CNN을 활용한
흉부 X-ray 이미지 분류

1조 | 김서현 김연성 서문홍 조건우 최지은

INDEX

01. Introduction

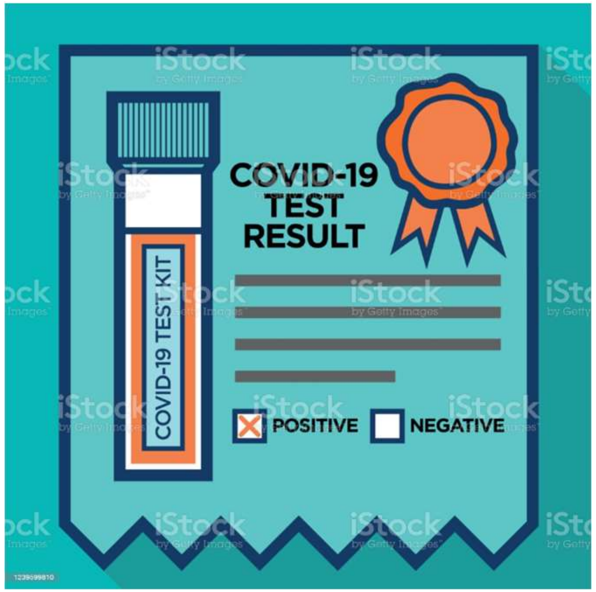
"코로나19"의 위험성 증대

⇨ 팬데믹 상황 속 비효율적인 대응이 이루어짐

기존의 문제 상황

코로나19 PCR 검사는 시간과 비용 측면에서 비효율적이다.

- 의료진들의 공공 인력 낭비, 의료 인력 부족
- 검사 결과를 확인하기 위해 하루라는 시간이 소요
- 약 8만원이라는 높은 검사 비용
- 검사 과정에 고통과 불쾌감 유발 가능
- 검사 장소에 여러 사람이 모이게 되어 또 다른 전파의 위험성 증대
- 일회성 검사도구로 인한 환경 오염



코로나19와 폐렴을 가시적으로 구분하기 어렵다.

- 비슷한 증상: 기침, 발열, 오한
- 감염경로의 유사성: 비말감염, 간접전파
- X-ray 상으로 구분 어려움 (데이터 수집 사례가 적음)

목표 설정

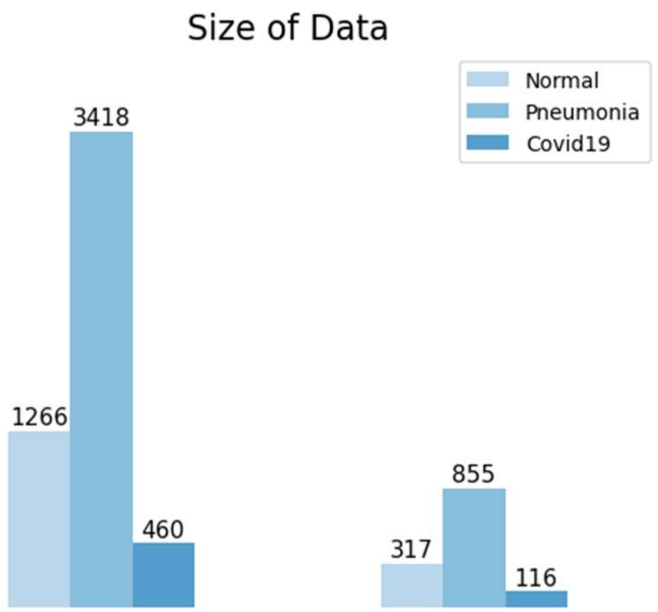
보다 효율적인 새로운 코로나19 판별 검사 필요
X-ray 사진만으로 바로 분류! 사람이 아닌 기계가 바로 분류하는 새로운 코로나19 검사법 제안

또한, 코로나19와 폐렴 데이터가 충분하지 않아서 적은 데이터로도 딥러닝을 할 수 있는 CNN 방식이 필요하다.

02. Data

EDA

- 데이터 세트는 train, test로 구성
- train, test 모두 3개의 하위 폴더(COVID19, PNEUMONIA, NORMAL)를 포함
- 데이터 세트에는 총 6432개의 X-ray 이미지가 포함되어 있으며 테스트 데이터는 전체 데이터의 20%를 차지



Train set 데이터 개수의 불균형

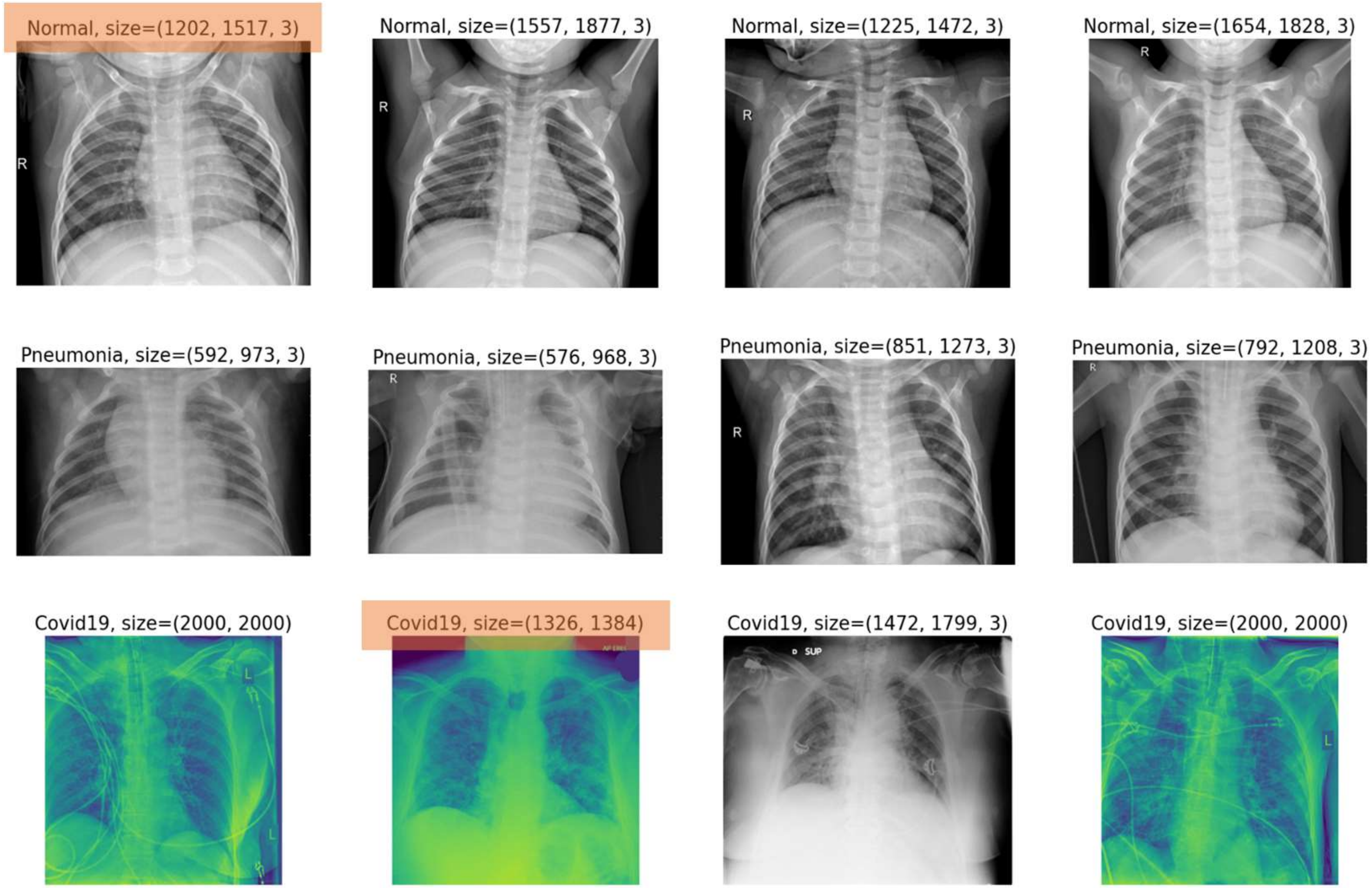
EDA

Data size

데이터 세트 별 X-ray 이미지 확인

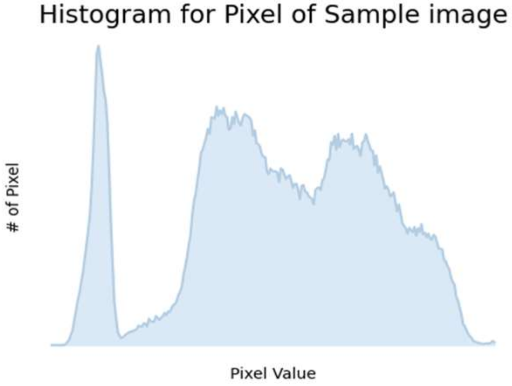
▼

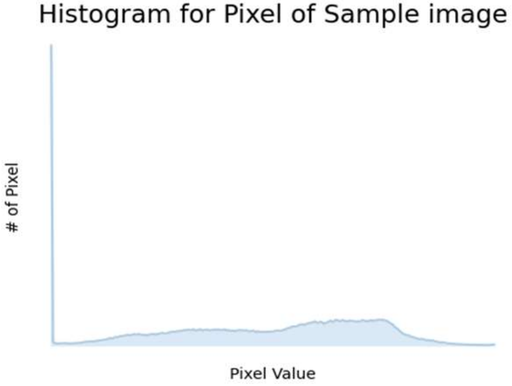
이미지 별 **픽셀 사이즈, 채널 수** 모두 다름



EDA

Image Histogram





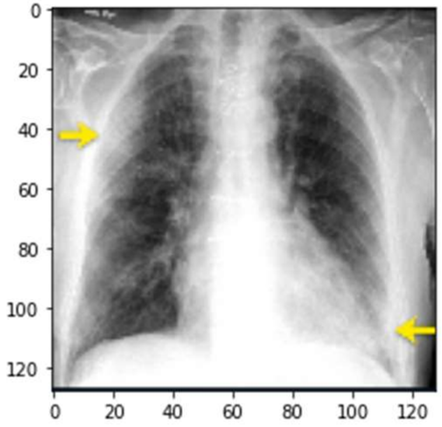
X-ray 이미지 별 Pixel의 분포를 알 수 있는 Image histogram 확인

▼

픽셀 분포가 불균등하여 이미지의 contrast가 떨어짐

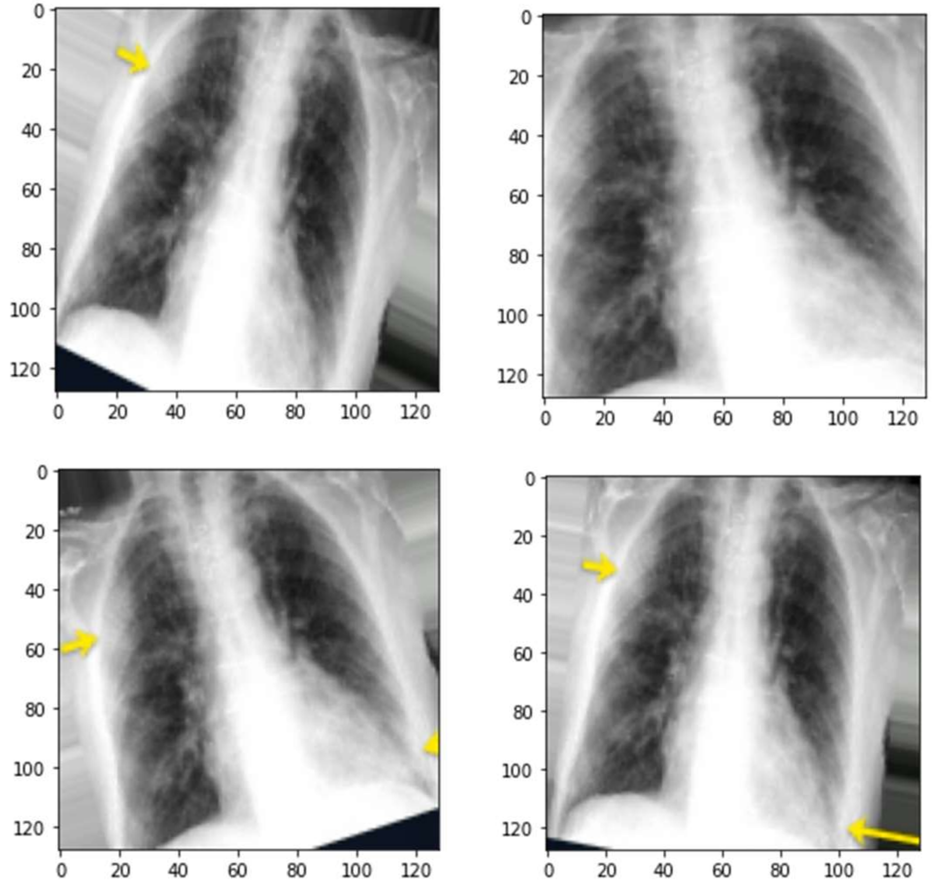
Data Preprocessing

Data Augmentation



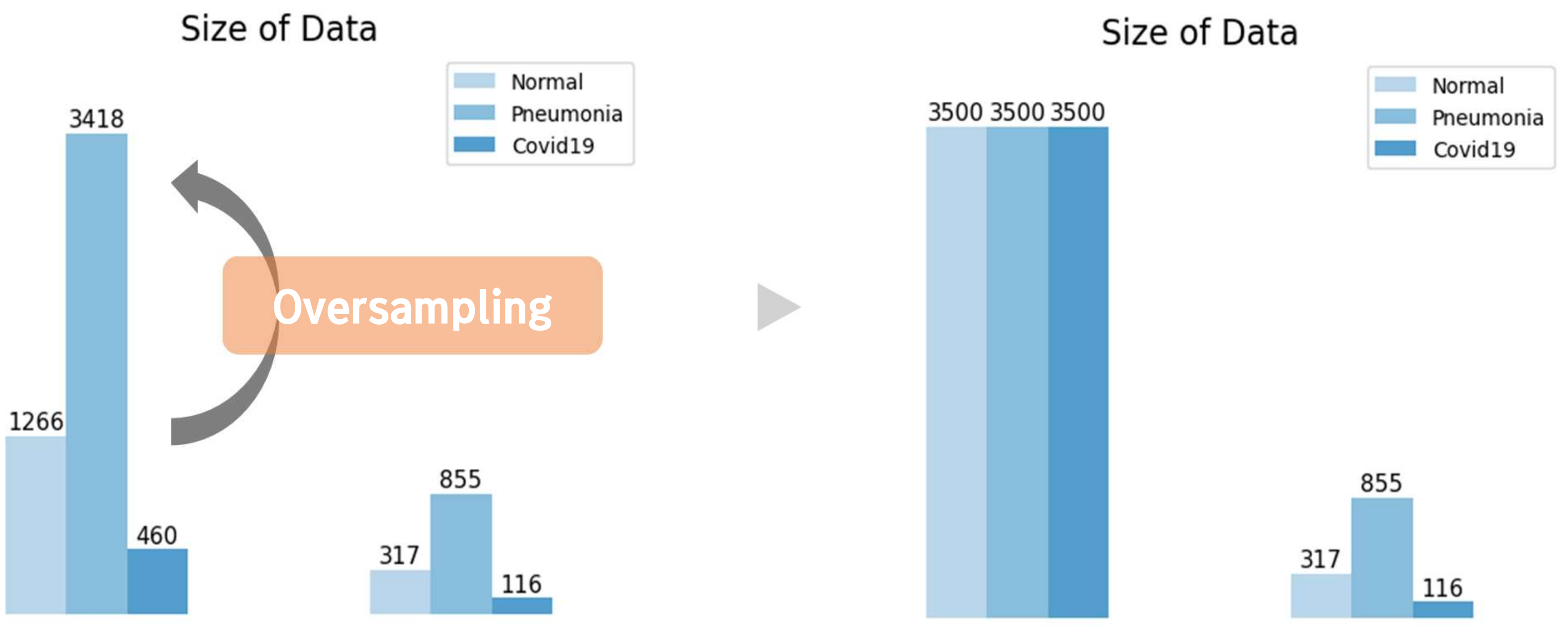
Size
(128, 128, 3)

