

# 뇌 영상 분석 및 뇌종양 진단 학습 모델 개발



저자 1 이 섬 재

저자 2 최 연 희

저자 3 한 혜 린

지도교수 감 진 규 교수님

---

## 목 차

1. 서론 .....	1
1.1. 연구 배경.....	1
1.2. 기존 문제점 .....	1
1.3. 연구 목표.....	2
2. 연구 배경.....	2
2.1. 데이터 수집 .....	2
2.2. 개발 환경.....	2
2.3. 데이터 전처리 .....	3
3. 연구 내용.....	4
3.1. 2D 모델.....	4
3.2. 3D 모델 .....	6
3.3. 교차 검증 (Cross Validation) .....	9
3.4. 시각화 (Confusion Matrix).....	10
4. 연구 결과 분석 및 평가.....	12
5. 결론 및 향후 연구 방향 .....	14
6. 구성원별 역할 및 개발 일정.....	15
7. 참고 문헌.....	16

---

## 1. 서론

### 1.1. 연구 배경

뇌 조직에 풍부하게 존재하는 신경교세포에서 악성 종양이 발생할 경우, 환자의 생명을 크게 위협할 수 있다. 교모세포종 (Glioblastoma, GMB) 으로 알려진 이 질환은 성인에게서 발생하는 뇌종양 중 가장 흔한 유형이며, 치명률이 가장 높다. 환자의 평균 예후는 표준적인 치료에 따라 14 개월이며, 치료되지 않는다면 약 4 개월 정도로 최악의 예후를 보이는 암이다.

지금까지 임상 종양 진단은 순수 형태학적-조직병리학적 분류를 많이 따랐지만 현재 개정된 세계보건기구(WHO)는 이를 통합 분자-세포유전학적 특성으로 전환했다. 이에 따라 손상된 DNA 를 수정하는 효소인 O[6]-methylguanine-DNA methyltransferase(MGMT) 프로모터에 메틸화가 되어 있을 경우 환자들의 예후가 좋아진다는 결과가 보고되었으며, 이는 화학 요법에 대한 반응성의 강력한 예측 변수인 것으로 나타났다. 따라서 새로 진단된 교모세포종에서 MGMT 프로모터 메틸화 상태는 치료 결정에 영향을 미칠 수 있다.<sup>[1]</sup>

### 1.2. 기존 문제점

현재 암의 유전적 특징을 분석하기 위해서는 조직 샘플을 추출하는 수술 과정이 필요하다. 수술 이후에도 종양의 유전적 특징을 분석하는데 많은 시간이 소모되고, 선택한 초기 요법의 유형과 결과에 따라 후속 수술이 필요할 수도 있다.

이에 따라 영상화를 통해 암의 유전을 정확히 예측하는 방법이 개발된다면, 환자의 치료 방법을 개선하고 수술 횟수를 최소화할 수 있을 것이다. 이를 통해 환자는 수술 전 새롭고 맞춤형 치료를 받을 수 있게 될 것이며, 이는 환자의 치료 및 생명 연장에 큰 영향을 끼치게 될 것이다.

---

### 1.3. 연구 목표

본 졸업 과제는 뇌 MRI 영상 데이터(mpMRI 데이터<sup>1</sup>)를 사용해 MGMT 프로모터 메틸화 상태를 분석하여, 교모세포종의 유전적 아형을 예측하는 학습 모델을 개발하는 것을 목표로 한다.

## 2. 연구 배경

### 2.1. 데이터 수집

mpMRI 데이터는 Kaggle competition 에서 제공하는 585 명의 환자 데이터를 사용한다. 각 데이터에는 FLAIR, T1 강조(조영제 투입 전), T1 강조(조영제 투입 후), T2 강조 순으로 촬영된 MRI 영상 데이터가 들어있다.

T1 강조 영상은 TR(Repetition Time) 과 TE(Echo Time) 를 짧게 하여 얻은 영상으로, 구조적 분석에 용이하다. 반대로, T2 강조 영상은 TR 과 TE 를 모두 길게 하여 얻은 영상으로, 병변 분석에 용이하다. FLAIR(FLuid Attenuated Inversion Recovery)는 T2 강조 영상의 일종으로, 뇌척수액(CSF)의 신호를 억제하는 기법이다. 이는 CSF 가 검은색으로 나타나 병변을 더 자세히 확인할 수 있다. 포함된 csv 파일에는 환자별 MGMT 프로모터 메틸화 여부가 기록되어있다.

### 2.2. 개발 환경

개발 언어로는 Python 을 선택했다. CPU 의 경우 Intel(R) Xeon(R) CPU E5-2620 v4 모델을 사용하였고, GPU 의 경우 NVIDIA GeForce GTX Titan Xp D5X 12GB 모델 7 개를 사용하였다. 또한, 해당 사양을 원활하게 사용하기 위해 CUDA 를 사용하였다. 마지막으로, 출력된 결과를 통해 모델의 전체적인 성능을 알아보기 위해 결과 값을 csv 파일로 추출하여 Google Drive 에 업로드한 뒤, Colab 을 통해 결과를 시각화하였다.

