



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2023년08월07일

(11) 등록번호 10-2564404

(24) 등록일자 2023년08월02일

(51) 국제특허분류(Int. Cl.)

A61B 5/00 (2021.01) A61B 5/02 (2006.01)

G06N 3/04 (2023.01) G06N 3/08 (2023.01)

G16H 50/20 (2018.01) G16H 50/50 (2018.01)

(52) CPC특허분류

A61B 5/7275 (2013.01)

A61B 5/02007 (2013.01)

(21) 출원번호 10-2021-0062071

(22) 출원일자 2021년05월13일

심사청구일자 2021년05월13일

(65) 공개번호 10-2022-0154479

(43) 공개일자 2022년11월22일

(56) 선행기술조사문헌

JP2010029434 A*

KR1020210002184 A*

KR1020190131267 A*

KR1020220121576 A

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

연세대학교 산학협력단

서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)

서울대학교병원

서울특별시 종로구 대학로 101(연건동)

(72) 발명자

원중윤

서울특별시 서대문구 연세로 50-1, 신촌세브란스병원 영상의학과교실(신촌동)

한기창

서울특별시 서대문구 연세로 50-1, 신촌세브란스병원 영상의학과교실(신촌동)

(뒷면에 계속)

(74) 대리인

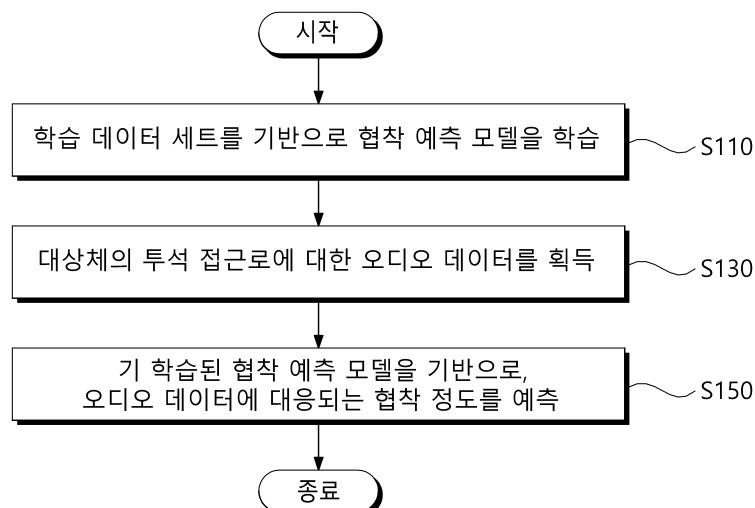
특허법인우인

전체 청구항 수 : 총 7 항

심사관 : 유창용

(54) 발명의 명칭 **합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법 및 장치****(57) 요약**

본 발명의 바람직한 실시예에 따른 합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법 및 장치는, 합성곱 신경망(convolutional neural network, CNN)을 포함하는 협착 예측 모델을 기반으로, 대상체의 투석 접근로에 대한 오디오 데이터로부터 해당 투석 접근로의 협착 정도를 예측함으로써, 보다 정확하게 투석 접근로의 협착 정도를 예측할 수 있으며, 이에 따라 추가적인 검사 및 처치를 안내할 수 있다.

대표도 - 도2

(52) CPC특허분류

A61B 7/04 (2013.01)

G06N 3/042 (2023.01)

G06N 3/08 (2023.01)

G16H 50/20 (2018.01)

G16H 50/50 (2018.01)

(72) 발명자

박재현

서울특별시 마포구 마포대로 195, 104동 201호(아
현동, 마포 래미안 푸르지오)

박인선

경기도 성남시 분당구 정자일로 135, C동 2521호(
정자동)

명세서

청구범위

청구항 1

컴퓨터에서 수행되는, 합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법으로서,

프로세서가, 시술 전에 획득한 투석 접근로에 대한 제1 오디오 데이터 및 시술 후에 획득한 투석 접근로에 대한 제2 오디오 데이터를 포함하는 학습 데이터 세트를 기반으로, 상기 협착 예측 모델을 학습하는 단계;

상기 프로세서가, 대상체의 투석 접근로에 대한 오디오 데이터를 획득하는 단계; 및

상기 프로세서가, 기 학습된 합성곱 신경망(convolutional neural network, CNN)을 포함하는 협착 예측 모델-여기서 상기 협착 예측 모델은 스펙트로그램(spectrogram)을 입력으로 하고, 협착 정도 값을 출력으로 함-을 기반으로, 상기 오디오 데이터에 대응되는 협착 정도를 예측하는 단계;

를 포함하고,

상기 협착 예측 모델 학습 단계는,

상기 학습 데이터 세트를 전처리하고,

상기 제1 오디오 데이터는 협착 정도가 50% 이상인 상태를 나타내는 제1 정답 레이블(label)로 하고 상기 제2 오디오 데이터는 협착 정도가 50% 미만인 상태를 나타내는 제2 정답 레이블(label)로 하여, 전처리한 상기 학습 데이터 세트를 기반으로 상기 협착 예측 모델을 학습하는 것으로 이루어지고,

상기 학습 데이터 세트의 전처리는,

상기 학습 데이터 세트에 포함된 오디오 데이터 각각에 대하여, 상기 오디오 데이터에서 미리 설정된 구간의 상기 오디오 데이터를 획득하고, 미리 설정된 구간의 오디오 데이터를 기반으로 스펙트로그램(spectrogram)을 획득하며, 획득한 상기 스펙트로그램(spectrogram)을 정규화하고, 정규화한 상기 스펙트로그램(spectrogram)을 수평 시프팅(horizontal shifting)하여 개수를 증량하며, 증량된 상기 스펙트로그램(spectrogram)의 크기를 미리 설정된 크기로 감소되도록 조정하여, 상기 학습 데이터 세트를 전처리하는 것으로 이루어지는,

합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법.

청구항 2

제1항에서,

상기 오디오 데이터 획득 단계는,

상기 오디오 데이터를 전처리하는 것으로 이루어지며,

상기 협착 정도 예측 단계는,

전처리된 상기 오디오 데이터를 상기 협착 예측 모델에 입력하고, 상기 협착 예측 모델의 출력값을 기반으로 상기 오디오 데이터에 대응되는 협착 정도를 예측하는 것으로 이루어지는,

합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법.

청구항 3

제2항에서,

상기 오디오 데이터 획득 단계는,

상기 오디오 데이터에서 미리 설정된 구간의 상기 오디오 데이터를 획득하고, 미리 설정된 구간의 상기 오디오 데이터를 기반으로 스펙트로그램(spectrogram)을 획득하며, 획득한 상기 스펙트로그램(spectrogram)을 정규화하고, 정규화한 상기 스펙트로그램(spectrogram)의 크기를 조절하는 것으로 이루어지는,

합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법.

청구항 4

삭제

청구항 5

삭제

청구항 6

삭제

청구항 7

삭제

청구항 8

제1항에서,

상기 협착 예측 모델 학습 단계는,

전처리한 상기 학습 데이터 세트를 미리 설정된 기준에 따라 훈련 데이터 세트, 튜닝 데이터 세트 및 검증 데이터 세트로 구분하고,

상기 협착 예측 모델을 상기 훈련 데이터 세트를 이용하여 학습하며, 학습된 상기 협착 예측 모델을 상기 튜닝 데이터 세트를 이용하여 튜닝하고, 튜닝된 상기 협착 예측 모델을 상기 검증 데이터 세트를 이용하여 검증하는 것으로 이루어지는,

합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법.

청구항 9

제1항 내지 제3항, 제8항 중 어느 한 항에 기재된 합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법을 컴퓨터에서 실행시키기 위하여 컴퓨터 판독 가능한 저장 매체에 저장된 컴퓨터 프로그램.

청구항 10

합성곱 신경망(convolutional neural network, CNN)을 이용하여 투석 접근로의 협착을 예측하는 협착 예측 장치로서,

합성곱 신경망(CNN)을 이용하여 투석 접근로의 협착을 예측하기 위한 하나 이상의 프로그램을 저장하는 메모리; 및

상기 메모리에 저장된 상기 하나 이상의 프로그램에 따라 합성곱 신경망(CNN)을 이용하여 투석 접근로의 협착을 예측하기 위한 동작을 수행하는 하나 이상의 프로세서;

를 포함하며,

상기 프로세서는,

시술 전에 획득한 투석 접근로에 대한 제1 오디오 데이터 및 시술 후에 획득한 투석 접근로에 대한 제2 오디오 데이터를 포함하는 학습 데이터 세트를 기반으로, 상기 협착 예측 모델을 학습하고,

대상체의 투석 접근로에 대한 오디오 데이터를 획득하고,

기 학습된 합성곱 신경망(CNN)을 포함하는 협착 예측 모델-여기서 상기 협착 예측 모델은 스펙트로그램(spectrogram)을 입력으로 하고, 협착 정도 값을 출력으로 함-을 기반으로, 상기 오디오 데이터에 대응되는 협착 정도를 예측하고,

상기 프로세서는,

상기 학습 데이터 세트를 전처리하고,

상기 제1 오디오 데이터는 협착 정도가 50% 이상인 상태를 나타내는 제1 정답 레이블(label)로 하고 상기 제2 오디오 데이터는 협착 정도가 50% 미만인 상태를 나타내는 제2 정답 레이블(label)로 하여, 전처리한 상기 학습 데이터 세트를 기반으로 상기 협착 예측 모델을 학습하고,

상기 학습 데이터 세트의 전처리는,

상기 학습 데이터 세트에 포함된 오디오 데이터 각각에 대하여, 상기 오디오 데이터에서 미리 설정된 구간의 상기 오디오 데이터를 획득하고, 미리 설정된 구간의 오디오 데이터를 기반으로 스펙트로그램(spectrogram)을 획득하며, 획득한 상기 스펙트로그램(spectrogram)을 정규화하고, 정규화한 상기 스펙트로그램(spectrogram)을 수평 시프팅(horizontal shifting)하여 개수를 증량하며, 증량된 상기 스펙트로그램(spectrogram)의 크기를 미리 설정된 크기로 감소되도록 조정하여, 상기 학습 데이터 세트를 전처리하는 것으로 이루어지는,

합성곱 신경망을 이용한 투석 접근로의 협착 예측 장치.

청구항 11

제10항에서,

상기 프로세서는,

상기 오디오 데이터를 전처리하며,

전처리된 상기 오디오 데이터를 상기 협착 예측 모델에 입력하고, 상기 협착 예측 모델의 출력값을 기반으로 상기 오디오 데이터에 대응되는 협착 정도를 예측하는,

합성곱 신경망을 이용한 투석 접근로의 협착 예측 장치.

청구항 12

삭제

청구항 13

삭제

발명의 설명

기술 분야

[0001] 본 발명은 합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법 및 장치에 관한 것으로서, 더욱 상세하게는 치료를 필요로 하는 임상적으로 유의미한 협착을 동반한 투석 접근로를 진단하는, 방법 및 장치에 관한 것이다.

배경 기술

[0002] 동정맥루 등과 같은 투석 접근로의 이상 여부 확인은 촉진과 청진에 많이 의존하고 있는 실정이다. 실제로 협착 부위 전후에 따라서 촉진을 했을 때 만져지는 진동(thrill)과 박동(pulsation)이 부위에 따라 큰 차이를 보이게 된다. 촉진상 진동은 청진기를 사용했을 때 가청 주파수 범위의 고강도 잡음(high pitch bruit)과 같은 진동음으로 들릴 수 있는데, 마찬가지로 잡음(bruit)의 유무 및 강도로 동정맥루의 협착과 폐쇄를 간접적으로 진단할 수 있으나, 청음에 숙달된 의사가 적을 뿐 아니라, 청음에 대한 판단에도 주관적인 요소가 많이 개입되어 혈관 확장술과 같은 치료를 요하는 의미있는 협착을 객관적으로 감별하기가 쉽지 않다.

선행기술문헌

특허문헌

(특허문헌 0001) 일본 공개특허공보 특개2010-029434호(2010.02.12.)

