메디컬 딥러닝 최종 보고서

문제 소개

먼저, KL grade를 구분하기에 Xray 방사선 이미지가 가장 효율적입니다.

이유는 초기에는 정상 소견을 보일 수 있으나 점진적으로 관절 간격의 감소가 나타나며 연골 아래 뼈의 음영이 짙어지는 경화 소견을 볼 수 있고, 더욱 진행되면 관절면의 가장 자리에 뼈가 웃자란 듯한 골극이 형성되고 관절면이 불규칙해집니다.

이차성 관절염의 경우 원인이 되는 과거 외상이나 질환의 흔적 혹은 변형 등이 관찰되기도 합니다.

다만 방사선학적 변화가 증상 및 활동력의 심한 정도를 그대로 반영하는 것은 아니어서 40세 이상에서 90%정도는 방사선학적으로 퇴행성 변화를 보이지만 이 중 30%정도만이 증상을 보이게 됩니다.

출처 : http://www.snuh.org/health/nMedInfo/nView.do?category=DIS&medid=AA000196

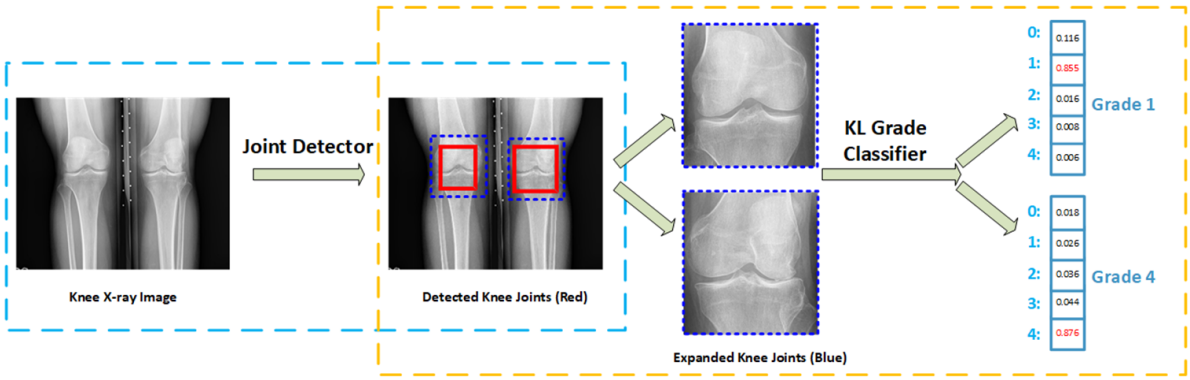
이미지 출처 : https://m.blog.naver.com/anzyme/221149697273

관련 연구

이러한 문제를 해결하기 위해 조사한 관련 연구는 다음과 같습니다.

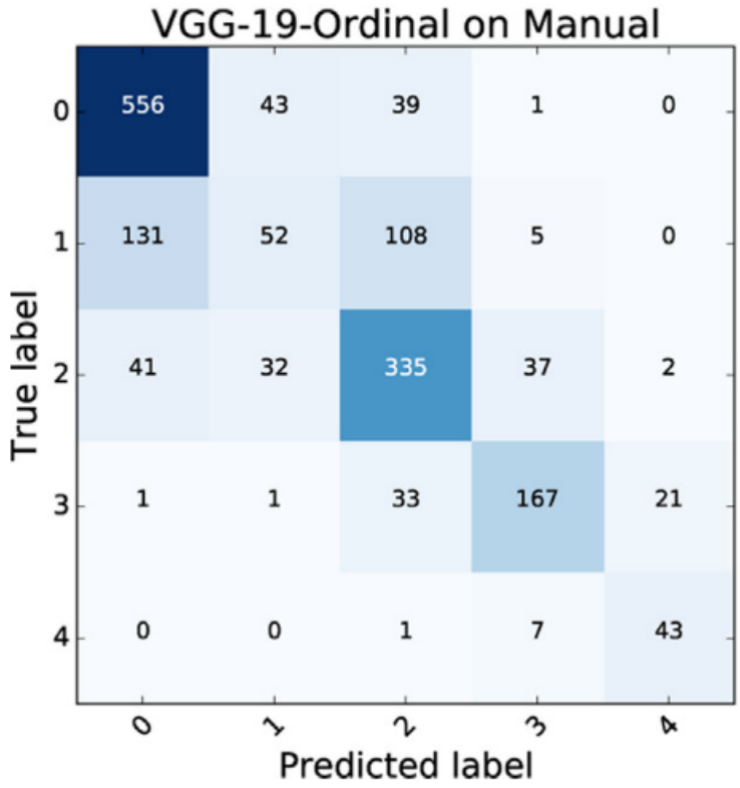
"Fully automatic knee osteoarthritis severity grading using deep neural networks with a novel ordinal loss"

