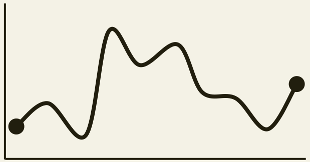
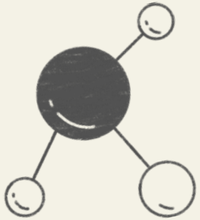


딥러닝

MRI 이미지 기반 암 환자 예측 모델 개발

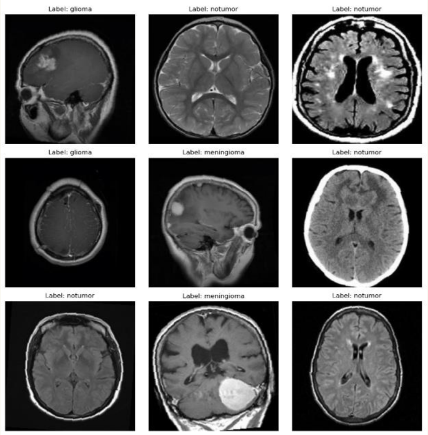
23.03 ~ 23.06



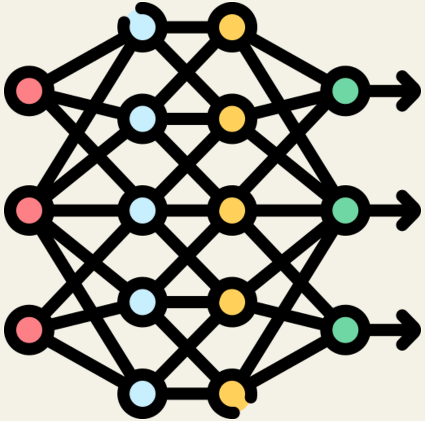
Introduction

다양한 뇌 종양을 진단할 수 있는 모델을 개발하고자 한다.
뇌 MRI 이미지를 분석하여 암 환자의 예후를 예측하고,
치료 전략을 수립하는 데 도움을 주고자 한다.