발명의 내용

해결하려는 과제

- [0003] 본 발명이 이루고자 하는 목적은, 합성곱 신경망(convolutional neural network, CNN)을 포함하는 협착 예측 모델을 기반으로, 대상체의 투석 접근로에 대한 오디오 데이터로부터 해당 투석 접근로의 협착 정도를 예측하는, 합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법 및 장치를 제공하는 데 있다.
- [0004] 본 발명의 명시되지 않은 또 다른 목적들은 하기의 상세한 설명 및 그 효과로부터 용이하게 추론할 수 있는 범위 내에서 추가적으로 고려될 수 있다.

과제의 해결 수단

- [0005] 상기의 목적을 달성하기 위한 본 발명의 바람직한 실시예에 따른 합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법은, 대상체의 투석 접근로에 대한 오디오 데이터를 획득하는 단계; 및 기 학습된 합성곱 신경망(convolutional neural network, CNN)을 포함하는 협착 예측 모델을 기반으로, 상기 오디오 데이터에 대응되는 협착 정도를 예측하는 단계;를 포함한다.
- [0006] 여기서, 상기 오디오 데이터 획득 단계는, 상기 오디오 데이터를 전처리하는 것으로 이루어지며, 상기 협착 정도 예측 단계는, 전처리된 상기 오디오 데이터를 상기 협착 예측 모델에 입력하고, 상기 협착 예측 모델의 출력 값을 기반으로 상기 오디오 데이터에 대응되는 협착 정도를 예측하는 것으로 이루어질 수 있다.
- [0007] 여기서, 상기 오디오 데이터 획득 단계는, 상기 오디오 데이터에서 미리 설정된 구간의 상기 오디오 데이터를 획득하고, 미리 설정된 구간의 상기 오디오 데이터를 기반으로 스펙트로그램(spectrogram)을 획득하며, 획득한 상기 스펙트로그램(spectrogram)을 정규화하고, 정규화한 상기 스펙트로그램(spectrogram)의 크기를 조절하는 것으로 이루어질 수 있다.
- [0008] 여기서, 시술 전에 획득한 투석 접근로에 대한 제1 오디오 데이터 및 시술 후에 획득한 투석 접근로에 대한 제2 오디오 데이터를 포함하는 학습 데이터 세트를 기반으로, 상기 협착 예측 모델을 학습하는 단계;를 더 포함할 수 있다.
- [0009] 여기서, 상기 협착 예측 모델은, 스펙트로그램(spectrogram)을 입력으로 하고, 협착 정도 값을 출력으로 할 수 있다.
- [0010] 여기서, 상기 협착 예측 모델 학습 단계는, 상기 학습 데이터 세트를 전처리하고, 상기 제1 오디오 데이터는 제1 정답 레이블(label)로 하고 상기 제2 오디오 데이터는 제2 정답 레이블(label)로 하여, 전처리한 상기 학습 데이터 세트를 기반으로 상기 협착 예측 모델을 학습하는 것으로 이루어질 수 있다.
- [0011] 여기서, 상기 협착 예측 모델 학습 단계는, 상기 학습 데이터 세트에 포함된 오디오 데이터 각각에 대하여, 상기 오디오 데이터에서 미리 설정된 구간의 상기 오디오 데이터를 획득하고, 미리 설정된 구간의 오디오 데이터를 기반으로 스펙트로그램(spectrogram)을 획득하며, 획득한 상기 스펙트로그램(spectrogram)을 정규화하고, 정규화한 상기 스펙트로그램(spectrogram)을 수평 시프팅(horizontal shifting)하여 개수를 증량하며, 증량된 상기 스펙트로그램(spectrogram)의 크기를 조절하여, 상기 학습 데이터 세트를 전처리하는 것으로 이루어질 수 있다.
- [0012] 여기서, 상기 협착 예측 모델 학습 단계는, 전처리한 상기 학습 데이터 세트를 미리 설정된 기준에 따라 훈련 데이터 세트, 튜닝 데이터 세트 및 검증 데이터 세트로 구분하고, 상기 협착 예측 모델을 상기 훈련 데이터 세트를 이용하여 학습하며, 학습된 상기 협착 예측 모델을 상기 튜닝 데이터 세트를 이용하여 튜닝하고, 튜닝된 상기 협착 예측 모델을 상기 검증 데이터 세트를 이용하여 검증하는 것으로 이루어질 수 있다.
- [0014] 상기의 기술적 과제를 달성하기 위한 본 발명의 바람직한 실시예에 따른 컴퓨터 프로그램은 컴퓨터 판독 가능한 저장 매체에 저장되어 상기한 합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법 중 어느 하나를 컴퓨터에서 실행시킨다.
- [0016] 상기의 목적을 달성하기 위한 본 발명의 바람직한 실시예에 따른 합성곱 신경망을 이용한 투석 접근로의 협착 예측 장치는, 합성곱 신경망(convolutional neural network, CNN)을 이용하여 투석 접근로의 협착을 예측하는 협착 예측 장치로서, 합성곱 신경망(CNN)을 이용하여 투석 접근로의 협착을 예측하기 위한 하나 이상의 프로그램을 저장하는 메모리; 및 상기 메모리에 저장된 상기 하나 이상의 프로그램에 따라 합성곱 신경망(CNN)을 이용

하여 투석 접근로의 협착을 예측하기 위한 동작을 수행하는 하나 이상의 프로세서;를 포함하며, 상기 프로세서는, 대상체의 투석 접근로에 대한 오디오 데이터를 획득하고, 기 학습된 합성곱 신경망(CNN)을 포함하는 협착 예측 모델을 기반으로, 상기 오디오 데이터에 대응되는 협착 정도를 예측한다.

[0017] 여기서, 상기 프로세서는, 상기 오디오 데이터를 전처리하며, 전처리된 상기 오디오 데이터를 상기 협착 예측 모델에 입력하고, 상기 협착 예측 모델의 출력값을 기반으로 상기 오디오 데이터에 대응되는 협착 정도를 예측할 수 있다.

[0018] 여기서, 상기 프로세서는, 시술 전에 획득한 투석 접근로에 대한 제1 오디오 데이터 및 시술 후에 획득한 투석 접근로에 대한 제2 오디오 데이터를 포함하는 학습 데이터 세트를 기반으로, 상기 협착 예측 모델을 학습할 수 있다.

[0019] 여기서, 상기 프로세서는, 상기 학습 데이터 세트를 전처리하며, 상기 제1 오디오 데이터는 제1 정답 레이블(label)로 하고 상기 제2 오디오 데이터는 제2 정답 레이블(label)로 하여, 전처리한 상기 학습 데이터 세트를 기반으로 상기 협착 예측 모델을 학습할 수 있다.

발명의 효과

[0020] 본 발명의 바람직한 실시예에 따른 합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법 및 장치에 의하면, 합성곱 신경망(convolutional neural network, CNN)을 포함하는 협착 예측 모델을 기반으로, 대상체의 투석 접근로에 대한 오디오 데이터로부터 해당 투석 접근로의 협착 정도를 예측함으로써, 보다 정확하게 투석 접근로의 협착 정도를 예측할 수 있으며, 이에 따라 추가적인 검사 및 처치를 안내할 수 있다.

[0021] 본 발명의 효과들은 이상에서 언급한 효과들로 제한되지 않으며, 언급되지 않은 또 다른 효과들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0022] 도 1은 본 발명의 바람직한 실시예에 따른 합성곱 신경망을 이용한 투석 접근로의 협착 예측 장치를 설명하기 위한 블록도이다.

도 2는 본 발명의 바람직한 실시예에 따른 합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법을 설명하기 흐름도이다.

도 3은 본 발명의 바람직한 실시예에 따른 협착 예측 모델의 학습 과정을 설명하기 위한 도면이다.

도 4는 도 3에 도시한 학습 데이터 세트의 전처리 과정을 설명하기 위한 도면이다.

도 5는 본 발명의 바람직한 실시예에 따른 협착 예측 모델을 이용한 협착 정도 예측 과정을 설명하기 위한 도면이다.

도 6은 도 5에 도시한 오디오 데이터의 전처리 과정을 설명하기 위한 도면이다.

도 7은 본 발명의 바람직한 실시예에 따른 협착 예측 모델 학습 과정과 협착 정도 예측 과정의 일례를 설명하기 위한 도면이다.

도 8은 본 발명의 바람직한 실시예에 따른 스펙트로그램(spectrogram) 획득 과정의 일례를 설명하기 위한 도면이다.

도 9는 도 8에 도시된 과정을 통해 획득한 스펙트로그램(spectrogram)의 일례를 나타내는 도면이다.

도 10은 도 8에 도시된 과정을 통해 획득한 스펙트로그램(spectrogram)의 일례를 나타내는 도면으로, 도 10의 (a)는 시술 전에 획득한 투석 접근로에 대한 오디오 데이터를 기반으로 획득한 스펙트로그램(spectrogram)을 나타내고, 도 10의 (b)는 시술 후에 획득한 투석 접근로에 대한 오디오 데이터를 기반으로 획득한 스펙트로그램(spectrogram)을 나타낸다.

도 11은 본 발명의 바람직한 실시예에 따른 협착 예측 모델의 성능을 설명하기 위한 도면으로, 도 11의 (a)는 혼동 행렬(confusion matrix)을 나타내고, 도 11의 (b)는 ROC(receiver operation characteristic) 곡선을 나타낸다.

발명을 실시하기 위한 구체적인 내용

- [0023] 이하, 첨부된 도면을 참조하여 본 발명의 실시예를 상세히 설명한다. 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나, 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있으며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하고, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다. 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성 요소를 지칭한다.
- [0024] 다른 정의가 없다면, 본 명세서에서 사용되는 모든 용어(기술 및 과학적 용어를 포함)는 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 공통적으로 이해될 수 있는 의미로 사용될 수 있을 것이다. 또한, 일반적으로 사용되는 사전에 정의되어 있는 용어들은 명백하게 특별히 정의되어 있지 않는 한 이상적으로 또는 과도하게 해석되지 않는다.
- [0025] 본 명세서에서 "제1", "제2" 등의 용어는 하나의 구성 요소를 다른 구성 요소로부터 구별하기 위한 것으로, 이들 용어들에 의해 권리범위가 한정되어서는 아니 된다. 예컨대, 제1 구성 요소는 제2 구성 요소로 명명될 수 있고, 유사하게 제2 구성 요소도 제1 구성 요소로 명명될 수 있다.
- [0026] 본 명세서에서 각 단계들에 있어 식별부호(예컨대, a, b, c 등)는 설명의 편의를 위하여 사용되는 것으로 식별 부호는 각 단계들의 순서를 설명하는 것이 아니며, 각 단계들은 문맥상 명백하게 특정 순서를 기재하지 않는 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 단계들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.
- [0027] 본 명세서에서, "가진다", "가질 수 있다", "포함한다" 또는 "포함할 수 있다" 등의 표현은 해당 특징(예컨대, 수치, 기능, 동작, 또는 부품 등의 구성 요소)의 존재를 가리키며, 추가적인 특징의 존재를 배제하지 않는다.
- [0029] 이하에서 첨부한 도면을 참조하여 본 발명에 따른 합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법 및 장치의 바람직한 실시예에 대해 상세하게 설명한다.
- [0031] 먼저, 도 1을 참조하여 본 발명의 바람직한 실시예에 따른 합성곱 신경망을 이용한 투석 접근로의 협착 예측 장치에 대하여 설명한다.
- [0032] 도 1은 본 발명의 바람직한 실시예에 따른 합성곱 신경망을 이용한 투석 접근로의 협착 예측 장치를 설명하기 위한 블록도이다.
- [0033] 도 1을 참조하면, 본 발명의 바람직한 실시예에 따른 합성곱 신경망을 이용한 투석 접근로의 협착 예측 장치(100)는 합성곱 신경망(convolutional neural network, CNN)을 포함하는 협착 예측 모델을 기반으로, 대상체의 투석 접근로(동정맥루 등)에 대한 오디오 데이터로부터 해당 투석 접근로의 협착 정도를 예측할 수 있다.
- [0035] 이를 위해, 협착 예측 장치(100)는 하나 이상의 프로세서(110), 컴퓨터 판독 가능한 저장 매체(130) 및 통신 버스(150)를 포함할 수 있다.
- [0036] 프로세서(110)는 협착 예측 장치(100)가 동작하도록 제어할 수 있다. 예컨대, 프로세서(110)는 컴퓨터 판독 가능한 저장 매체(130)에 저장된 하나 이상의 프로그램(131)을 실행할 수 있다. 하나 이상의 프로그램(131)은 하나 이상의 컴퓨터 실행 가능 명령어를 포함할 수 있으며, 컴퓨터 실행 가능 명령어는 프로세서(110)에 의해 실행되는 경우 협착 예측 장치(100)로 하여금 합성곱 신경망(CNN)을 이용하여 투석 접근로의 협착을 예측하기 위한 동작을 수행하도록 구성될 수 있다.
- [0037] 컴퓨터 판독 가능한 저장 매체(130)는 합성곱 신경망(CNN)을 이용하여 투석 접근로의 협착을 예측하기 위한 컴퓨터 실행 가능 명령어 내지 프로그램 코드, 프로그램 데이터 및/또는 다른 적합한 형태의 정보를 저장하도록 구성된다. 컴퓨터 판독 가능한 저장 매체(130)에 저장된 프로그램(131)은 프로세서(110)에 의해 실행 가능한 명령어의 집합을 포함한다. 일 실시예에서, 컴퓨터 판독 가능한 저장 매체(130)는 메모리(랜덤 액세스 메모리와 같은 휘발성 메모리, 비휘발성 메모리, 또는 이들의 적절한 조합), 하나 이상의 자기 디스크 저장 디바이스들, 광학 디스크 저장 디바이스들, 플래시 메모리 디바이스들, 그 밖에 협착 예측 장치(100)에 의해 액세스되고 원하는 정보를 저장할 수 있는 다른 형태의 저장 매체, 또는 이들의 적합한 조합일 수 있다.
- [0038] 통신 버스(150)는 프로세서(110), 컴퓨터 판독 가능한 저장 매체(130)를 포함하여 협착 예측 장치(100)의 다른 다양한 컴포넌트들을 상호 연결한다.

