건양대학교병원
KONYANG UNIV. HOSPITAL

KHD 2019

안저 촬영 영상 데이터

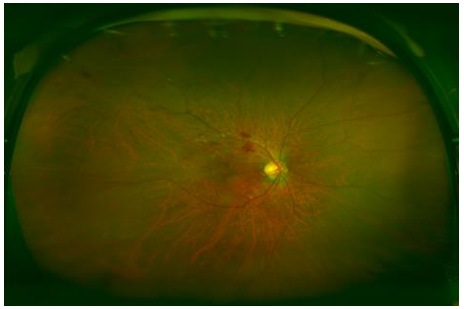
Contents

1

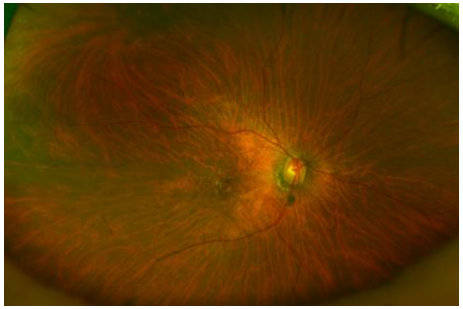
KHD 2019 개요

Overview

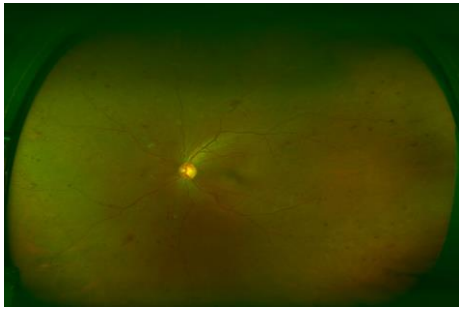
- 목표 : 안저 촬영영상 분류 모델 개발
  - 정상, 황반변성(AMD), 당뇨성망막병증(DMR), 망막정맥폐쇄(RVO) 로 구성
  - 4 multi class classification 을 목표로 함



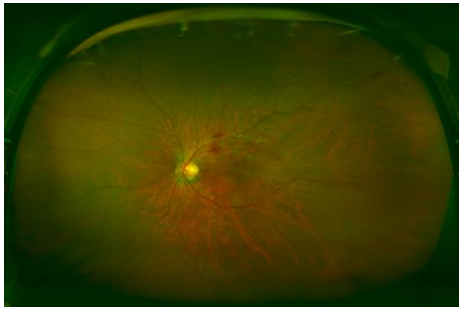
정상 광각 안저 영상



황반변성 광각 안저 영상(AMD)



당뇨망막병증 광각 안저 영상(DMR)



망막정맥폐쇄 광각 안저 영상(RVO)


WHO
WHAT
WHY
CONCEPT GENERATION
CONCEPT SCREENING
ITTERATIVE REQUIREMENTS IMPROVEMENT
MARKET RESEARCH
EXPENDITURES
YEAR

2

안저영상 데이터 셋

데이터셋 구성

DIRECTORY			CONTENTS
/train	/NORMAL	/Wide_NOR_xxx_R, …	- .Jpg 확장자의 이미지 - 3900×3072×3
	/AMD	/Wide_AMD_xxx_L, …	
	/RVO	/Wide_RVO_xxx_L, …	
	/DMR	/Wide_DMR_xxx_R, …	
/test	/NORMAL	/Wide_NOR_xxx_R, …	
	/AMD	/Wide_AMD_xxx_R, …	
	/RVO	/Wide_RVO_xxx_L, …	
	/DMR	/Wide_DMR_xxx_L, …	
/test_submit	/NORMAL	/Wide_NOR_xxx_L, …	
	/AMD	/Wide_AMD_xxx_L, …	
	/RVO	/Wide_RVO_xxx_R, …	
	/DMR	/Wide_DMR_xxx_R, …	

