

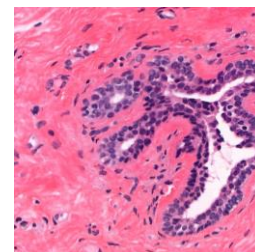
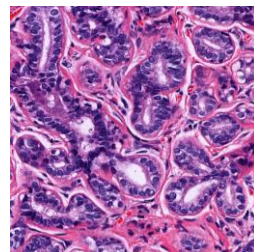
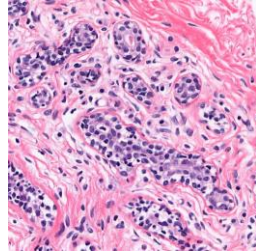
2020 Korea Health Datathon Medical Image

유방 병리 이미지

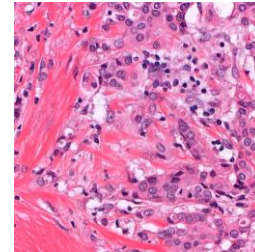
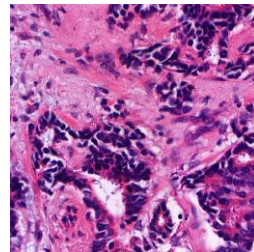
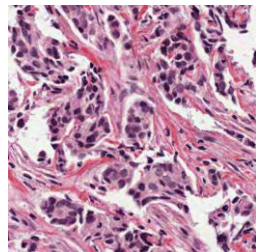
국립암센터