- [0039] 협착 예측 장치(100)는 또한 하나 이상의 입출력 장치를 위한 인터페이스를 제공하는 하나 이상의 입출력 인터페이스(170) 및 하나 이상의 통신 인터페이스(190)를 포함할 수 있다. 입출력 인터페이스(170) 및 통신 인터페이스(190)는 통신 버스(150)에 연결된다. 입출력 장치(도시하지 않음)는 입출력 인터페이스(170)를 통해 협착 예측 장치(100)의 다른 컴포넌트들에 연결될 수 있다.
- [0041] 그러면, 도 2 내지 도 6을 참조하여 본 발명의 바람직한 실시예에 따른 합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법에 대하여 설명한다.
- [0042] 도 2는 본 발명의 바람직한 실시예에 따른 합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법을 설명하기 흐름도이고, 도 3은 본 발명의 바람직한 실시예에 따른 협착 예측 모델의 학습 과정을 설명하기 위한 도면이며, 도 4는 도 3에 도시한 학습 데이터 세트의 전처리 과정을 설명하기 위한 도면이고, 도 5는 본 발명의 바람직한 실시예에 따른 협착 예측 모델을 이용한 협착 정도 예측 과정을 설명하기 위한 도면이며, 도 6은 도 5에 도시한 오디오 데이터의 전처리 과정을 설명하기 위한 도면이다.
- [0043] 도 2를 참조하면, 협착 예측 장치(100)의 프로세서(110)는 학습 데이터 세트를 기반으로 협착 예측 모델을 학습할 수 있다(S110).
- [0044] 여기서, 협착 예측 모델은 합성곱 신경망(CNN)을 포함하며, 스펙트로그램(spectrogram)을 입력으로 하고, 협착 정도 값을 출력으로 할 수 있다. 예컨대, 협착 정도 값은 투석 접근로의 협착 정도가 50% 이상일 확률을 나타내는 값으로, 0 ~ 1 사이의 값을 가질 수 있다.
- [0045] 그리고, 학습 데이터 세트는 혈관 확장술의 시술 전에 획득한 투석 접근로에 대한 제1 오디오 데이터 및 혈관 확장술의 시술 후에 획득한 투석 접근로에 대한 제2 오디오 데이터를 포함할 수 있다. 예컨대, 혈관 확장술을 시술하기 전에 환자의 투석 접근로(동정맥루 등)에 대한 오디오 데이터(20Hz ~ 1,000Hz 사이의 가청 주파수 대역의 소리)를 전자 청진기 등을 이용하여 획득할 수 있다. 이와 동일하게, 혈관 확장술을 시술한 후에 환자의 투석 접근로(동정맥루 등)에 대한 오디오 데이터를 전자 청진기 등을 이용하여 획득할 수 있다.
- [0046] 예컨대, 프로세서(110)는 도 3에 도시된 바와 같이, "학습 데이터 세트의 전처리 과정" -> "협착 예측 모델의 학습 과정" -> "협착 예측 모델의 튜닝 과정" -> "협착 예측 모델의 검증 과정"을 거쳐, 최종 협착 예측 모델을 학습할 수 있다.
- [0047] 즉, 프로세서(110)는 학습 데이터 세트를 전처리할 수 있다.
- [0048] 도 4를 참조하여 보다 자세하게 설명하면, 프로세서(110)는 학습 데이터 세트에 포함된 오디오 데이터 각각에 대해 아래와 같은 과정을 거쳐 전처리할 수 있다.
- [0049] 프로세서(110)는 오디오 데이터에서 미리 설정된 구간의 오디오 데이터를 획득할 수 있다. 예컨대, 프로세서(110)는 잡음 등의 영향을 제거하기 위해, 미리 설정된 구간(2초 ~ 8초 등)의 오디오 데이터를 추출할 수 있다.
- [0050] 프로세서(110)는 미리 설정된 구간의 오디오 데이터를 기반으로 스펙트로그램(spectrogram)을 획득할 수 있다. 예컨대, 프로세서(110)는 푸리에 변환(Fourier transform, FT) 등을 이용하여 오디오 데이터를 스펙트로그램(spectrogram)으로 변환할 수 있다.
- [0051] 프로세서(110)는 획득한 스펙트로그램(spectrogram)을 정규화(normalization)할 수 있다.
- [0052] 프로세서(110)는 데이터 증강(data augmentation)을 수행하기 전에, 정규화한 스펙트로그램(spectrogram)에서 불필요 영역(가장자리 경계 영역 등)을 제거할 수 있다.
- [0053] 프로세서(110)는 정규화한 스펙트로그램(spectrogram)을 수평 시프팅(horizontal shifting)하여 개수를 증량할 수 있다. 예컨대, 프로세서(110)는 스펙트로그램(spectrogram)을 기준으로 시간 축에서 복수회 수평 시프팅(horizontal shifting)하여 스펙트로그램(spectrogram)의 개수를 증가시킬 수 있다.
- [0054] 프로세서(110)는 증량된 스펙트로그램(spectrogram)의 크기를 조절할 수 있다. 예컨대, 프로세서(110)는 스펙트로그램(spectrogram)의 크기가 미리 설정된 크기(예컨대, 512 × 512 등)로 감소되도록 크기 조절을 할 수 있다.
- [0055] 그런 다음, 프로세서(110)는 제1 오디오 데이터는 제1 정답 레이블(label)로 하고 제2 오디오 데이터는 제2 정답 레이블(label)로 하여, 전처리한 학습 데이터 세트를 기반으로 협착 예측 모델을 학습할 수 있다.
- [0056] 여기서, 제1 정답 레이블은 투석 접근로의 협착 정도가 50% 이상인 상태를 나타내며, 예컨대 '1'로 설정될 수

있다. 제2 정답 레이블은 투석 접근로의 협착 정도가 50% 미만인 상태를 나타내며, 예컨대 '0'으로 설정될 수 있다.

- [0057] 보다 자세하게 설명하면, 프로세서(110)는 전처리된 학습 데이터 세트를 기반으로 아래와 같은 과정을 거쳐 협착 예측 모델을 학습할 수 있다.
- [0058] 프로세서(110)는 전처리한 학습 데이터 세트를 미리 설정된 기준에 따라 훈련 데이터 세트, 튜닝 데이터 세트 및 검증 데이터 세트로 구분할 수 있다. 예컨대, 프로세서(110)는 미리 설정된 비율인 "7:2:1"에 따라, 학습 데이터 세트의 제1 오디오 데이터 세트를 훈련 데이터 세트, 튜닝 데이터 세트 및 검증 데이터 세트로 구분하고, 학습 데이터 세트의 제2 오디오 세트를 훈련 데이터 세트, 튜닝 데이터 세트 및 검증 데이터 세트로 구분할 수 있다.
- [0059] 프로세서(110)는 협착 예측 모델을 훈련 데이터 세트를 이용하여 학습할 수 있다.
- [0060] 프로세서(110)는 학습된 협착 예측 모델을 튜닝 데이터 세트를 이용하여 튜닝할 수 있다.
- [0061] 프로세서(110)는 튜닝된 협착 예측 모델을 검증 데이터 세트를 이용하여 검증할 수 있다.
- [0063] 이후, 프로세서(110)는 대상체의 투석 접근로에 대한 오디오 데이터를 획득할 수 있다(S130).
- [0064] 예컨대, 프로세서(110)는 도 5에 도시된 바와 같이, "오디오 데이터의 획득 과정" -> "오디오 데이터의 전처리 과정"을 거쳐, 오디오 데이터를 획득할 수 있다.
- [0065] 즉, 프로세서(110)는 대상체의 투석 접근로에 대한 오디오 데이터를 획득할 수 있다. 예컨대, 협착 정도를 판단할 대상 환자의 투석 접근로(동정맥루 등)에 대한 오디오 데이터(20Hz ~ 1,000Hz 사이의 가청 주파수 대역의 소리)를 전자 청진기 등을 이용하여 획득할 수 있다.
- [0066] 그리고, 프로세서(110)는 획득한 오디오 데이터를 전처리할 수 있다.
- [0067] 도 6을 참조하여 보다 자세하게 설명하면, 프로세서(110)는 오디오 데이터에 대해 아래와 같은 과정을 거쳐 전처리할 수 있다.
- [0068] 프로세서(110)는 오디오 데이터에서 미리 설정된 구간의 오디오 데이터를 획득할 수 있다. 예컨대, 프로세서(110)는 잡음 등의 영향을 제거하기 위해, 미리 설정된 구간(2초 ~ 8초 등)의 오디오 데이터를 추출할 수 있다.
- [0069] 프로세서(110)는 미리 설정된 구간의 오디오 데이터를 기반으로 스펙트로그램(spectrogram)을 획득할 수 있다. 예컨대, 프로세서(110)는 푸리에 변환(FT) 등을 이용하여 오디오 데이터를 스펙트로그램(spectrogram)으로 변환할 수 있다.
- [0070] 프로세서(110)는 획득한 스펙트로그램(spectrogram)을 정규화(normalization)할 수 있다.
- [0071] 프로세서(110)는 정규화한 스펙트로그램(spectrogram)의 크기를 조절할 수 있다. 예컨대, 프로세서(110)는 스펙트로그램(spectrogram)의 크기가 미리 설정된 크기(예컨대, 512 × 512 등)로 감소되도록 크기 조절을 할 수 있다.
- [0073] 그런 다음, 프로세서(110)는 기 학습된 협착 예측 모델을 기반으로, 오디오 데이터에 대응되는 협착 정도를 예측할 수 있다(S150).
- [0074] 예컨대, 프로세서(110)는 도 5에 도시된 바와 같이, "전처리된 오디오 데이터의 입력 과정" -> "협착 예측 모델의 출력값 획득 과정" -> "협착 정도 예측 과정"을 거쳐, 대상체의 투석 접근로의 협착 정도를 예측할 수 있다.
- [0075] 즉, 프로세서(110)는 전처리된 오디오 데이터를 협착 예측 모델에 입력할 수 있다.
- [0076] 그리고, 프로세서(110)는 협착 예측 모델의 출력값을 기반으로 오디오 데이터에 대응되는 협착 정도를 예측할 수 있다.
- [0077] 예컨대, 협착 예측 모델의 출력값(즉, 협착 정도 값)이 "0.95"인 경우, 해당 대상체의 투석 접근로의 협착 정도가 50% 이상일 확률이 "95%"라는 것을 나타낸다. 이에 따라, 프로세서(110)는 해당 대상체의 협착 정도를 "95%"로 예측할 수 있다.
- [0078] 물론, 프로세서(110)는 협착 예측 모델의 출력값(즉, 협착 정도 값)을 미리 설정된 임계값(예컨대, 0.5 등)과 대비하여, 출력값(즉, 협착 정도 값)이 임계값 이상인 경우에는 "협착 의심" 등으로 해당 대상체의 협착 정도를

예측할 수 있고, 출력값(즉, 협착 정도 값)이 임계값 미만인 경우에는 "협착 아님" 등으로 해당 대상체의 협착 정도를 예측할 수도 있다.

