

---

# Introduction to Human Pose Estimation

---

2021. 02. 26

**Data Mining and Quality Analytics Lab**

조용원

# 발표자 소개

## ❖ 조용원

- 현재 석·박사 통합과정 5학기 재학 중 (지도교수: 김성범)
- E-mail: [gyj4318@korea.ac.kr](mailto:gyj4318@korea.ac.kr)
- 연구분야
  - ✓ Supervised semantic segmentation
  - ✓ Anomaly Detection and Localization




**종료** Mask R-CNN

---

2020년 01월 10일

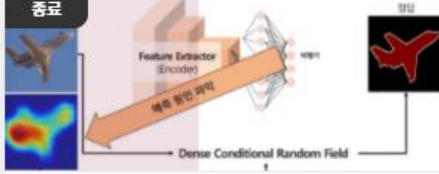
---

Mask R-CNN

발표자:  조용원


📅 2020년 1월 10일  
🕒 오후 1시 ~  
📍 고려대학교 신공학관 218호

세미나 정보 보기 →

**종료**  **발표**

---

Introduction to Weakly Supervised Sema

발표자:  조용원

📅 2020년 8월 21일  
🕒 오후 1시 ~  
📍 온라인  
▶ 온라인 비디오 시청 (YouTube)

세미나 정보 보기 →

# 목차

---

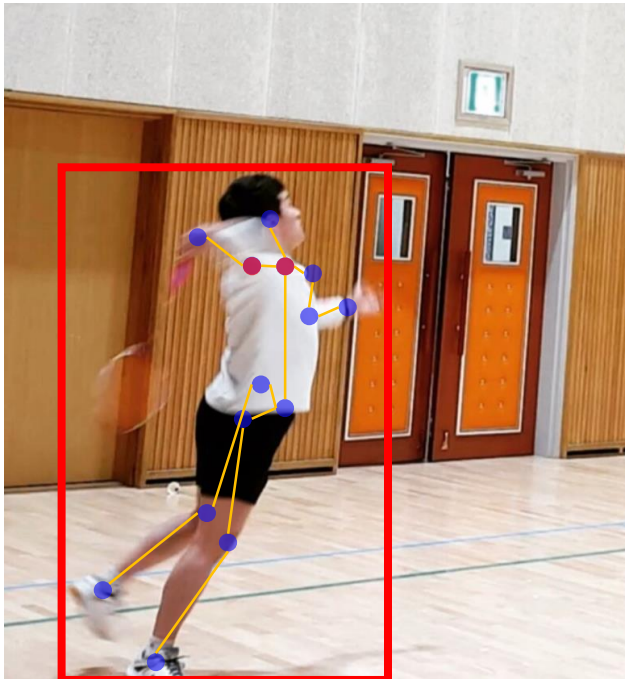
- **Introduction**
- **Single Person Pose Estimation**
- **Multi-Person Pose Estimation**
- **Conclusion**

# Introduction

## - Introduction

### ❖ 만약 인공지능(AI) 배드민턴 코치님을 개발한다면?

- 인간의 눈을 대체할 수 있는 **시각 기능**(사람이 어디에 있는지에 대한 위치 정보)
- 시각 기능을 통해 획득한 **자세에 대한 정보**(관절 중심점에 대한 위치정보)
- 올바른 배드민턴 자세에 대한 지식



이미지 입력

AI 코치님 조언



팔꿈치와 어깨 사이 각도가 없습니다.  
팔꿈치를 더 올린다면 더 강한 스매싱을 할 수 있습니다!

# Introduction

## - Introduction

### ❖ 만약 인공지능(AI) 배드민턴 코치님을 개발한다면?

- 인간의 눈을 대체할 수 있는 **시각 기능**(사람이 어디에 있는지에 대한 위치 정보)
- 시각 기능을 통해 획득한 **자세에 대한 정보**(관절 중심점에 대한 위치정보)
- 올바른 배드민턴 자세에 대한 지식



## Human Pose Estimation

이미지 입력

AI 코치님 조언



팔꿈치와 어깨 사이 각도가 없습니다.  
팔꿈치를 더 올린다면 더 강한 스매싱을 할 수 있습니다!

# Introduction

- Applications of human pose estimation

## ❖ Human pose estimation (HPE) 응용 분야

- 인간의 자세를 인식하는 각종 솔루션에 사용
  - ① 헬스와 같이 자세가 중요한 운동 관련 솔루션 (Motion analysis)
  - ② 물리치료사들에게 추가적인 정보를 제공하는 솔루션 (Medical assistances)
- Animation 생성

**Motion analysis**



**Medical assistances**



- <https://mobidev.biz/blog/human-pose-estimation-ai-personal-fitness-coach>  
- <https://cv-tricks.com/pose-estimation/using-deep-learning-in-opencv/>

# Introduction

- Applications of human pose estimation

## ❖ Human pose estimation (HPE) 응용 분야

- 인간의 자세를 인식하는 각종 솔루션에 사용
  - ① 헬스와 같이 자세가 중요한 운동 관련 솔루션 (Motion analysis)
  - ② 물리치료사들에게 추가적인 정보를 제공하는 솔루션 (Medical assistances)
- Animation 생성

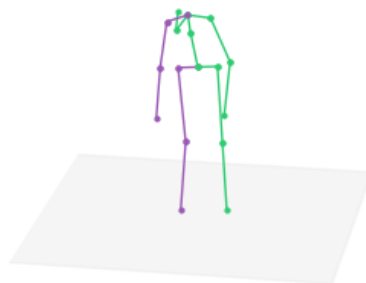
인간 움직임  
(입력 데이터)



관절 시각화



Animation 생성  
(출력 데이터)



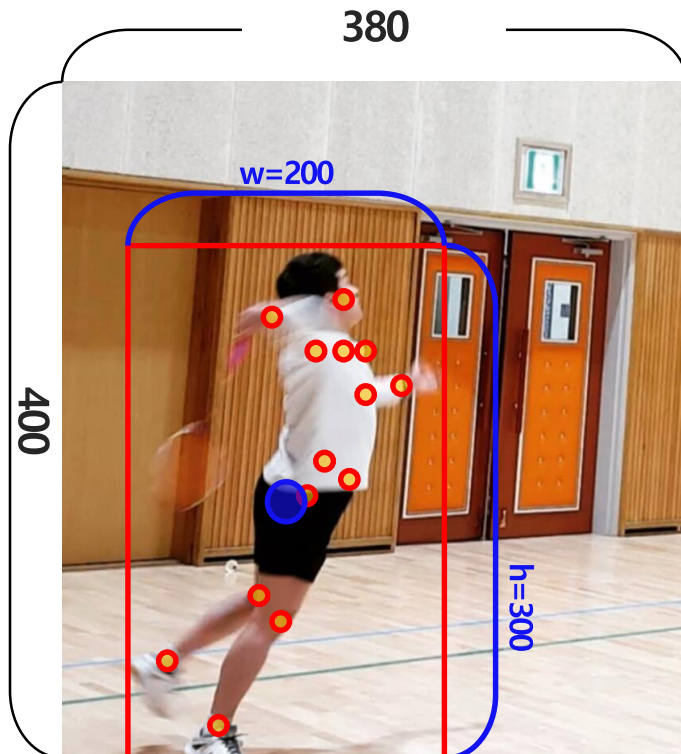
- <https://github.com/peterjq/OpenMMD>

# Introduction

- Input data and target (label)

## ❖ Human pose estimation (HPE) 입력 및 출력 데이터 예시

- 입력 데이터는 일반적으로 RGB 이미지
- 출력 데이터는 사람의 Bounding Box와 탐지하고자 하는 관절에 해당하는  $(x, y)$  좌표



- 사람에 대한 Bounding Box:  $(x, y, h, w)$ 
  - Box의 중심:  $(x, y) = (120, 15)$
  - 높이와 너비:  $h, w$
- 탐지하고자 하는 관절 개수 = 16개
  - ① 머리:  $(x, y) = (120, 370)$
  - ② 왼쪽 어깨:  $(x, y) = (120, 330)$
  - ③ 왼쪽 팔꿈치:  $(x, y) = (100, 330)$
  - ④ 왼쪽 손목:  $(x, y) = (80, 350)$
  - 
  - 
  - 
  - ⑬ 오른쪽 발목:  $(x, y) = (50, 70)$

