Assignment

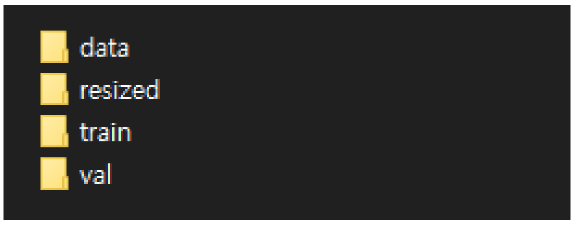
허재호

데이터 전처리 코드

```
import json
import os
import cv2
from PIL import Image
import glob
from tqdm import tqdm
import shutil
from sklearn.model_selection import train_test_split

def bbox_crop_images(json_data_path, images_path) :
    with open(json_data_path, 'r', encoding='utf-8') as j :
        json_data = json.load(j)

    files = os.listdir(images_path)
    for file in tqdm(files) :
        json_infos = json_data[file]
        annotations = json_infos['anno']
        for i, annots in enumerate(annotations) :
            label = annots['label']
            bbox = annots['bbox']
            os.makedirs(os.path.join("./data/metal_damged", 'data', label),
exist_ok=True)
            os.makedirs(os.path.join("./data/metal_damged", 'train', label),
exist_ok=True)
            os.makedirs(os.path.join("./data/metal_damged", 'val', label),
exist_ok=True)
            x, y, w, h = bbox
            img = cv2.imread(os.path.join(images_path, file))
            cropped_img = img[y:y+h, x:x+w]
            file_name = file.split(".")[0]
            new_file_path = os.path.join("./data/metal_damaged", "data",
label, f"{file_name}_cropped_{str(i).zfill(1)}.png")
            print(new_file_path)
            cv2.imwrite(new_file_path, cropped_img)

def expend2square(pil_image, background_color):
    width, height = pil_image.size
    if width == height:
        return pil_image
    elif width > height :
        result = Image.new(pil_image.mode, (width, width), background_color)
        result.paste(pil_image, (0, (width-height) // 2))
        return result
    else:
        result = Image.new(pil_image.mode, (height, height), background_color)
        result.paste(pil_image, ((height-width) // 2, 0))
        return result

def resize_with_padding(pil_image, new_size, background_color):
    img = expend2square(pil_image, background_color)
    img = img.resize((new_size[0], new_size[1]), Image.ANTIALIAS)
    return img

def resize_images(data_path) :
    img_path_list = glob.glob(os.path.join(data_path, "*","*.png"))
    for img_path in tqdm(img_path_list):
        try:
            dir, file = os.path.split(img_path)
            folder_name = dir.rsplit('\\')[1]
            os.makedirs(os.path.join("./data/metal_damaged/resized",
folder_name), exist_ok=True)
            name = os.path.basename(img_path).rsplit('.png')[0]
            img = Image.open(img_path).convert('RGB')
            img_new = resize_with_padding(img, (256,256), (0,0,0))
            save_file_name =
f"./data/metal_damaged/resized/{folder_name}/{name}.png"
            img_new.save(save_file_name, "png")
        except Exception as ex:
            print(f"Error occurs on : {file} with the reason of {ex}")

def file_split(src, dst) :
    files = glob.glob(os.path.join(src, "*", "*"))
    train_list, val_list = train_test_split(files, test_size=0.1)
    for file in train_list :
        folder_path = file.split("\\")[1]
        shutil.copy2(file, os.path.join(dst, "train", folder_path))
    for file in val_list :
        folder_path = file.split("\\")[1]
        shutil.copy2(file, os.path.join(dst, "val", folder_path))
    print("File Split Completed")

json_data_path = "./data/data/anno/annotation.json"
images_path = "./data/data/images"
bbox_crop_images(json_data_path, images_path)

resized_data_path = "./data/metal_damaged/data"
resize_images(resized_data_path)
```

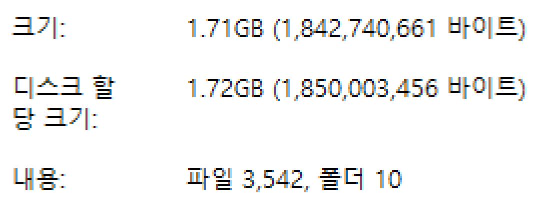
```
src = "./data/metal_damaged/resized"
dst = "./data/metal_damaged/"
file_split(src, dst)
```