---

<sup>1</sup> Multiparametric MRI. MRI 스캔의 한 방법으로 확산 가중 이미징, 관류 가중 이미징, 자기 공명 분광법을 사용해 조직의 미세 환경을 더 정확하게 측정할 수 있다.

---

### 2.3. 데이터 전처리

데이터는 train data 와 test data 로 나뉘어 있고 각 데이터는 환자 아이디 별로 분류되어 있다. 모델의 정확도를 예측하기 위해 train data 를 다시 8:2 로 나눠 train 용과 test 용으로 나누었다. 한 환자의 아이디에 4 가지 촬영 기법의 데이터 폴더가 있으며 폴더 안에 dicom 파일이 순서대로 정리되어 있다.

2D 모델과 3D 모델이 요구하는 데이터의 조건에 차이가 있어 용도에 맞게 두 그룹의 전처리를 진행하였다. 우선 min-max normalization 을 진행한 후 각자의 용도에 맞게 추가적인 전처리를 진행하였다.

2D 모델은 pre-trained model 을 사용하기 위해 해당 이미지를 순서대로 pydicom 으로 읽어 224\*224 로 크기를 조절하고 gray scale 인 이미지에 채널을 추가해 rgb scale 로 변경 후 png 파일로 저장하였다.

3D 모델의 학습을 위해서는 먼저 각 Dicom 파일에 저장된 ImageOrientation Patient 를 읽어 이미지가 촬영된 평면을 확인한 후 각 픽셀을 행렬화 하여 3 차원 행렬로 쌓았다. 그 후 ImagePositionPatient 로 이미지가 촬영된 방향을 확인해 방향이 반대로 저장되었을 경우 재정렬했다.

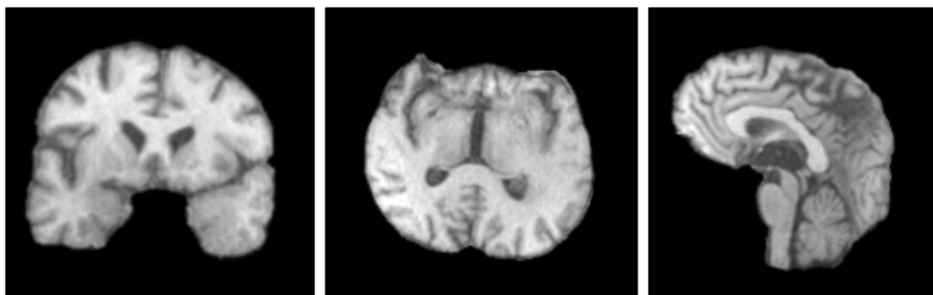


그림 1 각 평면에 따른 MRI 이미지

3D 이미지의 세 축 가운데 가장 긴 축을 확인한 후 그 축으로 입체를 64 등분해 단면을 64\*64 로 resizing 하고 다시 쌓았다.

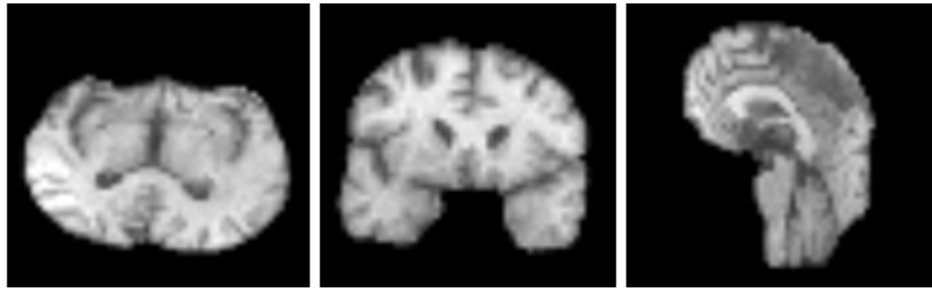


그림 2 전처리가 완료된 3D 이미지

편집을 위해 이미지를 3 차원 배열로 변경하여 dicom 파일로 바로 저장할 수 없었다. 이를 해결하기 위해 빈 dicom 파일을 만들어 행과 열, 프레임의 수를 64 로 지정하고 안의 픽셀 데이터에 편집이 완료된 3 차원 배열의 정보를 넣어 dicom 파일로 저장했다.

