

2021년 고양시 태권도 동작 데이터 데이터 톤 참고자료

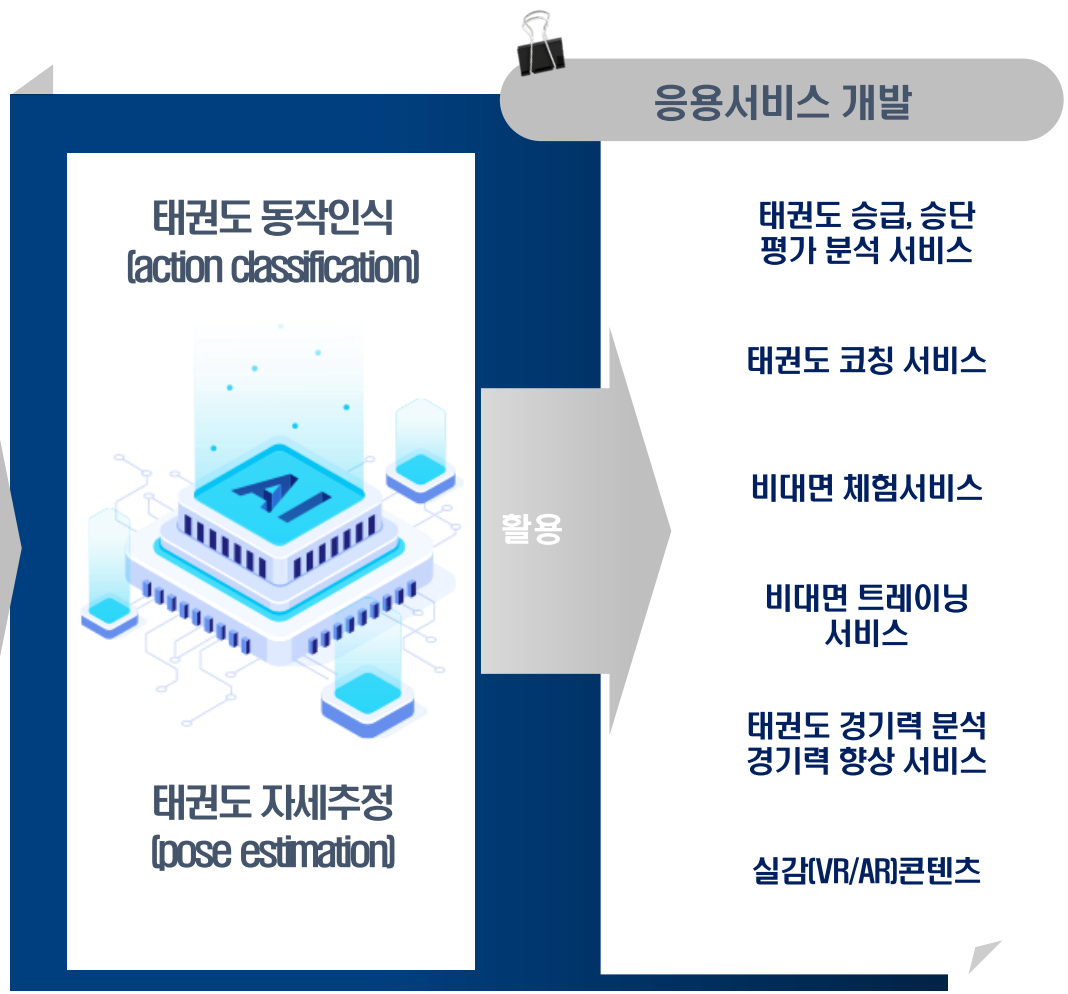
2021.11.15

1. 인공지능 모델 개발 개요

❖ 인공지능 모델 개발 목표

- 태권도 영상 데이터(입력) ▶ 태권도 동작인식(출력 : 동작명) 모델 개발
- 태권도 동작인식, 자세추정 모델을 통하여 최종적으로 승급승단, 코칭서비스 등의 개발에 활용

