된다. 이는 하나의 이미지에 여러 동일한 물체가 존재하면 개별적인 객체로 표현이 가능함을 말한다. 이는 그림 5에서 확인할 수 있다.

DeepLabv3+와 U-Net과 같은 모델은 입력된 영상에서 feature를 추출하는 부분인 encoder와 추출된 feature를 연산을 통해 원하는 마스크를 얻을 수 있도록 하는 decoder로 구성되어 있다. 두 모델에서는 encoder에서 얻어진 특징을 decoder를 통해 원하는 마스크를 얻을 수 있지만 multiclass segmentation에서의 클래스 간의 구분은 모델의 final layer의 output에 activation function인 softmax function을 통해 진행된다.

Mask R-CNN 모델은 Fast R-CNN [26] 그리고 Faster R-CNN [27]에서 발전한 모델이며 Fast R-CNN 그리고 Faster R-CNN은 detection 모델이다. 즉, Mask R-CNN은 detection 모델에서

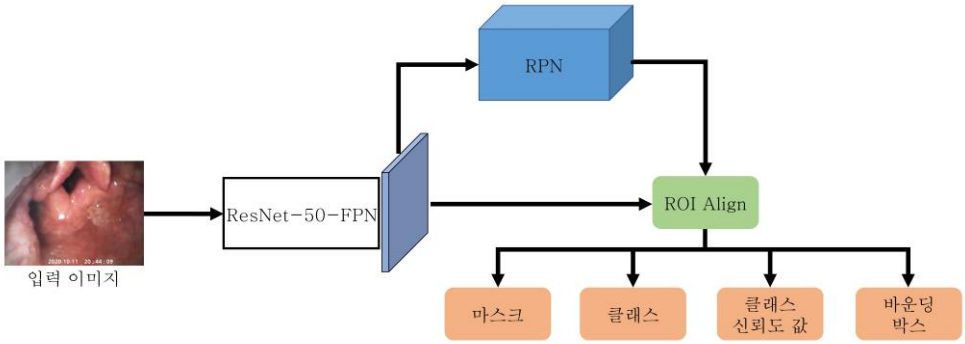
segmentation을 할 수 있도록 발전한 모델이고, YOLOv3 [15]와 같은 입력 데이터에 대해 1번의 과정을 통해 클래스, 클래스 신뢰도 값, 그리고 바운딩 박스가 얻어지는 1-stage detection 모델과 다른 2-stage detection 모델로 1차적으로 객체의 존재 유무를 파악하는 단계(배경과 객체의 존재를 구분하는 단계)가 존재하고 앞의 단계에서 객체가 존재하는 영역인 ROI (region of interest)를 구한다. 이렇게 얻어진 ROI 영역에서 모델의 클래스, 점수, 그리고 바운딩 박스를 계산하게 된다. 그렇기 때문에 동일한 클래스에 대해 서로 다른 객체로 나눌 수 있게 된다.

Mask R-CNN 모델이 동일 클래스인 객체를 각각 찾을 수 있는 이유는 그림 6에서 확인할 수 있는 region proposal layer (RPN)이 있기 때문이다. RPN을 통해 여러 개의 동일한 클래스의 객체가 존재하더라도 개별적으로 이를 나눌 수 있다.

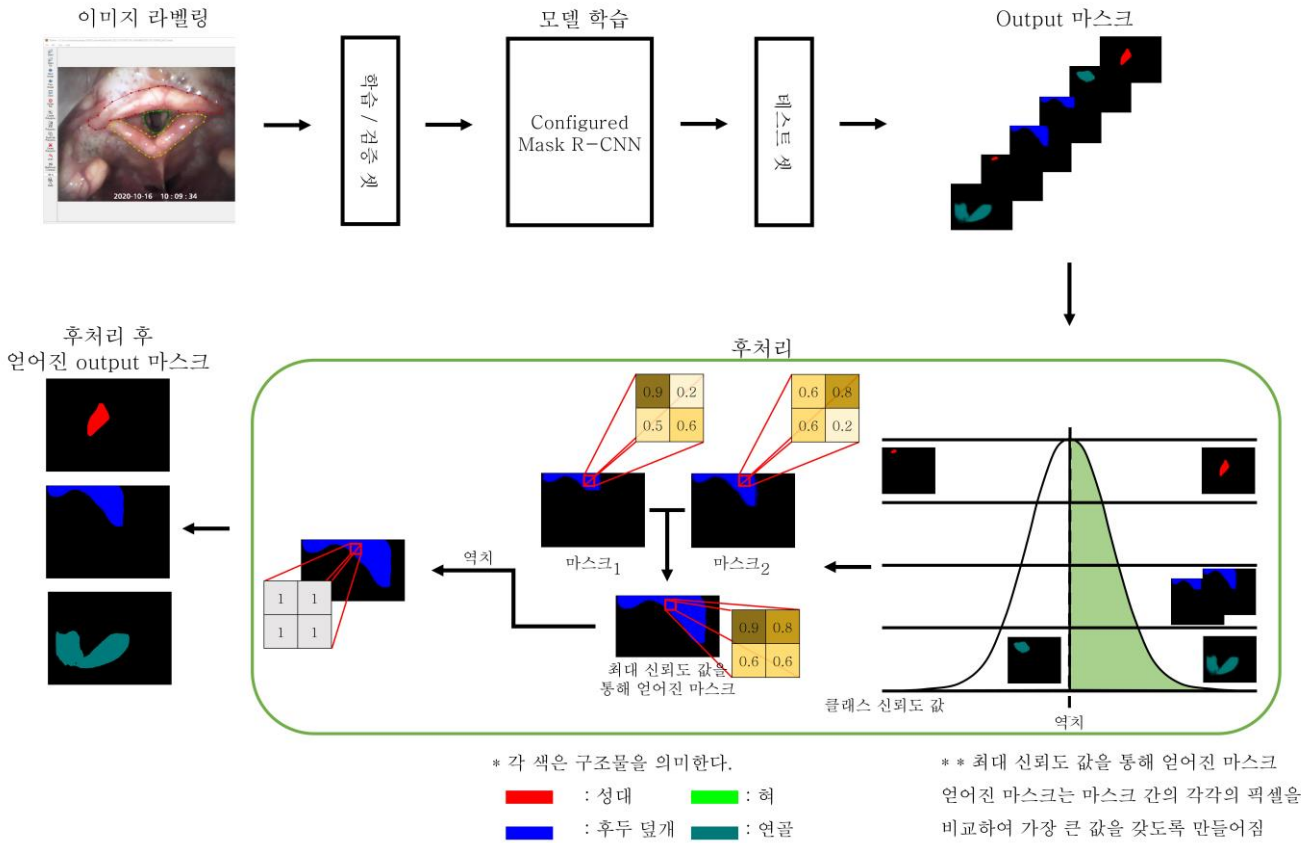
Semantic segmentation 모델은 모든 픽셀에 대해 각 픽셀이 어떤 클래스에 속하는지 구분하게 된다. 하지만 Mask R-CNN은 2-stage 모델로 객체가 존재할 수 있을 것 같은 영역인 ROI를 추출한다. ROI 영역에 대해서만 segmentation이 진행되기에 instance segmentation 모델인 Mask R-CNN을 본 연구에서 사용하였다. Mask R-CNN을 사용한 연구의 전체적인 과정은 그림 7과 같다.

Instance segmentation　　Semantic segmentation

[그림 5] Instance segmentation과 semantic segmentation 대한 예시 이미지. Instance segmentation은 각기 다른 객체로 표현되는 반면에 semantic segmentation은 동일한 객체에 대해 하나의 mask로 표현됨.



[그림 6] Mask R-CNN 모델의 구조.



[그림 7] Mask R-CNN을 이용한 연구의 전체적인 과정. Configured Mask R-CNN은 Mask R-CNN 학습과정에서 검증 셋을 검증할 때 나오는 여러 개의 마스크를 하나의 마스크로 만드는 과정을 거쳐서 학습된 모델을 의미함.

제 2 절 DeepLabv3+와 U-Net 모델 학습 과정

DeepLabv3+와 U-Net 모델은 segmentation models pytorch [28] 라이브러리를 이용해 모델을 구축하였다. 모델에 입력된 데이터에서 feature를 추출하는 부분인 인코더는 EfficientNet [29]으로 대체하여 사용했다.

EfficientNet은 모델을 구성할 때 모델의 깊이, 채널의 수, 그리고 입력 이미지의 해상도에 대해 최적화된 파라미터를 찾아 적용하여 모델을 구축하게 된다.

본 연구에서 사용된 EfficientNet 모델은 EfficientNet-B5를 사용하였다. 이는 모델의 입력 영상의 크기가 640 × 480으로 표2에서 이와 비슷한 입력 이미지의 해상도를 갖는 모델 중 (EfficientNet-B5 또는 EfficientNet-B7) 이를 인코더로 갖는 segmentation 모델 학습 시 batch size를 8 이상으로 설정할 수 있는 모델인 EfficientNet-B5를 선택하였다. EfficientNet-B5 모델을 인코더로 사용한 DeepLabv3+와 U-Net 모델을 본 논문에서는 EfficientNet-B5&DeepLabv3+ 그리고 EfficientNet-B5&U-Net이라한다. 구조물 segmentation을 위해 EfficientNet-B5 모델은 ImageNet으로 pre-trained된 가중치를 이용했다. EfficientNet-B5를 인코더로 사용한 EfficientNet-B5&DeepLabv3+와 EfficientNet-B5&U-Net 모델의 구조는 그림 8, 9와 같다. 표 2에서의 모델의 입력 이미지의 해상도는 가로/세로의 비율이 1:1이지만 본 연구에 사용된 모델의 입력 이미지의 해상도는 640 × 480, 비율이 4:3으로 1:1 비율이 되도록 입력 이미지 해상도의 수정은 진행하지 않았다. 이미지를 480 × 480 또는 640 × 640으로 리사이징을 진행하는 과정에서 보간이 이뤄진다. 보간이 이뤄지는

과정에서 픽셀 값이 변경되어 정보의 왜곡이 생기기 때문에 리사이징을 진행하지 않았다. 입력 이미지는 픽셀 값이 0 ~ 1의 값을 갖도록 scaling이 진행되었다.

EfficientNet-B5&DeepLabv3+ 그리고 EfficientNet-B5&U-Net 모델의 hyper parameter로 optimizer function은 Adam, learning rate은 0.0001, batch size는 8, epoch은 500, 학습 셋에 대한 loss function은 수식 (1)에서 확인할 수 있는 dice similarity coefficient loss (DSC loss)를 사용하였다.

$$\text{DSC loss} = 1 - \frac{2TP+1}{2TP+FP+FN+1} \quad (1)$$

수식 (1)의 TP는 true positive로 ground truth 마스크와 추론을 통해 얻어지는 마스크의 겹치는 영역을 의미한다. FP는 false positive로 추론을 통해 얻어지는 마스크만 존재하는 영역을 의미한다. FN은 false negative로 ground truth 마스크만 존재하는 영역을 의미한다. 이는 그림 10에서 확인할 수 있다. 모델은 매 epoch마다 검증 셋을 이용해서 평가가 이뤄진다. 검증 셋을 평가하는 방법은 loss function과 같은 DSC loss를 이용해 평가가 진행되며, 평가를 통해 얻어진 검증 loss가 가장 낮을 때 모델의 저장을 진행했다.

Learning rate은 epoch이 증가하면서 특정 조건을 달성하는 경우 감소하도록 만들었다. Learning rate이 감소하는 조건은 매 epoch마다 평가를 통해 얻어지는 검증 loss를 통해 모델의 저장이 30epoch 동안 이뤄지지 않는다면, learning rate이 0.1씩 감소하도록 설정되었다.