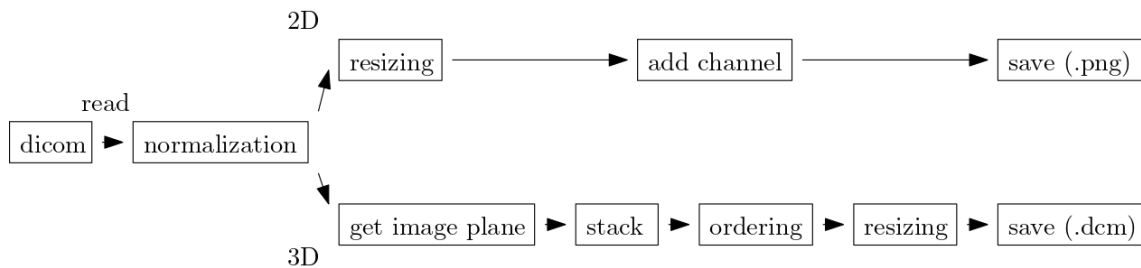


그림 3 전처리 과정

### 3. 연구 내용

#### 3.1. 2D 모델

본래 2D 모델을 사용할 때 imagenet 을 가중치로 사용하여 작업을 수행하려 하였다. 이를 위해서는 입력 값이  $224 \times 224 \times 3$  의 형태여야 하여, 원본 데이터를 해당 비율로 변환한 후 그림 파일 형태로 저장하였다. 변환된 데이터를 불러오는 데에는 성공하였으나, 이를 배열로 변환할 시 메모리 부족이 발생하여 최종적으로는 원본 데이터를 불러온 후 프로그램 상에서 이들을 전처리한 후 전체 배열에 저장하여 2D 모델의 입력 값으로 사용하기로 하였다. 기존 dicom 파일을 불러온 뒤, 해당 이미지들을

축소한 후 이를 Numpy 배열로 변환하였다. 이후 값의 범위를 조절하기 위해 값을 255로 나눠주었다. 최종적으로 이 값들을 입력 값으로 사용하였다.

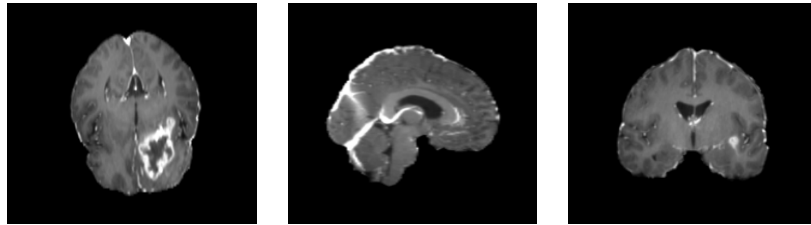


그림 4 3D Slicer 프로그램에 불러온 원본 데이터

2D 모델의 전체 계층 구조는 ResNet50 모델을 사용하였다. 해당 모델의 Optimizer 를 Adam, learning\_rate 를 0.001, loss 를 categorical\_crossentropy 로 설정한 후, epoch 를 100, batch\_size 를 1024 로 정한 뒤 학습을 진행하였다.

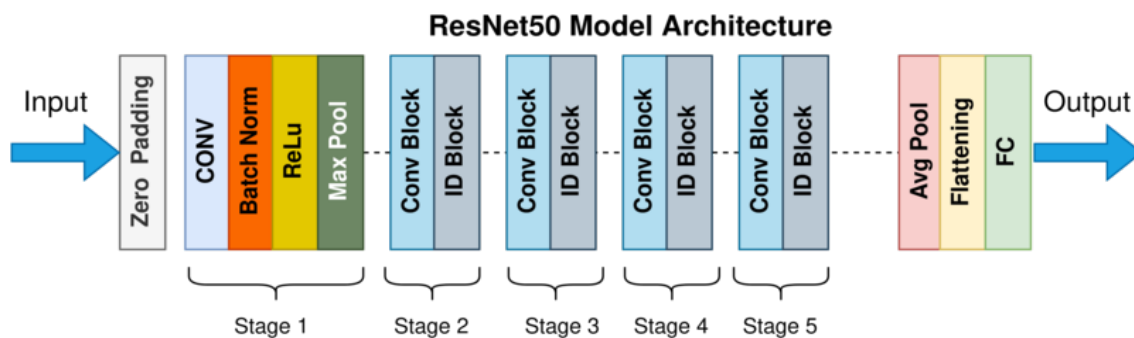


그림 5 2D 모델의 전체 계층 구조

학습 진행 시 epoch 당 평균 약 44 초를 소모했고, 학습 진행이 제일 효과적이었을 때의 상태는 accuracy 값이 0.8758, loss 값이 0.1957 이었다. 이후, 전체적인 학습 진행에 따른 2D 모델의 loss 값과 accuracy 값을 그래프로 나타내었다. 학습이 완료된 모델의 최초 예측 결과를 소수점으로 나타낸 뒤 이를 반올림하여 csv 파일로 추출하였다.

