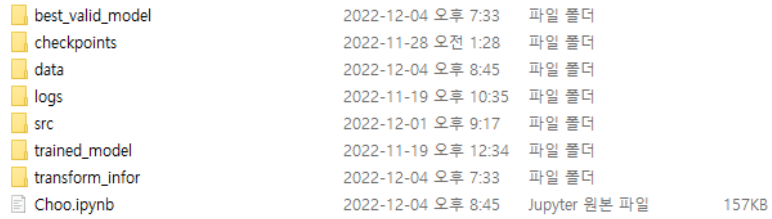
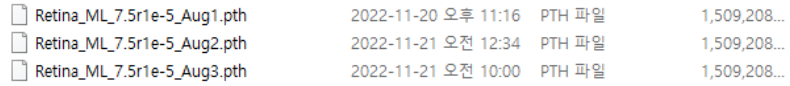
0. 폴더 설명



(1) checkpoints : epoch마다 모델 저장

-checkpoints 폴더 내부 구성



(2) data : 학습에 필요한 이미지 데이터를 모아놓은 폴더

-data 폴더 내부 구성

-Retina_Some_binary 폴더 내부 구성

best_valid_model : baseline으로 생성한 모델 중 가장 성능 높은 모델을 저장하는 폴더

logs : epoch마다 학습된 train, validation 및최종 test의 accuracy와loss정보가 저장된 폴더

src: 학습에 필요한 model, engines, utils 등의 구성 요소가 포함된 폴더

trained_model : 사용하지 않음

transform_infor : epoch마다 수행한 transform, learning rate 정보를 메모장 파일로 저장한 폴더

1. 코드 사용 설명

▾ 0. 학습 세팅

```
!pip install torchmetrics
!pip install optuna
!pip install foolbox
```

Colab 환경 기준으로 위의 모듈을 install 해야 합니다.

1) baseline

(1) 드라이브 경로 설정 및 라이브러리 설치

```
from google.colab import drive
drive.mount('/content/drive')

#==============구글 드라이브 경로 지정====================================
%cd /content/drive/MyDrive/RetinaProject_final/RetinaProject
#=========================================================================

!pip install torchmetrics
!pip install optuna
!pip install foolbox
```

Colab으로 코드를 실행하는 경우, 이곳에서 구글 드라이브 경로를 지정합니다.

(2) 이미지 경로 설정

```
#=============================================================================================
# 2) 이미지 데이터셋 불러오기 + 이미지 변환
#=============================================================================================
train_path = f'C:\\Users\\Bang\\JupyterProjects\\RetinaProject\\DeepLearningPytorchExample\\data\\{args.use_data}\\train'
test_path = f'C:\\Users\\Bang\\JupyterProjects\\RetinaProject\\DeepLearningPytorchExample\\data\\{args.use_data}\\test'
```

train_path와 test_path에서 이미지 폴더 경로를 설정합니다.
data/Retina_Some_binary data 폴더 내에 존재하는 train폴더와 test폴더를 각각 경로로 지정하면 됩니다.

(3) 최적 valid lossmodel 저장 경로

```
# 최적 loss인 model 저장
global best_valid_loss
if val_summary['loss'] < best_valid_loss:
    best_valid_loss = val_summary['loss']
    #————————————————————————————————————————————————————————————
    torch.save(model.state_dict(), f'C:\\Users\\Bang\\JupyterProjects\\RetinaProject\\DeepLearningPytorchExample\\best_valid_model\\{args.title}_{grid_count}.pt')
    save_best_param(args.transform_dir, train_transforms, test_transforms, args.lr, args.title)
torch.save(model.state_dict(), f'C:\\Users\\Bang\\JupyterProjects\\RetinaProject\\DeepLearningPytorchExample\\best_valid_model\\{args.title}.pt')
    #————————————————————————————————————————————————————————————
```

최적의 valid loss를 가지는 model을 저장할 폴더 경로를 지정합니다.
if문 내에서 경로 지정하는 부분은, 전체 GridSearch에서 가장 valid loss 성능이 가장 우수했던 모델을 저장합니다. if문 밖에서 경로 지정하는 부분은 전체 각 GridSearch 수행에서valid loss성능이 가장 우수했던 모델을 저장합니다.
best_valid_model/baseline/{args.load_title}_{grid_count}.pt 로 경로를 지정하면 됩니다.

