

2023년 제2회 K-ium 의료 인공지능 경진대회

Efficientnet기반의 뇌동맥류 여부 및 위치 진단 모델링

AIM

김지희¹, 명재홍², 이예승², 정계훈², 최호준²

¹강릉원주대학교 산업공학과

²강릉원주대학교 산업경영공학과

초록

본 팀에서는 익명화된 뇌혈관조형술 영상을 기반으로 뇌동맥류 여부 및 위치를 진단하는 SW를 python 프로그래밍 언어를 통해서 Efficientnet을 구축하였고, 익명화 의료 데이터셋 활용 인공지능(AI) 소프트웨어를 개발하기 위한 목적에서 시작하였다. 각 환자별 총 8개의 뇌혈관조형술 영상이 제시된 환경에서 train_set 폴더에 위치한 뇌혈관영상 8장을 환자의 인덱스에 맞게 분류하여 각각 환자에 맞는 디렉토리로 생성하여 추출하였다. 앞서 만든 디렉토리에 할당된 8장의 영상을 하나로 합쳐 w*h*d 형태인 3차원 데이터로 만들고 뇌동맥류 여부 데이터와 뇌동맥류 위치 진단 데이터를 각각 라벨링한 후 hdf5파일로 저장하는 과정을 수행하였다. 네트워크가 데이터를 처리하기 쉽게 하기위해 저장된 hdf5 파일을 0~1 사이의 실수값으로 정규화함으로써 색이 갖는 분포가 일정해진다. 뇌동맥류 여부 진단의 경우 정규화된 파일로 Efficientnet 모델을 학습한 후, 예측 결과를 출력하였다. 뇌동맥류 위치 진단의 경우 저장된 hdf5 파일을 정규화하고 정규화된 파일로 Efficientnet 모델을 학습한 후, 예측 결과의 각 변수마다 평균과 표준편차를 더하여 임계값을 설정하고 예측 결과를 임계값에 따라 0 또는 1로 변환하여 결과값을 출력하였다.

Abstract

Our team developed an AI software using anonymized cerebral angiography videos and Python programming language, aimed at diagnosing the presence and location of intracranial aneurysms. We constructed an EfficientNet model by leveraging a dataset of anonymized medical images. In this dataset, eight cerebral angiography images were provided for each patient. We categorized and extracted these images, aligning them with the corresponding patient indexes, and created directories for each patient. The eight images assigned to each patient were combined into a three-dimensional dataset of shape (width \times height \times depth). Subsequently, we labeled the dataset with information regarding the presence of intracranial aneurysms and their respective locations. The labeled dataset was then saved as an HDF5 file. To facilitate data processing by the network, we normalized the stored HDF5 file, scaling the values between 0 and 1, which ensured a consistent color distribution. For the diagnosis of cerebral aneurysm presence, we trained an EfficientNet model using the normalized file and generated predictions. As for the diagnosis of aneurysm location, we normalized the saved HDF5 file and trained an EfficientNet model using the normalized data. We computed the mean and standard deviation for each variable in the prediction results and established threshold values. Based on these thresholds, we transformed the prediction results into either 0 or 1 and outputted the final results accordingly.

Key Words : 뇌혈관조영술, hdf5, Efficientnet, 3차원 데이터

I. 서 론

주어진 데이터셋(2023_k_jum_composition)에는 환자별 디지털 감산 혈관조영술 표준 영상과 뇌동맥류의 존재 및 위치 정보를 담고 있다.