안저영상 질환 별 데이터 현황

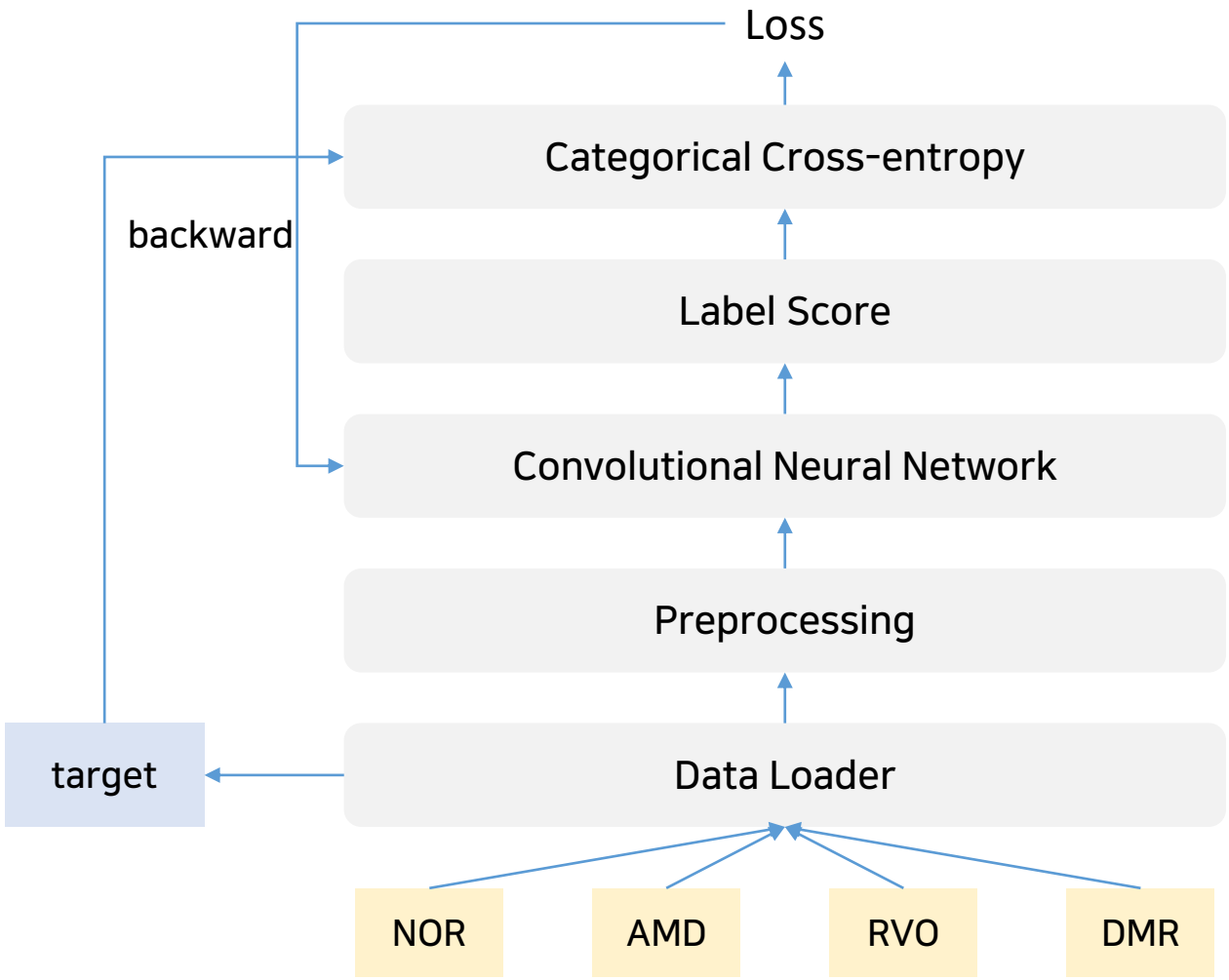
안저영상 데이터 제공 표 (2018-2019 건양대병원)

질환	Train 수량 (장)	Test 수량 (장)	비율 (%)
정상(Normal)	997	111	26
황반변성(AMD)	900	75	23
당뇨망막병증(DMR)	1,511	167	40
망막정맥폐쇄(RVO)	436	48	11