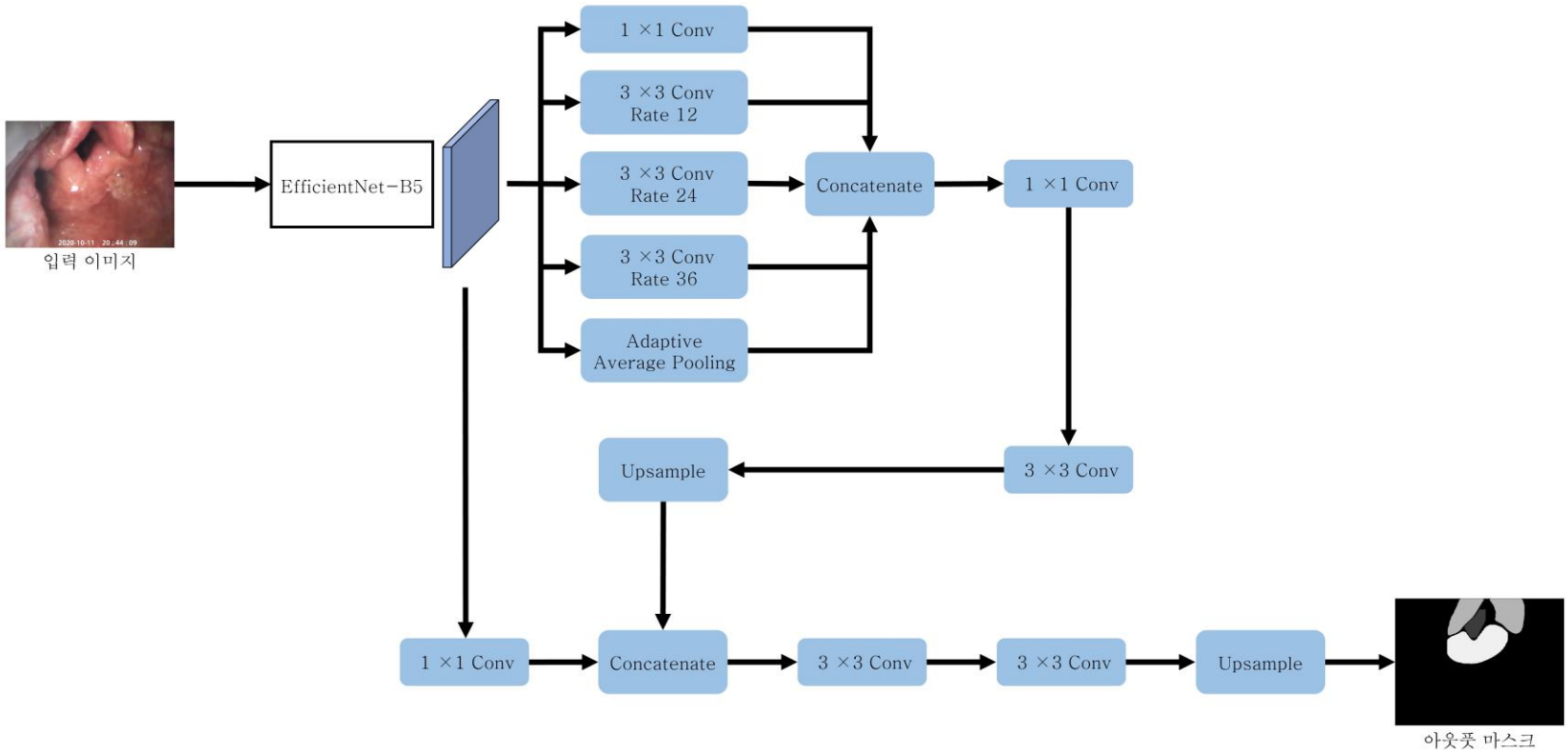
500epoch 동안 학습하도록 설정되지 않고 특정 조건을 만족하는 경우 학습이 종료되도록 설정하였다. 학습이 종료되도록 설정된 조건은

검증 loss를 통해 모델의 저장이 50epoch 동안 이뤄지지 않는 경우 학습이 종료되도록 설정되었다.

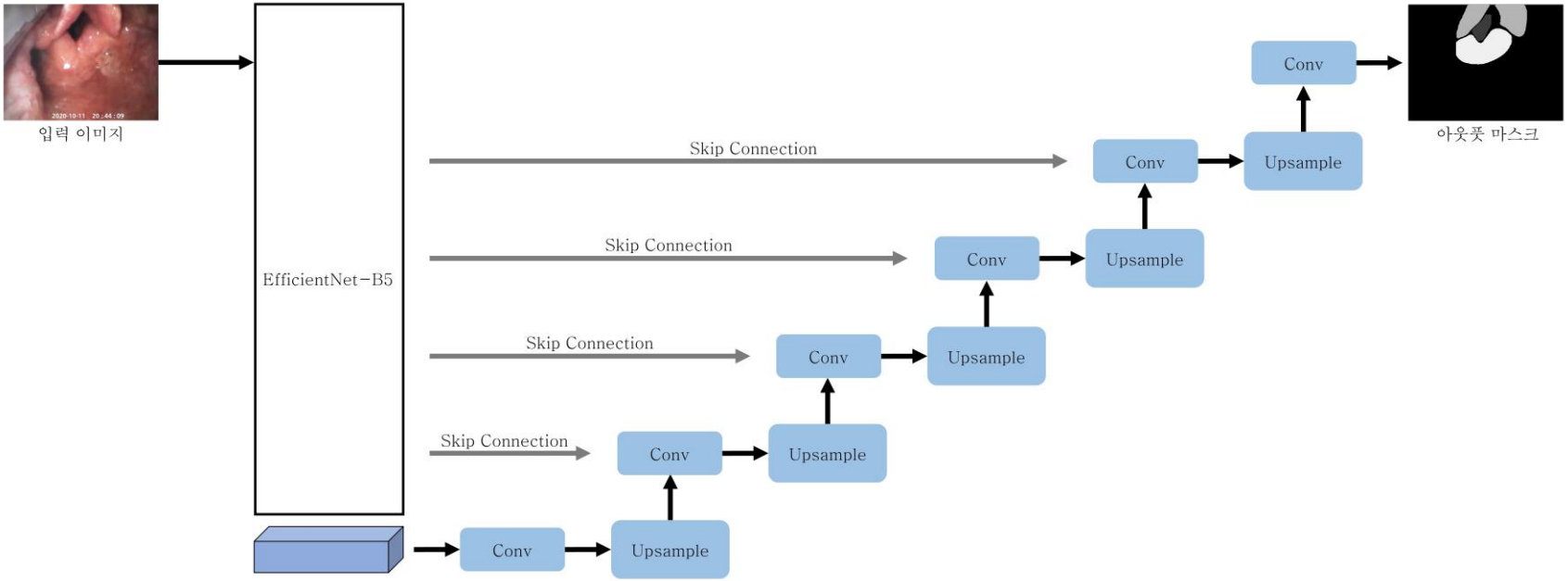
모델 학습 환경은 Ubuntu 20.04.3 LTS, AMD Ryzen 9 5900X, Nvidia RTX 3090 24GB, 그리고 RAM 48GB를 사용하였고 Python 3.7에 Pytorch 1.7.1.에서 학습되었다.

[표 2] EfficientNet 모델에 대한 입력 이미지의 해상도.

모델	이미지 해상도 (픽셀)
EfficientNet-B0	224 × 224
EfficientNet-B1	240 × 240
EfficientNet-B2	260 × 260
EfficientNet-B3	300 × 300
EfficientNet-B4	380 × 380
EfficientNet-B5	456 × 456
EfficientNet-B6	528 × 528
EfficientNet-B7	600 × 600



[그림 8] EfficientNet-B5&DeepLabv3+ 모델의 구조.



[그림 9] EfficientNet-B5&U-Net 모델의 구조.

[그림 10] Dice similarity coefficient loss에서 TP (true positive), FP (false positive), 그리고 FN (false negative)이 의미하는 영역을 나타낸 그림. 보라색은 ground truth 마스크의 영역, 노란색은 추론을 통해 얻어진 마스크, 그리고 하늘색은 ground truth 마스크와 추론을 통해 얻어진 마스크가 겹치는 영역을 의미함.

제 3 절 Mask R-CNN 모델 학습 과정

Mask R-CNN은 RPN을 통해 객체가 존재할 수 있을 수 있는 영역을 먼저 찾고, 이를 통해 얻어진 ROI를 통해 segmentation, classification, 그리고 detection을 진행하게 된다. 이를 통해 같은 클래스의 객체에 대해 서로 다른 객체로 나눠 detection과 segmentation을 진행할 수 있다. 하지만 segmentation을 진행하고자 하는 구조물 (성대, 후두 덮개, 연골, 그리고 혀)의 레이블링은 본 연구에서 구조물별로 하나의 마스크에 표현되었기 때문에 서로 다른 객체로 나눠 detection과 segmentation을 하는 것이 아닌 한 클래스에 대해 같은 객체로 segmentation을 진행한다.

본 연구에서는 사용된 Mask R-CNN 모델은 구조물 segmentation을 위해 backbone으로 ResNet-50-Feature Pyramid Network로 COCO로 pre-trained된 가중치를 이용했다. 입력 이미지는 0 ~ 1의 값을 갖도록 scaling이 진행된 후 정규화가 진행되었다.

Mask R-CNN모델의 hyper parameter로 optimizer function은 Adam, learning rate은 0.0001, batch size는 16, epoch은 500, 그리고 학습 셋의 loss function은 RPN loss와 ROI loss를 합한 loss를

사용하였다. 모델은 매 epoch마다 검증 셋을 이용해서 평가가 이뤄진다. 검증 셋을 평가하는 방법은 loss function과 같은 DSC loss를 이용해 평가가 진행되며, 평가를 통해 얻어진 검증 loss가 가장 낮을 때 모델의 저장을 진행했다.

검증 loss를 구하기 위해 DSC loss를 이용했다. 하지만 Mask R-CNN 모델은 클래스의 객체에 대해 서로 다른 객체로 나눠 detection과 segmentation이 가능하기 때문에 모델 학습 시 그림 11처럼 하나의 구조물에 대해 다른 객체로 모델이 판단하는 경우도 존재할 수 있다. 따라서 DSC loss를 구하기 위해 동일한 클래스에 대해 모델이 추론하여 얻어진 마스크들을 하나의 마스크로 만드는 과정을 추가하였다.

모델에서 입력 이미지에 대해 모델의 output은 객체가 존재하는 바운딩 박스, 객체의 클래스, 클래스 신뢰도 값, 그리고 객체의 마스크이다. 추론을 통해 얻어진 한 클래스에서의 여러 마스크는 아래의 과정을 통해 하나의 마스크로 만들어지는 과정을 거치고 DSC loss 값의 계산이 진행된다.

① 추론을 통해 얻어진 마스크와 클래스 신뢰도 값을 곱한다.
② ①의 단계에서 얻어진 마스크들의 픽셀 값을 비교해 가장 큰 픽셀 값을 새로운 마스크에 그 값으로 대체한다.
③ ②의 단계에서 얻어진 새로운 마스크와 대응하는 클래스의 ground truth 마스크를 통해 DSC loss를 계산한다.

이 과정은 그림 12를 통해 확인할 수 있다. 위의 과정 ①에서 마스크에 신뢰도 값을 곱하는 것은, 추론을 통해 얻어진 마스크의 픽셀 값은 추론을 통해 얻은 클래스에 대한 신뢰도 값을 의미하는 것이 아닌 마스크 내의 객체가 존재할 수 있는 것에 대한 신뢰도 값이기 때문이다.

