1. 데이터 추가 수집
   1. 180도까진 X
   2. 집는 과정까지 포함
2. 데이터 사진 프레임 따기 → 진행중
3. data augmentation
4. 젓가락 object detection → 필요성???
   1. yolox 는 적절하지 않음
   2. image segmentation으로
   3. sementic segmentation : <https://paperswithcode.com/task/semantic-segmentation>
5. 손가락 인식 - open api : hand tracking - mediapipe
   1. convLSTM (비교용) > FC-LSTM (1차원)
      1. 일기예보
      2. 이상행동탐지 (anamoly detection)
      3. <https://colab.research.google.com/github/keras-team/keras-io/blob/master/examples/vision/ipynb/conv_lstm.ipynb>
      4. <http://www.cs.toronto.edu/~nitish/unsupervised_video/> :moving mnist
      5. 올바른 손가락 입력한 후 모델 학습 그럼 올바르지 않은 것 입력하면 음,,, 집는 거만 문제가 아니라 손모양도 문제인디,,,
      6. 정확도라는게 있는가? batch loss는 있음
      7. <https://dacon.io/codeshare/2544> : 해빙 예측
      8. human action recognition의 경우 wearable device에서의 데이터 활용 경우가 많음 이미지아님
      9. <https://keras.io/examples/vision/conv_lstm/>
6. rule-based classifier ; binary image classification(image 분류)
   1. deep learning(CNN) : vggnet, alexnet
   2. machine learning : SVM

* <https://medium.com/analytics-vidhya/image-classification-using-machine-learning-support-vector-machine-svm-dc7a0ec92e01>
* <https://www.youtube.com/watch?v=nK-2k_ENgEc>
  1. CNN + rule-based classifier : <https://paperswithcode.com/paper/auto-deeplab-hierarchical-neural-architecture>

1. VIDEO
   1. object tracking
   2. swin transformer : object detection <https://www.youtube.com/watch?v=L3sH9tjkvKI>
   3. video swin transforemr : [**https://sjpyo.tistory.com/85**](https://sjpyo.tistory.com/85)
   4. <https://github.com/shwetabhardwaj44/EfficientVideoClassification_Youtube8M>
   5. [Efficient Video Classification Using Fewer Frames](https://paperswithcode.com/paper/efficient-video-classification-using-fewer) (LSTM)
   6. video action recognition - fine-grained 분류 문제
      1. Something-Something V1(Sth-Sth V1) 데이터셋 : 손동작 분류

<https://paperswithcode.com/dataset/something-something-v1>

* + 1. <https://seoilgun.medium.com/video-action-recognition-c322db5f1f13>

1. 앱 UI 디자인

1.augmentation → hand tracking → 젓가락 인식 → classfication 모델 학습

landmark data + y 값

normalization(회전-각도)

2.data normalization - convlstm 모양 변화 (input을 몇장으로 할것인가) + y 값

3.data normalization - 랜드마크 + lstm + y 값

-DB 만들때 새끼손가락빼기

-어떤 점이 연결되어있는지로 DB 만들기

ex) 손큰사람 손작은사람, 손 쏠려있는 것, 왼손잡이/오른손잡이

-

-손가락 인식 논문 확인

-handtracking 적용

-augmentation

8/4(수) 기존 손의 모습 없애고 handtracking이랑 젓가락만 남겨서 학습 (배경 black)

1. 처음 파지법부터 잘못된 경우
2. 파지법은 옳으나 집을때 이상한 경우

내일 질문 할 것

1. 작년에 수화 했던 팀. 핸드트래킹을 어떻게 변형해서 썼는지? + 수화했던 팀 학습 방법을 어떻게 했는지? convolution lstm을 씀
2. 핸드트래킹을 더 학습시켜서 정확도를 높일 수 있는지? 핸드트래킹 모델을 만들어서 더 학습시킬 수 있다.
3. 큰그림은 핸드트래킹을 어떻게 활용할지. 어떤 과정 속에서 활용할지. 상속
4. Convolution LSTM을 활용한다고 하면 prediction을 어떻게 둬야할지. 바른 자세, 바르지 않은 자세
5. (YOLO가 X였던 이유가 젓가락을 박스로 인식해서 안된다고 했던것

-> 어차피 segmentation으로 해도 손 관절 landmark와 같이 젓가락 landmark를 정해줘야 한다면, box로 인식 후 landmark를 제대로 찍는거랑, segmentation 후 landmark를 찍는거랑. 차이가 있을까요?) 정확하게 하려면 segmentation이 낫다. 하지만 공부할게 많고 못만들 것 같다. 할려고 하면 segmentation이 맞다.

1. 젓가락 인식 AI 만들때 학습 데이터를 어떻게 할까?
2. 전체적인 모델 빅픽쳐에 대해 조언 구하기.
3. 올바른 젓가락질을 학습을 통해 교정하기 A냐 B냐는 되지만 A가 어떻냐는 힘들다.
4. 손 관절, 젓가락 landmark 통해 교정하기

8. 젓가락 파지만으로 얼추 가능하지만 전부는 가능 X

-> 젓가락’질’ 까지 고려해야함

-> 학습

9. 라이브러리 소스코드를 뜯어보는 방법.

