

Classification of Bone Images

Binary Classification of Bone CT Images Between Normal / Fractured Bone
Using Convolutional Neural Network (CNN)



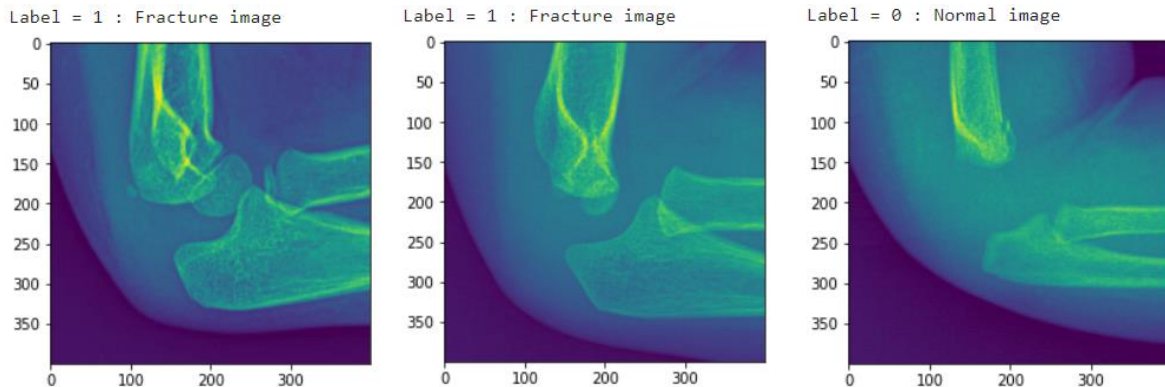
Division of Mechanical and
Biomedical Engineering

Taehee Kim

Classification of Bone Images

1. Objective

제공된 bone CT image를 입력 받아 자동으로 Normal/Fractured Bone으로 분류하는 Convolutional Neural Network (CNN)을 개발하는 것을 목표로 한다. 해당 프로젝트에서는 VGGNet, AlexNet, ResNet의 모델 구조를 참고하여 학습을 진행하였다.



2. Prepare Data

- Train-Val Split: 8:2

데이터의 학습 정도를 파악하기 위해 training set을 8:2의 비율로 나눈 후 80%의 데이터를 가지고 모델 학습을, 20%의 데이터를 가지고 모델 검증을 진행하였다. 학습에 사용된 데이터는 Fractured 140, Normal 141 개로 총 281개이며, 검증에 사용된 데이터는 Fractured 36, normal 36개로 총 72개이다.

- Crop: CenterCrop(400)

주어진 데이터 중 가장자리에는 R, L 등의 부위 정보가 표시되어 있는 데이터가 존재한다. 이와 더불어 가장자리보다 사진의 중심부에 주로 중요 정보, 즉 뼈의 fracture 여부를 확인하고자 하는 부분이 위치하게 된다. 이에 CenterCrop(400)을 활용해 사진의 중심을 기준으로 400x400의 크기로 이미지를 잘라 학습하였다.

- Data Augmentation: RandomHorizontalFlip()

전체 데이터를 클래스 별로 8:2로 나누어 총 281개 이미지를 학습에 사용하게 되었다. 그러나 이는 Deep Convolutional Neural Network를 학습시키기에 너무 적어 모델 학습이 잘 이루어지지 않았다. 이에 RandomHorizontalFlip()을 사용해 이미지를 좌우 반전시켜 학습 데이터를 증강시켜주었다.

