

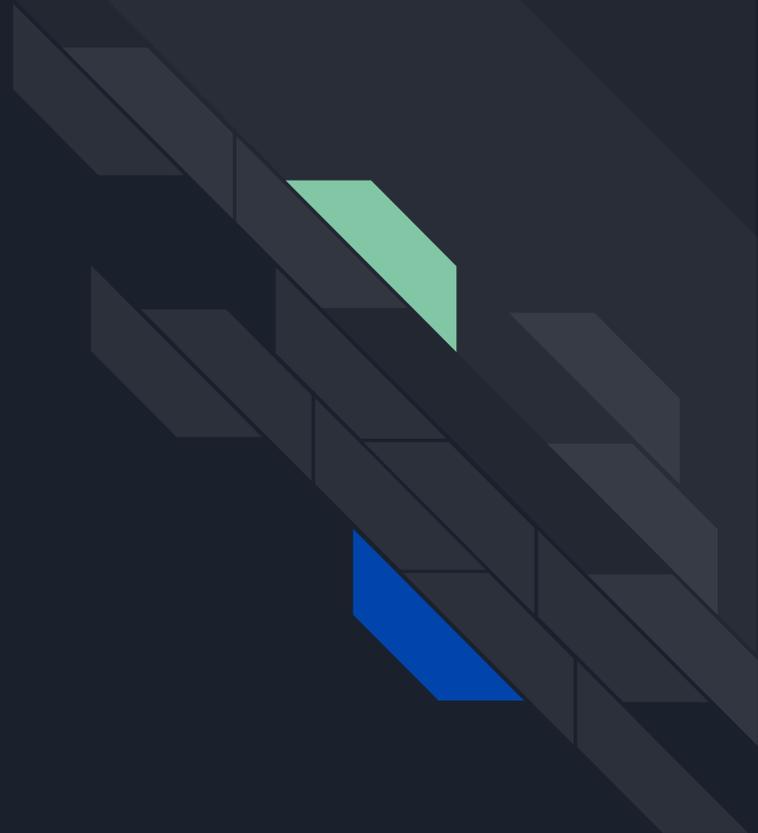


의료 영상 분석 개인 프로젝트

최경섭 - 데이터 과학자

발표 요약

- 개발자 소개
- 프로젝트 소개
- 데이터 분석
- 데이터 가공 / 학습 데이터 세트 생성
- 모델 학습
- 모델 튜닝 (현재 단계)
- 미래 계획





개발자 소개

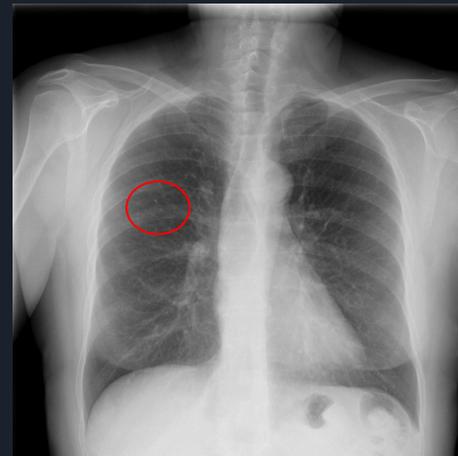
- 최경섭, 1994년생
- 캐나다 대학, 생물학 전공 준학사
- 총 개발 경력 2년 4개월
- 데이터 캠프, 데이터 과학자 인증 2024년 4월
- 딥 러닝/의료 영상 전문
- 웹, 머신 비전, 제조 머신러닝 경험