1. train_set

1) 환자별 디지털 감산 혈관조영술 표준 영상의 구성

- [Index]LI-A.jpg와 [Index]LI-B.jpg
→ 왼쪽 내경동맥에서 조영제를 주입하고 얻은 영상(서로 다른 각도 촬영)
- [Index]RI-A.jpg와 [Index]RI-B.jpg
→ 상기 LI-A, LI-B와 동일하나 오른쪽 내경동맥에서의 검사
- [Index]LV-A.jpg와 [Index]LV-B.jpg
→ 왼쪽 추골동맥에서 조영제를 주입하고 얻은 영상
- [Index]RV-A.jpg와 [Index]RV-B.jpg
→ 마찬가지로 LV-A, LV-B와 동일하나 오른쪽 추골동맥에서의 검사

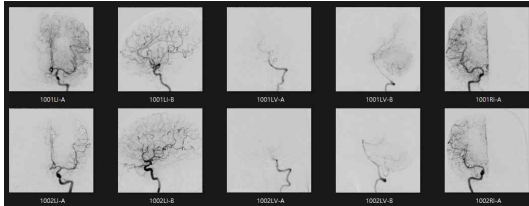


Fig. 1. 환자별 뇌혈관조영술 영상

2) 뇌동맥류의 존재 및 위치 정보의 구성

Index	Aneurysm	L_ICA	R_ICA	L_PCOM	R_PCOM	L_AntChor	R_AntChor	L_JCA	R_JCA	L_PCOM	R_PCOM	L_JCA	R_JCA	L_VA	R_VA	L_PICA	R_PICA	L_SCA	R_SCA	BA	L_PCA	R_PCA
1007	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1008	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1009	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1010	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1012	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1013	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1014	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1015	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1016	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1017	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1018	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1019	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1020	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 2. 환자별 뇌동맥류의 존재 및 위치 정보

(1) 뇌동맥류 여부 (1=TRUE 0=FALSE)

- Aneurysm

(2) 뇌동맥류 위치 정보 (1=Presence 0=Absence)

- L_JCA = left internal carotid artery
- R_JCA = right internal carotid artery
- L_PCOM = left posterior communicating artery
- R_PCOM = right posterior communicating artery
- L_AntChor = left anterior choroidal artery
- R_AntChor = right anterior choroidal artery

- L_ACA = left anterior cerebral artery
- R_ACA = right anterior cerebral artery
- L_ACOM = left-sided anterior communicating artery
- R_ACOM = right-sided anterior communicating artery
- L_MCA = left middle cerebral artery
- R_MCA = right middle cerebral artery
- L_VA = left vertebral artery
- R_VA = right vertebral artery
- L_PICA = left posterior inferior cerebellar artery
- R_PICA = right posterior inferior cerebellar artery
- L_SCA = left superior cerebellar artery
- R_SCA = right superior cerebellar artery
- BA = basilar artery
- L_PCA = left posterior cerebral artery
- R_PCA = right posterior cerebral artery

2. test_set

1) 뇌동맥류의 존재 및 위치 정보의 구성

Index	Aneurysm	L_ICA	R_ICA	L_PCOM	R_PCOM	L_AntChor	R_AntChor	L_JCA	R_JCA	L_PCOM	R_PCOM	L_JCA	R_JCA	L_VA	R_VA	L_PICA	R_PICA	L_SCA	R_SCA	BA	L_PCA	R_PCA
1021	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1022	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1023	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1024	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1025	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1026	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1027	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1028	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1029	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1030	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1031	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1032	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1033	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1034	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1035	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1036	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1037	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1038	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1039	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1040	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1041	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1042	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 2. 환자별 뇌동맥류의 존재 및 위치 정보

이 파일은 뇌동맥류 존재 여부와 위치 정보 모두 0으로 코딩되어 있다. 상기 test.csv에 0으로 코딩되었던 변수들에 참가팀의 인공지능 모형이 제시한 값들을 포함되어 출력한 파일이 output.csv 파일이다.

(1) 뇌동맥류 여부 (1=TRUE 0=FALSE)

- 뇌동맥류 여부에 대한 모형의 결과 값은 0과 1 사이의 실수로 출력한다.

(2) 뇌동맥류 위치 정보 (1=Presence 0=Absence)

- 21개의 위치 변수의 결과 값은 모두 0 또는 1의 정수로 출력한다.

본 팀이 구성한 프로그램은 'Anaconda'에서 지원하는 'jupyter notebook' 환경에서 작성되었으며, 환경 구성은 아래와 같다.

