10조 최종 보고서

1. Introduction

- Background

  2022 트렌드에 대해 찾아보면 “바른생활 루틴이”, “헬시 플레저” 등 자기 관리와 관련한 많은 단어들을 어렵지 않게 찾을 수 있다. 또한 최근 10년간 피트니스 센터가 54%나 증가하고, 코로나 시기에도 피트니스 센터가 늘었다는 기사도 어렵지 않게 확인 할 수 있었다. 그만큼 최근들어 사람들의 자기관리에 대한 관심도가 높아지고, 그 중에서도 눈에 띄게 달라진 것은 헬스장 이용 인원이 많아졌다는 것이다.
  또한 6월 7일 기준, 인스타그램에 #오운완 게시글이 140만여개, #운동기록 게시글이 150만여개나 될 정도로 예전보다 운동에 대한 관심도와 운동을 기록하고자 하는 니즈가 상당히 증가했다는 것을 실감할 수 있다. 이러한 시장의 분위기에 맞춰 시중에도 운동 기록을 도와주는 어플들이 많이 나와있다.
  그러나 대부분의 운동 기록 서비스들은 사용자가 어떤 운동을 했는지 기억하고 직접 캘린더에 작성해야 한다는 단점이 있다. 소수의 다른 운동 기록 서비스들은 추가로 구매한 디바이스를 활용하여 운동 횟수 및 시간을 측정하고 기록하는 기능을 제공하는 경우도 있지만, 추가 장비가 필요하다는 점과 센서 위에서 할 수 있는 운동만이 측정 대상이기 때문에 실용적이지 않다고 판단했다.
  이러한 기존 서비스들의 단점들을 보완하여 본 팀은 사용자가 본인들의 헬스 영상을 기록하며 운동 일지가 자동으로 기입되는 “득근득근 성장일기”를 기획하였다. “득근득근 성장일기”는 사용자들이 운동하는 동안 본인의 자세를 카메라로 촬영하여 운동하는 모습을 동영상으로 만들고, 촬영된 영상에서 이용자의 운동하는 자세를 분석하여 ‘어느’ 운동을 하는지 “득근득근 성장일기”에 자동으로 기록되는 기능을 제공한다.
  “득근득근 성장일기”에 운동 영상이 기록되고 자동으로 본인의 운동 일지가 작성되므로 기록하는 일의 귀찮음을 줄여주는 것은 물론 운동에 대한 열정도 더할 수 있을 것이고, 운동의 루틴과 일정을 편하게 관리 할 수 있을 것으로 기대한다.

- Requirements, assumptions, risks and constraints

  어떤 운동인지 판별하기 위해서는 주어진 운동 영상에 대해서 사람의 움직임을 확인할 수 있는 skeleton keypoints extraction 모델과 추출한 keypoints를 통해 운동의 종류를 예측해내는 모델이 필요하다. 따라서 본 프로젝트에서는 이 두 모델에 대해 병렬적으로 작업하기 위해 두 팀으로 나누어 진행하도록 할 것이다. 또한, 정확도 높은 exercise classifier 모델을 얻기 위해서는 skeleton keypoints extraction 모델에서 높은 정확도의 keypoints를 얻어내는 것이 선행 조건이라고 할 수 있다.

  skeleton keypoints extraction 모델링에 있어 예상되는 문제점과 제약 사항은 다음과 같다. 이미지로 학습하는 모델의 경우에는 많은 이미지 데이터가 요구되고 학습에 요구되는 시간이 매우 길다. 따라서 Pose Estimation에 자주 이용되는 pre-trained 모델이 필요할 것이다. 그 뿐만 아니라 데이터의 양이 방대하기 때문에 데이터 관리와 전처리에 많은 시간이 걸릴 것으로 예상된다.

  exercise classification 모델링에 있어 예상되는 문제점과 제약사항은 다음과 같다. 먼저 잘못된 운동 자세, 비슷한 운동 자세 데이터에 대해서는 분류 모델의 정확성은 낮을 것으로 보인다. 다음으로 운동 분류 모델의 주 모델인 lstm에서 sequence 크기가 동일해야 하기 때문에 영상 이미지 데이터를 모두 32 프레임으로 통일하여 좌표값을 뽑아내는 것이 필요하다. 또한 이 과정에서 운동하는 사람의 운동 속도에 따라서 같은 길이의 영상을 32개의 프레임으로 자르게 되면, 32개의 영상 이미지간의 좌표 변화 값이 다르기 때문에 AI hub 에서 제공하지 않은 외부 데이터에 대한 정확도가 낮을 것으로 예상된다.

- Project goals and success criteria

  운동 영상을 통해 영상 속 사람이 하고 있는 운동이 무엇인지 정확히 추정해내는 것이 목표이다. skeleton keypoints extraction 모델에서는 각 이미지들의 RMSE의 평균을 구한 값이 가로 픽셀의 수인 1920의 3%값인 57.6 이하면 성공이라고 판단할 것이다. 운동 분류 예측 모델에서는 전체 예측 정확도가 93%이상이면 성공이라고 판단할 것이다. 마지막으로 외부 데이터를 사용하여 keypoints 추출과 운동 분류를 진행했을 때 정확도가 80% 이상이면 최종적으로 성공적인 모델을 만들었다고 판단할 것이다.

