

2020 Korea Health Datathon

Medical Image

부비동 영상 데이터



Contents

I . 부비동영상 데이터 정의

II . 부비동영상 데이터 셋

III. 평가기준

A person's hands are holding a tablet over a desk. The desk has papers and a clipboard. The background is dark and blurred.

1

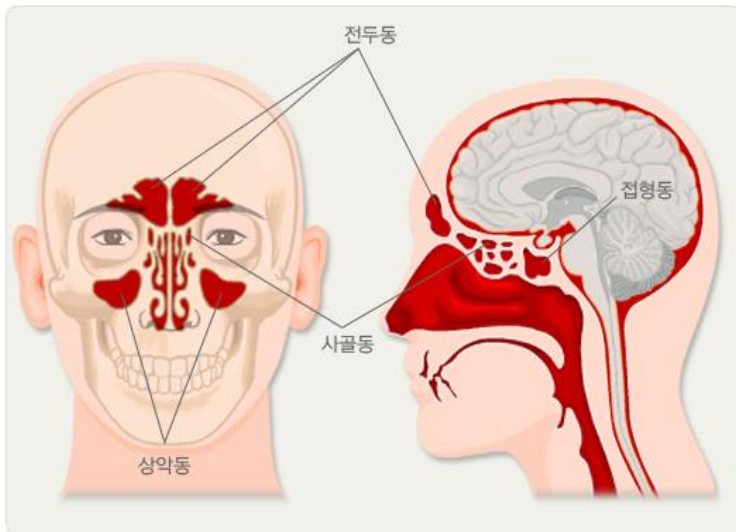
부비동영상 데이터 정의



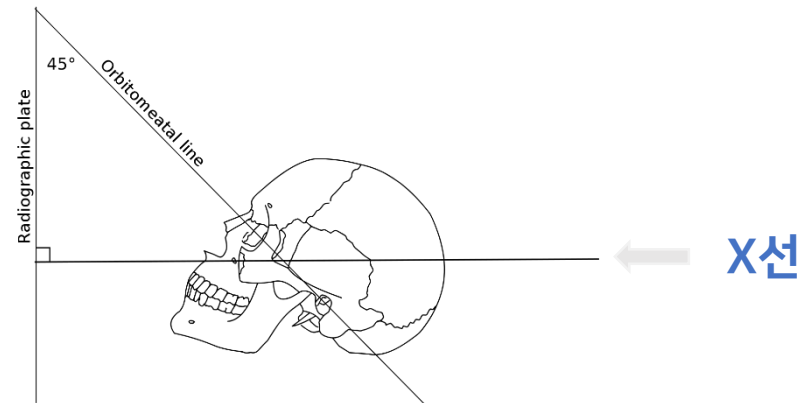
| 부비동 X-ray 영상

부비동 X-ray : 축농증, 감기, 알레르기 등으로 코막힘 등이 있을 때 검사

〈그림. 부비동의 구조와 위치〉

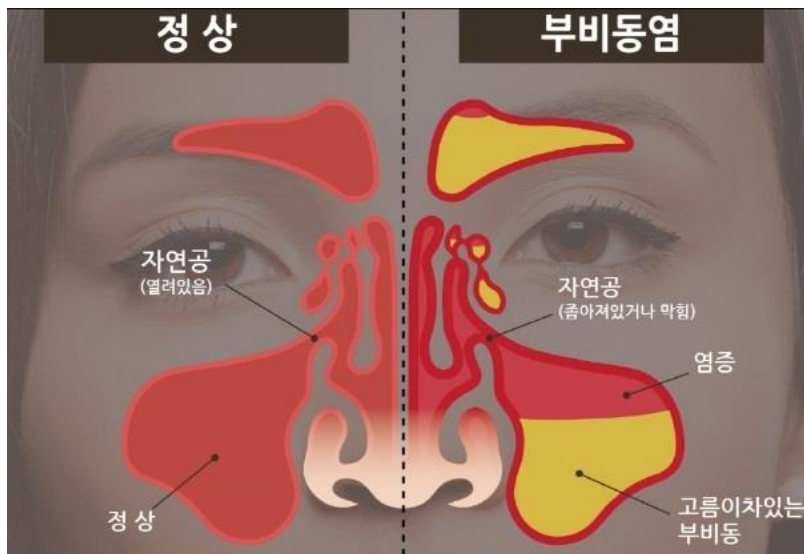


- 안면부의 전면과 측면을 촬영
- 각 부비동의 위치에 따라 촬영 각도 및 방법이 달라짐
- **Waters' view**
 - X선이 궤도 선에 대해 45° 각도를 이루어 조사
 - 상악동을 보기 위한 촬영 방법