- 플랫폼 : jupyter notebook
- GPU : NVIDIA GeForce RTX 3080
- GPU API : CUDA 11.2 (cudnn 8.1)

> conda env export -n mee -f environment.yml

위의 명령어를 실행하면 현재 활성화된 mee라는 이름의 Conda 환경에 관한 모든 정보가 environment.yml 파일로 내보내진다. 이 environment.yml 파일은 해당 환경에 설치된 모든 패키지와 버전, 그리고 필요한 설정 정보를 포함한다. 다른 사람이나 다른 컴퓨터에서 environment.yml 파일을 사용하여 동일한 환경을 생성하면, 동일한 패키지와 버전이 설치되어 원하는 환경을 재현할 수 있다.

> conda env create -f environment.yml

또한 environment.yml 파일을 공유하고, 위의 명령어를 실행하면 다른 사람이나 다른 컴퓨터에서 동일한 Conda 환경을 생성할 수 있다.

본 팀이 사용한 언어 및 라이브러리 버전은 다음과 같다.

1. 언어
 - Python 3.7.16
2. 라이브러리
 - numpy

Version: 1.21.6

License: BSD-3-Clause
- pandas

Version: 1.3.5

License: BSD-3-Clause
- tensorflow

Version: 2.10.0

License: Apache 2.0
- sklearn

Version: 1.0.2

License: BSD-3-Clause
- h5py

Version: 3.8.0

License: BSD 3-Clause

3. 사전학습 모델
 - EfficientnetB0

License: Apache 2.0

II. 관련 기술

인공지능 기술은 의료 영상 분석 분야에서 다양한 방식으로 활용되어 의료 전문가들의 진단과 치료 계획 수립에 도움을 주고 있다. 먼저, 영상 보조 검사 측면에서 인공지

능은 의료 영상을 정교하게 분석하여 의사나 방사선 전문가들이 진단을 더욱 정확하게 내릴 수 있도록 보조 역할을 한다. 이러한 기술은 의료 전문가들의 의사결정에 신속하고 높은 정확성을 제공하여 환자의 질병을 조기에 발견하고 적절한 치료 방안을 채택할 수 있도록 도움을 준다. 이러한 인공지능 기술의 도입으로 의료 분야의 정확성과 효율성을 크게 향상시키는 연구결과가 계속해서 발표되고 있다. 따라서, 본 보고서에서는 프로그램 생성 과정에서 활용된 기술에 대해 설명하고자 한다.

II - I. EfficientNet

본 프로그램에서 사용한 EfficientNet은 Google에서 개발한 컨볼루션 신경망(CNN) 아키텍처로, 이미지 분류 및 관련 작업에 사용되는 딥러닝 모델이다. EfficientNet은 모델의 크기를 조절하는 계수를 사용하여 모델의 효율성과 정확성을 균형있게 조절하는 효율적인 스케일링 방법을 제공한다. 또한 EfficientNet은 ImageNet 데이터셋에서 뛰어난 성능을 보여주었으며, 이를 기반으로 전이 학습(transfer learning)을 통해 다양한 컴퓨터 비전 작업에 활용될 수 있습니다. CNN에서 스케일링을 하는 대표적인 방법인 필터(채널)의 개수를 늘리는 wide scaling, layer의 개수를 늘리는 depth scaling (ResNet이 대표적), input이미지의 해상도를 높이는 resolution scaling을 조절하면서 모델을 설정한다. EfficientNet은 앞서 말한 3개의 scaling을 적절하게 균형을 잡는 compound scaling 방법을 사용한다. EfficientNet은 B0~B7의 모델로 구성되어있는데, 숫자가 커질수록 모델이 무거워지며, B0~B7의 차이를 만드는 것은 wide, depth, resolution과 dropout 총 4개의 하이퍼파라미터 값이다.

	width_ coefficient	depth_ coefficient	default_ resolution	dropout_ rate	output size
B0	1	1	224	0.2	1280
B1	1	1.1	240	0.2	1280
B2	1.1	1.2	260	0.3	1408
B3	1.2	1.4	300	0.3	1536
B4	1.4	1.8	380	0.4	1792
B5	1.6	2.2	456	0.4	2048
B6	1.8	2.6	528	0.5	2304
B7	2.0	3.1	600	0.5	2560