2. Data Preparation

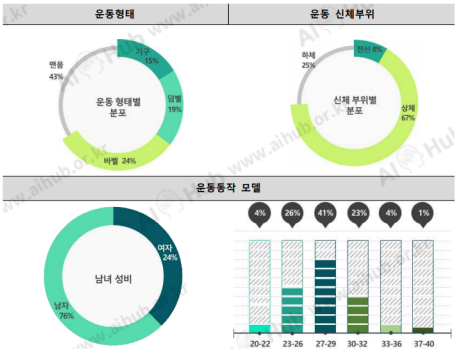
- Used data

  skeleton 추출 모델에서는 AI허브(aihub.or.kr)에 있는 피트니스 자세 이미지를 데이터로 사용하였다. 좀 더 구체적으로 설명하자면 먼저 한국인 평균의 ±20%의 체격을 가진 운동 경력 2~5년의 20~30대의 남녀 70여 명을 모델로 하여 40여 개의 운동 동작에 대해 200,000개의 영상(원천 데이터)을 얻었다. 다음으로 얻어진 영상으로부터 초당 1~3개의 영상 이미지를 추출하여, 총 300만 장 이상의 영상 이미지를 추출했고, 각 이미지에서의 24개의 keypoints와, 운동 정보를 JSON 파일들에 labeling 하였다.

  추가적으로 youtube 에서 운동당 1~2개 운동 자세 영상을 수집하여 외부데이터로 사용하였다. 수집한 외부데이터의 조건은 아래와 같다.
  - 약 15초 동안 앵글이 바뀌지 않는 영상
  - 운동하는 사람이 독립적으로 나오는 영상
  - 전신이 모두 나오는 영상

- Exploratory Data Analysis(EDA)

  아래 그림은 운동 형태, 운동 신체 부위, 운동 동작을 실행하는 모델의 성별과 나이에 따른 운동 영상의 Clip 분포와 dataset의 세부 구성 요소 및 24개 keypoint에 대한 정보이다.



NO	항목명		길이	타입	필수 여부	비고
	한글명	영문명				
	데이터셋정보	info			Y	JSON Object
1	데이터셋명	info.name	100	String	Y	
2	데이터셋상세설명	info.description	1000	String		
3	데이터셋URL	info.url	200	String		
4	데이터셋생성일자	info.date_created	100	String	Y	
	비디오정보	(VIDEO_NAME).jason		List	Y	각 비디오마다 하나의 JSON
5-1	비디오식별자	id	100	String	Y	
5-2	비디오시나리오식별자	s_id	100	String	Y	같은 시나리오를 공유하는 비디오
5-2	비디오너비	width	4	Number	Y	
5-3	비디오높이	height	4	Number	Y	
5-4	비디오명	name	100	String	Y	
5-5	비디오촬영정보	mat	64	Object		카메라 촬영정보
5-6	프레임 별 파일명	frames[]		List		
5-6-1	-	-	1000	String	Y	
5-7	프레임 별 어노테이션	infos[]		Object		
5-7-1	키포인트	keypoints		Object	Y	
5-7-2	운동 종류	category	1000	String	Y	
5-7-3	운동 상태	status	1000	String	Y	
5-8	비디오길이	length	4	Number		
5-9	비디오라이선스	license	100	String		
5-10	비디오촬영일자	date_captured	100	String		
	라이선스	licenses		List	Y	List of JSON Object
6-1	라이선스명	licenses[].name	100	String	Y	
6-2	라이선스URL	licenses[].url	200	String	Y	

keypoint 종류	설명	keypoint 종류	설명
nose	코	right_hip	오른쪽 둔부
left_eye	왼쪽 눈	left_knee	왼쪽 무릎
right_eye	오른쪽 눈	right_knee	오른쪽 무릎
left_ear	왼쪽 귀	left_ankle	왼쪽 발목
right_ear	오른쪽 귀	right_ankle	오른쪽 발목
left_shoulder	왼쪽 어깨	neck	목
right_shoulder	오른쪽 어깨	left_palm	왼쪽 손바닥
left_elbow	왼쪽 팔꿈치	right_palm	오른쪽 손바닥
right_elbow	오른쪽 팔꿈치	spine2(back)	척추2(등)
left_wrist	왼쪽 손목	spine1(waist)	척추1(허리)
right_wrist	오른쪽 손목	left_instep	왼쪽 발등
left_hip	왼쪽 둔부	right_instep	오른쪽 발등

그림 1. 데이터 분포

- Data cleaning / pre-processing
  - skeleton extraction model
    - train, validation, test set 구성
      1. Json파일로부터 각 이미지에서의 각 joint의 좌표를 추출하여 csv로 저장하였다.
      2. 각 운동 종류별로 80장 이상의 이미지를 선별하여 총 4096장의 keypoints좌표와 운동 종류를 담은 train.csv를 만들었다. 만든 csv에서 image name에 해당하는 이미지만 전체 이미지에서 추출하여 train set을 만들었다.
      3. 전체 dataset에서 train set를 제외한 뒤 위에서와 같은 방법으로 총 1600장의 정보를 담은 val.csv를 만들었고, 이를 바탕으로 validation set를 만들었다.

4. 마찬가지로 전체 dataset에서 train set과 validation set를 제외한 뒤 총 1250장의 정보를 담은 test.csv를 만들었고, 이를 바탕으로 test set을 만들었다.

- train 이미지 크기 조정

  트레이닝 성능을 높이기 위해 train dataset에서 keypoints들의 최대 최소값에서 10픽셀정도 마진을 둔 후 길이가 긴 쪽을 기준으로 정사각형으로 잘랐다. 이후 384x384로 resizing하여 모델 학습에 사용하였다.