3. Modeling

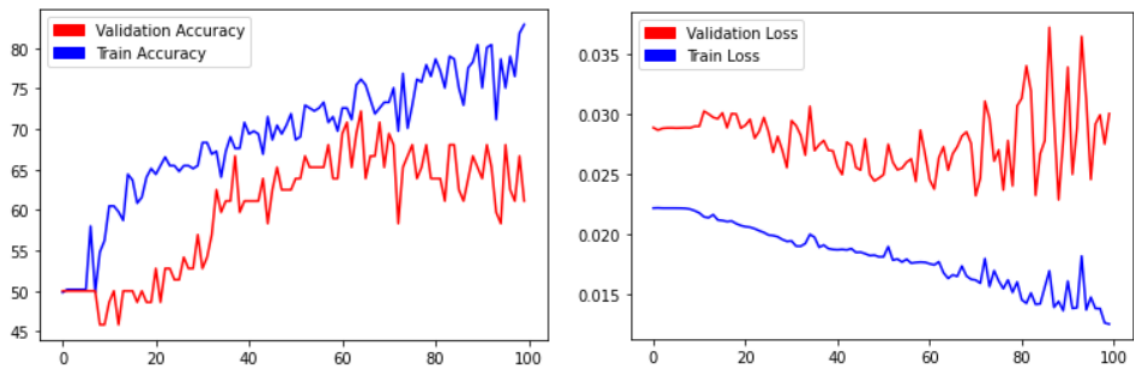
Bone fracture를 탐지하기 위한 CNN을 모델링하기 위해 VGGNet, AlexNet, ResNet 총 3종류의 모델 구조를 적용하였다. 하이퍼 파라미터 및 파라미터 변경을 통해 각 모델의 성능개선을 시도하였으며, Loss Function으로는 CrossEntropyLoss 함수를 사용하여 모델을 학습하였다.

VGGNet

“Very Deep Convolutional Neural Networks for Large-Scale Image Recognition (2015, K. Simonyan A. Zisserman)” 에서 제안된 VGGNet은 3 x 3 의 convolution layer를 깊게 중첩하여 쌓은 형식의 모델이다.

- VGGNet (ConvLayer: 8)

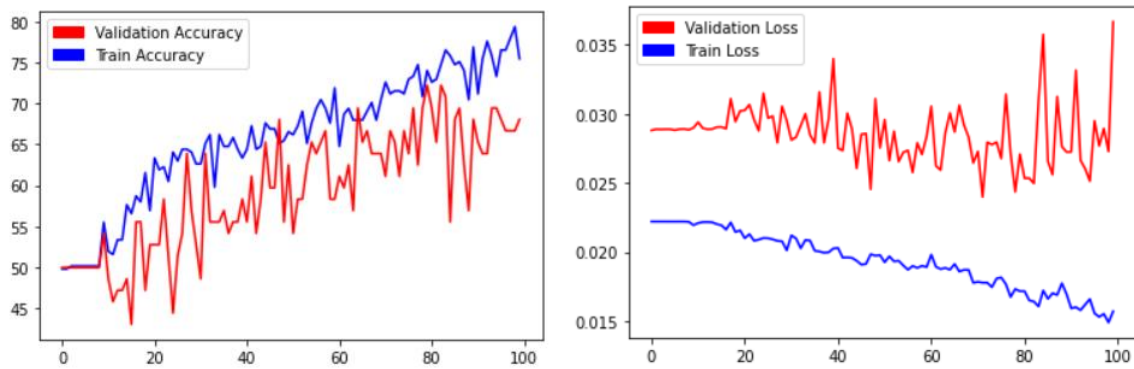
프로젝트의 baseline으로 주어진 VGGNet은 총 8개의 convolutional layer로 이루어져 있다. 이를 활용해 학습한 결과는 다음과 같다.



Accuracy와 Loss graph를 보면 학습이 잘 진행되다가 epoch 70 정도를 기점으로 overfitting이 진행되면서 generalization error 가 생긴 것을 볼 수 있다. validation accuracy가 가장 높았을 때는 epoch 65로, 해당 epoch에서의 weight를 저장해 평가한 결과 train accuracy 72.9535%, validation accuracy 72.2222%가 나왔다. train accuracy가 0.73인 것으로 보아 모델의 representational capacity를 높여야 할 필요성을 느꼈다.

- VGGNet (ConvLayer: 10)

기존의 VGGNet의 모델 capacity를 높이기 위해 convolutional layer를 추가하였고, 각 layer의 채널 수를 높여 학습해야 하는 hidden unit들의 수를 증가시켜 주었다. 결과로 convolutional layer의 수는 8개에서 10개로, 학습할 파라미터 수는 395,634개에서 7,638,210 증가한 모델이 생성되었다. 이를 활용해 학습한 결과는 다음과 같다.



