custom DNN 및 ResNet 을 이용한 폐렴 진단 성능 비교

Members :

박설희, 생명과학과 25학번, pshee050327@naver.com

이준승, 융합전자공학부 25학번, junseung114@naver.com

최선웅, 컴퓨터소프트웨어학부 24학번, sunwoong312@hanyang.ac.kr

김진태, 컴퓨터소프트웨어학부 20학번, jintea01@naver.com

0. 실행환경

- **CPU 모델**: Intel® Xeon® Gold 6336Y @ 2.40GHz
- **코어 수**: 24 cores / 48 threads
- **클럭 속도**: 최대 2.40GHz, 현재 800MHz (idle 상태)
- **캐시**: L3 캐시 36MB

1. Proposal (Option A)

Option A를 선택한 이유는 단순히 특정 알고리즘이나 이론을 정리하는 것을 넘어서, 실제로 폐렴 진단에 도움을 줄 수 있는 모델을 만들어보고 싶다는 생각에서 출발했기 때문입니다. ResNet18 같은 널리 알려진 모델과 비교해가며 제가 직접 설계한 모델의 성능을 평가함으로써, 현장에서 어떻게 활용될 수 있을지 고민해보고 싶었습니다.

2. Datasets

https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

2.1 개요

Kaggle에서 제공하는 **“Chest X-Ray Images (Pneumonia)”** 공개 데이터셋을 사용하였습니다. 이 데이터셋은 흉부 X-ray 이미지를 기반으로 **정상(Healthy)** 과 **폐렴(Pneumonia)** 을 이진 분류(Binary Classification)하는 데 활용됩니다. 데이터는 중국 광

저우 여성·아동 의료센터(Guangzhou Women and Children’s Medical Center)에서 소아 환자들을 대상으로 수집된 것입니다.

2.2 구성

데이터셋은 다음의 세 개 디렉토리로 구성되어 있습니다:

- **train/** – 모델 학습에 사용되는 이미지
- **val/** – 학습 중 모델의 성능을 점검하기 위한 검증용 이미지
- **test/** – 학습이 완료된 모델을 최종적으로 평가하기 위한 테스트 이미지

2.3 데이터 수

- **훈련 데이터(train)**
  - NORMAL: 1,341장
  - PNEUMONIA: 3,875장
- **검증 데이터(val)**
  - NORMAL: 8장
  - PNEUMONIA: 8장
- **테스트 데이터(test)**
  - NORMAL: 234장
  - PNEUMONIA: 390장

※ 정확한 수치는 데이터셋 버전에 따라 다를 수 있으므로, 실제 실험 시에는 직접 확인하여 사용하였습니다.

3. Metholodgy

본 프로젝트는 흉부 X-ray 이미지를 입력으로 받아 폐렴 여부를 분류하는 이진 분류 문제(Binary Classification Problem)입니다.

본 프로젝트에서는 흉부 X-ray 이미지를 기반으로 폐렴 여부를 분류하기 위해 세 가지 다른 수준의 분류 모델을 구현하고 성능을 비교하였습니다.

(1) 로지스틱 회귀(Logistic Regression)

(2) 커스텀 심층 신경망(Deep Neural Network, 이하 DNN)

(3) 사전학습된 ResNet(Residual Network)

각 모델은 학습 효율성, 예측 정확도, 학습 시간, 복잡도 등을 기준으로 비교 분석하였습니다.

3.1 모델 구성 및 특징

항목	**LogisticModel**	**DeepNN**	**ResNet18**
모델 구조	선형 회귀 (Linear)	다층 퍼셉트론 (MLP, Fully Connected Layers)	합성곱 신경망 (CNN, Residual Network)
입력 크기	`3 × 224 × 224` 전개 후 150528 픽셀	동일	동일
파라미터 수	적음 (단일 선형 계층)	중간 (4개의 FC 계층)	많음 (CNN 계층 + 잔차 블록 포함)
비선형성	없음	ReLU 활성화 함수 사용	ReLU 및 BatchNorm 포함
사전학습 여부	X	X	O (ImageNet으로 사전학습)
특징 추출 방식	전체 픽셀을 평탄화하여 입력	평탄화된 입력에 대해 점진적 특징 추출	계층적 특징 추출 (edge → texture → shape 등)
학습 속도	빠름 (가장 단순)	중간	느림 (가장 복잡)
성능 기대치	낮음 (단순 모델, 표현력 부족)	중간 (표현력 증가)	높음 (복잡한 이미지에서 성능 우수)
사용 목적	베이스라인 비교용	커스텀 성능 개선 모델	강력한 CNN 기반 비교 대상

