

태권도 초심자를 위한 AI의 DataSet 구축

조규철*, 김주연^o

*인하공업전문대학 컴퓨터정보과,

^o인하공업전문대학 컴퓨터정보과

e-mail: kccho@inhac.ac.kr*, kimjuyeon0912@gmail.com^o

Dataset Construction of Taekwondo Beginner AI

Cho Kyu Cheol*, Kim Ju Yeon^o

*Dept. of Computer Science, Inha Technical College,

^oDept. of Computer Science, Inha Technical College

● 요약 ●

세계 태권도 연맹은 국제 축구 연맹의 가입국과 동일한 수의 가입국을 보유할 만큼 태권도는 점점 더 세계적으로 나아가고 있다. 하지만 태권도의 교육방법은 예전과 다르지 않다. 도장의 관장이나 사범이 직접 자세를 눈으로 보고 판단하여 지도해야 한다. 본 연구는 기술이 발전하고 변화함에 따라 태권도를 조금 더 다양하고 흥미롭게 배울 수 있는 방법을 개발하고자 진행하였다. 본 논문에서는 피사체 모델을 촬영하여 이미지를 추출하고 이미지에서 사람의 관절 KeyPoint를 라벨링 한 후 이를 바탕으로 COCO 형식의 DataSet을 만들어낸다. 이후 이 DataSet을 기계에 학습을 시킨다면 초심자를 위한 교육용 태권도 AI가 만들어질 수 있다. 또한, 기계학습 이후 이 AI를 실제 교육현장에 적용하여 교육과정에 직접 사용할 수 있으며 이 AI를 바탕으로 교육용 게임 개발 등 다양한 방법으로 활용할 수 있을 것이라고 기대한다.

키워드: 딥러닝(Deep Learning), KeyPoint, Pose Estimation

I. 서론

현재 미국을 비롯한 많은 나라에서 태권도가 유초등생 학부모들에게 큰 인기를 얻는 추세이다. 또한, 2021년 12월 20일 기준, 세계 태권도 연맹의 가입국은 국제 축구 연맹(FIFA)의 가입국과 같은 211개이고 2021년 하계 도쿄 올림픽의 태권도 종목의 메달 결과를 보면 전체 메달을 중주국을 포함한 몇몇 국가가 독점하는 것이 아닌 21개 국가에서 다양하게 가져갔다[1]. 이처럼 태권도는 점점 더 세계화가 되어가지만, 태권도의 교육방법은 예전과 다르지 않다. 도장의 관장이나 사범이 아이들의 자세를 하나하나 보고 지도한다. 태권도를 처음 배우는 학생들이 흥미를 잃기 쉽다. 본 연구는 태권도의 학습을 더 흥미롭고 효과적으로 배울 수 있는 방법을 고안하고자 시작하였다.

Pose Estimation은 이미지나 비디오에서 사람의 관절(Skeleton) 위치를 측정하고 Pose를 추정하는 문제이다. Pose Estimation은 2D Pose 좌표를 추정하는 2D Pose Estimation과 RGB 또는 RGB-D 이미지를 사용하여 metric space의 3D Pose 좌표를 추정하는 Pose Estimation이 있다.

Pose를 추정하는 방식에는 영상이나 이미지에서 사람을 먼저 감지한 후 Bounding Box 내의 사람의 Pose를 추정하는 방식의 Top-down

방식과 관절 부위를 먼저 감지하여 노드를 서로 연결해 모든 사람의 자세를 추정하는 Bottom-up 방식이 있다[2].

본 논문에서는 이미지에서 2D Pose Estimation의 x, y의 좌표만을 가지는 2D Pose를 추정할 DataSet 구축을 목표로 한다. 이를 위해 COCO 형식의 KeyPoint를 참고하며 DataSet 이미지의 전처리 과정을 거치고 실제 COCO 형식의 모델을 export 하는 것을 목표로 한다.

