

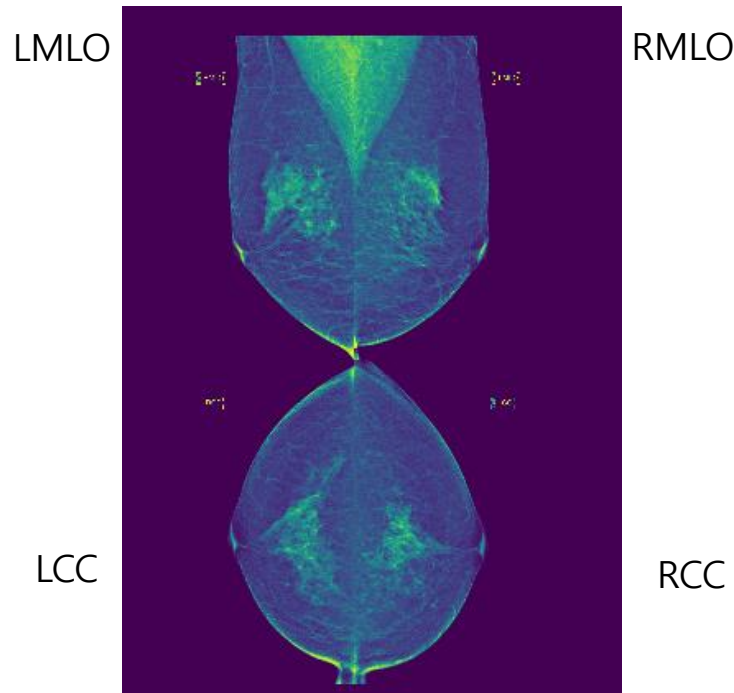
**KHD 2019**

**유방 촬영 영상 데이터**

**국립암센터**

# Overview

- 목표: 유방 촬영 이미지 분류 모델 개발
  - 정상군과 양성, 그리고 악성 종양을 가진 환자군들의 유방 촬영 이미지로 구성되어 있음
  - 본 대회는 악성 종양을 가진 환자군을 분류하는 효율적인 딥러닝 모델 개발이 목표임



# 유방 촬영 이미지 데이터

- 데이터 정의

- 정상 (normal) : 영상에서 이상소견이 없는 경우, 1년 이상이 지난 시점에 촬영된 추적 유방촬영에서 변화 없이 음성으로 판명된 경우
- 양성 (benign) : 영상에서 양성 병변이 확인된 경우, 1년 이상이 지난 시점에 촬영된 추적 유방촬영 영상에서 변화 없이 양성으로 판명된 경우
- 악성 (malignant) : 영상에서 악성 병변이 확인된 경우, 추적조직검사로 유방암 확정

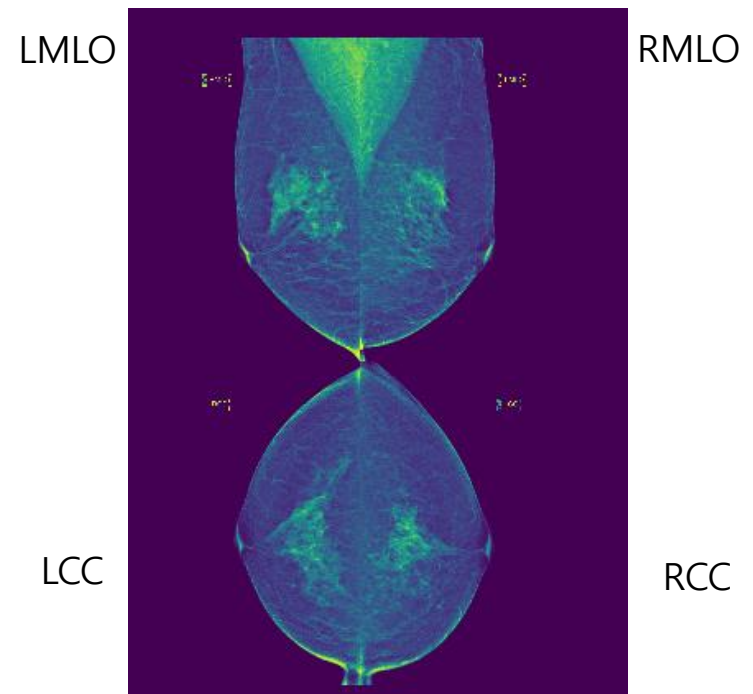
# 유방 촬영 이미지 데이터

- 데이터 형식

- 유방 이미지 : mammo.npz , 카테고리 3개 (benign, malignant, normal)
  - npz format – gzip 압축을 사용하여 배열 데이터를 저장하는 numpy

- 데이터 규모

- 학습용 데이터 (train)
  - 악성 (malignant): 500례 (25%)
  - 양성 (benign): 500례 (25%)
  - 정상 (normal): 1,000례 (50%)
- 검증용 데이터 (test)
  - 악성 (malignant): 50례 (25%)
  - 양성 (benign): 50례 (25%)
  - 정상 (normal): 100례 (50%)