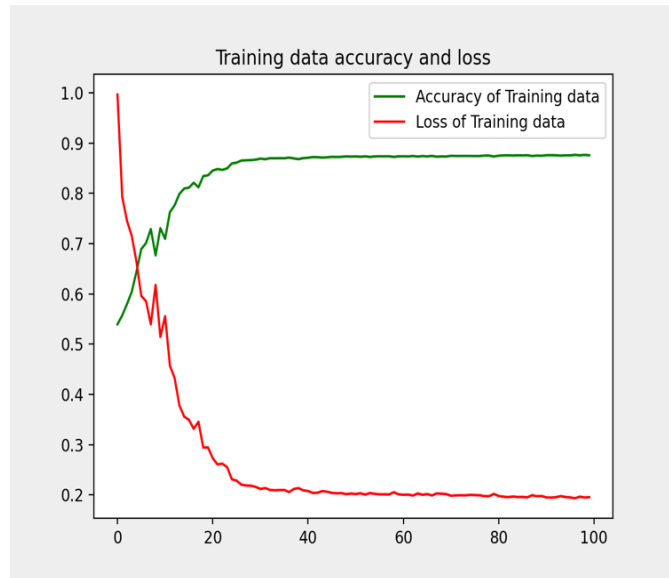


그림 6 2D 모델의 학습 그래프

### 3.2. 3D 모델

3D 모델은 2D 모델과 달리 전처리한 데이터를 사용하였다. 2.3 에서 전처리한 64\*64\*64 T1wCE dcm 데이터를 불러왔다. 불러온 데이터를 64\*64\*1 의 형태로 분리하여 배열의 형태로 변환한 뒤, 전체 배열에 저장하였다. 그 후 이를 3D 모델에 입력 값으로 사용하였다.

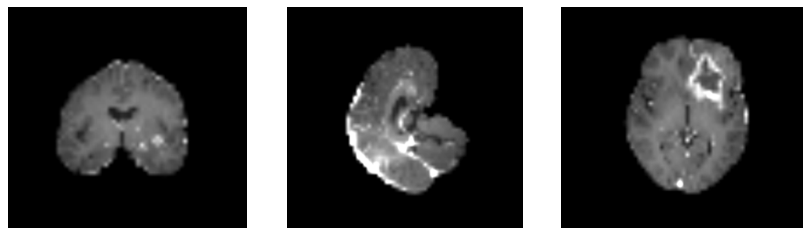


그림 7 64\*64\*64 데이터의 단면



3D 모델의 전체 계층 구조는 2D 모델의 전체 계층 구조와 동일하게 ResNet50 모델을 사용하였다.

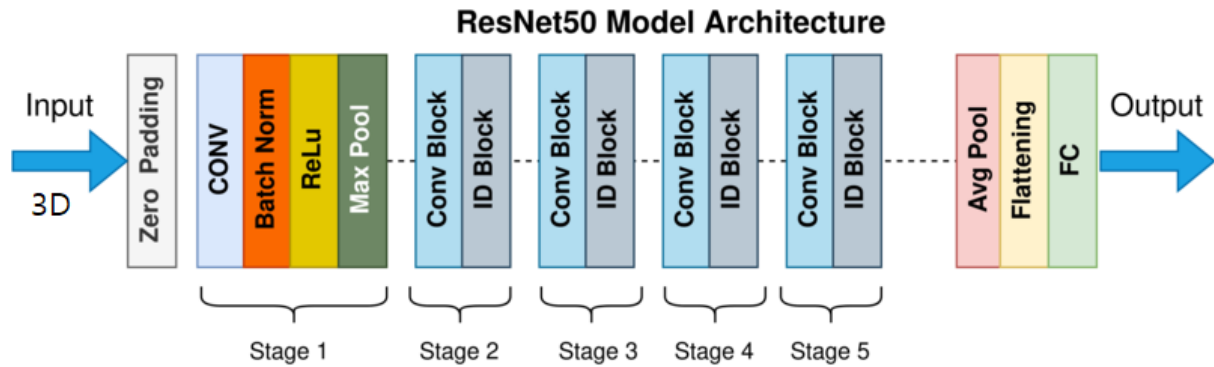


그림 8 3D 모델의 전체 계층 구조

ResNet은 기본적으로 Residual Block을 가지고, ResNet50에서는 1\*1, 3\*3, 1\*1 3개의 convolution layer를 사용한다. ResNet50의 전체적인 구조는 그림 10의 가운데 부분에 해당한다.

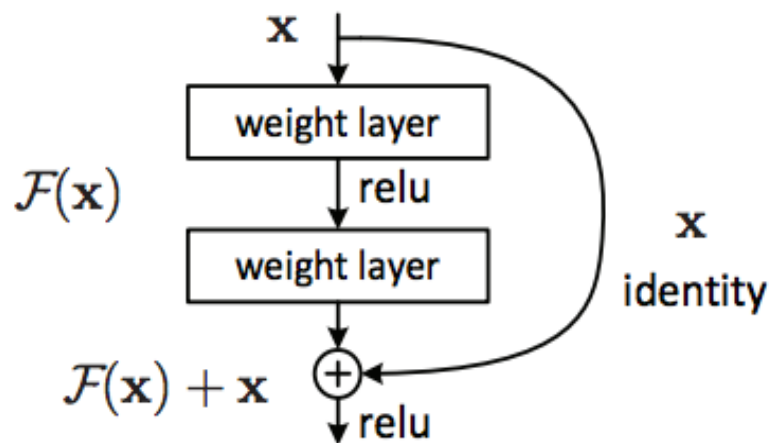


그림 9 ResNet의 Residual Block

해당 모델의 Optimizer, learning\_rate, loss를 각각 Adam, 0.001, categorical\_crossentropy로 설정한 후, epoch를 100, batch\_size를 1024로 정한 뒤 학습을 진행하였다.