로지스틱 모델은 가장 기본적인 NN의 형태로써 각 픽셀에 weight 를 mac 연산한 결과입니다. 최종 출력은 이진 분류 문제이기 때문에 BC 를 사용했습니다.

DNN 모델은 자체적으로 layer 개수, hidden unit 의 개수 등을 커스터마이징한 모델입니다. 다양한 조합을 시도한 결과 가장 성능이 우수하게 나온 값을 선정하였습니다. 추가적으로 하드웨어 친화적으로 개수를 설정하는 등의 테크닉이 들어갔습니다.

ResNet18모델은 Residual Network 계열의 심층 신경망으로, 18개의 레이어(17개의 Convolution + 1 Fully Connected)를 사용해 이미지 분류를 수행합니다. Skip Connection(잔차 연결)을 통해 gradient vanishing 문제를 완화하여 딥러닝 학습을 안정화합니다. 이미지넷 분류 등 다양한 컴퓨터 비전 문제에서 높은 성능을 보이며, 구조가 간단해 실험용 모델로 널리 사용되기에 비교군으로 설정했습니다.

3.2 데이터 전처리

- **크기 조정 (Resize)**
  - 모든 이미지를 `224 x 224` 픽셀로 크기 통일했습니다.
- **텐서 변환 (ToTensor)**
  - 이미지 데이터를 PyTorch가 처리할 수 있는 `Tensor` 타입으로 변환했습니다.
  - 픽셀 값 범위를 `[0, 255]`에서 `[0.0, 1.0]` 구간으로 Normalization 했습니다.

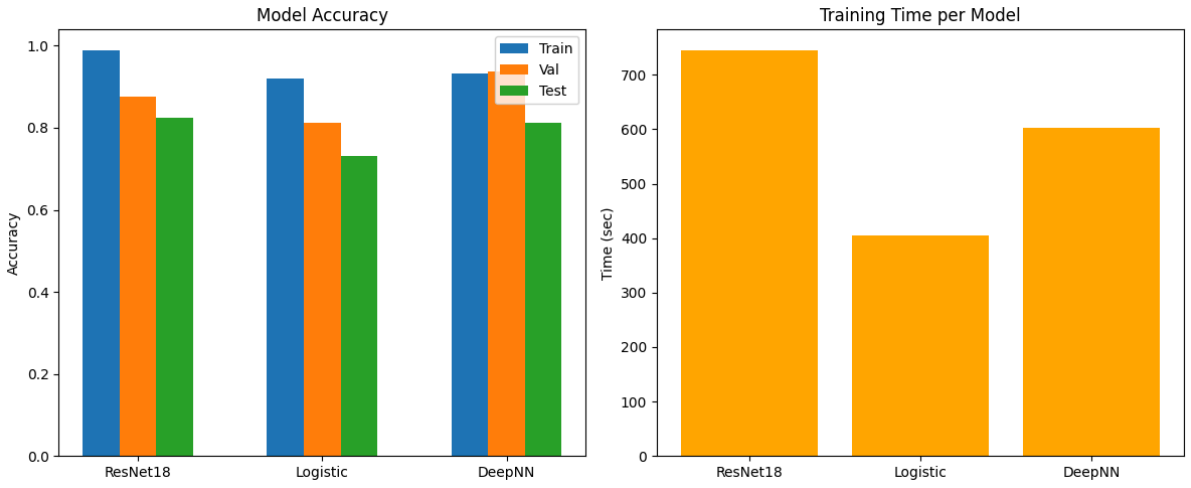
3.3 학습 및 평가 설정

- lr (learnning rate): 1e-4
- loss function: Cross-Entropy
- optimizer: Adam
- batch size: 32
- epoch: 10

cross-validation 및 early stopping, 추가적인 data augmentation 은 하지 않았습니다.