- 원천 데이터 정제
  - 올바른 운동 자세 선택
    이유 : 원천 데이터에서 올바르지 않은 자세에 대해서는 운동 분류에 있어 노이즈 데이터가 될 수 있어 제거하는 것이 맞다고 판단하였다.
    방법 : AI hub 원천데이터에서 운동 별로 5가지의 conditions을 두어 모두 true 인 운동 상태가 올바른 운동 상태로 선택하였다.
  - 오류를 포함한 데이터셋 수정 및 삭제
    - 원천 데이터 라벨링 오류
      이미지 명명법 : 운동상태 (3자리)-운동종류 (1자리)-자세 (1자리)-운동명 (2자리))
      ⇒ 이미지를 직접 확인하고 명명법에 위반되는 데이터는 명명법에 따라 재라벨링을 진행하였다.
      (history : 1. 운동상태가 다른 경우, 2. 운동 명이 잘못 된 경우)
  - frame 32 개 통일 → LSTM 최적화
    - frame 개수가 32개가 되지 않는 운동은 제거하여 진행하였다.
  - 애매한 동작 제거
    - 푸시업과 니 푸시업과 같이 특정 각도에서 key 포인트만으로 구분하기 힘든 운동 자세를 제외 시켰다.
- exercise classification model
  - 모델 input (csv)
    image(원천데이터 경로), keypoint 24가지(x, y를 분리하여 총 48개) , angle(A, B, C, D, E), exercise(운동종류 이름)로 구성되어 있다. skeleton extraction model에서 받은 keypoint에 angle 값과 exercise 를 라벨링 하여 csv 파일을 생성한다.
  - train, validation, test set 구성
    1. input 으로 받은 csv 데이터를 32개의 프레임 blocks 로 묶어 영상 이미지간의 sequence가 유지될 수 있도록 tensor 차원을 조정한다.
    2. block 단위로 묶인 데이터를 train, test set 으로 8:2 로 분할한다.
- 외부데이터 정제

  최종 평가를 위한 데이터로 외부데이터를 가지고 진행하였다. 수집한 외부 영상들은 15초의 영상으로 편집한 후 영상 총 32개의 프레임으로 잘라 영상 이미지를 추출한다.

3. Modeling

- skeleton extraction model
  - Model / Algorithm

    skeleton extraction model에서는 pre-trained된 모델인 HRNet과 human detector로 YOLO3을 이용하였다. 아래의 그림에서 HRNet은 Pose Estimation 분야에서 ResNet기반의 모델과 비교하여 볼 때, 높은 정확도와 낮은 연산량을 보임을 알 수 있다. 본 모델에서의 skeleton point 예측 결과는 운동 분류 모델에서의 정확도에 크게 영향을 주므로 높은 정확도를 보이는 HRNet을 pre-trained 모델로 선정하였다.

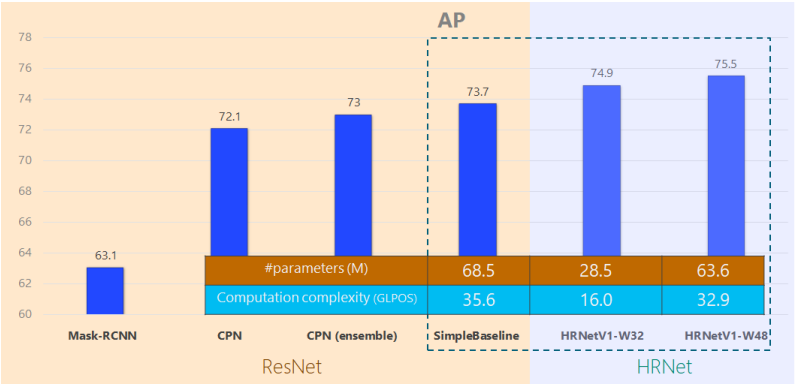


그림 2. Pose Estimation에서 ResNet과 HRNet의 성능 비교

HRNet의 구조는 아래의 그림과 같다.

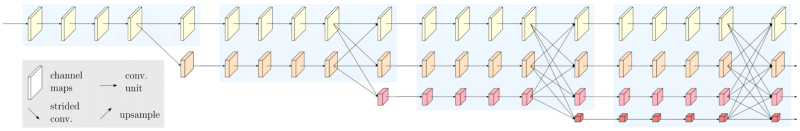


그림 3. HRNet의 구조

위의 구조에서 볼 수 있듯이 high-to-low resolution convolution이 parallel 하게 이루어지기 때문에 HRNet에서는 input 이미지의 해상도가 그대로 유지되어 skeleton keypoint extraction에 있어 높은 정확도를 가진다.

본 모델링에서는 HRNet의 pre-trained된 weight를 불러오고 final layer의 output channel을 48개로 조정한 뒤에 final layer에 대해서 학습을 진행시켰다. 먼저 Human detector로는 YOLO3을 이용하여 사람에 bounding box를 얻어내고, 사람이 있음을 인식했다면 모델에 대한 학습이 진행되었다. YOLO3를 통해 사람의 bounding box를 먼저 찾아내는 이유는 다음과 같다. 사람이 아닌 다른 사물이 움직이는 경우 정확한 skeleton을 추출해낼 수 없고, bounding box에 해당하는 부분에서만 모델링이 이루어지면 되므로 리소스 또한 절약할 수 있다. 따라서 본 모델에서는 위에서 설명한 방식을 채택하였고, YOLO3에서 사람의 bounding box를 찾지 못하는 경우, 해당 이미지에 대해서는 학습이 이루어지지 않는다.

- Hyperparameter tuning

$$RMSE = \sqrt{\frac{1}{24}\sum_{i=1}^{24}(x_i - x_i^{'})^2 + (y_i - y_i^{'})^2}$$

그림 4. 특정 이미지의 RMSE 공식

위 그림은 특정 이미지의 RMSE 값을 구하는 공식을 나타낸다. 그림에서 x, y는 Json파일에서 추출한 각 keypoint의 정답 x좌표, y좌표이고 x', y'는 모델로 예측한 각 keypoint의 x좌표, y좌표이다. hyperparameter 조정을 위해 epoch, learning rate을 다양한 값으로 설정한 뒤 각각의 상황에서 validation set의 이미지들의 RMSE를 구하였다. 그리고 모든 이미지의 RMSE의 평균을 구하여 그래프(그림 5)로 나타내었다.