해당 연구는 무릎의 관절염 분류에 관한 연구이며, 학습에 사용한 데이터 생성을 위해 YoloV2를 탐지에 사용하였고, 학습에 VGG19 모델을 활용하여서 관절염 등급을 분류한 연구입니다.



아래와 같은 정확도를 보이며, 해당 분야의 SOTA가 약 70%인 것으로 보았을 때,

SOTA와 유사한 정확도를 보인다는 것을 알 수 있습니다.



	정확도
정답(%)	30.42
오류(%)	69.58

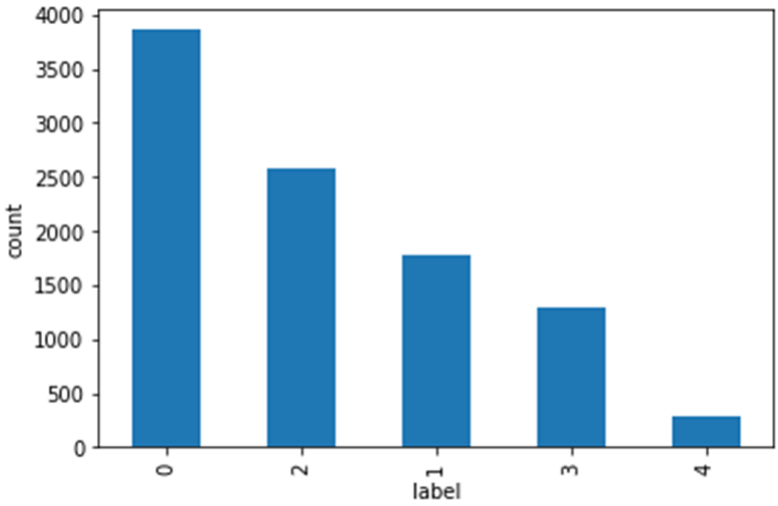
출처 : https://www.sciencedirect.com/science/article/abs/pii/S0895611118304956

Github : https://github.com/PingjunChen/KneeAnalysis

학습 데이터

"Fully automatic knee osteoarthritis severity grading using deep neural networks with a novel ordinal loss"에서 사용되었던 데이터 셋을 사용하여 모델의 학습에 사용하였습니다.

학습 데이터 세트의 분포는 아래와 같습니다.



0 = 39.41%(3857장), 1 = 18.09%(1770장), 2 = 26.34%(2578장), 3 = 13.14%(1286장), 4 = 3.01%(295장)

의 분포도와 개수로 구성되어 있으며, 0 > 2 > 1 > 3 > 4로 구성 되어있습니다.