II. 관련 연구

AI Hub 사이트의 개방 데이터에 '피트니스 자세 이미지'라는 모델이 공개되어있다. 이 모델은 운동 동작과 상태가 정의된 AI 모델을 공개하고 있다. 구축 내용을 살펴보면 30종의 동작(5개의 운동 상태)으로 구성되어 있으며, 다양한 형태의 피사체로부터 데이터를 수집하였다. 200,000만 개의 Clip을 촬영하였고, COCO 17 Skeleton KeyPoint로 가공되어 1개의 JSON 파일로 제공한다. 이 모델은 홈 트레이닝, AR/MR, 의료분야 등에 활용이 가능하다. 이

모델은 일상생활의 동작뿐만 아니라 운동 동작 및 자세 등과 같은 세밀한 행동을 인식할 수 있는 AI를 개발하여 데이터 인프라를 구축하기 위해 제작되었다[3].

III. 태권도 초심자를 위한 AI DataSet 구축

1. 개요

본 연구의 DataSet KeyPoint는 ‘cocodataset.org’ 사이트에서 제공하는 COCO 형식의 JSON 포맷 파일을 기반으로 하며 ‘cocodataset.org’ 사이트에서 제공하는 COCO DataSet[4]의 17 Skeleton KeyPoint는 Table 1과 같다.

Table 1. 17 Skeleton KeyPoint

Nose	Right_Eye	Left_Eye
	Right_Ear	Left_Ear
Right_Shoulder	Right_Elbow	Right_Wrist
Left_Shoulder	Left_Elbow	Left_Wrist
Right_Hip	Right_Knee	Right_Ankle
Left_Hip	Left_Knee	Left_Ankle

본 연구에서는 위의 17 Skeleton KeyPoint를 참고하여 15 Skeleton KeyPoint를 지정하였다. 지정한 15 Skeleton KeyPoint는 Table 2와 같다.

Table 2. 15 Skeleton KeyPoint

Head	Neck	Hip
Right_Shoulder	Right_Elbow	Right_Wrist
Left_Shoulder	Left_Elbow	Left_Wrist
Right_Hip	Right_Knee	Right_Ankle
Left_Hip	Left_Knee	Left_Ankle

데이터 모델 촬영 이미지 추출, 이미지 전처리 과정을 마친 이미지를 지정한 15 Skeleton KeyPoint를 라벨링하여 JSON 파일로 export 한다.

2. COCO DataSet

COCO DataSet은 이미지와 주석을 하나의 DataSet으로 만든 형태이고 여러 일상 이미지들의 집합이며 detection, segmentation, captioning 데이터의 집합체이다. ‘cocodataset.org’에서 제공하는 COCO DataSet은 Object detection, KeyPoint detection, Stuff segmentation, Panoptic segmentation, Image captioning 등 용도에 맞는 annotation을 가지고 있다. 또한 COCO DataSet은 JSON 포맷의 파일이며 DataSet의 버전, 설명 등 여러 정보를 가지고 있는 info, 라이선스 목록을 가지고 있는 licences, 이미지 목록과 이미지 각각의 정보들을 가지고 있는 images, 객체를 기준으로 가지는 segment mask와 box, 카테고리 정보 등을 가지고 있는 annotations, DataSet 내에 존재하는 카테고리 리스트인 categories까지 크게 5가지 구분[4]을 둘 수 있다. 각각의 큰 구분들이 가지고 있는 구성 정보는 Table 3과 같다.

Table 3. 사이트에서 제공하는 COCO 구성

info	description
	url
	version
	year
	contributor
licenses	url
	id
	name
	license
	file_name
images	coco_url
	height
	width
	date_captured
	flickr_url
	id
	annotation
area	
iscrowd	
imaged_id	
bbox	
category_id	
categories	id
	name
	supercategory

‘cocodataset.org’의 COCO DataSet은 Pose뿐만 아니라 사물 등 여러 카테고리를 가지고 있지만 본 논문에서 주요하게 참고할 detection은 사람의 Pose를 추정할 수 있는 KeyPoint detection이다.

3. 환경

본 논문의 DataSet 구축 과정은 크게 이미지 전처리 과정과 이미지 라벨링 단계로 나눌 수 있다. 이미지 전처리 단계에는 Pycharm IDE를 사용하였고 os, openCV, PIL 등의 라이브러리를 사용하였다.