Random
- Rotation
- Shear
- zoom



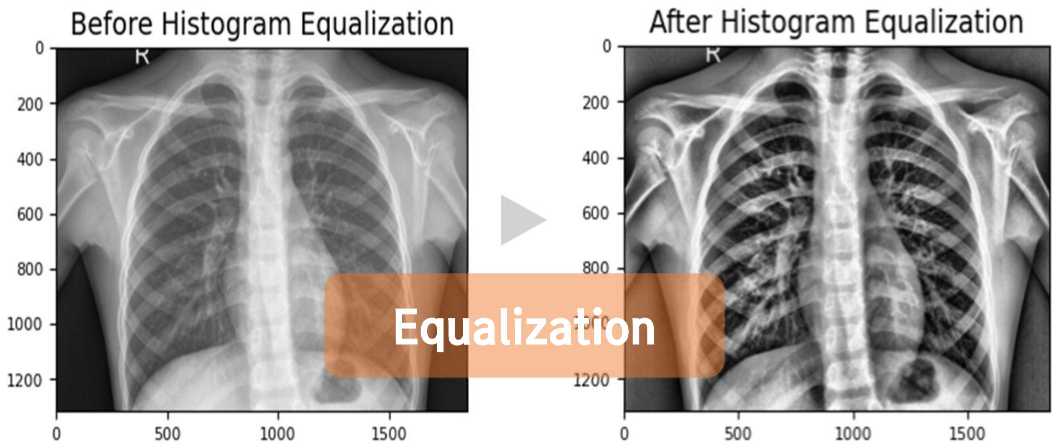
Data Preprocessing

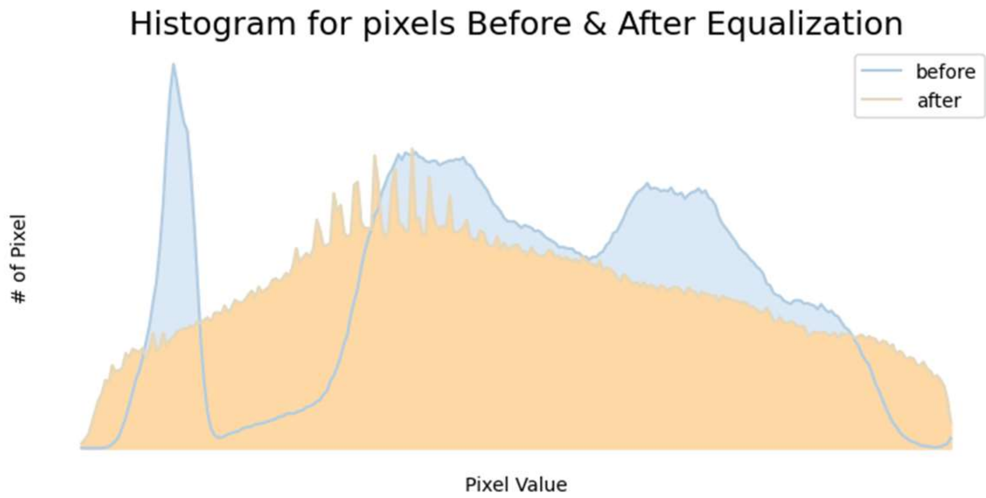
Oversampling



Data Preprocessing

Histogram Equalization

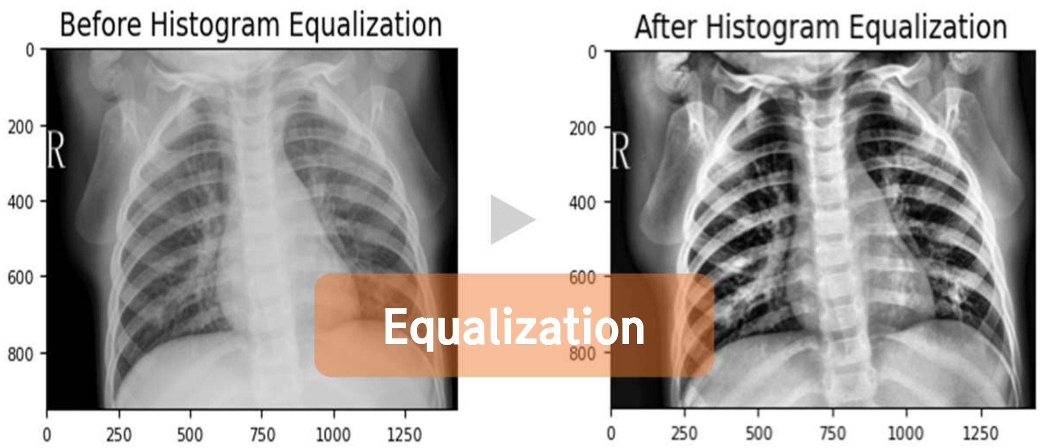


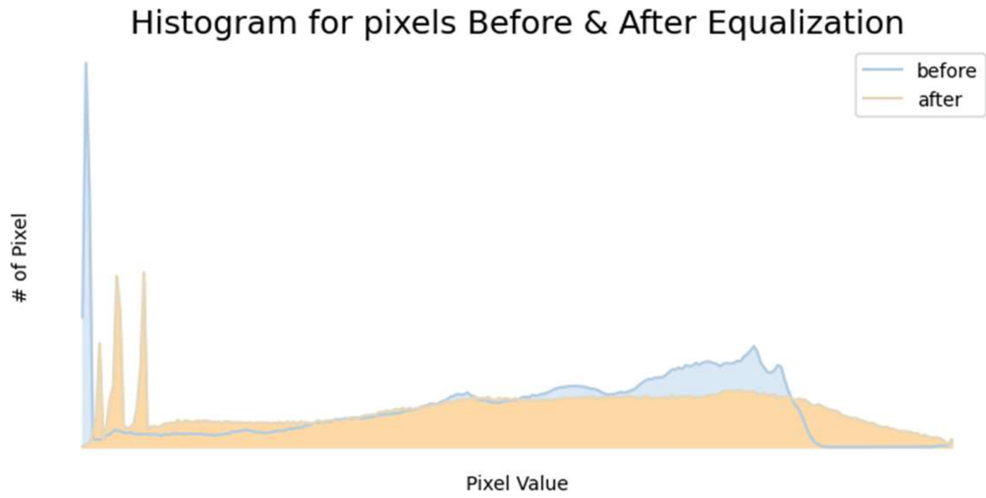


모든 intensity가 동일한 빈도수로 사용되도록 equalization

Data Preprocessing

Histogram Equalization





03. Modeling

Deep Learning Models

Models

총 9개의 **딥러닝 모델**을 사용하여 실험을 진행하였다.

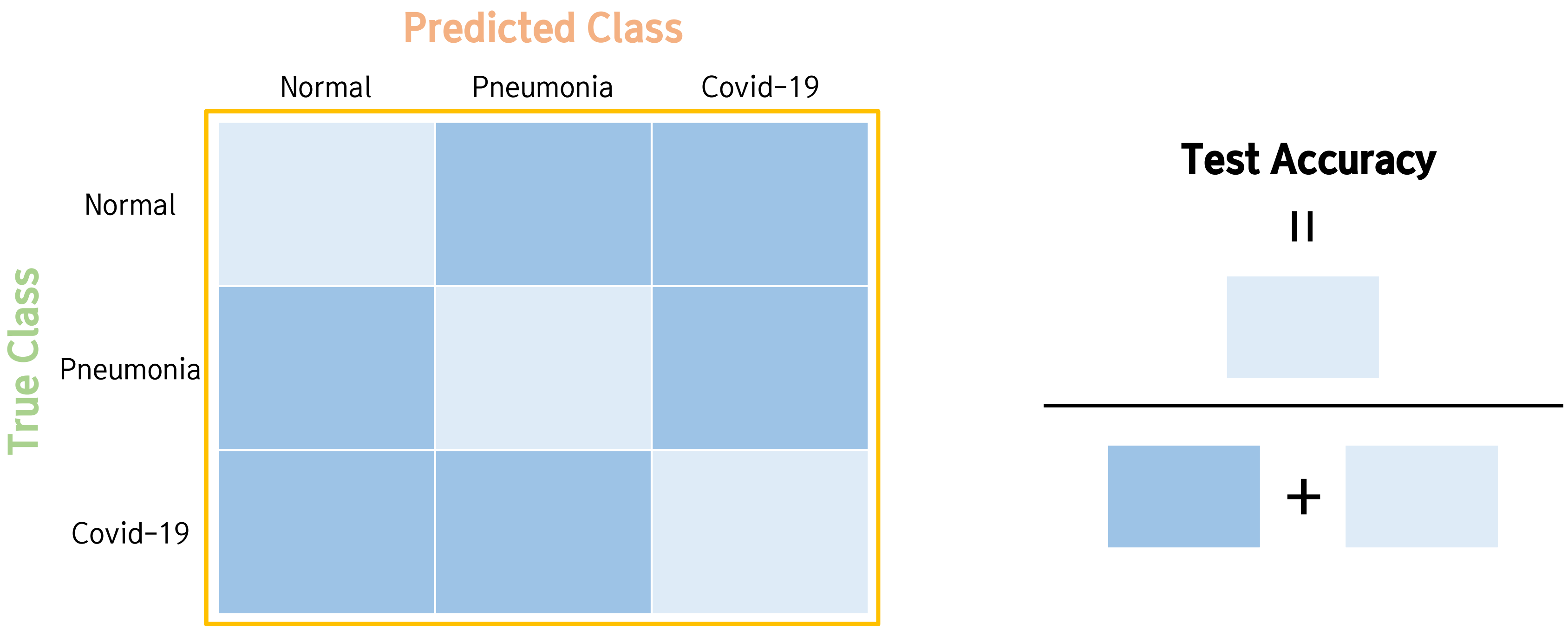
- 3 Conv. Layer
- AlexNet
- DenseNet
- ResNet34
- ResNet50
- VGG16
- VGG19
- Xception
- InceptionV3

Default Settings

- epochs = 100
- batch size = 32
- optimizer = SGD, RMSprop
(Learning Rate = 0.001)
- Validation Split = 0.2
- Early Stopping
(Validation loss, Patience=10)