layer name	34-layer	50-layer	101-layer
conv1	$7 \times 7, 64, \text{stride } 2$		
conv2_x	$3 \times 3 \text{ max pool, stride } 2$		
	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4_x	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$
conv5_x	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	average pool, 2048-d fc		

그림 10 ResNet50의 전체 구조

학습 진행 시 epoch 당 평균 약 26 초를 소모했고, 학습 진행이 제일 좋았을 때의 상태는 accuracy 값이 0.8776, loss 값이 0.1906 이었다. 전체적인 학습 진행에 따른 loss 값과 accuracy 값을 그림 11 에 그래프로 나타내었다.



그림 11 3D 모델의 학습 그래프

학습이 완료된 모델의 최초 예측 결과를 소수점으로 나타낸 뒤 이를 반올림하여 csv 파일로 추출하였다.

### 3.3. 교차 검증 (Cross Validation)

데이터의 편종을 막기 위해 별도의 세트로 구성된 학습 데이터 세트와 검증 데이터 세트에서 학습과 평가를 수행하였다. KFold 방법 가운데 StratifiedKFold 방법을 이용해 진행하였다. 전체 데이터 가운데 학습 데이터 세트와 검증 데이터 세트를 8:2 비율로 나누고, 학습 데이터 세트를 5 개의 fold 로 나누어 한 fold 를 골라 테스트 시키고, 나머지 부분을 학습에 사용하였다. 이 과정을 fold 만 바뀌가며 5 번 반복하여 데이터 세트에 있는 모든 Sample 이 테스트 될 수 있도록 하였다. 또한, Shuffle 을 True 로 설정하여 fold 를 나누기 전 데이터 세트를 무작위로 섞어서 Sampling 할 수 있게 진행하였으며, Random\_state 를 임의의 값인 42 로 설정하여, 작업을 여러 번 진행하여도 Sampling 결과를 유지할 수 있도록 하였다.

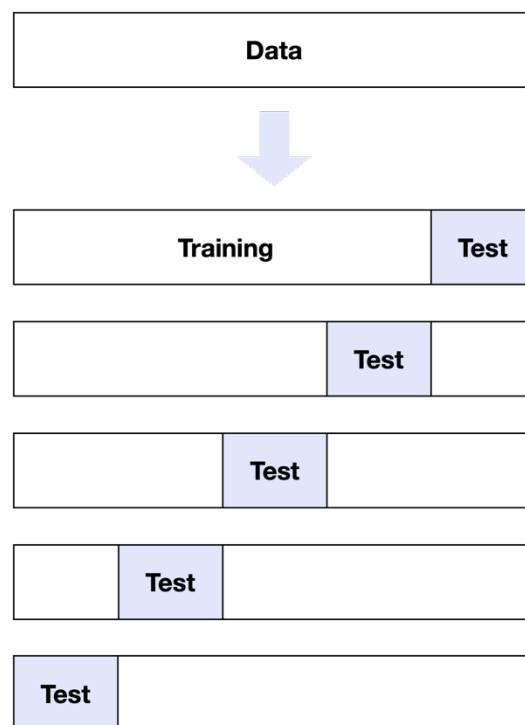


그림 12 StratifiedKFold Cross Validation

이로 하여금 데이터의 Overfitting 을 방지함과 동시에 모델의 올바른 일반화 성능을 추정할 수 있게 되었다.

### 3.4. 시각화 (Confusion Matrix)

StratifiedKFold 로 교차 검증을 진행함과 동시에 각각의 fold 가 학습될 때마다 fold 별 Confusion Matrix 를 png 파일로 출력하였다. Confusion Matrix 를 이용해 시각화한 이유는 학습을 통한 예측 성능을 측정하여 예측 Value 와 실제 Value 를 비교할 수 있고, 분류 모델의 성능 평가 지표를 보여줄 수 있기 때문이다.

아래의 그림에서 알 수 있듯 Actual Values 는 실제 값, Predictive Values 는 예측 값을 의미하며, T 는 True, F 는 False, P 는 Positive, N 은 Negative 를 의미한다. 다시 말해, TP 와 TN 은 실제 값을 맞게 예측한 부분이며, FP 와 FN 은 실제 값과 다르게 예측한 부분을 의미한다.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

그림 13 Confusion Matrix

Confusion Matrix 시각화를 위하여 sklearn 라이브러리를 사용하여 Confusion Matrix 를 위한 기능을 불러왔다. Normalize 값을 default 값인 None 으로 설정하여 정규화되지 않은 5 개의 Confusion Matrix 시각화 결과를 얻을 수 있었다.