이미지 라벨링 단계에서는 라벨링 할 Dataset의 생성 및 정보 저장을 위한 MongoDB, MongoDB 설치를 위한 Ubuntu 20.04 LTS, Keypoint를 라벨링하여 custom COCO DataSet을 만들기 위한 GitHub 오픈소스인 coco-annotator, coco-annotator 실행을 위한 Docker를 사용하였다.

4. DataSet 리스트

본 DataSet을 구축하는 과정에서 더욱 정확한 데이터 촬영 리스트를 작성하기 위해 촬영 리스트 작성에는 ‘2021 태권도 품새 경기규칙’[5]을 참고하였다. 이를 바탕으로 피사체 모델 촬영을 진행하였다. 그림 1은 본 단계에서 작성한 촬영 리스트 중 일부이다.

주춤서기	한손날열막기	2			
	한손날뚫열막기	2			
	양손날막기	2			
	산돌막기	1			
	안팔뚝뚫헤쳐막기	2			
	뚫뚫지르기	2			
	주먹표적찌르기	2			
	팔굽열막기	2			
	팔굽표적찌르기	2			
	큰뚫찌귀	2			
앞서기	두다리제자리주춤서기	1			
	아래막기	4			
	아래굴막기	4			
	뚫뚫막기	4			
	뚫뚫안막기	4			
	안팔뚝뚫헤쳐막기	4			

Fig. 1. 촬영 리스트 중 일부

```

3 file_directory = "C:/Users/admin/Desktop/HyperParameter/"
4 directory_names = os.listdir(file_directory)
5 for directory in directory_names :
6     file_path = "C:/Users/admin/Desktop/HyperParameter/" + directory
7     print(file_path)
8     file_names = os.listdir(file_path)
9     print(file_names)
10    i = 1
11    for name in file_names:
12        src = os.path.join(file_path, name)
13        dst = str(i) + '.jpg'
14        dst = os.path.join(file_path, dst)
15        os.rename(src, dst)
16        i += 1
    
```

Fig. 4. fileRename.py

5. 모델 촬영 및 이미지 전처리

모델 구축 시 학습시킨 동작을 기준으로 정확도를 측정하여 동작을 추정하는 만큼 최대한 정확한 동작을 촬영하기 위해 현재 태권도 품새 선수로 활동하고 있는 모델의 동작을 촬영하였으며, Python의 openCV, PIL 등의 라이브러리를 이용하여 영상에서 이미지를 추출하여 이미지 전처리 작업을 진행하였다. 그림 2는 촬영한 비디오에서 이미지를 추출하는 코드 중 일부이고, 그림 3은 추출한 이미지를 정사각형으로 resize 하는 코드 중 일부이다. 그림 4는 resize한 이미지의 이름을 바꾸어 주는 코드 중 일부이고, 그림 5는 그림 3의 코드를 실행한 결과이다.



Fig. 5. Result of fig 3 ImageSquare.py

```

5 while i <= 5327 :
6     count = 1
7     success = True
8     i = str(i)
9     os.mkdir('C:/Users/admin/Desktop/DataSet/' + i)
10    vidcap = cv2.VideoCapture('C:/Users/admin/Desktop/DataSet/IMG_' + i + '.mov')
11
12    while success :
13        success, image = vidcap.read()
14        if (int(vidcap.get(1)) % 5 == 0) :
15            cv2.imwrite('C:/Users/admin/Desktop/DataSet/' + i + '/%d.jpg' % count, image)
16            print('%d.jpg' % count)
17            count += 1
18        print(i + "번 영상 추출 완료")
19        vidcap.release()
20        print()
21        i = int(i)
22        i += 1
23        success = False
24
    
```