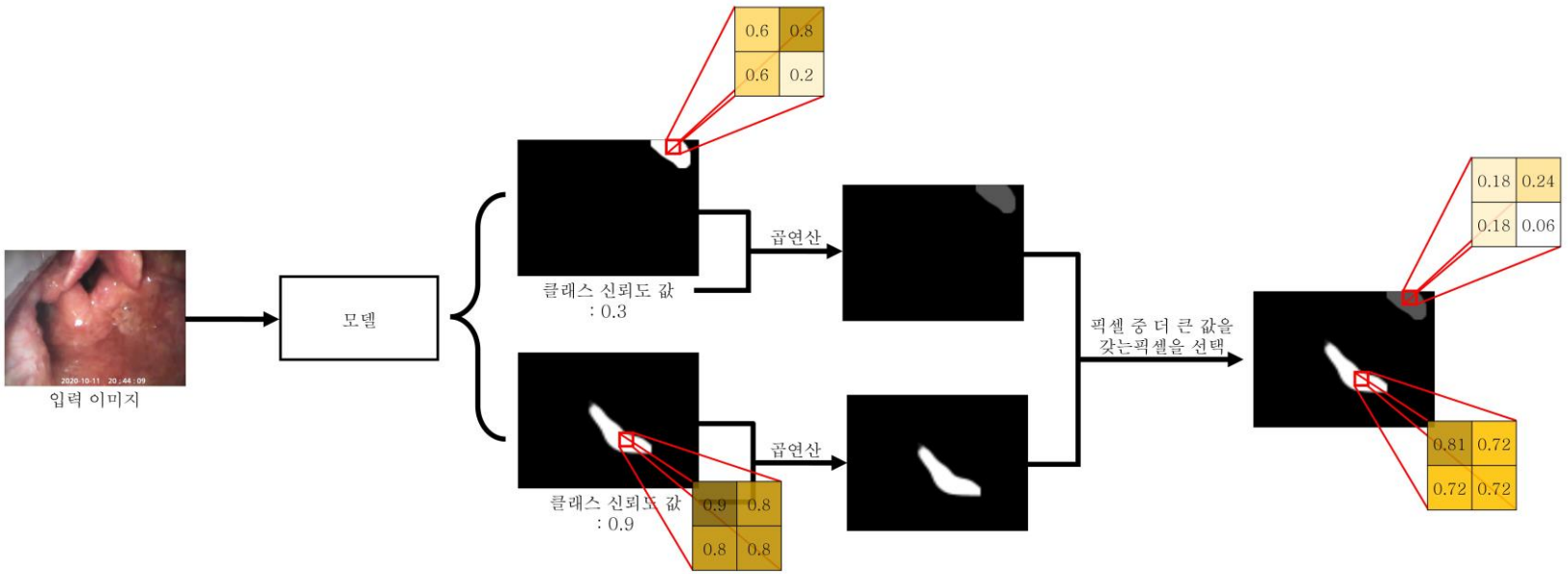
따라서 모델이 마스크에 대응하는 클래스에 대한 신뢰도 값 또한 검증 loss의 계산에 포함하기 위해 위와 같은 방식으로 하나의 마스크를 만들었다. 이렇게 학습된 Mask R-CNN 모델을 본 연구에서는 Configured Mask R-CNN이라 한다.

Learning rate은 epoch이 증가하면서 특정 조건을 달성하는 경우 감소하도록 만들었다. Learning rate이 감소하는 조건은 매 epoch마다 평가를 통해 얻어지는 검증 loss를 통해 모델의 저장이 30epoch 동안 이뤄지지 않는다면, learning rate이 0.1씩 감소하도록 설정되었다.

500epoch 동안 학습하도록 설정되지 않고 특정 조건을 만족하는 경우 학습이 종료되도록 설정하였다. 학습이 종료되도록 설정된 조건은 검증 loss를 통해 모델의 저장이 50epoch 동안 이뤄지지 않는 경우 학습이 종료되도록 설정되었다.

모델 학습 환경은 Ubuntu 20.04.3 LTS, AMD Ryzen 9 5900X, Nvidia RTX 3090 24GB, 그리고 RAM 48GB를 사용하였고 Python 3.7에 Pytorch 1.7.1.에서 학습되었다.

[그림 11] a와 b는 모델의 추론을 통해 나온 마스크 이미지, c는 모델의 입력 이미지. 동일한 구조물에 대해 여러 개의 마스크가 만들어질 수 있음. 파란색 영역은 후두 덮개를 의미함.



[그림 12] 학습 시 검증 셋을 통해 검증 loss를 구하는 과정에서 한 구조물에 대해 추론을 통해 얻어진 마스크들을 하나의 마스크로 만들고 이를 통해 검증 loss를 구하는 과정. 그림에 표시된 값은 실제 값을 의미하지 않음.

제 4 절 모델 성능 평가 방법

모델의 성능 평가를 위해 사용한 지표는 3가지이다. 첫번째 성능 지표는 DSC [30-32]를 사용해 추론을 통해 얻어진 마스크와 ground truth 마스크 간의 겹침의 정도를 계산해 평가를 진행하였다. DSC는 수식 (2)와 같이 계산된다. 수식 (2)에서의 TP, FP, 그리고 FN은 그림 10에서 표현되는 영역을 의미한다.

$$\text{Dice Similarity Coefficient (DSC)} = \frac{2TP}{2TP+FP+FN} \quad (2)$$

두번째 성능 지표는 detection 모델을 평가할 때 사용되는 평가방식을 사용하여 추론을 통해 얻어진 마스크로 구조물을 인식 가능한지 판단하였다. 세번째 성능 지표로 모델이 추론을 진행할 때의 FPS를 평가하였다.

두번째 성능 지표의 방법은 detection 모델을 평가할 때 사용할 때의 평가 지표로 detection 모델을 사용한 연구에서 추론을 통해 얻어진 바운딩 박스와 ground truth 바운딩 박스 간의 intersection over union (IoU)를 계산하여 IoU가 일정 값 이상이라면 추론을 통해 얻어진 바운딩 박스는 맞다고 한다. 이를 통해 본 연구에서는 추론을 통해 얻어진 마스크와 ground truth 마스크 간의 IoU가 특정 값 이상이라면 해당 추론을 통해 얻어진 마스크를 통해 구조물을 인식할 수 있다고 하였다. 기존 방식에서는 추론을 통해 얻어진 바운딩 박스와 ground truth 바운딩 박스를 통해 평가를 진행하지만 본 연구에서는 추론을 통해 얻어진 마스크와 ground truth 마스크를 이용해 평가를 진행한다.

한 클래스에 대해 추론을 통해 얻어진 마스크와 ground truth 마스크를 이용해 수식 (3)을 통해 IoU를 계산한다. 수식 (3)에서의 TP, FP, 그리고 FN은 그림 10에서 표현되는 영역을 의미한다.

$$\text{Intersection over Union (IoU)} = \frac{TP}{TP+FP+FN} \quad (3)$$

(TP:true positive, FP:false positive, FN:false negative)

아래의 조건에 따라 true positive (TP), false positive (FP), false negative (FN), 그리고 true negative (TN)을 구한다. [14, 33-35]

- True positive: 추론을 통해 얻어진 마스크와 ground truth 마스크의 IoU가 0.5 이상
- False positive: 추론을 통해 얻어진 마스크와 ground truth 마스크의 IoU가 0.5 미만 또는 ground truth 마스크는 없지만 추론을 통해 얻어진 마스크는 있는 경우
- False negative: ground truth 마스크는 있지만 추론을 통해 얻어진 마스크는 없는 경우
- True negative: ground truth 마스크와 추론을 통해 얻어진 마스크가 둘 다 없는 경우

위 조건을 통해 얻어진 TP, FP, FN, 그리고 TN을 통해 혼돈 행렬을 얻을 수 있고 이를 수식 (4) ~ (6)을 통해 accuracy, sensitivity, 그리고 specificity를 구할 수 있다.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (4)$$

$$\text{Sensitivity} = \frac{TP}{FN+TP} \quad (5)$$

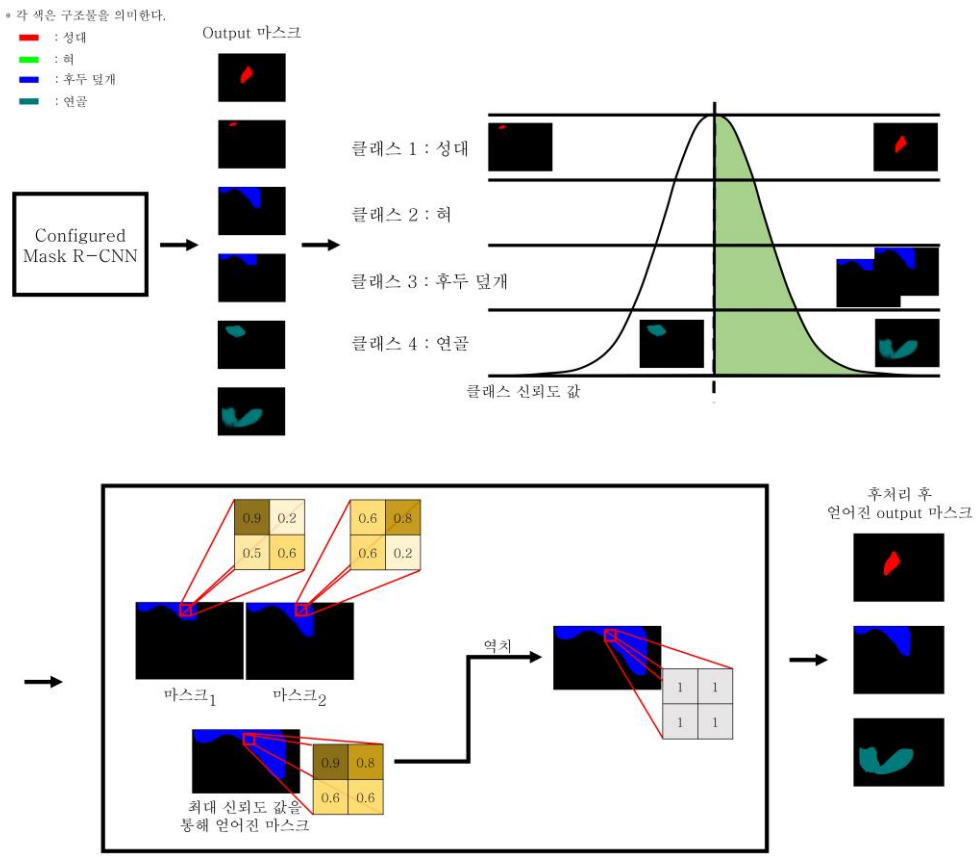
$$\text{Specificity} = \frac{TN}{FP+TN} \quad (6)$$

제 5 절 모델 성능 평가를 위한 Configured Mask R-CNN 모델의 후처리

앞의 3장 3절에서 Mask R-CNN 모델을 검증 셋으로 평가하여 모델을 저장하기 위해 추론을 통해 한 클래스에서 얻어지는 마스크들을 하나의 마스크로 만드는 과정을 진행했다. Configured Mask R-CNN 모델을 평가하기 위해 위와 비슷한 방법을 적용하여 각 클래스 별로 추론되어 나오는 마스크를 하나의 마스크로 만드는 과정을 진행했고 이 과정은 아래의 순서와 같다.

① 모델에서 추론을 통해 얻어지는 마스크를 각 클래스 별로 클래스 신뢰도 값에 따라 나눈다.
② 역치 값을 설정하여 클래스 신뢰도 값이 역차 값을 넘는 경우 마스크를 남기고 넘지 못하는 마스크는 버린다.
③ 각 클래스 별로 역치 값은 마스크에 대해 마스크끼리 각 픽셀 값을 비교하여 가장 큰 픽셀 값을 갖는 새로운 마스크를 만든다.
④ 만들어진 새로운 마스크에 역치 값을 적용하여 역치 값을 넘는 픽셀 값은 1로 그렇지 않은 픽셀 값은 0으로 처리한다.

위 과정을 통해 클래스 별로 한 개의 마스크를 만들 수 있다. 이 과정은 그림 13에서도 확인할 수 있다.



[그림 13] Configured Mask R-CNN 모델의 output 마스크를 후처리를 통해 클래스 별로 하나의 마스크로 만드는 과정. 그림에 표시된 값은 실제 값을 의미하지 않음.