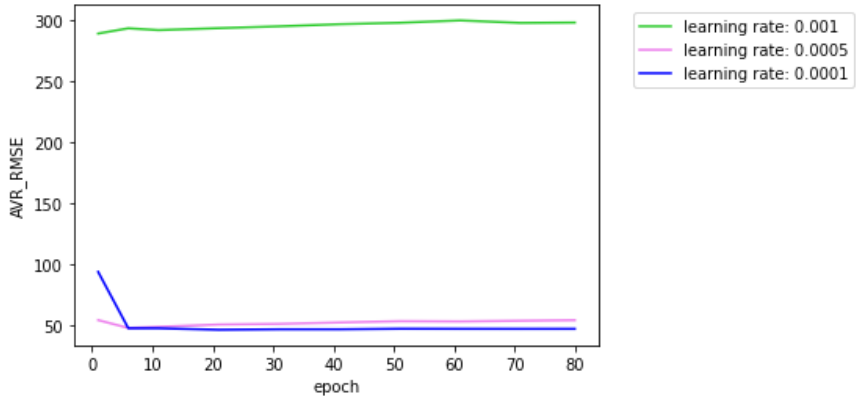


그림 5. learning rate와 epoch별 평균 RMSE

본 모델에서 조정한 hyperparameter는 다음과 같다.

- learning rate

  learning rate를 0.005, 0.001, 0.0001로 하여 학습을 진행하고 validation 과정을 거친 후, 가장 낮은 평균 RMSE를 보였던 learning rate를 파악한다. 그림 5를 보면 learning rate가 0.0001일 때가 전체적으로 가장 낮은 평균 RMSE를 보임을 알 수 있다.

- loss

  loss function으로는 각 skeleton 좌표의 오차가 큰 경우에 더 큰 penalty를 부여하기 위해 JointsMSELoss를 사용한다.

- optimizer

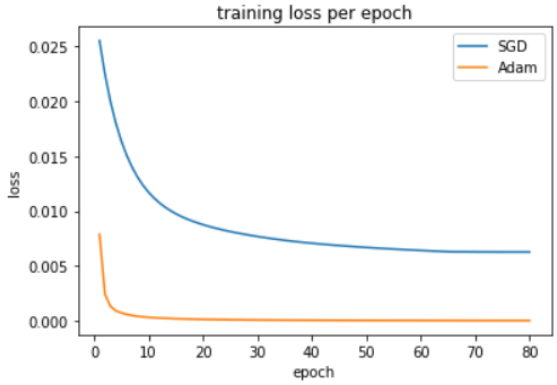


그림 6. Optimizer별 epoch당 training loss

  보편적으로 많이 쓰이는 SGD와 Adam을 비교해본 결과 Adam optimizer가 더 우수한 성능을 보였다. 따라서 최종 모델의 optimizer는 Adam으로 사용한다.

- pretrained_weight_path

  pre-trained된 HRNet의 가중치를 사용한다.

- model_channels

  model의 output 채널 수는 48개로 조정한다.

- model_nof_joints

  판별할 skeleton key point 개수는 24개로 한다.

- epochs

epoch은 80까지 진행되며 각 1, 6, 11, 21, 31, 41, 51, 61, 71, 80 epoch마다 학습된 가중치가 적용된 모델을 저장한다. 학습이 마무리가 되면 각 epoch에 대한 가중치가 저장된 모델을 불러오고 validation set을 이용하여 해당 모델의 validation이 진행된다. 그림 6은 앞서 언급한 하이퍼 파라미터를 확정하여 학습한 모델을 epoch 80까지 학습을 진행하였을 때 epoch별 평균 RMSE를 보여준다. epoch이 증가할수록 train RMSE의 평균은 계속해서 감소하는 경향을 보이고 validation RMSE의 평균은 21 epoch 이전에는 감소하다가 21 epoch 이후에는 다시 증가하는 경향을 보인다. 이 사실로 미루어보아 21 epoch이후의 모델은 overfitting의 가능성이 있음을 알 수 있다. 따라서, epoch 21일때의 모델을 최종 모델로 채택하였다.

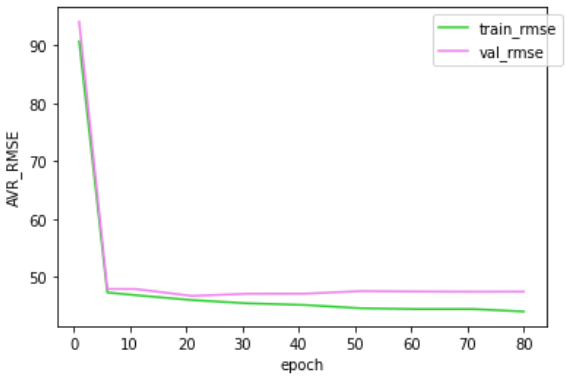


그림 7. epoch별 train RMSE의 평균과 validation RMSE의 평균

- exercise classification model
  - Model / Algorithm
    - 총 3,207개의 영상 중 batch_size(128)개의 영상을 임의로 추출하고 각 영상 별로 32개의 프레임, 각 프레임 별로 48개의 keypoint 데이터가 있습니다.
    - LSTM은 32개의 프레임을 하나의 sequence로 학습을 진행합니다.
    - 각 LSTM은 매 timestep마다 48개의 keypoint 데이터를 입력받아 마지막 time step의 output을 통해 운동의 종류를 예측합니다.
    - 최종적으로 14*5개의 class를 예측하므로 마지막에는 fully connected layer가 포함됩니다.
    - LSTM 모델 도식화