Table 1. B0-B7 모델 파라미터 기본값

EfficientNet B0는 EfficientNet 시리즈의 가장 기본이 되는 모델로, 넓이, 깊이 및 해상도 스케일링 계수가 모두 1로 설정되어 있고 비교적 작은 모델이지만 경량화된 구조와 효율적인 특징 추출 방식을 통해 다양한 컴퓨팅 환경에서 효과적으로 활용할 수 있다.

II - II. Adam

Adam(Adaptive Moment Estimation)은 딥러닝 학습에 널리 사용되는 최적화 알고리즘 중 하나로, 기울기 기반 최적화 알고리즘이다. Adam은 기존의 경사하강법 알고리즘에 모멘텀 최적화와 RMSProp 알고리즘을 결합한 형태이다. 이를 통해 학습 과정에서 빠른 수렴과 안정적인 최적화를 동시에 이루어낼 수 있다. Adam의 주요 특징과 작동 방식은 다음과 같다.

- 모멘텀 최적화: 기울기 하강법과 같이 이전 업데이트 방향을 보존하여 효과적인 경사 하강을 돕는 모멘텀 최적화를 도입한다. 모멘텀은 이전 업데이트의 방향과 크기를 보존함으로써 지역 최적점에서 빠르게 탈출하고 전역 최적점으로 수렴하도록 도움을 준다.
- RMSProp: 기존의 최적화 알고리즘 중 하나인 RMSProp은 학습률을 조정하여 학습의 안정성을 높이는 역할을 한다. 기울기의 크기에 따라 학습률을 조절하여 더 작은 학습률일수록 더 안정적으로 학습할 수 있다.
- 학습률 스케일링: Adam은 학습률을 동적으로 조정하여 학습 과정을 안정화하는데 기여한다. 초기 학습률에 영향을 받지 않고 적절한 학습률을 계산하여 적용한다.

Adam은 주로 딥러닝에서 사용되며, 매우 큰 데이터셋에서도 잘 작동하며, 빠른 수렴과 안정적인 학습을 제공하는 최적화 알고리즘이다.

II - III. binary_crossentropy

binary_crossentropy는 주로 이진 분류 문제에서 사용되는 손실 함수(loss function)이다. 이 함수는 신경망 모델이 이진 분류 작업을 수행할 때 예측값과 실제 라벨 사이의 차이를 측정하여 모델의 성능을 평가하고 개선하는 데 사용된다. 이 함수는 크로스 엔트로피(cross-entropy)의 이진 버전이며, 두 확률 분포 사이의 차이를 측정한다. 이진 분류에서는 보통 두 개의 클래스가 존재하며(예: 0과 1), binary_crossentropy 함수는 이 두 클래스에 대한 확률 분포 사이의 차이를 측정한다. 일반적으로, 모델의 출력은 0과 1 사이의 실수값으로 나타내며, 이 값을 확률값으로 해석하여 각 클래스에 속할 확률을 나타낸다. 보통 이진 분류 작업에서 높은 정확성을 얻는 데 도움을 준다. 이를 통해 모델이 정확한 예측을 하고, 분류 작업에서 뛰어난 성능을 발휘하도록 학습된다.

II - IV. HDF5

HDF5(Hierarchical Data Format version 5)는 대량의 데이터와 복잡한 계층 구조를 지원하는 파일 형식이다. 주로 의료 영상 데이터나 시계열 데이터, 실험 결과 등의 다양한 형태의 데이터를 효과적으로 저장하여 분석하고 처리하는 데 활용된다. HDF5 파일은 다차원 배열, 메타데이터, 속성 등을 저장하고 효율적으로 관리할 수 있다. HDF5 파일의 장점은 다음과 같다.

