# Computerized labeling method different Retinal degeneration from Optical Coherence Tomography Images (OCT) based on convolutional neural networks

김혜성, 장건
연세대학교 의공학부
comsicomet@yonsei.ac.kr geon8074@naver.com

## 요 약

황반변성은 노화, 유전적인 요인, 독성 염증 등에 의해 황반의 기능이 떨어지면서 시력이 감소되고, 심할 경우 시력을 완전히 잃기도 하는 질환이다. 황반변성의 종류로는 choroidal neovascularization (CNV), diabetic macular edema (DME), Drusen 이 존재한다. 일반적으로 CNN 에서는 층이 깊어지면 성능이 좋아지지만 일정 수준이상으로 층이 깊어지면 오히려 성능이 감소한다. 이와 같은 현상이 나타나는 이유는 역전파 단계에서 기울기 소실 문제가 발생하기 때문이다. 본 논문에서는 기울기 소실 문제를 해결하기 위해 ResNet 네트워크를 이용하여 황반변성을 분류하였다. 본 논문에서 학습시킨 분류기의 결과는 95% 이상의 정확도를 보였다. 향후 본 연구를 통해 학습된 ResNet18 의 층깊이를 증가시켜 더 높은 정확도를 갖는 분류기로 황반변성을 진단할 수 있다.

## 1. 서론

눈의 안쪽 망막의 중심부에 위치한 신경조직을 황반이라고 하는데, 시세포의 대부분이 이곳에 모여 있고 물체의 상이 맺히는 곳도 황반의 중심이므로 시력에 상당히 중요한 역할을 담당하고 있다. 황반변성은 노화, 유전적인 요인, 독성 염증 등에 의해 황반의 기능이 떨어지면서 시력이 감소되고, 심할 경우 시력을 완전히 잃기도 하는 질환이다. 노화로 인한 '연령관련 황반변성'(AMD, age-related macular degeneration)은 노란 침착물인 드루젠(drusen)이 제거되지 않고 쌓여서 생겨나는 질환이다. 여기서 드루젠은 세포 대사로 생긴 노폐물이 망막색소상피에 쌓여 있는 것으로, 드루젠이 많으면 세포가 변형되고 손상될 위험이 크며 이는 노화로 인해 생긴다.

본 연구에서 분류한 황반변성으로는 CNV, DME, Drusen 이다. 먼저 CNV 는 습성 황반변성(wet AMD)이며 망막 하단에 혈액이 관찰된다. DME 는 당뇨병에 의해 발생한 고혈당이 망막에서 말초 순환 장애를 발생시키는 합병증이다. DME 는 망막 내에서 굵은 혈액이 관찰되는 시각적 특징이 있다. Drusen 은 망막에 쌓이는 노란 침착물이며 연령관련 황반변성의 초기에 관찰된다. 드루젠의 시각적 특징으로 하단이 고르지 않고 돌출된 부분들이 관찰된다.

OCT(Optical coherence tomography)는 측정 대상의 비침습적 단층 촬영을 구현할 수 있는 새로운 이미징 기술이다.

최근 몇 년 동안 AlexNet, 32 VGG, 33 GoogLeNet, 34 및 ResNet35 와 같은 CNN 아키텍처의 변형이 개발되었다. 원칙적으로 이와 같은 모든 CNN 모델은 전이 학습을 기반으로 모델을 훈련하여 OCT 이미지 데이터 세트에 사용할 수 있다. 최근에는 기술이 비약적으로 발전하여 100 nm 가 넘는 넓은 파장 대역을 1 MHz 이상의 높은 반복률로 파장 가변이 가능한 광원이 보고되었으며, 2 차원 단층영상의 실시간 재현을 넘어서 3 차원 입체영상을 실시간으로 재현하는 수준까지 이르고 있다.[1]

딥러닝을 OCT 에 적용한 다음과 같은 연구가 선행되었다. ResNet50 을 사용 99.49% 정확도를 도출한 사례[2], U-Net 를 사용하여 88.1% 정확도를 도출한 사례[3], Xception 모델을 이용한 분류기로 99% 이상 정확도로 분류한 사례[4].

본 논문에서 convolutional neural networks (CNN)의 성능을 향상시키는 ResNet 네트워크를 이용하여 OCT 로 촬영한 황반변성의 병변을 분류하였다.

## 2. 황반 변성 분류기

### 2.1 망막 안저 사진

그림 1 은 OCT 로 촬영한 망막내부 사진들을 드루젠, 혈액이 관찰되는 모양, 정도에 따라 4 가지로 분류한 예시들이다.

본 연구에서 사용된 초기 데이터는 83,484 장이다. 실험이 진행될수록 데이터 분배를 달리하였다. 축소된 shortened 의 데이터 개수는 27,794 장, 최종적으로 연구에서 사용한 shortened2 의 데이터 개수는 24,800 장이다.