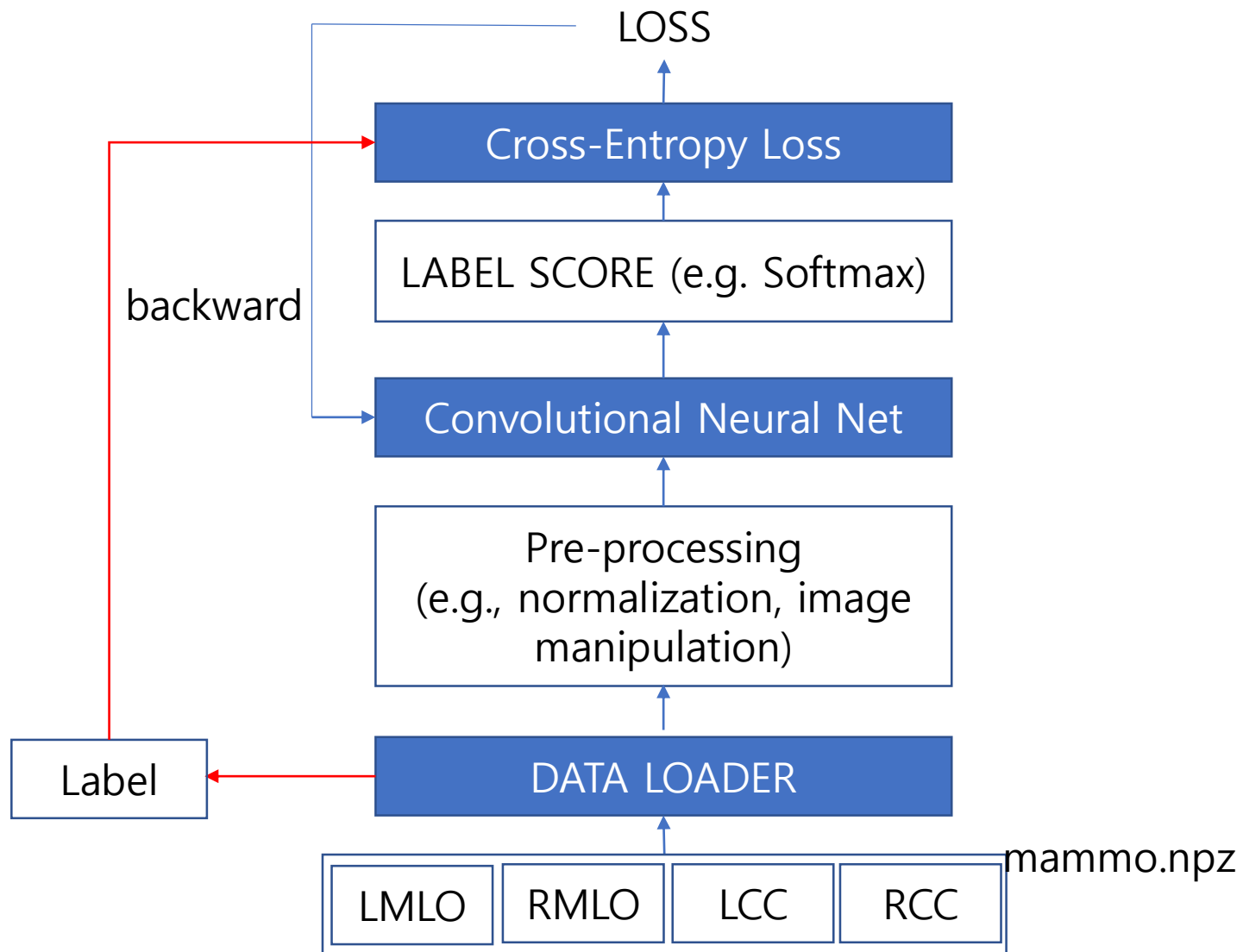


# 유방 촬영 이미지 데이터

## • 데이터 경로

DIRECTORY			FILES	CONTENTS
/train	/malign	/0XXXX	mammo.npz	<ul style="list-style-type: none"><li>- 4장의 이미지를 하나의 파일로 구성</li><li>- compressed numpy (.npz)</li><li>- 원본의 약 10% 사이즈</li><li>- Shape: (4,1,width,height)</li><li>- 순서: LMLO/RMLO/RCC/LCC</li></ul>
	/benign	/1XXXX		
	/normal	/2XXXX		
/test	/malign	/0XXXX		
	/benign	/1XXXX		
	/normal	/2XXXX		

# Baseline system



# Data loader

```
def data_loader (root_path):
    t = time.time()
    print('Data loading...')
    data_path = [] # data path 저장을 위한 변수
    labels=[] # 테스트 id 순서 기록
    ## 하위 데이터 path 읽기
    for dir_name, __ in os.walk(root_path):
        try:
            data_id = dir_name.split('/')[-1]
            int(data_id)
        except: pass
        else:
            data_path.append( dir_name )
            labels.append(int(data_id[0]))
    ## 데이터만 읽기
    data = [] # img저장을 위한 list
    for d_path in data_path:
        sample = np.load(d_path+'/mammo.npz')['arr_0']
        data.append(sample)
        data = np.array(data) ## list to numpy

    print('Dataset Reading Success \n Reading time',time.time()-t,'sec')
    print('Dataset:',data.shape,'np.array.shape(files, views, width, height)')

    return data, labels
```



# 모델: CNN 기반 예제

```
def cnn_basic():
    model = Sequential()
    model.add(Conv2D(filters=8, kernel_size=(2, 2), activation='relu', padding='same',
input_shape=(66,82,1))) ## shape size 정해주기
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.5))
    model.add(Conv2D(16, (2, 2), activation='relu', padding='same'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.5))
    model.add(Flatten())
    model.add(Dense(256, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))
    return model
```



# 성능 평가 척도

- F1 Score

- Precision과 Recall의 조화평균

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- Recall (sensitivity)

- 재현율이란 실제 True인 것 중에서 모델이 True라고 예측

$$Recall = \frac{TP}{TP + FN}$$

- Precision (Positive Predictive Value)

- 정밀도란 모델이 True라고 분류한 것 중에서 실제 True라고 예측

$$Precision = \frac{TP}{TP + FP}$$