

# 결과 요약지

대회명: 근골격 데이터

참가팀명	답러블	팀원수	2
참가주제	근골격 데이터		
<div>모델</div> <div>설명</div>	1.개발 환경		
	Shapely	1.8.0	<pip>
	SimpleITK	2.1.1	<pip>
	six	1.16.0	<pip>
	sklearn	0.0	<pip>
	sqlite	3.36.0	hc218d9a_0
	tensorboard	2.0.2	<pip>
	tensorflow-estimator	2.0.1	<pip>
	tensorflow-gpu	2.0.0	<pip>
	termcolor	1.1.0	<pip>
	terminado	0.12.1	<pip>
	testpath	0.5.0	<pip>
	threadpoolctl	3.0.0	<pip>
	tifffile	2021.11.2	<pip>
	tk	8.6.11	h1ccaba5_0
	tomli	1.2.2	<pip>
	tornado	6.1	<pip>
	tqdm	4.62.3	<pip>
	traitlets	5.1.1	<pip>
	typing_extensions	4.0.0	<pip>
	urllib3	1.26.7	<pip>
	wcwidth	0.2.5	<pip>
	webencodings	0.5.1	<pip>
	Werkzeug	2.0.2	<pip>
	wheel	0.37.0	pyhd3eb1b0_1
	widetsnbextension	3.5.2	<pip>
	wrapt	1.13.3	<pip>
	xz	5.2.5	h7b6447c_0
	zipp	3.6.0	<pip>
	zlib	1.2.11	
	2. 모델 설명		
	1.MISSON 1: 척추 X-ray 모델		
	1) 데이터		
	Train sets	Validation sets	Test sets
	173	43	24
			Total
			240

a) 학습용 데이터 가공

- .Dcm -> .jpg

- .json -> .png

b) 이미지 크기

- 512\*512

c) 코드 순서

- 1. data\_preprocessing.ipynb / 이미지 전처리
- 2. spine-base.ipynb / 학습
- 3. evaluation.ipynb / 검증

## 2. MISSION 2: 무릎 X-ray 모델

1) 데이터

class	data
Train	172
Validation	44
Test	24
Total	240

: validation set과 test set을 제외한 172 개 데이터로 학습하였음

: train과 validation은 8:2 비율로 나누었음

2) 모델 학습 파라미터

: learning rate가 너무 떨어져서 reduce\_lr callback은 사용하지 않음

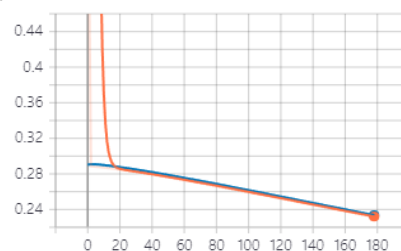
: 50 에폭 기준으로 earlystop을 사용하였으나 학습 loss가 계속해서 떨어져서 실행  
되진 않음

: 데이터가 적어 batch size는 32로 늘렸으며, 에폭은 1000으로 실행하였음

: 에폭 기준으로 log를 저장하여 tensorboard를 통해 학습 진행도를 꾸준히 확인하  
였음

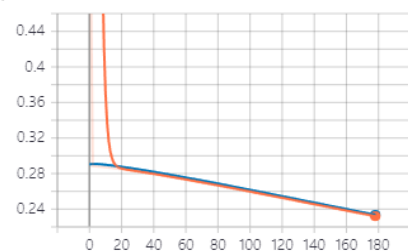
epoch\_loss

epoch\_loss



epoch\_mse

epoch\_mse



1.MISSON 1: 척추 X-ray 모델

1)테스트 데이터 로드

```
-----
Creating test images...
-----
0/24
img : 255
mask : 255
-----
normalization start...
-----
img : 1.0
mask : 1
mask : 0
loading done
predict test data
24/24 [=====] - 1s 56ms/step
(24, 512, 512, 1)
[0.0000000e+00 2.9802322e-08 5.9604645e-08 ... 9.9999988e-01 9.9999994e-01
 1.0000000e+00]
```

2) 딥러닝 모델 성능 평가 결과

```
acc avg : 0.9926
specificity avg : 0.9968
sensitivity avg : 0.8677
dsc avg : 0.8823
```

3) 척추 각도 및 거리 알고리즘 검증 평가

- L4-L5 각도 측정 비교 (왼쪽:딥러블 / 오른쪽:정답)