질병 예측 알고리즘 개발



환자 뇌 MRI

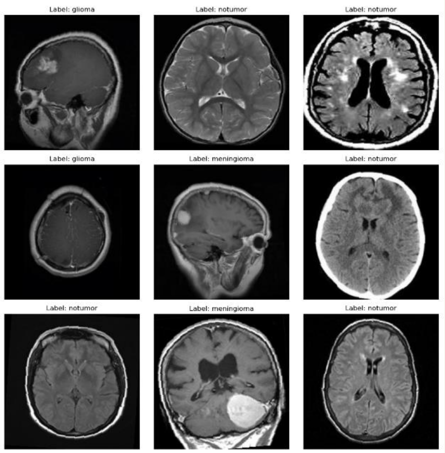


딥러닝 기반 모델

Glioma	글리오마
Meningioma	수막종
Pituitary	뇌하수체 질병
No tumor	비환자

질병 예측

Data set

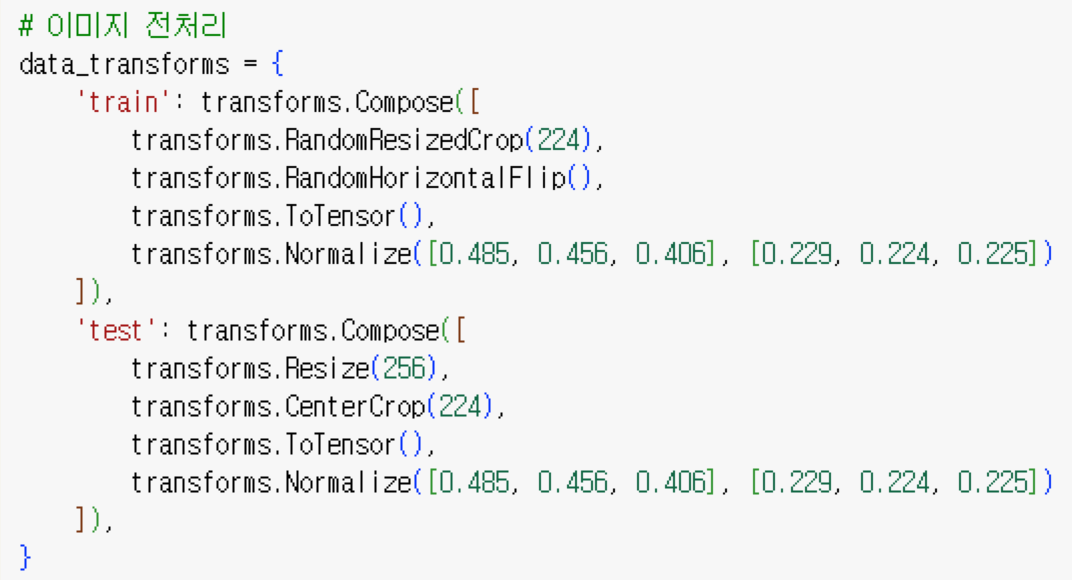


- Number of Samples: 5,712 images
- Number of Categories: 4
- 4 classes: glioma, meningioma, no tumor, pituitary.
- Data Source: Kaggle

Brain Tumor MRI Dataset

A dataset for classify brain tumors

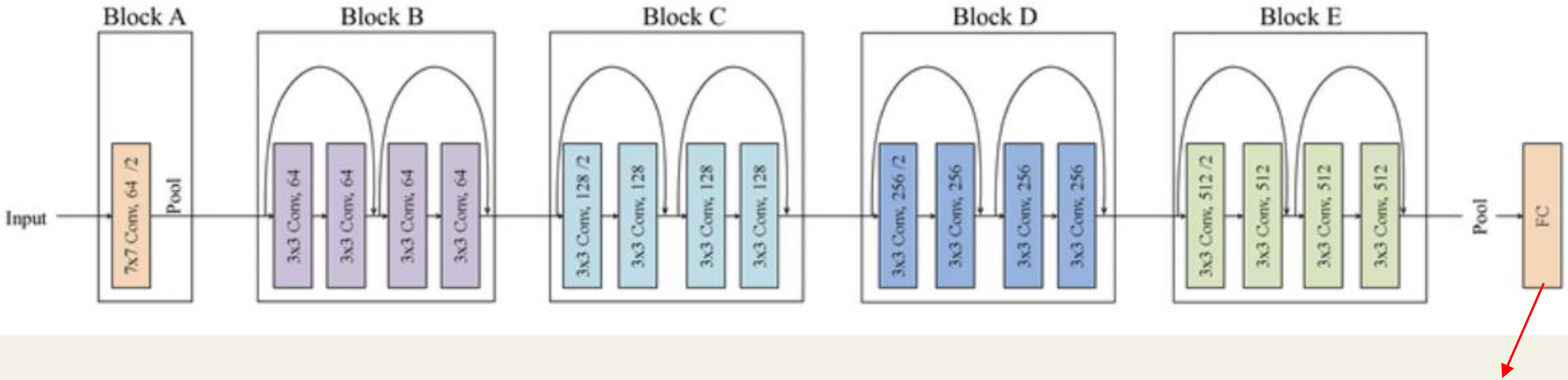
Pre-processing



```
# 이미지 전처리
data_transforms = {
    'train': transforms.Compose([
        transforms.RandomResizedCrop(224),
        transforms.RandomHorizontalFlip(),
        transforms.ToTensor(),
        transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
    ]),
    'test': transforms.Compose([
        transforms.Resize(256),
        transforms.CenterCrop(224),
        transforms.ToTensor(),
        transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
    ]),
}
```

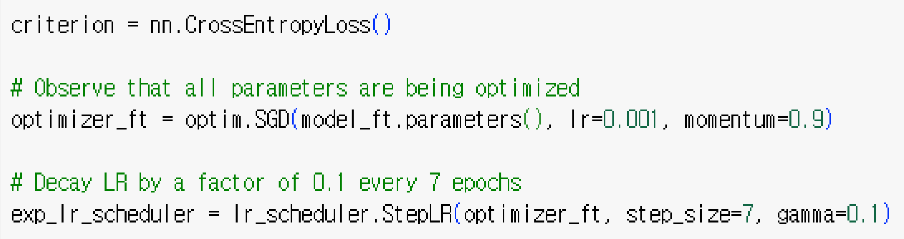
- 이미지 자르기 및 크기 조정
- 수평 반전
- Tensor로 변환
- 이미지 정규화