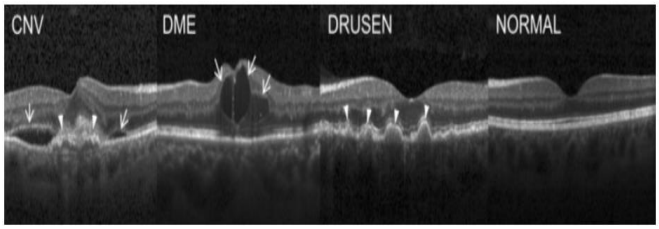



그림 1. OCT 로 촬영한 병변에 따른 안저 사진

### 2.2 전처리

딥러닝으로 안저 사진을 분류하기 위하여 pytorch 로 데이터 전처리를 진행하였다. OCT 촬영사진들은 흑백 사진이기에 Grayscale 을 사용하여 흑백으로 변환하였고, 400x400 으로 crop 한 뒤 평균 0.5, 표준편차 0.5 로 정규화를 실행하였다. CNV, DME, DRUSEN, NORMAL 4 개의 폴더에 들어있는 안저 사진들을 4 개의 레이블(label)로 라벨링(Labeling) 하였다. 배치 사이즈는 10, 코어 수는 4 개를 사용하여 데이터로더 변수에 저장하였다. drop_last 를 실행하여 마지막 배치의 여분의 데이터를 사용하지 않음으로써 오버피팅을 최소화하였다. 코어는 4 개를 사용하고, Google colab GPU 로 학습을 진행시켜 연산의 효율을 극대화하였다.

데이터를 Train : Validation : Test = 8 : 1 : 1 의 비율로 나누었다. 따라서 RenNet18 모델의 학습에 20000 개의 학습셋과 2400 개의 검증셋을 사용하고 테스트셋은 2400 개를 사용하여 모델을 학습시켰다.

### 2.3 분류기

ResNet18 은 잔여 블록(Residual Block)을 이용하기 때문에 기울기 소실 문제를 해결할 수 있다. 그림 2 에서 나타내었듯이 일반적으로 층이 깊은 CNN 에서 실제로 내재한 mapping 인 H(x)를 곧바로 학습하는 것이 어려우므로, 대신 잔여 블록을 이용해서 F(x)=H(x)-x 를 학습한다. 이를 통해 네트워크의 최적화(optimization) 난이도를 낮추며 기울기 소실 문제를 해결한다. 다시 말해 잔여 블록을 이용하면 층을 깊게 쌓아도 모델의 성능을 더욱 향상시킬수 있다.

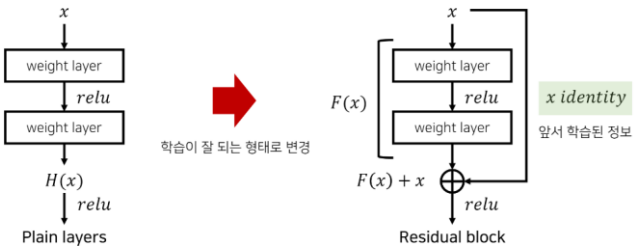



그림 2. residual block 설명[5]

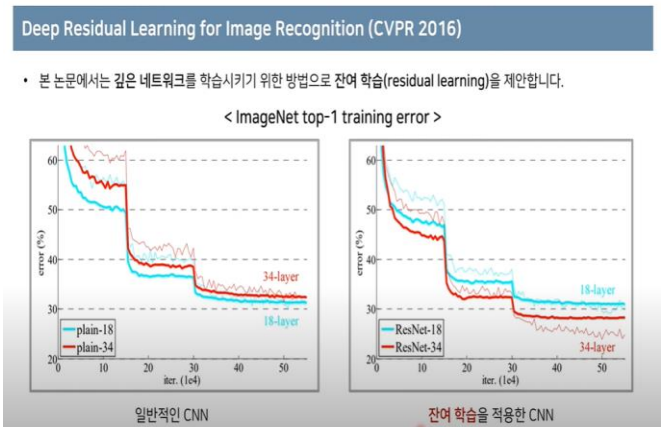



그림 3. 일반적인 CNN 과 잔여학습을 적용한 CNN 학습 오차비교[5]

ResNet18 을 이용한 분류기의 블록도를 그림 4 에 나타내었다. 전처리된 안저 사진의 사이즈는 400 x 400 x 1 이며 ResNet18 은 4 개의 block layer 를 사용하여 학습시킨 뒤 global average pooling 과 softmax 를 거쳐 결과를 도출하였다.