데이터 전처리 결과



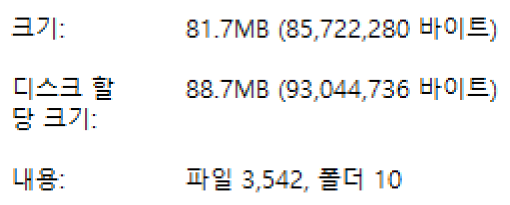
```
data
resized
train
val
```

Figure 1 Outputs



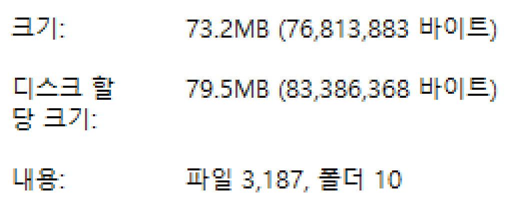
크기:	1.71GB (1,842,740,661 바이트)
디스크 할당 크기:	1.72GB (1,850,003,456 바이트)
내용:	파일 3,542, 폴더 10

Figure 2 Cropped data



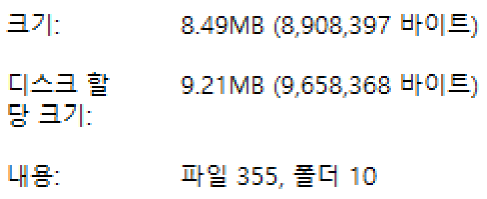
크기:	81.7MB (85,722,280 바이트)
디스크 할당 크기:	88.7MB (93,044,736 바이트)
내용:	파일 3,542, 폴더 10

Figure 3 Resized with padding data



크기:	73.2MB (76,813,883 바이트)
디스크 할당 크기:	79.5MB (83,386,368 바이트)
내용:	파일 3,187, 폴더 10