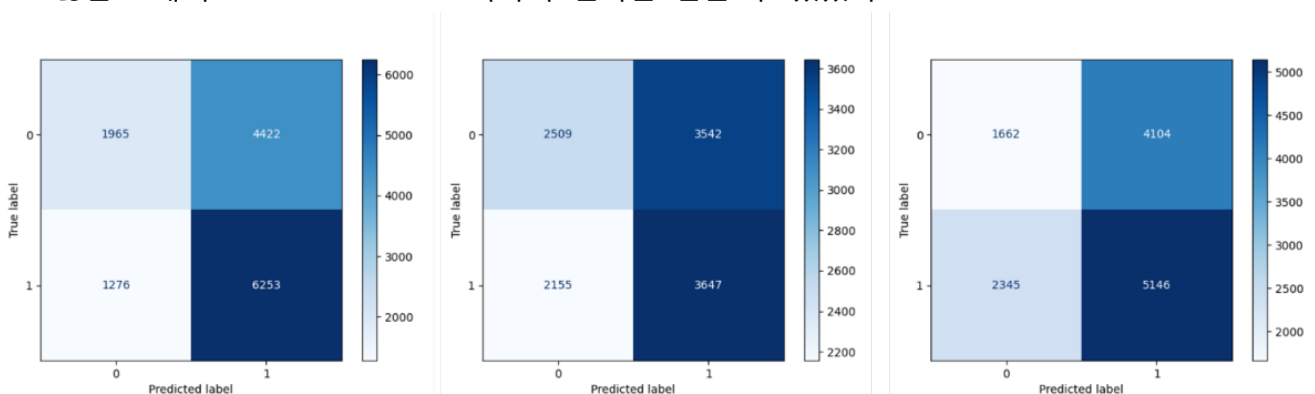


그림 14 좌측부터 순서대로 fold 0, 1, 2 로부터 얻어진 Confusion Matrix

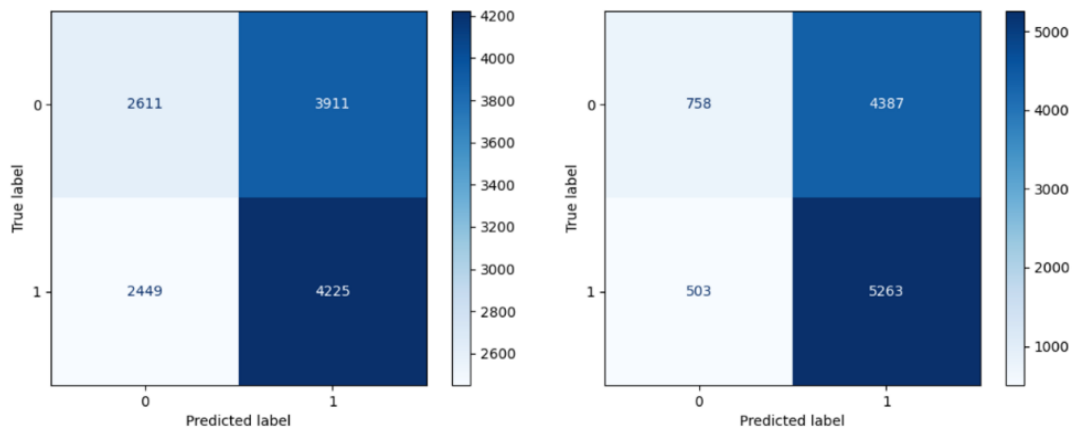


그림 15 좌측부터 순서대로 fold 3, 4 로부터 얻어진 Confusion Matrix

각각의 Confusion Matrix 에서의 TP, TN, FP, FN 값을 통해 Accuracy, F1 Score, Precision, Recall 를 얻을 수 있다. Accuracy 는 정확도로 전체 중 모델이 바르게 분류한 비율이고, Confusion Matrix 에서 대각선 부분을 의미한다. Precision 은 정밀도로 모델이 Positive 라 분류한 것 가운데 실제 값이 Positive 인 비율을 의미한다. 이는 Confusion Matrix 에서 열방향에 해당한다. Recall 은 재현도로 실제 값이 Positive 인 것 중 모델이 Positive 라 분류한 비율로, Confusion Matrix 의 행방향에 해당한다. F1 Score 는 Precision 과 Recall 의 조화평균이다.

<b>Accuracy</b>	=	$\frac{TP + TN}{TP + TN + FP + FN}$
<b>Precision</b>	=	$\frac{TP}{TP + FP}$
<b>Recall</b>	=	$\frac{TP}{TP + FN}$
<b>F1 score</b>	=	$\frac{2 \text{ Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

그림 16 Accuracy, F1 Score, Precision, Recall 수식

5 개의 Confusion Matrix 시각화 결과로부터 위 네 가지의 척도의 값을 구하였다. 이를 기반으로 numpy 라이브러리에서의 mean(), std() 함수를 사용하여 각 척도의 Average 와 Standard 값을 도출해내면 다음과 같다.