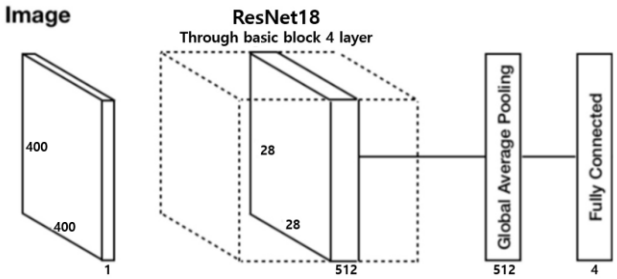



그림 4. ResNet18 분류기의 블록도

ResNet18 의 Optimizer 는 Adam 을 사용하고, 학습률은 0.001, eps 는 1e-08, weight_decay 는 0, amsgrad 은 False 로 설정한다. lr_scheduler.StepLR 라이브러리를 사용하여 스텝 사이즈는 7, 감마는 0.1 로 설정하여 7 번째 스텝마다 옵티마이져의 학습률에 0.1 을 곱하여 학습률을 업데이트 한다. 에폭은 20 으로 설정하여 전체 데이터에 대해 20 번 학습하도록 한다.

본 연구에서는 Meta Pseudo Labels 논문에 기재된 '모델들의 정확도 순위' 지표에 따라 pytorch 에서 제공하는 모델 중 상위 정확도를 가진 모델인 ResNet 을 선정하였다. 분류기를 ResNet18 로 설계하여 layer 가 깊어지면 모델의 성능은 더욱 향상하였다.

ResNet18 모델은 input layer, 2 개의 BasicBlock(Basicblock 은 4 개의 hidden layer 로 구성), output layer 로 설계한다.

| Method | # Params | Extra Data | ImageNet Top-1 | ImageNet Top-5 | ImageNet-ReaL [6] Precision@1 |
|---|---|---|---|---|---|
| ResNet-50 [24] | 26M | – | 76.0 | 93.0 | 82.94 |
| ResNet-152 [24] | 60M | – | 77.8 | 93.8 | 84.79 |
| DenseNet-264 [28] | 34M | – | 77.9 | 93.9 | – |
| Inception-v3 [62] | 24M | – | 78.8 | 94.4 | 83.58 |
| Xception [11] | 23M | – | 79.0 | 94.5 | – |
| Inception-v4 [61] | 48M | – | 80.0 | 95.0 | – |
| Inception-resnet-v2 [61] | 56M | – | 80.1 | 95.1 | – |
| ResNeXt-101 [78] | 84M | – | 80.9 | 95.6 | 85.18 |
| PolyNet [87] | 92M | – | 81.3 | 95.8 | – |
| SENet [27] | 146M | – | 82.7 | 96.2 | – |
| NASNet-A [90] | 89M | – | 82.7 | 96.2 | 82.56 |
| AmoebaNet-A [52] | 87M | – | 82.8 | 96.1 | – |
| PNASNet [39] | 86M | – | 82.9 | 96.2 | – |
| AmoebaNet-C + AutoAugment [12] | 155M | – | 83.5 | 96.5 | – |
| GPipe [29] | 557M | – | 84.3 | 97.0 | – |
| EfficientNet-B7 [63] | 66M | – | 85.0 | 97.2 | – |
| EfficientNet-B7 + FixRes [70] | 66M | – | 85.3 | 97.4 | – |
| EfficientNet-L2 [63] | 480M | – | 85.5 | 97.5 | – |
| ResNet-50 Billion-scale SSL [79] | 26M | 3.5B labeled Instagram | 81.2 | 96.0 | – |
| ResNeXt-101 Billion-scale SSL [79] | 193M | 3.5B labeled Instagram | 84.8 | – | – |
| ResNeXt-101 WSL [42] | 829M | 3.5B labeled Instagram | 85.4 | 97.6 | 88.19 |
| FixRes ResNeXt-101 WSL [69] | 829M | 3.5B labeled Instagram | 86.4 | 98.0 | 89.73 |
| Big Transfer (BiT-L) [33] | 928M | 300M labeled JFT | 87.5 | 98.5 | 90.54 |
| Noisy Student (EfficientNet-L2) [77] | 480M | 300M unlabeled JFT | 88.4 | 98.7 | 90.55 |
| Noisy Student + FixRes [70] | 480M | 300M unlabeled JFT | 88.5 | 98.7 | – |
| Vision Transformer (ViT-H) [14] | 632M | 300M labeled JFT | 88.55 | – | 90.72 |
| EfficientNet-L2-NoisyStudent + SAM [16] | 480M | 300M unlabeled JFT | 88.6 | 98.6 | – |
| Meta Pseudo Labels (EfficientNet-B6-Wide) | 390M | 300M unlabeled JFT | 90.0 | 98.7 | **91.12** |
| Meta Pseudo Labels (EfficientNet-L2) | 480M | 300M unlabeled JFT | **90.2** | **98.8** | 91.02 |

그림 5. Meta Pseudo Labels,
ImageNet 에 대한 모델들의 정확도 순위[6]

## 2.4 학습 결과

학습 결과, 학습 오차는 0.0858, 학습 정확도는 97.29%, 검증 오차는 0.2046, 검증 정확도는 93.46%로 나타났으며 최적의 검증 정확도는 94.33%, 학습 시간은 435 분 27 초로 학습이 완료되었다.