| 부비동염

부비동에 세균, 바이러스가 침투하여 염증이 발생한 질환으로, 흔히 '**축농증**'이라 불림



Sinusitis (부비동염)

1) Mucosal thickening

:상악동 내 점막 비후가 2mm 이상인 경우

2) Air fluid

:상악동 내 공기와 액체 음영이 가로로 명확히 보이는 경우

3) Haziness

:상악동 전체에 혼탁화 음영이 확인되는 경우

2

부비동영상 데이터 셋



| 부비동영상



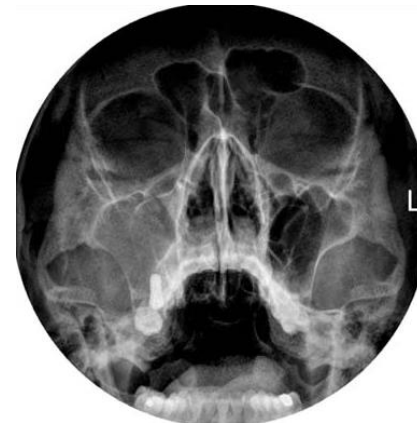
양측 상악동 정상



양측 상악동 Mucosal thickening



양측 상악동 Air fluid



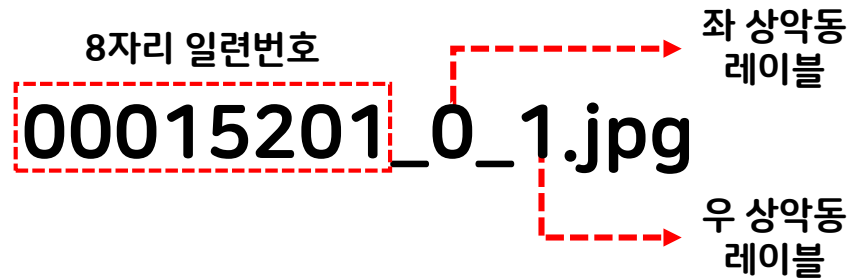
우측 상악동 Haziness

| 부비동영상 데이터 현황

부비동영상 데이터 제공 표 (건양대병원)

질환	Train 수량 (건)	Test 수량 (건)	비율 (%)
정상	2,500	250	78%
Mucosal thickening	360	40	13%
Air fluid	150	20	6%
Haziness	90	10	3%