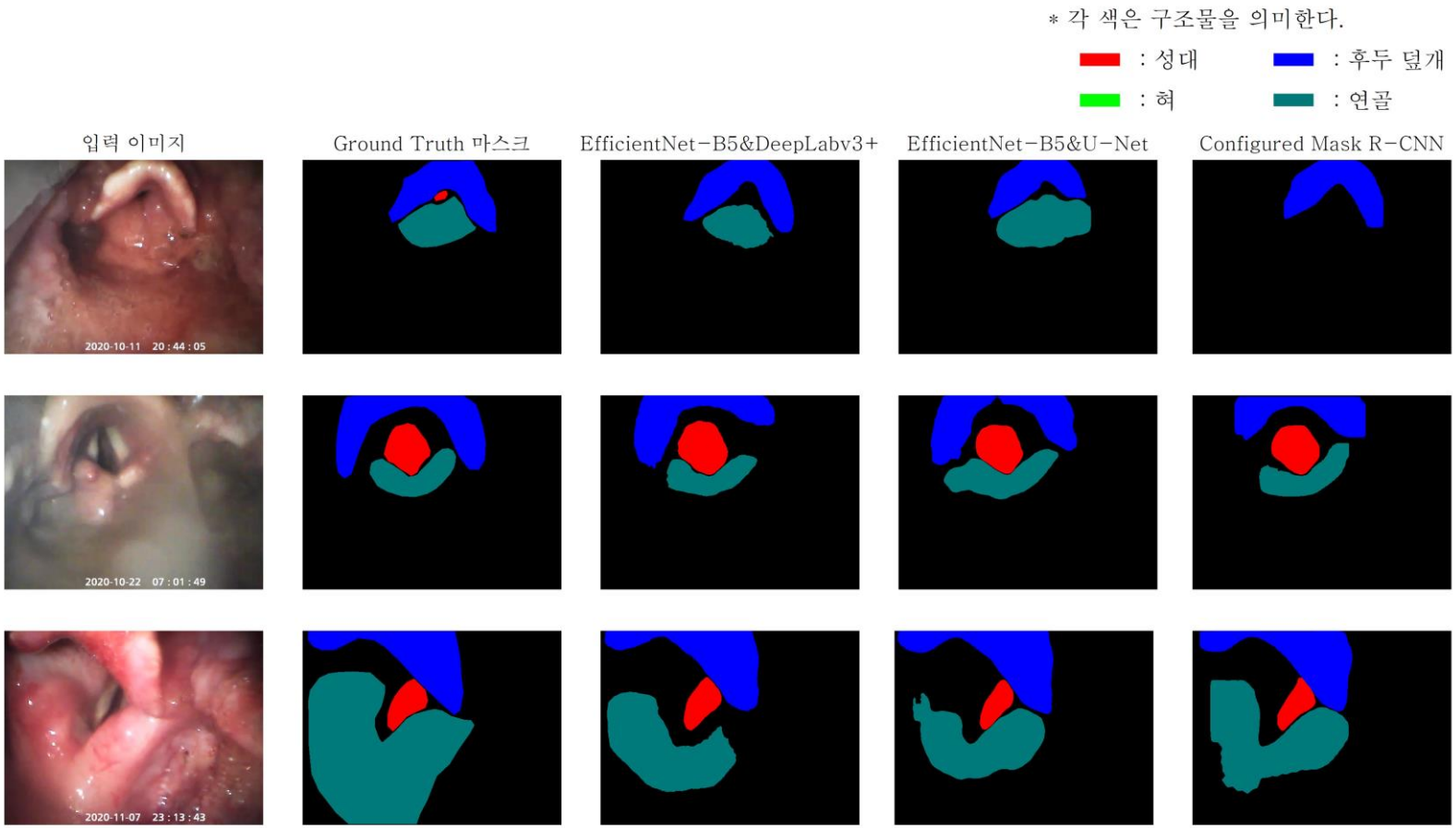
제 4 장 연구 결과

제 1 절 모델 성능 평가 결과

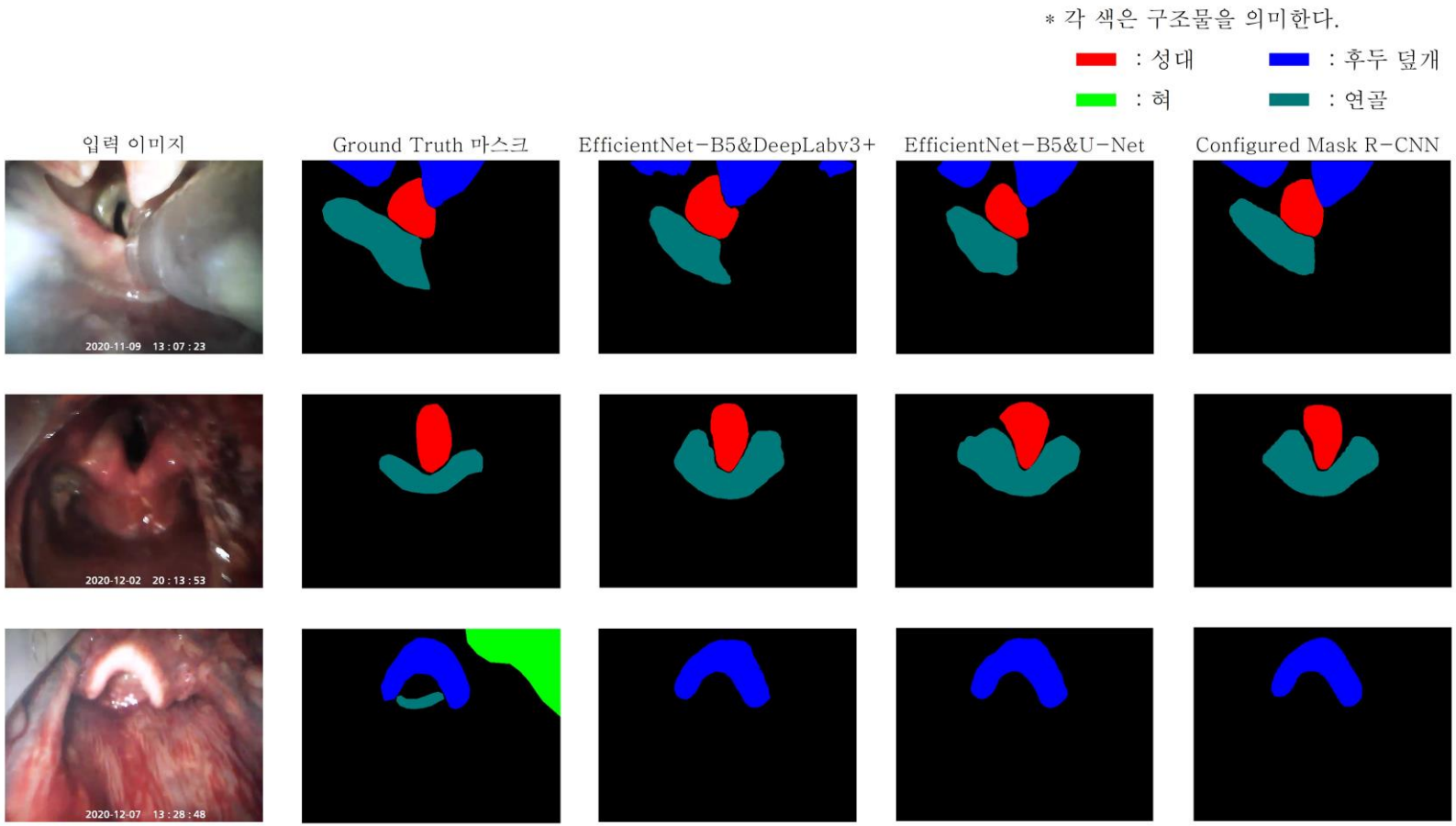
첫번째 모델 평가 방식은 DSC를 사용해 각 구조물 별 추론을 통해 얻어진 마스크와 ground truth 마스크 간의 겹침의 정도를 계산한 것으로 이의 결과는 표 3에서 확인할 수 있다. Configured Mask R-CNN 모델의 클래스 신뢰도 값과 마스크 신뢰도 값에 대한 역치 값은 0.5로 설정하였다. Configured Mask R-CNN, EfficientNet-B5&DeepLabv3+, 그리고 EfficientNet-B5&U-Net 모델의 output 마스크에 대한 역치 값은 0.5로 설정하였다. 성대 그리고 혀에 대해 높은 DSC를 보여주는 모델은 EfficientNet-B5&DeepLabv3+ 모델로 그 값은 0.766 그리고 0.3351이다. 후두 덮개에 대해 높은 DSC를 보여주는 모델은 Configured Mask R-CNN 모델로 그 값은 0.7677이다. 연골에 대해 높은 DSC를 보여주는 모델은 EfficientNet-B5&U-Net 모델로 그 값은 0.6906이다. 모델의 입력 이미지에 대한 추론을 통해 얻어지는 마스크는 그림 14와 15를 통해 확인할 수 있다.

[표 3] DSC 평가 지표를 이용해 모델의 성능을 비교한 결과.

모델	평가 지표	성대	후두 덮개	연골	혀
EfficientNet-B5&DeepLabv3+	DSC	0.766	0.7675	0.6539	0.3351
EfficientNet-B5&U-Net	DSC	0.7395	0.7581	0.6906	0.0
Configured Mask R-CNN	DSC	0.7207	0.7677	0.57	0.1167

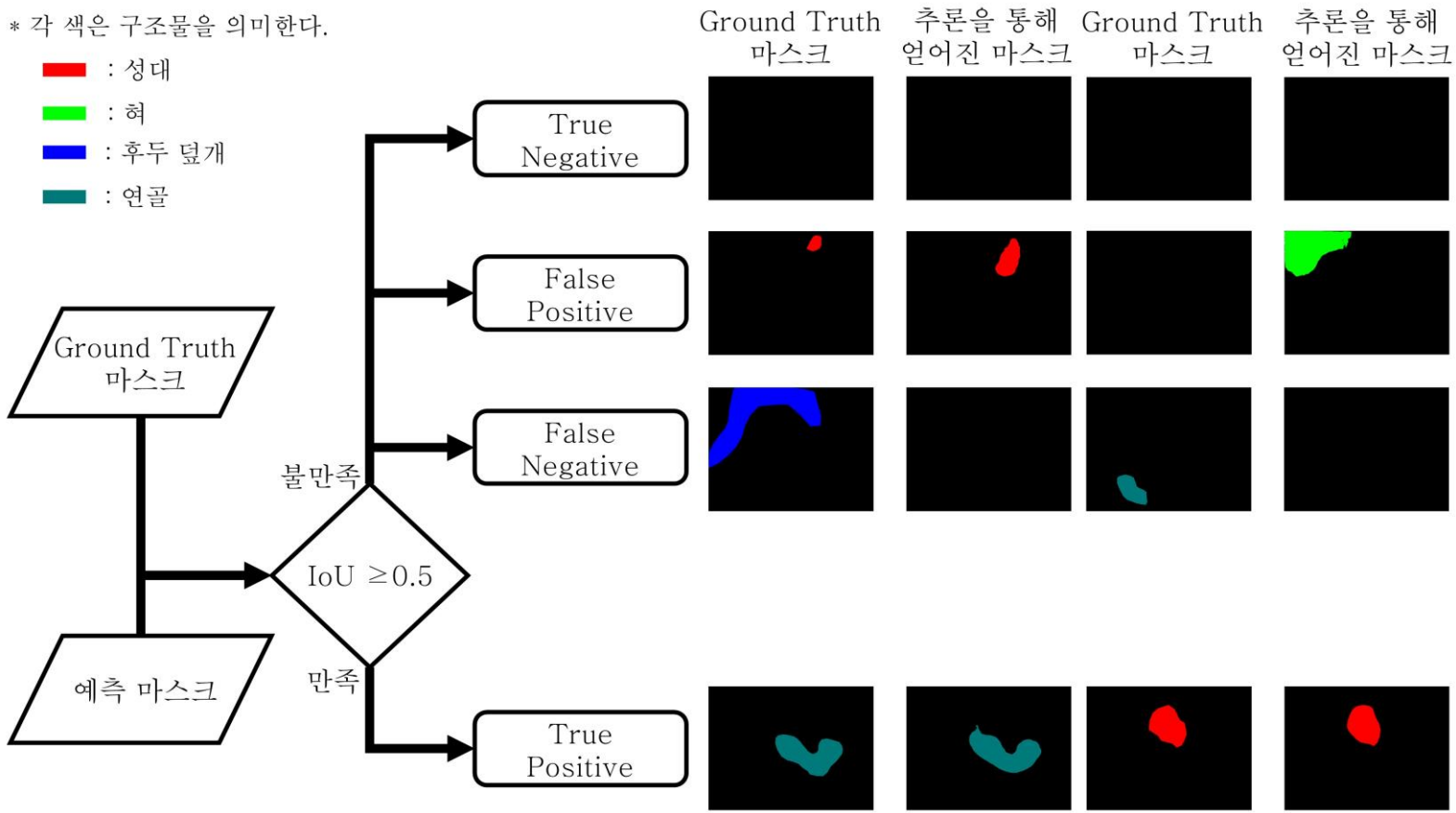


[그림 14] 입력 이미지에 대한 모델 별 추론을 통해 얻어진 마스크와 ground truth 마스크.