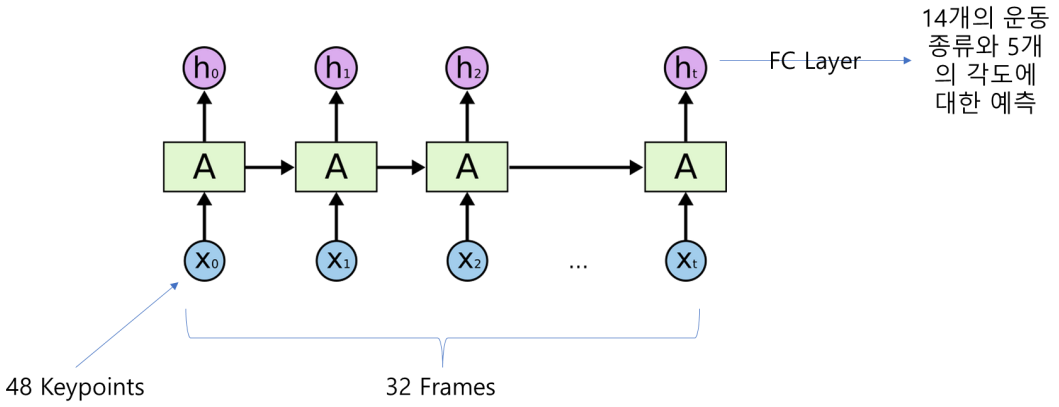


그림 8. LSTM 모델 도식화

    - many to one 방식의 LSTM 모델

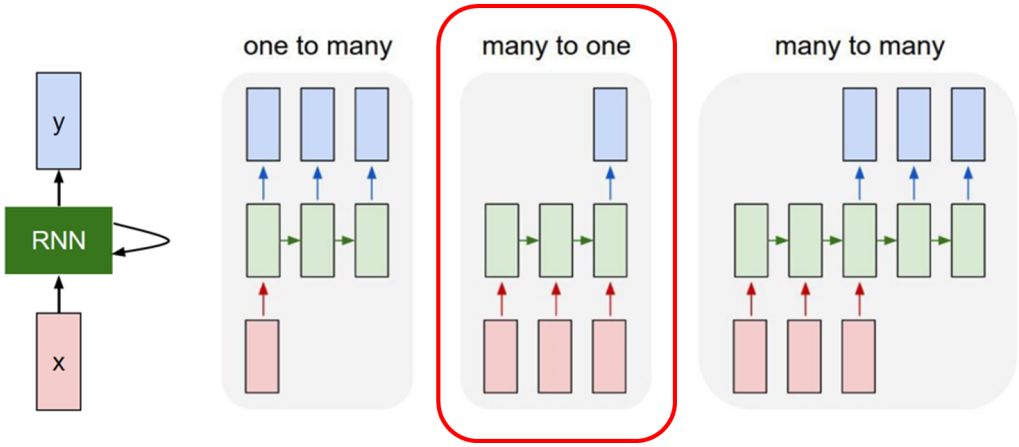


그림 9. many to one 방식

- Hyperparameter tuning
  - iteration 수 (1,000 vs 10,000 vs 20,000 vs 100,000)

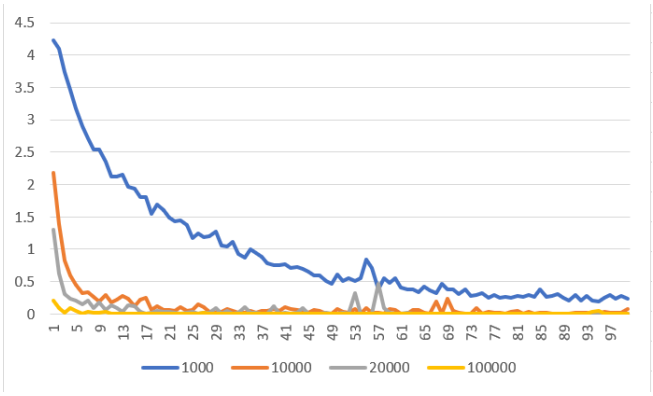
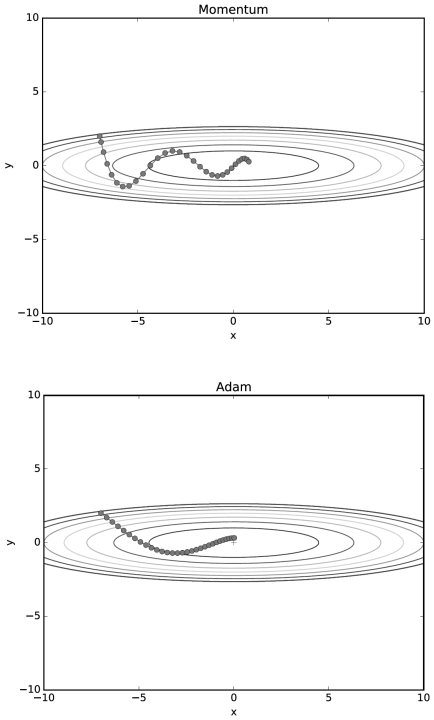


