

# 뇌종양 여부 분류 프로젝트

컴퓨터정보공학과 202447019신기재

2024. 12. 04

# 문제 정의

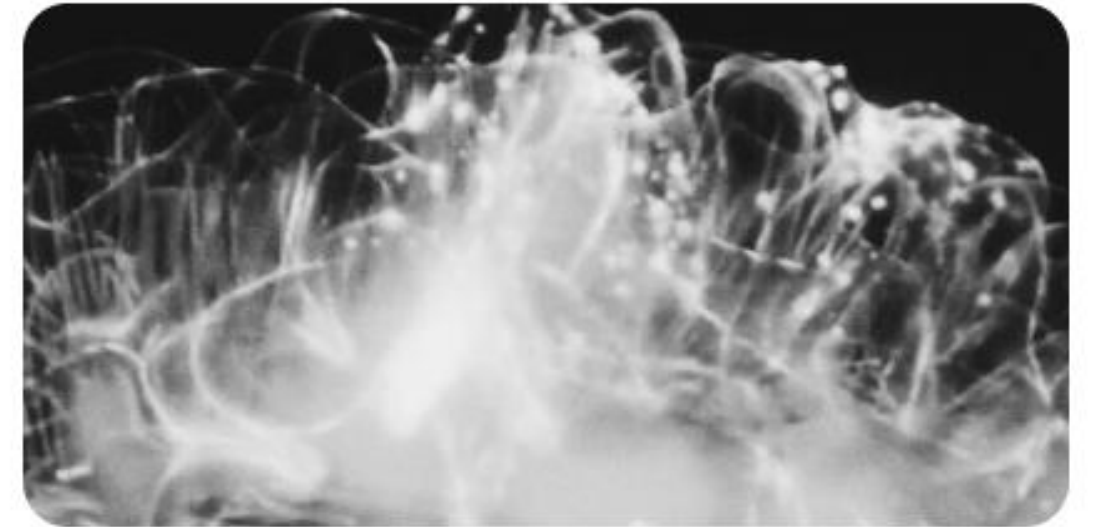


- MRI 이미지를 활용해 뇌종양 여부를 자동으로 분류하는 모델을 구현
- 입력 데이터 : 128x128 크기의 흑백 MRI 이미지
- 출력 데이터 : 종양의 존재 여부를 나타내는 이진 분류 결과 (Yes/No)