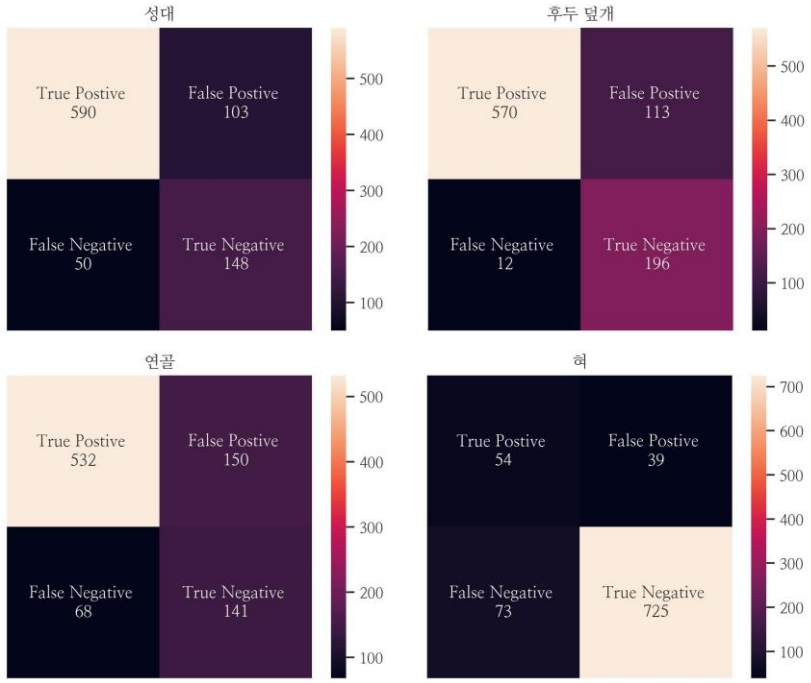


[그림 15] 입력 이미지에 대한 모델 별 추론을 통해 얻어진 마스크와 ground truth 마스크.

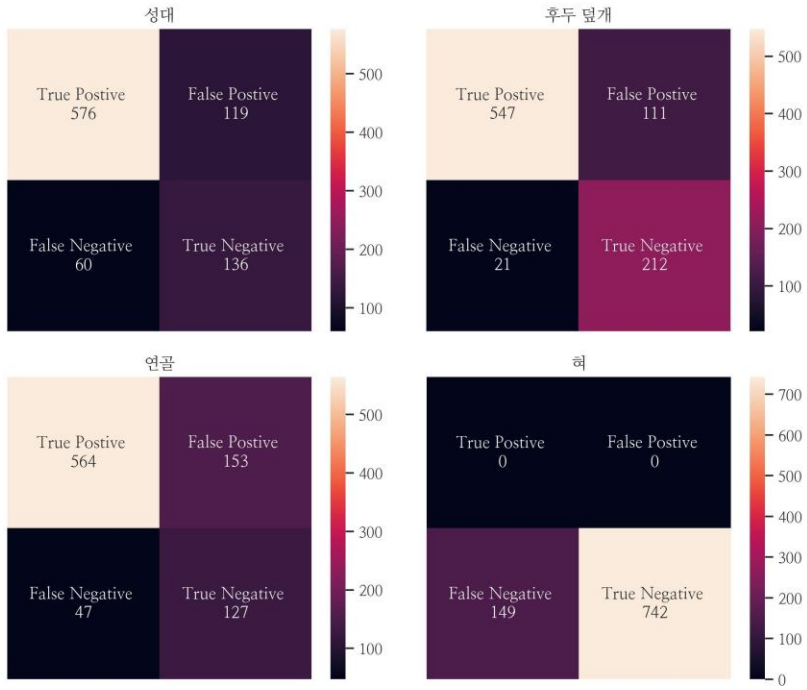
두번째 모델 평가 방식은 detection 모델을 평가할 때 사용되는 평가 지표를 도입하여 TP, FP, FN 그리고 TN을 구한 뒤 (그림 16) 혼돈 행렬 (그림 17-19)을 이용해 구조물 별 accuracy, sensitivity, 그리고 specificity를 구하였다 (표 4). Configured Mask R-CNN 모델의 클래스 신뢰도 값과 마스크 신뢰도 값에 대한 역치 값은 0.5로 설정하였다. Configured Mask R-CNN, EfficientNet-B5&DeepLabv3+, 그리고 EfficientNet-B5&U-Net 모델의 output 마스크에 대한 역치 값은 0.5로 설정하였다. 성대에 대해 EfficientNet-B5&DeepLabv3+ 모델이 0.8283과 0.9219으로 가장 높은 accuracy와 sensitivity를 보이고 Configured Mask R-CNN 모델이 0.8281로 가장 높은 specificity를 보였다. 후두 덮개에 대해 EfficientNet-B5&DeepLabv3+ 모델이 0.9794로 가장 높은 sensitivity를 보였고 Configured Mask R-CNN 모델이 0.862와 0.7517로 가장 높은 accuracy와 specificity를 보였다. 연골에 대해서는 EfficientNet-B5&U-Net 모델이 0.7755와 0.9231로 가장 높은 accuracy와 sensitivity를 보였고 Configured Mask R-CNN 모델이 0.7571로 가장 높은 specificity를 보였다. 혀에 대해서는 EfficientNet-B5&DeepLabv3+ 모델이 0.8743와 0.4252로 가장 높은 accuracy와 sensitivity를 보였고 EfficientNet-B5&U-Net 모델이 1.0으로 가장 높은 specificity를 보였다.



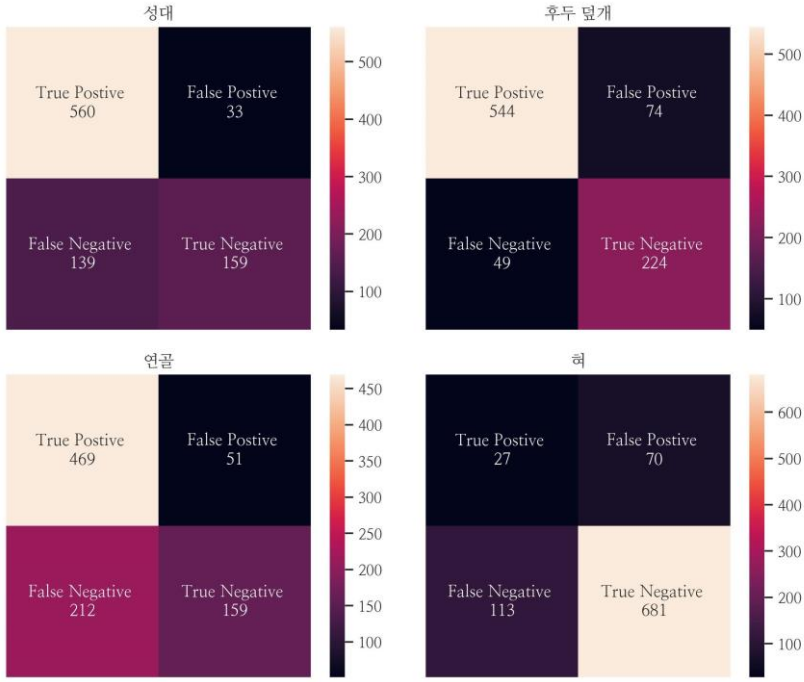
[그림 16] Detection 모델을 평가에 사용되는 방법을 도입하여 모델이 구조물을 인식할 수 있는 마스크를 만드는지를 확인하기 위한 방법을 나타낸 그림.



[그림 17] EfficientNet-B5&DeepLabv3+ 모델의 ground truth 마스크와 추론을 통해 얻어진 마스크 간의 관계를 통해 구해진 혼돈 행렬.



[그림 18] EfficientNet-B5&U-Net 모델의 ground truth 마스크와 추론을 통해 얻어진 마스크 간의 관계를 통해 구해진 혼돈 행렬.



[그림 19] Configured Mask R-CNN 모델의 ground truth 마스크와 추론을 통해 얻어진 마스크 간의 관계를 통해 구해진 혼돈 행렬.

[표 4] 모델의 구조물 별 accuracy, sensitivity, 그리고 specificity 결과.

모델	평가 지표	성대	후두 덮개	연골	혀
EfficientNet-B5&DeepLabv3+	Accuracy	0.8283	0.8597	0.7553	0.8743
	Sensitivity	0.9219	0.9794	0.8867	0.4252
	Specificity	0.5896	0.6343	0.4845	0.949
EfficientNet-B5&U-Net	Accuracy	0.7991	0.8519	0.7755	0.8328
	Sensitivity	0.9057	0.963	0.9231	0.0
	Specificity	0.5333	0.6563	0.4536	1.0
Configured Mask R-CNN	Accuracy	0.807	0.862	0.7048	0.7946
	Sensitivity	0.8011	0.9174	0.6887	0.1929
	Specificity	0.8281	0.7517	0.7571	0.9068

세번째 모델 평가 방식은 모델의 실시간 적용을 위한 모델 선정을 위해 추론 시의 FPS를 사용하였다. FPS는 데이터의 처리가 실시간으로 이뤄져야 하기 때문에 데이터 처리가 이뤄지는 시스템에 따라 결과가 달라진다. 시스템에 따른 FPS를 보기 위해 표 5에서와 같이 구성된 3개의 시스템을 테스트 셋을 이용해 추론 시 걸리는 시간을 통해 FPS 계산을 하여 성능 평가를 진행한다. 각 시스템에 대한 모델의 FPS는 표 6을 통해 확인할 수 있다. EfficientNet-B5&DeepLabv3+ 모델은 2번과 3번 시스템에서 GPU 메모리 부족으로 측정이 불가하였고 EfficientNet-B5&U-Net 모델은 3번 시스템에서 메모리 부족으로 측정이 불가하였다. 1번 시스템에서 초당 프레임 수는 Configured Mask R-CNN 모델, EfficientNet-B5&U-Net 그리고 EfficientNet-B5&DeepLabv3+ 순으로 각각 32FPS, 24FPS, 그리고 3FPS로 나타났다. 2번 시스템에서는 Configured Mask R-CNN 모델에서는 10FPS가 EfficientNet-B5&U-Net 모델에서는 12FPS를 보였으며 3번 시스템에서는 Configured Mask R-CNN 모델이 3FPS를 보였다.

[표 5] 초당 프레임 수 평가를 위한 시스템 구성.

시스템	종류	구성
1	데스크탑	• Ubuntu 20.04.3 LTS • AMD Ryzen 9 5900X • Nvidia RTX 3090 24GB, • RAM 48GB • Python 3.7 • Pytorch 1.7.1.
2	데스크탑	• Windows 10 Education • Intel i7-8700 • Nvidia RTX 2060 6GB, • RAM 32GB • Python 3.7 • Pytorch 1.7.1.
3	갤럭시 북 플렉스2 (NT950QDA-X72OB)	• Windows 11 Home • Intel i7-1165G7 • Nvidia GeForce MX450 • RAM 16GB • Python 3.7 • Pytorch 1.7.1.

[표 6] 모델의 테스트 셋에 대한 시스템 별 FPS 비교.

모델	평가 지표	시스템	
EfficientNet-B5&DeepLabv3+	FPS	1	3
		2	-
		3	-
EfficientNet-B5&U-Net	FPS	1	24
		2	12
		3	-
Configured Mask R-CNN	FPS	1	32
		2	10
		3	3

제 5 장 고 찰

제 1절 실험 결과 고찰

본 논문에서는 응급 상황에서 비디오 후두경을 이용해 촬영된 기관 내 삽관 영상을 이용해 구강 내 구조물을 segmentation하는 모델을 만들었다. 본 연구에서 사용된 데이터는 응급 상황에서의 환경을 반영하기 위해 수술 또는 진료 시에 보기 어려운 이물, 모션 블러, 빛 반사가 있는 경우의 데이터를 사용하였고, 이러한 경우에도 구조물의 segmentation이 이뤄질 수 있음을 확인하였다.

