

## 결과 요약지

대회명: 근골격 데이터

참가팀명	담미온	팀원수	3
참가주제	근골격 데이터		
모델 설명	<p>pip를 사용해 설치한 부분</p> <p>tensorflow gpu 2.0.0</p> <p>scikit-image 0.18.3</p> <p>numpy 1.19.2</p> <p>Spine 모델: Unet의 구조를 유지하면서 Residual 블록을 추가한 Resunet 모델을 사용했다. Residual블록은 3X3convolution, 3X3convolution을 적용시킨다. Resunet은 인코더, 브릿지, 디코더로 이루어져있다. 인코더는 512X512X1 이미지를 받아서 Residual 블록을 거치면서 각각의 특징들을 디코더에 넘겨준다(skip). 브릿지는 인코더의 마지막 출력을 받고 Residual 블록을 거친 후 디코더로 넘겨준다. 브릿지에서는 Upsampling한 특징과 인코더에서 Skip한 특징을 Concatenate하면서 디코더를 구성한다. 디코더는 512X512X1 마스크 된 이미지를 출력한다.</p> <p>Pain 모델: KeyPoint탐지에 유리한 Hourglass network를 사용하였다. Hourglass network는 Reidual 블록 대신 Bottleneck블록을 사용한 인코더와 디코더로 이루어져 있다. Bottleneck블록은 채널 수를 설정하여 1X1convolution, 3X3convolution, 1X1convolution을 적용시킨다. Input으로 512X512X1 이미지를 받으면 Resunet과 비슷하게 인코더에서 skip한 특징을 디코더에서 Upsampling 시키면서 Concatenate한다. 디코더는 16개의 Pain Keypoint를 출력한다.</p>		

<p>성능 평가 결과</p>	<ul style="list-style-type: none"> <li>Spain의 경우</li> </ul> <pre>[ ] -0.10656894080984736 -1.6875336274051738</pre> <p>39개의 validation 데이터 셋으로 평가를 진행하였다. 척추L4번과 척추L5번 사이의 각도와 간격에 대해 GT와 개발된 알고리즘으로 측정된 결과간에 R2 score가 각각 -0.10, -1.68 이 나왔다.</p> <ul style="list-style-type: none"> <li>Pain의 경우</li> </ul> <pre>Line 1 R2 Score: -4.851438379567778 Line 2 R2 Score: -2.4432105169579024 Line 3 R2 Score: -3.110664328217755 Line 4 R2 Score: -2.2087769563873225</pre> <p>60개의 validation 데이터 셋으로 평가를 진행하였다. 좌우 각각 내측 관절 간격과 외측 관절 간격에 대해 GT와 개발된 알고리즘으로 측정된 결과간에 R2 score가 각각 -4.85, -2.44, -3.11, -2.20이 나왔다.</p>
<p>기타 사항</p>	<ul style="list-style-type: none"> <li>Spine_final 코드 돌리는법</li> </ul> <ol style="list-style-type: none"> <li>1. data_pre와 train_data_loading 함수에 있는 변수 img_mask_path에 train img와 train mask 파일을 저장할 폴더 경로를 지정해주세요.</li> <li>2. augmentation 함수에 있는 변수 savepath에 augmentation 데이터 파일을 저장할 폴더를 지정해주세요.</li> <li>3. predict_val 함수에 있는 변수 test_img_mask_path에 Test img와 Test mask 파일을 저장할 폴더 경로를 지정해주세요.</li> <li>4. get_results 함수에 변수 pred_img_pat에 predict된 파일을 저장할 폴더 경로를 지정해주세요.</li> <li>5. main 함수에 train_path에 Train Dataset 경로, path에 check와 predict를 저장할 경로, test_path에 Test Dataset 경로를 지정해주세요.</li> </ol> <ul style="list-style-type: none"> <li>Pain_final 코드 돌리는법</li> </ul> <ol style="list-style-type: none"> <li>1. Pain_final1.ipynb 파일의 변수 Train_data_folder에 Test Dataset의 경로를 지정해주세요</li> </ol>

요.

2. np.save에 각 변수에 맞는 개인경로를 지정해주세요.

3. Pain\_final1.ipynb 파일의 변수 image\_array와 label\_array에 Pain\_final1.ipynb에서 Train Data를 저장한 경로를 지정해주세요

4. Evalute 부분의 각각 변수에 맞는 경로만 설정해주시면 됩니다.

● weight 파일로 평가만 하는경우

1. main문에서 test\_path에 Test Dataset 경로를 지정해주세요.

2. main문에서 imgs\_train, imgs\_mask\_train, imgs\_name =  
train\_data\_loading(train\_path, image\_size = image\_size)

model = deep(imgs\_train, imgs\_mask\_train, path, batch\_size = batch\_sizes, epochs  
= epochs, image\_size=image\_size)

이 2줄을 주석처리하세요.

3. # from tensorflow.keras import Model

# model = build\_model()

# model.compile(optimizer=Adam(lr=0.0001), loss=dice\_coef\_loss,  
metrics=['accuracy', sens, dice\_coef\_loss])

# model.load\_weights('/home/hackerton/jupyter/Spine\_data/pred/test1.h5')

이 4줄의 주석을 풀고 weights 파일의 경로를 지정해주세요.