예측 결과, 테스트 오차는 0.1801, 테스트 정확도는 95.21%, 테스트 시간은 3 분 32 초로 새로운 데이터에 대한 예측이 높은 정확도로 수행되었다.

# 3. 실험 결과 및 분석

본 연구는 다음과 같은 순서로 하이퍼 파라미터를 조절하여 경험적으로 결론을 도출하였다.

## 3.1 최적의 optimizer 선정

### 3.1-1 Optimizer 변경

Gradient Descent(GD)는 전체 학습셋을 사용하여 계산하므로 한 스텝을 내딛을 때, 전체 데이터에 대한 Loss function을 계산하므로 계산양이 늘어나 학습 시간이 오래 소요된다. Stochastic Gradient Descent(SGD)는 전체 학습셋을 미니 배치로 나눠서 loss function을 계산하여 계산속도를 늘릴 수 있기에 SGD를 첫 optimizer로 사용하였다.

### 3.1-2 Optimizer를 SGD에서 Adam으로 변경

SGD의 Local minium 문제, 즉 손실함수의 최솟값이 아닌 극솟값이 되는 문제가 발생하여 테스트 결과가 좋지 않았다. 이 경우 기울기가 0이 되어 더 이상 매개변수가 갱신되지 않아 학습이 멈추기 때문에 최신 기울기와 모멘텀을 반영할 수 있도록 optimizer를 조절하였다.

Adam optimzer는 RMSProp, Momentum을 포함한다. RMSProp은 기울기를 지수 가중 이동 평균을 사용하여 최신 기울기들을 더 크게 반영되도록 하며, Momentum은 gradient값이 0인 곳에서도 관성에 의해 업데이트를 수행될 수 있도록 한다.

이를 통해 방향과 스텝사이즈를 보다 적절하게 조

절하고자 optimizer를 Adam으로 변경하였다. 표1에서 볼 수 있듯이 Adam을 사용하여 정확도가 상승하고 학습시간을 단축하였다.

**표1. Optimizer에 따른 정확도 결과**

| Optimizer 실험 | SGD | ADAM |
|---|---|---|
| 학습 정확도 | 89.35% | 89.58% |
| 검증 정확도 | 85.55% | 85.69% |
| 테스트 정확도 | 73.03% | 74.12% |
| 학습시간 | 357분 | 257분 |

### 3.2 최적의 학습률 선정

아래 그림처럼 학습률이 지나치게 작으면 최종 성능에 이르는 데 걸리는 학습 시간이 굉장히 오래 걸릴 뿐만 아니라 최소값이 아닌 극솟값에 빠지는 문제가 발생한다. 반면, 학습률이 지나치게 크면 입력층으로 갈수록 전파되는 오차가 되려 폭증하는 현상으로 인해 최소값을 지나칠 수 있다.

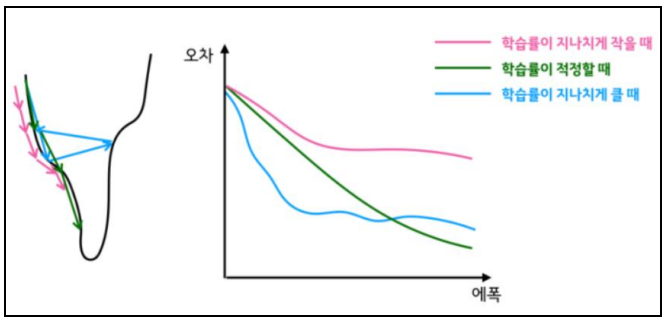



그림6. 학습률에 따른 오차 변화도[7]

**표2. 학습률에 따른 정확도 도표**

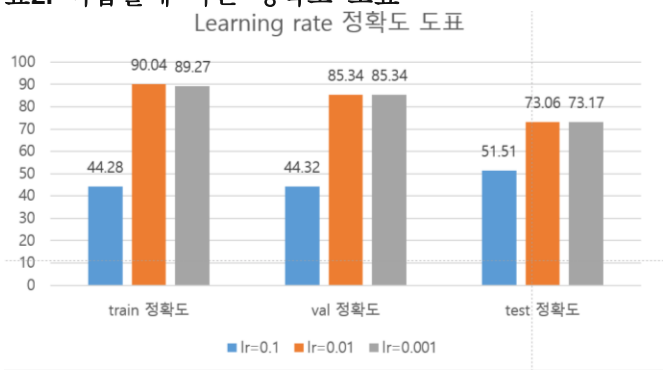



표2는 학습률을 달리 설정하여 실험한 결과이다.