Configured Mask R-CNN 모델의 클래스 및 마스크 신뢰도 값의 역치 값에 따른 DSC 값의 변화와 EfficientNetB5&DeepLabv3+ 그리고 EfficientNetB5&U-Net 모델의 마스크 신뢰도 값의 역치 값에 따른 DSC 값의 변화는 그림 20 ~ 22에서 확인할 수 있다. Configured Mask R-CNN 모델의 역치 값에 따른 DSC 값은 성대, 후두 덮개, 그리고 연골 3개의 구조물에 대해 클래스 및 마스크 신뢰도 값에 대해 역치 값의 0.5로 설정되었을 때 가장 높은 값을 보였다. 혀의 경우 클래스 신뢰도에 대한 역치 값이 0.8일 때 마스크 신뢰도에 대한 역치 값이 0.5일 때 가장 높은 DSC 값을 보였다. Configured Mask R-CNN 모델의 클래스 및 마스크 신뢰도 값에 대해 역치 값을 모든 구조물에 대해 통일하기 위해 0.5로 설정하였다. EfficientNetB5&DeepLabv3+ 모델의 경우 구조물에 따라 DSC 값의 최대값이 서로 다르게 나타났다. 성대와 연골은 역치 값이 0.5일 때, 후두 덮개는 역치 값이 0.7일 때, 그리고 혀는 역치 값이 0.6일 때 DSC 값이 높게 나타났다. 4개의

구조물에 대해 역치 값 0.5가 2회로 많이 나타나 EfficientNetB5&DeepLabv3+ 모델의 마스크 신뢰도 값에 대한 역치 값을 0.5로 설정하였다. EfficientNet-B5&U-Net 모델의 역치 값에 따른 DSC 값은 역치 마스크 신뢰도 값에 대해 역치 값의 0.5로 설정되었을 때 혀를 제외한 모든 구조물에서 높은 값을 보였다. 혀 구조물의 경우 모든 역치 값에 대해 동일한 값을 나타냈기 때문에 EfficientNet-B5&U-Net 모델의 마스크 신뢰도 값에 대한 역치 값은 0.5로 설정하였다.

많은 연구들에서 DSC를 segmentation 모델의 성능 지표로 활용하지만 DSC를 통해서는 추론을 통해 얻어지는 마스크로 구조물을 인식할 수 있는지에 대해서는 확인하기 어렵다. 따라서 본 연구에서는 detection 모델에서 IoU를 이용해 TP, FP, FN, 그리고 TN을 나누는 방법을 도입하여 추론을 통해 얻어지는 마스크로 구조물을 인식할 수 있는지를 판단하였다. 다만 일반적인 detection 모델에서 IoU를 계산하기 위해 바운딩 박스를 사용하지만 본 연구에서는 마스크를 이용해 계산을 진행하였다. 바운딩 박스를 이용해 IoU를 계산한다면 구조물이 존재하지 않는 영역이 계산에 들어가게 되어 불필요한 영역에 대해서도 고려하게 되지만 마스크를 이용해 계산을 진행하게 된다면 제한된 영역에 대해 계산이 이뤄지기 때문에 구조물의 존재 유무에 대한 파악이 더 명확하게 된다.

표 3의 결과만을 통해 성대, 후두 덮개, 그리고 연골에 대해 모델의 성능을 본다면 EfficientNet-B5&DeepLabv3+이 구조물을 가장 잘 segmentation했다고 판단할 수 있다. 하지만 표 4의 specificity 결과를 본다면 EfficientNet-B5&DeepLabv3+과 EfficientNet-B5&U-Net 모델이 성대, 후두 덮개, 그리고 연골에 대해 Configured Mask R-CNN 모델보다 더 많은 FP를 갖는 것을 확인할 수 있다. 다만 Configured

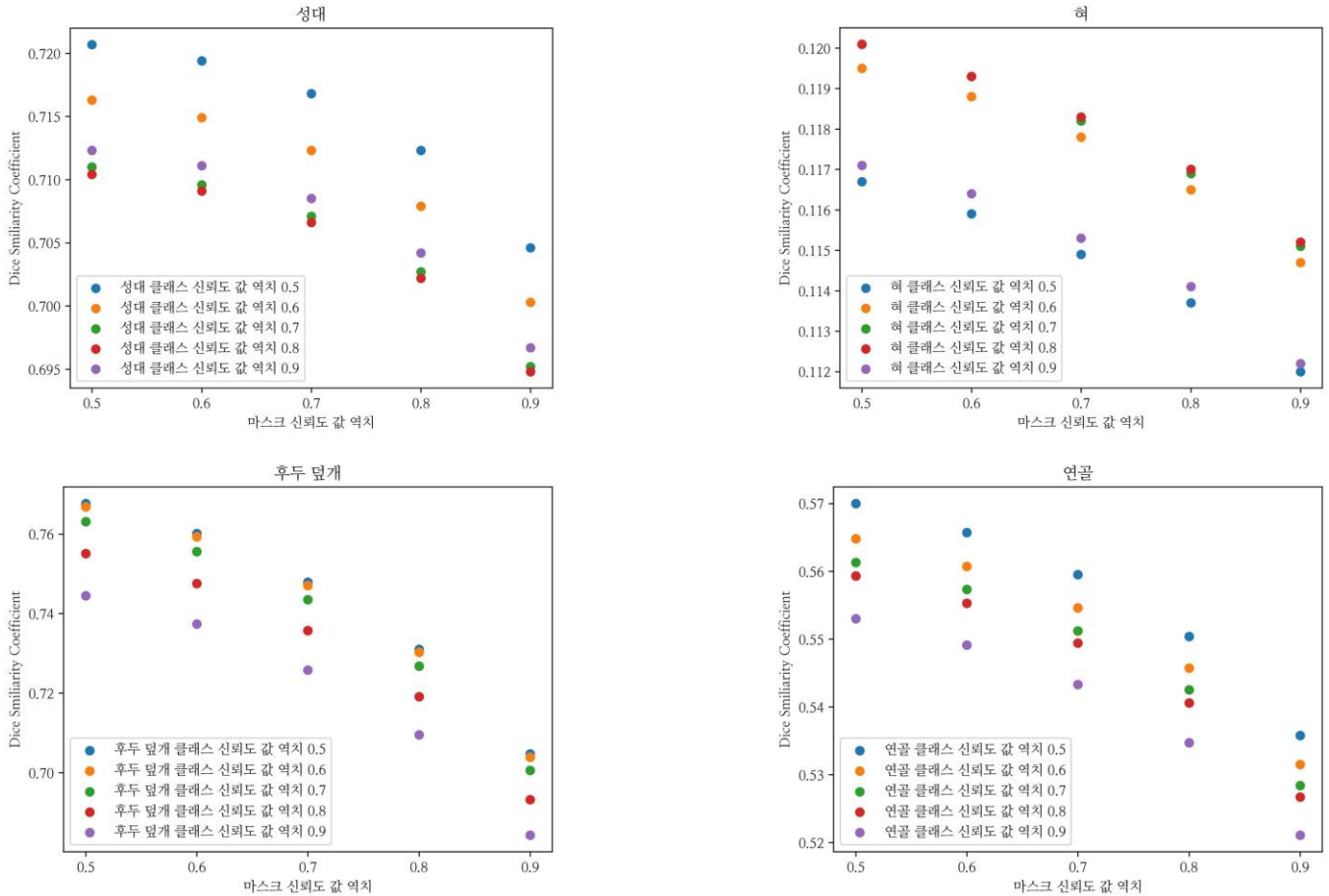
Mask R-CNN 모델의 경우 성대, 후두 덮개, 그리고 연골에 대한 sensitivity가 낮은 것을 통해 FN가 더 많다는 것을 확인할 수 있다. 이는 EfficientNet-B5&DeepLabv3+과 EfficientNet-B5&U-Net 모델은 이미지에 대해 모든 픽셀을 segmentation하는 반면에 Configured Mask R-CNN 모델은 RPN을 통해 구조물의 존재 유무 및 영역을 찾기 때문에 FP가 더 적게 나타낸다고 판단된다. 이를 통해 사용할 모델을 선정하는데 있어 accuracy, sensitivity, 그리고 specificity가 고려된다면 모델을 선정하는 데에 있어 선택의 폭이 넓어질 수 있을 것으로 생각된다.

혀 구조물의 경우 다른 구조물들에 비해 DSC 값이 낮게 나타나는 것을 확인할 수 있다. 이는 혀에 점막 또는 혈액과 같은 이물질 및 빛 반사가 존재하고 혀의 앞면과 뒷면에 따라 해부학적 구조가 다르기 때문이다. 이는 그림 23에서 확인할 수 있다. 이렇게 서로 다른 특성을 보임에도 같은 혀로 레이블링이 진행되었다. 또한 혀의 레이블링 된 데이터의 수가 다른 구조물에 비해 적은데 그 이유는 일반적으로 기관 내 삽관 시행 시 후두경의 날로 혀를 옆으로 치워야 성대가 노출되므로 혀는 삽관 초반에만 짧게 등장하게 된다. 이와 같은 이유로 혀의 DSC가 다른 구조물인 성대, 후두 덮개, 그리고 연골 보다 낮은 DSC를 보이는 것으로 판단된다.

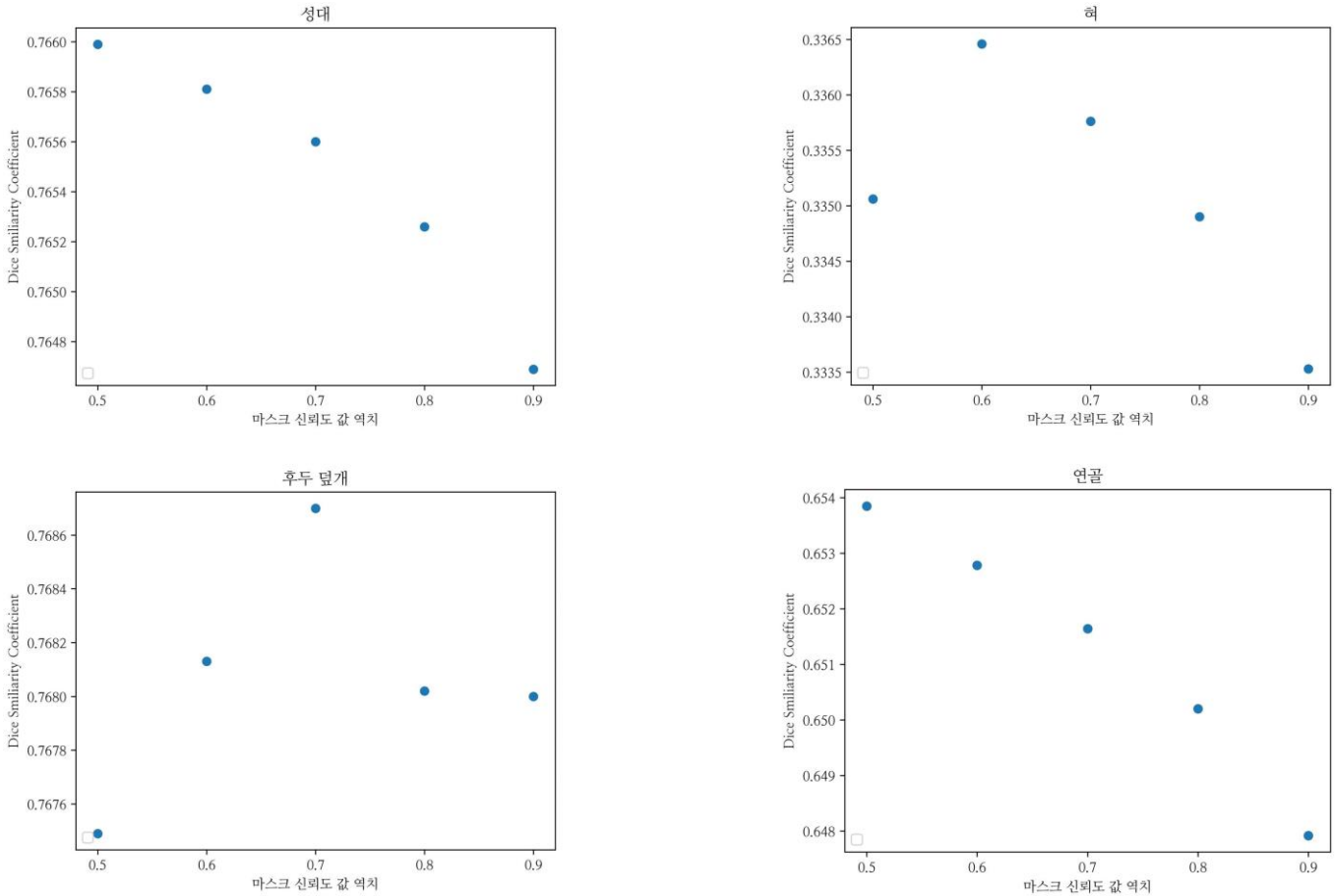
본 연구에서는 모델의 성능 지표로 사용되는 시스템에 따른 모델의 FPS를 비교하였다. 이는 기관 내 삽관이 이미 완료된 이후의 영상에서 segmentation을 진행하는 것이 아닌 실시간으로 영상에 모델을 적용하여 segmentation을 진행하려는 경우, 사용하는 시스템에 따른 모델의 실시간 동작 여부를 비교하기 위해 진행되었다. 표 5와 표 6을 통해 알 수 있듯이 사용하는 시스템에 따라 모델이 동작하지 않을 수도 있으며, 모델이 동작하더라도 실시간으로 처리가 불가능할 수 있다.