- 확장성: 대규모 데이터를 처리하거나 다양한 유형의 데이터를 저장하기 용이하다. 또한, 데이터 요소들을 계층 구조로 구성하여 데이터의 구조를 유연하게 확장할 수 있다.
- 효율성: 데이터 압축 기능을 활용하여 데이터 입출력 속도를 향상시킨다. 이는 대량의 데이터를 빠르게 읽고 쓸 수 있도록 도와준다.
- 이식성: 다양한 운영 체제와 프로그래밍 언어에서 지원되어 데이터의 이식성이 높다. 이러한 특성은 다양한 환경에서 데이터를 공유할 수 있게 한다.

III. 데이터 전처리

```
import os
import re
import shutil

image_folder =
"C:/Users/2023_k_jum_composition/train_set/train_set"
output_folder = 'C:/Users/Downloads/classified_images'

os.makedirs(output_folder, exist_ok=True)

classified_folders = {}

for filename in os.listdir(image_folder):
    file_path = os.path.join(image_folder, filename)
    matches = re.findall(r'\d{4}', filename)
    if len(matches) > 0:
        image_index = matches[0]
    else:
        image_index = "others"
    class_folder = os.path.join(output_folder, image_index)
    if image_index not in classified_folders:
        os.makedirs(class_folder, exist_ok=True)
        classified_folders[image_index] = class_folder
    shutil.copy(file_path, class_folder)
```

- image_folder: 이미지 데이터가 저장된 폴더 경로를 지

정한다.

- output_folder: 이미지 데이터를 분류하여 저장할 폴더 경로를 지정한다. 이곳에 환자 Index별로 디렉토리가 생성된다.
- os.makedirs: output_folder 경로에 디렉토리를 생성한다.
- classified_folders: 이미지 파일을 분류한 디렉토리들을 저장할 디렉터리 변수이다.

이후 반복문을 통해 이미지 데이터 폴더의 각 파일을 순회한다. 각 파일의 이름에서 숫자 네 자리를 추출하여 환자 Index로 사용한다. 이후 해당 Index를 이름으로 하는 디렉토리를 생성하고, 해당 디렉토리로 이미지 파일을 복사한다. 이 코드를 통해 이미지 데이터를 환자 Index별로 분류하여 디렉토리를 생성하고, 각 디렉토리에 해당하는 이미지 파일을 복사할 수 있다.

위의 과정을 거쳐 생성된 환자별 디렉토리 속 8장의 이미지들을 한 장의 이미지로 압축하여 효율적으로 분석하기 위해 hdf5 파일 형태로 변환하였다. 변환하는 이유는 흑백 이미지의 RGBA(R은 빨간색, G는 초록색, B는 파란색, A는 투명도) 채널을 조절하면 이미지 정보 손실의 위험이 있기 때문이다. 흑백 이미지는 보통 단일 채널(그레이스케일)로 구성되어 있기 때문에 RGB 채널의 값이 흑백 이미지에 대해 의미를 가지지 않는다.

따라서, 흑백 이미지의 구조나 모양에 영향을 주지 않기 위해 hdf5 파일 형태로 8장의 뇌 이미지를 1장의 이미지로 결합하였다. 이로 인해 이미지 데이터의 일관성과 완전성을 유지하면서 원하는 작업을 수행할 수 있고 효율적인 데이터 저장, 계층적 구조 지원, 다양한 데이터 타입 지원, 확장성 및 이식성을 제공하여 이미지 데이터 관리 및 처리 작업을 향상시킬 수 있다. hdf5 파일을 생성함으로써 이미지를 (256, 256, 8) 크기의 3차원 형태로 이미지를 변환하여 1127(총 환자 수)*8개(각 환자당 제시되는 이미지 개수)의 이미지들을 1127개로 축소시킨 후 뇌동맥류 여부(Aneurysm 값인 0(=FALSE)과 1(=TRUE)을 3차원 이미지에 라벨링 하였다.