# 데이터 셋



## Brain MRI Images for Brain Tumor Detection



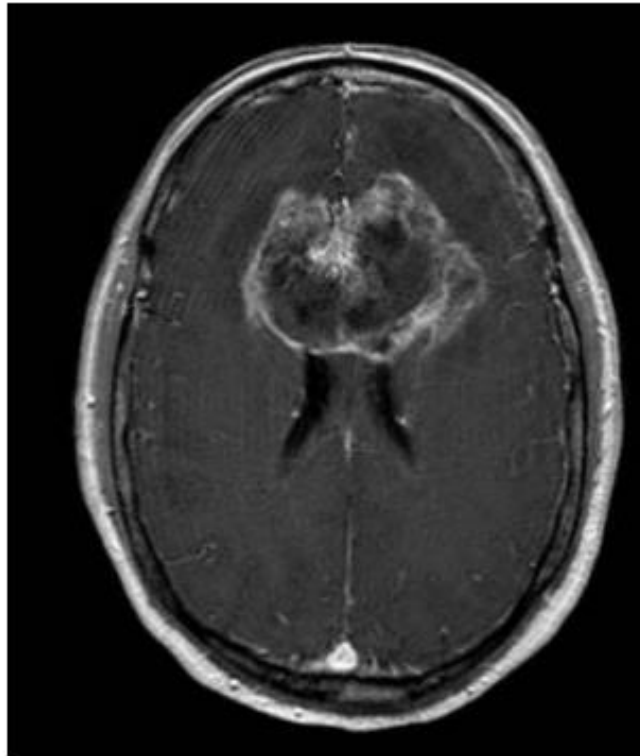
두 개의 클래스(뇌종양 여부)에 해당하는 MRI 이미지로 구성

출처 : 캐글

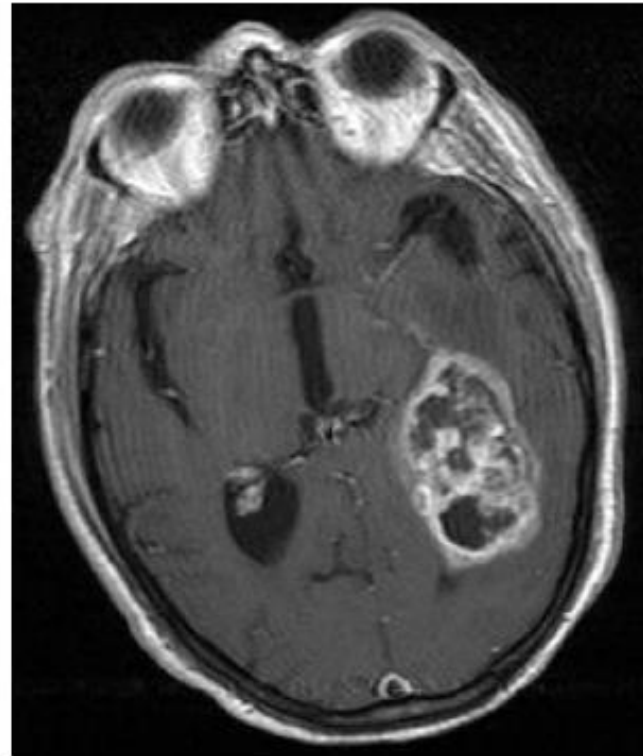
# 데이터 셋



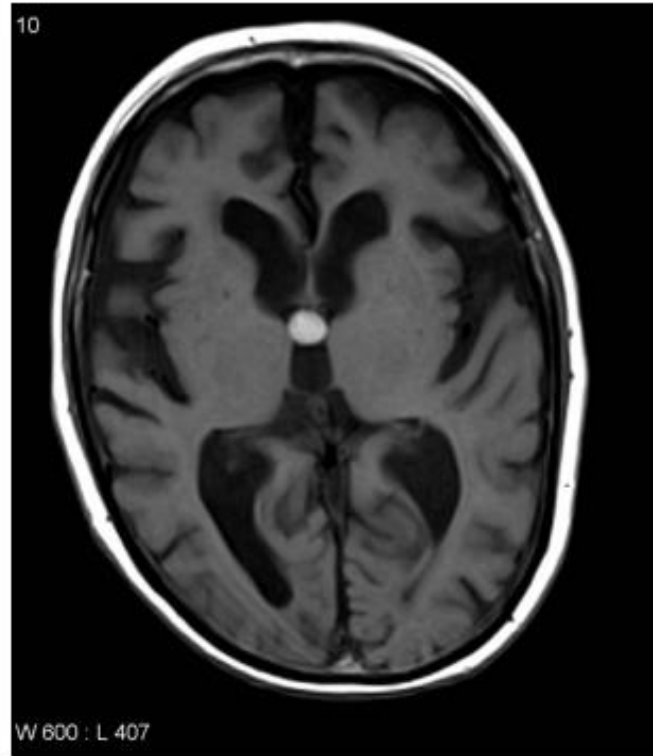
Yes



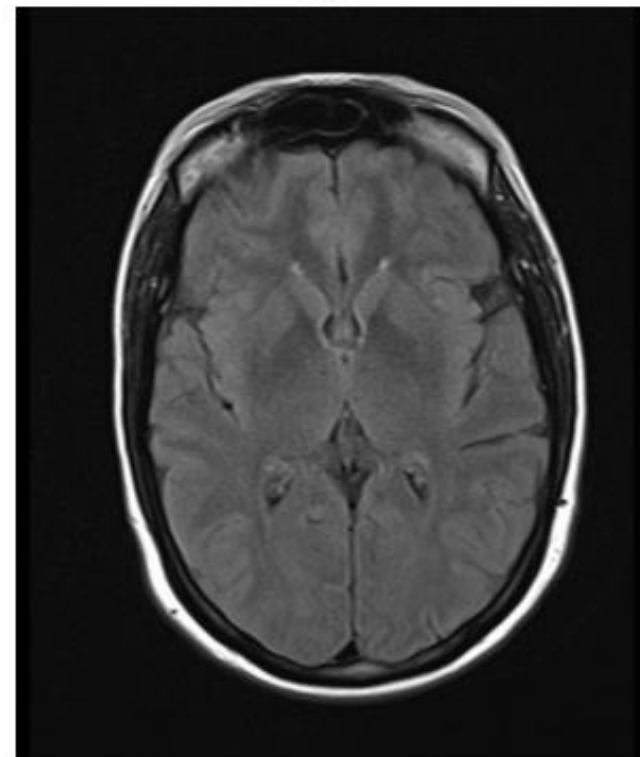
Yes



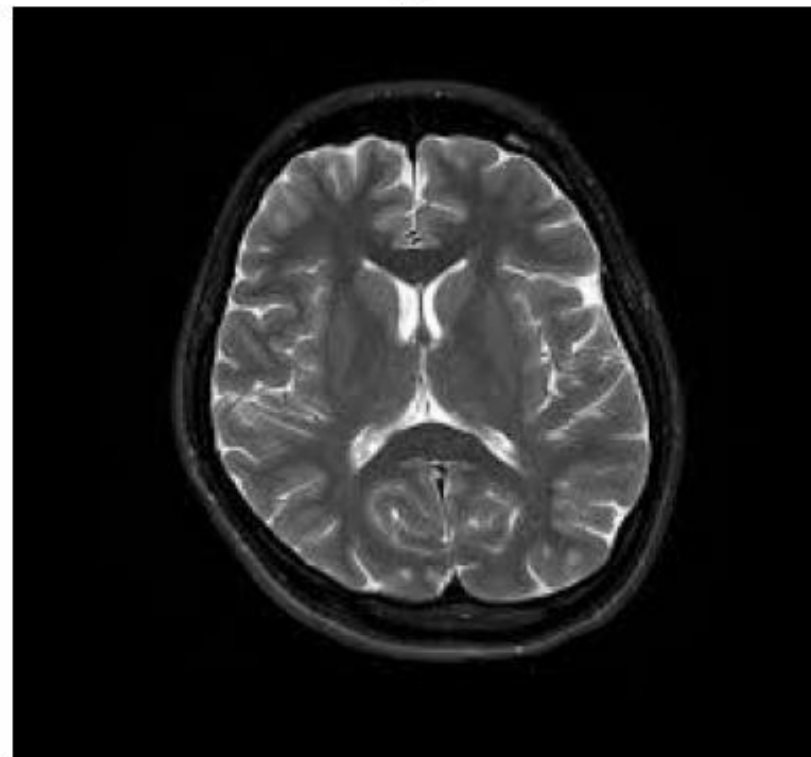
Yes



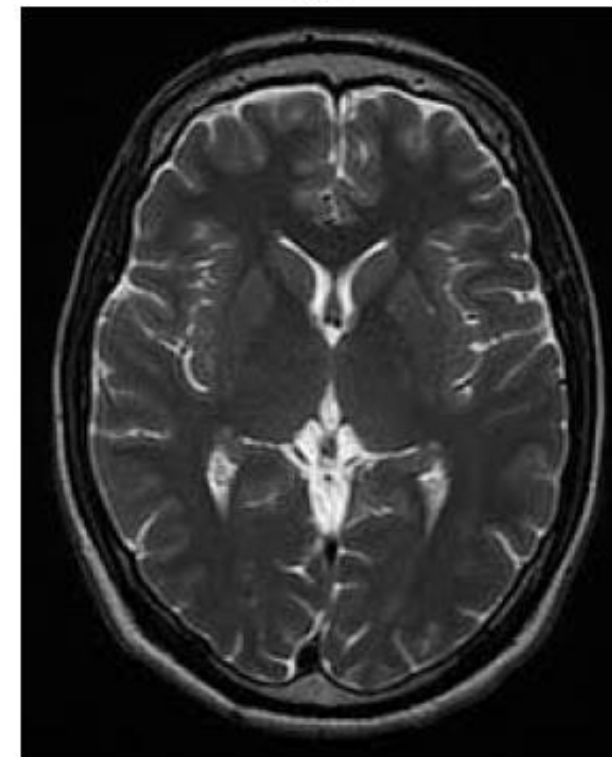
No



No



No



- Yes : 뇌종양 양성

- No : 뇌종양 음성

# 데이터 전처리



# 데이터 변환

```
transform = transforms.Compose([
    transforms.Grayscale(num_output_channels=1), # 흑백으로 변환
    transforms.Resize((128, 128)),             # 크기 조정
    transforms.RandomHorizontalFlip(p=0.5),     # 50% 확률로 좌우 반전
    transforms.RandomRotation(15),             # -15 ~ +15도 사이에서 랜덤 회전
    transforms.RandomAffine(degrees=0, translate=(0.1, 0.1)), # 랜덤 위치 이동
    transforms.ColorJitter(brightness=0.2, contrast=0.2),    # 밝기 및 대비 변화
    transforms.ToTensor(),                       # 텐서로 변환
    transforms.Normalize((0.5,), (0.5,))        # 정규화
])
```

- 흑백 변환, 크기 조정, 데이터 증강 적용

# 모델 구조 (CNN)



## Convolutional Layers

- 첫 번째 레이어: 입력 채널 1 → 출력 채널 32 (3x3 필터, 패딩 1)
- 두 번째 레이어: 입력 채널 32 → 출력 채널 64 (3x3 필터, 패딩 1)
- 각 레이어 뒤에 ReLU 활성화 함수와 MaxPooling(2x2) 적용

