

딥 러닝 기반 프로젝트

# 뇌종양 MRI 분류 및 예측 모델 개발 및 시각화

2025. 01. 14

휴먼교육센터

이 찬

# 목차

## LIST

---

### 1. 프로젝트 소개

- a 추진 배경 및 목적
- b 과제 범위
- c 구축 대상 선정 기준
- d AI 예측 분석 모델 적용 대상

### 2. 주요 결과물

- a 데이터 수집 및 전처리
- b 모델 설계 및 학습
- c 예측 결과 및 평가
- d 프로토타이핑

# 01 프로젝트 개요 및 현황

사업과제: 딥러닝 기반 뇌종양 MRI 분류 시스템 개발 및 시각화

## 추진 배경 및 목적

01

의료 데이터의 디지털화와 함께 AI 기반 진단 보조 시스템에 대한 수요 증가, 특히 뇌종양 진단 분야에 정확하고 신속한 분석 도구의 필요성 대두

02

뇌종양의 조기 진단과 정확한 유형 분류는 환자의 치료 계획 수립과 예후에 핵심적인 역할을 함

03

MRI이미지 판독은 고도의 전문성과 시간이 요구되는 작업으로, 의료진의 업무 부담을 경감하고 진단의 정확성을 보조할 수 있는 자동화된 시스템이 필요

04

의료 AI 시스템의 실용화를 위해서는 높은 정확도뿐만 아니라 결과에 대한 설명 가능성과 시각화가 중요하므로, 이를 고려한 통합적인 시스템 개발이 필요

# 01 프로젝트 개요 및 현황

과제 범위

과제구분		내용
AI	AI기반 뇌종양MRI 분류 시스템 구현	원시 데이터 수집 및 데이터셋 구축
		데이터 전처리, 표준화, 상관관계 분석
		예측모델 선정(CNN) 및 학습
		정확도, 분류 보고서, 혼동 행렬을 통한 평가 지표를 활용한 모델 성능 평가
		웹 프로토타입(Streamlit) 구축
		예측모델 시각화 및 웹기반 시스템 구축
		테스트

# 01 프로젝트 개요 및 현황

## 구축 대상 선정 기준

### 데이터 접근성 및 활용성

- 공개된 의료 이미지 데이터셋의 활용 가능성 검토
- 이미지 해상도, 촬영 각도 등 데이터 품질의 일관성
- 각 종양 유형별 충분한 데이터 샘플 확보

### 예측모델 개발 효율성

- CNN 모델 적용에 적합한 이미지 특성 보유
- 학습 데이터의 크기와 모델 복잡도의 균형
- GPU 자원을 효율적으로 활용할 수 있는 데이터 구조

### 문제 해결 기여도 및 경제성

- 의료진의 진단 보조 도구로서의 활용 가능성
- 신속하고 정확한 초기 진단 지원 여부
- 의료 교육 및 연구 목적으로의 확장 가능성

# 01 프로젝트 개요 및 현황

AI 예측 분석모델 적용 대상

환경관리기능	수집데이터	예측모델인자	AI예측 분석 대상
뇌종양 분류	뇌종양 데이터셋의 MRI 이미지4가지 종양 유형별 분류 이미지각 종양 당 300~400개의 이미지	이미지 특성: 픽셀값, 크기 (150x150), 채널(RGB)종양 유형: 신경교종, 수막종, 뇌하수체 종양, 종양x데이터 증강: 회전, 반전, 확대/축소	4가지 종양 유형 분류클래스별 예측 확률분류 정확도 평가

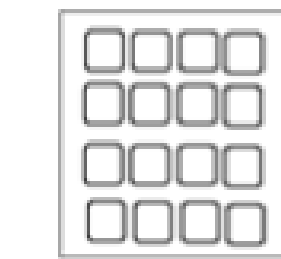
# 01 프로젝트 개요 및 현황

AI 예측 분석모델 구축 프로세스

## 데이터 수집



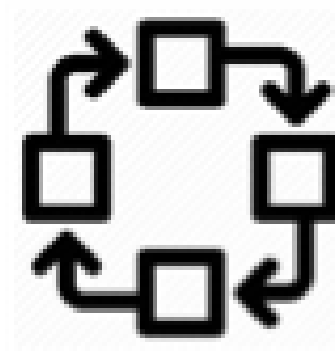
- MRI 이미지 데이터셋 구축
- 4개 클래스의 뇌종양 이미지 확보
- 훈련용/테스트용으로 분리된 데이터 구성