Figure 4 Train data



크기:	8.49MB (8,908,397 바이트)
디스크 할당 크기:	9.21MB (9,658,368 바이트)
내용:	파일 355, 폴더 10

Figure 5 Test data

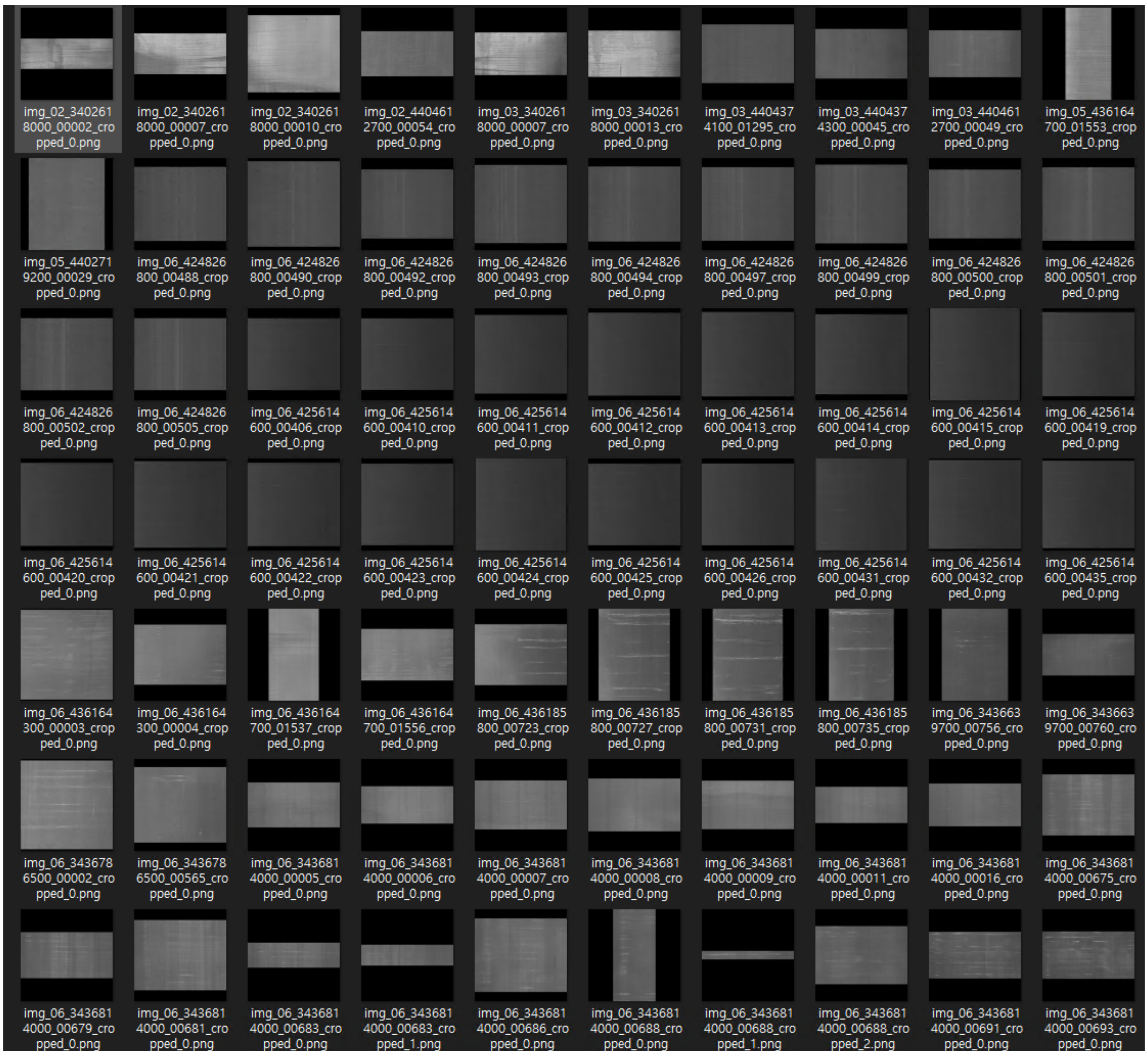

img_02_3402618000_00002_cropped_0.png
img_02_3402618000_00007_cropped_0.png
img_02_3402618000_00010_cropped_0.png
img_02_4404612700_00054_cropped_0.png
img_03_3402618000_00007_cropped_0.png
img_03_3402618000_00013_cropped_0.png
img_03_4404374100_01295_cropped_0.png
img_03_4404374300_00045_cropped_0.png
img_03_4404612700_00049_cropped_0.png
img_05_436164700_01553_cropped_0.png
img_05_4402719200_00029_cropped_0.png
img_06_424826800_00488_cropped_0.png
img_06_424826800_00490_cropped_0.png
img_06_424826800_00492_cropped_0.png
img_06_424826800_00493_cropped_0.png
img_06_424826800_00494_cropped_0.png
img_06_424826800_00497_cropped_0.png
img_06_424826800_00499_cropped_0.png
img_06_424826800_00500_cropped_0.png
img_06_424826800_00501_cropped_0.png
img_06_424826800_00502_cropped_0.png
img_06_424826800_00505_cropped_0.png
img_06_425614600_00406_cropped_0.png
img_06_425614600_00410_cropped_0.png
img_06_425614600_00411_cropped_0.png
img_06_425614600_00412_cropped_0.png
img_06_425614600_00413_cropped_0.png
img_06_425614600_00414_cropped_0.png
img_06_425614600_00415_cropped_0.png
img_06_425614600_00419_cropped_0.png
img_06_425614600_00420_cropped_0.png
img_06_425614600_00421_cropped_0.png
img_06_425614600_00422_cropped_0.png
img_06_425614600_00423_cropped_0.png
img_06_425614600_00424_cropped_0.png
img_06_425614600_00425_cropped_0.png
img_06_425614600_00426_cropped_0.png
img_06_425614600_00431_cropped_0.png
img_06_425614600_00432_cropped_0.png
img_06_425614600_00435_cropped_0.png
img_06_436164300_00003_cropped_0.png
img_06_436164300_00004_cropped_0.png
img_06_436164700_01537_cropped_0.png
img_06_436164700_01556_cropped_0.png
img_06_436185800_00723_cropped_0.png
img_06_436185800_00727_cropped_0.png
img_06_436185800_00731_cropped_0.png
img_06_436185800_00735_cropped_0.png
img_06_3436639700_00756_cropped_0.png
img_06_3436639700_00760_cropped_0.png
img_06_3436786500_00002_cropped_0.png
img_06_3436786500_00565_cropped_0.png
img_06_3436814000_00005_cropped_0.png
img_06_3436814000_00006_cropped_0.png
img_06_3436814000_00007_cropped_0.png
img_06_3436814000_00008_cropped_0.png
img_06_3436814000_00009_cropped_0.png
img_06_3436814000_00011_cropped_0.png
img_06_3436814000_00016_cropped_0.png
img_06_3436814000_00675_cropped_0.png
img_06_3436814000_00679_cropped_0.png
img_06_3436814000_00681_cropped_0.png
img_06_3436814000_00683_cropped_0.png
img_06_3436814000_00683_cropped_1.png
img_06_3436814000_00686_cropped_0.png
img_06_3436814000_00688_cropped_0.png
img_06_3436814000_00688_cropped_1.png
img_06_3436814000_00688_cropped_2.png
img_06_3436814000_00691_cropped_0.png
img_06_3436814000_00693_cropped_0.png