(4) 학습된 최적 valid loss model의 경로

```
#=============================================================================================
# 6. Model Test
#=============================================================================================
# 학습된 모델 불러오기
model.load_state_dict(torch.load(f'C:\\Users\\Bang\\JupyterProjects\\RetinaProject\\DeepLearningPytorchExample\\best_valid_model\\{args.title}.pt'))
```

최적의 valid loss를 가지는 model을 불러오기 위해 저장된 파일 경로를 지정합니다.
위에서 각 GridSearch 수행에서 학습된 모델의 test 결과를 추출하기 위해 위의 (3)번 과정에서 if 문 밖에서 경로를 지정하는 코드와 같은 경로로 설정합니다.
best_valid_model/baseline/{args.load_title}_{grid_count}.pt 로 경로를 지정하면 됩니다.

2) DCGAN

```
In [ ]: #=================구글 드라이브 경로 지정=================================
        %cd /content/drive/MyDrive/ColabNotebooks/team_project
        #=========================================================================
```

```
/content/drive/MyDrive/ColabNotebooks/team_project
```

코랩으로 사용하신다면 이곳에 구글 드라이브 경로를 지정해 주세요

```
# 데이터셋의 경로

#======================원본 이미지 저장 경로=====================
dataroot = "GAN_data/Retina_Some_binary/train/DR"
#================================================================
```

훈련에 사용한 원본 이미지가 저장되어 있는 경로입니다. data/Retina_Some_binary data 폴더의 train 폴더 경로를 지정해 주세요.

```
done = epoch * len(dataloader) + i

if done % sample_interval == 0:

    #============================이미지 저장 경로 ====================================
    save_image(fake.data[1], f"G_DR/{done+1}.png", nrow=1, normalize=True)
    #=================================================================================
```

DCGAN으로 생성한 가짜 망막 이미지가 저장되는 경로와 저장되는 파일 이름입니다.

2_1)GAN

```
In [2]: #====================구글 드라이브 경로 설정=======================
        %cd /content/drive/MyDrive/ColabNotebooks/team_project
        #================================================================
```

```
/content/drive/MyDrive/ColabNotebooks/team_project
```

코랩으로 사용하신다면 이곳에 구글 드라이브 경로를 지정해 주세요.

```
#====================================훈련 데이터 저장 경로=============================================
train_dataset = datasets.ImageFolder(root="GAN_data/Retina_Some_binary/train/DR", transform=transforms_train)
#=====================================================================================================

dataloader = torch.utils.data.DataLoader(train_dataset, batch_size=10, shuffle=True, num_workers=4)
```

```
/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloader.py:566: UserWarning: This DataLoader will create 4 worker pro
```

훈련에 사용한 원본 이미지가 저장되어 있는 경로입니다. data/Retina_Some_binary data 폴더의 train 폴더 경로를 지정해 주세요.

```
        # 판별자(discriminator) 업데이트
        d_loss.backward()
        optimizer_D.step()

        done = epoch * len(dataloader) + i
        if done % sample_interval == 0:
            #=================================생성된 이미지 저장 경로=============================
            save_image(generated_imgs.data[1], f"generated_DR_images/{done}.png", nrow=1, normalize=True)
            #=====================================================================================

    # 하나의 epoch이 끝날 때마다 로그(log) 출력
    print(f"[Epoch {epoch}/{n_epochs}] [D loss: {d_loss.item():.6f}] [G loss: {g_loss.item():.6f}] [Elapsed time: {time.time() - star
```

```
[Epoch 0/100] [D loss: 0.551939] [G loss: 0.499678] [Elapsed time: 442.28s]
[Epoch 1/100] [D loss: 0.595386] [G loss: 1.416674] [Elapsed time: 477.16s]
[Epoch 2/100] [D loss: 0.565664] [G loss: 0.548363] [Elapsed time: 512.08s]
```

GAN으로 생성한 가짜 망막 이미지가 저장되는 경로와 저장되는 파일 이름입니다.