[0080] 그러면, 도 7 내지 도 11을 참조하여 본 발명의 바람직한 실시예에 따른 합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법의 일례와 성능에 대하여 설명한다.

[0081] 도 7은 본 발명의 바람직한 실시예에 따른 협착 예측 모델 학습 과정과 협착 정도 예측 과정의 일례를 설명하기 위한 도면이고, 도 8은 본 발명의 바람직한 실시예에 따른 스펙트로그램(spectrogram) 획득 과정의 일례를 설명하기 위한 도면이며, 도 9는 도 8에 도시된 과정을 통해 획득한 스펙트로그램(spectrogram)의 일례를 나타내는 도면이고, 도 10은 도 8에 도시된 과정을 통해 획득한 스펙트로그램(spectrogram)의 일례를 나타내는 도면으로, 도 10의 (a)는 시술 전에 획득한 투석 접근로에 대한 오디오 데이터를 기반으로 획득한 스펙트로그램(spectrogram)을 나타내고, 도 10의 (b)는 시술 후에 획득한 투석 접근로에 대한 오디오 데이터를 기반으로 획득한 스펙트로그램(spectrogram)을 나타내며, 도 11은 본 발명의 바람직한 실시예에 따른 협착 예측 모델의 성능을 설명하기 위한 도면으로, 도 11의 (a)는 혼동 행렬(confusion matrix)을 나타내고, 도 11의 (b)는 ROC(receiver operation characteristic) 곡선을 나타낸다.

[0082] 도 7을 참조하면, 본 발명의 바람직한 실시예에 따른 합성곱 신경망을 이용한 투석 접근로의 협착 예측 방법의 일례는 크게 "이미지 전처리 과정(도 7에 도시된 Image Preprocessing)" 및 "딥 러닝 프로세스(도 7에 도시된 Deep Learning Process)"로 이루어지는 협착 예측 모델 학습 과정과, "환자의 오디오 데이터 전처리 과정(도 7에 도시된 User)", "딥 러닝 프로세스(도 7에 도시된 Deep Learning Process)" 및 "환자의 협착 정도 예측 과정(도 7에 도시된 Output)"으로 이루어지는 협착 정도 예측 과정을 포함할 수 있다.

[0083] 협착 예측 모델에 입력되는 데이터(협착 예측 모델의 학습에 이용되는 학습 데이터 세트에 포함된 오디오 데이터 또는 협착 정도를 판단할 대상체의 오디오 데이터)는 멜 스펙트로그램(mel spectrogram)으로, 512×512 크기를 가지고 RGB 채널 개수가 3개인 이미지 파일일 수 있다.

[0084] 전자 청진기 등을 통해 환자의 투석 접근로(동정맥루 등)의 청음을 녹음한 오디오 파일을 획득할 수 있다. 약 10초 동안 녹음을 진행하였으나, 사람이 직접 녹음하는 경우, 오디오 파일의 재생 시간이 파일별로 차이가 있을 수 있고, 녹음을 시작할 때와 끝낼 때 청진기를 만지는 등 잡음이 오디오 파일에 들어갈 수 있기 때문에, 이와 같은 잡음 등을 제거하기 위해서 각 오디오 파일의 2초부터 8초가 되는 시간, 즉 6초간의 오디오 데이터를 실제 사용하였다.

표 1

[0085] `trim_wav(DATA_DIR + fname, DATA_DIR2 + fname, 2, 8)`

[0086] 이후, 오디오 파일을 특정 샘플링 레이트(sampling rate)에 따라 샘플링을 하여 뉴메릭 넘버(numeric number)를 어레이(array) 형태로 저장할 수 있다.

표 2

[0087] `y, sr = librosa.load(.wav)`

[0088] 여기서, sr를 특정 값으로 설정할 수 있지만 본 발명에서는 네이티브 샘플링 레이트(native sampling rate)를 사용했기에 sr=None으로 설정하였다.

[0089] 이렇게 되면 샘플링 interval(x-축, time) vs. amplitude(y-축) 그래프를 만들 수 있는데 이는 분석에는 그리 유용하지 못한다. 소리는 기본적으로 특정 주파수를 가진 sin 함수들의 합이라고 볼 수 있는데, 위에서 구한 y 파형을 주파수 분석을 통해서 특정 시간에 각 주파수 성분이 어떻게 구성이 되어 있는지 확인할 수 있는데, 이 방법이 바로 푸리에 변환(FT)이며, 즉 푸리에 방정식을 풀면, amplitude vs. time 그래프를 frequency vs. time 그래프로 바꿀 수 있고, 본 발명에서는 푸리에 변환(FT)으로 STFT(short time Fourier transform)를 사용하였다.

표 3

[0090]

```
S = librosa.feature.melspectrogram(y=y, n_mels=40, n_fft=input_nfft, hop_length=input_stride,
fmin=fmin, fmax=fmax)
```

[0091]

여기서, fmax는 분석 범위를 결정하는 maximum frequency인데, 보통 Nyquist 법칙에 따라 maximum frequency는 sampling rate/2인 값으로 결정한다.

[0092]

이런 방식으로 STFT를 했을 때 얼마나 나누어 분석을 진행할 것인지 여부는 도 8에 도시된 Hop Length로 결정할 수 있다. n_fft가 분석할 FFT length(또는 window length)이고, 이는 25 msec으로 결정하였다. Hop Length는 10 msec으로 해서 1 칸당 15 msec(Overlap Length)가 겹치도록 설정을 하였다.

[0093]

이 방식으로 도 9에 도시된 바와 같은 멜 스펙트로그램(mel spectrogram)을 획득할 수 있다. X 축은 time이고, Y-축은 frequency이며, 특정 시간대의 특정 주파수의 세기, 데시벨은 색상으로 표현할 수 있다.

[0094]

하지만, 실제 멜 스펙트로그램(mel spectrogram)을 이용해서 학습을 시킬 때는 특징 추출(feature extraction)을 해야 하는데, 멜 스펙트로그램(mel spectrogram)에서 색상으로 표현되는 스펙트로그램(spectrogram)의 파워(power)의 분간 능력을 키우고 데이터의 균일성을 위해서는 스펙트로그램(spectrogram)의 정규화(normalization)가 필요하다. 즉, 녹음할 때마다 들어가는 잡음의 정도가 다를 수 있고, 협착 정도에 따라 특정 주파수에서 더 큰 파워(power)의 음파가 녹음될 수 있기 때문에, 그리고 협착 예측 모델을 학습시킬 때 특징(feature)을 최대한 잘 인식할 수 있도록 하기 위해, 스펙트로그램(spectrogram)의 정규화(normalization)를 진행하여야 한다.

표 4

[0095]

```
def normalize_mel(S):
    return np.clip((S-min_level_db)/-min_level_db, 0, 1)
def norm_mel(a):
    norm_log_S = normalize_mel(librosa.power_to_db(a, ref=np.max))
    return norm_log_S
S = librosa.feature.melspectrogram(y=y, n_mels=40, n_fft=input_nfft,
    hop_length=input_stride, fmin=fmin, fmax=fmax)
S_re = norm_mel(S)
```

[0096]

위와 같은 방식으로 각 오디오 파일로부터 멜 스펙트로그램(mel spectrogram)을 획득할 수 있고, 혈관 확장술의 시술 전과 시술 후의 스펙트로그램(spectrogram)의 예시는 도 10에 도시된 바와 같다. 도 10의 (a)는 혈관 확장술의 시술 전 멜 스펙트로그램(mel spectrogram)이고, 도 10의 (b)는 혈관 확장술의 시술 후 멜 스펙트로그램(mel spectrogram)이다. 시술 후 투석 접근로(동정맥루 등)의 협착 정도가 호전되면서 높은 주파수에서 더 큰 파워(power)의 스펙트로그램(spectrogram)이 보이는 것을 확인할 수 있다. 실제로 시술 전과 시술 후에 녹음한 청음을 들으면 시술 후 청음이 더 크고 잘 들리게 개선되는 것을 확인할 수 있다.

[0097]

이렇게 얻은 멜 스펙트로그램(mel spectrogram)은 혈관 확장술의 시술 전에 얻은 경우, 협착 정도가 50% 이상(실제 혈관 조영술에서 동정맥루 협착 부위와 정상 혈관의 직경 값을 비교하여 계산함)이므로 제1 정답 레이블인 "pre (1)"로 레이블(label)해서 폴더에 저장하고, 혈관 확장술의 시술 후에 얻은 경우, 협착 정도가 50% 미만(실제 혈관 조영술에서 협착 정도가 50% 미만으로 확인된 경우)이므로 제2 정답 레이블인 "post (0)"으로 레이블(label)해서 폴더에 저장하였다.

[0098]

그리고, 도 9에 도시된 바와 같이, 멜 스펙트로그램(mel spectrogram)의 가장자리 영역을 둘러싸고 있는 하얀 경계가 있다. 이 경계를 더 극명하게 보이게 하기 위해 가장자리 파란선은 임의적으로 표시하였다.

[0099]

협착 예측 모델을 학습시키기 위해서는 멜 스펙트로그램(mel spectrogram)의 데이터 수를 증량해야 하는데, 이때 수평 시프팅(horizontal shifting) 방법을 사용한다. 고양이를 인식하는 합성곱 신경망을 개발할 때는, 원본 고양이 사진에 여러 각도를 주거나 vertical/horizontal flip과 같은 기법을 써서 사진을 증폭시켜 학습시킬 수 있지만, 멜 스펙트로그램(mel spectrogram)은 일반적인 고양이 사진과 달리 x축, y축, z축 값과 의미가 정해져 있는 vector 그림이기에, 데이터 증량이 가능한 방법은 수평 시프팅(horizontal shift) 방법 밖에 없다. 수

평 시프트(horizontal shift)로 얻은 데이터는 현실적으로 같은 환자를 녹음해도 녹음 시작 시간과 종료 시간이 다르면 얻을 수 있는 결과물이기 때문에 학습 데이터로 이용하여도 무방하다. 즉, 수평 시프팅(horizontal shifting)을 사용한 이유는 현실에서도 녹음 시작 시간과 종료 시간에 따라 멜 스펙트로그램(mel spectrogram)이 x축(=time)을 따라 이동할 수 있기 때문이다. 멜 스펙트로그램(mel spectrogram)에 보이는 반복적인 피크(peak)가 $\sin(x)$ 또는 $\cos(x)$ 함수라고 생각했을 때, 녹음 시간 범위를 어떻게 설정하냐에 따라 캡처되는 웨이브(wave)가 $\sin(x+a)$ 또는 $\cos(x+a)$ 처럼 보일 수 있다.

[0100] 이러한 데이터 증량은 아래와 같은 ImageDataGenerator를 이용하였다.

표 5

[0101]

```
from keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img
data_aug_gen = ImageDataGenerator(rescale=1./255,
                                   rotation_range= 0,
                                   width_shift_range=0.9,
                                   height_shift_range= 0,
                                   shear_range=0,
                                   #zoom_range=[0.8, 2.0],
                                   horizontal_flip= False,
                                   vertical_flip= False,
                                   fill_mode='wrap')
# 이 for는 무한으로 반복되기 때문에 원하는 반복 횟수를 지정하여, 지정된 반복 횟수가 되면 빠져나오도록 해야한다.
for batch in data_aug_gen.flow(x, batch_size=1, save_to_dir =DATA_DIR9, save_prefix='aug',
                               save_format='png'):
    i += 1
    if i > 50:
        break
```

[0102]

위의 코드를 보면 수평 시프팅(horizontal shifting)을 위해 width shift range를 0.9로 설정한 것을 볼 수 있다. 위의 코드에서는 증량을 50배($i > 50$)로 설정, 즉, 멜 스펙트로그램(mel spectrogram) 이미지 한 개를 가지고 x축으로 이동하여 50개의 이미지를 생성하였으나, 50은 임의적으로 정한 값이며, 50배에 한정할 필요는 없다.