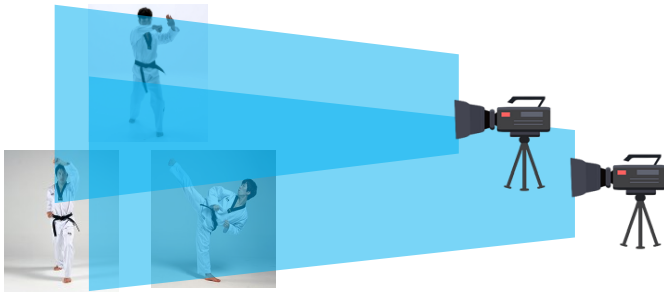
태권도 품새영상
태권도 경기영상



2. 태권도 데이터 구성

❖ 태권도 데이터 구성

원시데이터 - 영상데이터



- 태권도의 품새 영상을 8개 방향에서 촬영
- 품새 : 태극1장 ~ 8장

원천데이터

동작시작점

동작중간점


동작종료점



- 품새를 구성하는 개별/단위동작을 동작시작, 중간, 종료점 이미지로 분할

3. 태권도 학습데이터 파일명 규칙

❖ 태권도 학습데이터 파일명 규칙

 00000005-S01.jpg

▪ 시작점 라벨링 데이터

 00000005-M01.jpg

▪ 중간점 이미지 데이터

 00000005-S01.jpg 00000005-E01.jpg

▪ 종료점 라벨링 데이터

4. 학습데이터 파일내용 (예시)

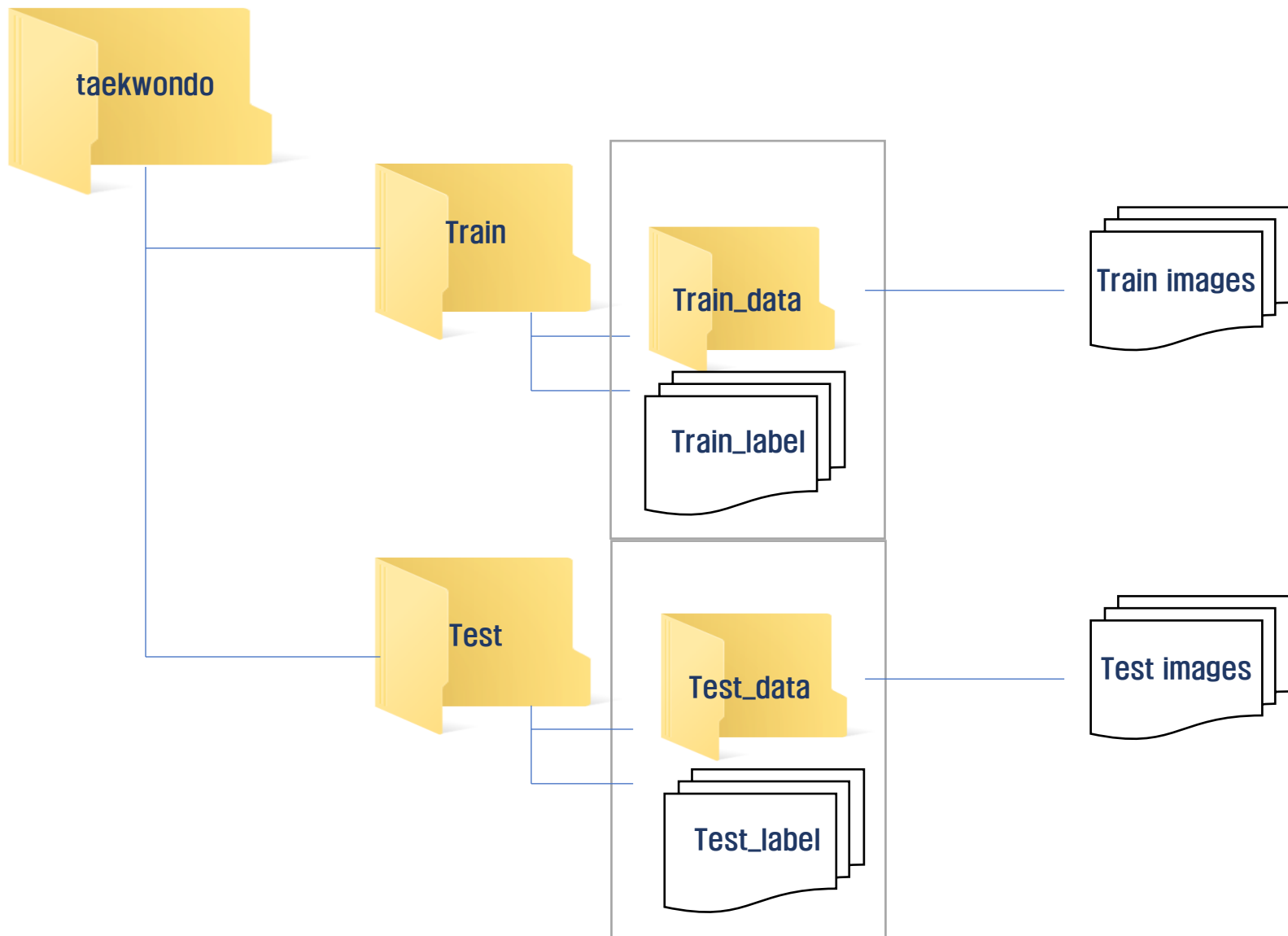
```
00000403 0
00000404 0
00000405 0
00000406 0
00000407 0
00000408 0
00000409 0
00000410 0
00000411 0
00000412 0
00000413 0
00000414 0
00000415 0
00000416 0
00000417 0
00000418 0
00000419 0
00000420 0
00000421 0
00000422 0
```