4. Evaluation & Analysis

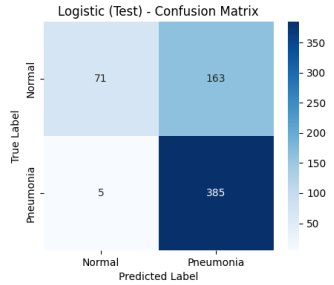
4.1 정확도(Accuracy) 및 학습 시간 비교

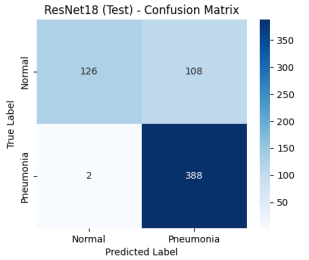


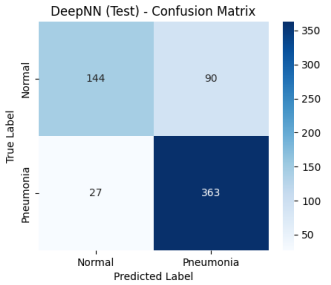
단순히 모든 정확로 Accuracy 를 그래프로 확인한 결과 RexNet18 이 가장 정확한 성능임을 볼 수 있습니다. 이후의 섹션(4.2, 4.3) 에서는 조금 더 세부적으로 class 별로 어떻게 결과가 나왔는지 확인해보도록 하겠습니다.

모델별 Training time 을 살펴보면 ResNet18 이 가장 오래 걸리는 모습을 볼 수 있습니다. layer 가 더 깊어서 당연한 결과라고 생각이 됩니다. ResNet18 모델은 약 720초(12분), Logistic 모델은 약 400초(6분40초), DNN모델은 약 600초(10분) 소요되는 것을 볼 수 있습니다.

4.2 Confusion Matrix 분석 (Test Set 기준)







아래는 분류 결과를 표로 작성했습니다.

모델	진짜 정상 → 정상	진짜 정상 → 폐렴	진짜 폐렴 → 정상	진짜 폐렴 → 폐렴
Logistic	71	163	5	385
DeepNN	144	90	28	363
ResNet18	126	108	2	388

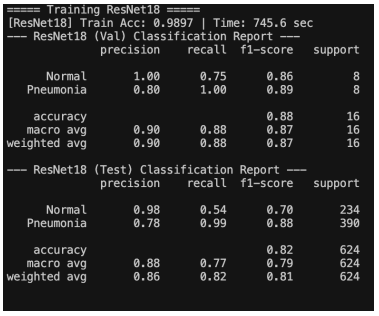
추가적으로 진단수치를 정확도, Recall, Precision 을 표로 계산한 결과 입니다.

모델	정확도	재현율(Recall)	정밀도 (Precision)	특이사항
Logistic	73.1%	**98.7%**	**70.2%**	Pneumonia 민감하지만 오탐 많음
DeepNN	81.3%	93.1%	80.1%	Normal 분류에 강함
ResNet18	**82.4%**	**99.5%**	78.2%	**폐렴 감지에 매우 우수**, 오탐 많음

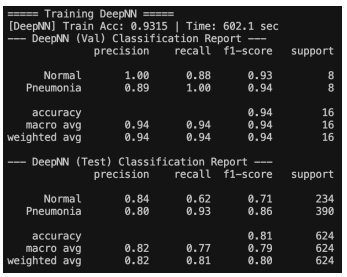
의료 시스템의 특성상 폐렴 진단에서는 recall 값이 중요한 지표로 볼 수 있습니다. ResNet18 모델은 99.5%의 recall로 가장 우수한 성능을 보여주었으며, Logistic 모델은 98.7%로 그 뒤를 이었습니다. 그러나 Logistic 모델의 precision 값은 70.2%로 상대적으로 낮아, 위양성(false positive)이 발생할 가능성이 상당히 높은 것으로 분석됩니다. 보다

종합적인 성능 평가를 위해, 이어지는 분석(섹션 4.3)에서 각 class 별 F1 Score 를 확인하여 Precision과 Recall의 조화 평균을 살펴보겠습니다. 추가적으로 class 별 진단수치도 함께 파악하겠습니다.