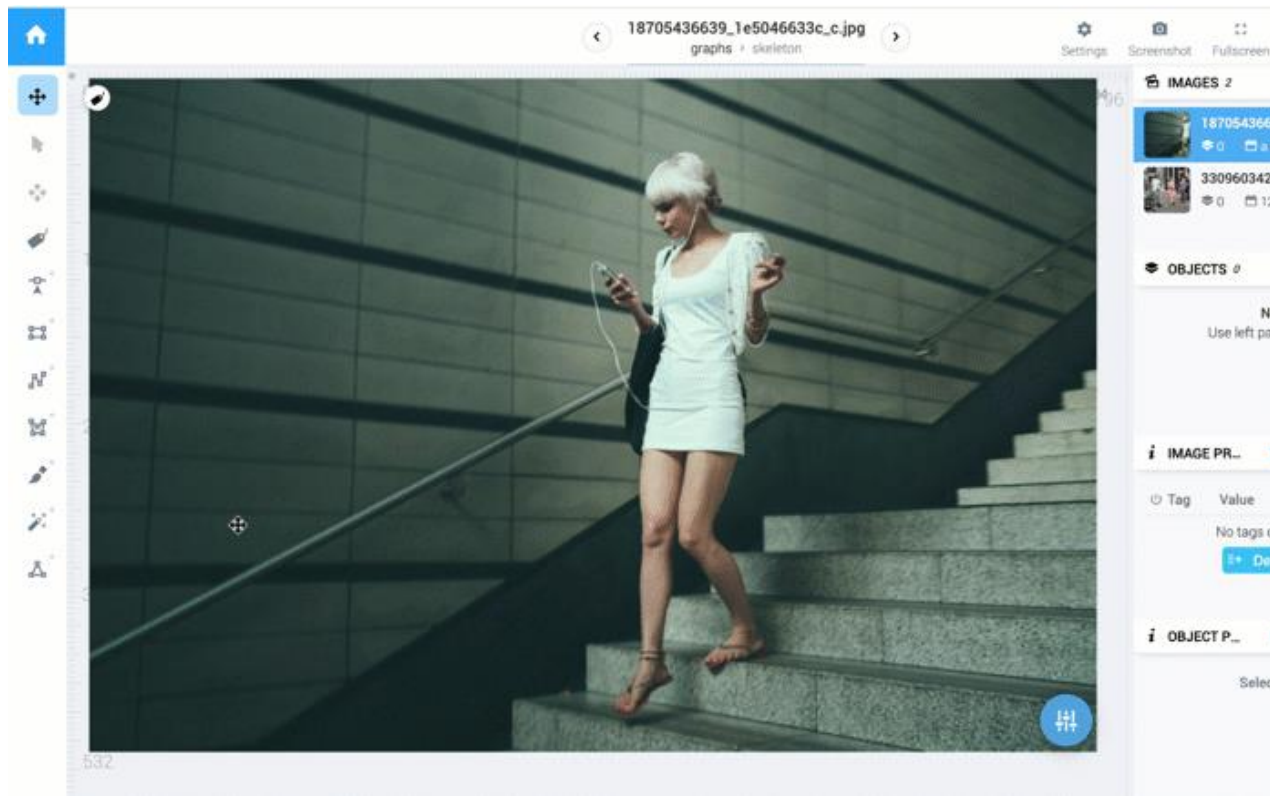


# Introduction

- Dataset generation and preparations

## ❖ Human pose estimation (HPE) 데이터 셋 레이블링 도구 소개

- 도구 명: Supervisely (<https://supervise.ly/>)
- 학생들에게는 무료로 제공되고 있으며 기업의 경우 별도의 계약 필요



# Introduction

- Hierarchy of HPE

## ❖ Human pose estimation (HPE) 모델 계층

- 입력 이미지 내 사람 **한 명만 존재**하는 경우: Single person pose estimation
  - Direct regression: 관절 별 좌표를 예측
  - Heatmap based estimation: 특정 관절이 존재할 만 한 곳을 Heatmap 형태로 출력
- 입력 이미지 내 사람이 두 명 이상 존재하는 경우: Multi-person pose estimation
  - Top-down approach: 사람을 우선적으로 탐지 후 탐지 결과 내에서 관절별 좌표를 예측
  - Bottom-up approach: 탐지하고자 하는 관절에 대한 위치 예측 후 사람 별로 나누는 과정 진행

Direct regression



Heatmap based estimation



# Introduction

## - Hierarchy of HPE

### ❖ Human pose estimation (HPE) 알고리즘 계층도

- 입력 이미지 내 사람 한 명만 존재하는 경우: Single person pose estimation
  - Direct regression: 관절 별 좌표를 예측
  - Heatmap based estimation: 특정 관절이 존재할 만 한 곳을 Heatmap 형태로 출력
- 입력 이미지 내 사람이 **두 명 이상 존재**하는 경우: Multi-person pose estimation
  - Top-down approach: 사람을 우선적으로 탐지 후 탐지 결과 내에서 관절별 좌표를 예측
  - Bottom-up approach: 탐지하고자 하는 관절에 대한 위치 예측 후 사람 별로 나누는 과정 진행



# Introduction

## - Hierarchy of HPE

### ❖ Human pose estimation (HPE) 알고리즘 계층도

- 입력 이미지 내 사람 한 명만 존재하는 경우: Single person pose estimation
  - Direct regression: 관절 별 좌표를 예측
  - Heatmap based estimation: 특정 관절이 존재할 만 한 곳을 Heatmap 형태로 출력
- 입력 이미지 내 사람이 **두 명 이상 존재**하는 경우: Multi-person pose estimation
  - Top-down approach: 사람을 우선적으로 탐지 후 탐지 결과 내에서 관절별 좌표를 예측
  - Bottom-up approach: 탐지하고자 하는 관절에 대한 위치 예측 후 사람 별로 나누는 과정 진행



# Introduction

---

- Hierarchy of HPE

## ❖ Human pose estimation (HPE) 알고리즘 계층도

- 입력 이미지 내 사람 **한 명만 존재**하는 경우: Single person pose estimation
  - **Direct regression: 관절 별 좌표를 예측**
  - Heatmap based estimation: 특정 관절이 존재할 만 한 곳을 Heatmap 형태로 출력
- 입력 이미지 내 사람이 **두 명 이상 존재**하는 경우: Multi-person pose estimation
  - **Top-down approach: 사람을 우선적으로 탐지 후 탐지 결과 내에서 관절별 좌표를 예측**
  - Bottom-up approach: 탐지하고자 하는 관절에 대한 위치 예측 후 사람 별로 나누는 과정 진행

# Single Person Pose Estimation

- Direct regression method

## ❖ DeepPose: Human Pose Estimation via Deep Neural Networks

- 2014년 IEEE conference on Computer Vision and Pattern Recognition에서 발표
- 저자들은 Google 소속이며 2021년 2월 22일 기준 1920회 인용
- Deep Learning을 HPE 분야에 최초로 적용한 논문

### DeepPose: Human Pose Estimation via Deep Neural Networks

Alexander Toshev  
toshev@google.com  
Google

Christian Szegedy  
szegedy@google.com  
Google



Figure 1. Besides extreme variability in articulations, many of the joints are barely visible. We can guess the location of the right arm in the left image only because we see the rest of the pose and anticipate the motion or activity of the person. Similarly, the left body half of the person on the right is not visible at all. These are examples of the need for *holistic reasoning*. We believe that DNNs can naturally provide such type of reasoning.

and in the recent years a variety of models with efficient inference have been proposed ([6, 18]).

The above efficiency, however, is achieved at the cost of limited expressiveness – the use of local detectors, which reason in many cases about a single part, and most importantly by modeling only a small subset of all interactions between body parts. These limitations, as exemplified in Fig. 1, have been recognized and methods reasoning about pose in a holistic manner have been proposed [15, 20] but with limited success in real-world problems.