다음으로, 이미지 데이터를 0과 1 사이의 범위로 조정하기 위해 이미지 데이터를 255로 나누는 정규화 작업을 수행하였다. 이미지 정규화는 데이터를 특정 범위로 조정하여 모델 학습에 도움을 주는 중요한 전처리 과정이다. 이 작업을 수행한 이유는 이미지 데이터가 8비트로 표현되는 경우, 픽셀값의 최댓값이 255이기 때문이다. 따라서, 모든 픽셀값을 0과 1 사이의 값으로 변환하기 위해 255로 나누었다.

IV. 모델 생성

Table 1.에서 B0값을 default로 설정하면 EfficientNet B0 모델이 만들어진다. input 데이터인 hdf5 파일의 크기 (256, 256, 8)에 맞게 모델의 파라미터를 아래와 같이 수정하였다.

parameter	value
input_channels	8
default_resolution	256
width_coefficient	1
depth_coefficient	1
dropout_rate	0.2

Table 2. B0 모델 파라미터 수정

EfficientNet의 핵심은 MBConvBlock(Mobile Inverted Residual Bottleneck Block)인데 입력 데이터를 처리하고 중요한 특징을 추출한다. MBConvBlock은 입력값과 다양한 매개변수를 받고 다양한 크기의 특징을 추출하고 채널 간 상호작용을 강화한다. 최종적으로, 1x1 컨볼루션과 배치 정규화를 통해 출력 채널 수를 조정하는 역할을 하는데 이러한 과정을 반복하여 EfficientNet 모델을 구성한다. 우리의 모델에서는 7개의 다른 크기의 커널을 사용하여 다양한 크기의 특징을 추출할 수 있게 설계되었다.

EfficientNet모델은 반복적으로 MBConvBlock을 적용하며 각 반복에서는 MBConvBlock 함수가 호출되고, 필요한 컨볼루션, 배치 정규화, 활성화 함수 등이 적용된다.

Stage	Operator	Resolution	Channels	Layers
1	Conv 3x3	256x256	32	1
2	MBConv1, k3x3	128x128	16	1
3	MBConv1, k3x3	128x128	24	2
4	MBConv1, k5x5	64x64	40	2
5	MBConv1, k3x3	32x32	80	3
6	MBConv1, k5x5	32x32	112	3
7	MBConv1, k5x5	16x16	192	4
8	MBConv1, k3x3	8x8	320	1
9	Conv1x1&Pooling&FC	8x8	1280	1

Table 3. B0 모델 파라미터 수정

이 과정을 통해 네트워크는 이미지에서 더 복잡한 특징을 학습할 수 있게 된다. 반복적인 MBConvBlock 적용 후에는 최종적인 분류를 위한 레이어들이 추가되는데 이 과정에는 1x1 컨볼루션, 배치 정규화, 활성화 함수, 전역 평균 풀링, 드롭아웃, 밀집 레이어가 사용되어 더 정확한 예측을 할 수 있다.

모델을 구축하는 단계에서 num_classes=2로 설정하여 분류할 클래스의 수를 표현하였는데, 이는 뇌동맥류 여부 진단에서 0과 1, 총 2개의 클래스를 구분하기 위한 설정이다. 최종 출력은 1 또는 0으로 나타내는 이진 분류 결과이다. 이를 위해 sigmoid activation function을 사용했다.

뇌동맥류 존재 여부 데이터의 모델 컴파일 과정에서는

Adam 옵티마이저를 활용하여 학습을 진행하였다. 기본적으로 제공되는 학습률은 0.001이지만, 본 프로그램에서는 학습률을 0.0001로 조정하였다. 그 이유는 학습률이 0.001과 0.0001의 두 가지 경우에 따라 모델의 성능 차이가 나타났으며, 이를 통해 학습률이 모델의 학습 및 일반화 능력에 영향을 미침을 확인하였기 때문이다. 따라서, 성능 향상을 위해 Adam 옵티마이저의 학습률을 0.0001로 설정하였다. 본 프로그램은 뇌동맥류 여부를 이진 분류하는 작업이기 때문에 binary_crossentropy를 손실 함수로 선택하였다. 이 손실 함수는 뇌동맥류의 존재 여부를 정확하게 분류하는 모델을 학습하기에 적합하며, 모델이 환자의 영상 데이터를 효과적으로 분석하고 클래스를 예측할 수 있다. 학습 단계에서 batch_size=8, epoch=10, validation_split=0.3으로 설정하여 데이터를 훈련 세트와 검증 세트로 나누고, 모델을 학습시켜 예측 결과를 출력하였다. 이를 통해 모델이 학습된 가중치를 기반으로 입력된 뇌혈관조형술 영상의 뇌동맥류 여부를 예측하고 출력하였다.