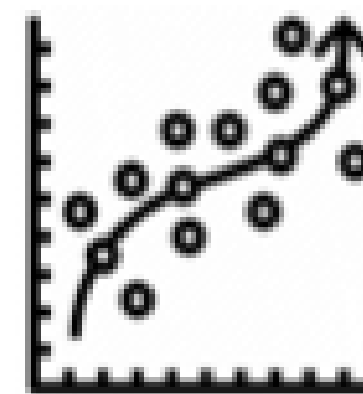
## 데이터 수집

- 이미지 크기 표준화 (150x150픽셀)
- 픽셀 정규화(0~1 스케일링)
- 데이터 증각 기법 적용(회전, 반전, 확대/축소)

## 자동화 예측 분석



- CNN 아키텍처 채택
- 4개의 Conv2D-MaxPooling 블록 구성
- Dropout층을 통한 과적합 방지
- Dense 층을 통한 최종 분류 구현



- 배치 크기 32, 50 에포크로 학습 진행
- Early Stopping을 통한 최적의 학습 시점 선정
- 모델 가중치 저장 및 관리

## 사용자 분석 및 시각화



- 정확도 측정
- 혼동 행렬을 통한 성능 분석
- 오분류 사례 및 개선점 도출
- 이후, Streamlit을 통한 웹 인터페이스 구축 및 시각화

# 02 주요 결과물

## 데이터 수집 - Kaggle 데이터셋 자동 다운로드 구현

```
from kaggle.api.kaggle_api_extended import KaggleApi
import os
import zipfile

def download_and_extract_data():
    dataset_name = "masoudnickparvar/brain-tumor-mri-dataset"

    download_dir = os.path.join(os.getcwd(), 'tumor_dataset')
    os.makedirs(download_dir, exist_ok=True)

    zip_file_path = os.path.join(download_dir, 'brain-tumor-mri-dataset.zip')

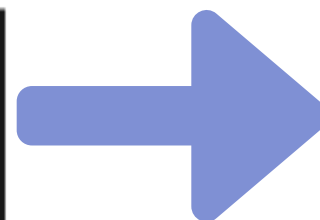
    if not os.path.exists(zip_file_path):
        api = KaggleApi()
        api.authenticate()

        print("데이터셋 다운로드 중...")
        api.dataset_download_files(dataset_name, path=download_dir, unzip=False)

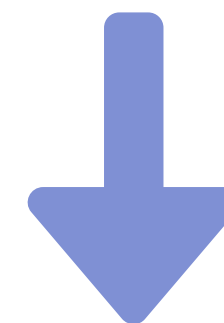
    extract_dir = os.path.join(download_dir, 'brain_tumor_dataset')
    if not os.path.exists(extract_dir):
        print("압축 해제 중...")
        with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
            zip_ref.extractall(extract_dir)
        print("압축 해제 완료")

    return extract_dir

data_dir = download_and_extract_data()
```



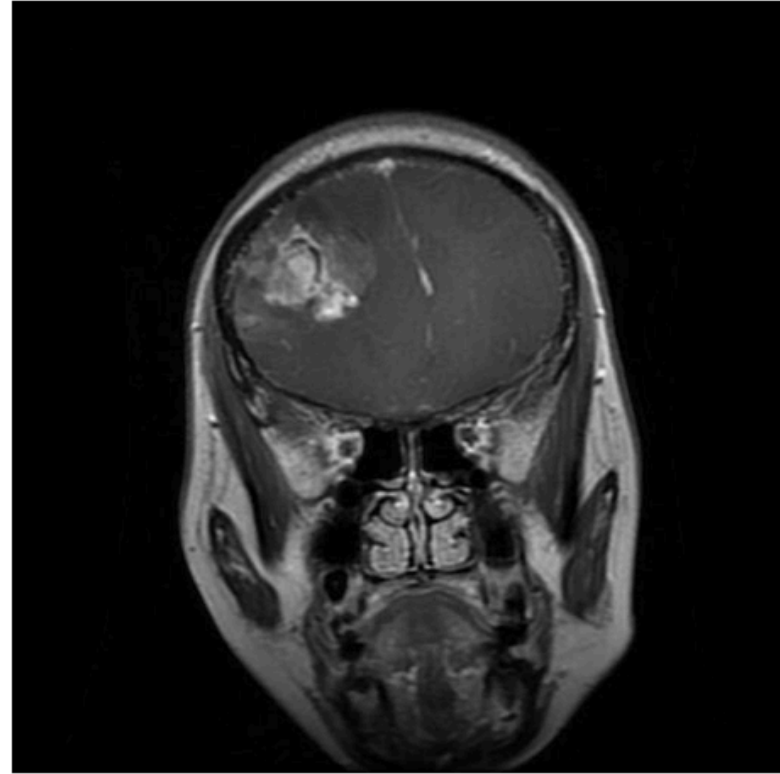
```
데이터셋 다운로드 중...
Dataset URL: https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset
압축 해제 중...
압축 해제 완료
```