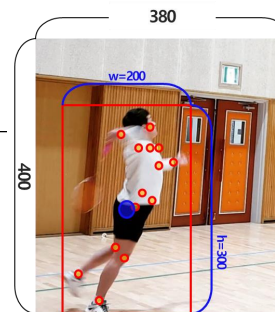
In this work we ascribe to this holistic view of human pose estimation. We capitalize on recent developments of deep learning and propose a novel algorithm based on a

# Single Person Pose Estimation

- Direct regression method (stage 1)

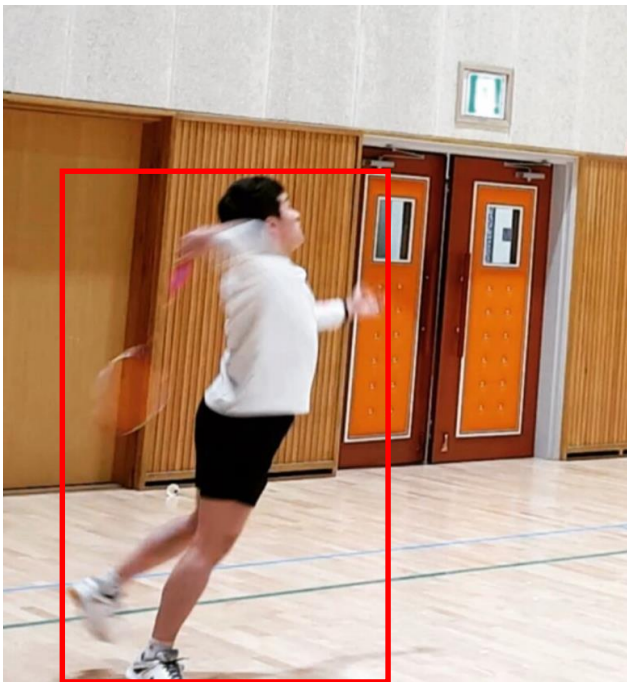
## ❖ DeepPose 모델 – 데이터 입력에서 예측까지 과정

- 입력 데이터: RGB 이미지
- Bounding Box를 사용해 사람이 존재하는 영역만 추출
  - Bounding Box  $(x, y, w, h) = (120, 150, 200, 300)$



- 사람에 대한 Bounding Box:  $(x, y, h, w)$ 
  - > Box의 중심:  $(x, y) = (120, 15)$
  - > 높이와 너비:  $h, w$
- 탐지하고자 하는 관절 개수 = 16개
  - ① 머리:  $(x, y) = (120, 370)$
  - ② 왼쪽 어깨:  $(x, y) = (120, 330)$
  - ③ 왼쪽 팔꿈치:  $(x, y) = (100, 330)$
  - ④ 왼쪽 손목:  $(x, y) = (80, 350)$
  - ⋮
  - ⑯ 오른쪽 발목:  $(x, y) = (50, 70)$

## 입력 RGB 이미지

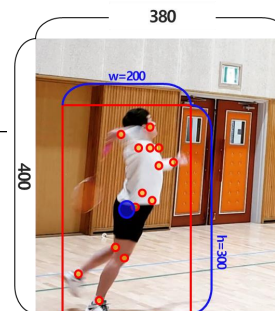


# Single Person Pose Estimation

- Direct regression method (stage 1)

## ❖ DeepPose 모델 - 데이터 입력에서 예측까지 과정

- Bounding Box를 사용해 사람이 존재하는 영역만 추출
  - Bounding Box  $(x, y, w, h) = (120, 150, 200, 300)$
- 관절별 좌표는 추출 전(前) 이미지 내 좌표이기 때문에 변환 필요

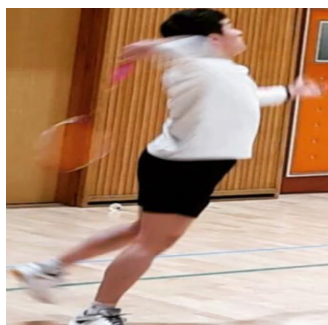


- 사람에 대한 Bounding Box:  $(x, y, h, w)$ 
  - Box의 중심:  $(x, y) = (120, 150)$
  - 높이와 너비:  $h, w$
- 탐지하고자 하는 관절 개수 = 16개
  - ① 머리:  $(x, y) = (120, 370)$
  - ② 왼쪽 어깨:  $(x, y) = (120, 330)$
  - ③ 왼쪽 팔꿈치:  $(x, y) = (100, 330)$
  - ④ 왼쪽 손목:  $(x, y) = (80, 350)$
  - ⋮
  - ⑯ 오른쪽 발목:  $(x, y) = (50, 70)$

추출된 사람  
이미지



고정된 크기로  
크기 변환



Size=(220, 220, 3)

관절별 좌표

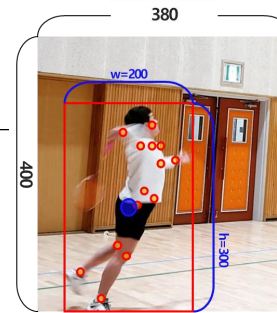
변환된 관절별 좌표

	관절별 좌표		변환된 관절별 좌표	
	X 좌표	Y 좌표	변환된 X 좌표	변환된 Y 좌표
머리	120	370		
왼쪽 어깨	120	330		
왼쪽 팔꿈치	100	330		
왼쪽 손목	80	350		
.....	.....	.....		
오른쪽 발목	50	70		



# Single Person Pose Estimation

- Direct regression method (stage 1)



- 사람에 대한 Bounding Box: (x, y, h, w)
  - > Box의 중심: (x, y)=(120, 15)
  - > 높이와 너비: h, w
- 탐지하고자 하는 관절 개수 = 16개
  - ① 머리: (x, y) = (120, 370)
  - ② 왼쪽 어깨: (x, y) = (120, 330)
  - ③ 왼쪽 팔꿈치: (x, y) = (100, 330)
  - ④ 왼쪽 손목: (x, y) = (80, 350)
  - ⋮
  - ⑯ 오른쪽 발목: (x, y) = (50, 70)

## ❖ DeepPose 모델 – 데이터 입력에서 예측까지 과정

- Bounding Box를 사용해 사람이 존재하는 영역만 추출
  - Bounding Box (x, y, w, h) = (120, 150, 200, 300)
- 관절별 좌표는 추출 전(前) 이미지 내 좌표이기 때문에 변환 필요

### • 머리에 대한 좌표 변환

- 변환된 X 좌표 =  $\frac{1}{200} (120 - 120)$
- 변환된 Y 좌표 =  $\frac{1}{300} (370 - 300)$

### • 왼쪽 팔꿈치에 대한 좌표 변환

- 변환된 X 좌표 =  $\frac{1}{200} (100 - 120)$
- 변환된 Y 좌표 =  $\frac{1}{300} (330 - 300)$

관절별 좌표

변환된 관절별 좌표

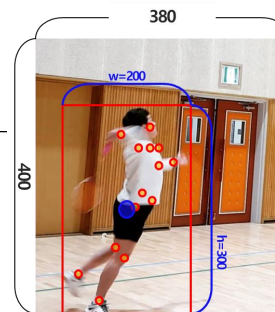
	관절별 좌표		변환된 관절별 좌표	
	X 좌표	Y 좌표	변환된 X 좌표	변환된 Y 좌표
머리	120	370	0	0.73
왼쪽 어깨	120	330		
왼쪽 팔꿈치	100	330	- 0.1	0.60
왼쪽 손목	80	350		
.....	.....	.....		
오른쪽 발목	50	70		

# Single Person Pose Estimation

- Direct regression method (stage 1)

## ❖ DeepPose 모델 – 데이터 입력에서 예측까지 과정

- Bounding Box를 사용해 사람이 존재하는 영역만 추출
  - Bounding Box (x, y, w, h) = (120, 150, 200, 300)
- 관절별 좌표는 추출 전(前) 이미지 내 좌표이기 때문에 변환 필요



- 사람에 대한 Bounding Box: (x, y, h, w)
  - Box의 중심: (x, y)=(120, 15)
  - 높이와 너비: h, w
- 탐지하고자 하는 관절 개수 = 16개
  - ① 머리: (x, y) = (120, 370)
  - ② 왼쪽 어깨: (x, y) = (120, 330)
  - ③ 왼쪽 팔꿈치: (x, y) = (100, 330)
  - ④ 왼쪽 손목: (x, y) = (80, 350)
  - ⋮
  - ⑯ 오른쪽 발목: (x, y) = (50, 70)

$$N(y_i, b) = \begin{pmatrix} \frac{1}{w} & 0 \\ 0 & \frac{1}{h} \end{pmatrix} (y_i - b_c)$$

- $i$ : 관절 인덱스 ( $i = 1, 2, \dots, K$ )
- $w$ : Bounding box 너비
- $h$ : Bounding box 높이
- $y_i$ :  $i$ 번째 관절에 대한 좌표
- $b$ : Bounding box
- $b_c$ : Bounding box 중심 좌표