모델의 규모가 증가한 만큼 더 좋은 성능을 기대하였으나 모델의 깊이에 비해 학습 데이터 양이 부족했던 탓인지 학습 과정 중에서의 validation accuracy 및 loss의 oscillation이 더 심하게 나타났다. 해당 모델은 epoch 83에서 가장 높은 validation accuracy를 보였고, 이 때의 weight를 저장해 평가한 결과는 train accuracy 70.8185%, validation accuracy 72.2222% 이다. 이는 baseline VGGNet과 비교했을 때 validation accuracy가 동일하며, train accuracy는 오히려 낮아져 조금 underfitting 된 상태이다.

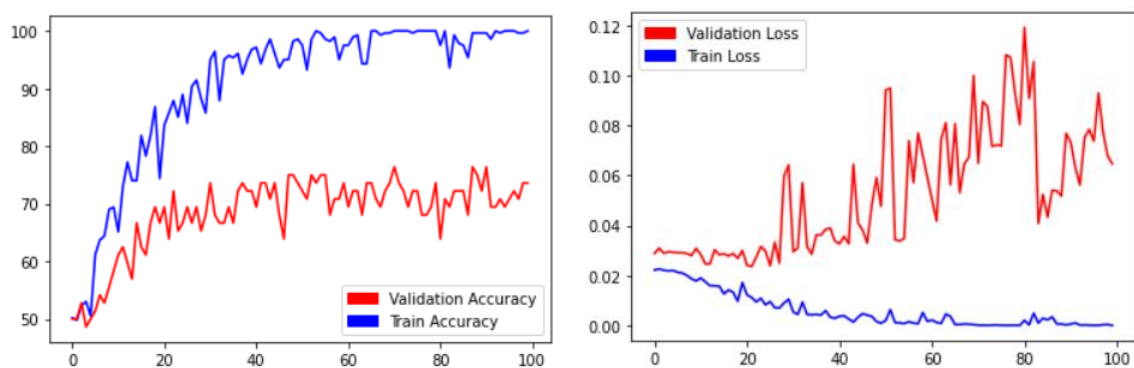
AlexNet

“ImageNet Classification with Deep Convolutional Neural Networks (2012, A. Krizhevsky I. Sutskever)”에서 제안된 AlexNet은 총 5개의 convolutional layer와 3개의 fully connected layer로 구성되어 있는 모델이다.

Input layer → **Conv1** → MaxPool1 → Norm1 → **Conv2** → MaxPool2 → Norm2 → **Conv3** → **Conv4** → **Conv5** → MaxPool3 → Dropout1 → **FC1** → Dropout2 → **FC2** → **Output Layer (SoftMax)**

- AlexNet (Dropout: 0.5, λ : 0)

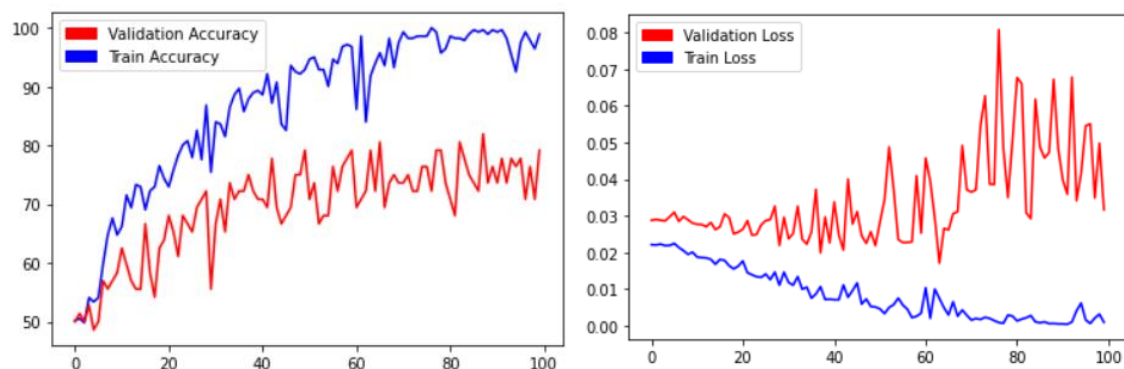
AlexNet 논문에서 제시한 모델 구조를 동일하게 구현하되 400 x 400의 input image에 맞게 첫 번째 fully connected layer의 input을 조정해 주었다. 기본 AlexNet 구조를 활용해 학습한 결과는 다음과 같다.