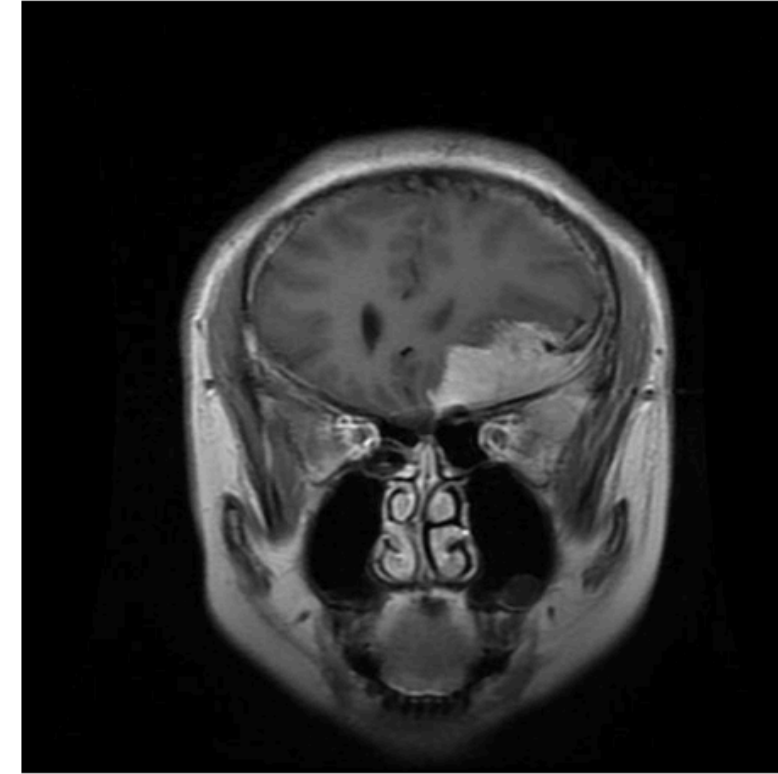
```
▼ AI_PROJECT
  > __pycache__
  > .venv
  > dataset ●
  > font
  > model
  > report
  ▼ tumor_dataset ●
    ▼ brain_tumor_dataset ●
      > Testing ●
      > Training
```