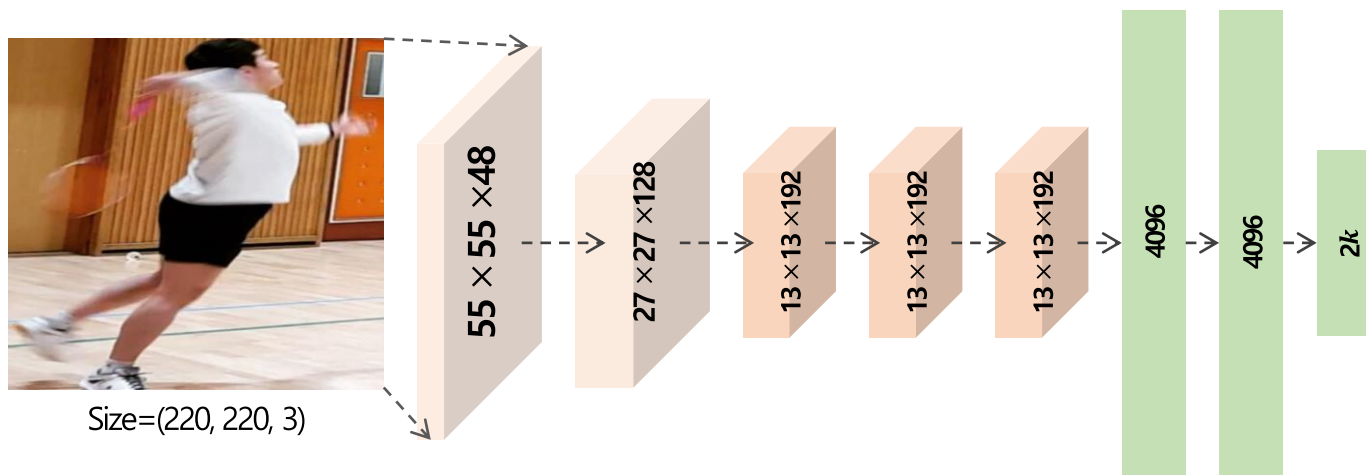
	관절별 좌표		변환된 관절별 좌표	
	X 좌표	Y 좌표	변환된 X 좌표	변환된 Y 좌표
머리	120	370	0	0.73
왼쪽 어깨	120	330	0	0.60
왼쪽 팔꿈치	100	330	- 0.1	0.60
왼쪽 손목	80	350	- 0.2	0.67
.....	.....	.....	.....	.....
오른쪽 발목	50	70	- 0.35	- 0.27

# Single Person Pose Estimation

- Direct regression method (stage 1)

## ❖ DeepPose 모델 – 데이터 입력에서 예측까지 과정

- 처리된 입력 데이터를 이용해 예측값을 산출하는 모델
- AlexNet 기반 특징 추출기를 사용해 Representation 벡터 산출
- 해당 벡터를 Fully connected layer에 입력하여 관절별 예측 값 산출
  - $2k$ :  $k$ 개 관절별  $(x, y)$



# Single Person Pose Estimation

- Direct regression method (stage 1)

$$N(y_i, b) = \begin{pmatrix} \frac{1}{w} & 0 \\ 0 & \frac{1}{h} \end{pmatrix} (y_i - b_c)$$

- $i$ : 관절 인덱스 ( $i = 1, 2, \dots, K$ )
- $w$ : Bounding box 너비
- $h$ : Bounding box 높이
- $y_i$ :  $i$ 번째 관절에 대한 좌표
- $b_c$ : Bounding box 중심 좌표

	관절별 좌표		변환된 관절별 좌표	
	X 좌표	Y 좌표	변환된 X 좌표	변환된 Y 좌표
머리	120	370	0	0.73
왼쪽 어깨	120	330	0	0.60
왼쪽 팔꿈치	100	330	-0.1	0.60
왼쪽 손목	80	350	-0.2	0.67
.....	.....	.....	.....	.....
오른쪽 발목	50	70	-0.35	-0.27

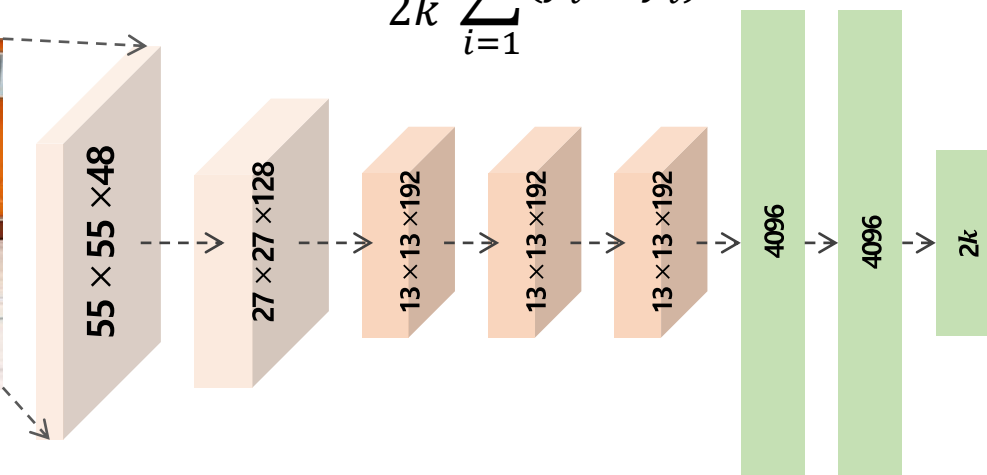
## ❖ DeepPose 모델 – 데이터 입력에서 예측까지 과정

- 해당 벡터를 Fully connected layer에 입력하여 관절별 예측 값 산출
  - $2k$ :  $k$ 개 관절별 ( $x, y$ )
- Mean squared error(MSE) 손실 함수를 사용
- 관절별 대략적인 위치 파악

$$MSE = \frac{1}{2k} \sum_{i=1}^{2k} (y_i - \hat{y}_i)^2$$



Size=(220, 220, 3)



	예측 ( $\hat{y}_i$ )	실제 ( $y_i$ )
1	0.02	0
2	0.01	0.73
3	-0.03	0
4	0	0.60
...	.....	.....
...	.....	.....
$2k$	-0.3	-0.27

# Single Person Pose Estimation

- Direct regression method (stage  $s, s \geq 2$ )

$$N(y_i, b) = \begin{pmatrix} \frac{1}{w} & 0 \\ 0 & \frac{1}{h} \end{pmatrix} (y_i - b_c)$$

- $i$ : 관절 인덱스 ( $i = 1, 2, \dots, K$ )
- $w$ : Bounding box 너비
- $h$ : Bounding box 높이
- $y_i$ :  $i$ 번째 관절에 대한 좌표
- $b_c$ : Bounding box 중심 좌표

	관절별 좌표		변환된 관절별 좌표	
	X 좌표	Y 좌표	변환된 X 좌표	변환된 Y 좌표
머리	120	370	0	0.73
왼쪽 어깨	120	330	0	0.60
왼쪽 팔꿈치	100	330	-0.1	0.60
왼쪽 손목	80	350	-0.2	0.67
.....	.....	.....	.....	.....
오른쪽 발목	50	70	-0.35	-0.27

## ❖ 관절별 대략적 위치 → 관절별 위치 예측모델 학습

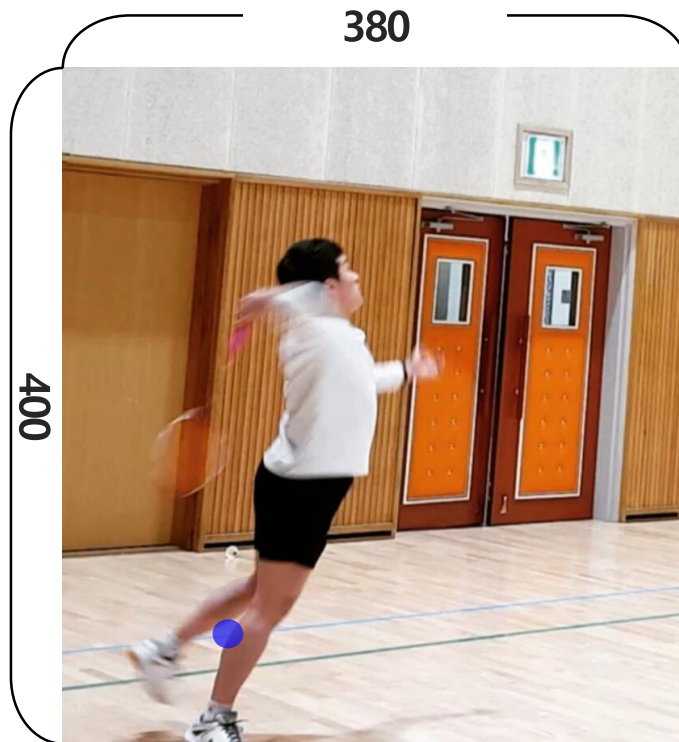
- 앞에서 산출된 예측 관절 위치를 실제 이미지 내 위치로 역 변환
  - 오른쪽 엉덩이 위치 예측 값 (-0.3, -0.7) → 실제 이미지 내 위치로 변환

### • 오른쪽 엉덩이에 대한 좌표 역 변환

- $-0.3 = \frac{1}{200} (x - 120) \rightarrow x = 60$
- $-0.7 = \frac{1}{300} (y - 300) \rightarrow y = 90$



Size=(220, 220, 3)