· 이미지 셋 이름

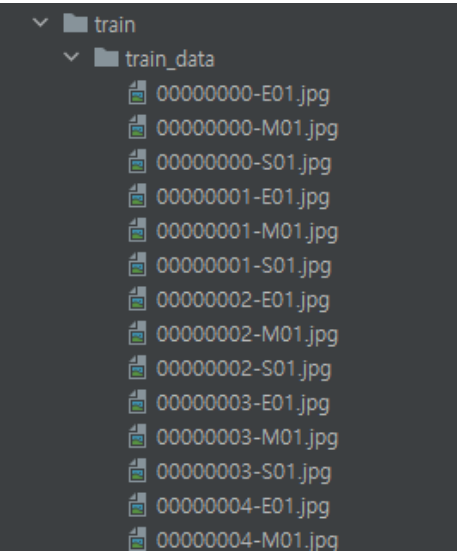
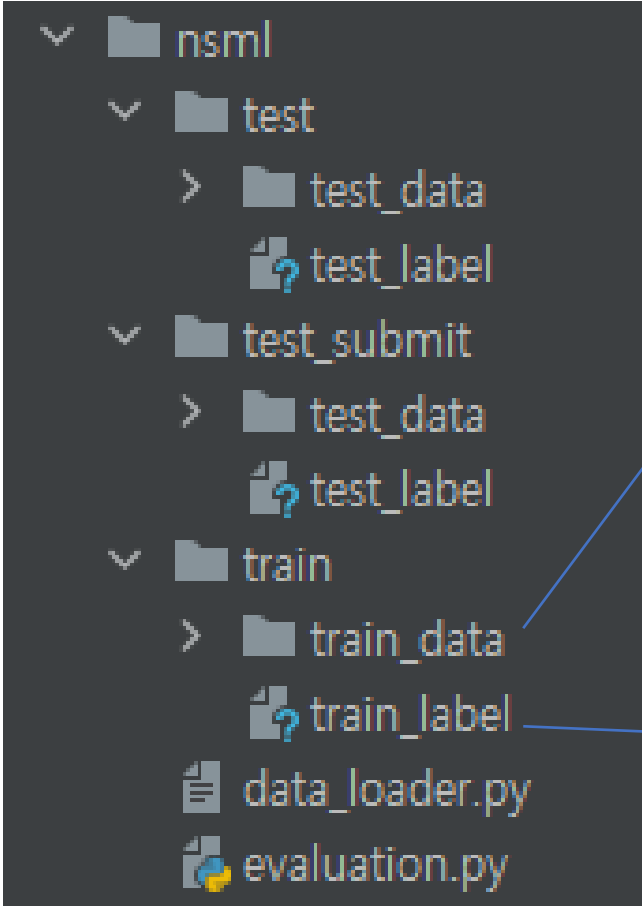
· 동작 라벨 번호