Fig. 2. CaptureVideo.py

```

7 while flag :
8     os.chdir('C:/Users/admin/Desktop/DataSet/%d' % i) # 해당 폴더로 이동
9     files = os.listdir(os.getcwd()) # 현재 폴더에 있는 모든 파일을 list로 불러오기
10
11    for file in files:
12        img = Image.open(file) # 이미지 불러오기
13        img_size = img.size # 이미지의 크기 측정
14        # 책사각형의 이미지가 256x512 이라면, img_size = (256,512)가 된다.
15        x = img_size[0] # 넓이값
16        y = img_size[1] # 높이값
17
18        if x != y:
19            size = max(x, y)
20            resized_img = Image.new(mode='RGB', size=(size, size), color=(255, 255, 255))
21            offset = (round((abs(x - size) / 2), round((abs(y - size) / 2)))
22            resized_img.paste(img, offset)
23            resized_img.save(file + "_resized.png")
24            print(file + "_resized.png")
25
26    i = i + 1
27    if i == 5327 :
28        flag = False
    
```

Fig. 3. ImageSquare.py

6. 이미지 라벨링

이미지 라벨링은 GitHub에 공개되어있는 라벨링 툴인 coco-annotator를 이용하였다. coco-annotator 내의 카테고리 생성을 통하여 동작 8가지의 카테고리를 생성하였고 전처리 작업을 거친 이미지에 KeyPoint를 라벨링을 진행하였다. 그림 6은 한 이미지에 대한 라벨링 결과이다.

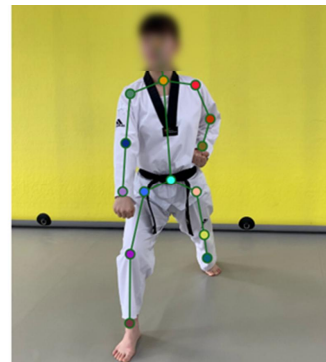


Fig. 6. Image Labeling

7. COCO DataSet 구축

본 논문에서 구축한 JSON 형식 파일의 구성은 표 4와 같다. 다만, 본 논문에서 구축한 JSON 파일과 COCO 홈페이지에서 제공하는 JSON 파일의 구성과는 차이가 있다.

Table 4. custom COCO 구성

images	id
	dataset_id
	category_id
	path
	width
	height
	file_name
	annotated
	annotating
	num_annotation
	metadata
	deleted
	milliseconds
categories	id
	name
	supercategory
	color
	metadata
	keypoint_colors
	keypoints
annotation	id
	image_id
	category_id
	segmentation
	area
	bbox
	iscrowed
	color
	keypoints
	metadata
	num_keypoints

후 학습시키고 학습을 시키지 않은 동작 들을 더 추가할 것이다. 동작을 추가하여 모델이 어느 정도 완성이 되면 실제 교육현장에서 사용이 가능한 모델이 될 것이고 완성된 모델을 사용하여 교육용 프로그램이나 교육용 게임을 만들 수 있을 것으로 기대된다.

REFERENCES

- [1] <http://www.worldtaekwondo.org/>
- [2] <https://paperswithcode.com/task/pose-estimation>
- [3] <https://aihub.or.kr/>
- [4] <https://cocodataset.org/#home>
- [5] <http://www.koreataekwondo.org/>

그림 7은 본 논문에서 구축한 DataSet의 JSON 파일 중 일부이다.

```

"images": [
  {
    "id": 199,
    "dataset_id": 2,
    "category_ids": [
    ],
    "path": "/datasets/TK/1.jpg",
    "width": 1920,
    "height": 1920,
    "file_name": "1.jpg",
    "annotated": false,
    "annotating": [
    ],
    "num_annotations": 0,
    "metadata": [
    ],
    "deleted": false,
    "milliseconds": 0,
    "events": [
    ],
    "regenerate_thumbnail": false
  },
  {
    "id": 7,
    "image_id": 199,
    "category_id": 1,
    "segmentation": [
    ],
    "area": 0,
    "bbox": [
      0,
      0,
      0,
      0
    ],
    "iscrowed": false,
    "isbbox": false,
    "color": "#e07490",
    "keypoints": [
      909,
      396,
    ]
  }
]
    
```

Fig. 7. JSON of DataSet

IV. 결론

현재까지는 8개 동작의 영상에서 추출한 이미지의 라벨링 정보로 Custom DataSet을 구축하는 과정까지 진행하였다. 향후 연구에서는 본 연구에서 구축한 JSON 파일을 학습이 가능한 형태로 가공한