개요

- 목표: 악성/양성 분류 모델 개발
- 샘플 이미지



양성 샘플

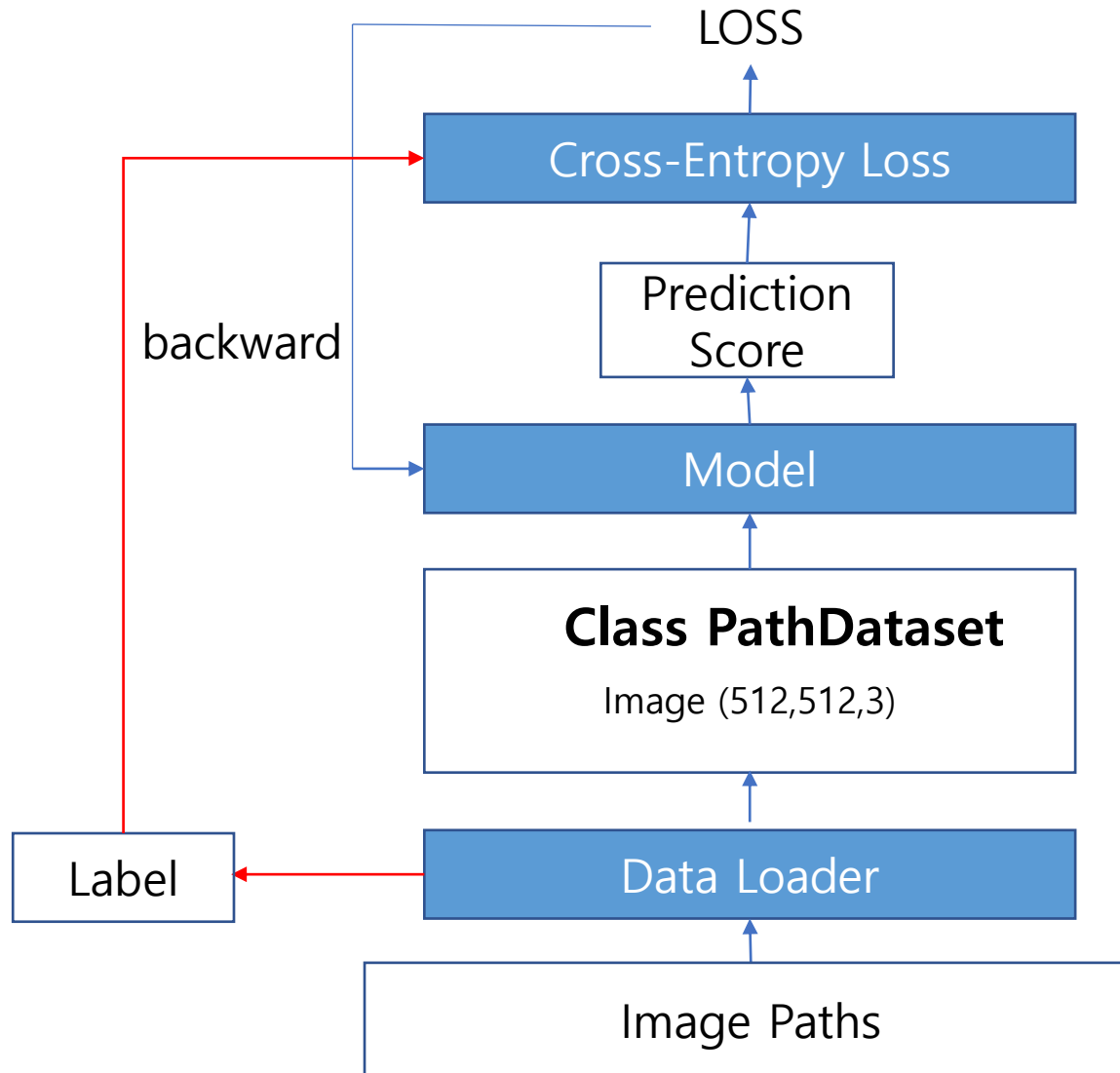


악성 샘플

- 제공 데이터 구성

| 데이터 셋 | 양성 / 악성 |
|-------|---------------|
| 트레이닝 | 5,600 / 2,400 |
| 테스트 | 1,400 / 600 |

기본 코드구조



주요 코드 - 1

- 데이터 로더 (main.py)
 - 이미지 경로를 읽어 이미지를 구성하는 코드
 - Preprocessing을 통해서 이미지의 분류에 대한 성능을 향상 가능

```
class PathDataset(Dataset):
    def __init__(self, image_path, labels=None, test_mode= True):
        self.len = len(image_path)
        self.image_path = image_path
        self.labels = labels
        self.mode = test_mode

    def __getitem__(self, index):
        im = cv2.imread(self.image_path[index])
        im = im.reshape(3, im.shape[0], im.shape[1])

        ### REQUIRED: PREPROCESSING ###

        if self.mode:
            return torch.tensor(im, dtype=torch.float32)
        else:
            return torch.tensor(im, dtype=torch.float32), \
                torch.tensor(self.labels[index] , dtype=torch.long)

    def __len__(self):
        return self.len
```



주요 코드 - 2

- 모델 (arch.py)
 - 효율적인 병리영상 분류를 위한 모델 설계

```
class CNN (nn.Module):
    def __init__(self, ):
        super(CNN, self).__init__()

        self.cnn_layer = nn.Sequential(
            nn.Conv2d(3,6, kernel_size=3, stride=2, padding=1),
            nn.BatchNorm2d(6),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.MaxPool2d(kernel_size=3, stride=2),
            nn.Conv2d(6,12, kernel_size=3, stride=2, padding=1),
            nn.BatchNorm2d(12),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.MaxPool2d(kernel_size=3, stride=2),
            nn.Conv2d(12,15, kernel_size=3, stride=2, padding=1),
            nn.BatchNorm2d(15),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.MaxPool2d(kernel_size=3, stride=2),
        )

        self.fc = nn.Sequential(
            nn.Linear(735, 64),
            nn.BatchNorm1d(64),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(64, 32),
            nn.BatchNorm1d(32),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(32, 2),
            nn.Softmax(dim=1)
        )

    def forward(self, x):
        out = self.cnn_layer(x)
        out = out.reshape(out.size(0), -1)
        out = self.fc(out)
        return out
```



평가 척도

- 6가지 지표의 평균

- Accuracy, Specificity, Sensitivity, Precision, Negative predictable value, F1 score
 - 참고: en.wikipedia.org/wiki/Positive_and_negative_predictive_values

A / B Track 공통사항

- Pre-trained model 사용 가능
- 시간당 최대 1번 commit
- 모델 용량 300MB 이하
- 문의사항 답변 시간 :
9/21 (월) 15:00 – 18:00,
9/22~24 (화~목) 09:00 – 18:00,
9/25 (금) 09:00–17:00