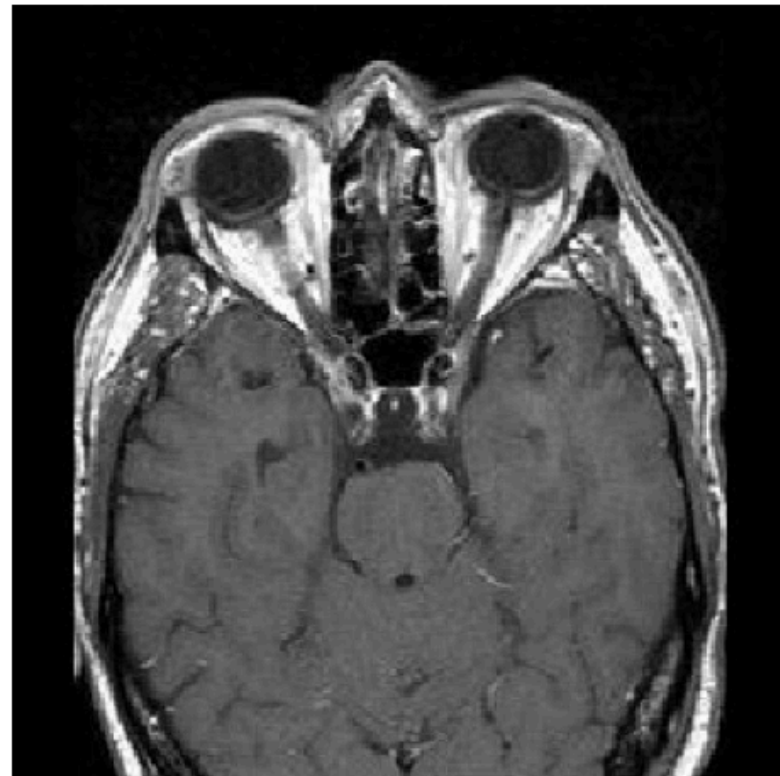
glioma 샘플



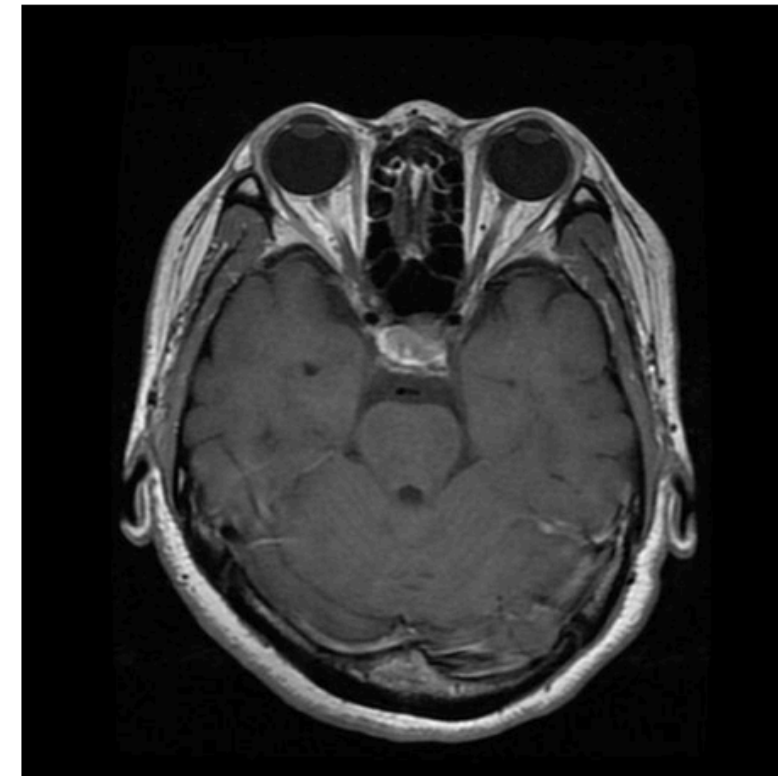
meningioma 샘플



notumor 샘플

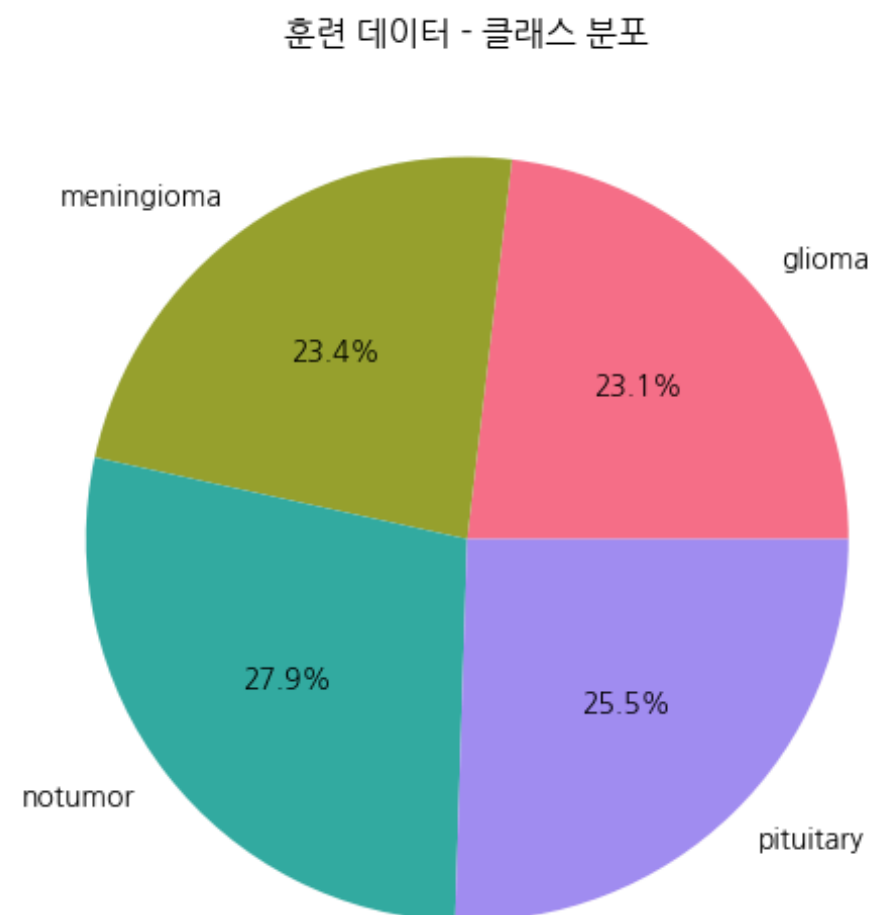
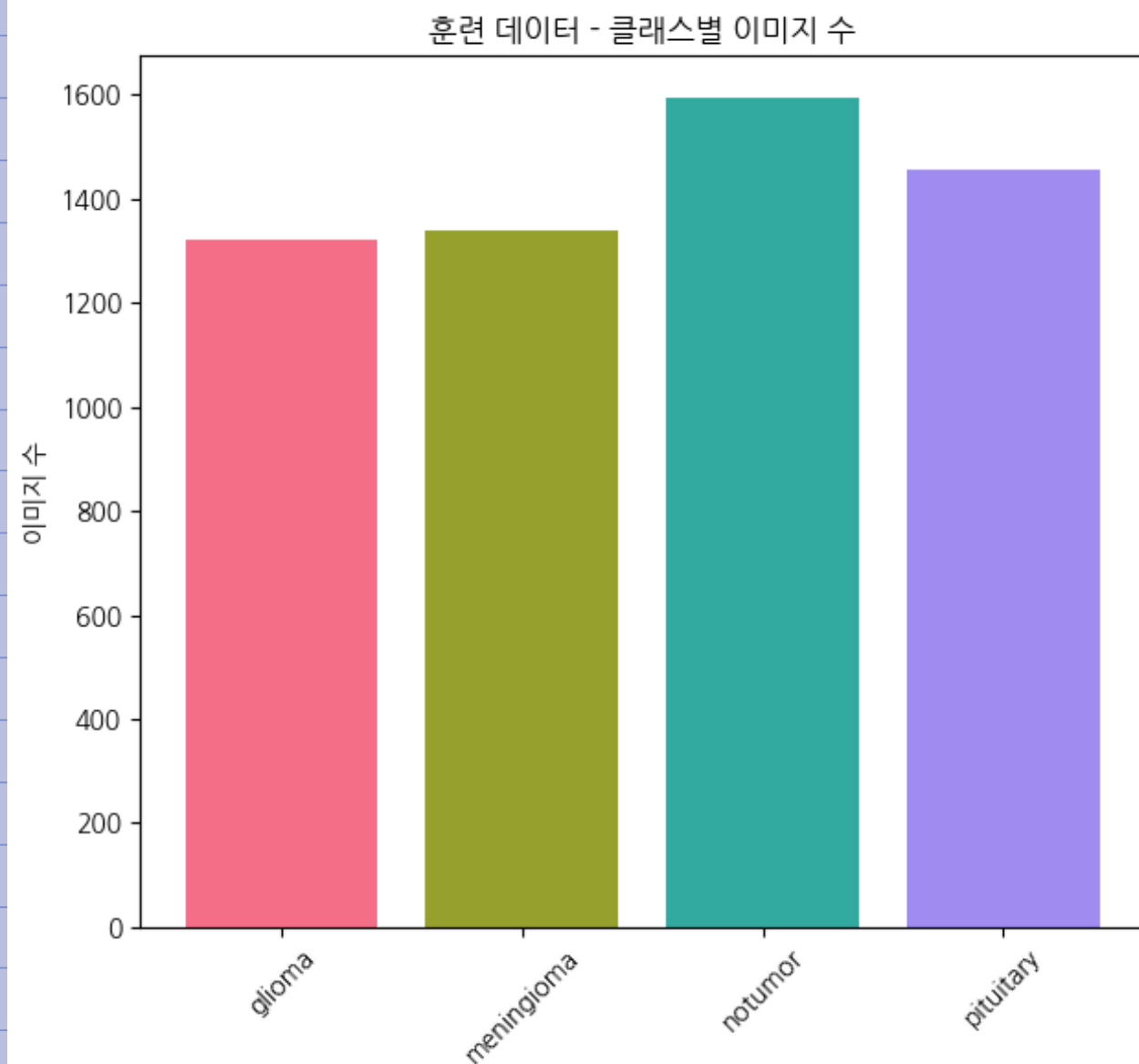