**초기조건**

학습률를 제외한 다른 파라미터는 동일하게 하였다. 에폭은 60, 배치 사이즈는 32, 7에폭마다 학습률을 0.1씩 곱하여 감소시켰다.

**결과 및 분석**

학습률이 0.1일 때 정확도가 매우 낮고 학습률이 0.01과 0.001일 때 정확도는 비슷하지만 학습 초기에 노이즈가 많다.

**결론**

따라서 적절한 학습률을 0.001 로 선정하였다.

### 3.2 최적의 에폭 수 선정

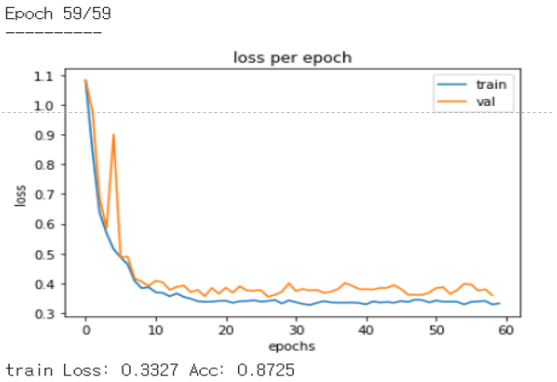



그림 7. 에폭 60 의 정확도 결과

**초기조건**

60epochs 학습 실행, Lr_scheduler.StepLR 라이브러리를 사용하여 7번째 에폭마다 학습률을 0.1배씩 감소시킨다.

**결과 및 분석**

그림8을 참고하여 학습이 진행함에 따라 20번째 에폭부터 학습 정확도가 눈에 띄게 증가하지 않고 검증 오차가 감소되지 않음. 오히려 검증 오차가 진동(oscillating)하는 모습을 보인다.

**최종 결론**

적절한 에폭 수를 20으로 선정하였다.

### 3.3 데이터 재분배

Shortened data

| 단위:개수 | CNV | DME | DRUSEN | NORMAL |
|---|---|---|---|---|
| Train | 10,012 | 3,102 | 2,364 | 7,113 |
| Val | 1,249 | 387 | 296 | 886 |
| Test | 872 | 377 | 280 | 856 |

Shortened2 data

| 단위:개수 | CNV | DME | DRUSEN | NORMAL |
|---|---|---|---|---|
| Train | 5000 | 5000 | 5000 | 5000 |
| Val | 600 | 600 | 600 | 600 |
| Test | 600 | 600 | 600 | 600 |

그림 8. 데이터 재분배 변화

**변경 배경**

축소된 데이터로 딥러닝 실행한 많은 표본에서 공통적으로 학습 정확도가 테스트 정확도와 검증 정확도보다 낮게 나왔다. 즉, 편향된 결과값이 출력되었고 데이터 분배를 문제 요인으로 의심하였다.

**초기조건**

그림9에서처럼 Shortened version을 shortened2 version으로 변경한 것 이외에는 모두 동일하다. 에폭 수는 60, 배치 사이즈는 30, 초기 학습률은 0.001로 설정하였다.

**결과 및 분석**

그림10의 결과에 알 수 있듯이 Shortened version과 shortened2 version의 학습 정확도와 검증 정확도의 결과는 비슷한 수준이다.

그러나 shortened2 version의 테스트 정확도가 대폭 상승하였다.

**최종 결론**

데이터 재분배로 편향된 결과값을 바로잡을 수 있었다. 결론적으로 shortened2 데이터의 사용을 결정하였다.