Accuracy 그래프를 보면 train accuracy가 100%의 정확도 근처에 있는 와중에 validation accuracy는 70% 전후에서 oscillate한다는 것을 볼 수 있다. Epoch 71에 validation accuracy가 가장 높게 나왔으며, 이때의 모델 weight를 저장해 평가한 결과는 train accuracy 100%, validation accuracy 76.3889이다. train과 validation accuracy의 차이로 보아 overfitting이 일어났으며, generalization error를 해결해야 할 것으로 보인다.

- AlexNet (Dropout: 0.8, λ : 0)

Generalization error를 해결하기 위해 모델의 capacity를 감소시키고자 fully connected layer에서의 dropout value를 0.5에서 0.8로 높여 p_{keep} 을 줄여주었다.

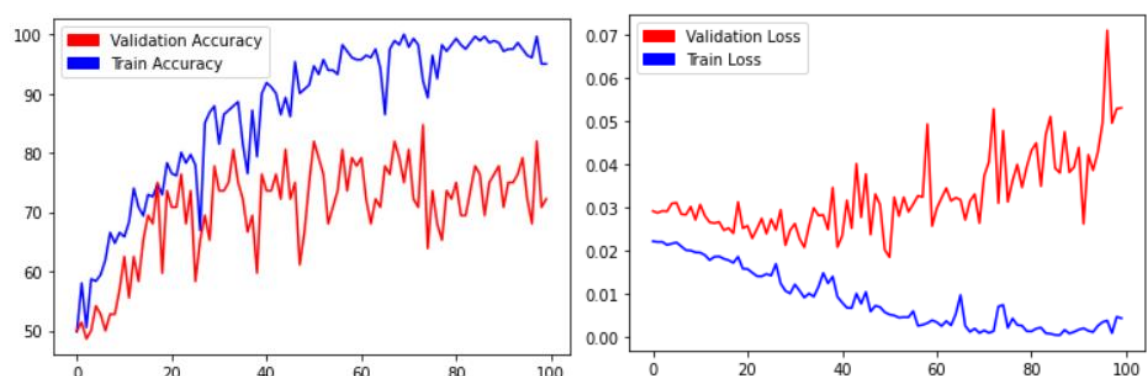


Accuracy graph를 보았을 때 기존의 AlexNet보다 Validation Accuracy가 train accuracy를 보다 잘 따라간 모습을 보였다. best validation accuracy는 epoch 88번째에서 나왔으며 이때의 weight를 저장해 모델을 평가한 결과 train accuracy 100%, validation accuracy 81.944%로 기존 모델보다 정확도가 5% 상승한 모습을 보였다.

- AlexNet (Dropout: 0.8, λ : 1e-4)

앞서 fully connected layer의 dropout을 증가함으로써 모델의 training error를 유지하며 generalization error를 줄이는 것에 성공하였다. 그러나 여전히 training accuracy가 100%인 것에 비해 validation accuracy가 81.944%로 generalization error를 조금 더 줄이고자 dropout을 증가시킨 모델에 추가적으로 L2 Regularization을 적용해 학습을 진행하였다.

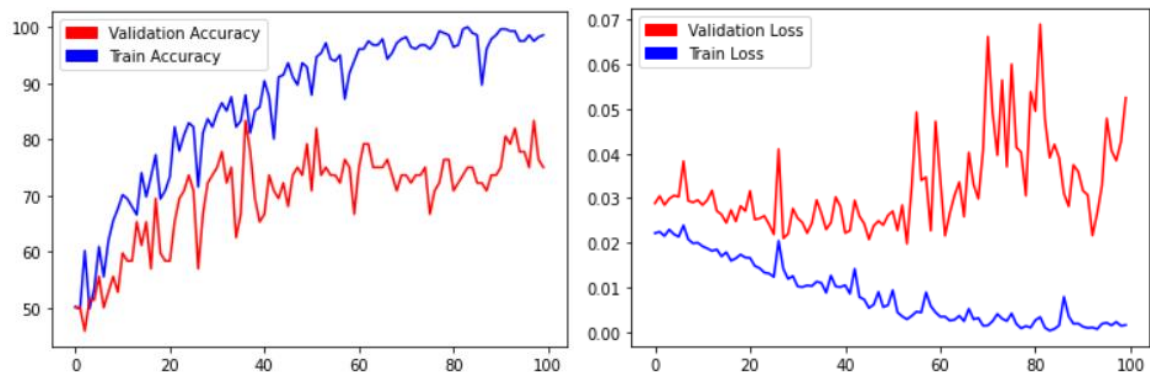
아래는 L2 Regularization 에서의 lambda 값이 1e-4일때의 학습한 결과이다.