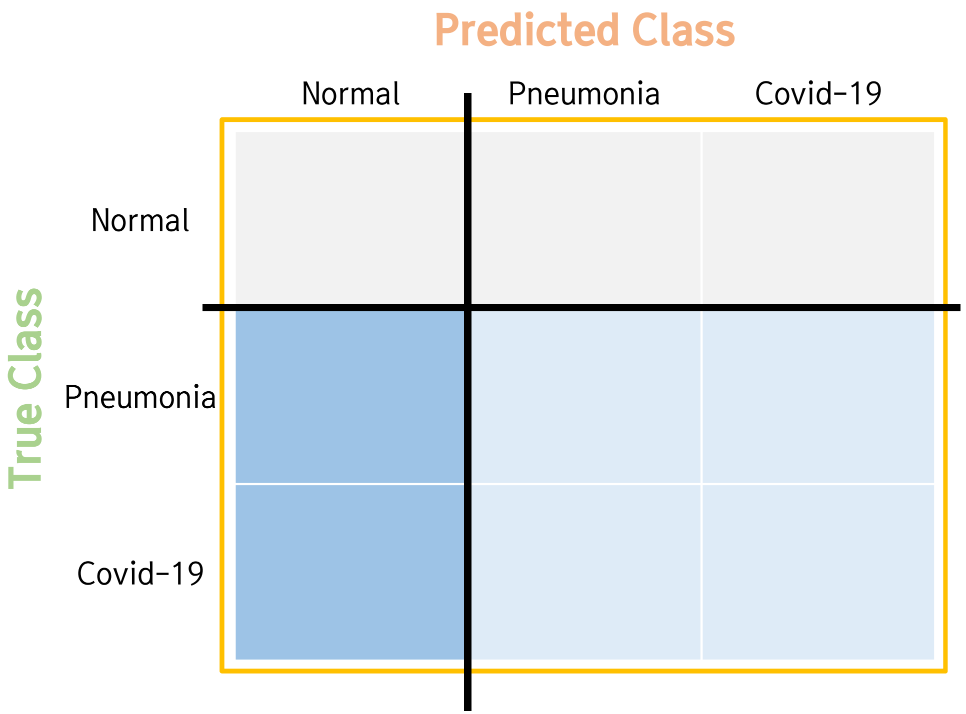
Confusion matrix

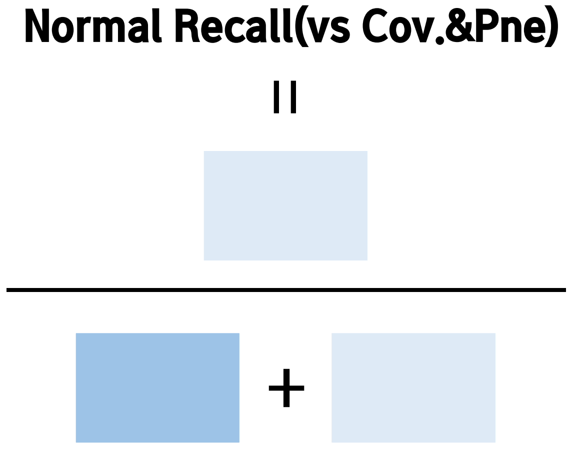
Confusion Matrix



Confusion matrix

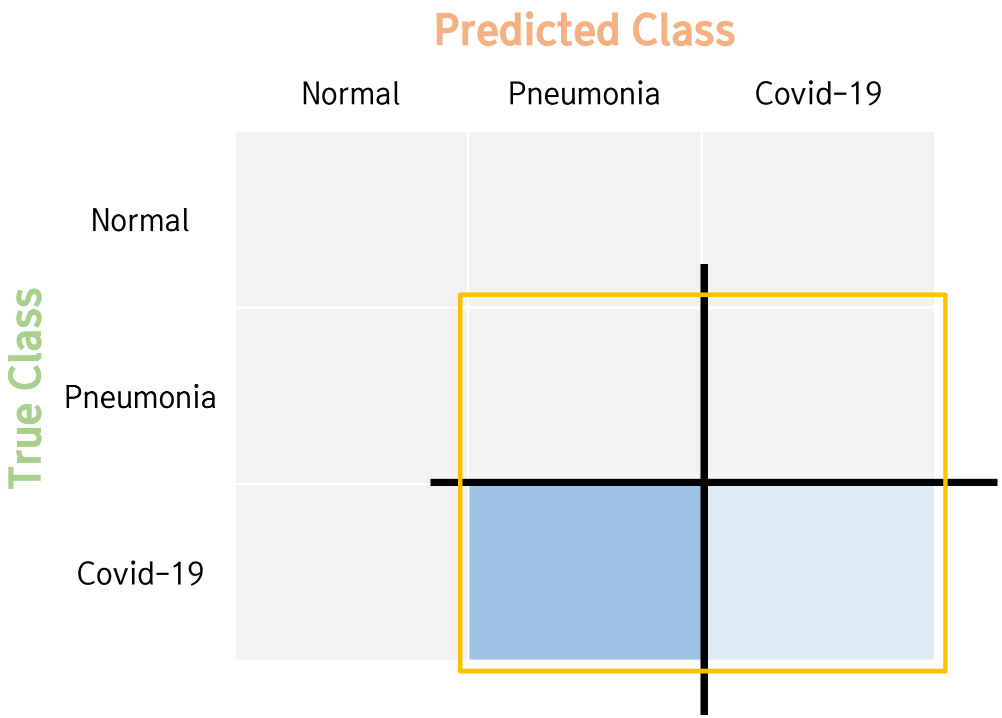
Confusion Matrix

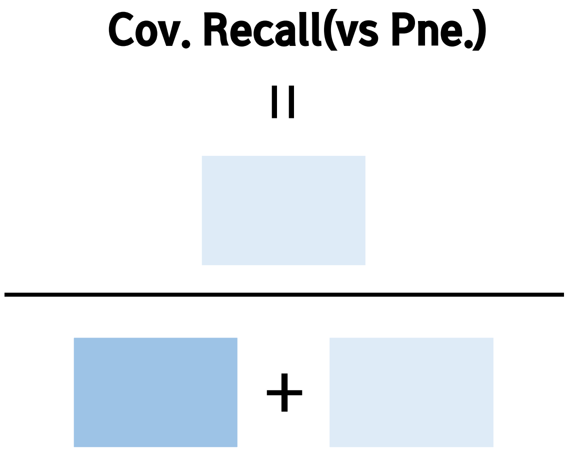




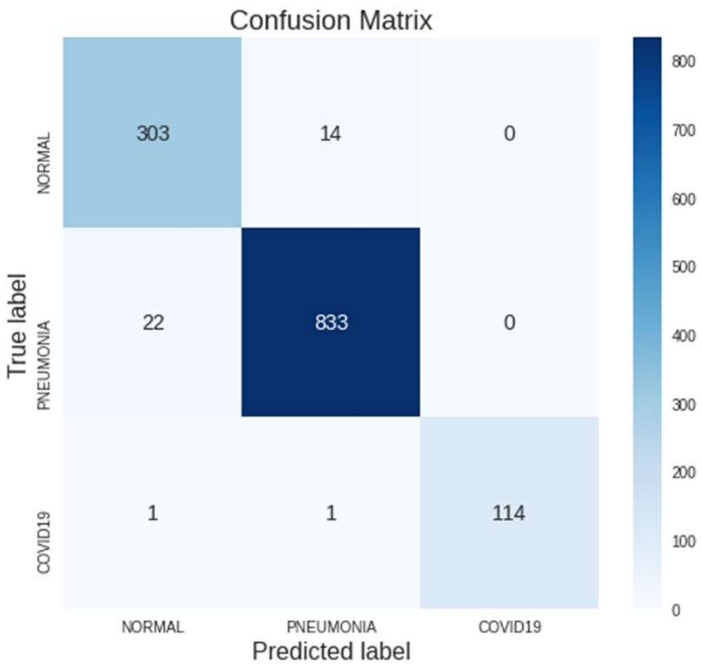
Confusion matrix

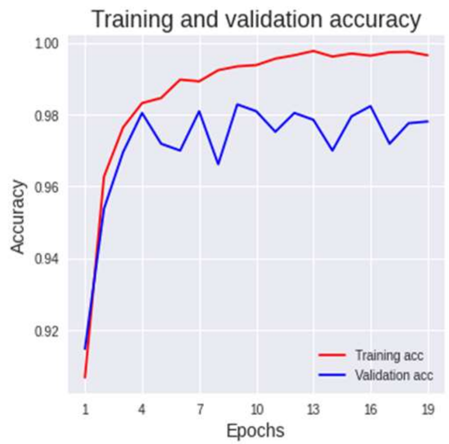
Confusion Matrix

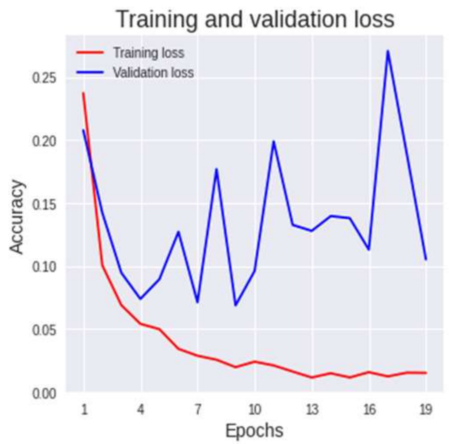


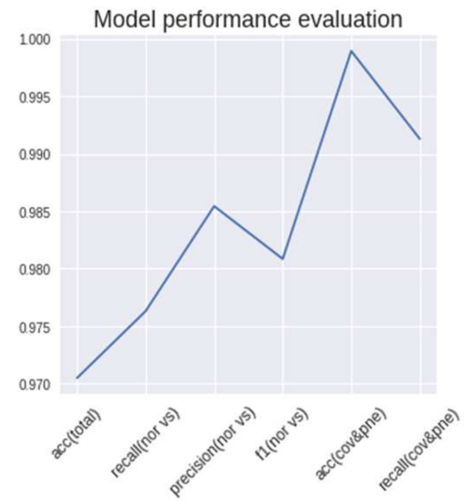


EX. VGG19


Confusion Matrix
True label
Predicted label
NORMAL
PNEUMONIA
COVID19
303
14
0
22
833
0
1
1
114
800
700
600
500
400
300
200
100
0


Training and validation accuracy
Accuracy
Epochs
Training acc
Validation acc
1.00
0.98
0.96
0.94
0.92
1
4
7
10
13
16
19


Training and validation loss
Accuracy
Epochs
Training loss
Validation loss
0.25
0.20
0.15
0.10
0.05
0.00
1
4
7
10
13
16
19


Model performance evaluation
1.000
0.995
0.990
0.985
0.980
0.975
0.970
acc(total)
recall(nor vs)
precision(nor vs)
f1(nor vs)
acc(cov&pne)
recall(cov&pne)

Summary

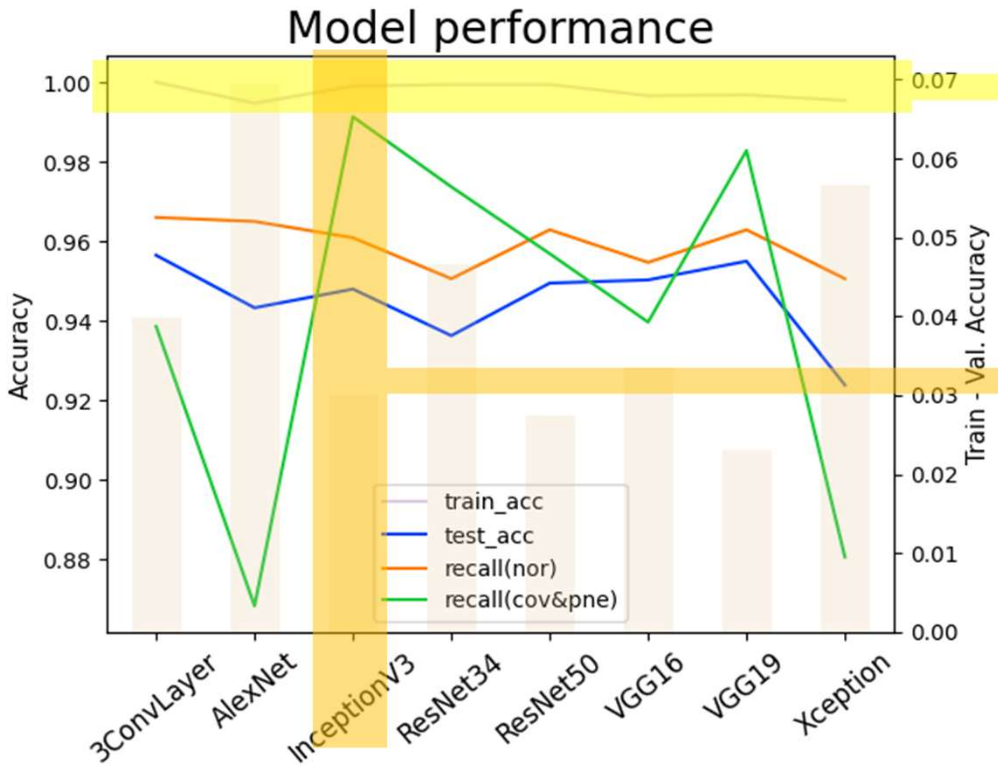
Value	Train Accuracy	Validation Accuracy	Test Accuracy	Recall (Normal)	Recall (COVID19 vs PNEUMONIA)
3 Conv. Layer	0.9980	0.9557	0.9371	0.9485	0.9181
VGG16	0.9964	0.9833	0.9681	0.9711	1.0000
VGG19	0.9966	0.9781	0.9704	0.9763	0.9913
ResNet34	0.9995	0.9757	0.9588	0.9742	0.9824
ResNet50	0.9983	0.9029	0.8812	0.9711	1.0000
AlexNet	0.9984	0.9638	0.9503	0.9629	0.9913
InceptionV3	0.9996	0.9714	0.9456	0.9577	0.9739
Xception	0.9982	0.9724	0.9526	0.9660	0.9652
DenseNet50	0.9982	0.9700	0.9417	0.9423	1.0000

Summary

Model performance graph

Summary

Model performance graph



Training accuracy ≈ 1

→ Overfitting의 위험성

InceptionV3

Ensemble

Soft voting

3 Conv. Layer
AlexNet
DenseNet50
ResNet34
ResNet50
VGG16
VGG19
Xception
InceptionV3

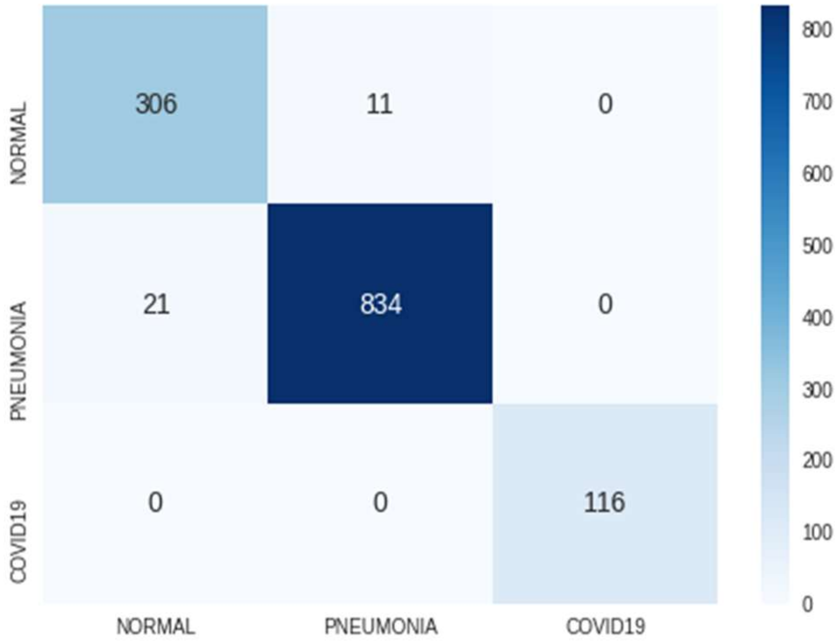
All possible Combinations

Soft Voting

Ensemble

Soft voting

3 Conv. Layer	AlexNet	DenseNet50
ResNet34	ResNet50	VGG16
VGG19	Xception	InceptionV3



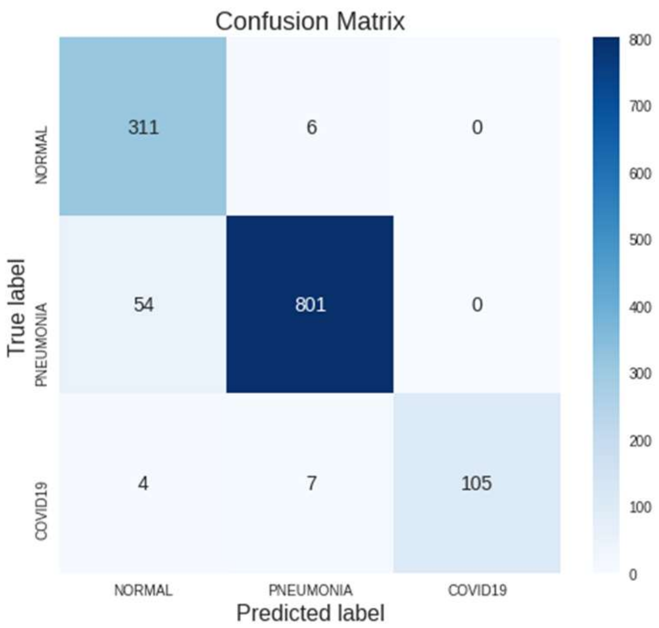
Test accuracy = 0.9756
Normal Recall(vs Cov.&Pne) = 0.9784
Cov. Recall(vs Pne.) = 1

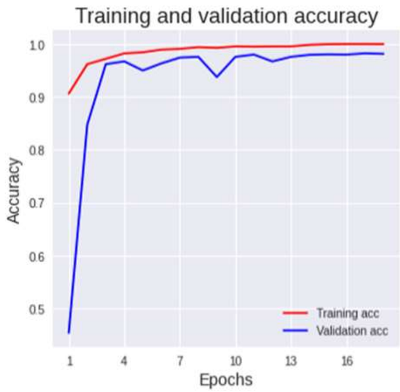
Fine tuning – InceptionV3

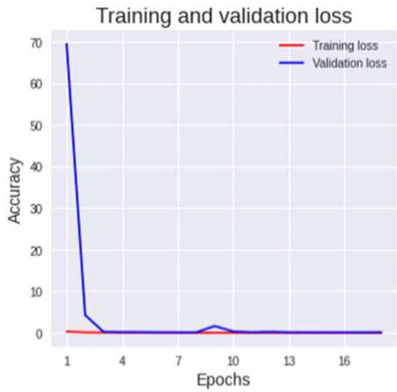
Model (RMSprop)	Freezing pre-trained model layer			Dense layer		Regularizer		
	0%(True)	50%	100%(False)	O	O + Dropout(0.5)	L1(0.1)	L2(0.1)	L1 + L2(0.1)
①	O	X	X	X	X	X	X	X
②	X	O	X	X	X	X	X	X
③	X	X	O	X	X	X	X	X
④	O	X	X	O	X	X	X	X
⑤	O	X	X	X	O	X	X	X
⑥	O	X	X	O	X	O	X	X
⑦	O	X	X	O	X	X	O	X
⑧	O	X	X	X	X	X	X	O
⑨	X	O	X	X	O	X	X	O

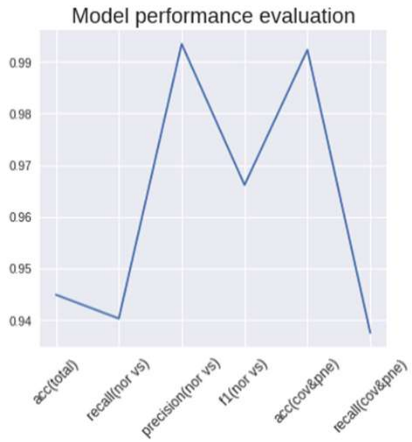
EX. Fine tuning

② Freezing pre-trained model layer: 50%







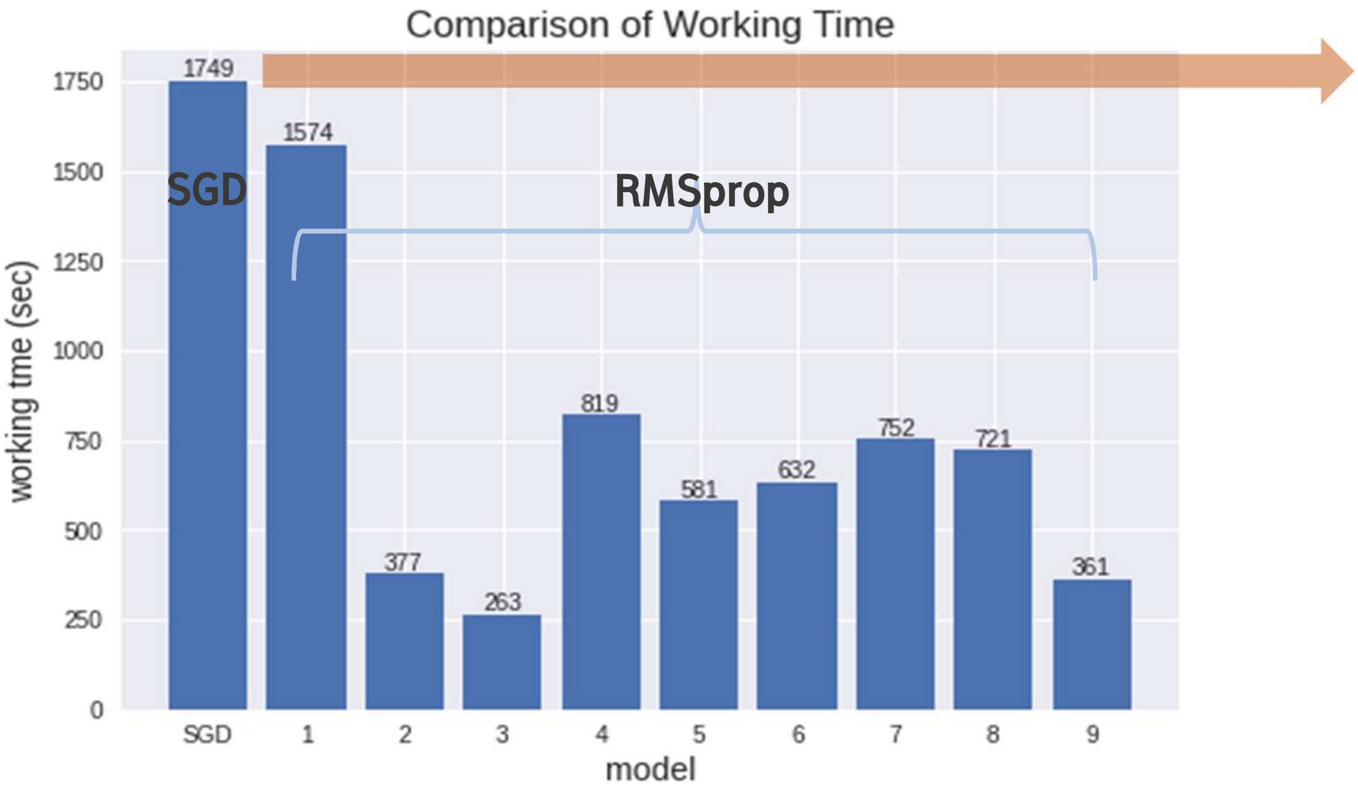


Summary

Model #	Train Accuracy	Validation Accuracy	Test Accuracy	Recall (nor. vs)	recall (cov&pne)	Train – Val Accuracy	time(s)
SGD	0.9990	0.9690	0.9480	0.9609	0.9913	0.0300	1749
①	1.0000	0.9876	0.9658	0.9763	0.9828	0.0124	1575
②	0.9999	0.9814	0.9449	0.9403	0.9375	0.0185	378
③	0.9833	0.9271	0.9177	0.9351	0.9123	0.0562	264
④	0.9989	0.9800	0.9612	0.9722	0.9914	0.0189	820
⑤	0.9998	0.9829	0.9589	0.9670	0.9913	0.0169	581
⑥	1.0000	0.9814	0.9464	0.9609	0.9914	0.0186	632
⑦	1.0000	0.9833	0.9682	0.9856	0.9914	0.0167	753
⑧	1.0000	0.9838	0.9674	0.9701	1.0000	0.0162	722
⑨	0.9371	0.9852	0.9635	0.9743	0.9828	0.0481	361