3

샘플모델

시스템 베이스라인


Loss
Categorical Cross-entropy
backward
Label Score
Convolutional Neural Network
Preprocessing
target
Data Loader
NOR
AMD
RVO
DMR

CNN 모델 샘플 코드

```python
def dataset_loader(img_path, rescale=RESCALE, resize_factor=RESIZE):

    t1 = time.time()
    print('Loading training data...₩n')

    p_list = [os.path.join(dirpath, f) for dirpath, dirnames, files in os.walk(img_path) for f in files if all(s in f for s in ['.jpg'])]
    p_list.sort()

    images = []
    labels = []
    for i, p in enumerate(p_list):
        im = cv2.imread(p, 3)
        if not (resize_factor == 1.):
            im = image_preprocessing(im, rescale=rescale, resize_factor=resize_factor)
        images.append(im)

        l = Label2Class(p.split('/')[-2])
        labels.append(l)

    images = np.array(images)
    labels = np.array(labels)

    t2 = time.time()
    print('Dataset prepared for' ,t2 -t1 ,'sec')
    print('Images:' ,images.shape ,'np.array.shape(files, views, width, height)')
    print('Labels:', labels.shape, ' among 0-3 classes')

    return images, labels
```

CNN 모델 샘플 코드

```python
def cnn_sample(in_shape):    # Example CNN

    ## Sample code
    # in_shape = (, , 3)
    num_classes = 4

    model = Sequential()
    model.add(Conv2D(filters=16, kernel_size=(5, 5), padding='same', input_shape=in_shape))
    model.add(BatchNormalization(axis=-1))
    model.add(ReLU())

    model.add(Conv2D(filters=32, kernel_size=(5, 5), padding='same'))
    model.add(BatchNormalization(axis=-1))
    model.add(ReLU())
    model.add(ZeroPadding2D(padding=((0, 0), (0, 1))))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=64, kernel_size=5, padding='same'))
    model.add(ReLU())
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=128, kernel_size=6, padding='same'))
    model.add(BatchNormalization(axis=-1))
    model.add(ReLU())
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(filters=256, kernel_size=5, padding='same'))
    model.add(BatchNormalization(axis=-1))
    model.add((ReLU()))

    model.add(Flatten())
    model.add(Dense(100, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))

    return model
```

4

평가기준

NSML Leaderboard

Categorical accuracy

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

Sensitivity

: 모든 positive case 중 positive라고 예측한 결과

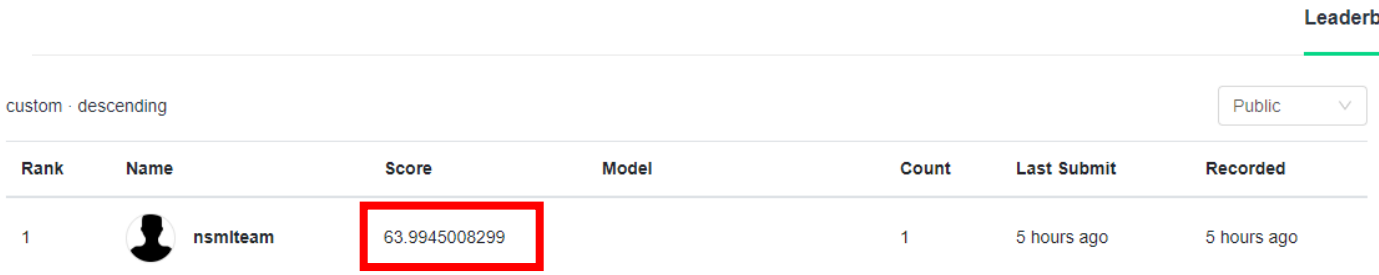
$$Sensitivity = \frac{TP}{TP + FN}$$

Specificity

:모든 negative case 중 negative라고 예측한 결과

$$specificity = \frac{TN}{TN + FP}$$

NSML Leaderboard



Score : aa.aannnnpppp

AccuracySensitivitySpecificity

→ Accuracy를 기준으로 ranking 1-5까지는 sensitivity와 specificity 함께 평가

ThanKYou!