그림 10. iteration/100 당 loss 그래프

    - 처음에는 빠르게 결과를 보기 위해 iteration을 1000번으로 학습을 진행시켜 보았는데, 성능이 매우 저조했다. 이후 iteration을 10배씩 증가시켜본 결과 100,000번에서 가장 좋은 성능을 냈다. 위의 그래프는 각 iteration에서 (iteration/100) 번에 한 번씩 loss값을 출력한 그래프이다.
  - batch size (batch_size = 64 vs 128 vs 256)
    - batch_size를 64, 128, 256 으로 조정해가며 학습을 해 본 결과 128인 경우 가장 좋은 성능을 보였다.
  - image sequence 분할 (n_steps = 32 vs 16)
    - 각 영상별로 32개의 프레임으로 구성되어 있는데, 이것을 반으로 쪼개서 한 sequence에 담아 데이터의 양을 두 배로 늘리는 방법을 시도해보았다. 데이터의 증가로 성능이 개선될 것이라는 기대와는 달리 오히려 성능이 더 나빴다. 운동을 한 회당 나눈 것이 아니라 단순히 시간에 따라 나눈것이기 때문에, 각 sequence별 운동의 횟수가 달라지는 등 모델의 학습에 방해요인이 있을 것이라 판단하여 이후로는 32개의 프레임 전부를 한 sequence로 사용하였다.
  - Optimizer (SGD vs AdamW)
    -



- SGD는 모멘텀을 이용해서 로컬 미니마를 벗어나기 위한 장치를 마련했다. 그러나 모멘텀 역시 미니멈 근처에서 멈추기 힘들다는 문제를 안고 있다.
- Adam은 로컬 미니마를 벗어나는 데 필요한 모멘텀 방식과 learning rate가 adaptive한 RMSProp 방식의 장점을 취합해 놓은 Optimizer이다. 그러나 L2 regularization 텀이 추가된 loss func를 Adam을 이용해서 최적화 할 경우, 의도와 달리 일반화 효과과 떨어질 수 있다고 한다. 그래서 Optimizer 후보군으로 Adam 대신 AdamW를 선정하였다.
- 참고한 모델에서 사용한 SGD 대신 AdamW를 사용해보았더니 iteration 초기 loss값이 5배 이상 줄어들고 최종 loss는 100배 이상 감소하는 결과를 확인하였고, train & test accuracy 역시 1% 이상씩 증가하였다. 이후로 optimizer로 AdamW를 사용.

- LR Scheduler: None Sheduler

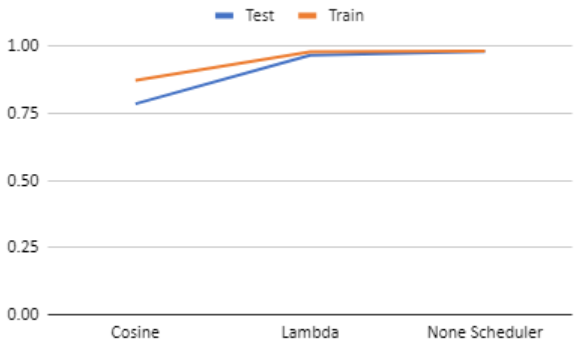


그림 11. Scheduler 별 train/test accuracy 그래프

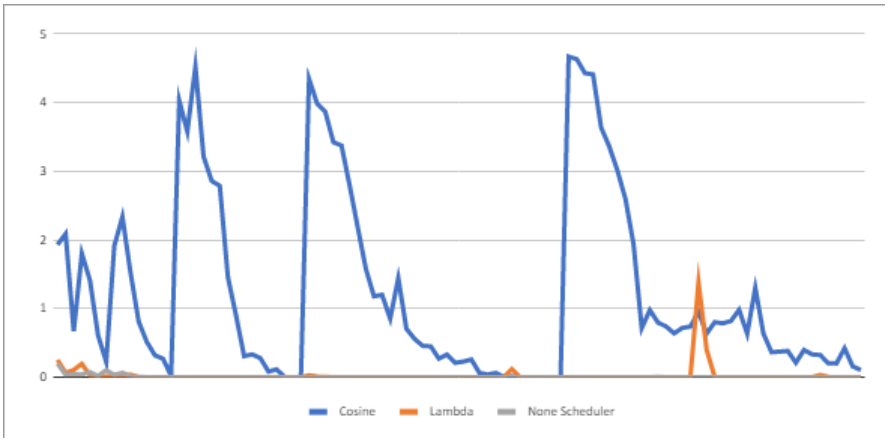


그림 12. Scheduler 별 loss 그래프

위의 그래프는 iteration=100,000번 돌리는 동안 1000번 단위로 한 번씩 loss 값을 기록해서 나타낸 그래프이다. Learning rate scheduler로써 단조감소하는 LambdaLR과 주기를 갖고 증가와 감소를 반복하는 CosineAnnealingLR을 두 가지를 모두 사용해서 모델을 구현해보았으나, 두 경우 모두 scheduler를 사용하지 않을 때 보다 최종 accuracy가 낮아서 최종모델은 learning rate scheduler를 포함시키지 않았다.

- 최종 선정 모델 설명
  - 최종 선정 모델은 각 하이퍼 파라미터 중 Test Accuracy가 가장 높은 최적의 파라미터들을 모두 사용하는 모델로 선정하였다.(iteration=100,000 / batch_size=128 / n_steps=32 / Optimizer=AdamW / LRScheduler=None)

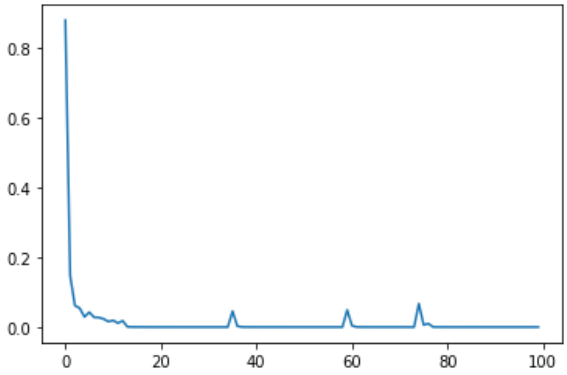


그림 13. 최종 선정 모델 epoch-loss 그래프

4. Evaluation

- skeleton extraction model

  최종 모델에 test set을 적용하여 확인해본 결과 평균 RMSE는 **약 35.86**이었는데, 이는 앞서 언급했던 성공 기준인 57.6 이하이므로 목표치보다 초과 달성하였음을 확인할 수 있다.