Summary

Model performance time



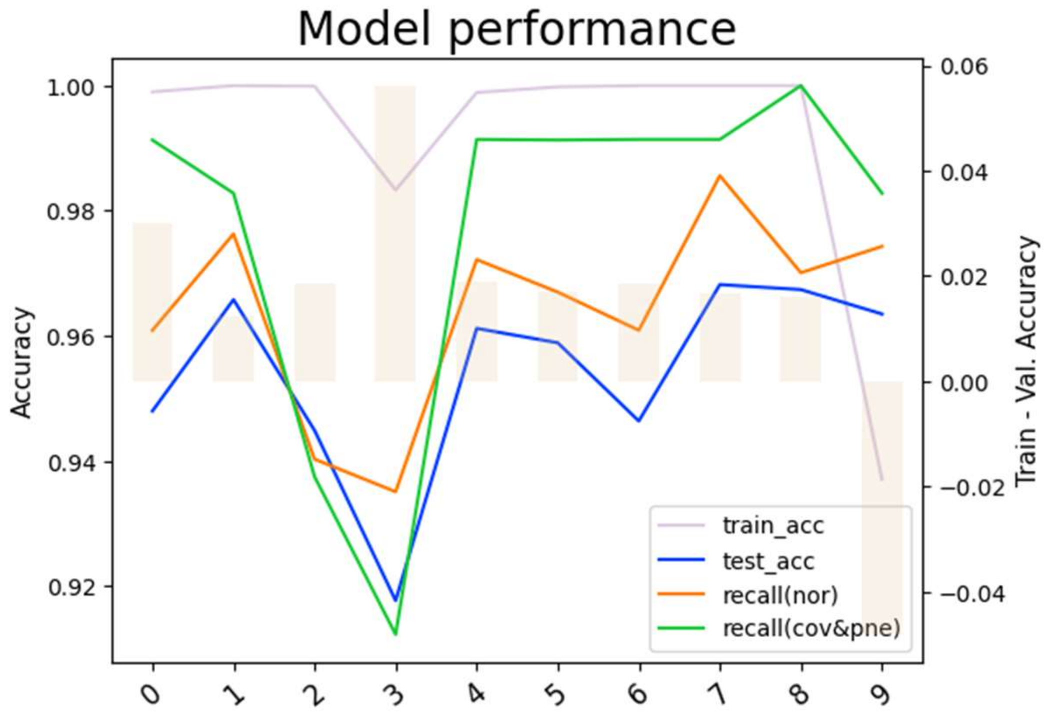
SGD vs RMSprop

- 고속 Optimizer
- learning rate 자동 조절

Working Time ↓

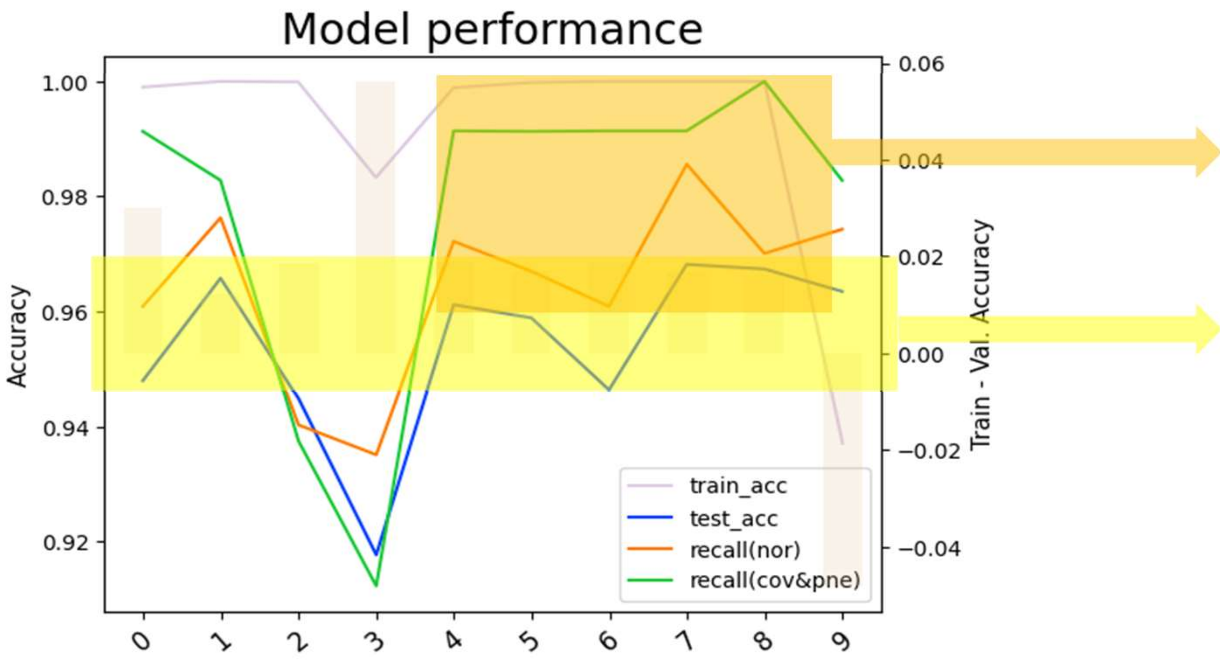
Summary

Model performance graph



Summary

Model performance graph



Test Acc. & Recall ↑

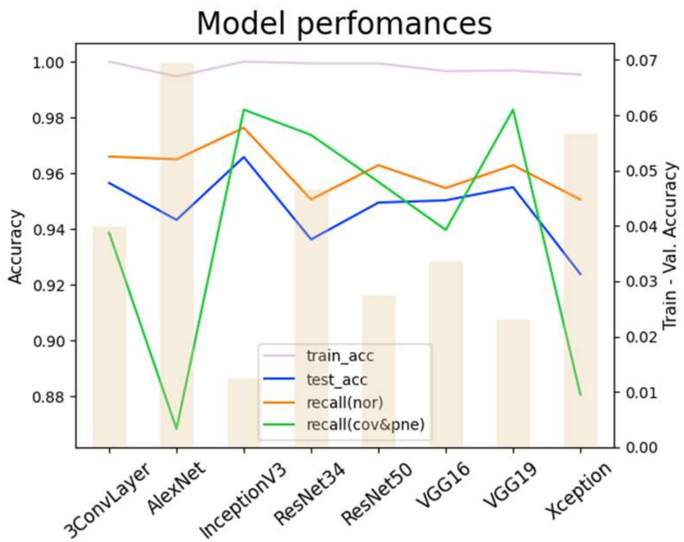
Training – Validation Acc. ↓

→ Overfitting ↓

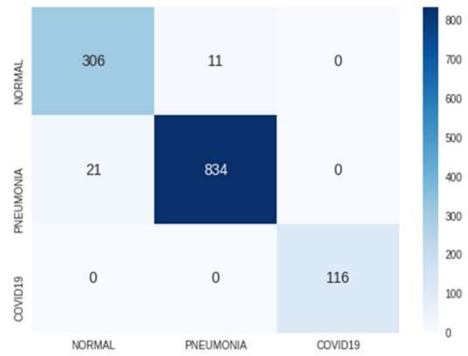
04. Results

Results

Summary



Ensemble (Soft voting)

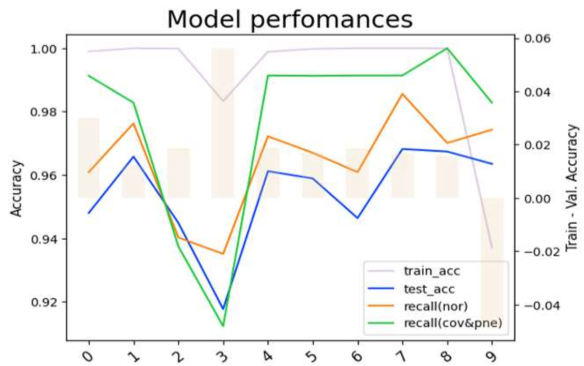


Test Acc. = 0.9756

Normal Recall(vs Cov.&Pne.) = 0.9784

Cov. Recall(vs Pne.) = 1

Fine tuning



Test Acc. = 0.9682

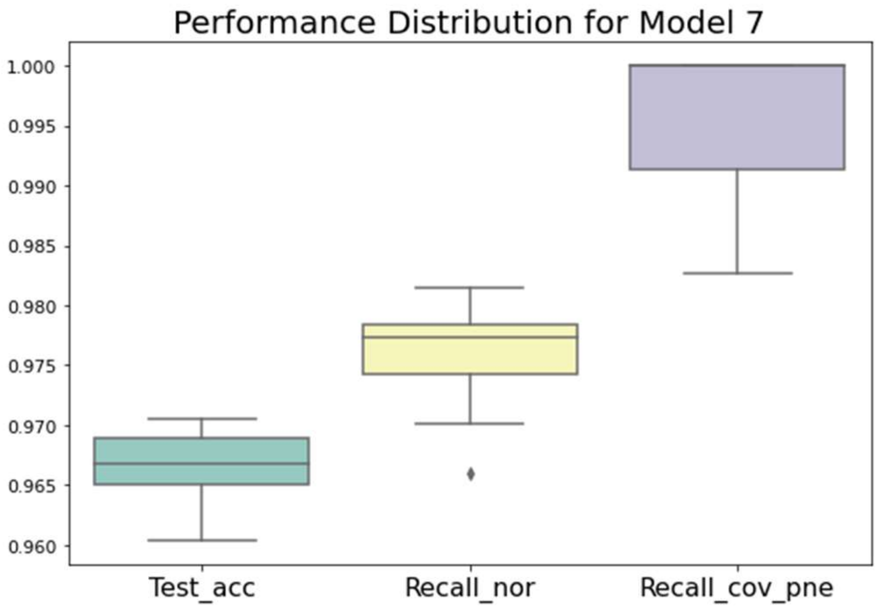
Normal Recall(vs Cov.&Pne.) = 0.9856

Cov. Recall(vs Pne.) = 0.9914

⑦ InceptionV3 + Regularizer(L2)

Performance Distribution

앞서 선정한 Model ⑦(InceptionV3 + Regularizer(L2))에 대하여 50번의 반복 시행을 통하여 모델의 Performance의 Distribution을 구해보았다.



Mean Value :

Test_acc	0.9668
Recall_nor	0.9764
Recall_cov_pne	0.9955

≈

Test Acc. = 0.9682

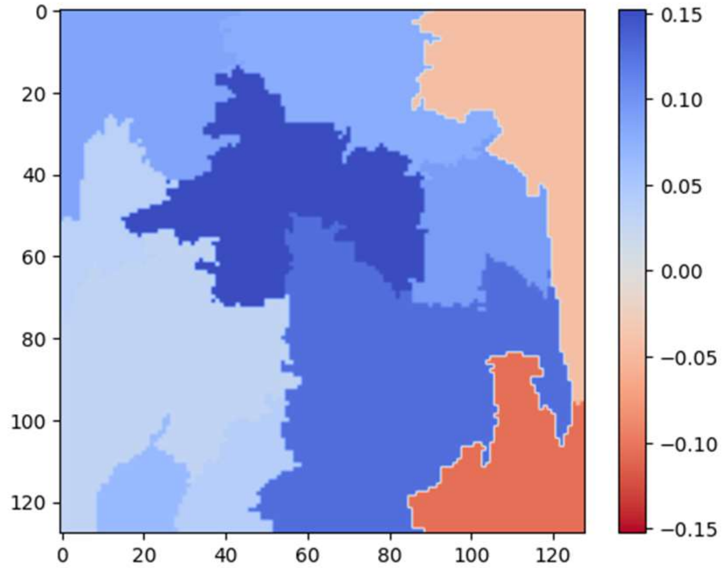
Normal Recall(vs Cov.&Pne.) = 0.9856

Cov. Recall(vs Pne.) = 0.9914

Performance Distribution의 평균값이 앞서 구한 값과 비슷하지만 다소 차이가 존재함

Visualization

Prediction Visualization by LIME



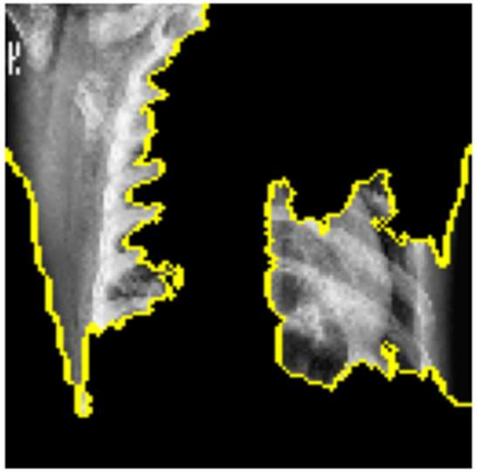
Predicted class의 Local Area에 따른 Prediction rate