# 모델 구조 (CNN)



## Fully Connected Layers

- Flatten 후 첫 번째 Fully Connected Layer:  $64 \times 32 \times 32 \rightarrow 128$  뉴런
- 드롭아웃으로 과적합 방지
- 출력 레이어:  $128 \rightarrow 2$  (출력 클래스 수)
- 활성화 함수: ReLU
- 손실 함수: CrossEntropyLoss
- 최적화 알고리즘: Adam

# 하이퍼 파라미터 설정



```
# 하이퍼파라미터 설정  
batch_size = 16  
epochs = 30  
learning_rate = 0.001
```

```
Epoch [30/30], Loss: 0.4732473698946146  
Validation Accuracy: 78.43137254901961%
```



# 하이퍼 파라미터 변경



```
# 하이퍼파라미터 설정  
batch_size = 32  
epochs = 40  
learning_rate = 0.0001
```

```
Epoch [40/40], Loss: 0.4699926887239729  
Validation Accuracy: 82.3529411764706%
```

# 하이퍼 파라미터 변경



# 하이퍼파라미터 설정

batch\_size = 16

epochs = 30

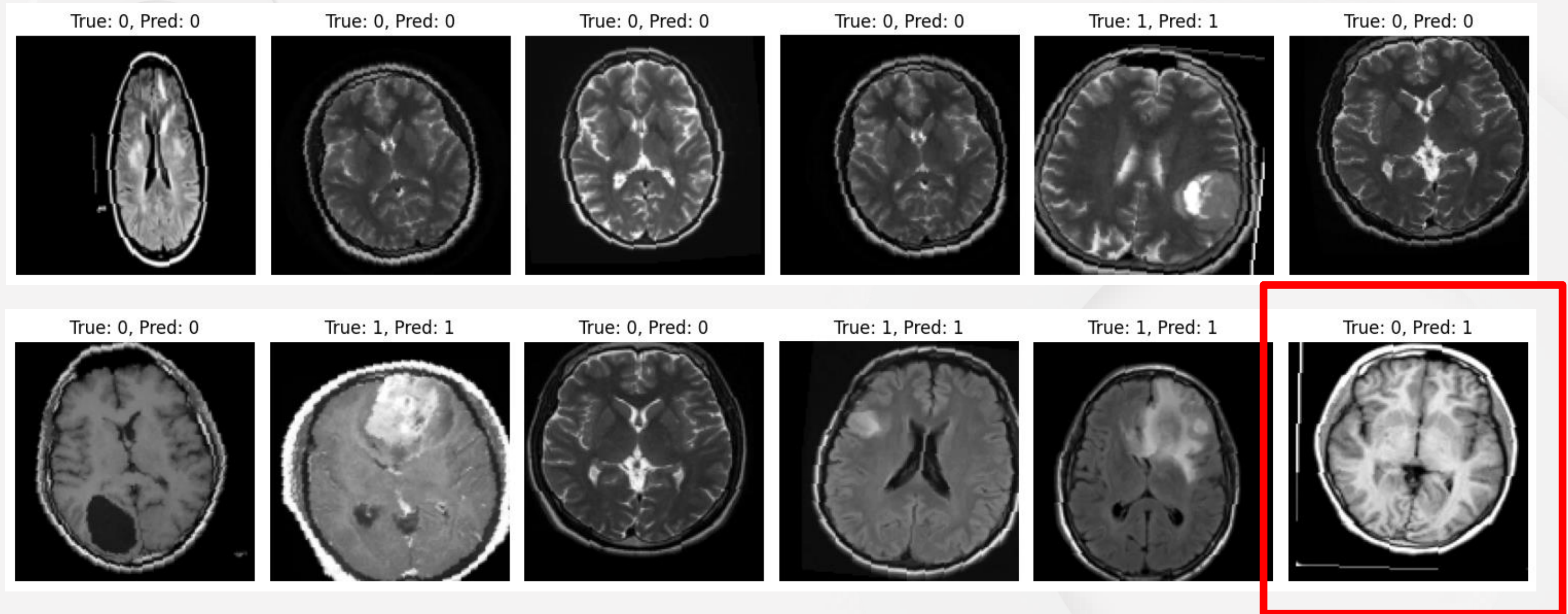
learning\_rate = 0.0005



초기 설정에서 학습률만 변경

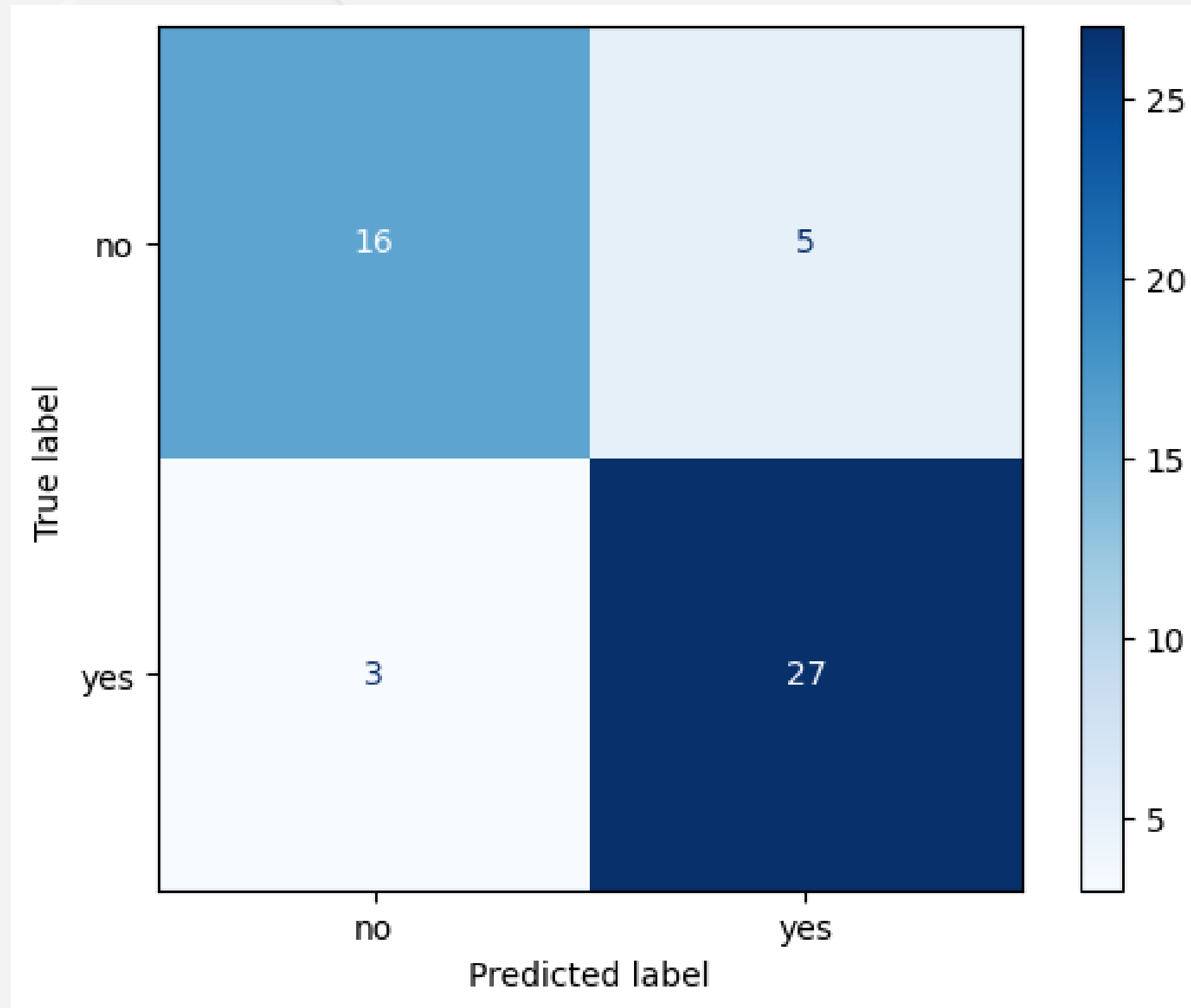
Epoch [30/30], Loss: 0.39339682918328506  
Validation Accuracy: 86.27450980392157%

# 예측 결과 시각화



- 대체로 잘 예측한 것으로 보임

# 모델 성능 시각화 (혼동 행렬)



**True Positive (TP): 27**  
실제 yes, 예측 yes 수

**False Positive (FP): 5**  
실제 no, 예측 yes 수

**False Negative (FN): 3**  
실제 yes, 예측 no 수

**True Negative (TN): 16**  
실제 no 예측 no 수

# 결론



- 모델 정확도 86%, 대체로 잘 분류
- 학습 과정에서 배치 크기, 학습률 등 하이퍼 파라미터를 조정함으로써 모델 성능 개선
- 규모가 큰 데이터 셋을 사용하여 진행하면 더 좋은 성능을 보일 것으로 예상

**Thank you  
For listening**