  아래의 그림은 skeleton key point의 예측 결과 예시이다.

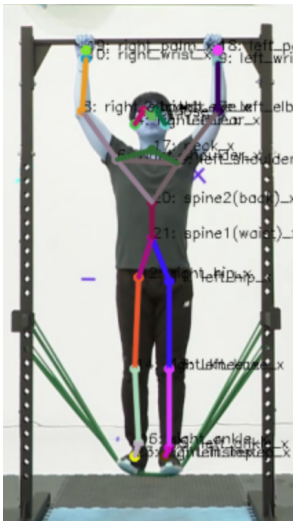


그림 14. skeleton extracted image

또한, yolo3에서 사람을 찾지 못하여 학습을 못한 데이터도 있었지만 전체 데이터 양 중, 대략 0.2%에 해당하는 데이터 양이었기 때문에 데이터 손실은 크지 않았다.

- novel or unique findings, patterns, insights

  test set을 통해 각 이미지의 RMSE를 측정한 결과 다른 운동들에 비해 특정 종류들의 운동들(크런치, 바이시클 크런치, 시저크로스, 힙쓰러스트, 푸시업, 니푸시업)의 RMSE 값이 100 이상으로 유독 높은 것을 발견할 수 있었다. 이러한 운동들의 이미지를 확인하여 보니 운동 기구나 다른 물체로 인해 가려져서 또는 촬영 각도로 인해 신체 부위의 일부가 보이지 않는 이미지들 이었다.

- exercise classification model
  - 머신 러닝 분류 모형의 평가 방법으로는 정확도(Accuracy)와 Heat map을 사용하였다. 평가를 위해 각 운동명과 해당되는 운동의 Accuracy를 출력하였고, 모든 운동의 정확도를 가시적으로 확인할 수 있는 Heat map을 사용하였다. Heat map과 출력된 Accuracy를 바탕으로 미루어봤을때, Accuracy가 1인 운동들로는 덤벨 체스트 플라이, 라잉 레그 레이즈, 바이시클 크런치, 푸시업, 딥스, 로잉머신으로 나타났다. 또한 Accuracy 기준 하위 5가지 운동으로는 크런치, 바벨 로우, 바벨 런지, 바벨 데드리프트, 프런트 레이즈가 나왔으며 각 운동의 정확도는 0.9400, 0.9487, 0.9565, 0.9714, 0.9762이다.
  - novel or unique findings, patterns, insights

    운동 종류 분류에 대한 정확도가 1이 아닌 운동은 14가지 운동 중에서 8가지가 확인된다. Accuracy가 1이 아닌 운동 종류들을 열거해보면 프런트 레이즈, 바벨 로우, 바벨 데드리프트, 바벨 스쿼트, 바벨 런지, 크런치, 힙쓰러스트, Y - exercise이다. 그 중에서 가장 Accuracy가 낮은 운동인 크런치의 Accuracy는 0.9400으로 절대 낮은 수치가 아니라는 것을 알 수 있다. 이러한 사실들로 미루어 보어 본 팀에서 설계한 모델은 성공적으로 운동 종류를 분류한다는 것을 방증하고 있다. 더불어 Heat Map을 자세하게 보면 크런치의 몇몇 동작에 대한 좌표가 바이시클 크런치로 분류된다는 것을 확인할 수 있다. 하지만 그렇게 판단한 좌표가 두드러지 않은 것은 아니고, 실제로 해당 운동 결정 모델의 input data인 csv 파일을 살펴보면 sequence 단위로 keypoint가 변화하는 추이가 비슷하다는 것을 확인할 수 있다.

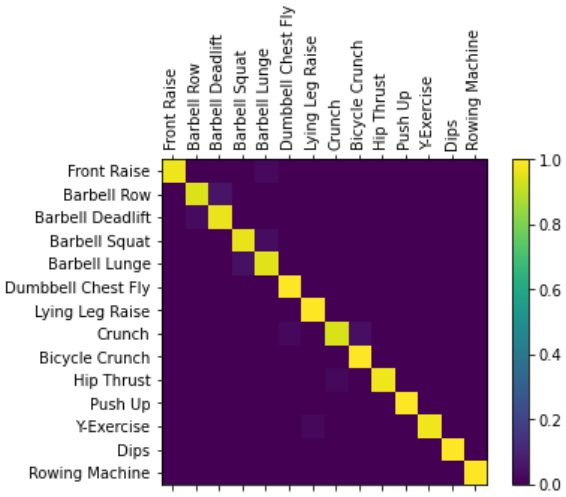


그림 15. 최종 선정 모델의 각 운동별 정확도에 대한 Heat Map

- External data 테스트
  - 외부영상을 이용한 예측 결과

```
(0m 0s) external_dataset/12/01.jpg / 바벨 로우 ✓
(0m 0s) external_dataset/14/01.jpg / 바벨 데드리프트 ✓
(0m 0s) external_dataset/15/01.jpg / 바벨 런지 X (['바벨 스쿼트'])
(0m 0s) external_dataset/16/01.jpg / 바벨 런지 ✓
(0m 0s) external_dataset/22/01.jpg / 크런치 ✓
(0m 0s) external_dataset/23/01.jpg / 바이시클 크런치 ✓
(0m 0s) external_dataset/25/01.jpg / 덤벨 체스트 플라이 X (['힙쓰러스트'])
(0m 0s) external_dataset/27/01.jpg / 푸시업 ✓
(0m 0s) external_dataset/29/01.jpg / Y - Exercise ✓
(0m 0s) external_dataset/30/01.jpg / 덤벨 체스트 플라이 ✓
(0m 0s) external_dataset/34/01.jpg / 바벨 런지 X (['딥스'])
(0m 0s) external_dataset/39/01.jpg / 바벨 로우 X (['케이블 크런치'])
(0m 0s) external_dataset/41/01.jpg / 로잉머신 ✓
(0m 0s) external_dataset/9/01.jpg / 프런트 레이즈 ✓
```