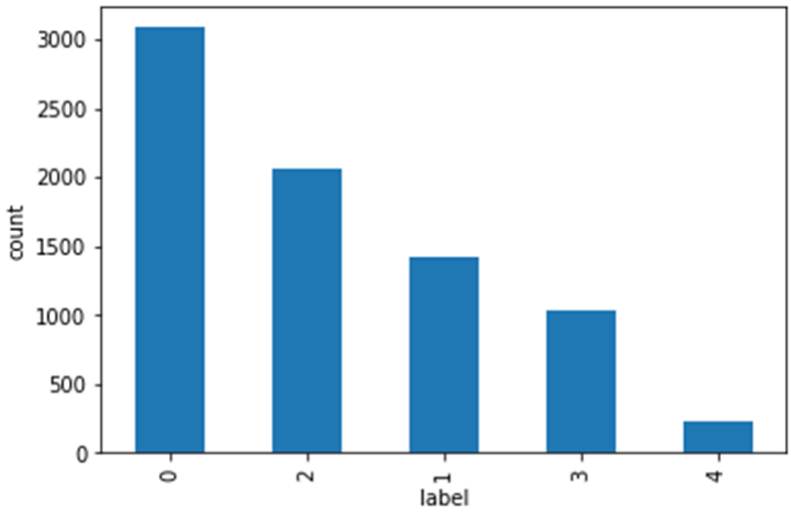
DPhi 제공 데이터

학습 데이터를 검증 데이터로 사용

검증 데이터 세트로는 DPhi "Data Sprint #35: Osteoarthritis Knee X-ray"에서 제공하는 데이터 세트의 학습 데이터를 학습과정에서의 검증 데이터로 사용하였으며, 모델의 성능 검증을 위해서 DPhi의 테스트 데이터 세트를 활용하였습니다.

출처 : https://dphi.tech/challenges/data-sprint-35-osteoarthritis-knee-x-ray/81/overview/about

먼저 사용된 검증 데이터 세트에 대해 먼저 설명하겠습니다.



위의 이미지와 같은 분포를 가지고 있으며,

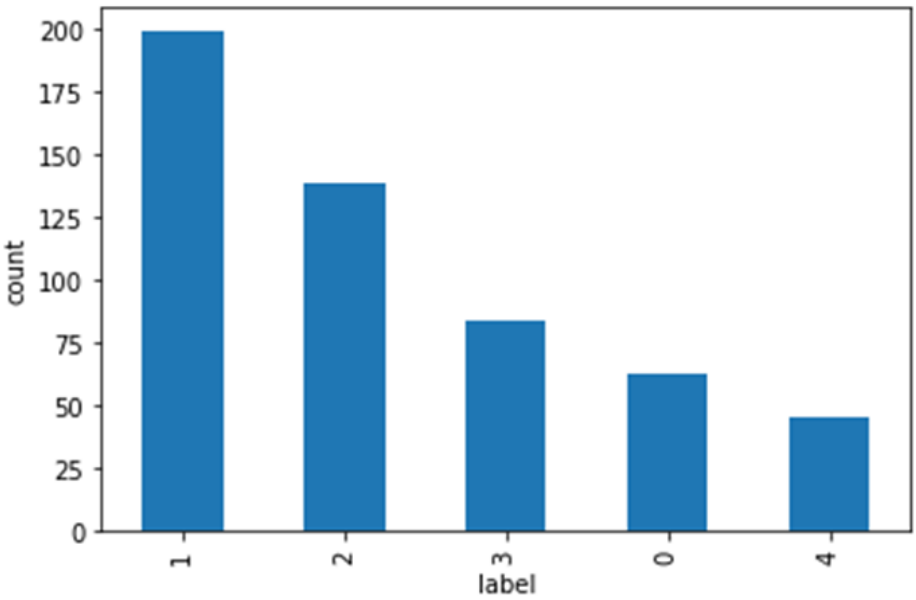
0 = 3085장, 1 = 1416장, 2 = 2062장, 3 = 1029장, 4 = 236장으로 0 > 2 > 1 > 3 > 4로 구성 되어있습니다.