[0103]

하지만, 도 9에 도시된 바와 같이, 하얀 가장자리 영역이 있으면 수평 시프트(horizontal shift)를 했을 때 왼쪽이나 오른쪽 끝에 있는 하얀 세로선이 중간에 끼어들어서 멜 스펙트로그램(mel spectrogram) 데이터를 훼손시킬 수 있기 때문에 이러한 하얀 가장 자리 영역을 없애는 전처리를 수행한다.

표 6

[0104]

```
#trim image to remove white border (no need to designate white 225 225 225, uses pixel (0,0))
def trim(f):
    bg = Image.new(f.mode,f.size, f.getpixel((0,0)))
    diff = ImageChops.difference(f, bg)
    diff = diff = ImageChops.add(diff, diff, 2.0, -100)
    bbox = diff.getbbox()
    if bbox:
        return im.crop(bbox)
```

[0105]

그리고, 하얀 가장자리 영역의 왼쪽과 오른쪽 끝에 보이지 않는 검은 선도 함께 제거를 한다.

표 7

[0106]

```
#crop sides to remove streaky vertical line
def crpim(im):
    width, height = im.size
    img = im
    img_res = img.crop((10,0,width-10,height))
    return img_res
```

[0107]

이렇게 얻은 멜 스펙트로그램(mel spectrogram)은 아래 방식으로 확인을 해보면 크기가 2328 × 909이다.

표 8

[0108]

```
import cv2
im = cv2.imread(DATA_DIR6 + 'postex.wav.png')
h, w, c = im.shape
print('width: ', w)
print('height: ', h)
print('channel:', c)

width: 2328
height: 909
channel: 3
```

[0109]

이미지의 크기가 너무 크면 협착 예측 모델이 어레이(array)를 단계적으로 압축할 때, 시작 이미지에서 너무 큰 영역이 압축이 되어서 마지막 레이어에 이르렀을 때 시술 전과 시술 후의 멜 스펙트로그램(mel spectrogram)의 값이 유의한 차이를 보이지 못할 가능성이 있다. 또한, 협착 예측 모델의 학습 시간이 오래 걸리기 때문에, 직 사각형인 위와 같은 이미지들을 정사각형 사진으로 크기를 조절한다.

표 9

[0110]

```
img_width, img_height = 512 , 512
#512 x 512 upper limit?
img_channel = 3
img_shape = (img_width, img_height, img_channel)
n_classes = 2
epochs = 10
batch_size = 15
def read_img(img_file_path, height = img_height, width = img_width):
    tmp_img = imageio.imread(img_file_path)
    tmp_img = tmp_img[:, :, :3] # get rgb channels
    tmp_img = tmp_img.astype('float32') #change data type
    tmp_img -= np.min(tmp_img)
    tmp_img /= np.max(tmp_img)
    tmp_img = cv2.resize(tmp_img, (width, height), interpolation = cv2.INTER_CUBIC)
    return tmp_img
```

[0111]

위의 코드는 512 × 512로 멜 스펙트로그램(mel spectrogram)의 크기를 조절한 예시이며, 위와 같이 크기 조절 후 협착 예측 모델에 입력하게 된다.

[0113]

학습 데이터 세트는 7:1:2의 비율에 따라 훈련 데이터 세트, 튜닝 데이터 세트 및 검증 데이터 세트로 구분하여 분석을 진행하였다. 이를 위해 train_test_split function을 사용하였으며, train_test_split은 random하게 폴더 내 파일 리스트에 대한 어레이(array)를 training, tuning, validation subset으로 나누어준다.

FIG 10

[0114]

```

ratio_train = 0.70
ratio_tune = 0.10
ratio_val = 0.20
filelist_pre = os.listdir(PRE_PATH)
filelist_post = os.listdir(POST_PATH)
Y_pre = np.zeros(len(filelist_pre))
Y_post = np.ones(len(filelist_post))
filelist_train_tune, filelist_val, Y_train_tune, Y_val = train_test_split(filelist, Y,
stratify = Y, test_size = ratio_val, random_state=SEED)
filelist_train, filelist_tune, Y_train, Y_tune = train_test_split(filelist_train_tune,
Y_train_tune,
stratify = Y_train_tune,
test_size = (ratio_tune/(1-ratio_val)), random_state=SEED)

```

[0115]

그러면, 예를 들어, filelist_tune에는 다음과 같은 melspectrogram.png 파일들이 어레이(array) 형태로 들어 가게 된다.

FIG 11

[0116]

```

print(filelist_tune)

['aug_0_4960.png' 'aug_0_1335.png' 'aug_0_4483.png' 'aug_0_2761.png'
'aug_0_6022.png' 'aug_0_460.png' 'aug_0_8006.png' 'aug_0_8392.png'
'aug_0_8322.png' 'aug_0_1380.png' 'aug_0_9992.png' 'aug_0_8180.png'
'aug_0_4821.png' 'aug_0_9618.png' 'aug_0_5742.png' 'aug_0_4492.png'
'aug_0_6548.png' 'aug_0_9888.png' 'aug_0_2143.png' 'aug_0_7711.png'
'aug_0_6265.png' 'aug_0_483.png' 'aug_0_6907.png' 'aug_0_2448.png'
'aug_0_9725.png' 'aug_0_5616.png' 'aug_0_1087.png' 'aug_0_5973.png'
'aug_0_813.png' 'aug_0_6349.png' 'aug_0_8544.png' 'aug_0_9848.png'
'aug_0_5402.png' 'aug_0_159.png' 'aug_0_9178.png' 'aug_0_4356.png'
'aug_0_7508.png' 'aug_0_6779.png' 'aug_0_2304.png' 'aug_0_4412.png'
'aug_0_8247.png' 'aug_0_3615.png' 'aug_0_9967.png' 'aug_0_6395.png'
'aug_0_7328.png' 'aug_0_6707.png' 'aug_0_8903.png' 'aug_0_921.png'
'aug_0_1947.png' 'aug_0_7142.png' 'aug_0_5883.png' 'aug_0_217.png'
'aug_0_3444.png' 'aug_0_7394.png' 'aug_0_1708.png' 'aug_0_8178.png'
'aug_0_1137.png' 'aug_0_4933.png' 'aug_0_4119.png' 'aug_0_403.png'
'aug_0_4120.png' 'aug_0_6206.png' 'aug_0_3864.png' 'aug_0_8954.png'
'aug_0_2758.png' 'aug_0_4700.png' 'aug_0_1780.png' 'aug_0_8847.png'
'aug_0_642.png' 'aug_0_9361.png' 'aug_0_7775.png' 'aug_0_4778.png'
'aug_0_6093.png' 'aug_0_1316.png' 'aug_0_374.png' 'aug_0_7731.png'
'aug_0_6636.png' 'aug_0_9439.png' 'aug_0_7850.png' 'aug_0_8797.png']

```

[0117]

위의 파일들은 pre=0, post=1이라고 하면, 아래와 같은 형태의 바이너리 어레이(binary array)가 된다.

FIG 12

[0118]

```

print(Y_tune)

[0. 1. 1. 1. 0. 0. 0. 1. 1. 0. 1. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0.
 1. 1. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0. 1. 0. 1. 1. 0. 1. 1. 1. 0.
 1. 1. 1. 1. 0. 0. 1. 1. 0. 1. 1. 0. 0. 1. 0. 1. 1. 0. 0. 1. 1.
 1. 1. 1. 1. 1. 0. 0. 0.]

```


[0119] 즉, tuning에 file list에 들어간 aug_0_4960.png의 경우 pre(시술 전에 얻음 청음)이기에 0이고 aug_0_1335.png의 경우 post(시술 후 얻음 청음)이기에 1로 어레이(array)에 저장되어 있다.

[0120] 합성곱 신경망(CNN) 모델인 ResNET50 모델을 이용하여 본 발명에 따른 협착 예측 모델의 성능을 시험하였다.

[0121] ResNET50은 일반적인 합성곱 신경망(CNN) 모델과 같이, 입력 레이어, 합성곱 레이어, 최대 풀링(max pooling) 레이어, 평균 풀링(average pooling) 레이어 및 출력 레이어로 이루어진다. 여기서, 합성곱 레이어는 50개 레이어로 이루어져 멜 스펙트로그램(mel spectrogram)에서 영상 특징을 추출한다. 최대 풀링(max pooling) 레이어는 합성곱 레이어에서 추출된 특징을 서브 샘플링(sub-sampling)하여 시스템 안전성과 효율성을 높인다. 평균 풀링(average pooling) 레이어는 파라미터(parameter) 수를 줄인다. 출력 레이어는 아래와 같은 값을 출력한다.

[0122] 즉, 출력 레이어를 통해 출력되는 값은 50% 이상의 투석 접근로 협착에 대한 협착 예측 모델의 예측 능력 및 진단 성적에 대한 값을 출력할 수 있다. 예컨대, 아래의 예시와 같이, sensitivity, specificity, positive predictive value, negative predictive value, accuracy 등에 대한 값을 출력할 수 있다. 이를 기반으로 도 11에 도시된 바와 같은 혼동 행렬(confusion matrix) 및 ROC(receiver operation characteristic) 곡선을 획득할 수 있으며, ROC 곡선을 통해 진단능의 AUC(area under the curve) 값을 산출할 수 있다.

표 13

[0123]	TN = 70 / FP = 0 FN = 18 / TP = 72 sensitivity: 80.0 % specificity: 100.0 % Accuracy >> 88.75%
--------	--

[0124] 검증 데이터 세트에 포함된 특정 환자의 멜 스펙트로그램(mel spectrogram)으로부터 50% 이상의 투석 접근로(동정맥루 등) 협착을 의심해야 하는지 여부를 YES/NO식으로 결과를 얻을 수 있다.

[0125] 협착 예측 모델을 돌리면, 출력으로 각 멜 스펙트로그램(mel spectrogram) 마다 0, 1 표시로 50% 미만 또는 50% 이상의 협착 여부를 알 수 있다.

표 14

[0126]	['aug_0_8169.png'] [0.94346315]
--------	------------------------------------

[0127] ResNet50 모델의 경우 네트워크의 출력값이 x가 되도록 $H(x)-x$ 를 최소화하는 방향으로 학습을 진행하기 때문에, 출력이 "0.94346315"로 나왔지만, 이는 1에 근접한 값으로 이러한 경우 50% 이상 협착이 있다고 보면 된다. 이러한 경우 모델을 돌릴 때 print("YES")로 출력할 수 있다. 반대로 아래의 경우 0에 근접한 값이기에 0 또는 50% 이상의 협착이 아닌 50% 미만의 협착으로 인식하고 print("NO")로 출력할 수 있다.

표 15

[0128]	['aug_0_8464.png'] [0.05653682]
--------	------------------------------------

[0129] 50% 이상의 유의미한 협착이 의심되는 경우, 투석 접근로(동정맥루 등) 협착에 대한 추가 검사를 추천하게 되며, "YES"인 경우, "혈액 투석 접근로에 심각한 협착이 의심되므로 혈액 투석이 제대로 이루어지지 않을 가능성 있습니다. 도플러 초음파나 혈관 조영술과 같은 추가 검사를 요하므로 가까운 병원으로 방문하시길 바랍니다." 등과 같은 권장 사항이 함께 출력될 수도 있다.

[0130] 검증 데이터 세트에 포함된 특정 환자의 멜 스펙트로그램(mel spectrogram)이 50% 이상의 투석 접근로(동정맥루 등) 협착으로 의심되면, 협착 예측 모델이 얼마만큼 의심하는지를 %값으로 출력할 수 있다.