표 1 Confusion\_Matrix\_fold 별 Accuracy, Precision, Recall, F1 Score 값

Confusion_Matrix_fold	Accuracy	Precision	Recall	F1 Score
0	0.590	0.307	0.606	0.407
1	0.519	0.414	0.537	0.467
2	0.513	0.288	0.414	0.339
3	0.518	0.400	0.516	0.450
4	0.551	0.147	0.601	0.236

표 2 Average\_Accuracy, Average\_Precision, Average\_Recall, Average\_F1 Score 값

Avg_Accuracy	Avg_Precision	Avg_Recall	Avg_F1 Score
0.538	0.311	0.534	0.379

표 3 Std\_Accuracy, Std\_Precision, Std\_Recall, Std\_F1 Score 값

std_Accuracy	std_Precision	std_Recall	std_F1 Score
0.029	0.095	0.069	0.084

## 4. 연구 결과 분석 및 평가

2D 모델과 3D 모델의 MGMT\_Value 최종 예측 값을 원본 데이터의 MGMT\_Value 와 비교하여 시각화한 결과를 그림 17 과 그림 18 에 나타내었다. 최종 예측 값과 원본 데이터 값을 각각 csv 파일로 추출한 뒤, 이를 Google Colab 을 통해 비교하였다.

2D 모델과 3D 모델 모두 MGMT\_Value 최초 예측 값의 대다수가 0.5 를 넘었고, 이를 반올림하여 최종적으로 대부분의 MGMT\_Value 를 1 이라 예측하였다. 다만 정확도가 그리 높지는 않았는데, 해당 이유를 두 가지로 추측하였다.

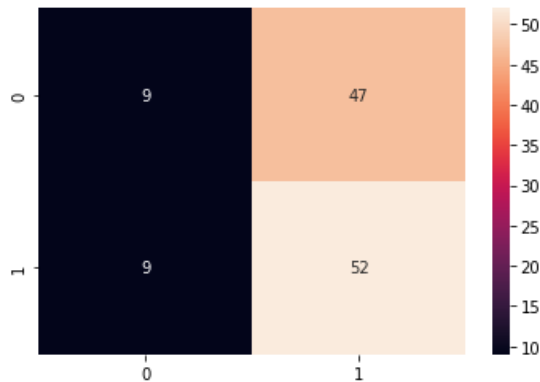


그림 17 2D 모델 Confusion Matrix

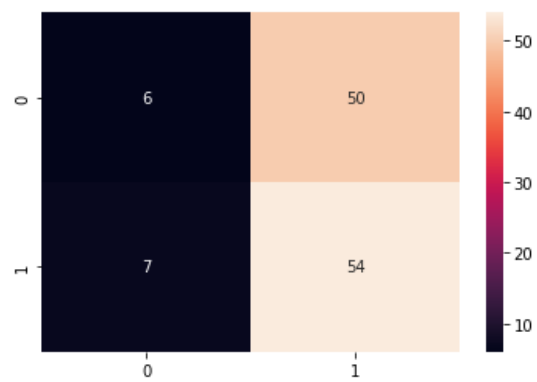


그림 18 3D 모델 Confusion Matrix

표 4 2D 모델 및 3D 모델 평가 지표

Model	Accuracy	Precision	Recall	F1 Score	AUC
2D Model	0.521	0.525	0.852	0.650	0.506
3D Model	0.512	0.519	0.885	0.654	0.496

첫번째, 원본 데이터와 MGMT\_Value 사이의 관계가 모호했다는 점이다. 그림 19 는 MGMT\_Value 가 1 인 데이터, 그림 20 은 MGMT\_Value 가 0 인 데이터인데, 두 데이터 사이에서 뚜렷한 차이를 발견하지 못했다.

두번째, 원본 데이터를 과도하게 압축했다는 점이다. 원본 데이터의 경우 512\*512 크기로 저장되어 있었으나 이를 64\*64 크기로 변환하여 사용하게 되면서 과도한 압축이 발생했다.