프로젝트 소개

01 딥러닝 사용하여 가슴 영상에서 폐렴 식별

02 Tensorflow, Pytorch, openCV, Pandas 라이브러리 사용

03 현재 모델 튜닝 단계 - 총 소요 기간 약 70시간





프로젝트 소개

프로젝트 구조

-연구

-보고서

-데이터

-피쳐

-모델

-테스트

-시각화

├ data	
├ external	<- 타사 소스의 데이터.
├ interim	<- 변환된 중간 데이터.
├ processed	<- 모델링을 위한 최종 데이터 세트.
└ raw	<- 원본 데이터.
├ docs	<- 기본 Sphinx 프로젝트; 자세한 내용은 sphinx-doc.org 참조
├ models	<- 훈련된 모델, 모델 예측 또는 모델 요약
└ Carmine400i70a.h5	<- 후방-전방 흉부 엑스레이에서 페렴 감지 70% 정확도
├ notebooks	<- Jupyter 노트북. 이름 규칙은 번호(정렬을 위해), 작성자의 이니셜, 그리고 짧은 설명, 예: `1.0-jqp-initial-data-exploration`.
├ references	<- 데이터 사전, 설명서 및 기타 모든 설명 자료.
└ research_papers	<- PDF 형식의 학술 연구 논문
├ reports	<- HTML, PDF, LaTeX 등으로 생성된 분석
└ figures	<- 보고서에 사용될 그래프 및 그림
├ src	<- 이 프로젝트에서 사용되는 소스 코드.
├ __init__.py	<- src를 Python 모듈로 만들
├ data	<- 데이터 다운로드 또는 생성 스크립트
└ make_dataset.py	<- 원본 데이터를 모델링을 위한 특징으로 변환하는 스크립트
├ features	<- 원본 데이터를 모델링을 위한 특징으로 변환하는 스크립트
├ build_features.py	<- 페렴 진단을 위한 800개의 이미지 데이터 세트를 생성하는 스크립트
└ create_sliced.py	<- 슬라이스된 CSV 데이터 세트를 생성하는 스크립트
├ models	<- 모델을 훈련하고 훈련된 모델을 사용하여 예측하는 스크립트
├ predict_model.py	<- 이미지를 통해 페렴을 예측하기 위해 Carmine400 모델을 사용하는 스크립트
└ train_model.py	<- 흉부 엑스레이에서 페렴을 감지하기 위해 400개의 이미지를 사용하여 Carmine400 모델을 훈련하는 스크립트
├ tests	<- 자동화된 테스트를 실행하는 스크립트
├ carmine_test.py	<- Carmine 모델을 테스트하는 스크립트
└ tangerine.py	<- Tangerine 모델을 테스트하는 스크립트
├ visualization	<- 탐색적 및 결과 지향 시각화를 생성하는 스크립트
├ exploratory.py	<- 탐색적 데이터 분석을 수행하는 스크립트
├ image_count.py	<- "raw/img" 폴더 내 이미지 수를 세는 스크립트
└ visualize.py	<- OpenCV를 사용하여 흉부 엑스레이를 시각화하는 스크립트
├ LICENSE	
├ Makefile	<- `make data` 또는 `make train`과 같은 명령을 포함한 Makefile
└ README.md	<- 이 프로젝트를 사용하는 개발자를 위한 최상위 README.



데이터 분석

- BIMCV-COVID19+ 데이터세트는 발렌시아 지역 의료 영상 은행(BIMCV)의 의료 영상 데이터뱅크에서 CR, DX, CT 이미지 포함된 대규모 데이터 세트입니다.

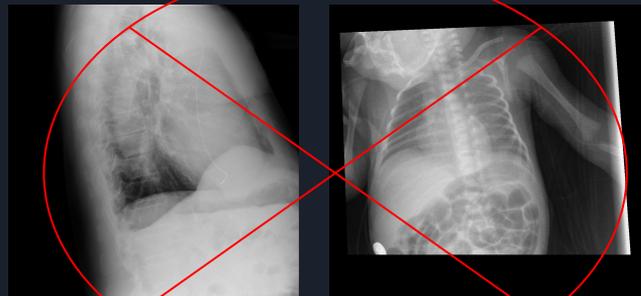
- 23,521 이미지 파일(약 152GB)

- EDA 작업, Pandas 라이브러리로 `.head()`, `.describe()`, `.shape`, `.info()` 사용

- 26020열, 28행, Null 데이터 존재

데이터 가공 / 학습 데이터 세트 생성

- “정상” 이미지 200, “폐렴” 이미지 200
- Projection == “PA” (전 방향 검사)
- Pediatric == “No” (소아과 제외)

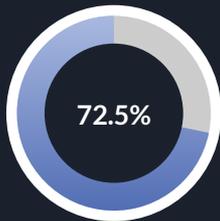


모델 학습

- Carmine400i70aDeploy.h5 모델
- Tensorflow, Keras
- ResNet50 모델 기반 (CNN, 50 Layers)



Validation loss



Accuracy
(모델 정확도)

The screenshot shows a Jupyter Notebook environment with a file explorer on the left and a code editor on the right. The code defines a ResNet50 model, compiles it with Adam optimizer, and trains it for 10 epochs. The output shows training progress and final metrics.

```
layer.trainable = False # Freeze layers of the pre-trained model

model = Sequential([
    base_model,
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(num_classes, activation='softmax')
])

return model

# Define input shape and number of classes
input_shape = (224, 224, 3) # Assuming input size for the model is 224x224 with 3 channels (RGB)
num_classes = 2 # Number of classes (normal and abnormal)

# Create the model
model = create_resnet_model(input_shape, num_classes)

# Compile the model
model.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model
history = model.fit(X_train, y_train, batch_size=32, epochs=10, validation_data=(X_val, y_val))

# Evaluate the model
loss, accuracy = model.evaluate(X_val, y_val)
```

Run: traindd x

```
10/10 ----- 11s 1s/step - accuracy: 0.6018 - loss: 0.6630 - val_accuracy: 0.7250 - val_loss: 0.6133
Epoch 8/10
10/10 ----- 11s 1s/step - accuracy: 0.6437 - loss: 0.6411 - val_accuracy: 0.7000 - val_loss: 0.6083
Epoch 9/10
10/10 ----- 11s 1s/step - accuracy: 0.6119 - loss: 0.6475 - val_accuracy: 0.7125 - val_loss: 0.6175
Epoch 10/10
10/10 ----- 11s 1s/step - accuracy: 0.6529 - loss: 0.6365 - val_accuracy: 0.7250 - val_loss: 0.5930
3/3 ----- 2s 699ms/step - accuracy: 0.7258 - loss: 0.5900
Validation Loss: 0.5930191874504089, Validation Accuracy: 0.7250000238418579

Process finished with exit code 0
```



모델 튜닝 (현재 단계)

- 데이터셋 확대 60%
- Input Shape 확대 60%
- 데이터 품질 향상



웹 배포





미래 계획





감사합니다

발표 끝입니다