표 16

[0131]

```
['aug_0_8169.png']
[0.94346315]
```

[0132]

이 환자의 멜 스펙트로그램(mel spectrogram)의 경우, 협착 예측 모델이 94% 정도로 50% 이상의 협착이 있을 것으로 예측한다는 것을 의미한다.

[0133]

아래의 코드와 같은 ResNet50 모델을 이용한 협착 예측 모델의 학습 과정, 튜닝 과정 및 검증 과정은 아래와 같다.

표 17

[0134]

```
base_model = ResNet50(weights=None, include_top=True, input_shape=img_shape)
output = tf.keras.layers.Dense(n_classes, activation='softmax',
name='final_layer')(base_model.output)
model = tf.keras.models.Model(inputs=[base_model.input], outputs=[output])
model.summary()

n_classes = 2
epochs = 10
batch_size = 20
```

[0135]

batch_size는 샘플을 한번 학습시킬 때 사용한 샘플 개수고, epoch는 ResNet의 50개 레이어를 몇 번 앞뒤로 오고 가며 학습을 거칠지 여부이다. 즉, epochs=10이면, 전체 데이터를 10번 사용해서 학습을 하겠다는 것이다. 이러한 값들은 고정되어 있지 않으며 모델을 최적화 시키기 위해 batch_size와 특히 epoch 값을 여러 번 수정해야 한다. Epoch의 경우 값이 너무 작으면 모델이 데이터에 underfitting이 되는 경향이 발생하고 너무 크면 overfitting이 되는 문제가 발생한다.

[0136]

예시로, 100개의 멜 스펙트로그램(mel spectrogram)이 있다면 batch size가 20이니까 한번의 iteration 마다 20개의 데이터에 대해 학습하기 때문에 1 epoch = 100 / batch size = 5 iteration이 되고 40 epoch이면 200번의 iteration을 하게 된다.

[0137]

학습 시마다 model을 update하기 위해 사용한 optimizer는 아래와 같은 Keras SGD(stochastic gradient descent)를 사용하였다. optimizer 또한 사용할 수 있는 종류가 RMSprop, Adam, Adadelta 등 많으며, 본 발명에서는 SGD를 사용하였지만, SGD에 국한되지는 않는다. 협착 예측 모델을 학습시킬 때 일반적으로 learning rate은 0.1 ~ 0.01과 같은 값을 쓰며, momentum은 0.9로 많이 설정한다. 본 발명에서는 learning rate로 0.02를 설정하였다.

표 18

[0138]

```
##optimizer and loss##
opt = SGD(learning_rate=0.02, momentum=0.9, decay=1e-2/epochs)
metrics = ['accuracy']
model.compile(loss='binary_crossentropy', optimizer=opt, metrics=metrics)
```

[0139]

optimizing의 기본 원리는 학습률을 "처음에는 크게, 그리고 점점 작게"하는 것이다(참고 문헌: Qian Ning, On the momentum term in gradient descent learning algorithms, Neural networks 12.1 (1999): 145-151).

[0140]

모멘텀은 학습률의 값 자체는 같지만, 파라미터를 변경해갈 때 모멘텀 항이라는 조정항을 사용해 유사적으로 "처음에는 크게 그리고 점점 작게"라는 개념을 표현한다.

[0141]

오차 함수 E에 대한 신경망 모델의 파라미터를 θ 라 하고 θ 에 대한 E의 경사를 $\nabla_{\theta}E$, 파라미터의 차 $\Delta\theta^{(t)}$ 를 식 (1)라고 하면, 스텝 t에서 모멘텀을 사용해 파라미터를 변경해 가는 식이 (2)와 같다.

[0142] $\gamma \Delta \Theta^{(t-1)}$: 모멘텀항

[0143] 계수 $\gamma (<1)$ 는 일반적으로 0.5나 0.9와 같은 값을 설정한다.

$$[0144] \Delta \Theta^{(t)} = \Delta \Theta^{(t-1)} - \gamma \Delta \Theta^{(t-1)} \quad (1)$$

$$[0145] \Delta \Theta^{(t)} = -\eta \nabla_{\Theta} E(\Theta) + \gamma \Delta \Theta^{(t-1)} \quad (2)$$

[0146] 즉, learning rate과 momentum을 이와 같이 설정하면, 초기 epoch에 빠른 속도로 정확도를 올릴 수 있다.

[0147] 이렇게 파라미터를 모두 설정하고, 아래와 같이 협착 예측 모델을 fitting 또는 학습을 시킨다.

표 19

```
[0148] n_points = len(filelist_train) #train data 개수 (string length)
nb_tune_samples = len(filelist_tune) #tune data 갯수

model_history = model.fit(generator_train_fx(),
                           steps_per_epoch = n_points // batch_size,
                           epochs=epochs,
                           verbose=1,
                           callbacks=callbacks_list,
                           validation_data=generator_tune_fx(),
                           validation_steps = nb_tune_samples // batch_size)

def generator_train_fx():
    while True:
        for i in range(len(filelist_train) // batch_size): #step
            batch_img = np.zeros((batch_size, img_height, img_width, img_channel))
            batch_smk = np.zeros((batch_size, 2), dtype=np.float16)
            for j in range(batch_size): #batch size
                filename = filelist_train[i*batch_size+j]
                label = Y_train[i*batch_size+j]
                img = read_img(get_img_path(filename, label), img_height, img_width)

                if label == 1.0: #post
                    batch_smk_tmp=[1., 0.]
                elif label == 0.0: #pre
                    batch_smk_tmp=[0., 1.]
                batch_img[j] = img
                batch_smk[j] = batch_smk_tmp
            yield batch_img, batch_smk
```

[0149] pre, post 파일로 나누어서 이미 멜 스펙트로그램(mel spectrogram)을 구분해서 저장해 놓았기 때문에, PRE_PATH + filename에서 불러오는 경우 1, POST_PATH + filename에서 불러오는 경우 0으로 학습을 하면서 모델을 만든다.

[0150] Tuning 또는 미세 조정 단계에서는 epoch 마다 완성된 모델이 tuning-set 데이터를 넣었을 때 보이는 정확도(accuracy)를 토대로 가장 tuning-set 데이터에 대한 accuracy가 높은 모델을 고르게 된다. 예를 들어, 시험용으로 epochs = 10 으로 설정하고 돌리면 다음과 같은 결과를 얻을 수 있다.

표 20

[0151]

```

Epoch 1/10
27/27 [=====] - 49s 1s/step - loss: 0.6919 - accuracy: 0.5981 -
val_loss: 0.6877 - val_accuracy: 0.5625
Epoch 2/10
27/27 [=====] - 36s 1s/step - loss: 0.6827 - accuracy: 0.5981 -
val_loss: 0.6857 - val_accuracy: 0.5625
Epoch 3/10
27/27 [=====] - 36s 1s/step - loss: 0.6713 - accuracy: 0.5981 -
val_loss: 0.6853 - val_accuracy: 0.5625
Epoch 4/10
27/27 [=====] - 37s 1s/step - loss: 0.6296 - accuracy: 0.7584 -
val_loss: 0.6860 - val_accuracy: 0.5625
Epoch 5/10
27/27 [=====] - 37s 1s/step - loss: 0.5501 - accuracy: 0.9664 -
val_loss: 0.6908 - val_accuracy: 0.5625
Epoch 6/10
27/27 [=====] - 37s 1s/step - loss: 0.4679 - accuracy: 0.9205 -
val_loss: 0.7316 - val_accuracy: 0.5625
Epoch 7/10
27/27 [=====] - 37s 1s/step - loss: 0.3894 - accuracy: 0.9239 -
val_loss: 0.7642 - val_accuracy: 0.5625

Epoch 00007: ReduceLRonPlateau reducing learning rate to 0.009999999776482582.
Epoch 8/10
27/27 [=====] - 37s 1s/step - loss: 0.3528 - accuracy: 0.9278 -
val_loss: 0.4207 - val_accuracy: 0.8500
Epoch 9/10
27/27 [=====] - 37s 1s/step - loss: 0.2593 - accuracy: 0.9831 -
val_loss: 0.3553 - val_accuracy: 0.9000
Epoch 10/10
27/27 [=====] - 37s 1s/step - loss: 0.2822 - accuracy: 0.9471 -
val_loss: 0.5197 - val_accuracy: 0.8000

```

[0152]

여기서, accuracy는 training-set에 대한 모델의 accuracy이며, val_accuracy가 tuning-set에 대한 모델의 accuracy이다. 위에서 설명했듯이 epoch값이 작으면 underfitting의 문제가 발생하고, epoch값이 크면 overfitting의 문제가 발생할 수 있다. 따라서, Epoch 1/10에서 epoch 10/10으로 갈수록 accuracy는 매우 호전(0.5981 → 0.9471)되지만 tuning-set에 대한 val_accuracy는 epoch 9/10에서 peak를 찍고 epoch 10/10에서는 0.800으로 다소 감소하는 것을 알 수 있다. 그 이유는 training set melspectrogram에 너무 overfitting된 나머지, tuning set melspectrogram을 입력했을 때 fitting이 잘되지 않아 accuracy가 감소하기 때문이다. 따라서, 통상적으로 accuracy와 val_accuracy가 가장 좋은 epoch의 model을 결정하는 것이 tuning 단계이다. 위의 예시에서는 Epoch 9/10 모델로 결정하는 것이 tuning이 된다.

[0153]

그래서 Epoch 9 train weight를 결정한 뒤 validation set에 model을 적용해서 예측을 얼마나 잘하는지를 살펴본다.

표 21

[0154]

```

model.load_weights('/content/drive/My
Drive/AVFstudy/weights/20210112412/train_weights_epoch_009.h5')
#change the file directory of the selected weights
Y_pred = model.predict(generator_validation_fx(), steps=len(filelist_val)//batch_size + 1)
Y_pred = Y_pred[: len(filelist_val),:]
print(filelist_val)
print(Y_pred)

```

[0155] 그러면, 위에서 설명한 출력값을 얻을 수 있다.

[0158] 본 실시예들에 따른 동작은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능한 저장 매체에 기록될 수 있다. 컴퓨터 판독 가능한 저장 매체는 실행을 위해 프로세서에 명령어를 제공하는데 참여한 임의의 매체를 나타낸다. 컴퓨터 판독 가능한 저장 매체는 프로그램 명령, 데이터 파일, 데이터 구조 또는 이들의 조합을 포함할 수 있다. 예컨대, 자기 매체, 광기록 매체, 메모리 등이 있을 수 있다. 컴퓨터 프로그램은 네트워크로 연결된 컴퓨터 시스템 상에 분산되어 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수도 있다. 본 실시예를 구현하기 위한 기능적인(Functional) 프로그램, 코드, 및 코드 세그먼트들은 본 실시예가 속하는 기술 분야의 프로그래머들에 의해 용이하게 추론될 수 있을 것이다.