추가 테스트 데이터

이병대 교수님 께서 보내주신 데이터를 모델의 테스트 데이터로 사용하였으며,

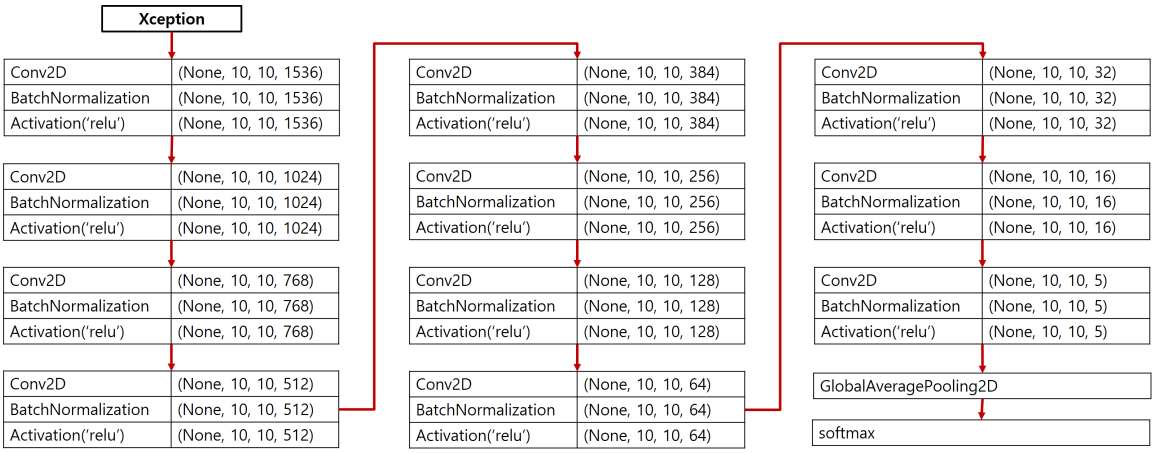
위의 이미지를 직접 손으로 Crop 후 224 사이즈로 Resize하여 아래와 같은 이미지로 구성된 테스트 데이터 세트를 제작하였습니다.

제작된 테스트 데이터 세트는 다음과 같은 분포를 가지며, 1 > 2 > 3 > 0 > 4로 구성되어있습니다.



다음으로는 진행한 실험에 대한 설명을 드리겠습니다.

모델 구조



제가 구성한 모델의 구조는 위의 이미지와 같으며, 기존의 Xception 모델을 활용하였고, 끝 단의 Layer에 filter의 개수를 조밀히 추가하여서 특징을 좀 더 세분화하고자 하였습니다.

실험 환경

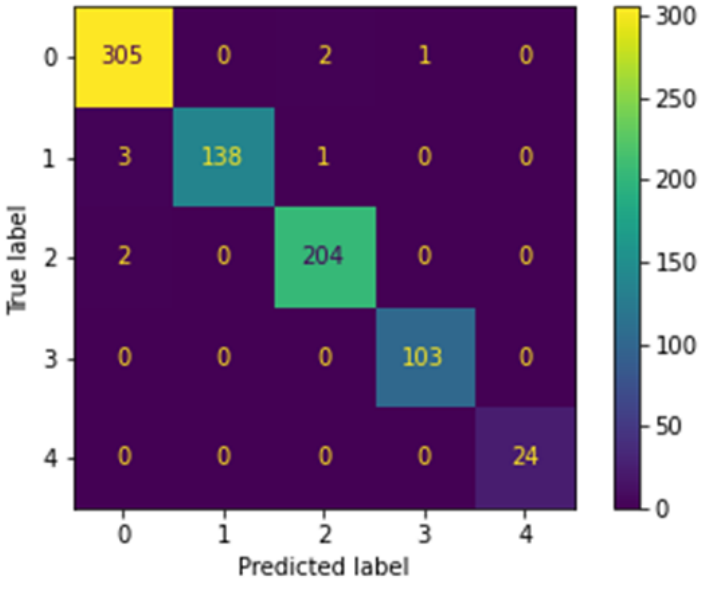
Experiment	
OS	Windows 10
GPU	GeForce RTX 3090
CUDA	11.2
cuDNN	8.1
Tensorflow	2.6.0
Tensorflow-gpu	2.6.0
Keras	2.6.0

Hyperparameter	
Model	Xception
Batch size	8
Optimizer	Adam (lr = 0.00001, decay = 0.0001)
Train Epochs	135 (Early stopping)
Plus Train Epochs	50
ImageDataGenerator	Rescale = 1./255
Image size	224x224