74번째 Epoch에 가장 높은 validation accuracy인 84.722%가 나왔다. 그러나 dropout만 적용했을 때의 모델은 training error를 유지하면서 effective capacity를 증가시켰던 반면에 regularization parameter를 적용한 결과 train accuracy 94.6619%, Train Loss 0.0058로 training error가 증가하였다. 또한, 모델 학습 과정에서 accuracy가 많이 oscillate 하는 모습을 보였다.

- AlexNet (Dropout: 0.8, λ : 1e-3)

L2 Regularization의 lambda 값이 1e-3 때의 학습한 결과이다.



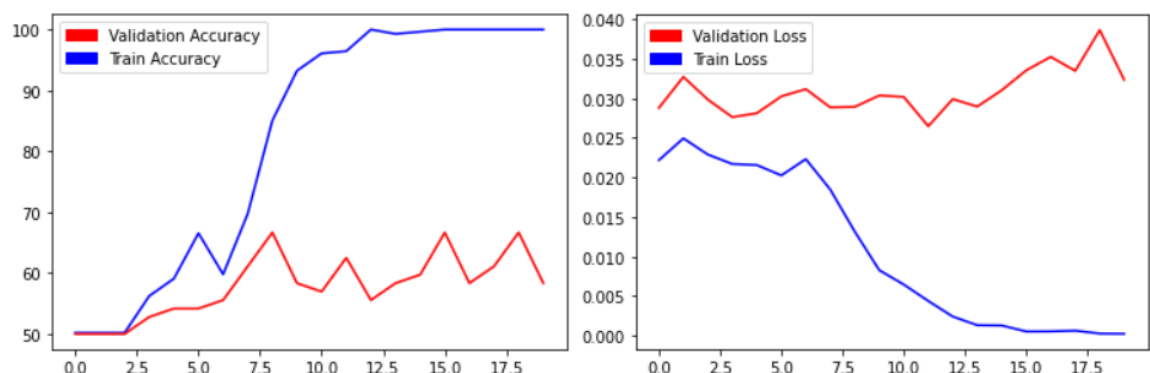
Regularization을 1e-4에서 1e-3으로 증가시킨 결과 가장 높은 validation accuracy에서의 모델은 train accuracy는 87.5445%, validation accuracy 83.33%로 $\lambda=1e-4$ 일 때보다 성능이 낮았다. 반면에 $\lambda=1e-4$ 일 때보다 validation accuracy가 조금 더 안정적으로 수렴하는 모습을 보였다.

ResNet

“Deep Residual Learning for Image Recognition (2016, K. He X. Zhang S. Ren J. Sun)” 에서 제안된 ResNet은 이전 layer의 feature map을 다음 layer의 feature map에 더해주는 ‘Skip Connection’ 구조를 활용하는 모델이다.

- ResNet (ConvLayer: 25)

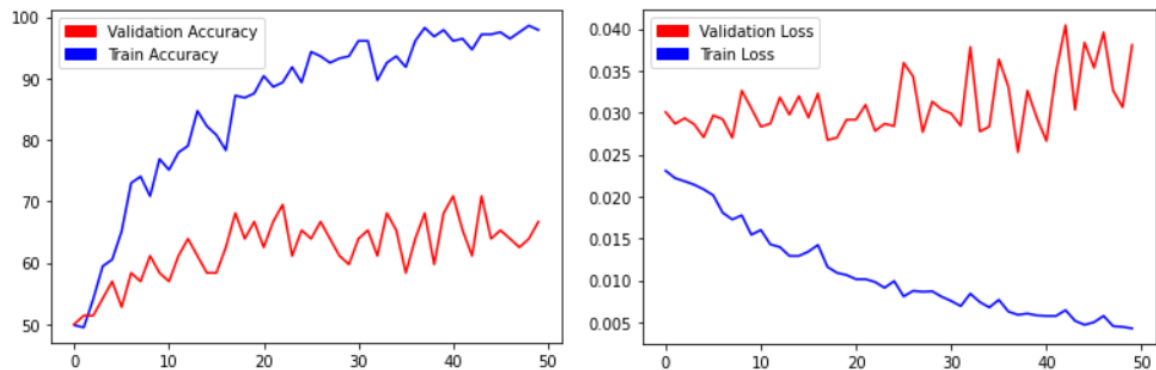
25개의 convolutional layer와 1개의 fully connected layer를 가진 ResNet 모델로 학습하였다.