실제 이미지

# Single Person Pose Estimation

- Direct regression method (stage  $s, s \geq 2$ )

$$N(y_i, b) = \begin{pmatrix} \frac{1}{w} & 0 \\ 0 & \frac{1}{h} \end{pmatrix} (y_i - b_c)$$

- $i$ : 관절 인덱스 ( $i = 1, 2, \dots, K$ )
- $w$ : Bounding box 너비
- $h$ : Bounding box 높이
- $y_i$ :  $i$ 번째 관절에 대한 좌표
- $b_c$ : Bounding box 중심 좌표

	관절별 좌표		변환된 관절별 좌표	
	X 좌표	Y 좌표	변환된 X 좌표	변환된 Y 좌표
머리	120	370	0	0.73
왼쪽 어깨	120	330	0	0.60
왼쪽 팔꿈치	100	330	-0.1	0.60
왼쪽 손목	80	350	-0.2	0.67
.....	.....	.....	.....	.....
오른쪽 발목	50	70	-0.35	-0.27

## ❖ 관절별 대략적 위치 → 관절별 위치 예측모델 학습

- 앞에서 산출된 예측 관절 위치를 실제 이미지 내 위치로 역 변환
  - 오른쪽 엉덩이 위치 예측 값 (-0.3, -0.7) → 실제 이미지 내 위치로 변환
  - 왼쪽 어깨 위치 예측 값 (0.2, 0.2) → 실제 이미지 내 위치로 변환

### • 오른쪽 엉덩이에 대한 좌표 역 변환

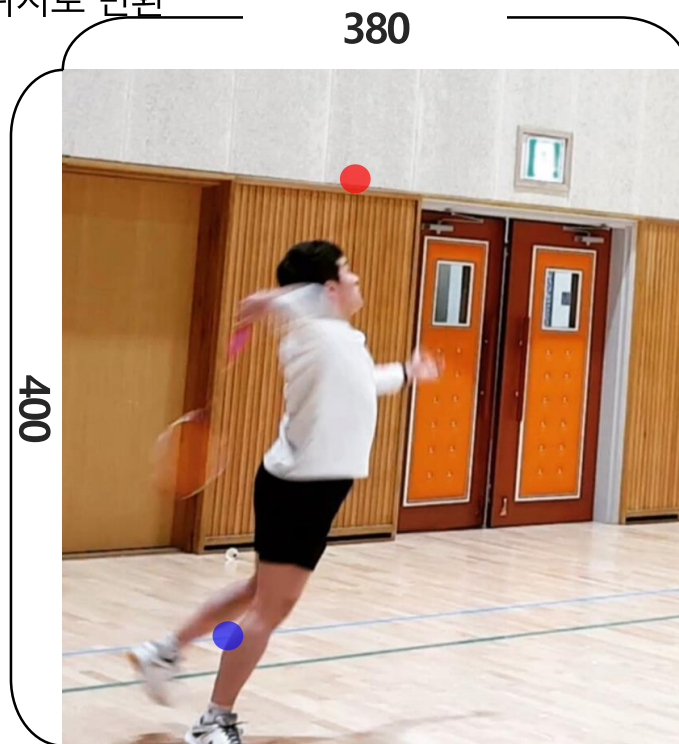
- $-0.3 = \frac{1}{200}(x - 120) \rightarrow x = 60$
- $-0.7 = \frac{1}{300}(y - 300) \rightarrow y = 90$

### • 왼쪽 어깨에 대한 좌표 역 변환

- $0.2 = \frac{1}{200}(x - 120) \rightarrow x = 160$
- $0.2 = \frac{1}{300}(y - 300) \rightarrow y = 360$



Size=(220, 220, 3)



실제 이미지

# Single Person Pose Estimation

- Direct regression method (stage  $s, s \geq 2$ )

$$N(y_i, b) = \begin{pmatrix} \frac{1}{w} & 0 \\ 0 & \frac{1}{h} \end{pmatrix} (y_i - b_c)$$

- $i$ : 관절 인덱스 ( $i = 1, 2, \dots, K$ )
- $w$ : Bounding box 너비
- $h$ : Bounding box 높이
- $y_i$ :  $i$ 번째 관절에 대한 좌표
- $b_c$ : Bounding box 중심 좌표

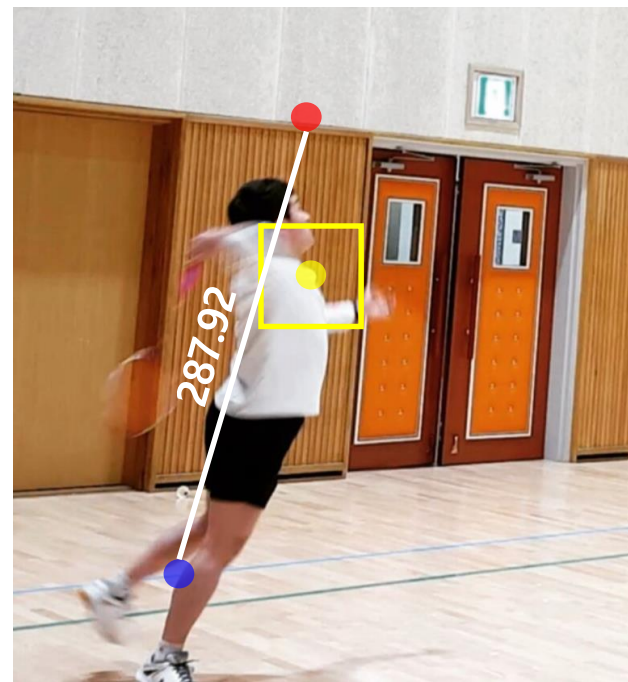
	관절별 좌표		변환된 관절별 좌표	
	X 좌표	Y 좌표	변환된 X 좌표	변환된 Y 좌표
머리	120	370	0	0.73
왼쪽 어깨	120	330	0	0.60
왼쪽 팔꿈치	100	330	-0.1	0.60
왼쪽 손목	80	350	-0.2	0.67
.....	.....	.....	.....	.....
오른쪽 발목	50	70	-0.35	-0.27

## ❖ 관절별 대략적 위치 → 관절별 위치 예측모델 학습

- 실제 이미지 내에서 왼쪽 어깨 예측 값과 오른쪽 엉덩이 예측 값 사이 거리 계산
  - 거리 =  $\sqrt{(160 - 60)^2 + (360 - 90)^2} = 287.92$
- 왼쪽 어깨 관절 실제 위치를 중심으로 하는 Bounding box 생성
- Bounding box의 너비와 높이 =  $\delta * 287.92$



Size=(220, 220, 3)



실제 이미지

# Single Person Pose Estimation

- Direct regression method (stage  $s, s \geq 2$ )

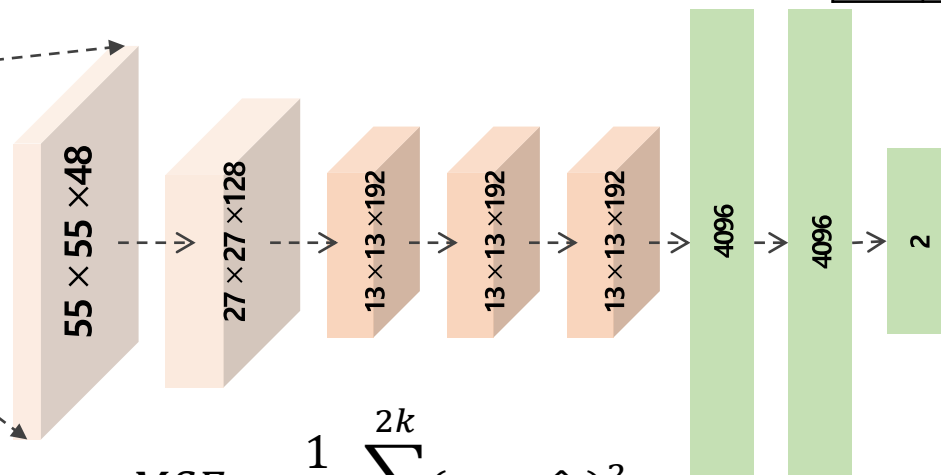
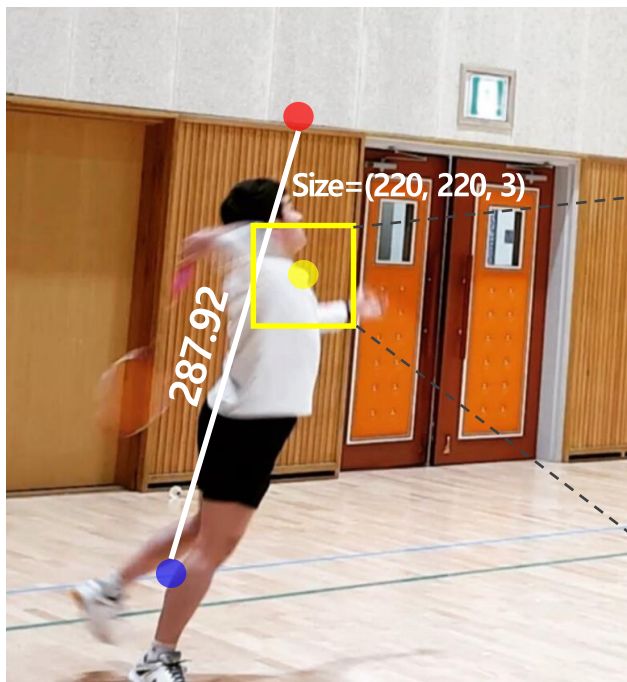
## ❖ 관절별 대략적 위치 → 관절별 위치 예측모델 학습