pituitary 샘플



# 02 주요 결과물

## 데이터셋의 시각화 및 분석



- 클래스 별 데이터 분포도 확인
- 최대 차이가 no tumor 와 glioma 사이 약 4.8%로 양호한 수준
- 균형 잡힌 데이터 분포 -> 특정 클래스에 편향되지 않을 것
- 각 클래스 당 1300~1600개의 이미지 -> 딥 러닝에 적절한 수준

# 02 주요 결과물

## 데이터 수집 - Kaggle 데이터셋 자동 다운로드 구현

```
# 클래스 및 이미지 개수 확인
classes = sorted([item.name for item in train_dir.glob('*/*') if item.is_dir()])
train_image_count = len(list(train_dir.glob('*/*.jpg')))
test_image_count = len(list(test_dir.glob('*/*.jpg')))

print(f"발견된 클래스: {classes}")
print(f"학습 이미지 개수: {train_image_count}")
print(f"테스트 이미지 개수: {test_image_count}")
```

✓ 0.1s

발견된 클래스: ['glioma', 'meningioma', 'notumor', 'pituitary']  
학습 이미지 개수: 5712  
테스트 이미지 개수: 1311

```
# 각 클래스별 이미지 개수 확인
for class_name in classes:
    train_images = len(list(train_dir.glob(f'{class_name}/*')))
    test_images = len(list(test_dir.glob(f'{class_name}/*')))
    print(f"{class_name}: 학습 {train_images}개, 테스트 {test_images}개 이미지")
```

✓ 0.0s

glioma: 학습 1321개, 테스트 300개 이미지  
meningioma: 학습 1339개, 테스트 306개 이미지  
notumor: 학습 1595개, 테스트 405개 이미지  
pituitary: 학습 1457개, 테스트 300개 이미지

```
# 이미지 크기 및 배치 크기 설정
img_height = 150
img_width = 150
batch_size = 32
```