3. NoisyStudent_crawling

```
# Move to the project directory

#====================구글 드라이브 경로=================================================
%cd /content/drive/MyDrive/RetinaProject_final/RetinaProject
#=======================================================================================
```

Colab 으로 사용하신다면 이곳에 구글 드라이브 경로를 지정해 주세요.

```
# Jupyter 환경
args = easydict.EasyDict({
        "title" : "Retina_ML_7.5r1e-5_Aug3_NoiseStudent",
        #==============baseline 최고성능 모델 title=====================
        "load_model" : "Retina_ML_7.5r1e-5_Aug3",
        #================================================================
        "device" : "cuda",
        "root" : "data",
        "batch_size" : 32, # !!!
        "num_workers" : 2,
        "epochs" : 100, # !!!
```

best_valid_model/baseline 폴더에 저장된 모델의 이름을 작성해 주세요

```
#================================원본 망막 데이터 경로=================================================
train_path = "/content/drive/MyDrive/RetinaProject_final/RetinaProject/data/Retina_Some_binary/train"
test_path = "/content/drive/MyDrive/RetinaProject_final/RetinaProject/data/Retina_Some_binary/test"
#=======================================================================================================
```

원본 망막 데이터를 불러오는 경로입니다. data/Retina_Some_binary data 폴더 내에 존재하는 train폴더와 test폴더를 각각 경로로 지정하면 됩니다.

```
print( len(test_dataset) , len(test_dataset))

#=======================================================================================================

#=====================================크롤링 데이터 저장 경로=============================
stu_path = "/content/drive/MyDrive/RetinaProject_final/RetinaProject/data/StudentData"
#=======================================================================================================

stu_dataset1 = torchvision.datasets.ImageFolder(
```

크롤링 한 데이터를 불러오는 경로입니다. data/crawling data폴더로 경로를 지정하면 됩니다.

+) 학생 모델 -> 선생 모델 이동

```
#=======================================================================================

#=================================================학습된 baseline 모델 불러오기=========================================
model.load_state_dict(torch.load(f"/content/drive/MyDrive/RetinaProject_final/RetinaProject/best_valid_model/{args.load_model}.pt"))
#=======================================================================================================================

model.eval()
```

baseline으로 학습된 모델을 불러오는 경로입니다.

이곳에 best_valid_model/baseline/{args.load_model}.pt 로 경로를 지정하면 됩니다.

```
    # 최적 loss인 model 저장
    if val_summary['loss'] < best_valid_loss:
        best_valid_loss = val_summary['loss']
        #==================================================학습 모델 저장 경로==========================================================
        torch.save(model.state_dict(), f"/content/drive/MyDrive/RetinaProject_final/RetinaProject/NoiseStudent_model/{args.title}.pt")
        #=============================================================================================================================

    # save model
```

학습된 모델을 저장하는 경로입니다.

best_valid_model/NoisyStudent_crawling/{args.title}.pt 로 경로를 지정하면 됩니다.

```
# 학습된 모델 불러오기

#==============================================학습된 모델 저장 경로==========================================================
model.load_state_dict(torch.load(f"/content/drive/MyDrive/RetinaProject_final/RetinaProject/NoiseStudent_model/{args.title}.pt"))
#=============================================================================================================================
```

저장된 모델을 평가하기 위해 불러오는 경로입니다.

best_valid_model/NoisyStudent_crawling/{args.title}.pt 로 경로를 지정하면 됩니다.