4.3 Classification Report (Test Set 기준)



```
===== Training ResNet18 =====
[ResNet18] Train Acc: 0.9897 | Time: 745.6 sec
--- ResNet18 (Val) Classification Report ---
              precision    recall  f1-score   support

      Normal       1.00      0.75      0.86         8
   Pneumonia       0.80      1.00      0.89         8

    accuracy                           0.88        16
   macro avg       0.90      0.88      0.87        16
weighted avg       0.90      0.88      0.87        16

--- ResNet18 (Test) Classification Report ---
              precision    recall  f1-score   support

      Normal       0.98      0.54      0.70       234
   Pneumonia       0.78      0.99      0.88       390

    accuracy                           0.82       624
   macro avg       0.88      0.77      0.79       624
weighted avg       0.86      0.82      0.81       624
```



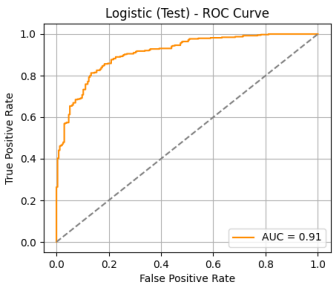
```
===== Training DeepNN =====
[DeepNN] Train Acc: 0.9315 | Time: 602.1 sec
--- DeepNN (Val) Classification Report ---
              precision    recall  f1-score   support

      Normal       1.00      0.88      0.93         8
   Pneumonia       0.89      1.00      0.94         8

    accuracy                           0.94        16
   macro avg       0.94      0.94      0.94        16
weighted avg       0.94      0.94      0.94        16

--- DeepNN (Test) Classification Report ---
              precision    recall  f1-score   support

      Normal       0.84      0.62      0.71       234
   Pneumonia       0.80      0.93      0.86       390

    accuracy                           0.81       624
   macro avg       0.82      0.77      0.79       624
weighted avg       0.82      0.81      0.80       624
```

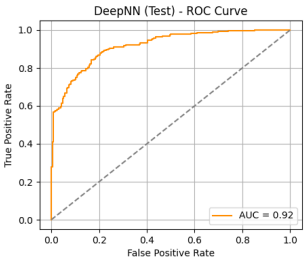
```
===== Training Logistic =====
[Logistic] Train Acc: 0.9209 | Time: 404.1 sec
--- Logistic (Val) Classification Report ---
              precision    recall  f1-score   support

      Normal       1.00      0.62      0.77         8
   Pneumonia       0.73      1.00      0.84         8

    accuracy                           0.81        16
   macro avg       0.86      0.81      0.81        16
weighted avg       0.86      0.81      0.81        16

--- Logistic (Test) Classification Report ---
              precision    recall  f1-score   support

      Normal       0.93      0.30      0.46       234
   Pneumonia       0.70      0.99      0.82       390

    accuracy                           0.73       624
   macro avg       0.82      0.65      0.64       624
weighted avg       0.79      0.73      0.68       624
```

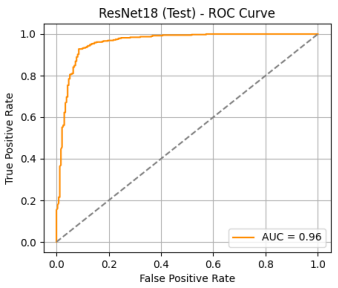
- **ResNet18** 모델에서 폐렴환자에 대한 Recall 값은 0.99 입니다. normal class 에 대해서는 0.54의 값을 지닙니다. 즉, 폐렴환자 진단에 매우 뛰어나지만, 정상 마저 폐렴으로 오진할 가능성이 높습니다.
- **DeepNN** 모델에서 폐렴환자에 대한 Recall 값은 0.93 입니다. normal class 에 대해서는 0.62의 값을 지닙니다.
- **Logistic** 모델은 위의 두 모델에 대해서 그 어떤 부분도 좋은점을 찾을 수가 없습니다.