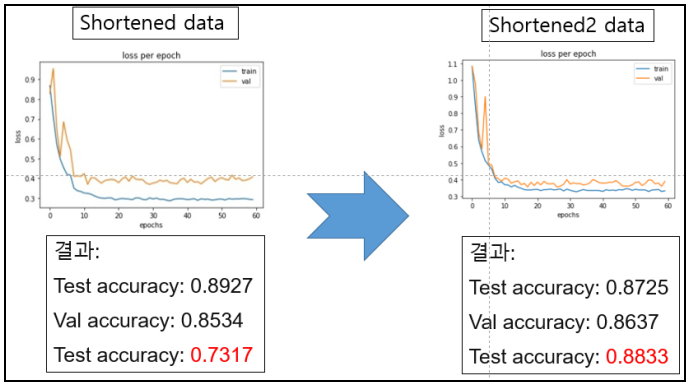


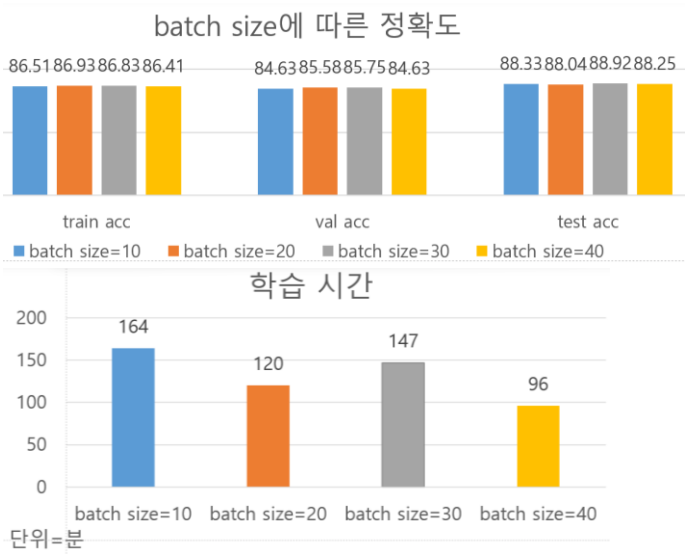

그림 9. 데이터 재분배 변화

### 3.4 최적의 배치 사이즈 선정

배치 사이즈가 클수록 최종 성능에 도달하는데 걸리는 총 훈련 시간이 단축된다. 그러나 배치 사이즈가 크면 오차를 최소화하는 방향이 이미 정해졌을 확률이 크기 때문에 극솟값(local minimum)에서 벗어나기가 힘들다. 경사하강법을 사용할 때 극솟값이 최소값이 아니라는 사실을 분간할 수 없어서 오차를 줄일 수 있는 여지가 있음에도 학습효과가 나타나지 않는다.

즉, 배치 사이즈가 크면 훈련시간은 단축되지만 최종 정확도는 떨어진다[8].

**표 3. 배치 사이즈에 따른 정확도와 학습시간**




**초기조건**

batch size를 제외한 모든 조건 동일하다.

**결과 및 분석**

표3에서 나타내었듯이 배치 사이즈에 따른 정확

도는 유의미한 차이가 없다. 그러나 배치 사이즈가 클수록 전반적으로 학습시간이 단축하는 효과가 발생한다.

**최종 결론**

Colab 에서 수용가능한(out of memory 로 세션이 중단되지 않는) 수준에서, 최대한의 배치 사이즈가 효율성이 좋다.

## 3.5 crop size 키우기

**튜닝 배경**

데이터의 픽셀크기는 매우 다양하므로, 본 실험 이전에는 픽셀크기를 224x224로 통일시켰다.

그림 11에서 데이터 사진을 살펴보면 224x224의 crop사진으로 병변을 분류가능한 부분이 포함되지 않을 가능성이 있다. 그러므로 본 실험에서는 crop size를 최대로 키워서 학습하고자 한다.

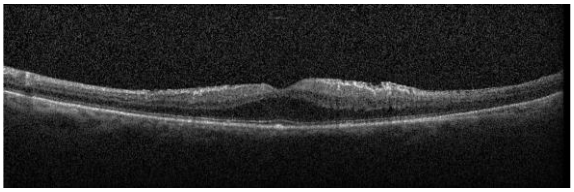


그림 10. 1536x496, 512x512 사이즈 이미지

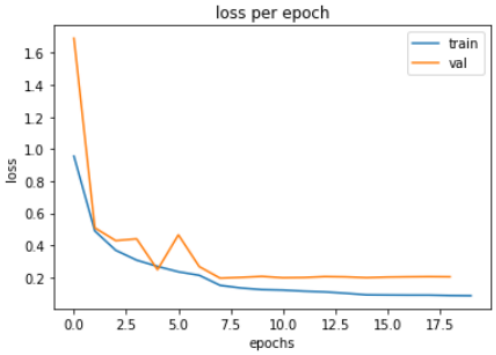



그림 11. 변경된 Crop size 의 모델 학습 결과

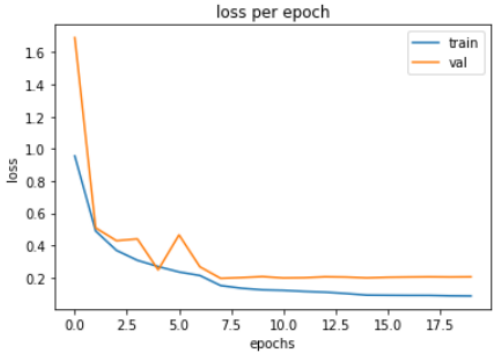



그림 12. 변경된 Crop size 의 모델 테스트 결과

**초기조건**

Centercropsize 메소드를 이용하여 구글 코랩에서 메모리 부족의 문제가 발생하지 않는 선에서 400x400 픽셀까지 crop size를 키웠다. 배치 사이즈는 10, 에폭 수는 20, 초기 학습률은 0.001로 설정하였다.