4. NoisyStudent_DCGAN

```
# Mount goole drive
drive.mount('/content/drive')

# Move to the project directory

#======================구글 드라이브 경로 지정=================================
%cd /content/drive/MyDrive/RetinaProject_final/RetinaProject
#==============================================================================
```

Colab을 사용하신다면 이곳에 구글 드라이브 경로를 지정해 주세요.

```
# Jupyter 환경
args = easydict.EasyDict({
        "title" : "Retina_ML_7.5r1e-5_Aug3_NoiseStudent_GAN",
        #===============baseline 최고 성능 모델명=========================
        "load_model" : "Retina_ML_7.5r1e-5_Aug3",
        #=================================================================
        "device" : "cuda",
        "root" : "data",
        "batch_size" : 32, # !!!
        "num_workers" : 2,
        "epochs" : 100, # !!!
```

이곳에 best_valid_model/baseline 폴더에 저장된 모델의 이름을 작성해 주세요

```
#============================원본 망막 데이터 저장 경로=================================================================
train_path = "/content/drive/MyDrive/RetinaProject_final/RetinaProject/data/Retina_Some_binary/train"
test_path = "/content/drive/MyDrive/RetinaProject_final/RetinaProject/data/Retina_Some_binary/test"
#=======================================================================================================================

train_dataset = torchvision.datasets.ImageFolder(
    train_path,
    transform = train_transforms
)
```

원본 망막 데이터를 불러오는 경로입니다. data/Retina_Some_binary data 폴더 내에 존재하는 train폴더와 test폴더를 각각 경로로 지정하면 됩니다.

```
#================================GAN 데이터 저장 경로=================================================
stu_path_GAN = "/content/drive/MyDrive/RetinaProject_final/RetinaProject/data/StudentData_GAN"
#=====================================================================================================
```

DCGAN으로 생성된 데이터가 저장된 경로입니다. data/DCGAN data 로 경로를 지정해 주시면 됩니다.

```
#=============================================================================

#=============================================baseline 최고 성능 모델 저장 경로=================================
model.load_state_dict(torch.load(f"/content/drive/MyDrive/RetinaProject_final/RetinaProject/best_valid_model/{args.load_model}.pt"))
#=======================================================================================================================
model.eval()

train_dataset_stu_GAN = copy.deepcopy(train_dataset)
one = 0
zero = 0
for i in range(len(stu_dataset_GAN)):
```

baseline으로 학습된 모델을 불러오는 경로입니다.

이곳에 best_valid_model/baseline/{args.load_model}.pt 로 경로를 지정하면 됩니다.

```
    val_logger.add_scalar('Accuracy', val_summary['metric'], epoch + 1)

    # 최적 loss인 model 저장
    if val_summary['loss'] < best_valid_loss:
        best_valid_loss = val_summary['loss']
        #=============================================학습된 모델 저장 경로=================================
        torch.save(model.state_dict(), f"/content/drive/MyDrive/RetinaProject_final/RetinaProject/NoiseStudent_GAN_model/{args.title}.pt")
        #=======================================================================================================
        
    # save model
```

학습된 모델을 저장하는 경로입니다.

best_valid_model/NoisyStudent_DCGAN/{args.title}.pt 로 경로를 지정하면 됩니다.

```
#=============================================학습된 모델 저장 경로=================================
model.load_state_dict(torch.load(f"/content/drive/MyDrive/RetinaProject_final/RetinaProject/NoiseStudent_GAN_model/{args.title}.pt"))
#=======================================================================================================

# 모델 성능 측정
test_summary = evaluate(test_loader, model, loss_fn, metric_fn, args.device)

# write log
test_logger.add_scalar('Loss', test_summary['loss'], epoch + 1)
```

저장된 모델을 평가하기 위해 불러오는 경로입니다.

best_valid_model/NoisyStudent_DCGAN/{args.title}.pt 로 경로를 지정하면 됩니다.