알고리즘 :	2.362533155041766	정답 json:	4.344206705540099
알고리즘 :	18.19124762617083	정답 json:	13.960041705303624
알고리즘 :	12.091730838198394	정답 json:	22.92778449233107
알고리즘 :	85.04354194946474	정답 json:	5.296476359053976
알고리즘 :	14.102234856631028	정답 json:	15.48811530885612
알고리즘 :	5.606165951466643	정답 json:	5.738305930431428
알고리즘 :	16.09081634885221	정답 json:	6.67223639679098
알고리즘 :	11.587330647526375	정답 json:	5.987122109357242
알고리즘 :	4.535599991337212	정답 json:	3.59310660190373
알고리즘 :	116.56505117707799	정답 json:	15.889254141936588
알고리즘 :	28.73979529168807	정답 json:	14.086299765850043
알고리즘 :	4.960980910189332	정답 json:	7.633546356604524
알고리즘 :	11.164032811584164	정답 json:	40.88918642166332
알고리즘 :	20.80679101271124	정답 json:	21.31603658398785
알고리즘 :	1.3639275316029233	정답 json:	25.45895408447377
알고리즘 :	164.74488129694217	정답 json:	14.147875499226789
알고리즘 :	11.051703271602694	정답 json:	24.99335012404527
알고리즘 :	10.549611570812248	정답 json:	12.299888159267864
알고리즘 :	157.01128319791923	정답 json:	1.3180774040926184
알고리즘 :	11.913692605255847	정답 json:	6.174831242746487
알고리즘 :	5.95983911071301	정답 json:	19.194369147206494
알고리즘 :	12.260599273398618	정답 json:	12.444139418923376
알고리즘 :	0.7518305506137538	정답 json:	12.060330462123556
알고리즘 :	13.274162221417749	정답 json:	14.49912046550896

성능 평가  
결과

L4-L5 각도에 대한 r2 score= -0.2427490574760589

- L4-L5 거리 측정 비교 (왼쪽:답러블 / 오른쪽:정답)

알고리즘 :	13.892443989449804	정답 json:	46.32493928760188
알고리즘 :	19.72941965694886	정답 json:	50.695167422546305
알고리즘 :	14.0089257261219	정답 json:	63.50590523722971
알고리즘 :	41.00304866714181	정답 json:	72.47068372797375
알고리즘 :	15.628499608087784	정답 json:	84.48076704197234
알고리즘 :	14.212670403551895	정답 json:	8.134123677446759
알고리즘 :	27.613402542968153	정답 json:	54.56189146281496
알고리즘 :	16.3783393541592	정답 json:	92.91393867445294
알고리즘 :	20.09975124224178	정답 json:	2.5536186089547512
알고리즘 :	196.05420168922674	정답 json:	8.232131315765074
알고리즘 :	167.08381130438698	정답 json:	88.60022573334675
알고리즘 :	18.117670931993437	정답 json:	89.1852005660132
알고리즘 :	16.3783393541592	정답 json:	36.05551275463989
알고리즘 :	30.700162866017504	정답 json:	15.329706716046461
알고리즘 :	11.672617529928752	정답 json:	42.95346318982906
알고리즘 :	28.460498941515414	정답 json:	34
알고리즘 :	16.15549442140351	정답 json:	99.12618221237011
알고리즘 :	14.534441853748634	정답 json:	16.327694662750158
알고리즘 :	16.560495161679196	정답 json:	10.28869671046824
알고리즘 :	18.66815470259447	정답 json:	68.59300255857006
알고리즘 :	18.34393632784414	정답 json:	127.58134659894448
알고리즘 :	14.637281168304447	정답 json:	36.22154055254967
알고리즘 :	6.576473218982953	정답 json:	43.86342439892262
알고리즘 :	13.46291201783626	정답 json:	8.671272859275044

L4-L5 거리에 대한 r2 score= -0.7571963519993772

## 2.MISSON 2: 무릎 X-ray 모델

Point 1 Distance Error:	41.219 ± 13.622	(mm)
Point 2 Distance Error:	39.318 ± 14.277	(mm)
Point 3 Distance Error:	50.157 ± 34.303	(mm)
Point 4 Distance Error:	49.005 ± 35.477	(mm)
Point 5 Distance Error:	73.058 ± 28.962	(mm)
Point 6 Distance Error:	76.29 ± 28.275	(mm)
Point 7 Distance Error:	103.435 ± 21.587	(mm)
Point 8 Distance Error:	112.782 ± 20.962	(mm)