Figure 6 Images

학습 결과

```
Epoch [1/20], Train Loss : 1.4055, Train Acc : 0.5723, Val Loss : 0.4810, Val Acc : 0.8507
Epoch [2/20], Train Loss : 0.5005, Train Acc : 0.8431, Val Loss : 0.4443, Val
Acc : 0.8676
Epoch [3/20], Train Loss : 0.3955, Train Acc : 0.8798, Val Loss : 0.3422, Val
Acc : 0.8873
Epoch [4/20], Train Loss : 0.3266, Train Acc : 0.8987, Val Loss : 0.2809, Val
Acc : 0.9070
Epoch [5/20], Train Loss : 0.3325, Train Acc : 0.8968, Val Loss : 0.3390, Val
Acc : 0.8958
Epoch [6/20], Train Loss : 0.2797, Train Acc : 0.9112, Val Loss : 0.3260, Val
Acc : 0.9155
Epoch [7/20], Train Loss : 0.2768, Train Acc : 0.9118, Val Loss : 0.2991, Val
Acc : 0.9127
Epoch [8/20], Train Loss : 0.2678, Train Acc : 0.9156, Val Loss : 0.2359, Val
Acc : 0.9211
Epoch [9/20], Train Loss : 0.2422, Train Acc : 0.9247, Val Loss : 0.2757, Val
Acc : 0.9239
Epoch [10/20], Train Loss : 0.2405, Train Acc : 0.9200, Val Loss : 0.3190, Val
Acc : 0.9183
Epoch [11/20], Train Loss : 0.2332, Train Acc : 0.9216, Val Loss : 0.2994, Val
Acc : 0.9070
Epoch [12/20], Train Loss : 0.2131, Train Acc : 0.9278, Val Loss : 0.4208, Val
Acc : 0.9099
Epoch [13/20], Train Loss : 0.2243, Train Acc : 0.9272, Val Loss : 0.3137, Val
Acc : 0.9155
Epoch [14/20], Train Loss : 0.1893, Train Acc : 0.9426, Val Loss : 0.3080, Val
Acc : 0.9155
Epoch [15/20], Train Loss : 0.1893, Train Acc : 0.9366, Val Loss : 0.2732, Val
Acc : 0.9239
Epoch [16/20], Train Loss : 0.1808, Train Acc : 0.9432, Val Loss : 0.3007, Val
Acc : 0.9042
Epoch [17/20], Train Loss : 0.1699, Train Acc : 0.9435, Val Loss : 0.3201, Val
Acc : 0.8986
Epoch [18/20], Train Loss : 0.1726, Train Acc : 0.9426, Val Loss : 0.2417, Val
Acc : 0.9268
Epoch [19/20], Train Loss : 0.1743, Train Acc : 0.9404, Val Loss : 0.2570, Val
Acc : 0.9155
Epoch [20/20], Train Loss : 0.1649, Train Acc : 0.9463, Val Loss : 0.2431, Val
Acc : 0.9380
```

테스트 결과

```
Test Set : Acc [333/355] 94%
```

힉습에 사용한 하이퍼 파라미터

```
epochs = 20
criterion = CrossEntropyLoss().to(device)
optimizer = Lion(model.parameters(), lr=1e-4, weight_decay=1e-2)
```