**결과 및 분석**

학습 정확도는 97.29%, 테스트 정확도는 95.21%로 출력되었다.

**최종 결론**

정확도가 상당히 상승하였고 다른 예시들을 견주어 보아도 만족스러운 결과를 도출하였다.

## 3.6 모델 평가

모델을 평가하는 지표인 Sensitivy, Specificity, Accuracy 는 다음과 같다.

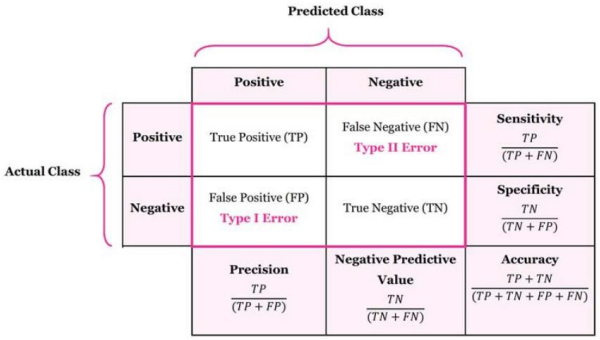



그림 13. Confusion matrix comprehesion[9]

최종적인 연구결과의 Sensitivity, Specificity, Accuarcy 는 다음과 같다.

**표 4. 각 클래스의 모델 평가 지표**

| | sensitivity | specificity | accuracy |
|---|---|---|---|
| CNV | 89.83% | 98.78% | 96.54% |
| DME | 98.67% | 98.44% | 98.50% |
| DRUSEN | 96.67% | 98.11% | 97.75% |
| NORMAL | 98.67% | 99.28% | 99.13% |

### 3.7 오류 데이터 확인

Confusion matrix 를 살펴본 결과 오류가 자주 발생한 부분은 CNV 와 DRUSEN 의 병변 분류에서 나타났다. 오류가 발생한 사진들을 직접 관찰한 결과 CNV 의 시각적 특징인 혈액의 분포가 나타나지 않았고 DRUSEN 의 특징들과 상당히 유사한 모습이 나타났다. 또한, OCT 로 촬영한 사진들의 이미지 왜곡, 회전, 잘림 현상들이 관찰되었다. 이를 통해, OCT 데이터 자료들의 근본적인 문제가 딥러닝 모델의 오류를 발생시킨 것으로 판단된다.

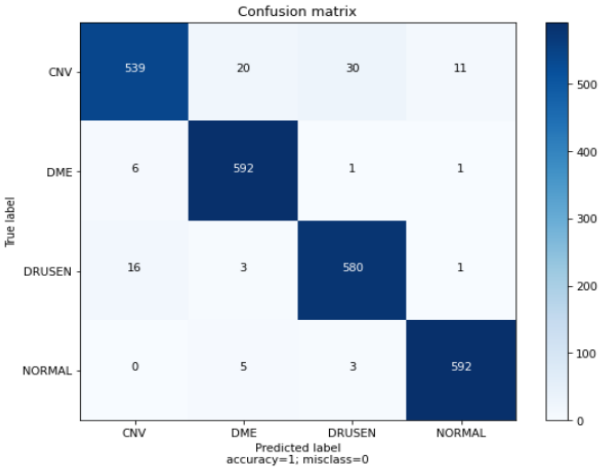



그림 14. Confusion maxtrix result

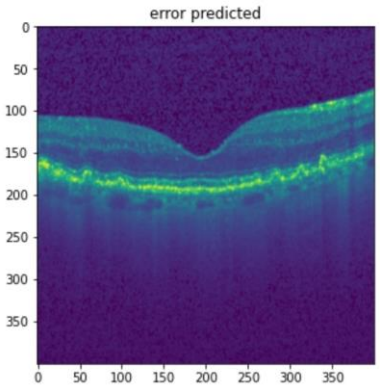



그림 15. DRUSEN 과 식별이 안되는 CNV 데이터

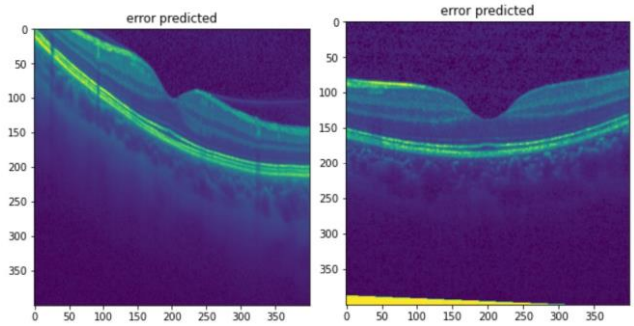



그림 16. 이미지 왜곡, 회전, 잘림 현상

## 4. 결론

본 논문에서는 CNN 의 층을 깊게 쌓음에 따라 발생하는 문제인 기울기 소실의 문제를 해결하기 위해 잔여 블록을 사용한 DNN 인 ResNet18 모델을 구축하였다. 일반적인 CNN 은 기울기 소실의 문제로 인해 더 깊은 층을 쌓는 것이 불가하나 본 연구에서 사용된 ResNet 모델은 향후 레이어를 더 추가하여 정확도를 향상시킬 가능성을 제시한다.