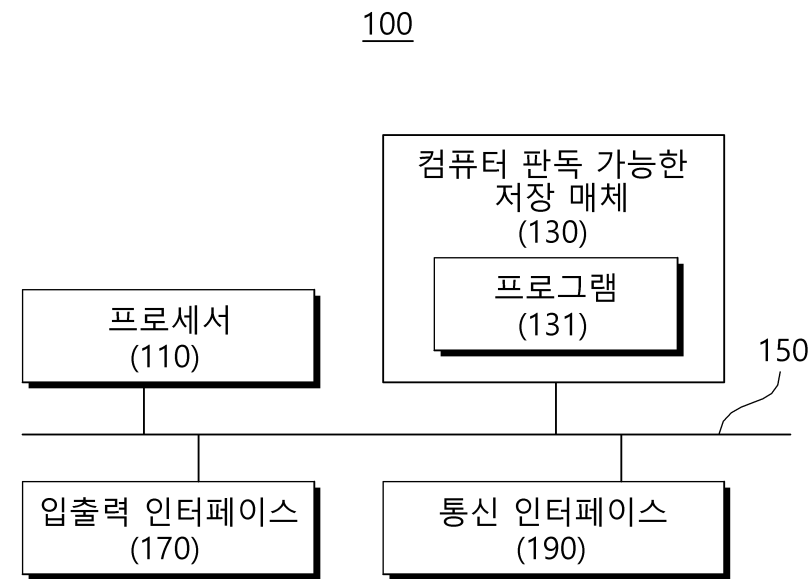
[0159] 본 실시예들은 본 실시예의 기술 사상을 설명하기 위한 것이고, 이러한 실시예에 의하여 본 실시예의 기술 사상의 범위가 한정되는 것은 아니다. 본 실시예의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 실시예의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

부호의 설명

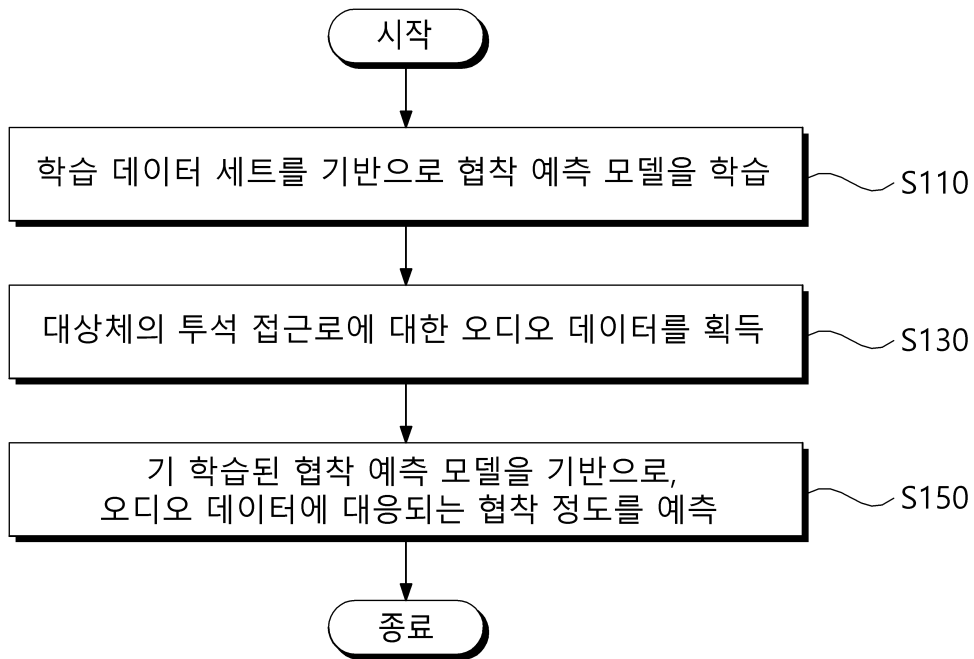
- [0160] 100 : 협착 예측 장치,
110 : 프로세서,
130 : 컴퓨터 판독 가능한 저장 매체,
131 : 프로그램,
150 : 통신 버스,
170 : 입출력 인터페이스,
190 : 통신 인터페이스