F1 score 는 recall 과 precision 의 조화평균값입니다. 위양성의 가능성, 즉, false positive 역시 무시할 수 없는 값이기 때문에 이를 반영한 수치라고 볼 수 있습니다. 허나, 의료시스템은 환자를 놓치지 않는 것이 우선순위여야 한다고 생각하기 때문에 종합적으로 ResNet18 이 더 나은 모델이라고 판단됩니다.

4.4 ROC Curve & AUC 분석







ROC 곡선은 성능 비교를 시각적으로 비교할 수 있습니다.

AUC 값은 해당 곡선의 아래 면적 입니다. 즉, 그래프가 좌상단에 몰려있을수록 더 나은 성능임을 알 수 있습니다. python 을 이용해 값을 얻은 결과

- **ResNet18**: AUC(Test) = 0.96
- **DeepNN**: AUC(Test) = 0.92
- **Logistic**: AUC(Test) = 0.91

ResNet 18 모델이 가장 좋은 성능을 얻어냈다는 것을 볼 수 있습니다.

5. Related Work

5.1 한양대학교 AI+X:머신러닝 과목

'Ai+x:머신러닝' 과목에서 본 matplotlib 관련 tutorial 들을 참고해서 실제 코드에 적용하는데 큰 도움이 되었습니다. 특히 데이터 분포, 학습 곡선, 모델 비교 결과 등 시각적인 자료를 도출해낼 때 참고한 예제가 많았습니다.

5.2 블로그

https://ai-com.tistory.com/entry/ML-분류-성능-지표-Precision정밀도-Recall재현율

각종 분류 성능 지표의 의미를 파악하는 용도로 사용했습니다.

https://diseny.tistory.com/entry/ROC-곡선-아주-쉽게-이해하기

roc 곡선의 의미를 파악하는 용도로 사용했습니다.

5.3 논문

https://arxiv.org/pdf/1512.03385

ResNet18 의 구조를 파악하기위해 읽은 논문입니다.

6. Conclusion: Discussion

이번 프로젝트를 통해 직접 개발한 모델과 기존의 ResNet18 모델을 비교한 결과, ResNet18 모델이 가장 높은 성능을 보여주었습니다. 이러한 결과를 통해 의료 인공지능 시스템을 구축할 때 이미 검증된 모델을 가져와 사용하는 것이 직접 개발하는 것보다 효율적일 수 있다는 점을 알 수 있었습니다. 비단 의료 인공지능 시스템 뿐만 아닙니다. AI를 활용한 다양한 서비스를 개발함에 있어서 직접 모델을 만들고, 다양한 파라미터 값들을 수정해가며 최적의 모델을 찾는것도 의미가 있겠지만, 높은 확률로 시중에 있는 모델을 가져다 사용하는 것이 시간적으로, 또 비용적으로 저렴할 수 있다는 것을 알았습니다.

이는 소프트웨어 개발에서 검증된 라이브러리를 가져다 사용하는 것이 안정적이고 빠르게 결과를 낼 수 있는 것과 같은 맥락입니다. 따라서 앞으로의 연구와 실무에서는 직접 모델을 개발하기보다는 기존에 검증된 모델을 활용해 fine-tuning을 하거나 성능 개선을 위한 추가적인 방법을 적용하는 방향이 더 합리적이라고 생각합니다.

7. Role

- **박설희**: 데이터셋 수집 및 전처리 자료 조사, 실험 결과 분석 및 시각화
- **이준승**: 영상 촬영, 실험 데이터 기록, 발표 자료 보조, 최종 보고서 작성
- **최선웅**: 코드 작성(데이터셋 로딩, 모델 학습, 평가), 데이터 전처리 구현, 방법론 설계 및 제안
- **김진태**: 코드 실행 및 디버깅, 하드웨어 및 소프트웨어 환경 설정, 보고서 검토