학습률의 경우, 60 에폭, 7 step 마다 0.1 배씩 학습률을 감소시키는 조건에서 학습률이 0.1 인 경우는 학습률이 지나치게 크므로 입력층으로 갈수록 전파되는 오차가 되려 폭증하는 현상이 발생한다. 이로 인해 정확도가 떨어지는 결과를 보였다. 최적의 에폭을 찾기 위해서는 경험적으로 실험하여 극솟값을 찾아내야 함을 알 수 있었다.

데이터의 경우, 83,484 개의 데이터를 1/3 로 줄여 제한된 기간 내에 더 많은 하이퍼파라미터를 조절할 수 있도록 하였다. 데이터를 1/3 로 줄이는 경우 테스트 정확도가 학습정확도와 검증 정확도보다 10% 가량 낮게 나타나는 것에 착안하여 데이터의 분배에 문제가 있는지 고려하였다. 데이터를 일정한 정수 비율로 떨어지게 재분배함으로써 테스트 정확도를 15% 가량 높일 수 있었다. 이는 데이터의 편향 또한 모델의 성능을 좌우하는데 영향을 줄 수 있음을 시사한다.

에폭의 경우, 일정 에폭 이상부터는 학습 정확도가 크게 증가하지 않고, 검증 오차 또한 크게 감소하지 않는 구간을 찾을 때까지 에폭을 늘려가면서 모델을 학습시켜야 한다.

배치 사이즈의 경우, 배치 사이즈 변화에 따른 정확도에서는 큰 차이가 나타나지 않았으며, 단지 배치 사이즈가 클수록 전체적으로 학습 시간을 단축시킬 수 있었다. 다른 조건이 모두 동일한 경우, 배치 사이즈가 지나치게 작아 계산하는 효율을 감소시키지 않는 선에서 crop size 를 최대로 키우는 것이 정확도가 가장 높게 나타나는 결과를 보였다.

결론적으로 모델의 성능을 향상시키기 위해서는 하이퍼파라미터를 경험적으로 조절해야 하며 각각의 하이퍼파라미터 사이의 관계를 파악해야한다. 또한 주어진 GPU 환경과 기간 내에 최적의 결과를 도출할 수 있는 방향으로 연구를 진행해야 함을 알 수 있었다. 최종적인 코드와 결과는 깃허브에 업로드하였다[10].

향후 과제로는 CNN 의 layer 를 더 깊게하여 정확도를 높이고, 황반변성의 다양한 병번 분류뿐만 아니라 추가징후 등을 검출 할 수 있는 시스템을 개발할 계획이다. 이로써 향후 본 연구를 통해 망막병증의 총괄적인 진단이 가능할 것으로 생각된다.

## 감사의 글

많은 도움을 주신 이현우 조교님과 지도편달해 주신 양세정 교수님께 감사드립니다.

## 참고문헌

[1] 표세욱, et al. "치의학 분야에 대한 광간섭 단층영상기기 (optical coherence tomography) 의 적용 가능성 고찰. " *대한치과보철학회지* 55.1 (2017): 100-110.

[2] Yankui Sun, Haoran Zhang, and Xianlin Yao "Automatic diagnosis of macular diseases from OCT volume based on its two-dimensional feature map and convolutional neural network with attention mechanism," Journal of Biomedical Optics 25(9), 096004 (16 September 2020).

[3] L. Huang, X. He, L. Fang, H. Rabbani and X. Chen, "Automatic Classification of Retinal Optical Coherence Tomography Images With Layer Guided Convolutional Neural Network," in IEEE Signal Processing Letters, vol. 26, no. 7, pp. 1026-1030, July 2019, doi: 10.1109/LSP.2019.2917779.

[4] 이희도, 김종수, 권윤형, 김윤경. (2019). 딥러닝 기법을 이용한 미숙아 망막병증 분류 시스템. 한국정보기술학회논문지, 17(10), 17-24.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778

[6] Pham, Hieu, et al. "Meta pseudo labels." arXiv preprint arXiv:2003.10580 (2020).

[7] https://www.kakaobrain.com/blog/113

[8] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *International conference on machine learning*. PMLR, 2015.

[9] https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html

[10] https://github.com/Hyeseong0317/Pytorch_Capstone_Design/blob/main/Capstone_%EB%81%9D.ipynb