뇌동맥류 위치 데이터의 모델 컴파일 과정에서도 Adam을 사용하였으며, 손실 함수 또한 binary_crossentropy를 사용하여 학습을 진행하였다. 학습 단계에서 batch_size=8, epoch=7, validation_split=0.3으로 설정하여 데이터를 훈련 세트와 검증 세트로 나누고, 모델을 학습시켜 예측 결과(predictions)를 출력하였다. 출력된 predictions 값은 0과 1 사이의 실수값으로 구성되어 있는데, 이 실수값을 0 또는 1로 변환하기 위해 각 변수에 해당하는 평균과 표준편차를 더하여 임계값으로 설정하고 predictions을 임계값에 따라 0 또는 1로 변환하는 과정을 거쳐 뇌동맥류 위치 정보가 들어있는 데이터 프레임에 업데이트하였다.

V. 모델 훈련 결과

batch_size=8, epoch=10, validation_split=0.3의 파라미터로 뇌동맥류 존재 여부 모델 훈련 결과는 아래와 같다.

Epoch	accuracy
1	0.5241
2	0.5508
3	0.5888
4	0.6536
5	0.6967
6	0.7500
7	0.8147
8	0.8299
9	0.8579
10	0.8731

Table 4. 뇌동맥류 존재 여부 모델 훈련 결과

epoch가 늘어날수록 accuracy가 증가하는 것을 확인할 수 있다. 다음은 batch_size=8, epoch=7, validation_split=0.3의

파라미터로 뇌동맥류 위치 모델 훈련 결과이다. 위치 변수는 총 21개로 출력된 accuracy 중 가장 높은 값만 나타낸 결과이다.

variable	accuracy
L_JCA	0.9302
R_JCA	0.9480
L_PCOM	0.9860
R_PCOM	0.9835
L_AntChor	0.9835
R_AntChor	0.9937
L_ACA	0.9924
R_ACA	0.9911
L_ACOM	0.9734
R_ACOM	0.9835
L_MCA	0.9594
R_MCA	0.9505
L_VA	0.9911
R_VA	0.9937
L_PICA	0.9987
R_PICA	0.9987
L_SCA	0.9962
R_SCA	0.9975
BA	0.9860
L_PCA	0.9975
R_PCA	1.0000

Table 5. 뇌동맥류 위치 모델 훈련 결과

이 모델의 결과도 epoch가 늘어날수록 accuracy가 증가하는 것을 확인할 수 있었다.

최종 output.csv를 생성하기 위해 예측 결과를 데이터 프레임 형태로 저장함으로써 보다 구조화된 형식으로 결과를 분석할 수 있다. groundtruth.csv 데이터에는 총 50개의 정답 데이터가 포함되어 있다. 예측 결과를 groundtruth.csv와 비교하기 위해 50개의 데이터를 추출하는 과정을 수행하였다. test.csv 파일을 활용하여 데이터 프레임에 추출한 50개의 예측 결과값을 입력하는 과정을 수행한 후, 뇌동맥류 여부 및 위치 정보를 예측한 결과를 담은 CSV 파일을 생성하였다.

References

[1]

<https://github.com/swotr/tf2-keras-imagenet/blob/master/model.py>

[2]

<https://github.com/qubvel/efficientnet/blob/master/efficientnet/model.py>

[3]

<https://blog.naver.com/PostView.naver?blogId=qwopqwop200&logNo=222053989967&parentCategoryNo=&categoryNo=1&viewDate=&isShowPopularPosts=true&from=search>