실험 결과

참고한 모델의 정확도

참고한 모델의 검증 데이터(DPhi 학습 데이터)에 대한 정확도

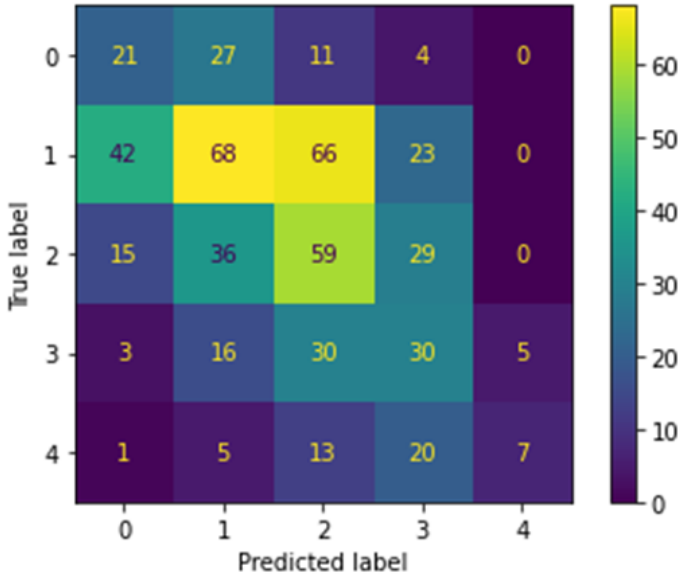


	정확도
정답(%)	0.989
오답(%)	0.011

DPhi의 테스트 데이터에 대한 정확도

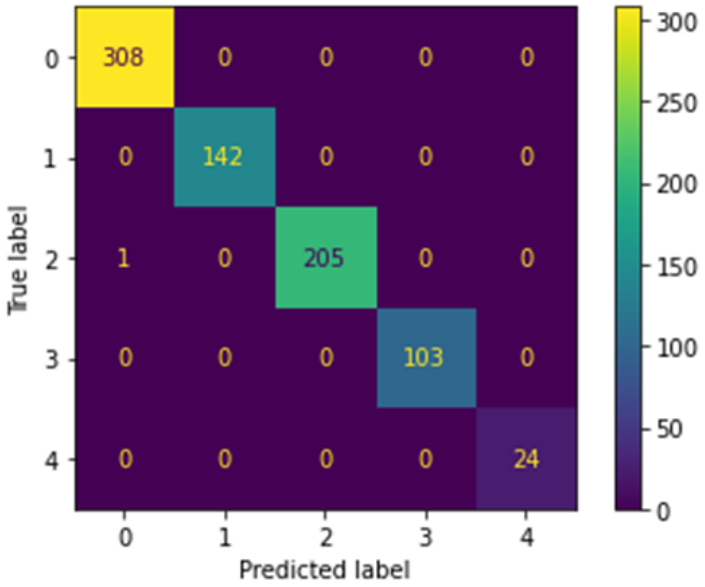
77813	Accepted 06/12/21 06:53 am	98.26	No summary

추가 테스트 데이터에 대한 정확도



	정확도
정답(%)	0.348
오답(%)	0.652

검증 데이터를 통한 Confusion Matrix



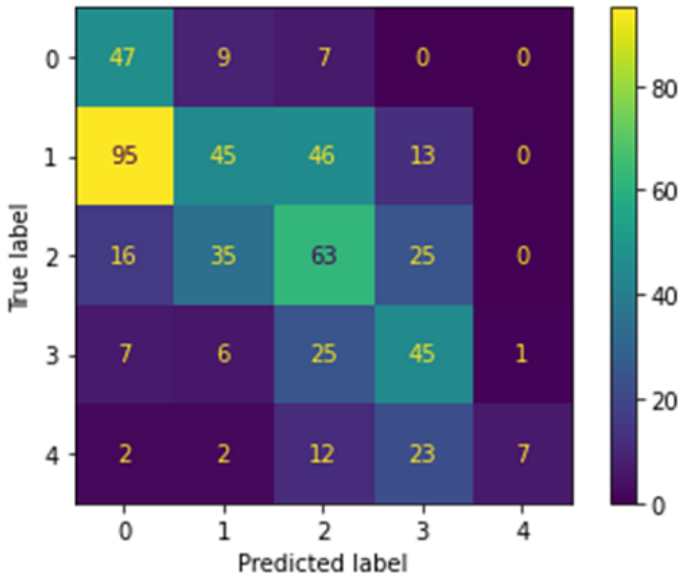
	정확도
정답(%)	0.9986
오답(%)	0.0014

다음과 같은 결과를 보았을 때에는 검증 데이터에 대해서는 99.86%의 분류를 해내었고,

아래에서 볼 수 있듯 test data에 대한 정확도는 99.8468%로 상당히 높은 결과를 나타났습니다.