유튜브를 통해 수집한 외부 영상을 우리의 모델에 넣어본 결과 keypoint 데이터를 비교적 잘 추출하였고, 그 keypoint 데이터를 통해 운동 종류를 예측한 결과 14개의 운동에 대해서 10개의 운동을 예측 성공하였다. 아무래도 내부 데이터를 이용했을 때보다(약 97%)는 눈에 띄게 정확도가 낮아졌지만 나름 유의미한 예측(약 71%)을 했다고 볼 수 있다. 그러나 아무래도 서비스로 출시하기에는 부족한 성능이고, 이 부분에 대한 고찰은 Conclusion에서 후술하겠다.

5. Conclusion

- Summary

  먼저 AI허브에서 얻은 운동 영상 이미지 dataset 을 전처리 후 train, validation, test set 으로 나누었다. 다음으로 train set을 통해 hrnet 을 이용한 스켈레톤 추출 모델을 학습하였고 validation set 을 통해 hyperparameter 를 조정하였다. 그리고 최종적으로 결정한 스켈레톤 추출 모델을 사용하여 test set의 평균 RMSE 를 구하여 모델을 평가하였다. 다음으로 올바른 운동 자세에 해당하는 약 32만 장의 이미지에 대해 최종 모델을 사용하여 keypoints의 좌표를 출력하였다. 출력한 keypoints에 대해 추가적으로 전처리를 진행한 후 미니배치를 통해 임의로 뽑은 데이터를 lstm을 활용한 운동 분류 모델의 input 으로 넣어 여러번 반복하여 학습 시킨다. 학습된 모델을 사용하여 외부 데이터의 운동 종류와 촬영 각도에 따라 분류해내고 최종적으로 운동 종류만을 예측한다. 결과로는 이미지의 경로와 예측한 운동 종류를 반환하여 출력한다.

- Main advantages of our models
  - 신체 부위가 잘 보이는 이미지의 keypoint 예측에서는 높은 정확도를 보인다.
  - 여러 각도에 대한 데이터를 통해 다양한 방향에서 찍은 영상에 대해서도 올바른 운동 분류가 가능하다.
- Discussion about limitation
  - skeleton extraction model에서 전체 dataset의 크기(약 550만 장의 이미지)에 비해 실제 사용한 dataset(약 7500장)의 크기가 너무 작다.

    학습할 때 사용한 colab pro+의 런타임 시간은 최대 12시간 정도이다. 그러나 train set의 크기가 지금보다 커질 경우 학습 시간이 12시간을 넘어 학습에 실패할 가능성이 존재했다. 그러한 이유로 인해 전체 dataset 대비 낮은 비율의 dataset을 사용할 수 밖에 없었다.
  - skeleton extraction model에서 예측에 실패한 경우

그림 16. 예측에 실패한 이미지

위의 그림은 skeleton extraction model에서 특정 keypoint 예측에 실패한 경우를 보여준다.

1. 첫 번째 경우는 외부 데이터를 사용했을때 발생하였다. 이 외부 데이터에는 여러 사람이 존재하지만 본 모델에서는 한 사람만 찾도록 되어 있기 때문에 예측에 실패했다고 판단된다.
2. 두 번째 이미지의 경우 신체 부위의 일부가 물체에 의해 가려졌기 때문에 예측에 실패했다고 판단된다.
3. 가장 큰 이유로 위 두 가지 상황을 나타내는 train dataset이 충분하지 못했기 때문이라고 생각한다.

  - 비슷한 운동 동작 분류 시, 다시 말해서 관절 포인트의 변화가 비슷한 운동들을 분류할 때는 낮은 정확도를 보인다. 예를 들어 바벨로우와 덤벨 벤트오버 로우의 경우 운동 기구만 다를 뿐 관절 포인트의 움직임은 정확히 일치하는 운동이다. 이러한 부분은 만약 우리의 모델을 서비스화하게 된다면 그전에 image semantic segmentation을 통해 운동기구를 인식해서 그에 따라서 운동을 분류하는 기능을 추가해야할 것으로 보인다.
  - 앞서 언급했듯이, 서비스화 하기에는 외부 데이터를 활용했을 때의 운동 종류 예측 정확도가 매우 부족하다.
    1. 모델을 학습시키는 데에 5개의 각도만으로 한정을 했기에 촬영 각도에 대한 유연성이 부족했을 것이라 판단된다.
    2. 또한, 운동 기구에 신체의 일부가 가려지는 경우 keypoint 추출이 부정확해지는 점 역시 예측 결과에 영향을 미쳤을 것이다.

    이미지 데이터를 각도를 더 다양화하고 데이터의 양을 증가시키는 방안과 2D keypoint로 학습시킨 모델이 다양한 촬영 각도에 비교적 취약하므로 영상으로부터 3D keypoint를 추출해내서 학습시키는 방안이 해결책이 될 수 있을 것으로 보인다.
  - 서비스화 하기에는 기능이 부족하다. 추후 운동 횟수 측정 기능까지 추가된다면 서비스화도 고려해볼 수 있을 것으로 생각된다.