I 데이터셋 구성



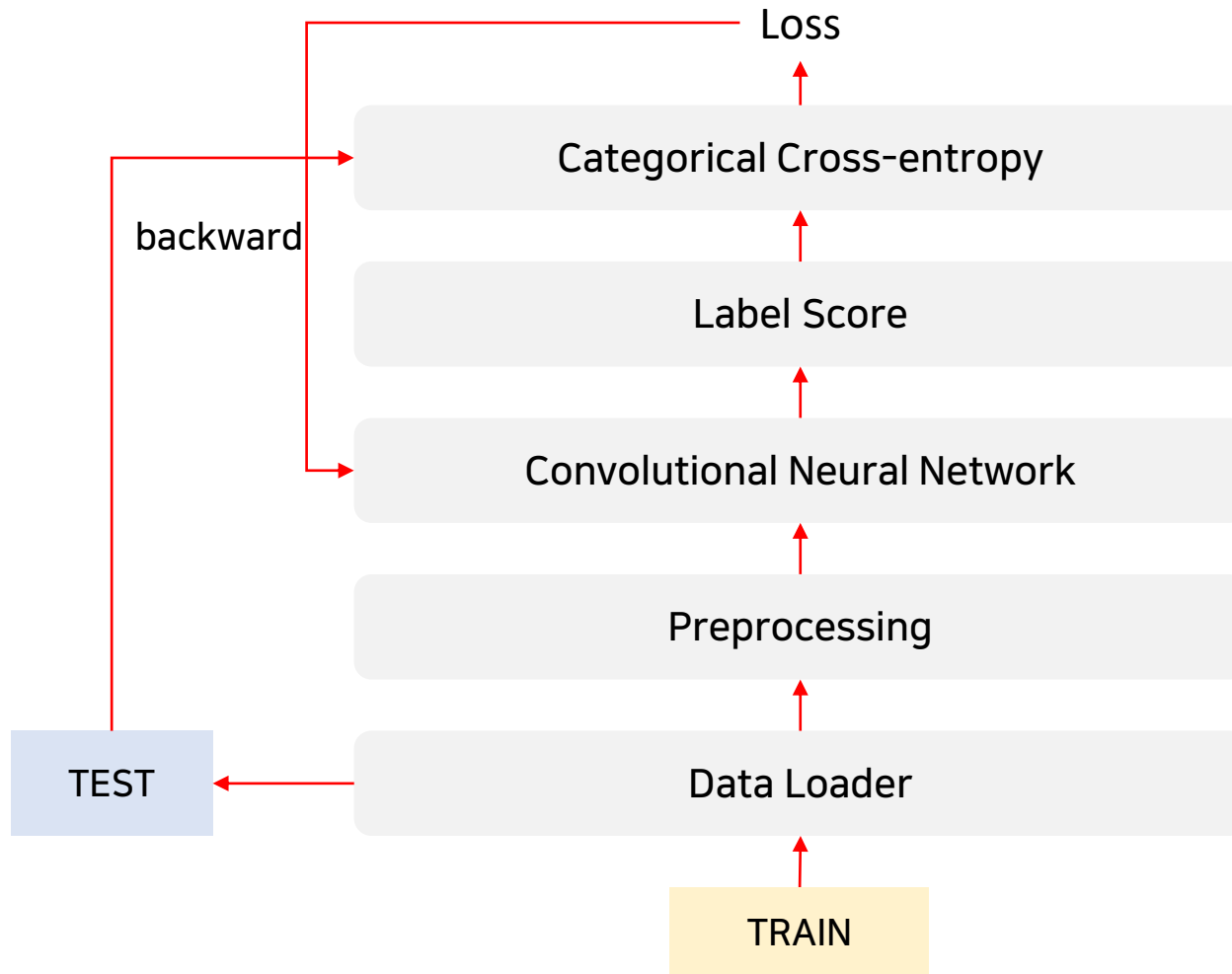
DIRECTORY		CONTENTS
/train	/00000001_0_0.jpg	1) Jpg 확장자 2) 600*600*3 - 원본 영상 비율에 따라 h, w값은 상이함 - 최대값은 600 유지
	/00000002_x_1.jpg	
	/00000003_2_3.jpg	
/test	/00000011_0_0.jpg	
	/00000012_x_1.jpg	
	/00000013_2_3.jpg	
/test_submit	/00000021_0_0.jpg	
	/00000022_x_1.jpg	
	/00000023_2_3.jpg	