Epoch 13만에 100% training accuracy에 도달하며 타 모델들보다 훨씬 더 빠르게 수렴하였다. 그러나 representational capacity에 비해 effective capacity는 많이 떨어지며 overfitting이 일어나는 것을 볼 수 있었다.

- ResNet (ConvLayer: 15, λ : 1e-2)

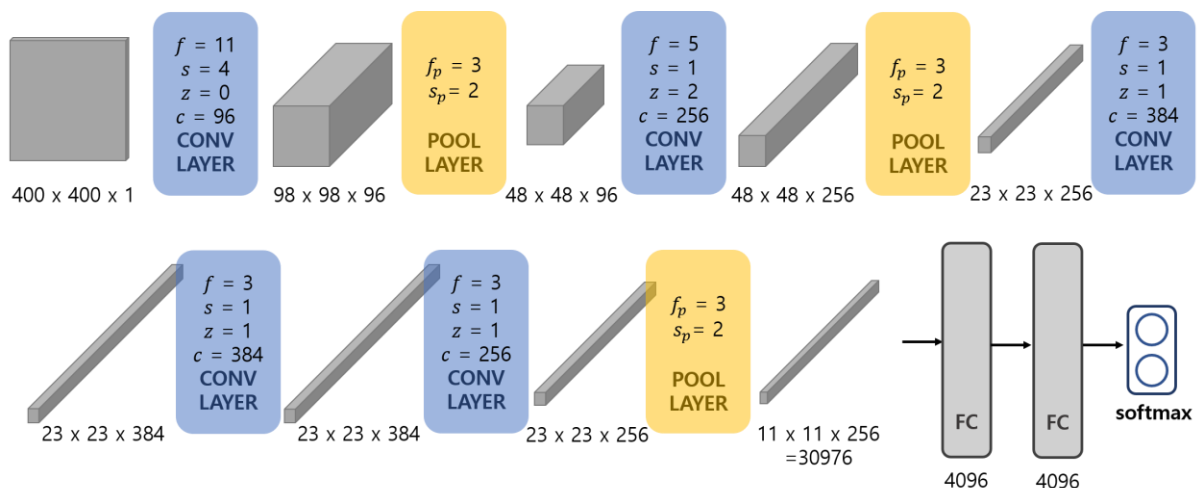
Overfitting을 줄이기 위해 layer 수를 총 26개에서 16개로 줄여 모델의 크기를 낮춰주었고, $\lambda=1e-2$ 의 L2 Regularization을 적용해 주었다.



Accuracy 와 loss 그래프를 살펴보았을 때, 기존의 ResNet26으로 학습을 진행했을 때 보다 validation 수치들이 train을 어느 정도 따라가는 양상을 보였다. Epoch 41에서 가장 높은 validation accuracy를 보였으며 이때 학습 정확도가 96.7972%, 검증 정확도가 70.8333% 이다.

4. Conclusion

VGGNet, AlexNet, ResNet의 구조 및 하이퍼 파라미터들을 변경해가며 총 8번의 학습을 진행한 결과 Dropout과 L2 Regularization을 적용한 AlexNet이 가장 좋은 성능을 보였다. 이에 최종적으로 400x400의 CT 이미지를 입력으로 받고, Fully-Connected Layer에서의 dropout rate가 0.8인 변형된 형태의 AlexNet을 사용하였다. 최종 모델의 구조는 다음과 같다.



Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 96, 98, 98]	11,712
ReLU-2	[-1, 96, 98, 98]	0
MaxPool2d-3	[-1, 96, 48, 48]	0
BatchNorm2d-4	[-1, 96, 48, 48]	192
Conv2d-5	[-1, 256, 48, 48]	614,656
ReLU-6	[-1, 256, 48, 48]	0
MaxPool2d-7	[-1, 256, 23, 23]	0
BatchNorm2d-8	[-1, 256, 23, 23]	512
Conv2d-9	[-1, 384, 23, 23]	885,120
ReLU-10	[-1, 384, 23, 23]	0
Conv2d-11	[-1, 384, 23, 23]	1,327,488
ReLU-12	[-1, 384, 23, 23]	0
Conv2d-13	[-1, 256, 23, 23]	884,992
ReLU-14	[-1, 256, 23, 23]	0
MaxPool2d-15	[-1, 256, 11, 11]	0
Flatten-16	[-1, 30976]	0
Dropout-17	[-1, 30976]	0
Linear-18	[-1, 4096]	126,881,792
ReLU-19	[-1, 4096]	0
Dropout-20	[-1, 4096]	0
Linear-21	[-1, 4096]	16,781,312
ReLU-22	[-1, 4096]	0
Linear-23	[-1, 2]	8,194

Total params: 147,395,970
 Trainable params: 147,395,970
 Non-trainable params: 0

Input size (MB): 0.61
 Forward/backward pass size (MB): 37.64
 Params size (MB): 562.27
 Estimated Total Size (MB): 600.52

Classification Report					
	precision	recall	f1-score	support	
0	0.90	0.78	0.84	36	
1	0.80	0.92	0.86	36	
accuracy			0.85	72	
macro avg	0.85	0.85	0.85	72	
weighted avg	0.85	0.85	0.85	72	
L2 Regularization $\lambda=1e-4$					
	precision	recall	f1-score	support	
0	0.83	0.83	0.83	36	
1	0.83	0.83	0.83	36	
accuracy			0.83	72	
macro avg	0.83	0.83	0.83	72	
weighted avg	0.83	0.83	0.83	72	
L2 Regularization $\lambda=1e-3$					

가장 높은 검증 데이터의 정확도를 보여준 학습 결과는 L2 Regularization $\lambda=1e-4$ 일 때이지만 $\lambda=1e-3$ 일 때의 validation loss 가 더 낮게 나온다는 점, f1-score를 살펴볼 시 $\lambda=1e-3$ 일 때가 양 클래스를 보다 균등하게 예측한다는 점에서 더 안정적이라 판단하였다. 이에 Dropout 0.8, $\lambda=1e-3$ 를 적용해 학습한 AlexNet 모델을 활용해 Test Data의 Class를 예측하였다.

Model	Hyper Parameters	Epochs (saved ep)	Accuracy	F1-Score	Loss	# Params
VGGNet	ConvLayer: 8 $\lambda = 0$ $lr = 0.0001$	100(65)	(Train: 0.73) Val: 0.72	0.72	-	395,634
	ConvLayer: 10 $\lambda = 0$ $lr = 0.0001$	100(83)	(Train: 0.71) Val: 0.72	0.72	-	7,638,210
AlexNet	Dropout 0.5 $\lambda = 0$ $lr = 0.0001$	100(71)	(Train: 1.00) Val: 0.76	0.76	-	147,395,970
	Dropout 0.8 $\lambda = 0$ $lr = 0.0001$	100(88)	(Train: 1.00) Val: 0.82	0.82	-	147,395,970
	Dropout 0.8 $\lambda = 1e-4$ $lr = 0.0001$	100(74)	(Train: 0.95) Val: 0.85	0.85	(Train: 0.0059) Val: 0.0254	147,395,970
	Dropout 0.8 $\lambda = 1e-3$ $lr = 0.0001$	100(37)	(Train: 0.88) Val: 0.83	0.83	(Train: 0.0084) Val: 0.0167	147,395,970
ResNet	ConvLayer: 25 $\lambda = 0$ $lr = 0.0001$	20(9)	(Train: 0.87) Val: 0.67	0.66	-	3,152,370
	ConvLayer: 15 $\lambda = 1e-2$ $lr = 0.0001$	50(41)	(Train: 0.97) Val: 0.71	0.71	(Train: 0.0053) Val: 0.0321	213,234