도면

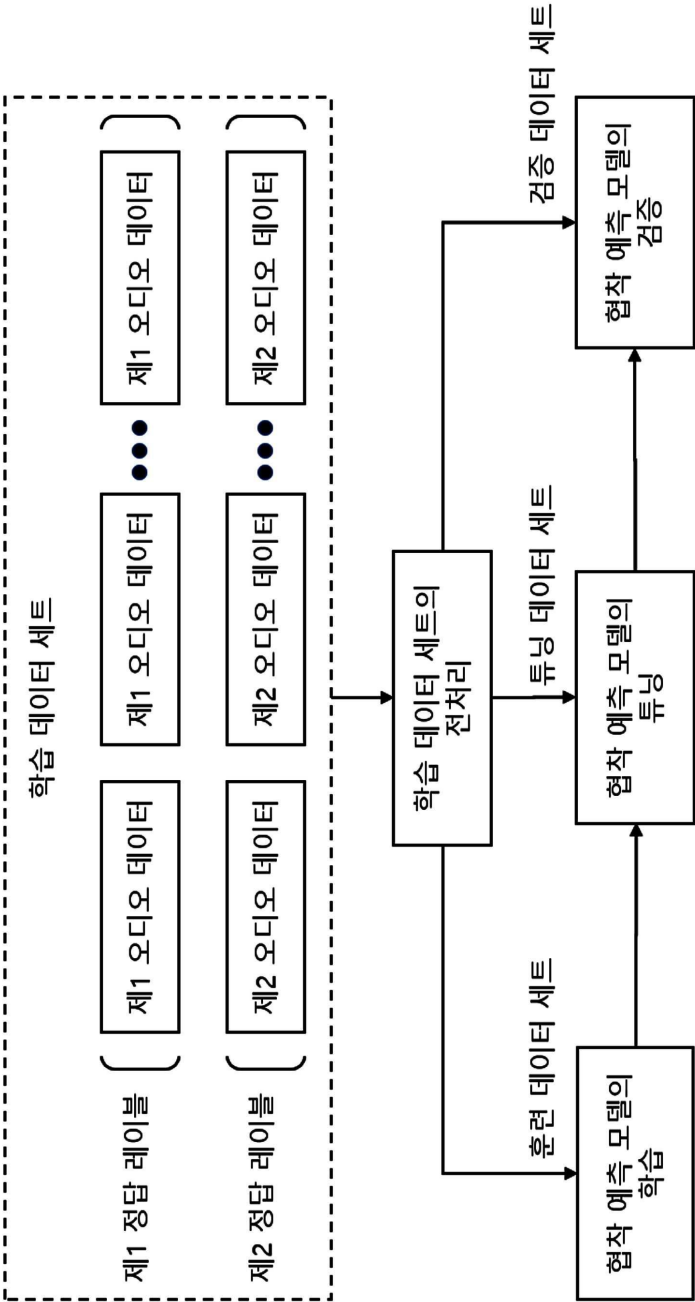
도면1



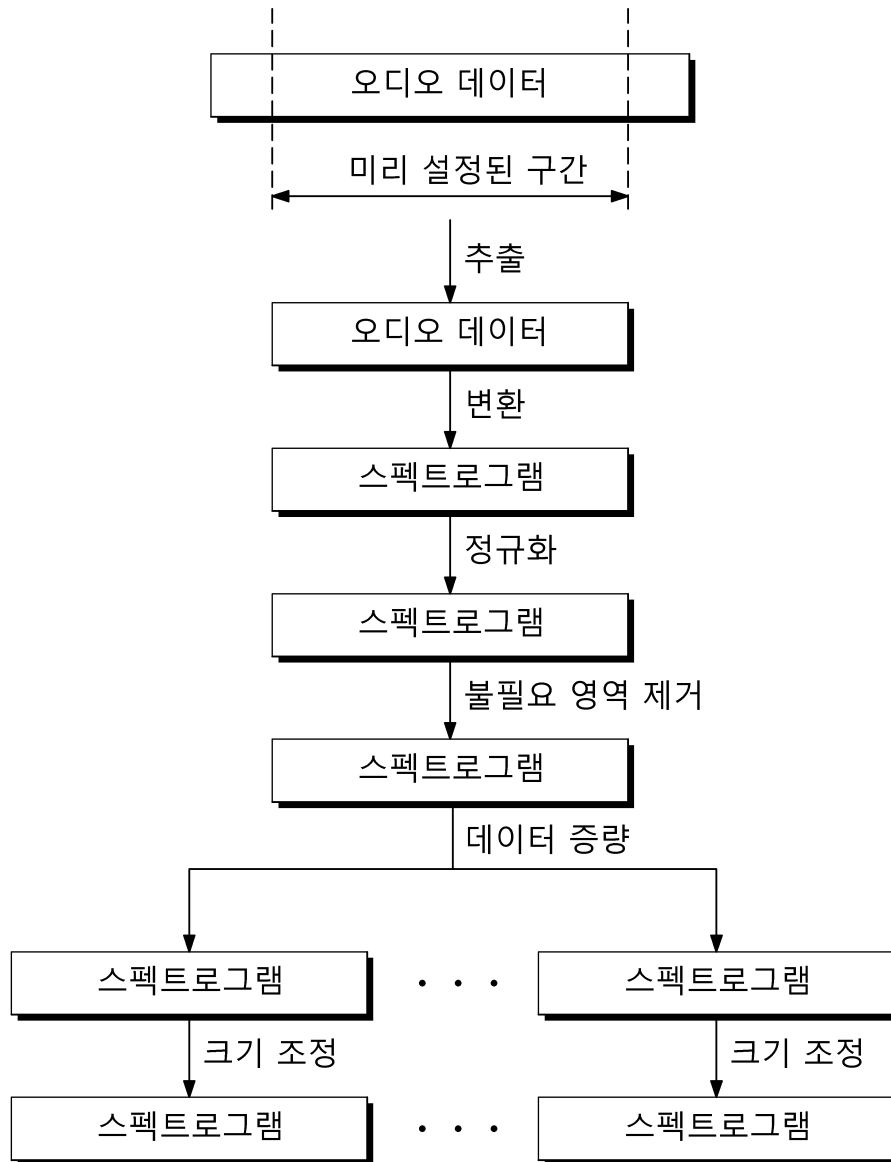
도면2



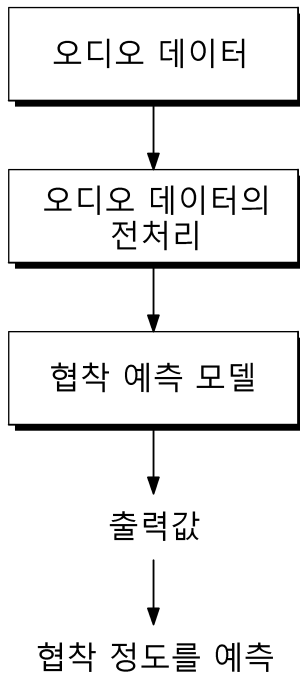
도면3



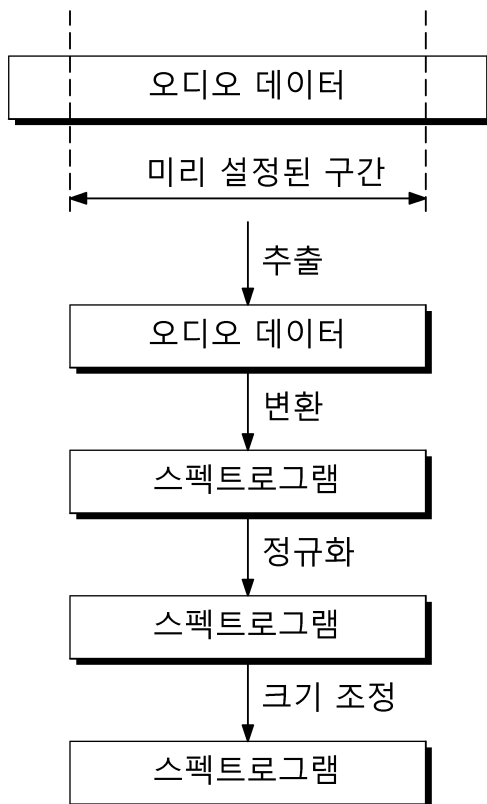
도면4



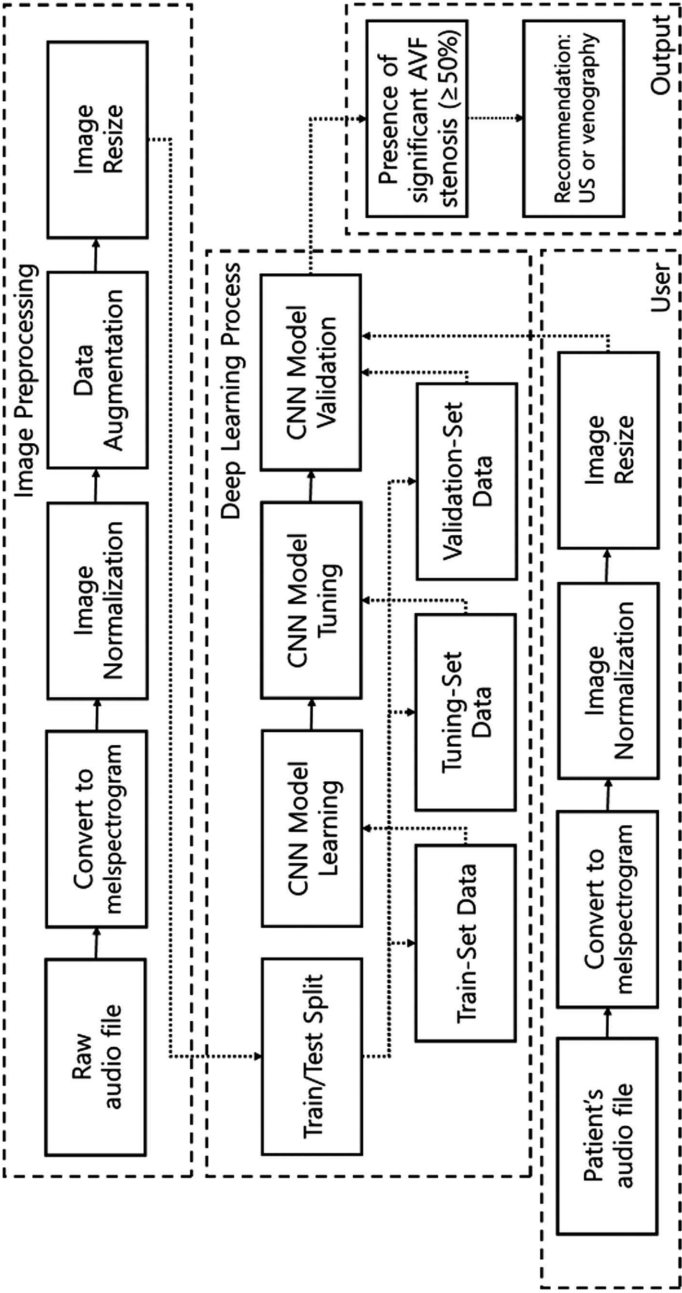
도면5



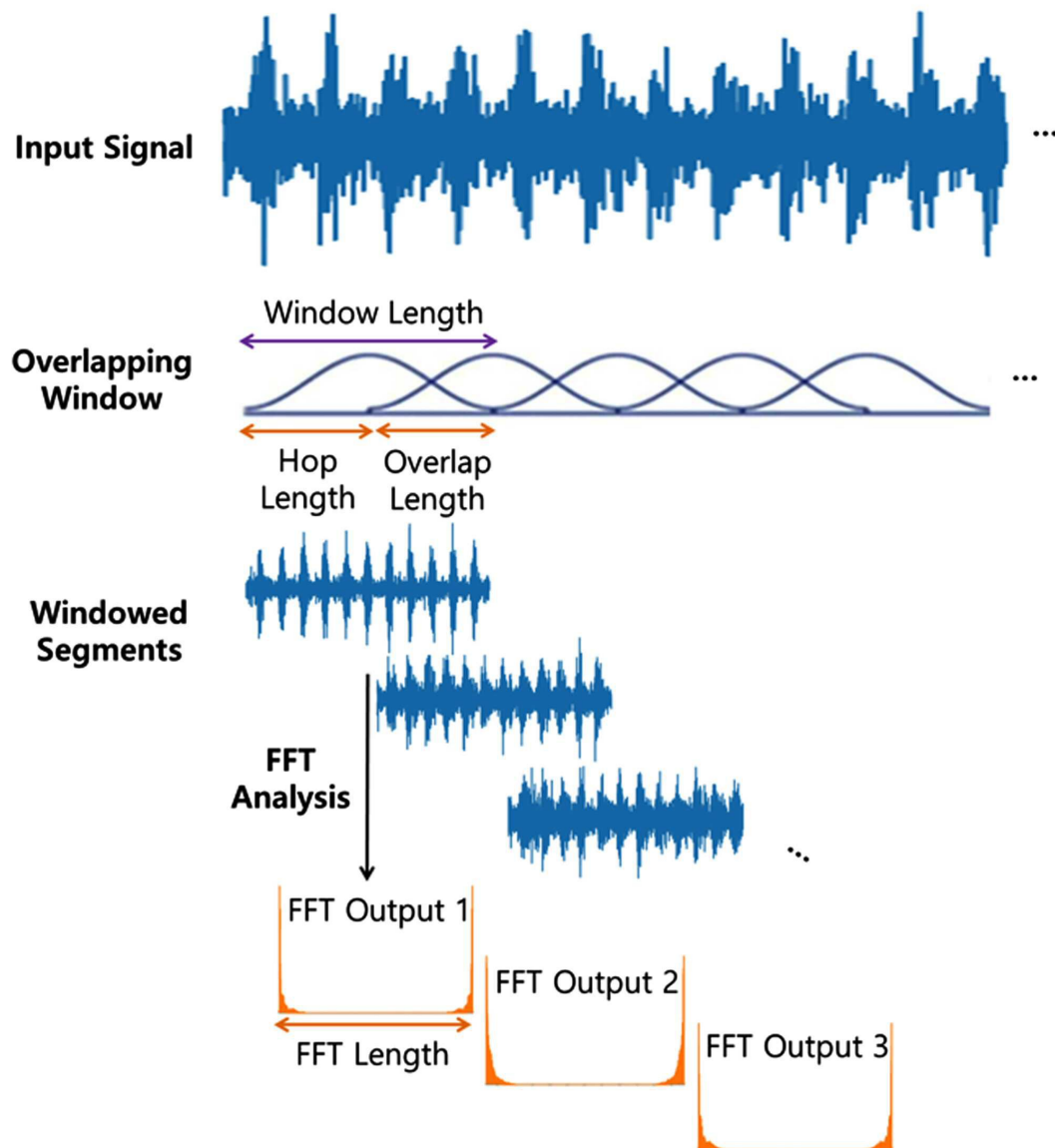
도면6



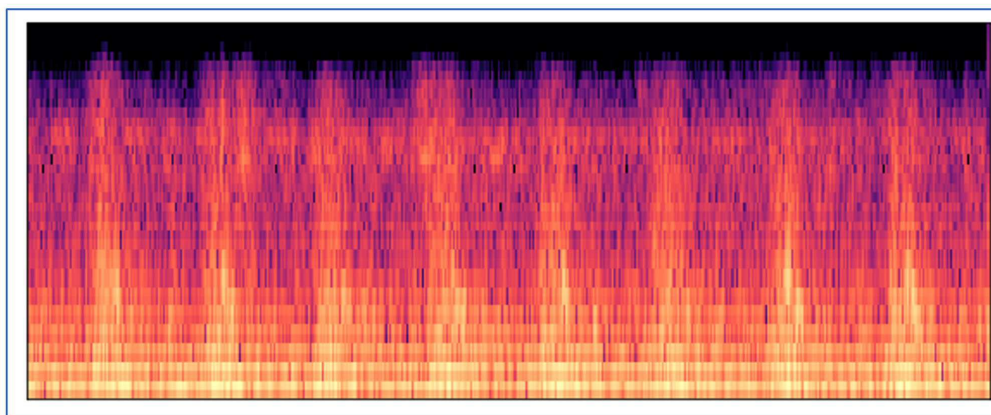
도면7



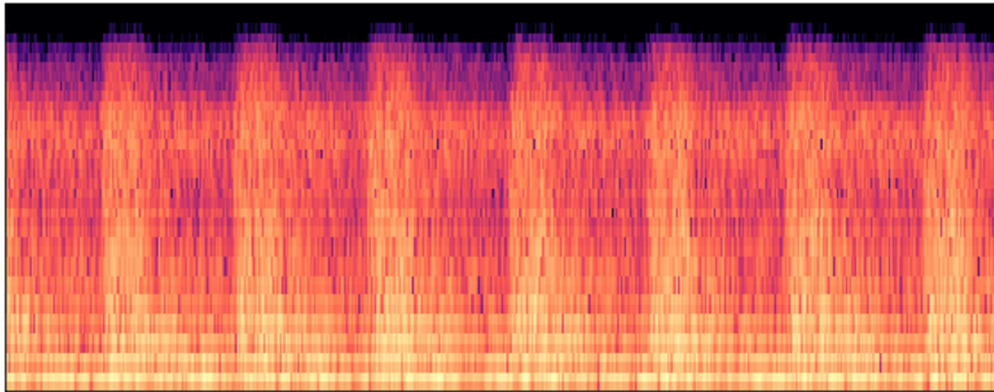
도면8



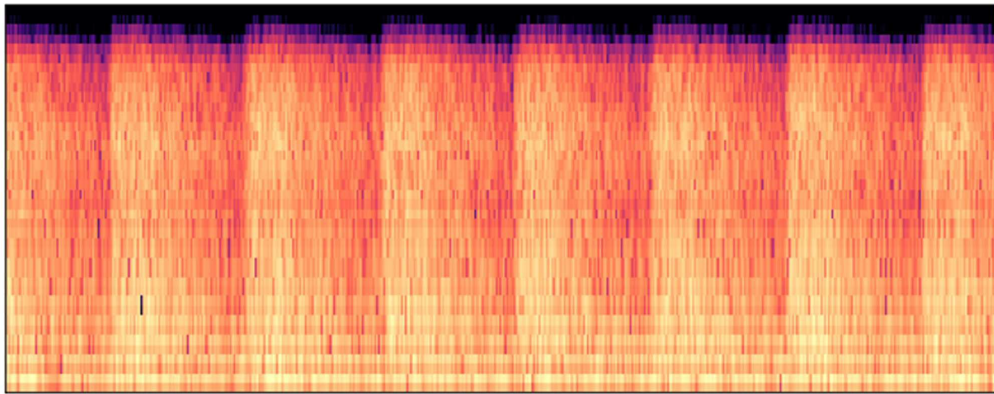
도면9



도면10

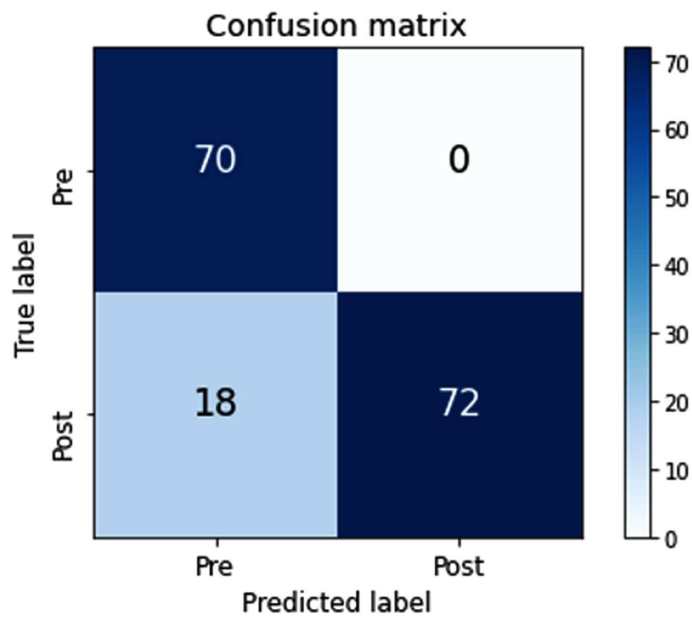


(a)

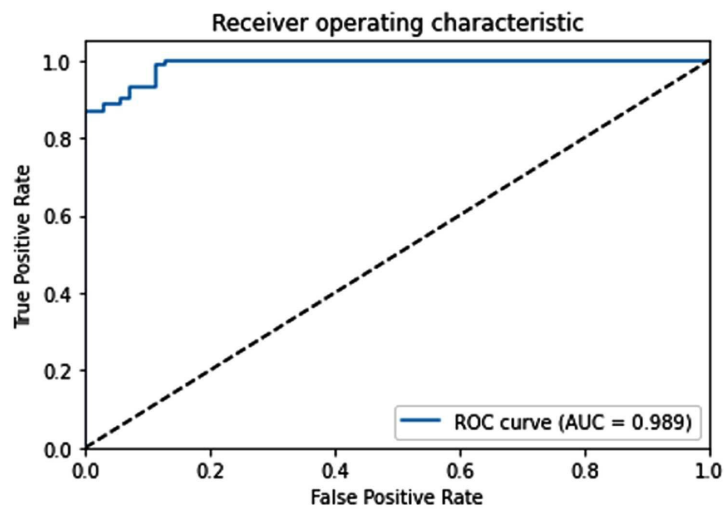


(b)

도면11



(a)



(b)