- Bounding box의 너비와 높이 =  $\delta * 287.92$
- Stage 1에서 관절 위치를 변환하는 동일한 과정을 진행
- 앞에서 학습한 모형에 새로운 Bounding box를 입력해 예측 값 산출 및 학습 진행

$$N(y_i, b) = \begin{pmatrix} \frac{1}{w} & 0 \\ 0 & \frac{1}{h} \end{pmatrix} (y_i - b_c)$$

- $i$ : 관절 인덱스 ( $i = 1, 2, \dots, K$ )
- $w$ : Bounding box 너비
- $h$ : Bounding box 높이
- $y_i$ :  $i$ 번째 관절에 대한 좌표
- $b_c$ : Bounding box 중심 좌표

	관절별 좌표		변환된 관절별 좌표	
	X 좌표	Y 좌표	변환된 X 좌표	변환된 Y 좌표
머리	120	370	0	0.73
왼쪽 어깨	120	330	0	0.60
왼쪽 팔꿈치	100	330	-0.1	0.60
왼쪽 손목	80	350	-0.2	0.67
.....	.....	.....	.....	.....
오른쪽 발목	50	70	-0.35	-0.27



	예측 ( $\hat{y}_i$ )	실제 ( $y_i$ )
1	0.1	0.1
2	0.25	0.23

$$MSE = \frac{1}{2k} \sum_{i=1}^{2k} (y_i - \hat{y}_i)^2$$



# Single Person Pose Estimation

- Direct regression method (전체 학습)

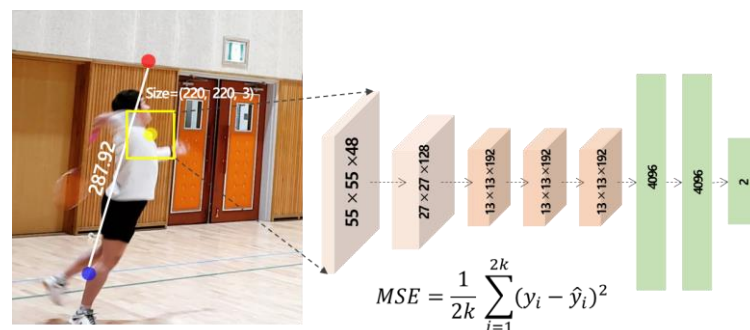
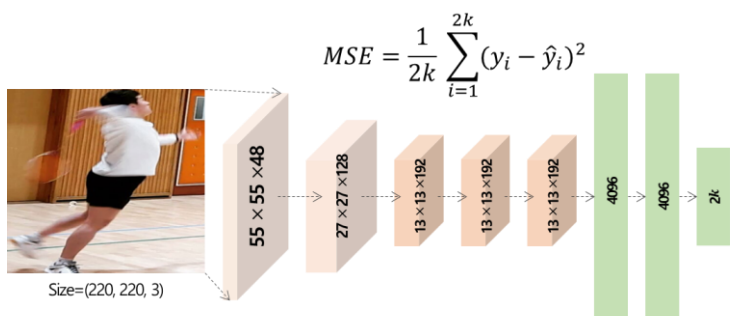
## ❖ DeepPose 학습 과정

### 1. Stage 1 모델을 학습

- 입력 이미지 내 관절의 대략적인 위치를 파악하기 위함

### 2. 관절별로 Stage 2 모델을 학습

- 관절별로 개별적인 모델 사용 (탐지하고자 하는 관절 수:  $k$ 개 → 모델 개수:  $k$ 개)
- 관절별 모델을 학습 해야하기에 많은 학습 시간, 추론 시간을 가지는 문제점이 존재함
- 여러 관절 사이의 관계를 고려한다고 할 수 없음

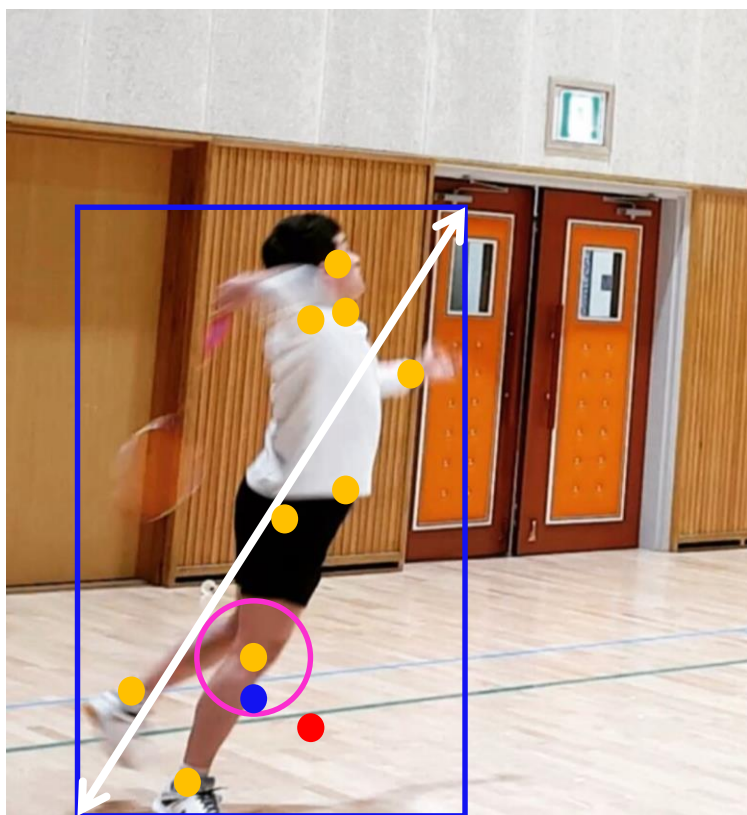


# Single Person Pose Estimation

- Evaluation metric for direct regression method

## ❖ Human Pose Estimation 평가 지표 및 계산 방식

- Percent of Detected Joints (PDJ) 지표



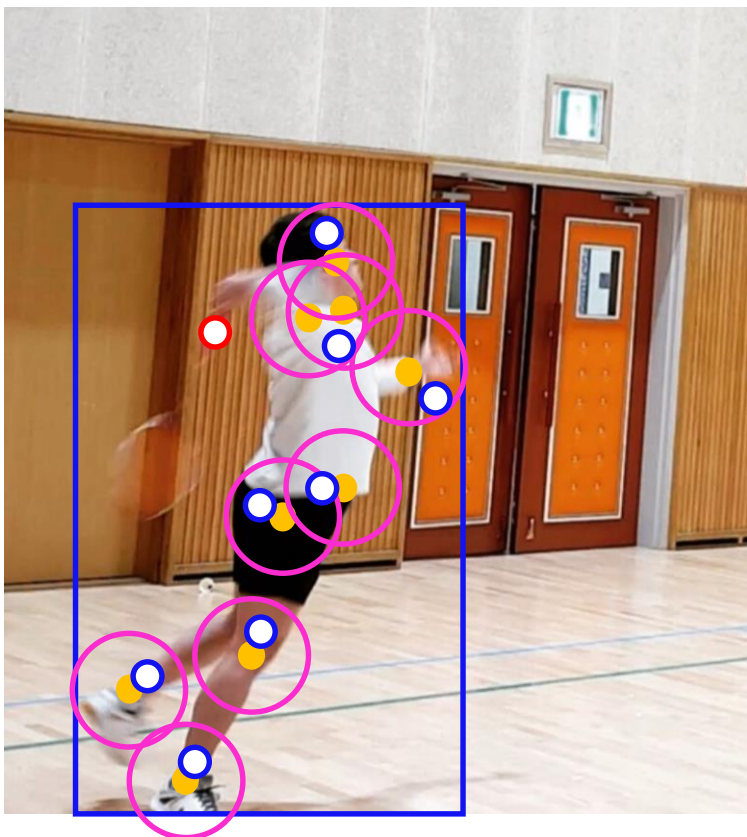
- ① 몸통 길이(흰 선)을 계산
- ② (특정 임계값 \* 길이)가 반지름인 원 생성
  - 반지름=임계값\*길이 =  $0.05 * 500 = 25$
- ③ 예측 위치가 원 내부에 있는지 여부 확인
  - 원 **내부에** 있는 경우 (옳은 결과): 1
  - 원 **외부에** 있는 경우 (틀린 결과): 0