Visualization

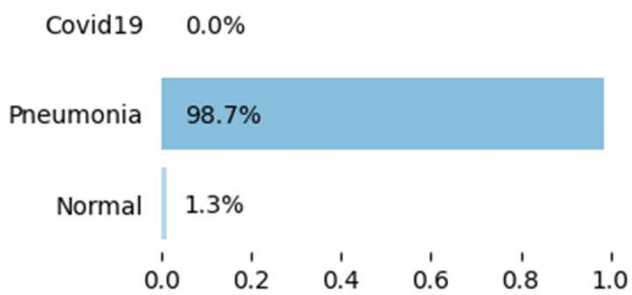
Prediction Visualization by LIME

Actual class : Pneuomnia

Visualization of detection



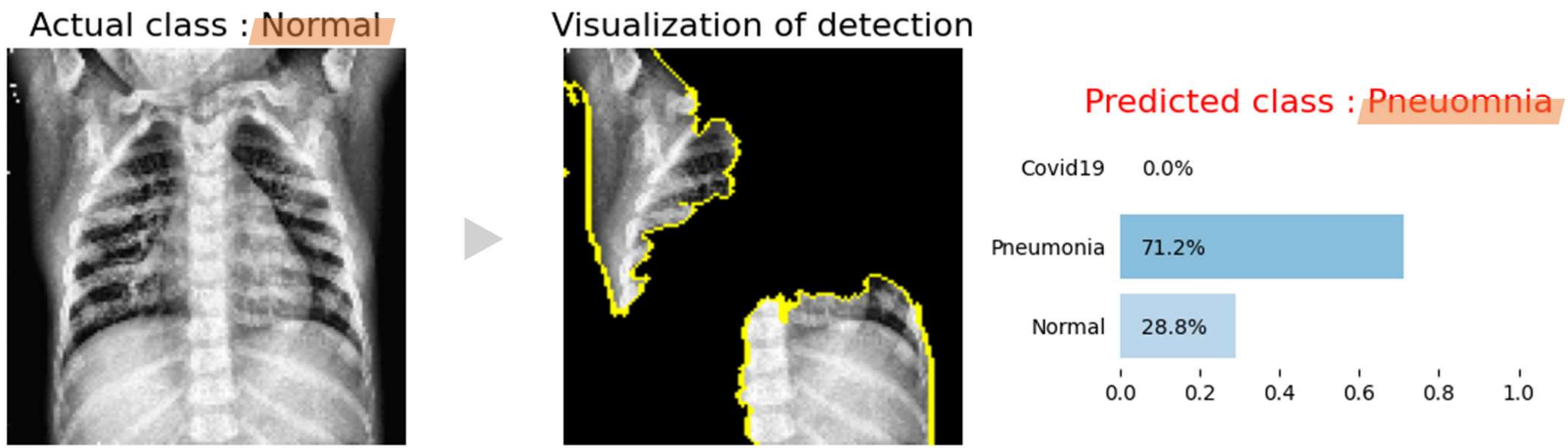
Predicted class : Pneuomnia



Predicted class의 예측값이 높은 Local area와 각 class에 대한 Prediction rate

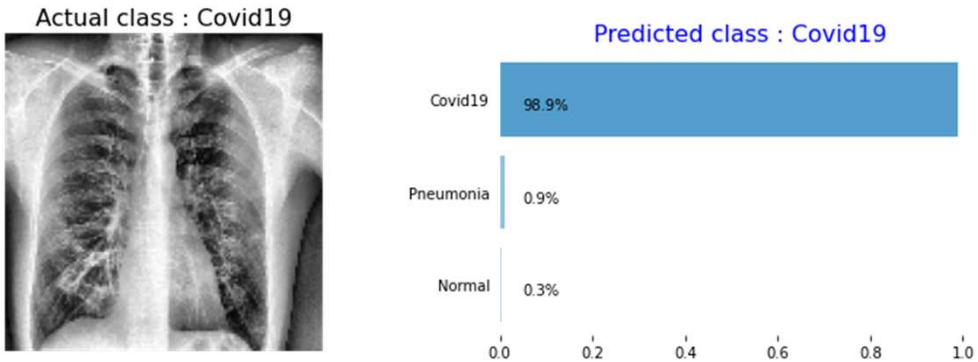
Visualization

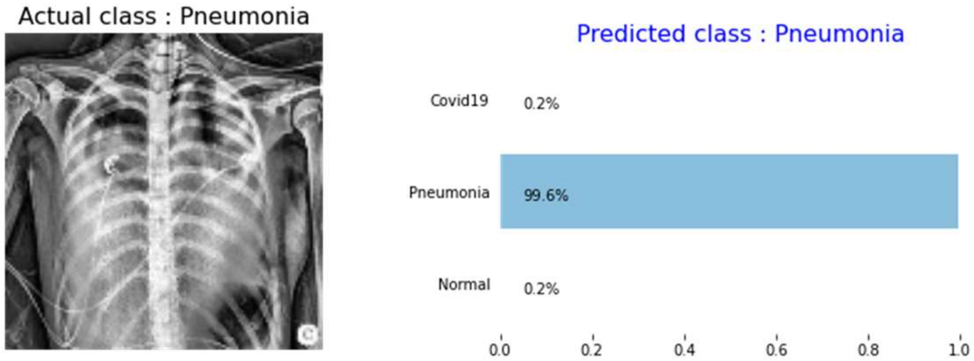
Prediction Visualization by LIME

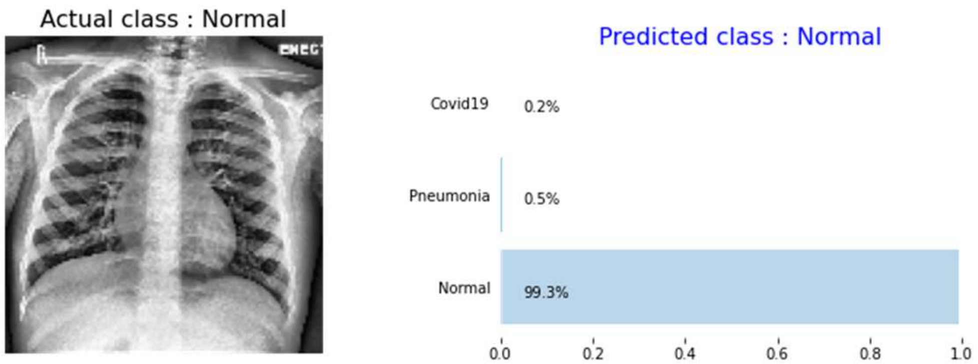


Model Test

구글에서 Random하게 구한 x-ray 이미지(Normal / Pneumonia / Covid-19) 총 9개에 대하여 분류 성능을 테스트 해보았다.







9개 이미지 중 8개에 대해 예측 성공!!

Limit

Limit

- **한정된 Computing Power**

  Measure에 대한 보다 정교한 값을 구하기 위해서는 Distribution을 구하는 것이 좋다.

  그러나 제한적인 메모리와 GPU로 인해 비교적인 적은 횟수로 딥러닝 모델을 Fitting해야 했다.

  이러한 이유로 하이퍼파라미터 튜닝에 어려움이 있었다.

- **충분하지 않은 Dataset**

  실험을 진행한 Dataset의 Data 숫자는 약 5,000개 였다.

  따라서 모델을 Fitting할 때마다 Measure에 대한 편차가 컸다.

Conclusion

X-ray를 활용한 코로나 진단법의 장점

1) 타인과 대면할 필요가 없어 위험이 감소되어 안전성이 증가한다.
2) 검사를 위한 시간과 비용을 절약할 수 있다.
3) 일회용 사용을 줄여 환경오염을 줄일 수 있다.
4) 폐렴과 쉽게 구분된다.
5) 한정된 의료 자원을 우선순위가 높은 환자에게 배분할 수 있다.

CNN을 활용한 이미지 분류 검사법의 의의

1) CT, MRI 등 다른 영상의학적 촬영 장비에 개발한 모형을 적용하여 의료 혁신을 이룰 수 있다.
2) 데이터가 축적될수록 정확도가 빠르게 개선될 것으로 기대된다.
3) 적은 비용으로도 일정 수준의 진료를 기대할 수 있다.

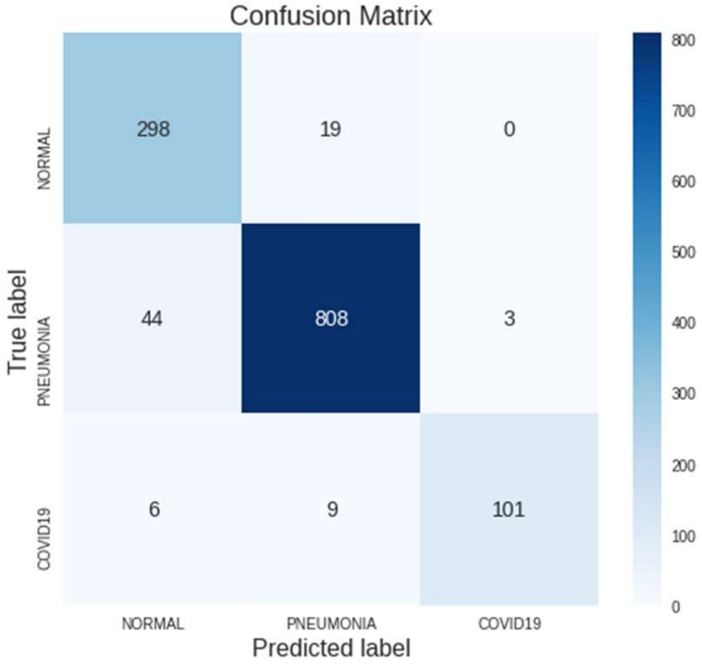
X-ray 이미지 분류 검사법을 사용하면 코로나19 판별의 효율성이 개선되고 정확도 높은 코로나 판별이 가능하다. 또한 CNN을 활용한 검사법은 다양한 질병의 검사 방식으로 확장 가능하다.

감사합니다

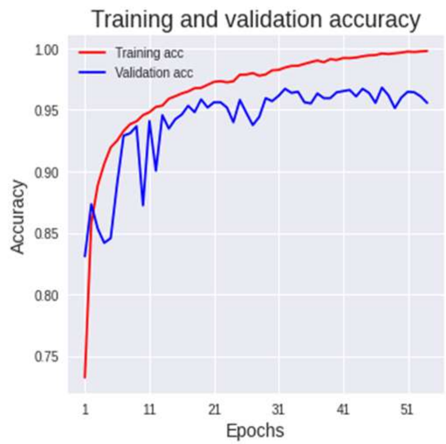
(부록) 예측 성능 도표

모델별 예측 성능 도표

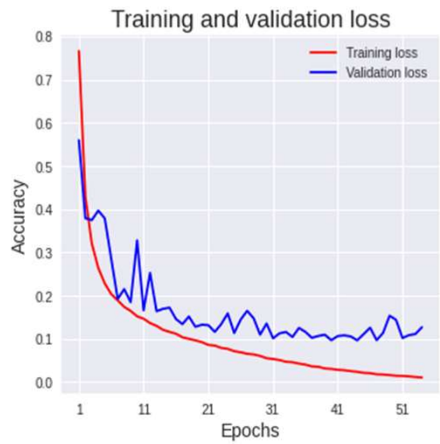
3 Conv. Layer



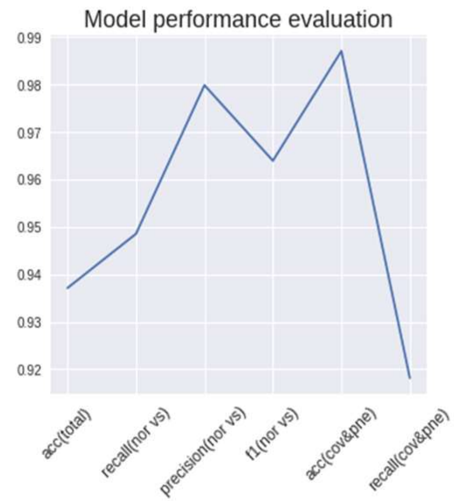
Confusion Matrix
True label
NORMAL
PNEUMONIA
COVID19
298
19
0
44
808
3
6
9
101
NORMAL
PNEUMONIA
COVID19
Predicted label
800
700
600
500
400
300
200
100
0



Training and validation accuracy
Training acc
Validation acc
Accuracy
Epochs

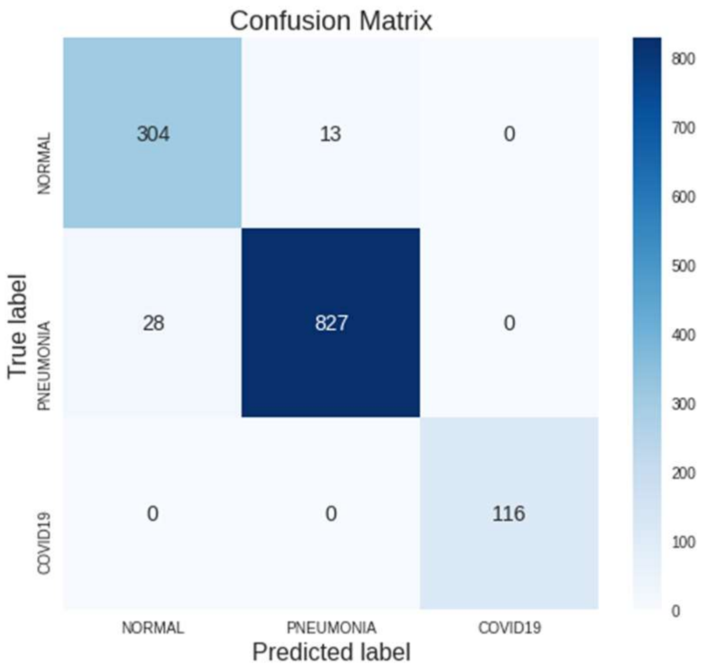


Training and validation loss
Training loss
Validation loss
Accuracy
Epochs

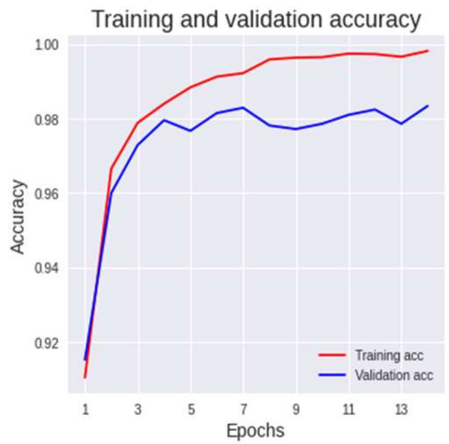


Model performance evaluation
acc(total)
recall(nor vs)
precision(nor vs)
f1(nor vs)
acc(cov&pne)
recall(cov&pne)

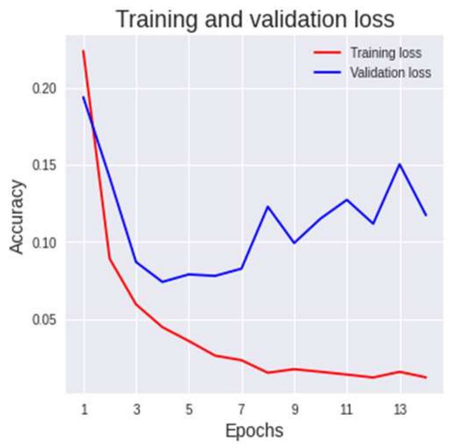
VGG16



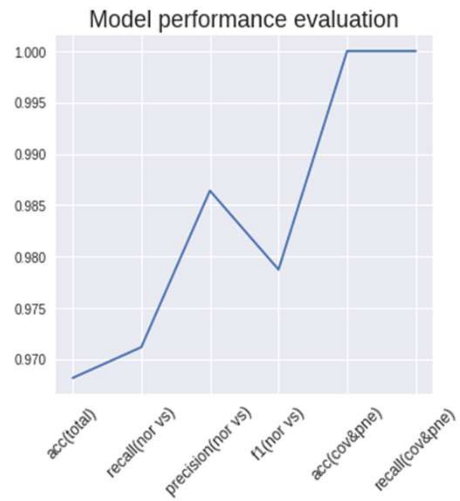
Confusion Matrix
NORMAL
PNEUMONIA
COVID19
True label
304
13
0
28
827
0
0
0
116
NORMAL
PNEUMONIA
COVID19
Predicted label
800
700
600
500
400
300
200
100
0



Training and validation accuracy
Accuracy
1.00
0.98
0.96
0.94
0.92
1
3
5
7
9
11
13
Epochs
Training acc
Validation acc

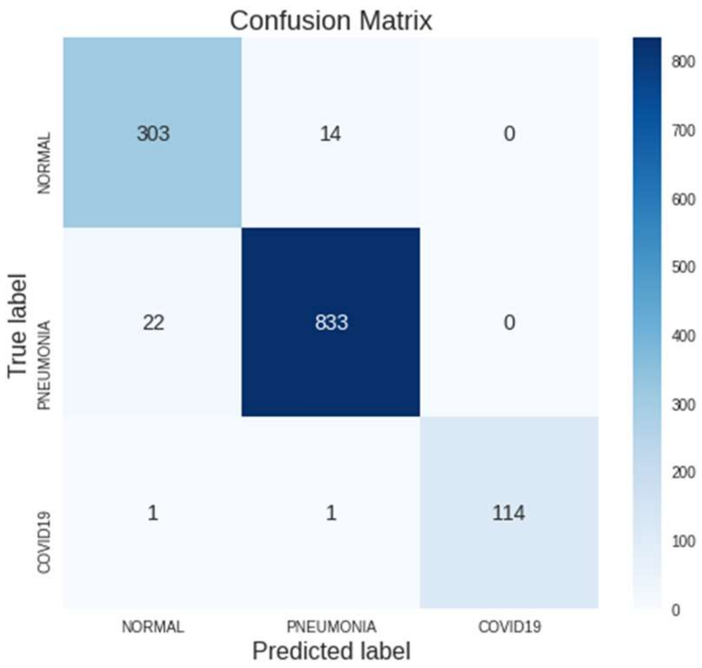


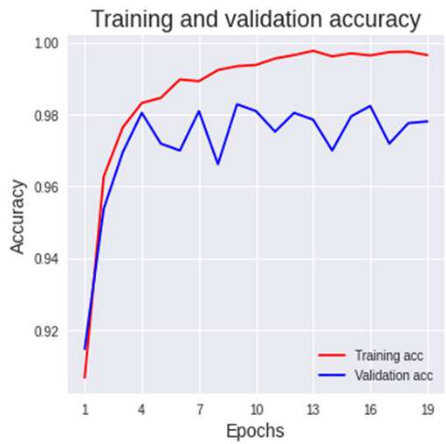
Training and validation loss
Accuracy
0.20
0.15
0.10
0.05
1
3
5
7
9
11
13
Epochs
Training loss
Validation loss

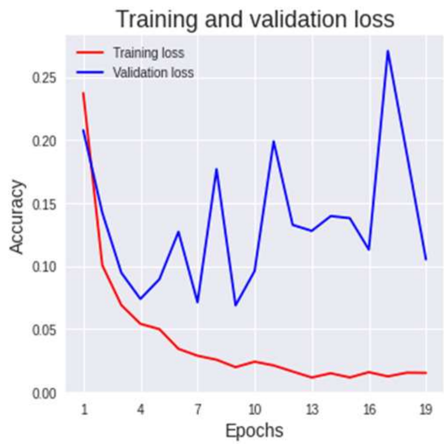


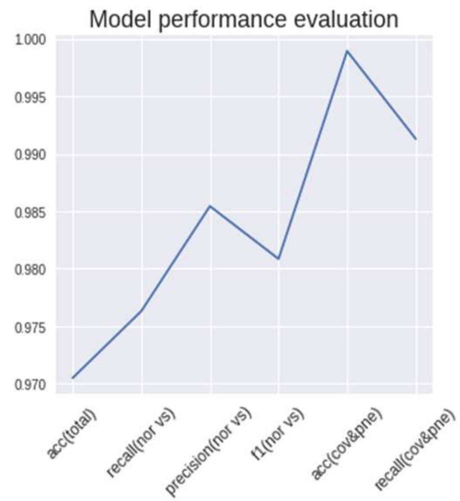
Model performance evaluation
1.000
0.995
0.990
0.985
0.980
0.975
0.970
acc(total)
recall(nor vs)
precision(nor vs)
f1(nor vs)
acc(cov&pne)
recall(cov&pne)