Line 1 R2 Score: -0.3647943864840868  
Line 2 R2 Score: 0.028861979417636574  
Line 3 R2 Score: -0.0999505455589833  
Line 4 R2 Score: -7.2516232102451745

Line 1 Distance Error:	1.553 ± 1.157	(mm)
Line 2 Distance Error:	1.377 ± 0.937	(mm)
Line 3 Distance Error:	1.517 ± 0.974	(mm)
Line 4 Distance Error:	4.55 ± 1.758	(mm)

: 24 개 데이터로 모델 성능 평가 실행, R2 score 와 같이 상관관계를 평가하는 점수를

	<p>출력하기에는 조금 적은 데이터로 생각됨.</p> <p>: 60개 데이터로 모델 성능 평가를 실행하면 보다 높은 점수가 나올 것이라 생각됨.</p>
기타 사항	<p><b>1.MISSON 1: 척추 X-ray 모델</b></p> <p>첨부한 3.validation 코드에서</p> <p>먼저 딥러닝 성능파트 부분에서 밑에 있는 코드를 모두 실행 그 중</p> <ol style="list-style-type: none"> <li>1) Predit_save 함수의 pred_img_path에 딥러닝이 예측한 이미지를 저장할 경로 및 이름 설정</li> </ol> <pre>def predict_save(pred_list,name_list):     pred_img_path = '../dataset/spine/ex1/train/pred/'     if not os.path.isdir(pred_img_path):         os.makedirs(pred_img_path)      imgs = pred_list     for i in range(imgs.shape[0]):         img = imgs[i]         img[img &lt;= 0.5] = 0         img[img &gt; 0.5] = 255         img = array_to_img(img)         img.save(pred_img_path+"%s_pred.png" %(name_list[i]))</pre> <ol style="list-style-type: none"> <li>2) Args에 훈련데이터 경로, 테스트데이터 경로 설정, load_model 함수에 딥러닝 모델이 있는 경로 설정</li> </ol> <pre>args = easydict.EasyDict({     "train_path" : "../dataset/spine/ex1/train/",     "test_path": "../dataset/spine/ex1/test/",     "image_size": 512,     "epochs":100,     "batch_size":4 })  model=load_model("../dataset/spine/ex1/train/pred/ap_aug_exp1.h5",custom_objects={'dice_coef_loss': # model = multi_gpu_model(model,gpus=2) test_name = predict_val(model,args.test_path,image_size = args.image_size)</pre> <p>➔ 딥러닝 결과 산출</p>

척추 L4-L5 사이 간격 및 분절 각도 측정 알고리즘 성능평가 파트에서 밑에 있는 코드를 모두 실행 그 중

- 3) Get\_result 함수에서 딥러닝이 예측한 이미지가 들어있는 경로 설정(path변수) 및 test\_path변수에서 이름 알맞게 설정

```
def get_results(test_name):  
    angle_list = []  
    dist_list = []  
    angle_list.clear()  
    dist_list.clear()  
    len(test_name)  
    for i in range(len(test_name)):  
        # 예측 이미지 경로  
        path="../dataset/spine/ex1/train/pred/"  
        test_path=path+test_name[i]+"_pred.png"  
        # test_name=glob.glob(test_path+"/*.png")
```

- 4) args에서 훈련 및 테스트셋 경로 설정

```
args = easydict.EasyDict({  
    "train_path" : "../dataset/spine/ex1/train/",  
    "test_path": "../dataset/spine/ex1/test/",  
    "image_size": 512,  
    "epochs":100,  
    "batch_size":4  
})  
  
angle_list,dist_list = get_results(test_name)  
get_score(angle_list,dist_list,test_name,args.test_path)
```

➔ r2 score 산출

## 2.MISSION 2: 무릎 X-ray 모델

2-1. 테스트 데이터 개수만 변경하여 np.zeros 변수 넣어주시면 됩니다.

: original\_size\_label = np.zeros((테스트셋 개수, 8, 2))

: original\_size\_predict = original\_size\_predict.reshape(테스트셋 개수, 8, 2)

2-2. predict, label, file\_name, reduce\_ratio, pixel\_spacing 변수 npy 파일 경로 설정해주세요.

2-3. test set npy 파일 형성 시 512 사이즈로 제작해주시면 됩니다.

: 1\_load\_data.ipynb 파일에서 test set 이미지를 512 사이즈로 resize하여 npy 파일 생성 >

	3_evaluation.ipynb 파일에서 성능 평가
--	-------------------------------