-> ex) handtracking의 정확도를 높이기 위해 더 뜯어보려고 했으나, 손 인식, landmark 찍는 알고리즘 등은 결국 불러와서 시행하길래, 그 불러오는 함수나 클래스를 어떻게 뜯어볼 수 있는지.

landmark 몇개 찍을 것인지, 잘찍은건지

feature extraction

올바른 젓가락질 기준 (괄호 속 숫자는 landmark 번호)

1. 밑에 젓가락은 2개 포인트에서 고정됨.

첫번째 포인트: 약지의 첫 관절(7), 두번째 포인트: 엄지와 검지 사이의 골짜기(2번 근처)

밑에 젓가락은 상하좌우 운동 X. 고정됨.

1. 위에 젓가락은 3(+1)개의 포인트에서 고정됨.

첫번째 포인트: 중지의 tip(12)과 첫 관절(11) 사이(관절에 더 가까움)에 고정

두번째 포인트: 검지의 두번째(6), 세번째(5) 관절 사이에 젓가락 지지

세번째 포인트: 엄지 손가락(4, 3 사이부분)으로 첫번째, 두번째 포인트 중간을 누르면서 고정

(네번째 포인트: 검지의 tip(8) 부분을 젓가락과 접촉시킨다. 이것은 젓가락을 아래로 움직일때 사용하는 부분.)

1. 아래 젓가락은 움직임 X. 위 젓가락은 위로 움직일 땐 중지를 통해(2.의 첫번째 포인트), 아래로 움직일 땐 검지(2.의 네번째 포인트)를 통해 움직인다.

무엇이 올바른 젓가락질과 올바르지 않은 젓가락질이 구분하는 특징이 될까?

1. 올바른 젓가락질 -> 중지와 약지 사이의 거리가 다른 손가락에 비해 큼. 즉, 엄지, 검지, 중지 사이의 거리와 약지, 새끼손가락의 거리가 중지와 약지 사이의 거리에 비해 작음.  
   ex) 중지-약지 tip의 거리/검지-중지 tip의 거리 의 값. 크면 올바른, 작으면 올바르지 않은.
2. 검지-중지의 각도 차이와 중지-약지의 각도 차이를 활용  
   이때 기울기 차이를 사용하면 안됨  
   sin 함수로 각도에 대한 정비례 관계를 구하던가, 아니면 log값을 취한 것을 사용해야할듯 하다.  
   이때 사용해야 하는 포인트는 검지(6, 7), 중지(10, 11), 약지(14, 15) 생각중..  
   먼저 포인트를 이용한 기울기 계산 -> 세개의 기울기값의 차이를 구함 -> 기울기 차이는 tan -tan 값이므로 이것을 삼각함수 공식을 이용해 사잇각을 sin 값이나 각도(°)로 표현 -> sin 값이나 각도(°)값을 하나의 feature로 사용  
   하지만 올바르지 않은 젓가락질이 워낙에 다양한 형태이기 때문에 가장 대표적인 올바르지 않은 젓가락질에 잘 적용될듯 하다.

Normarlization 방법이 무엇이 될 수 있을지

1. 손 크기를 표준화 시킨다  
   어떤 부분을 기준으로 할지 앞으로 토의
2. 한개의 landmark를 (0,0)으로 고정시키고 다른 점들의 위치를 수정한다.

---------------------------------------------------------------------------------------------------------------------------

17 18 19 20

각자 염려되는 것

1. 생각 보다 코드 수정의 어려움

ex) 영상 입력을 array에 바꿔야 하는 것 등...

1. 우리가 너무 다운되는것 아닌가? 낙담에 대해 좀 염려

ex) 선생님의 말 한마디에 너무 휘둘리지 않도록

1. 선생님께 물고 늘어져야 할 것.
2. 프로젝트 갈아 엎는 것에 대해 너무 두려워 하지 말기

-> 배운것이 아닌 분야라 힘들어 하는 것이니, 배운 것으로 해도 좋다는 마인드로

1. 침묵을 두려워하지 않기, 말하는것도 두려워하지 않는것

시계열/RNN/LSTM

1. RNN (Recurrent Neural Network)은 시계열 또는 자연어와 같은 시퀀스 데이터를 모델링하는 데 강력한 신경망 클래스이다.
2. tensorflow 시계열(ts) 예시 : <https://www.tensorflow.org/tutorials/structured_data/time_series>

=> 이전의 데이터를 바탕으로 앞으로의 값을 예측함

----------------------------------------------------------------------------------------------------------------

=> 한 동영상!의 데이터를 바탕으로 class를 분류함

1. Code Error 나 가이드가 없는 코드는 어떻게 해야합니까 ㅠㅠ