따라서 실시간으로 모델을 적용하여 데이터를 처리하기 위해서는 어떤 시스템을 사용할 것인지에 대해 고려해야 한다.

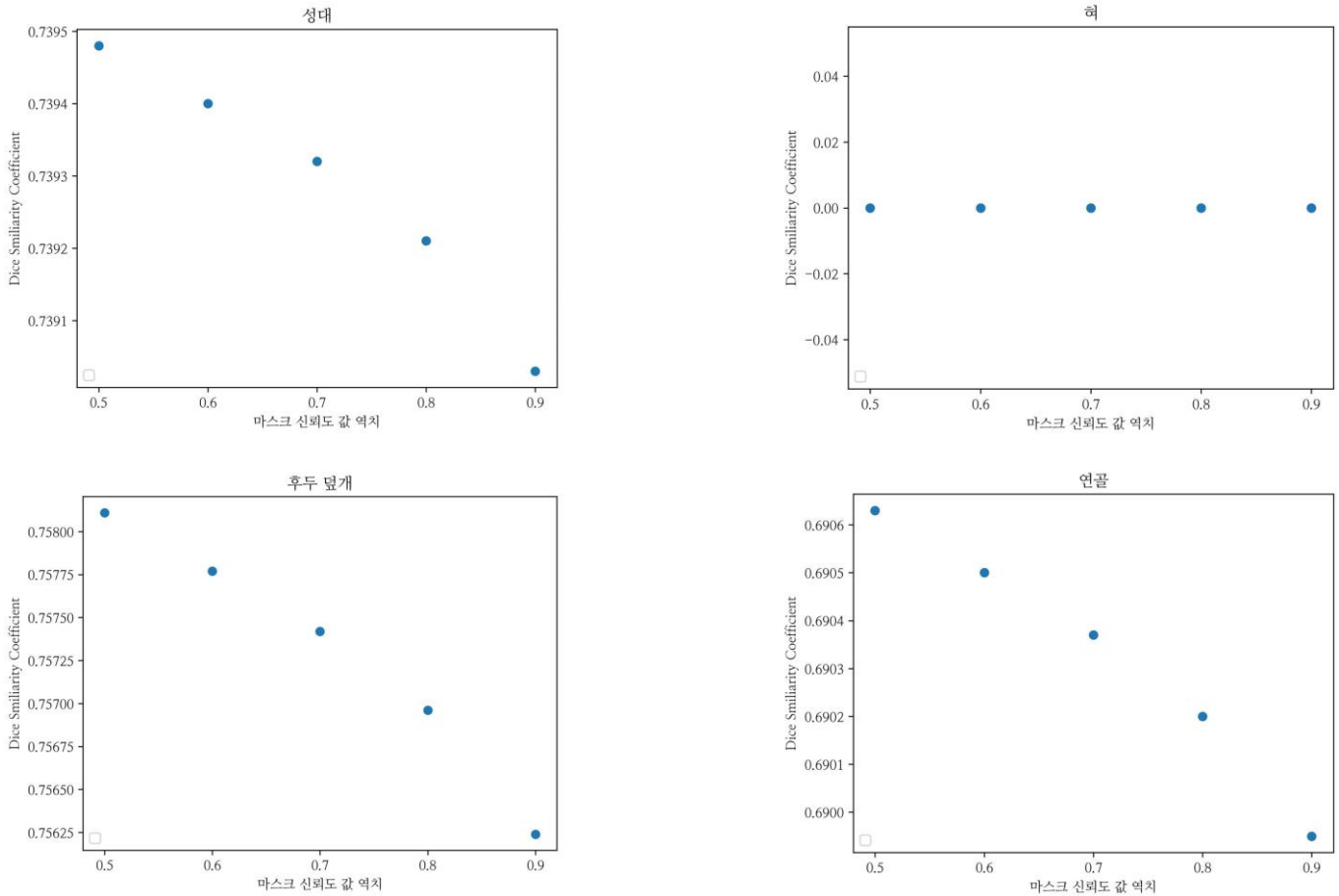
본 연구에서 사용된 영상의 경우 초당 프레임 수가 30이기 때문에 실시간으로 segmentation을 진행하기 위해서는 모델이 추론 시의 FPS는 30이상을 보여주어야 한다. 영상을 처리하여 정지 영상으로 만들고 이를 시스템에 전달하여 모델이 segmentation을 진행하고 다시 이를 전달하는 과정에서 시스템에 걸리는 부하 및 시간을 고려하지 않는다면, 연구에 사용된 시스템 중에서 이를 만족하는 모델은 Mask R-CNN 모델이다.



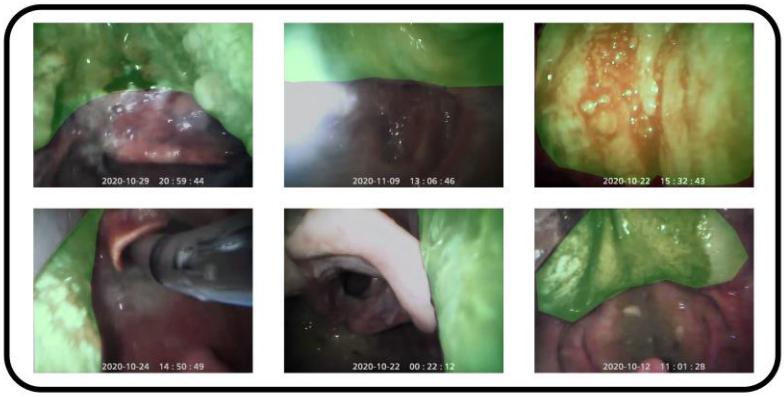
[그림 20] Configured Mask R-CNN 모델의 클래스/마스크 신뢰도 값의 역치에 따른 dice similarity coefficient 변화를 나타낸 그림.



[그림 21] EfficientNetB5&DeepLabv3+모델의 마스크 신뢰도 값의 역치에 따른 dice similarity coefficient 변화를 나타낸 그림.



[그림 22] EfficientNetB5&U-Net모델의 마스크 신뢰도 값의 역치에 따른 dice similarity coefficient 변화를 나타낸 그림.

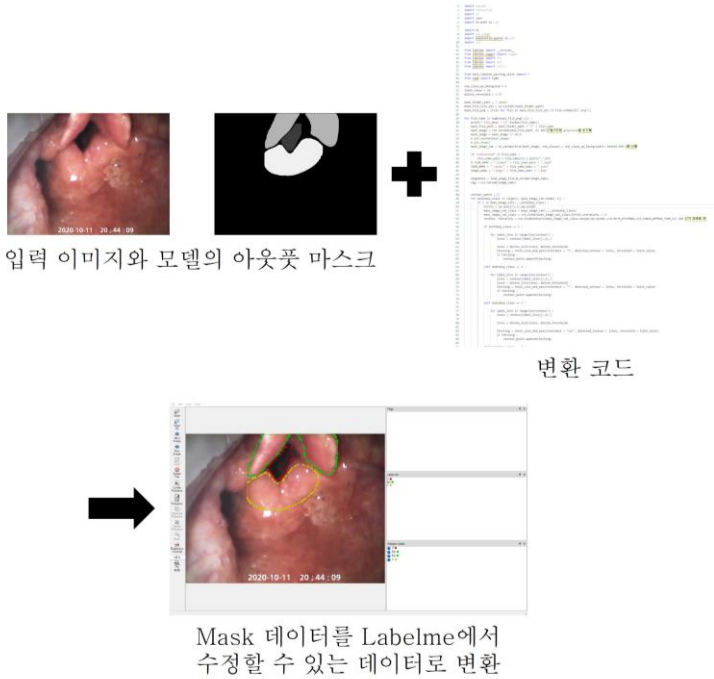


[그림 23] 혀의 앞과 뒷면 그리고 혀에 점막 또는 혈액과 같은 이물질 및 빛 반사가 존재하는 경우에 대한 예시 이미지. 초록색 부분은 혀를 나타냄.

제 2절 한계점 및 발전 방향

본 연구에는 몇 가지 제한점이 있다. 첫째, 동영상에서 추출된 전체 이미지를 사용하지 않고 특정 구간 내의 이미지를 사용했다는 것이다. 이는 모델이 동영상 전체 시간에 대해 segmentation을 진행하는 경우 체외의 물체에 대해 구조물이라고 인식할 수 있다. 따라서 추후 모델을 개발할 때, 본 연구에서 설정한 구간 이외의 이미지도 모델에 추가하여 학습을 진행하거나 모델에 데이터를 입력하기 전에 classification 모델을 추가해 segmentation 모델에 입력할 데이터를 선별할 수 있도록 할 수 있을 것이다. 둘째, 응급 상황에서의 촬영된 데이터는 CPR을 진행하면서 촬영한 동영상이 포함되어 있어서 모션 블러가 포함된 이미지가 존재한다. 실시간으로 기관 내 삽관 영상에 적용한다면 모션 블러가 존재하는 이미지 또한 segmentation이 필요하다. 따라서 본 연구에서는 모션 블러가 존재하는 이미지를 제외하지 않고 모델 개발에 사용하였다. 추후 연구에서는 데이터의 생성 [36] 또는 이미지에 존재하는 노이즈를 제거하는데 사용되기도 하는 Generative Adversarial Network (GAN) [37]을 이용하여 모션 블러가 적거나

없는 이미지에 적용하여 모션 블러가 존재하는 이미지를 만들어 robust한 모델을 만드는데 활용할 수 있을 것이다. 또는 모션 블러가 존재하는 이미지를 classification하고 분류된 이미지에 GAN을 적용하여 모션 블러를 줄일 수 있을 것이다. 셋째, 데이터 레이블링은 시간이 많이 소요되는 작업이다. 모델의 추론 과정을 통해 얻어지는 마스크를 모델의 개발에 사용할 수 있다면 레이블링에 소요되는 시간 및 작업자의 피로도를 줄 일 수 있을 것이다. 추후 연구에서는 본 연구에서는 활용되지 않았지만, 그림 24에서 확인할 수 있듯이 마스크 파일을 Labelme에서 읽을 수 있는 JSON 파일로 변환하는 코드를 개발하였다. 이를 활용하여 모델을 통해 얻어진 데이터를 수정 및 개선하여 레이블링을 진행하는 시간 및 작업자의 피로도를 줄임으로서 데이터 수집을 용이하게 할 수 있을 것이다.



[그림 24] 모델의 추론을 통해 얻은 마스크 파일을 Labelme가 읽을 수 있는 JSON 파일로 변환할 수 있도록 만듦.