{ '기본준비': 0,
 '뒷굽미하고 바깥막기': 1,
 '뒷굽미하고 손날바깥막기': 2,
 '앞굽미하고 당겨지르기': 3,
 '앞굽미하고 바깥손안막고 지르기': 4,
 '앞서고 지르기': 5,
 '앞차고 뒷굽미하고 바깥막기': 6,
 '앞차고 앞굽미하고 지르기': 7,
 '앞차고 앞서고 아래막고 지르기': 8,
 '열서고 메주먹내려치기': 9 }

5. 데이터 저장구조



6. 데이터 저장구조 예시



1	00000403	0
2	00000404	0
3	00000405	0
4	00000406	0
5	00000407	0
6	00000408	0
7	00000409	0
8	00000410	0
9	00000411	0
10	00000412	0
11	00000413	0
12	00000414	0
13	00000415	0
14	00000416	0

7. 2021년 태권도 데이터톤 진행용 데이터

동작명	사용되는 품새	데이터수량 / 비율			
		Training set	Test set	Validation set	합계
기본준비	태극1장 1품-18품	777 / 77%	111 / 11%	111 / 11%	999
앞서고 지르기	태극1장 3품-5품 태극1장 8품-10품	777 / 77%	111 / 11%	111 / 11%	999
뒷굽이하고 손날바깥막기	태극3장 8품-10품 태극5장 10품-12품 태극8장 21품-23품	777 / 77%	111 / 11%	111 / 11%	999
앞차고 앞서고 아래막고 지르기	태극3장 20품-21품	777 / 77%	111 / 11%	111 / 11%	999
뒷굽이하고 바깥막기	태극4장 9품-11품	777 / 77%	111 / 11%	111 / 11%	999
옆서고 매주먹내려치기	태극5장 3품-5품	777 / 77%	111 / 11%	111 / 11%	999
앞굽이하고 바탕손안막고 지르기	태극6장 19품-20품	777 / 77%	111 / 11%	111 / 11%	999
앞차고 앞굽이하고 지르기	태극6장 8품-10품 태극4장 7품	777 / 77%	111 / 11%	111 / 11%	999
앞차고 뒷굽이하고 바깥막기	태극6장 3품-5품-14품-16품	777 / 77%	111 / 11%	111 / 11%	999
앞굽이하고 당겨지르기	태극8장 7품-9품	777 / 77%	111 / 11%	111 / 11%	999

8. 베이스라인 시스템

- Environment
 - Docker Image : nvcr.io/nvidia/pytorch:20.03-py3
 - PIP
 - Tqdm
 - Pandas
 - pillow

9. DATA LOADER

- **Data Loader**

주어진 학습 또는 테스트 데이터를 불러오기 위해 Pytorch의 DataLoader 클래스를 상속 받는 클래스를 만들어 한번에 데이터를 불러옴.

- 시작, 중간, 끝부분에 해당되는 이미지를 불러와 (9,W,H) 형태로 병합

```
def __getitem__(self, idx):
    img_origin_path = list(self.img_labels.keys())[idx]
    img_path_S1 = os.path.join(self.img_dir, img_origin_path) + '-S01.jpg'
    img_path_M1 = os.path.join(self.img_dir, img_origin_path) + '-M01.jpg'
    img_path_E1 = os.path.join(self.img_dir, img_origin_path) + '-E01.jpg'
    image_id = img_path_S1.split('-')[0].split('/')[-1]
    image_S1 = Image.open(img_path_S1)
    image_M1 = Image.open(img_path_M1)
    image_E1 = Image.open(img_path_E1)
    label = self.img_labels[img_origin_path]

    if self.transform:
        image_S1 = self.transform(image_S1)
        image_M1 = self.transform(image_M1)
        image_E1 = self.transform(image_E1)
        image = torch.cat((image_S1, image_M1, image_E1))
    if self.target_transform:
        label = self.target_transform(label)
    label = int(label)
    return image_id, image, label
```

10. 모델예제

- Directory Structure
- Train – images
 - -- labels
- Test – images
 - -- labels

11. 평가지표

- Accuracy
 - 전체 데이터 중 정확하게 예측한 데이터의 비율

$$\frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives}$$

		예측 클래스 (Predicted Class)	
		Negative(0)	Positive(1)
실제 클래스 (Actual Class)	Negative(0)	TN (True Negative)	FP (False Positive)
	Positive(1)	FN (False Negative)	TP (True Positive)