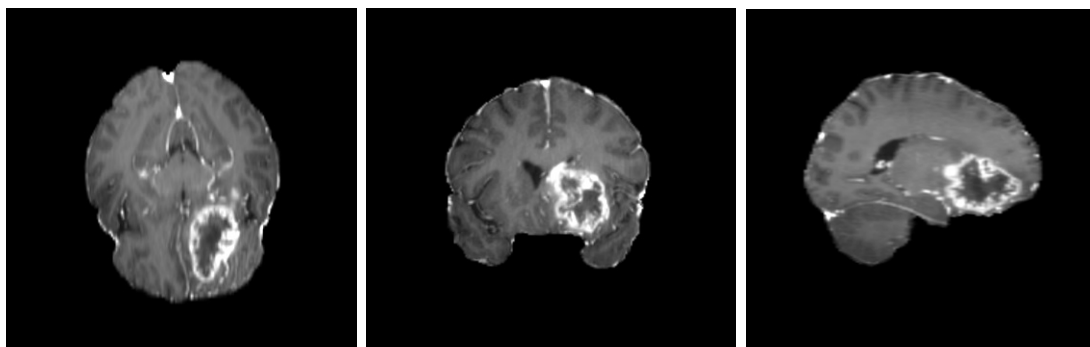


그림 19 MGMT\_Value 가 1 인 데이터

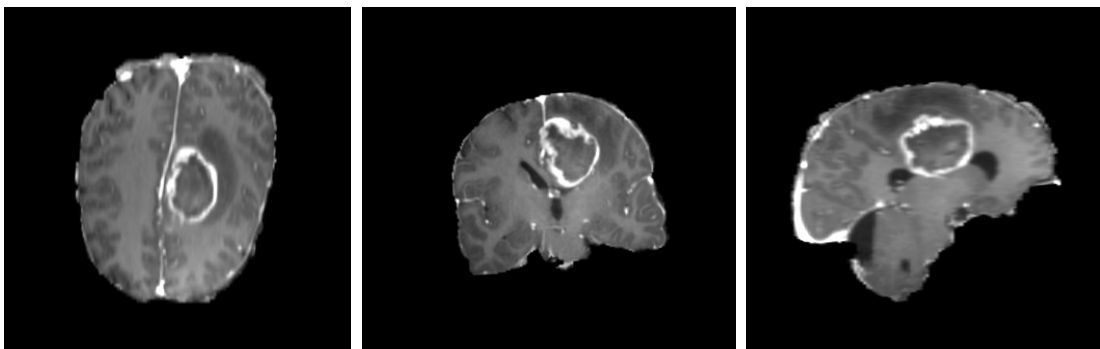


그림 20 MGMT\_Value 가 0 인 데이터

## 5. 결론 및 향후 연구 방향

우리는 뇌 MRI 영상 데이터를 사용해 MGMT 프로모터 메틸화 상태를 분석하여, 교모세포종의 유전적 아형을 예측하는 학습 모델을 개발하고자 하였다. 학습을 위한 데이터는 Radiological Society of North America 으로부터 제공받아 전처리 후 학습에 사용하였다. 이로써 데이터 전처리, 모델 설계 및 학습, 검증, 시각화까지 학습 모델 개발 과정 전체를 경험해보았다.

데이터 전처리의 경우 64\*64\*64 크기로 변경하며 데이터의 손실이 발생하였는데, 크기의 감소 폭을 줄여 데이터의 손실을 최소화한다면 더욱 더 향상된 결과치를 확인할 수 있었을 것이라 생각한다. 뿐만 아니라 전처리 과정에서 3D 이미지의 비율 일그러짐이



발생하였는데, 추후 일그러짐이 학습에 미치는 영향을 분석하고, 전처리를 다시 시도하여 추가 학습을 진행할 계획이다.

## 6. 구성원별 역할 및 개발 일정

표 5. 구성원별 역할

이름	역할
이섬재	<ul style="list-style-type: none"> <li>- 학습 모델 개발 및 최적화</li> <li>- 데이터 전처리 모듈 수정</li> <li>- 예측 결과 도출 및 시각화</li> </ul>
최연희	<ul style="list-style-type: none"> <li>- 학습 모델 테스트</li> <li>- 학습 모델 최적화 및 오류 수정</li> <li>- 데이터 전처리 및 시각화</li> </ul>
한혜린	<ul style="list-style-type: none"> <li>- 학습 모델 테스트</li> <li>- 학습 모델 최적화 및 오류 수정</li> <li>- 데이터 전처리</li> </ul>
공통	<ul style="list-style-type: none"> <li>- 보고서 작성 및 검토</li> <li>- 발표 및 시연 준비</li> </ul>

표 6 개발 일정

5월			6월				7월				8월					9월			
16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29	5	12	19	26
관련 지식 습득																			
			학습 모델 개발																
			알고리즘 작성 및 테스트																
									중간보고										
											데이터 전처리 및 시각화								
											학습 모델 최적화 및 오류 수정								
																최종 테스트			
																		최종보고 및 발표준비	

---

## 7. 참고 문헌

- [1] Baid, U., Ghodasara, S., Mohan, S., Bilello, M., Calabrese, E., Colak, E., ... & Bakas, S. "The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification". arXiv preprint arXiv:2107.02314, 2021.
- [2] Chaitra Dayananda, Bumshik Lee, "A comparative analysis on segmentation of brain MRI using CNN models" The Journal of Korean Institute of Next Generation Computing, vol.16, no.3, pp. 63-76, 2020. (in Korean)
- [3] Bijen Khagi. "Brain MRI segmentation and Alzheimer's disease classification using CNN" , Doctoral dissertation, chosun university general graduate school, 2019. (in Korean)
- [4] Sawlani, V., Patel, M. D., Davies, N., Flinham, R., Wesolowski, R., Ughratdar, I. & Poptani, H, "Multiparametric MRI: practical approach and pictorial review of a useful tool in the evaluation of brain tumours and tumour-like lesions", Insights into imaging, 11(1), 1-19, 2020.
- [5] ResNet, wikimedia commons, Available:  
<https://commons.wikimedia.org/wiki/File:ResNet50.png>
- [6] ResNet50, Aditi Rastogi, <https://blog.devgenius.io/resnet50-6b42934db431>