# Single Person Pose Estimation

- Evaluation metric for direct regression method

## ❖ Human Pose Estimation 평가 지표 및 계산 방식

- Percent of Detected Joints (PDJ) 지표



- ① 몸통 길이(흰 선)을 계산
- ② (특정 임계값 \* 길이)가 반지름인 원 생성
  - 반지름=임계값\*길이 =  $0.05 * 500 = 25$
- ③ 예측 위치가 원 내부에 있는지 여부 확인
  - 원 **내부에** 있는 경우 (옳은 결과): 1
  - 원 **외부에** 있는 경우 (틀린 결과): 0
- ④ 예시 예측 결과에 대한 PDJ 계산
  - **○** 의 개수: 8개
  - **○** 의 개수: 1개
  - $PDJ = (\text{파란 원 개수}) / (\text{전체 관절 수}) = 8/9$

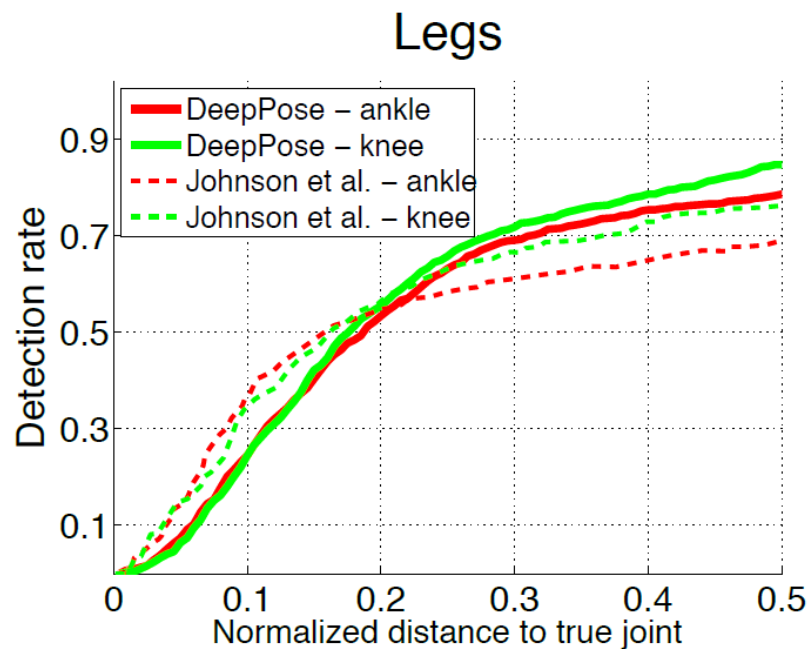
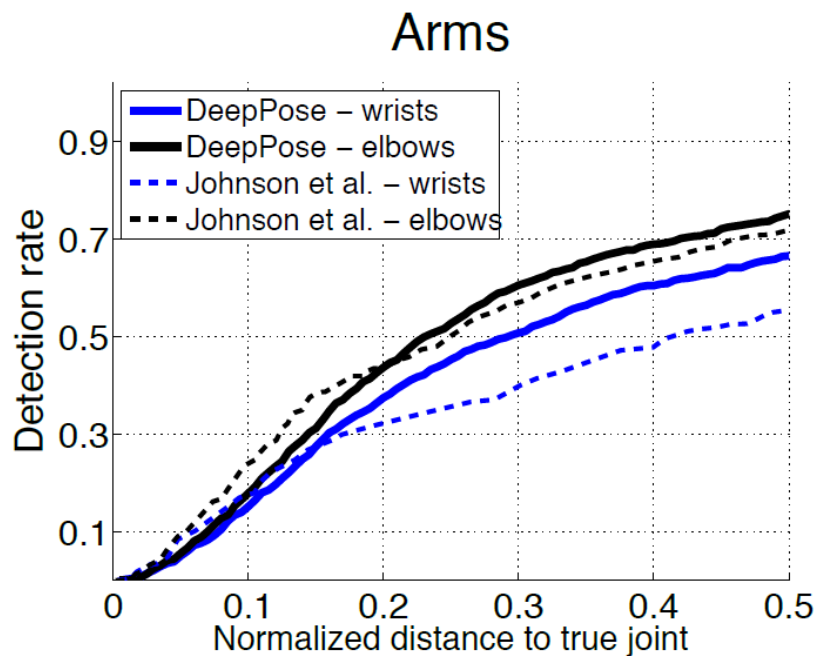
- <https://ctkim.tistory.com/101?category=906618>

# Single Person Pose Estimation

- Evaluation for direct regression method

## ❖ 실험 결과

- Percent of Detected Joints (PDJ) 지표를 Detection rate으로 표기
- 임계값을 0에서 0.5까지 증가시키며 성능을 확인
- 기존 state-of-the-art 모델보다 뛰어난 성능을 보이는 것을 네 관절에서 확인 가능

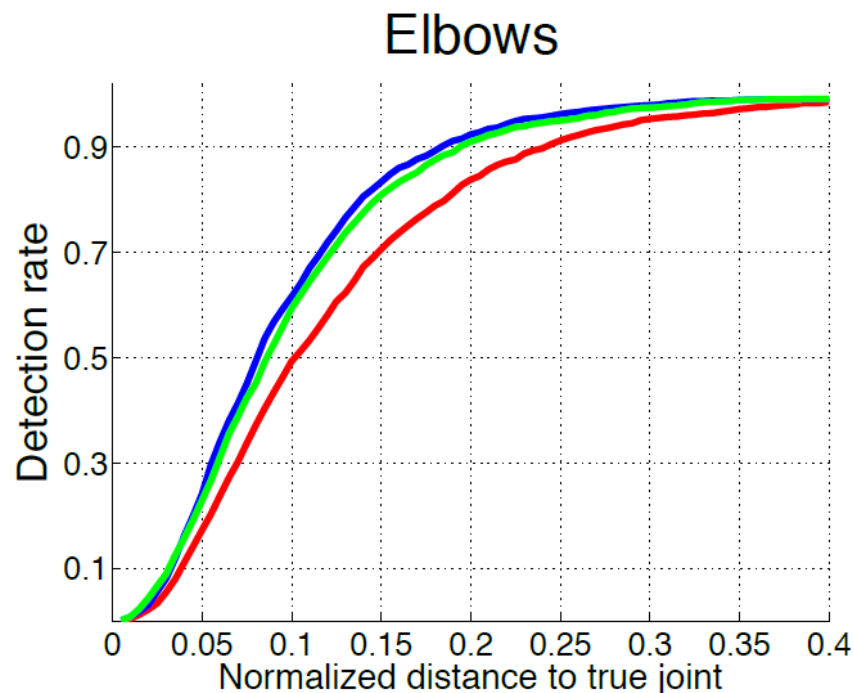
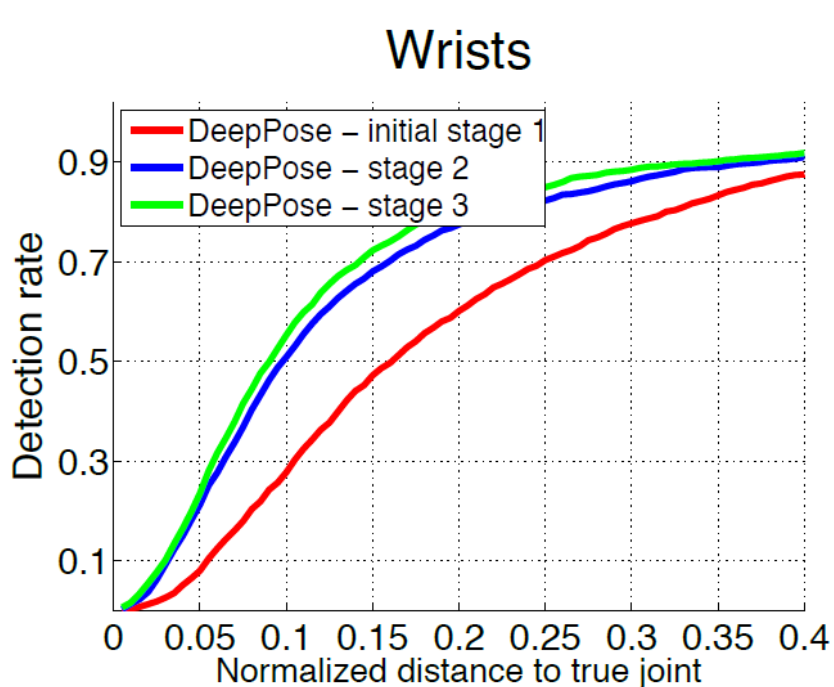