✓ 0.0s

```
# 데이터셋 생성
```

```
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    str(train_dir),
    image_size=(img_height, img_width),
    batch_size=batch_size
)
```

```
test_ds = tf.keras.preprocessing.image_dataset_from_directory(
    str(test_dir),
    image_size=(img_height, img_width),
    batch_size=batch_size
)
```

✓ 0.4s

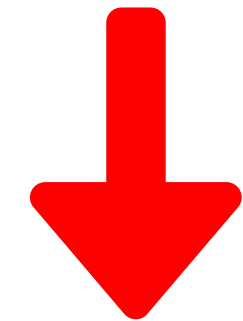
Found 5712 files belonging to 4 classes.  
Found 1311 files belonging to 4 classes.

데이터의 클래스 및 이미지 개수 확인 후 이미지 크기 조정 -> 데이터셋 생성

# 02 주요 결과물

데이터 전처리 - 증강 레이어 정의

```
# 데이터 증강 레이어 정의
data_augmentation = tf.keras.Sequential([
    layers.RandomFlip("horizontal"),
    layers.RandomRotation(0.2),
    layers.RandomZoom(0.2),
])
```



데이터의 수가 균형잡혔기  
때문에 클래스 가중치 부여  
는 건너뛴

RandomFlip: 이미지를 수평으로 무작위 뒤집기

RandomRotation: 이미지를  $\pm 20\%$  범위로 무작위 회전

RandomZoom: 이미지를  $\pm 20\%$  범위로 무작위 확대/축소

이러한 방식을 통해 훈련 데이터의 다양성 증가 및 모델의 일반화 성능 향상

# 02 주요 결과물

## 모델 정의 - CNN 모델

```
# CNN 모델 구성 - 뇌종양 분류에 맞게 더 깊은 모델 구성
model = tf.keras.Sequential([
    data_augmentation,
    layers.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(128, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(256, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Dropout(0.3),
    layers.Flatten(),
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(num_classes, activation='softmax')
])
```

점진적으로 증가하는 필터 수(32-→64-→128-→256)

여러 번의 MaxPooling으로 특징 맵 크기 감소

2번의 Dropout으로 과적합 방지

# 02 주요 결과물

## 모델 구조 확인

```
# 모델 구조 출력
model.summary()
✓ 0.0s
```

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
sequential (Sequential)	(None, 150, 150, 3)	0
rescaling (Rescaling)	(None, 150, 150, 3)	0
conv2d (Conv2D)	(None, 150, 150, 32)	896
max_pooling2d (MaxPooling2D)	(None, 75, 75, 32)	0
conv2d_1 (Conv2D)	(None, 75, 75, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 37, 37, 64)	0
conv2d_2 (Conv2D)	(None, 37, 37, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 18, 18, 128)	0

conv2d_3 (Conv2D)	(None, 18, 18, 256)	295168
max_pooling2d_3 (MaxPooling2D)	(None, 9, 9, 256)	0
dropout (Dropout)	(None, 9, 9, 256)	0
flatten (Flatten)	(None, 20736)	0
dense (Dense)	(None, 256)	5308672
dropout_1 (Dropout)	(None, 256)	0

# 02 주요 결과물

## 모델 학습

```
# Early Stopping 콜백 설정
callbacks = [
    tf.keras.callbacks.EarlyStopping(
        monitor='val_loss',
        patience=5,
        restore_best_weights=True
    )
]
```

✓ 0.0s

```
# 모델 학습
epochs = 20
history = model.fit(
    train_ds,
    validation_data=test_ds,
    epochs=epochs,
    callbacks=callbacks
)
```