VGG19

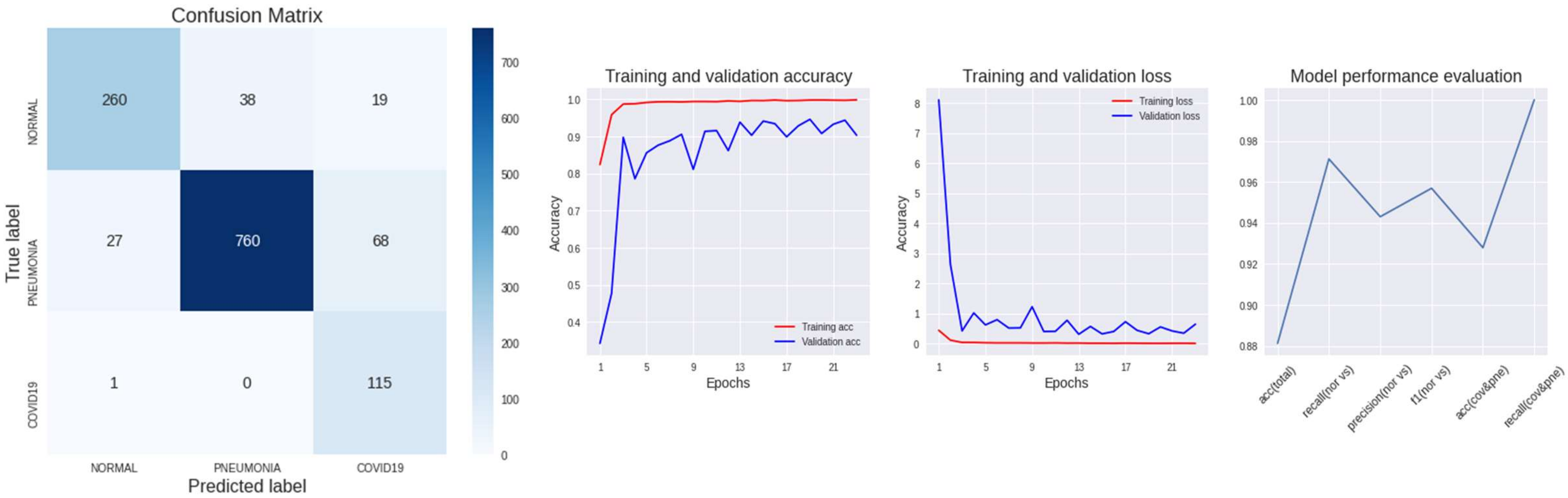

Confusion Matrix
True label
NORMAL
PNEUMONIA
COVID19
303
14
0
22
833
0
1
1
114
Predicted label
800
700
600
500
400
300
200
100
0


Training and validation accuracy
Accuracy
Epochs
Training acc
Validation acc
1.00
0.98
0.96
0.94
0.92
1
4
7
10
13
16
19


Training and validation loss
Accuracy
Epochs
Training loss
Validation loss
0.25
0.20
0.15
0.10
0.05
0.00
1
4
7
10
13
16
19

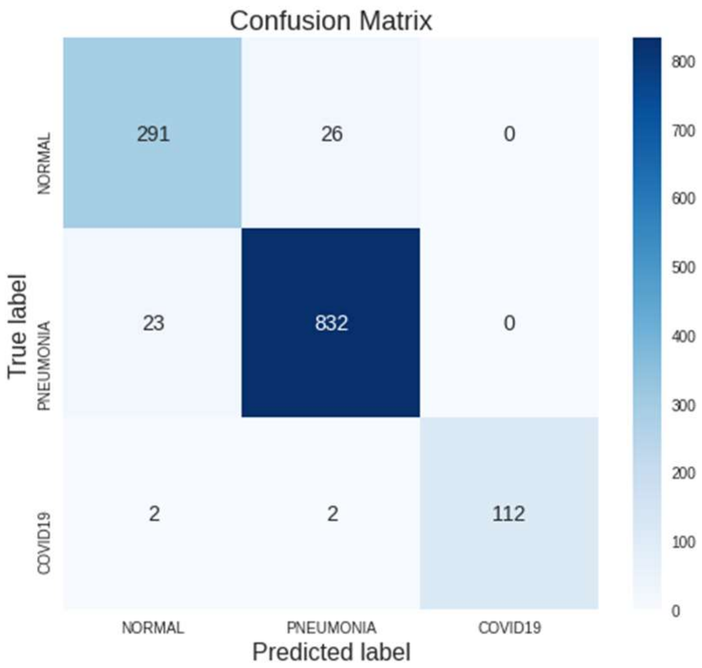

Model performance evaluation
1.000
0.995
0.990
0.985
0.980
0.975
0.970
acc(total)
recall(nor vs)
precision(nor vs)
f1(nor vs)
acc(cov&pne)
recall(cov&pne)

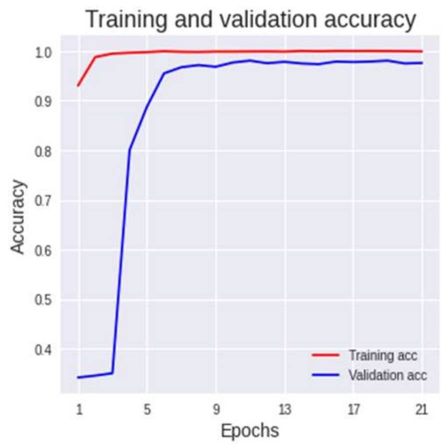
ResNet34

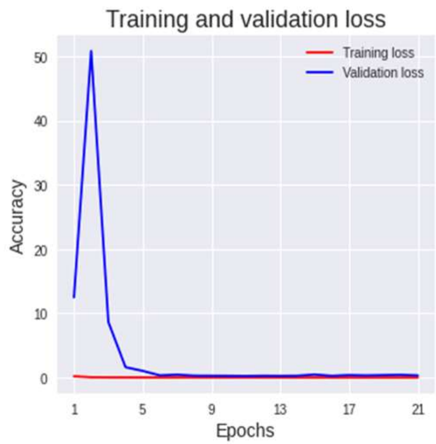


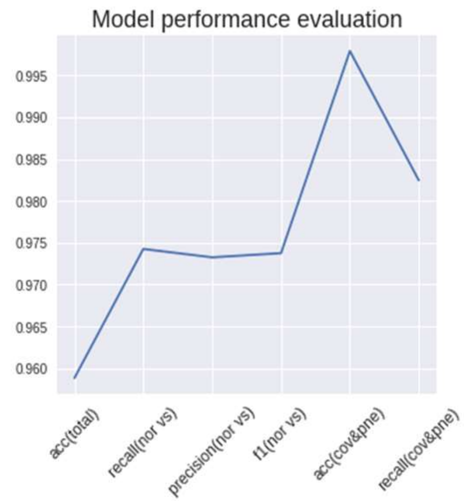
Confusion Matrix
True label
NORMAL
PNEUMONIA
COVID19
260
38
19
27
760
68
1
0
115
Predicted label
700
600
500
400
300
200
100
0
Training and validation accuracy
Accuracy
Epochs
Training acc
Validation acc
Training and validation loss
Training loss
Validation loss
Model performance evaluation
acc(total)
recall(nor vs)
precision(nor vs)
f1(nor vs)
acc(cov&pne)
recall(cov&pne)

ResNet50

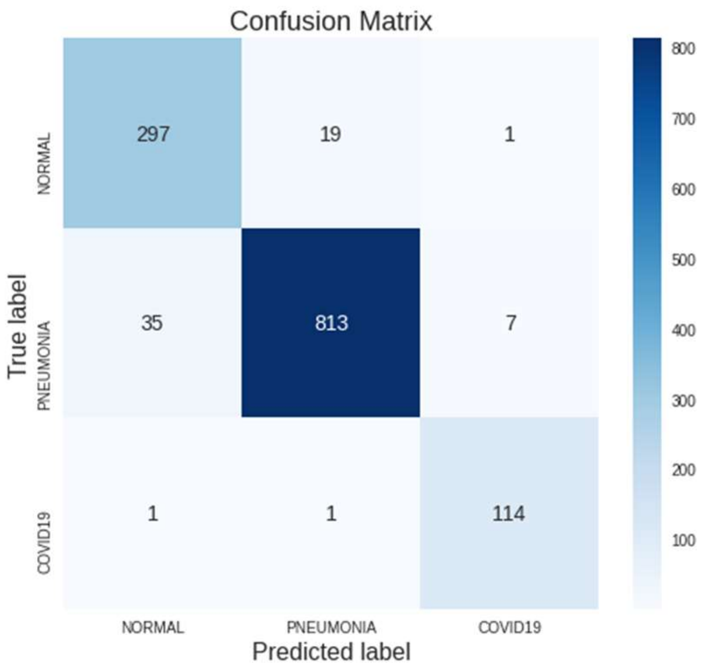

Confusion Matrix
291
26
0
23
832
0
2
2
112
NORMAL
PNEUMONIA
COVID19
True label
NORMAL
PNEUMONIA
COVID19
Predicted label
800
700
600
500
400
300
200
100
0


Training and validation accuracy
Accuracy
Epochs
Training acc
Validation acc
1.0
0.9
0.8
0.7
0.6
0.5
0.4
1
5
9
13
17
21

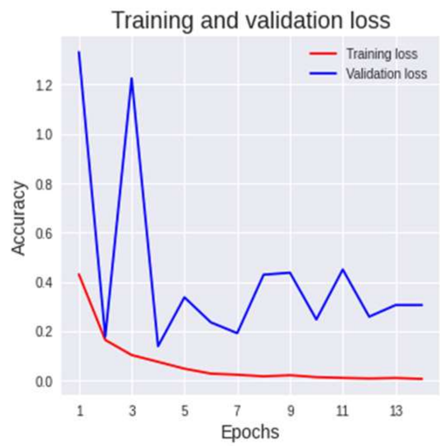

Training and validation loss
Accuracy
Epochs
Training loss
Validation loss
50
40
30
20
10
0
1
5
9
13
17
21

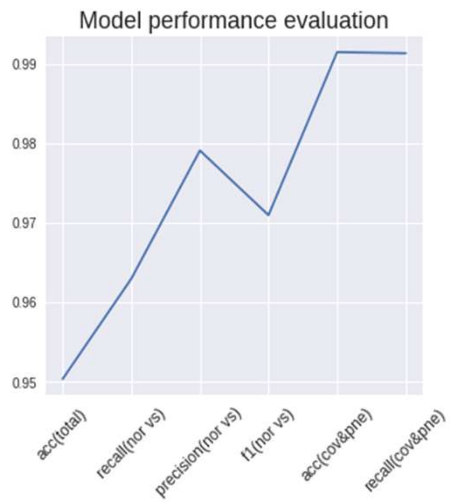

Model performance evaluation
0.995
0.990
0.985
0.980
0.975
0.970
0.965
0.960
acc(total)
recall(nor vs)
precision(nor vs)
f1(nor vs)
acc(cov&pne)
recall(cov&pne)

AlexNet

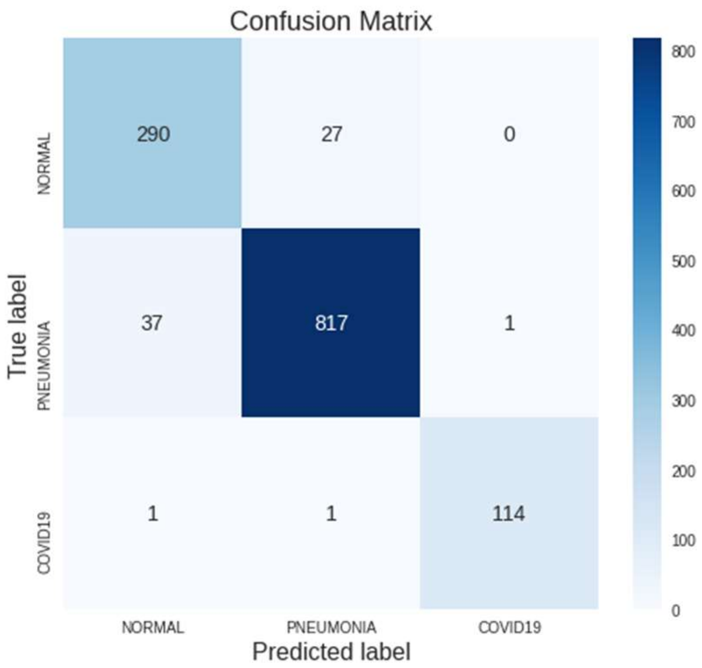

Confusion Matrix
NORMAL
PNEUMONIA
COVID19
True label
297
19
1
35
813
7
1
1
114
NORMAL
PNEUMONIA
COVID19
Predicted label
800
700
600
500
400
300
200
100

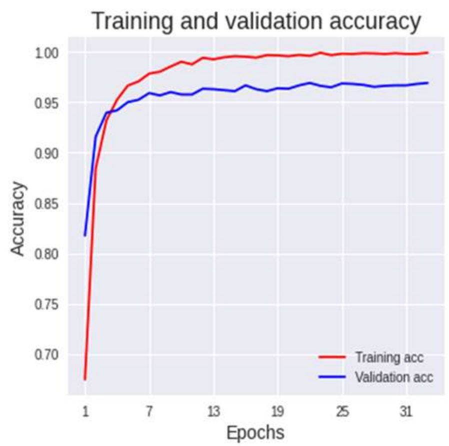

Training and validation accuracy
Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
1
3
5
7
9
11
13
Epochs
Training acc
Validation acc

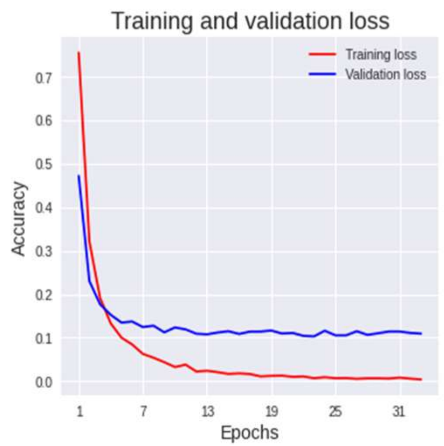

Training and validation loss
Accuracy
1.2
1.0
0.8
0.6
0.4
0.2
0.0
1
3
5
7
9
11
13
Epochs
Training loss
Validation loss

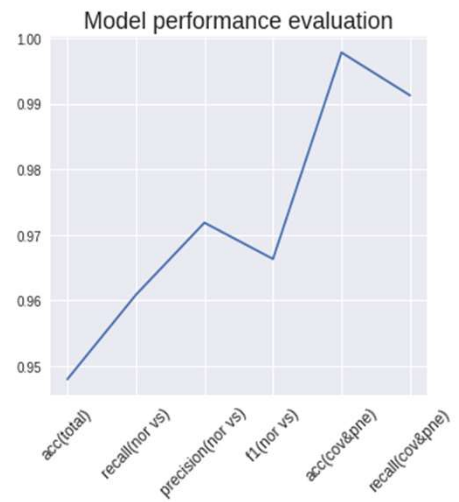

Model performance evaluation
0.99
0.98
0.97
0.96
0.95
acc(total)
recall(nor vs)
precision(nor vs)
f1(nor vs)
acc(cov&pne)
recall(cov&pne)