Model (ResNet-18)



출력 노드 4

Fine-tuning

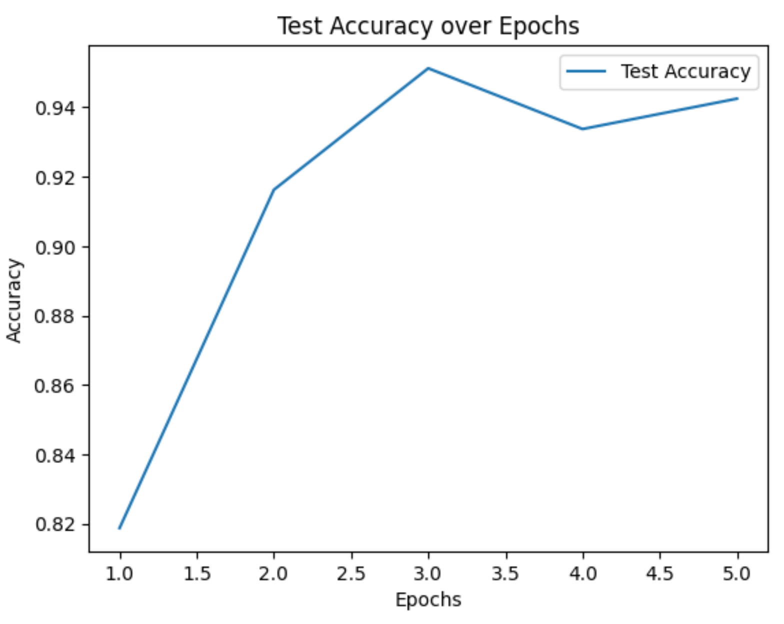


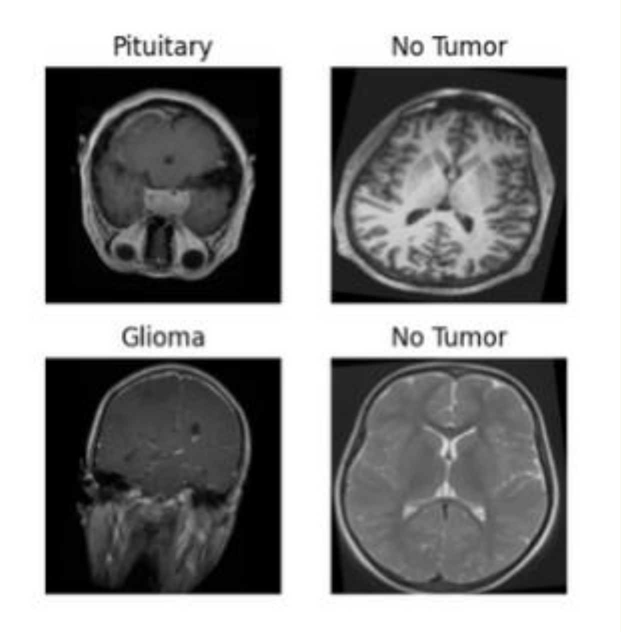
```
criterion = nn.CrossEntropyLoss()

# Observe that all parameters are being optimized
optimizer_ft = optim.SGD(model_ft.parameters(), lr=0.001, momentum=0.9)

# Decay LR by a factor of 0.1 every 7 epochs
exp_lr_scheduler = lr_scheduler.StepLR(optimizer_ft, step_size=7, gamma=0.1)
```