3

샘플모델

| 시스템 베이스라인



| CNN 모델 샘플 코드 (keras)

```
def SampleModelKeras(in_shape, num_classes):
```

```
    model = Sequential()
    model.add(Conv2D(filters=64, kernel_size=(3, 3), padding='same', input_shape=in_shape))
    model.add(BatchNormalization(axis=-1))
    model.add(ReLU())
    model.add(Conv2D(filters=64, kernel_size=(3, 3), padding='same'))
    model.add(BatchNormalization(axis=-1))
    model.add(ReLU())
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=128, kernel_size=(3, 3), padding='same'))
    model.add(BatchNormalization(axis=-1))
    model.add(ReLU())
    model.add(Conv2D(filters=128, kernel_size=(3, 3), padding='same'))
    model.add(BatchNormalization(axis=-1))
    model.add(ReLU())
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=256, kernel_size=(3, 3), padding='same'))
    model.add(BatchNormalization(axis=-1))
    model.add(ReLU())
    model.add(Conv2D(filters=256, kernel_size=(3, 3), padding='same'))
    model.add(BatchNormalization(axis=-1))
    model.add(ReLU())

    model.add(Flatten())
    model.add(Dense(256*4*4, activation='relu'))
    model.add(Dense(512, activation='relu'))
    model.add(Dense(64, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))

    return model
```

CNN 모델 샘플 코드 (pytorch)

class SampleModelTorch(nn.Module):

```
def __init__(self, num_classes=4):
    super(SampleModelTorch, self).__init__()

    self.layer1 = nn.Sequential(nn.Conv2d(1, 64, kernel_size=3, padding=2), nn.BatchNorm2d(64), nn.ReLU(),
                                nn.Conv2d(64, 64, kernel_size=3, padding=2), nn.BatchNorm2d(64), nn.ReLU(),
                                nn.MaxPool2d(kernel_size=2, stride=2))

    self.layer2 = nn.Sequential(nn.Conv2d(64, 128, kernel_size=3, padding=2), nn.BatchNorm2d(128), nn.ReLU(),
                                nn.Conv2d(128, 128, kernel_size=3, padding=2), nn.BatchNorm2d(128), nn.ReLU(),
                                nn.MaxPool2d(kernel_size=2, stride=2))

    self.layer3 = nn.Sequential(nn.Conv2d(128, 256, kernel_size=3, padding=2), nn.BatchNorm2d(256), nn.ReLU(),
                                nn.Conv2d(256, 256, kernel_size=3, padding=2), nn.BatchNorm2d(256), nn.ReLU())

    self.fc = nn.Sequential(nn.Linear(256 * 4 * 4, 512), nn.Linear(512, 64), nn.Linear(64, num_classes))

def forward(self, x):
    x = self.layer1(x)
    x = self.layer2(x)
    x = self.layer3(x)
    x = x.reshape(x.size(0), -1)
    x = self.fc(x)
    return x
```



4

평가기준

| NSML Leaderboard

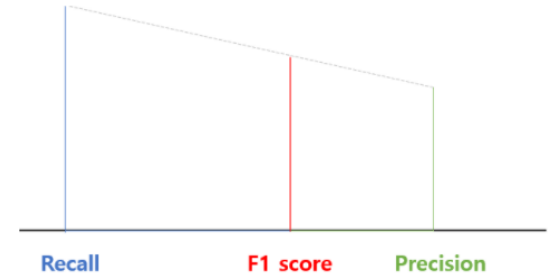
F1 score : recall(재현율)과 precision(정밀도)의 조화 평균.

Recall : true라고 분류된 것 중 실제 true인 비율

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

Precision : 실제 true 중 true로 분류된 비율

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



$$F1\ score = \frac{2}{recall^{-1} + precision^{-1}} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

Weighted F1 score

Leaderboard score

$$= \frac{(1 \times F1\ score_0) + (2 \times F1\ score_1) + (3 \times F1\ score_2) + (4 \times F1\ score_3)}{10}$$

* $F1\ score_n$: F1 score for class n

| NSML Leaderboard

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



Than **KY**ou!