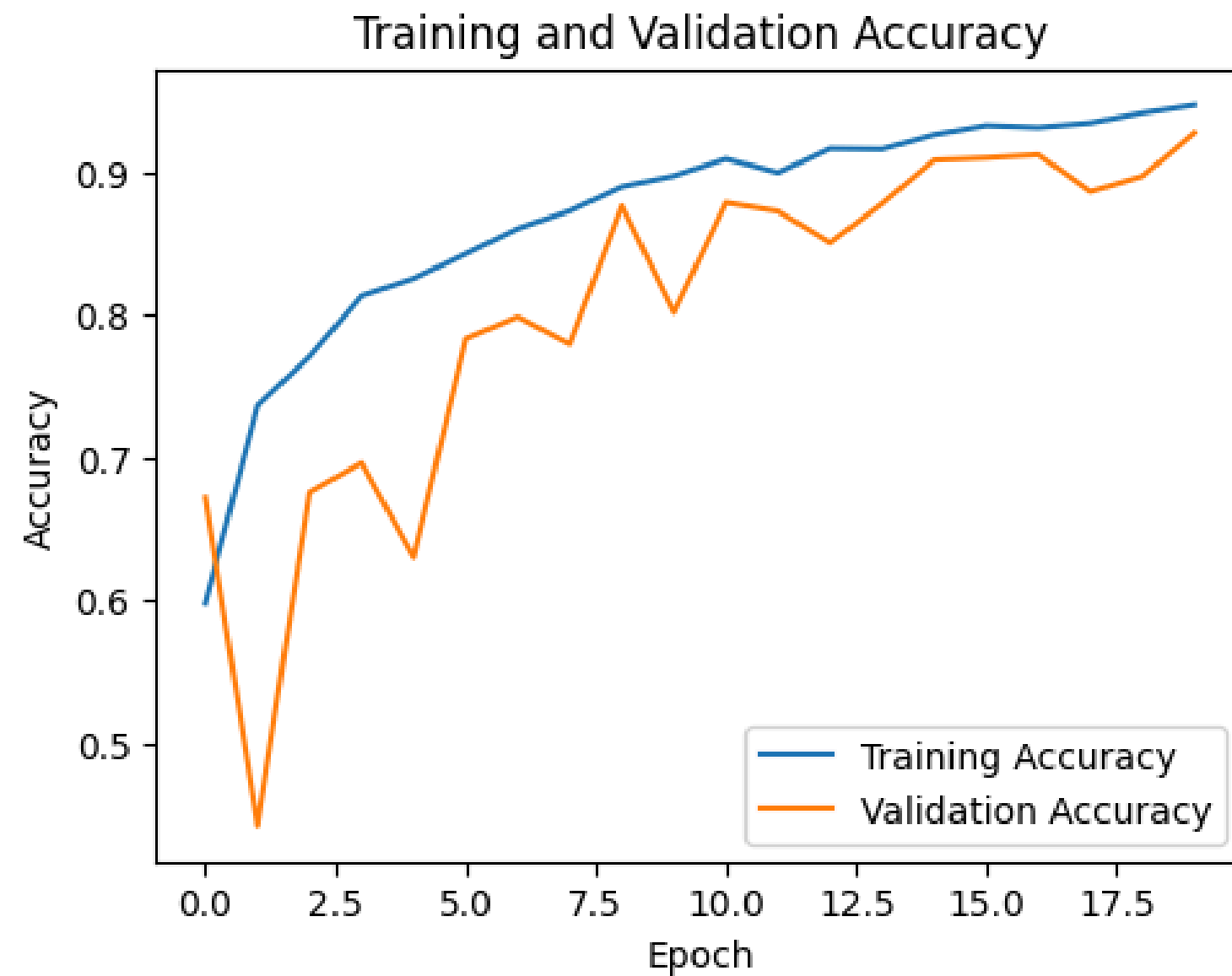
EarlyStopping을 사용했음에도 총 20번을 학습?

Epoch 20/20

179/179 [=====] - 187s 1s/step - loss: 0.1541 - accuracy: 0.9470 - val\_loss: 0.1989 - val\_accuracy: 0.9275

# 02 주요 결과물

모델 학습



Validation Loss, 즉 training 데이터의 20% 인 검증용 데이터의 정확도는 처음엔 큰 차이가 났으나 학습 할 수록 정확도가 큰 폭으로 상승