1	ideal		99.8468	07/12/21 08:00 pm

하지만, 추가 테스트 데이터의 결과는



	정확도
정답(%)	0.3898
오답(%)	0.6102

다음과 같이 약 39%의 결과를 보이면서 아주 낮은 성능을 보였습니다.

정확도에 대한 결과 분석

학습 데이터의 분포는

0 = 39.41%, 1 = 18.09%, 2 = 26.34%, 3 = 13.14%, 4 = 3.01%, 0 > 2 > 1 > 3 > 4

로 구성 되어있으며,

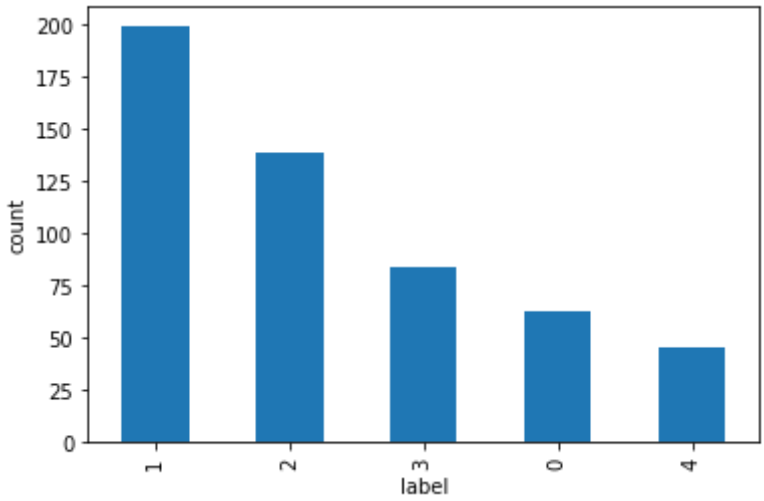
DPhi의 test dataset의 분포는

0 = 39.42%, 1 = 18.07%, 2 = 26.35%, 3 = 13.12%, 4 = 3.01%, 0 > 2 > 1 > 3 > 4

로 구성 되어있습니다.

이러한 분석을 통해서 두 가지의 데이터 세트가 동일한 분포를 가지고 있었고, 학습 데이터와 테스트 데이터의 분포가 엄청나게 유사하기 때문에 Data leakage 현상으로 인해 정확도가 높게 나온것 같습니다.

이렇게 생각하는 이유는



저희가 사용한 테스트 데이터셋의 분포는 1 > 2 > 3 > 0 >4 순으로 기존의 테스트 데이터 셋과 다른 분포를 가지고 있으며, 테스트 데이터셋은 예측된 분포를 보시면 0 > 2 > 3 > 1 > 4로 학습 데이터의 분포와 동일하게 예측의 분포가 맞춰지는 것을 알 수 있습니다.

label	사용된 데이터의 개수	모델이 예측한 라벨별 개수
0	63	167
1	199	97
2	139	153
3	84	106
4	46	8

결과적으로, 99.80%로 엄청나게 높은 정확도가 나온 이유는 Data Leakage 현상과 학습 데이터와 테스트 데이터의 분포가 동일하였기 때문이라고 생각됩니다.

이러한 현상을 해소하기 위해서 기존의 학습 데이터는 유지하고 학습에 사용하고 추가 테스트 데이터에 대해 데이터 증강을 진행 후 검증 데이터로 사용하였으나, 학습 과정에서 validation loss가 계속해서 40% 내외를 유지하며 학습이 이루어지지 않았습니다. 테스트 데이터에 대해 검증을 위해서 기존의 참고하였던 모델에도 적용해 보았으나, 오히려 정확도가 더 떨어지는 것으로 보았을 때 제가 직접 구성한 테스트 데이터 세트가 잘못 제작되었다고 생각합니다.