InceptionV3

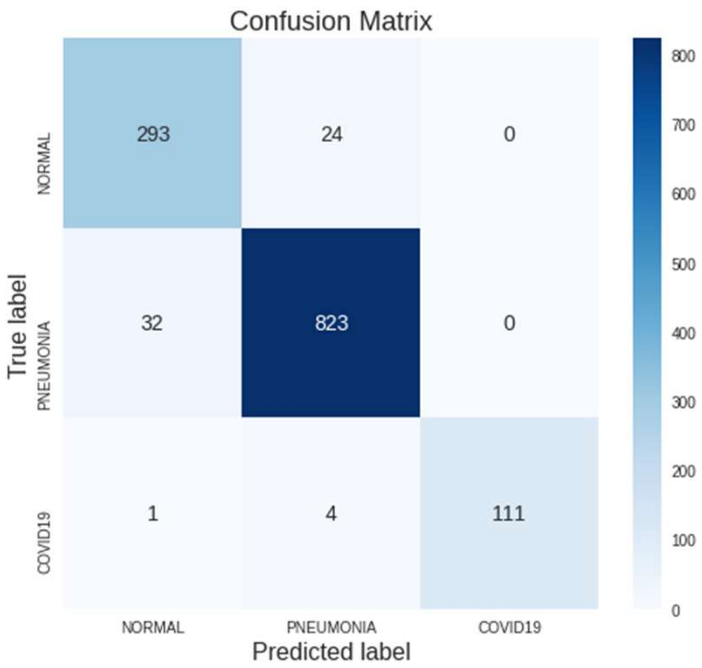


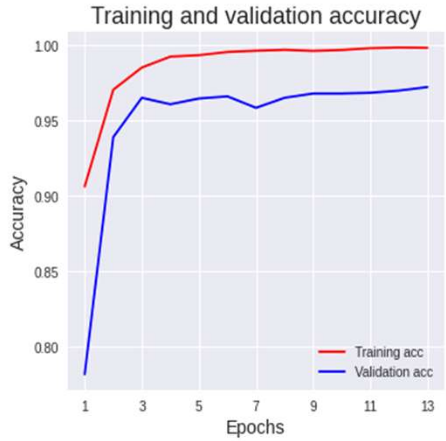


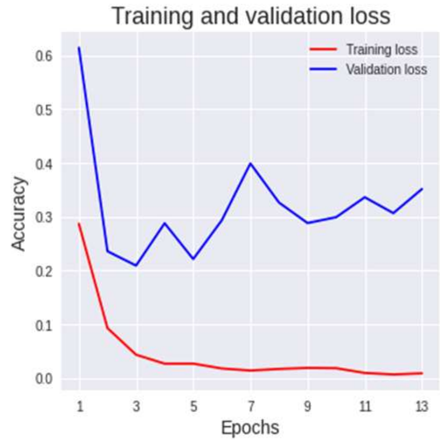


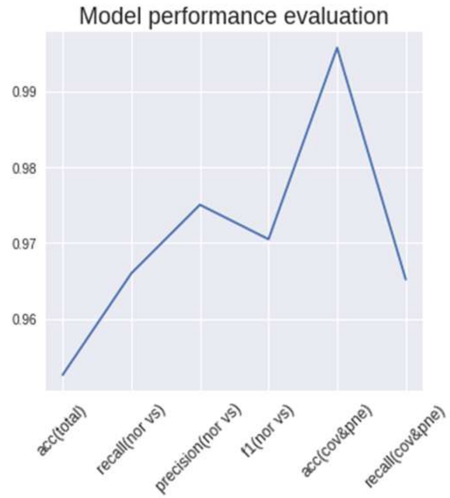


Xception

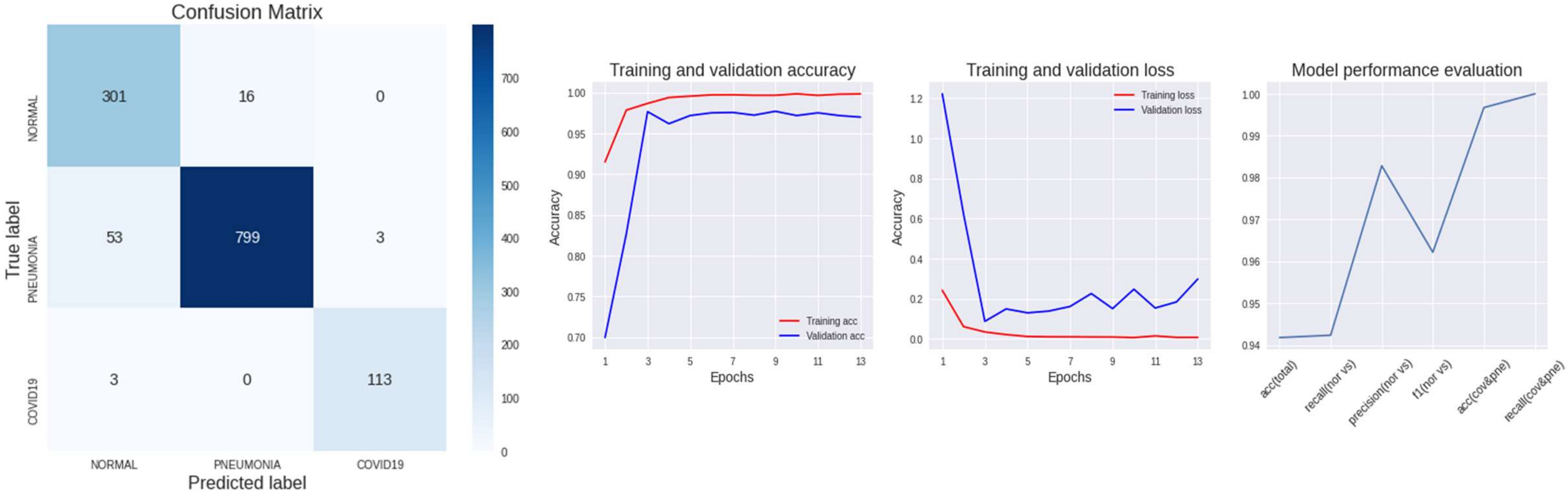

Confusion Matrix
True label
NORMAL
PNEUMONIA
COVID19
293
24
0
32
823
0
1
4
111
NORMAL
PNEUMONIA
COVID19
Predicted label
800
700
600
500
400
300
200
100
0


Training and validation accuracy
Accuracy
1.00
0.95
0.90
0.85
0.80
Training acc
Validation acc
1
3
5
7
9
11
13
Epochs


Training and validation loss
Accuracy
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Training loss
Validation loss
1
3
5
7
9
11
13
Epochs


Model performance evaluation
0.99
0.98
0.97
0.96
acc(total)
recall(nor vs)
precision(nor vs)
f1(nor vs)
acc(cov&pne)
recall(cov&pne)

DenseNet50

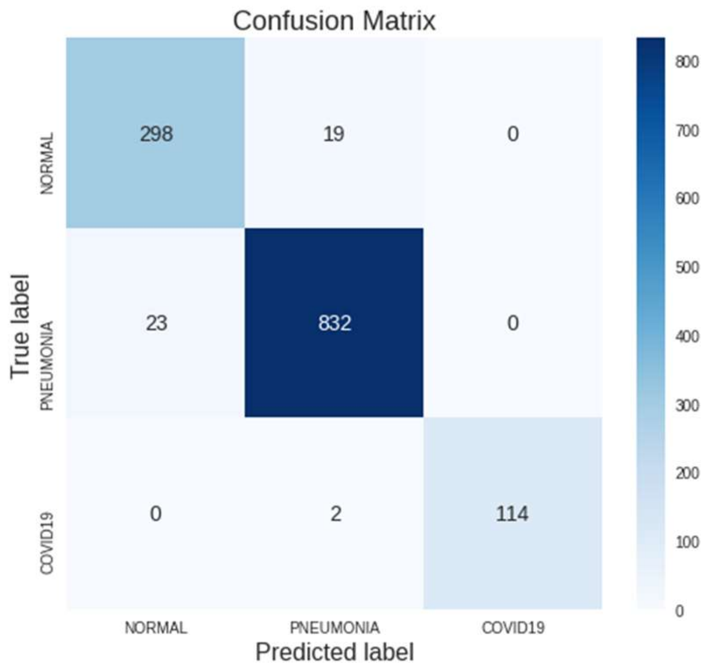


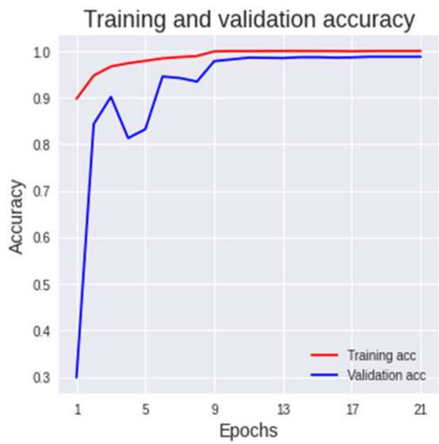
Confusion Matrix
True label
Predicted label
NORMAL
PNEUMONIA
COVID19
301
16
0
53
799
3
3
0
113
Training and validation accuracy
Accuracy
Epochs
Training acc
Validation acc
Training and validation loss
Accuracy
Epochs
Training loss
Validation loss
Model performance evaluation
acc(total)
recall(nor vs)
precision(nor vs)
f1(nor vs)
acc(cov&pne)
recall(cov&pne)

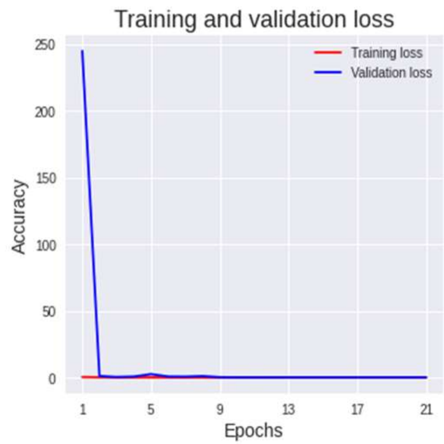
InceptionV3 – Fine tuning 예측 성능 도표

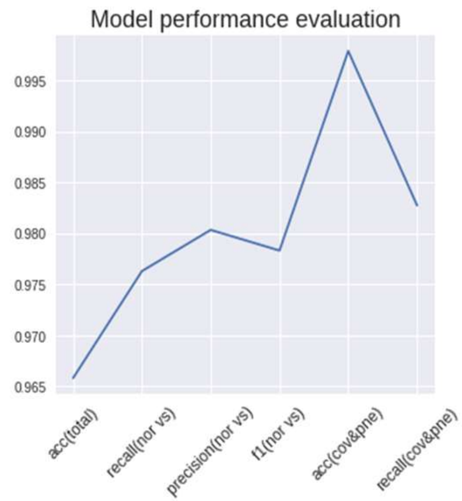
Fine tuning - Inception

① Freezing pre-trained model layer: 0%(True)



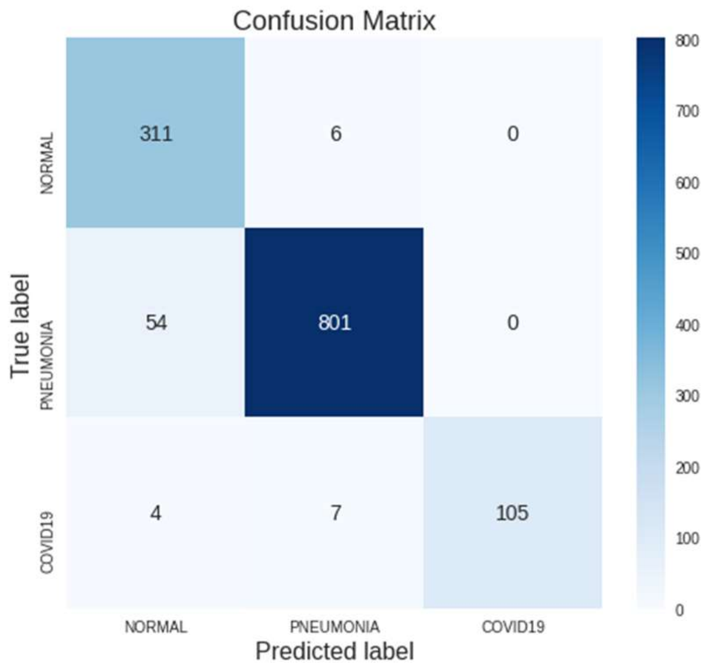


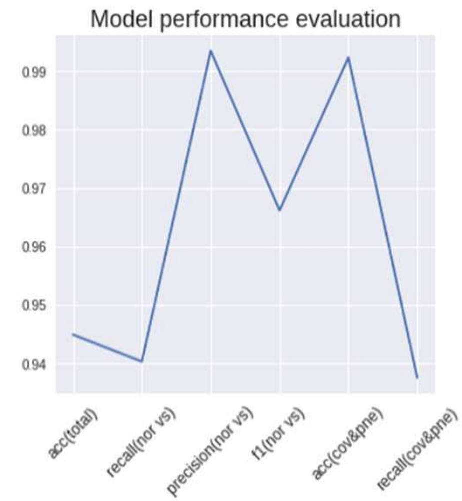




Fine tuning - Inception

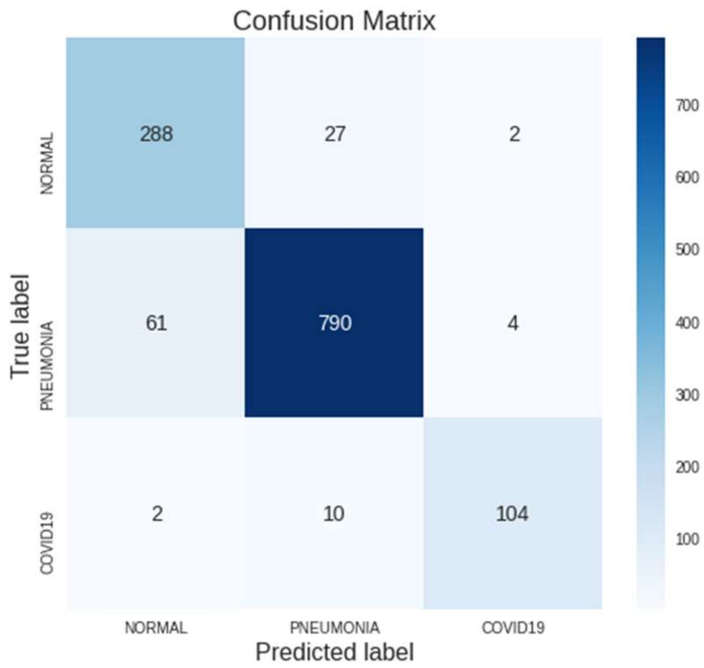
② **Freezing pre-trained model layer: 50%**

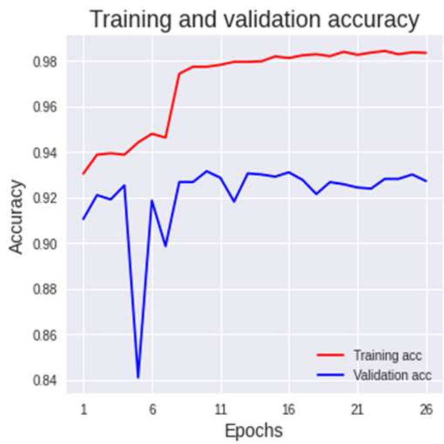


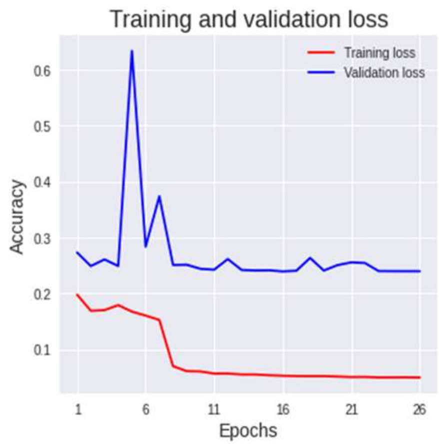


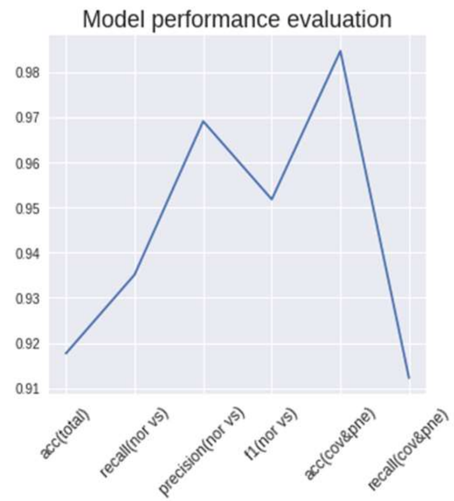
Fine tuning - Inception

③ Freezing pre-trained model layer: 100%(False)