제 6 장 결 론

본 연구는 기관 내 삽관 자동화 시스템 개발을 위한 초석으로 실제 상황에서 촬영된 데이터에서 구조물을 인식하여 segmentation하는 알고리즘의 개발을 위해 시작되었다. 성대, 후두 덮개, 연골, 그리고 혀의 구조물 segmentation을 위해 Mask R-CNN, DeepLabv3+ 그리고 U-Net 모델을 사용했다. DeepLabv3+ 그리고 U-Net 모델의 backbone으로 EfficientNet-B5를 사용한 DeepLabv3+ 그리고 U-Net 모델을 본 논문에서는 EfficientNet-B5&DeepLabv3+ 그리고 EfficientNet-B5&U-Net이라 한다. Mask R-CNN 모델의 경우 추론 과정에서 하나의 클래스에 대해 여러 개의 output 마스크가 생성될 수 있다. 학습과정에서 검증 셋을 이용해 모델의 검증을 진행하는 과정에서 하나의 클래스에 대해 추론을 통해 얻어진 여러 개의 마스크를 하나의 마스크로 만드는 과정을 진행하였고, 이를 통해 학습된 모델을 Configured Mask R-CNN이라 한다. 학습된 Configured Mask R-CNN은 하나의 클래스에 대해 추론을 통해 얻어지는 여러 개의 마스크를 하나의 마스크로 만드는 후처리 과정이 추가로 진행되었다. 모델의 평가를 위해 DSC를 이용하였고, 추론을 통해 얻어진 마스크로 구조물을 인식할 수 있는지 보기 위해 detection 모델에서 사용하는 평가 방법을 적용했다. 그리고 시스템에 따른 모델 별 추론 시 FPS를 비교하여 모델을 실시간으로 활용할 수 있는 모델을 확인하였고 이를 통해 실시간 활용을 위한 시스템을 확인할 수 있었다. 본 연구를 통해 기존에 진행된 연구들과 다르게 실제 응급 상황에서 진행된 기관 내 삽관 데이터를 이용해 구조물의 segmentation이 진행될 수 있음을

확인하였다. 추후 자동화 시스템을 위한 구조물 인식 알고리즘으로 본 연구가 활용될 수 있을 것이다. 그리고 개발된 딥러닝 모델을 응급 상황에서 촬영된 영상에 적용하여 구조물이 segmentation된 영상을 얻을 수 있다. 구조물이 segmentation된 영상을 통해 경험이 적은 의료 종사자들이 구조물에 대한 이해를 높일 수 있을 것이라 생각된다.

참고 문헌

[1] J. Peters *et al.*, "First-pass intubation success rate during rapid sequence induction of prehospital anaesthesia by physicians versus paramedics," *European Journal of Emergency Medicine,* vol. 22, no. 6, pp. 391-394, 2015.

[2] W. E. Hurford, "Techniques for endotracheal intubation," *International Anesthesiology Clinics,* vol. 38, no. 3, pp. 1-28, 2000.

[3] C. Matava, E. Pankiv, S. Raisbeck, M. Caldeira, and F. Alam, "A convolutional neural network for real time classification, identification, and labelling of vocal cord and tracheal using laryngoscopy and bronchoscopy video," *Journal of medical systems,* vol. 44, no. 2, pp. 1-10, 2020.

[4] R. M. Levitan, J. W. Heitz, M. Sweeney, and R. M. Cooper, "The complexities of tracheal intubation with direct laryngoscopy and alternative intubation devices," *Annals of Emergency Medicine,* vol. 57, no. 3, pp. 240-247, 2011.

[5] J.-B. Paolini, F. Donati, and P. Drolet, "video-laryngoscopy: another tool for difficult intubation or a new paradigm in airway management?," *Canadian Journal of Anesthesia/Journal canadien d'anesthésie,* vol. 60, no. 2, pp. 184-191, 2013.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234-241, 2015.

[7] H. Ding, Q. Cen, X. Si, Z. Pan, and X. Chen, "Automatic glottis segmentation for laryngeal endoscopic images based on U-Net," *Biomedical Signal Processing and Control,* vol. 71, p. 103116, 2022.

[8] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems,* vol. 30, 2017.

[9] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929,* 2020.

[10] X. Pan, W. Bai, M. Ma, and S. Zhang, "RANT: A cascade reverse attention segmentation framework with hybrid transformer for laryngeal endoscope images," *Biomedical Signal Processing and Control,* vol. 78, p. 103890, 2022.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961-2969, 2017.

[12] C.-L. Chin, C.-L. Chang, Y.-C. Liu, and Y.-L. Lin, "AUTOMATIC SEGMENTATION AND INDICATORS MEASUREMENT OF THE VOCAL FOLDS AND GLOTTAL IN LARYNGEAL ENDOSCOPY IMAGES USING MASK R-CNN," *Biomedical Engineering: Applications, Basis and Communications,* vol. 33, no. 04, p. 2150027, 2021.

[13] J. Ren *et al.*, "Automatic recognition of laryngoscopic images using a deep-learning technique," *The Laryngoscope,* vol. 130, no. 11, pp.

E686-E693, 2020.
[14] M. A. Azam *et al.*, "Deep Learning Applied to White Light and Narrow Band Imaging Videolaryngoscopy: Toward Real-Time Laryngeal Cancer Detection," *The Laryngoscope,* vol. 132, no. 9, pp. 1798-1806, 2022.
[15] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767,* 2018.
[16] G. Hoyer, S. Runnels, and S. Merchant, "Advanced Video Laryngoscope and Automatic Data Collection System," 2020.
[17] H. Xiong *et al.*, "Computer-aided diagnosis of laryngeal cancer via deep learning based on laryngoscopic images," *EBioMedicine,* vol. 48, pp. 92-99, 2019.
[18] P. He, R. Jain, J. Chambost, C. Jacques, and C. Hickman, "Semantic Video Segmentation for Intracytoplasmic Sperm Injection Procedures," *arXiv preprint arXiv:2101.01207,* 2021.
[19] J. Born *et al.*, "Accelerating detection of lung pathologies with explainable ultrasound image analysis," *Applied Sciences,* vol. 11, no. 2, p. 672, 2021.
[20] M. Grammatikopoulou *et al.*, "CaDIS: Cataract dataset for surgical RGB-image segmentation," *Medical Image Analysis,* vol. 71, p. 102053, 2021.
[21] D. Liang *et al.*, "Coronary angiography video segmentation method for assisting cardiovascular disease interventional treatment," *BMC medical imaging,* vol. 20, no. 1, pp. 1-8, 2020.
[22] H. Yao, R. W. Stidham, Z. Gao, J. Gryak, and K. Najarian, "Motion-based camera localization system in colonoscopy videos," *Medical Image Analysis,* vol. 73, p. 102180, 2021.
[23] *VirtualDub*. (1.10.4), [Online], Available: https://www.virtualdub.org
[24] *Labelme: Image Polygonal Annotation with Python*. (4.5.13), Github repository. [Online]. Available: https://github.com/wkentaro/labelme
[25] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801-818, 2018.
[26] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440-1448, 2015.
[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems,* vol. 28, 2015.
[28] *Segmentation Models Pytorch*. (0.3.0), Github repository. [Online]. Available: https://github.com/qubvel/segmentation_models.pytorch
[29] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105-6114, 2019.
[30] G. Luo, Q. Yang, T. Chen, T. Zheng, W. Xie, and H. Sun, "An optimized two-stage cascaded deep neural network for adrenal segmentation on CT images," *Computers in Biology and Medicine,* vol. 136, p. 104749, 2021.

[31] J. Chen *et al.*, "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306,* 2021.
[32] K.-H. Uhm *et al.*, "Deep learning for end-to-end kidney cancer diagnosis on multi-phase abdominal computed tomography," *npj Precision Oncology,* vol. 5, no. 1, p. 54, 2021.
[33] A. Nogueira-Rodríguez *et al.*, "Real-time polyp detection model using convolutional neural networks," *Neural Computing and Applications,* vol. 34, no. 13, pp. 10375-10396, 2022.
[34] Y. Li, "Detecting Lesion Bounding Ellipses with Gaussian Proposal Networks," *Cham: Springer International Publishing, in Machine Learning in Medical Imaging*, pp. 337-344, 2019.
[35] M. Zlocha, Q. Dou, and B. Glocker, "Improving RetinaNet for CT lesion detection with dense masks from weak RECIST labels," in *International conference on medical image computing and computer-assisted intervention*, pp. 402-410, 2019.
[36] D. Yoon *et al.*, "Colonoscopic image synthesis with generative adversarial network for enhanced detection of sessile serrated lesions using convolutional neural network," *Scientific Reports,* vol. 12, no. 1, p. 261, 2022.
[37] I. Goodfellow *et al.*, "Generative adversarial networks," *Communications of the ACM,* vol. 63, no. 11, pp. 139-144, 2020.

Abstract

A Study on the Segmentation of Anatomical Structure in Emergent Endotracheal Intubation Using Deep-Learning Algorithm

Seung Jae Choi
Interdisciplinary Program in Bioengineering
The Graduate School
Seoul National University

This study is about segmenting vocal cord, epiglottis, corniculate cartilage, and tongue by deep learning from the data acquired from emergency department using video laryngoscope in Endotracheal Intubation (ETI) process.

This study was conducted using Mask R-CNN, DeepLabv3+, and U-Net models for segmentation. The Mask R-CNN model can generate multiple masks for each structure through inference. As a result of the model, several masks are made for each structure. For vocal cord, epiglottis, corniculate cartilage, and tongue in the oral cavity, one mask was labeled for each structure. Therefore, masks for each structure are necessary to be made into one mask.

In this paper, several masks for each structure, output of the Mask R-CNN model, were made into a single mask respectively. The performance of the model was verified using same evaluation

methods for all three models. Dice similarity coefficient, method used to evaluate detection model and frames per second were used as an evaluation method.

The ETI images used in this study, include foreign objects around the airway, motion blur, and light reflection were used to reflect the actual endotracheal intubation environment.

Through this study, it was confirmed that the data reflecting the actual situation could be segmented using deep learning and found the model that could be used in real-time inferencing. As the first study using data taken in emergency, the algorithm developed in this study can be applied to the actual video to obtain video which the structures are segmented. From the videos, which the structures were segmented, it is thought that less experienced medical workers can improve their understanding of structures in the oral cavity and can be used as a cornerstone for establishing a remote intubation assistance system and developing an automatic intubation system.

Keywords : Deep Learning, Endotracheal Intubation, Image Segmentation, Image Processing
Student Number : 2021-27700

감사의 글

항상 곁에서 저를 지원해주시고 지지해주시는 부모님과 가족들에게 감사 인사를 드립니다. 옳은 방향으로 나아갈 수 있도록 조언을 해주시고 힘들 때에도 버틸 수 있도록 곁에 있어 주셨습니다. 가족들이 주신 은혜에 다시금 감사드리며, 자랑스럽다고 말하실 수 있도록 열심히 노력하겠습니다.

지난 2년 동안 부족한 저를 지원해주시고, 보살펴 주시며 지도해주신 김성완 교수님께 진심으로 감사드립니다. 김성완 교수님께서 이끌어 주셔서 한 명의 연구자로서 성장할 수 있었습니다. 언제나 존경하는 마음으로 은혜를 잊지 않으며 살아가겠습니다. 연구의 방향을 잡아 주시고 지도해 주시며 도움을 주신 김대곤 교수님께 감사합니다. 김대곤 교수님께서 주신 의학적 관점의 지도를 통해 성장할 수 있었습니다. 귀한 시간을 내 주셔서 학위 심사를 맡아 주신 이정찬 교수님, 김성완 교수님, 그리고 조민우 교수님 감사합니다.

대학원 생활을 함께 해 주신 생체 모델링 및 제어 (BMC) 연구실의 모든 연구원 분들께 감사합니다. 대학원 생활에 적응할 수 있도록 조언과 도움을 주신 조민우 교수님, 김영곤 교수님, 임민혁 교수님, 그리고 전병준 교수님 감사합니다. 인턴 및 석사 과정 동안 도움을 주고 함께 해준 김병수 선배, 윤 단 선배, 김영균 선배, 종현, 승연, 서이, 그리고 재우에게 감사합니다. 덕분에 즐거운 연구실 생활을 할 수 있었습니다. 서울대학교병원 융합의학기술원에서 함께 보낸 연구원분들께 감사합니다.

본 논문은 한국연구재단의 연구결과로 수행되었음 (2021R1C1C1010352). 본 연구는 정부 (과학기술정보통신부, 산업통상자원부, 보건복지부, 식품의약품안전처)의 재원으로

범부처전주기의료기기연구개발사업단의 지원을 받아 수행된 연구임 (과제고유번호 : 1711174462, RS-2020-KD000123). 본 연구는 정부(과학기술정보통신부, 산업통상자원부, 보건복지부, 식품의약품안전처)의 재원으로 범부처전주기의료기기연구개발사업단의 지원을 받아 수행된 연구임 (과제고유번호 : 1711173837, RS-2021-KD000006).