## 02 주요 결과물

분류 보고서 / 혼동 행렬을 통한 예측값 vs 실제값 확인

```
from sklearn.metrics import classification_report
import numpy as np

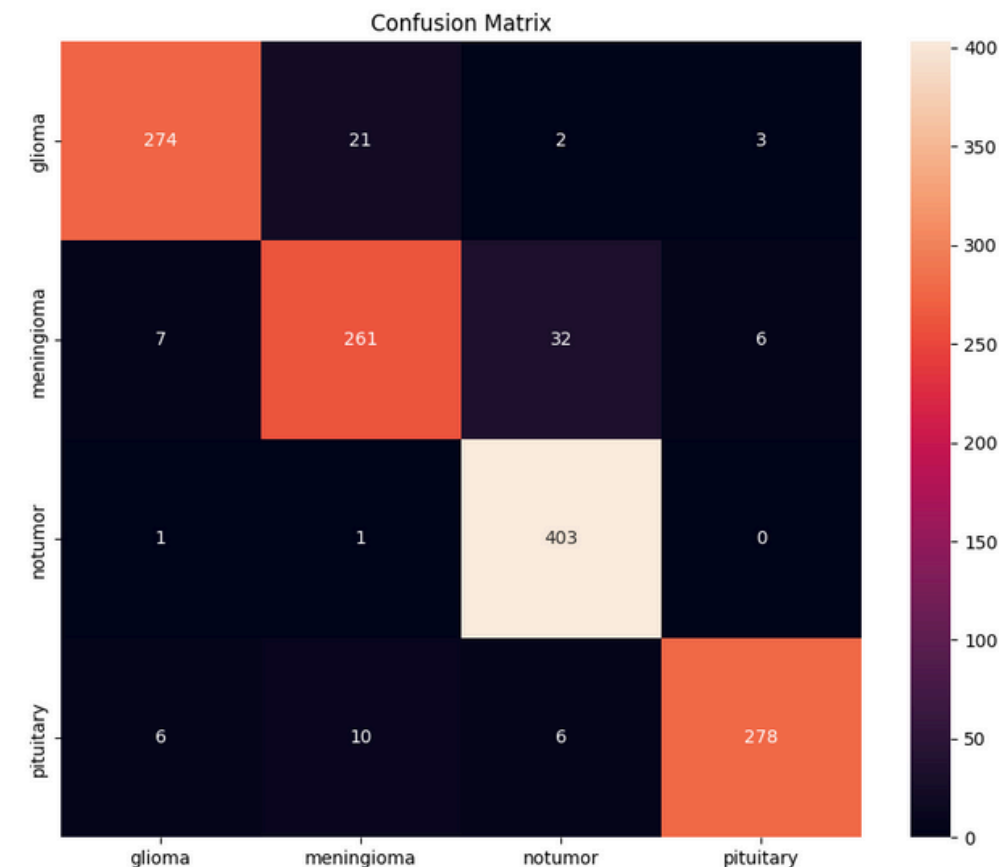
# 예측값 생성
y_pred = []
y_true = []

for images, labels in test_ds:
    pred = model.predict(images)
    pred_classes = np.argmax(pred, axis=1)
    y_pred.extend(pred_classes)
    y_true.extend(labels)

# 분류 보고서 생성
print(classification_report(y_true, y_pred,
                             target_names=['glioma', 'meningioma', 'notumor', 'pituitary']))
```

클래스별 성능 (Classification Report)

Glioma: 정밀도 95%, 재현율 91%, F1-score 93%  
Meningioma: 정밀도 89%, 재현율 85%, F1-score 87%  
No Tumor: 정밀도 91%, 재현율 100%, F1-score 95%  
Pituitary: 정밀도 97%, 재현율 93%, F1-score 95%



No Tumor 클래스가 가장 정확하게 분류됨 (403개 정확 분류)  
Meningioma 클래스가 상대적으로 다른 클래스와 혼동되는 경우가 많음  
주요 오분류 사례:

Meningioma를 No tumor로 오분류: 32건  
Glioma를 Meningioma로 오분류: 21건  
Pituitary를 Meningioma로 오분류: 10건

# 02 주요 결과물

이미지 업로드를 통한 예측 실행

## 분류 가능한 종양 유형

1. Glioma (신경교종)
2. Meningioma (수막종)
3. No Tumor (종양 없음)
4. Pituitary (뇌하수체 종양)

## 사용 방법

1. MRI 이미지 파일을 업로드합니다.
2. 자동으로 분석이 시작됩니다.
3. 분석 결과와 확률을 확인합니다.

## 뇌종양 MRI 분류 시스템

MRI 이미지를 업로드하면 종양의 유형을 분석해드립니다.

MRI 이미지를 업로드하세요



Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

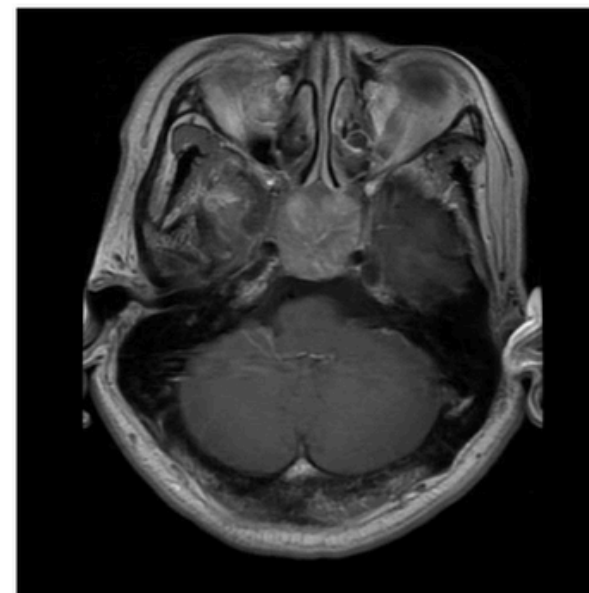
Browse files



Te-pi\_0024.jpg 27.7KB



## 업로드된 MRI 이미지



## 분석 결과



## 진단 결과

PITUITARY



## 신뢰도

47.21%



## 상세 분석

glioma: 17.61%

meningioma: 17.63%

notumor: 17.55%

# THANK YOU

---

감사합니다