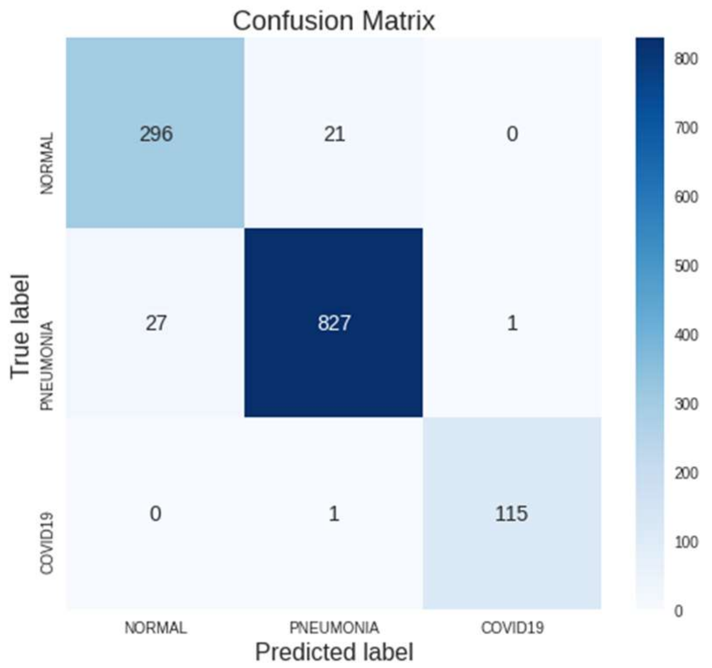


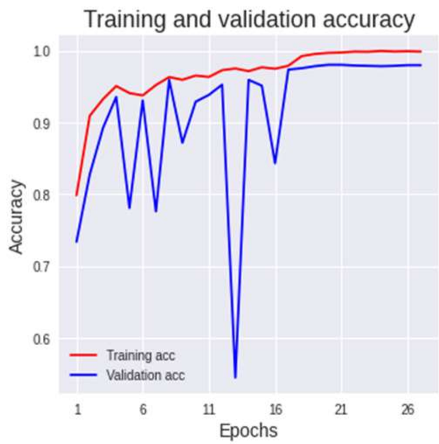


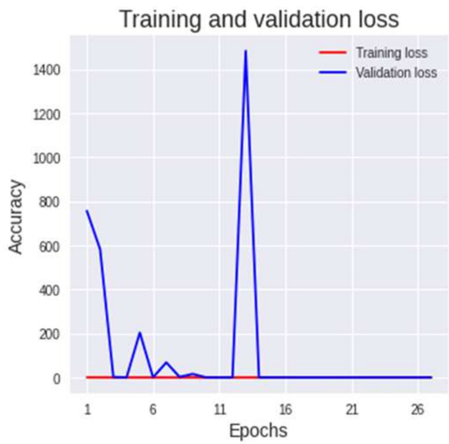


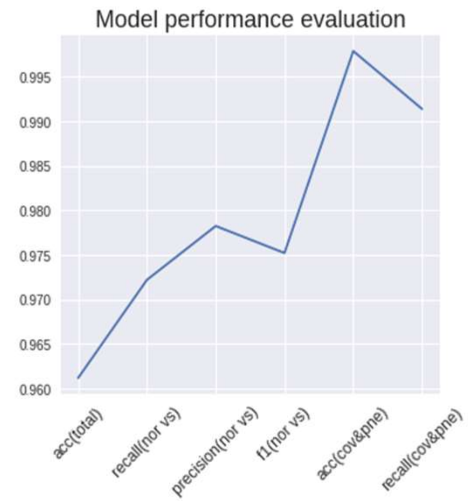
Fine tuning - Inception

④ Dense layer: O



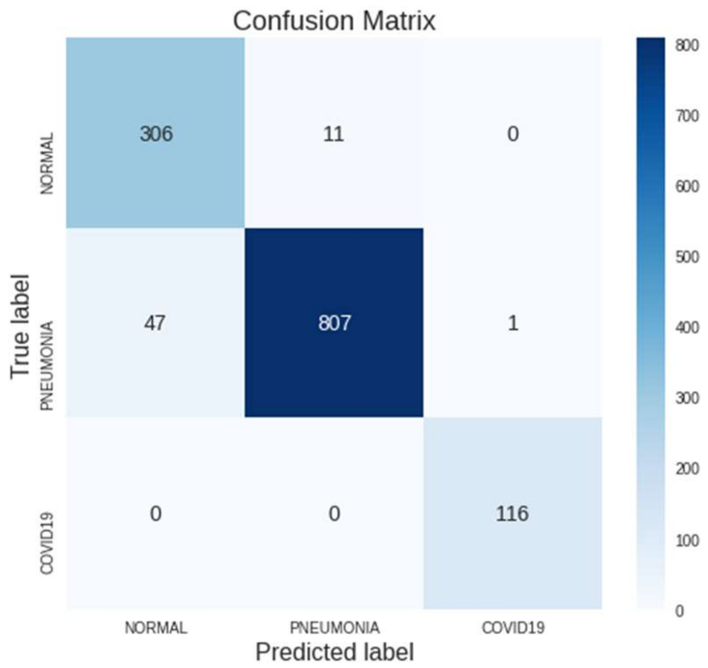


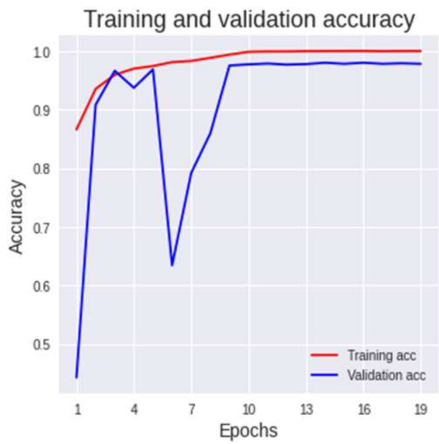


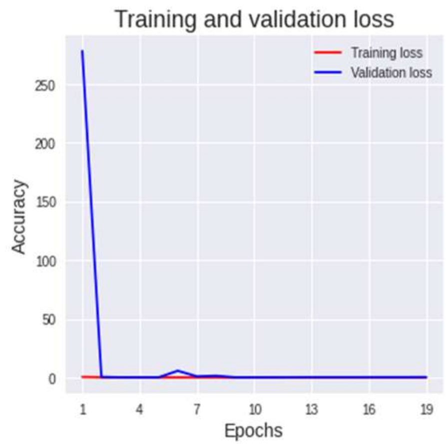


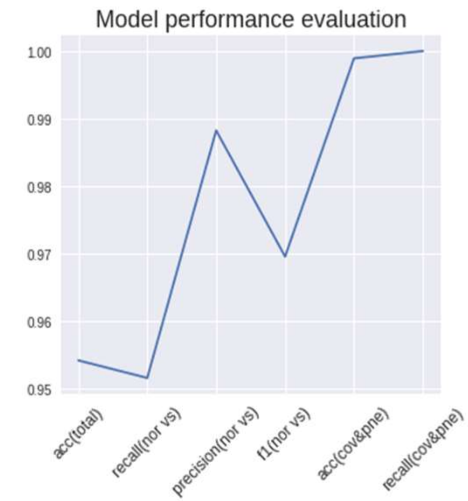
Fine tuning - Inception

⑤ Dense layer: O + Dropout(0.5)



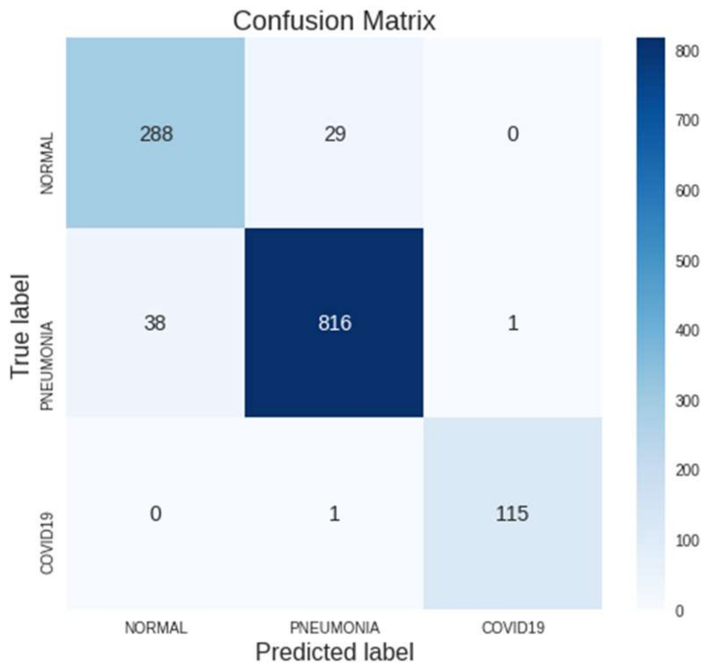


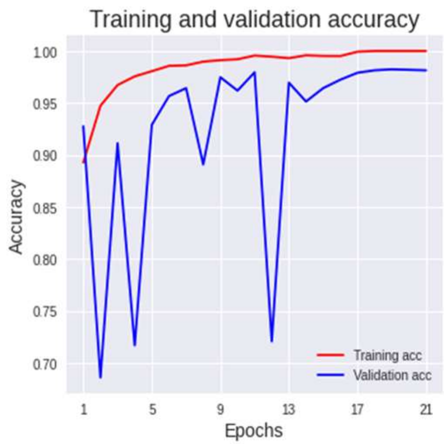


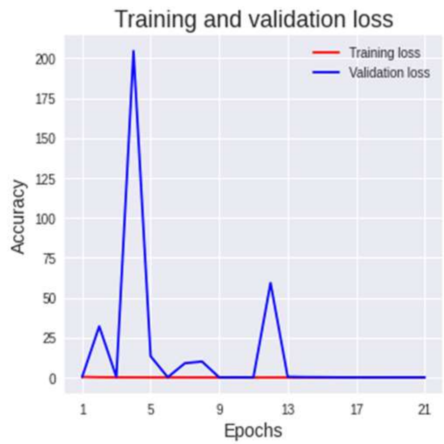


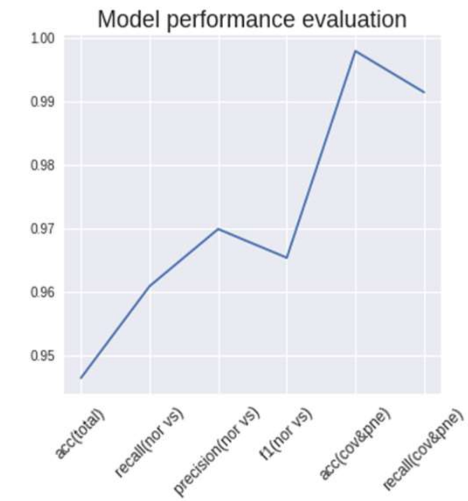
Fine tuning - Inception

⑥ Regularizer: L1



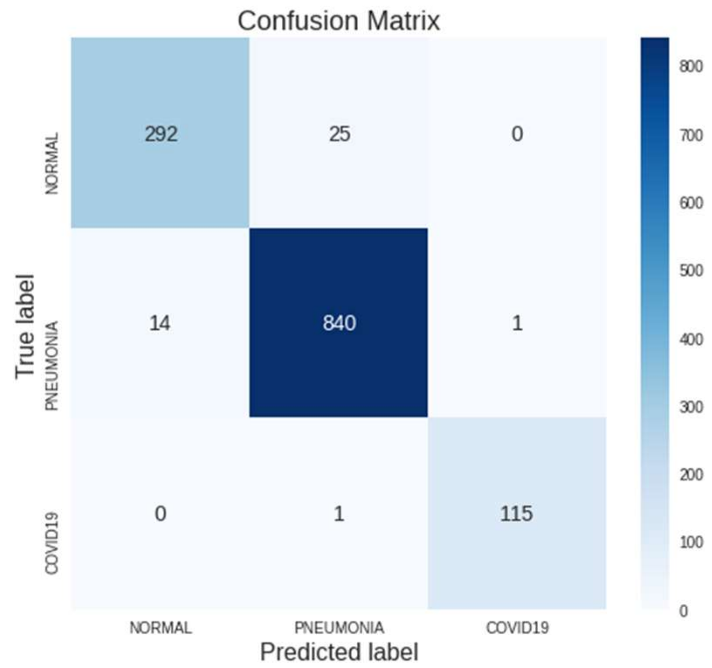


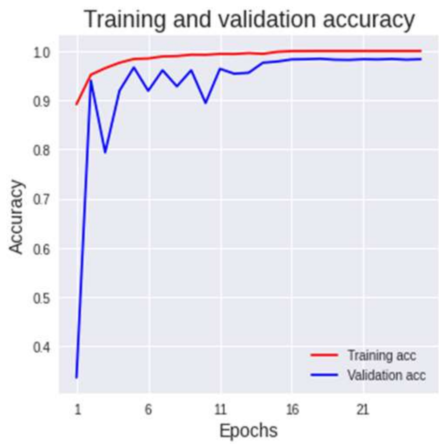


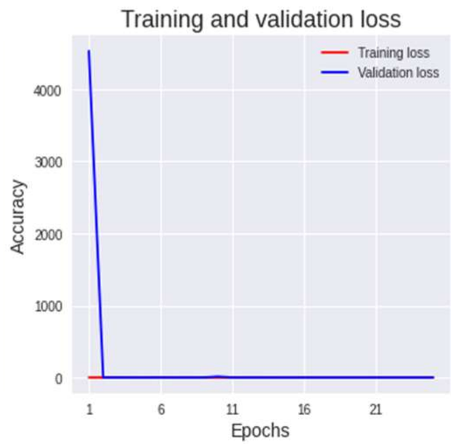


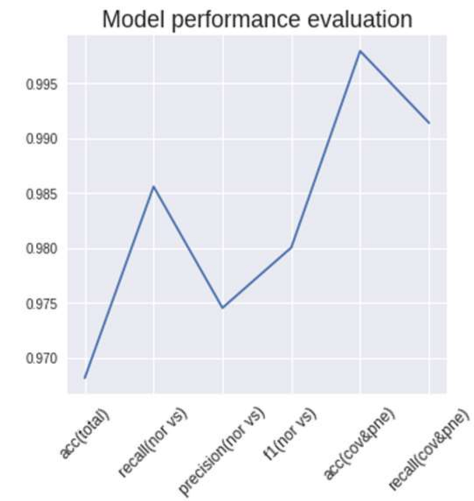
Fine tuning - Inception

⑥ Regularizer: L2



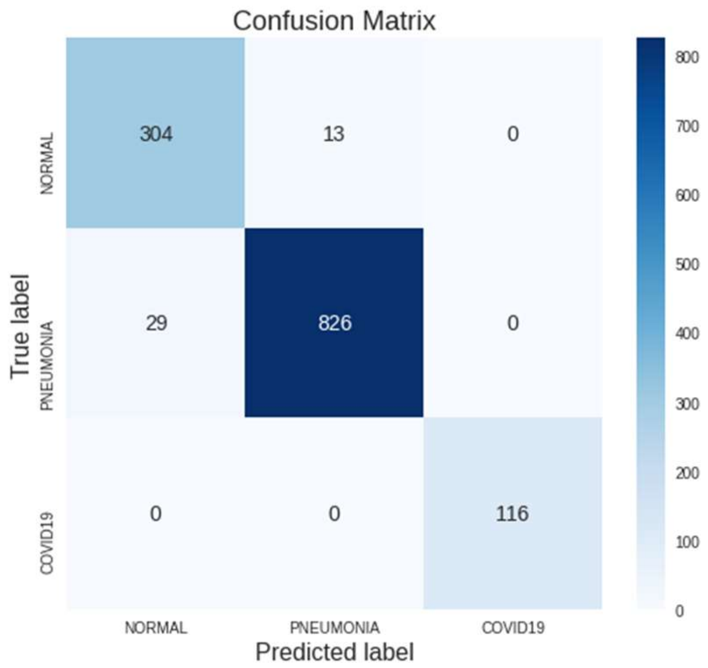


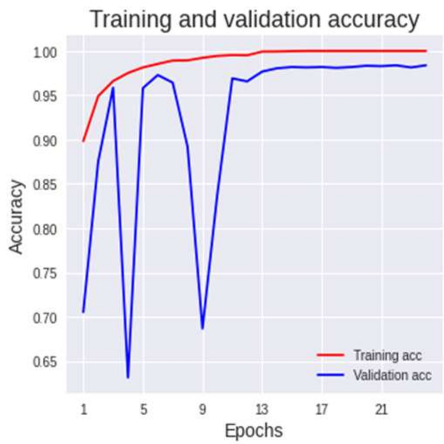


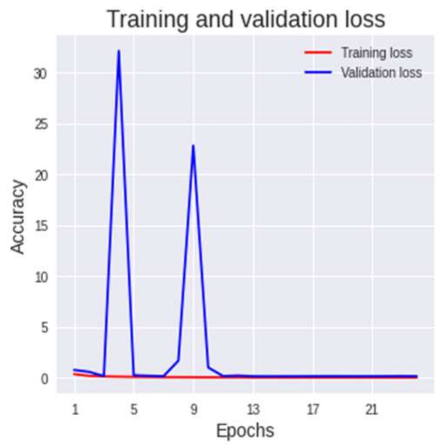


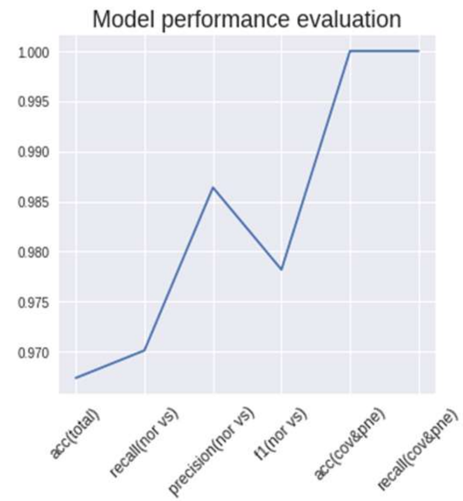
Fine tuning - Inception

⑥ Regularizer: L1 + L2









Fine tuning - Inception

⑨ Freezing pre-trained model layer: 50% / Dense layer : O + Dropout(0.5) / Regularizer: L1 + L2