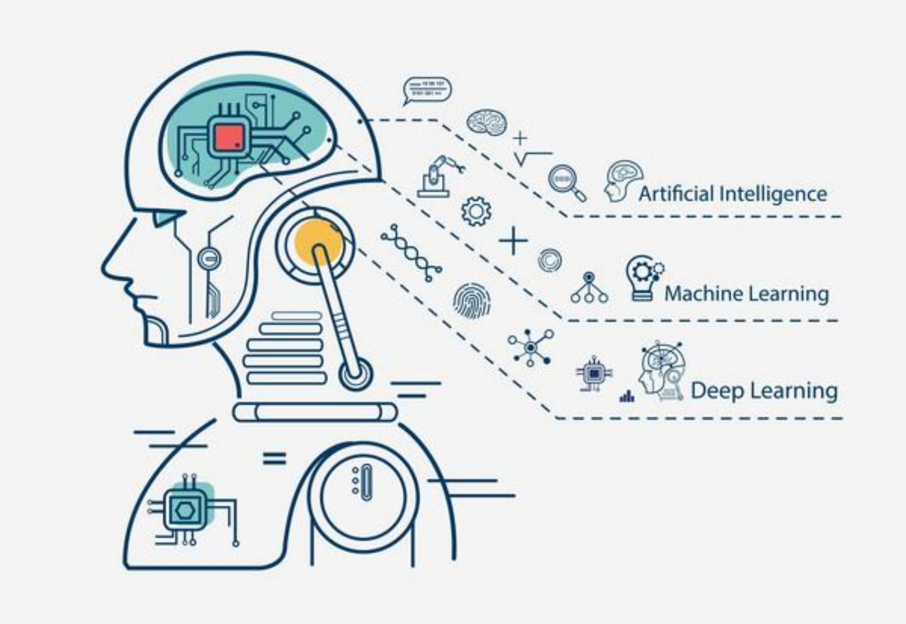
- 손실함수: cross entropy
- Optimizer: SGD

  (lr = 0.001, momentum=0.9)
- 학습률 조절: 7 epochs 마다 0.1 감소

Result

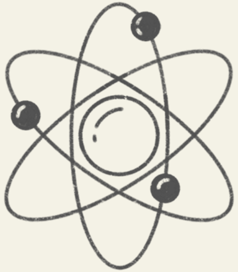

Test Accuracy over Epochs
Test Accuracy
Accuracy
0.82
0.84
0.86
0.88
0.90
0.92
0.94
Epochs
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


Pituitary
No Tumor
Glioma
No Tumor



Lessons

- 전이 학습
- 신경망 구축
- 의료 영상 처리



감사합니다

CREDITS: This presentation template was created by **Slidesgo,** and includes icons by **Flaticon,** and infographics & images by **Freepik**

```
Input
  |
conv1 (7x7, 64) + BN + ReLU
  |
maxpool (3x3, stride 2)
  |
Layer1
  |
  |-- BasicBlock
  |     |-- conv1 (3x3, 64) + BN + ReLU
  |     |-- conv2 (3x3, 64) + BN
  |     +-- Shortcut Connection
  |
  |-- BasicBlock
  |     |-- conv1 (3x3, 64) + BN + ReLU
  |     |-- conv2 (3x3, 64) + BN
  |     +-- No Shortcut Connection
  |
Layer2
  |
  |-- BasicBlock
  |     |-- conv1 (3x3, 128, stride 2) + BN + ReLU
  |     |-- conv2 (3x3, 128) + BN
  |     +-- Shortcut Connection (1x1, stride 2)
  |
  |-- BasicBlock
  |     |-- conv1 (3x3, 128) + BN + ReLU
  |     |-- conv2 (3x3, 128) + BN
  |     +-- No Shortcut Connection
```

```
Layer3
  |
  |-- BasicBlock
  |     |-- conv1 (3x3, 256, stride 2) + BN + ReLU
  |     |-- conv2 (3x3, 256) + BN
  |     +-- Shortcut Connection (1x1, stride 2)
  |
  |-- BasicBlock
  |     |-- conv1 (3x3, 256) + BN + ReLU
  |     |-- conv2 (3x3, 256) + BN
  |     +-- No Shortcut Connection
  |
Layer4
  |
  |-- BasicBlock
  |     |-- conv1 (3x3, 512, stride 2) + BN + ReLU
  |     |-- conv2 (3x3, 512) + BN
  |     +-- Shortcut Connection (1x1, stride 2)
  |
  |-- BasicBlock
        |-- conv1 (3x3, 512) + BN + ReLU
        |-- conv2 (3x3, 512) + BN
        +-- No Shortcut Connection
```