# Single Person Pose Estimation

- Evaluation for direct regression method

## ❖ 실험 결과

- Percent of Detected Joints (PDJ) 지표를 Detection rate으로 표기
- 임계값을 0에서 0.4까지 증가시키며 성능을 확인
- Stage 1만 진행하는 경우보다 관절별 모델 학습 시 관절 탐지 성능 증가



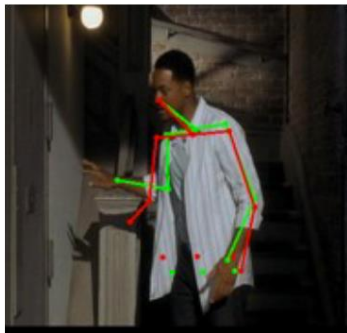
# Single Person Pose Estimation

- Evaluation for direct regression method

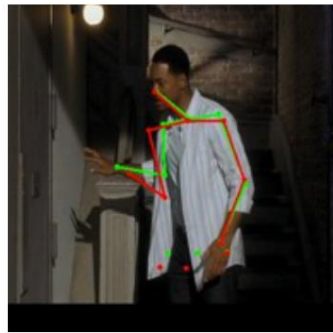
## ❖ 탐지 결과 시각화

- 초록색 선: 실제 관절 및 관절별 연결선
- 빨간색 선: 예측 관절 및 관절별 연결선

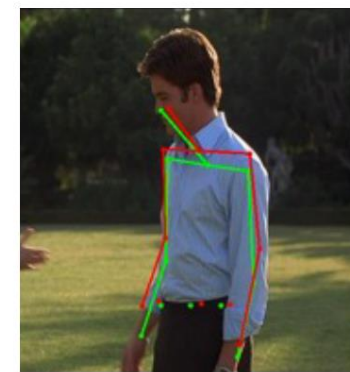
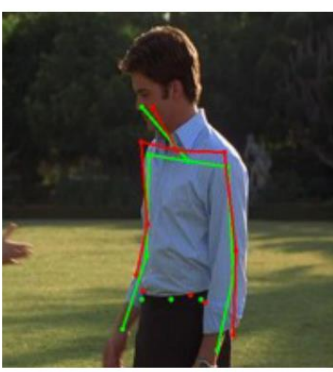
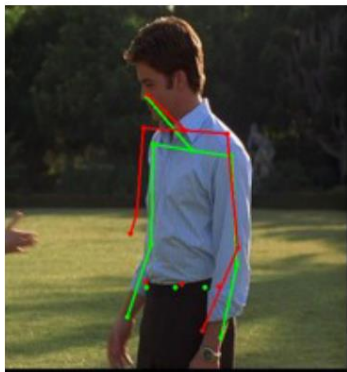
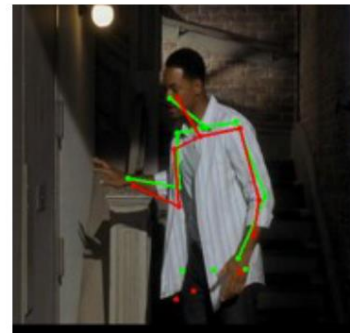
Initial stage 1



stage 2



stage 3



# Multi-Person Pose Estimation

- Top-down approach

## ❖ Mask R-CNN

- 2017년 IEEE conference on Computer Vision and Pattern Recognition에서 발표
- 저자들은 Facebook AI Research 소속이며 2021년 2월 22일 기준 10228회 인용
- Object Detection, Instance segmentation, **Human pose estimation**을 하나의 모델로 가능

## Mask R-CNN

Kaiming He Georgia Gkioxari Piotr Dollár Ross Girshick  
Facebook AI Research (FAIR)

### Abstract

*We present a conceptually simple, flexible, and general framework for object instance segmentation. Our approach efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover,*

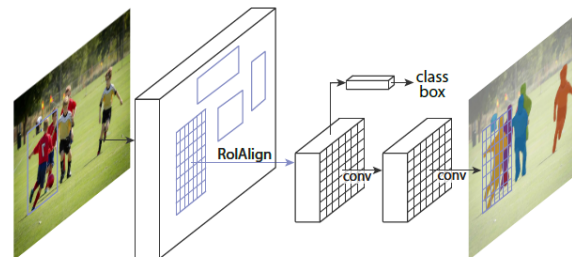


Figure 1. The Mask R-CNN framework for instance segmentation.

# Multi-Person Pose Estimation

- Top-down approach

## ❖ Multi-Person pose estimation

- 입력 이미지 내 사람이 **두 명 이상 존재**하는 경우
  - Top-down approach: 사람을 우선적으로 탐지 후 탐지 결과 내에서 관절별 좌표를 예측
  - Bottom-up approach: 탐지하고자 하는 관절에 대한 위치 예측 후 사람 별로 나누는 과정 진행



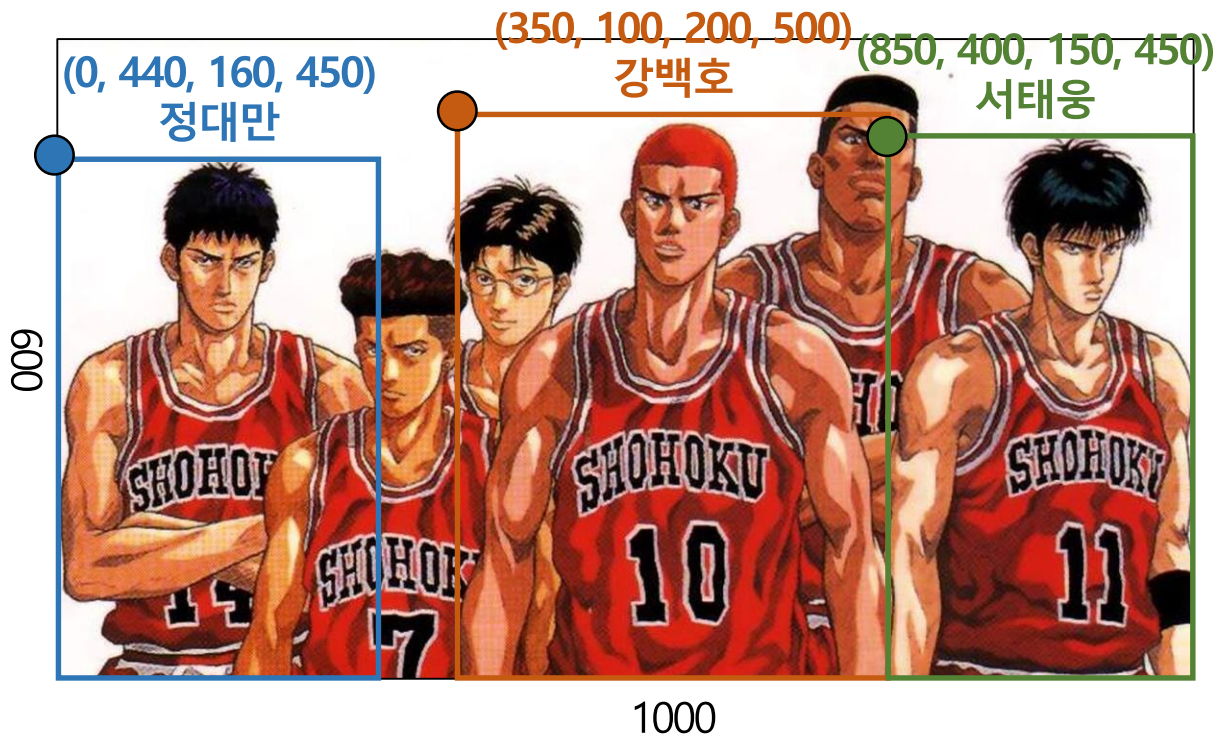


# Multi-Person Pose Estimation

- Overview of Mask R-CNN

## ❖ Object Detection and Instance segmentation

- 두 문제 모두 일반적으로 RGB 이미지를 입력 데이터로 사용
- Object detection: 탐지하고자 하는 범주에 대해 Bounding box regression & classification
- Instance segmentation: 관심있는 객체를 찾고 찾은 객체에 대해 Pixel-wise classification



## • 탐지하고자 하는 인물

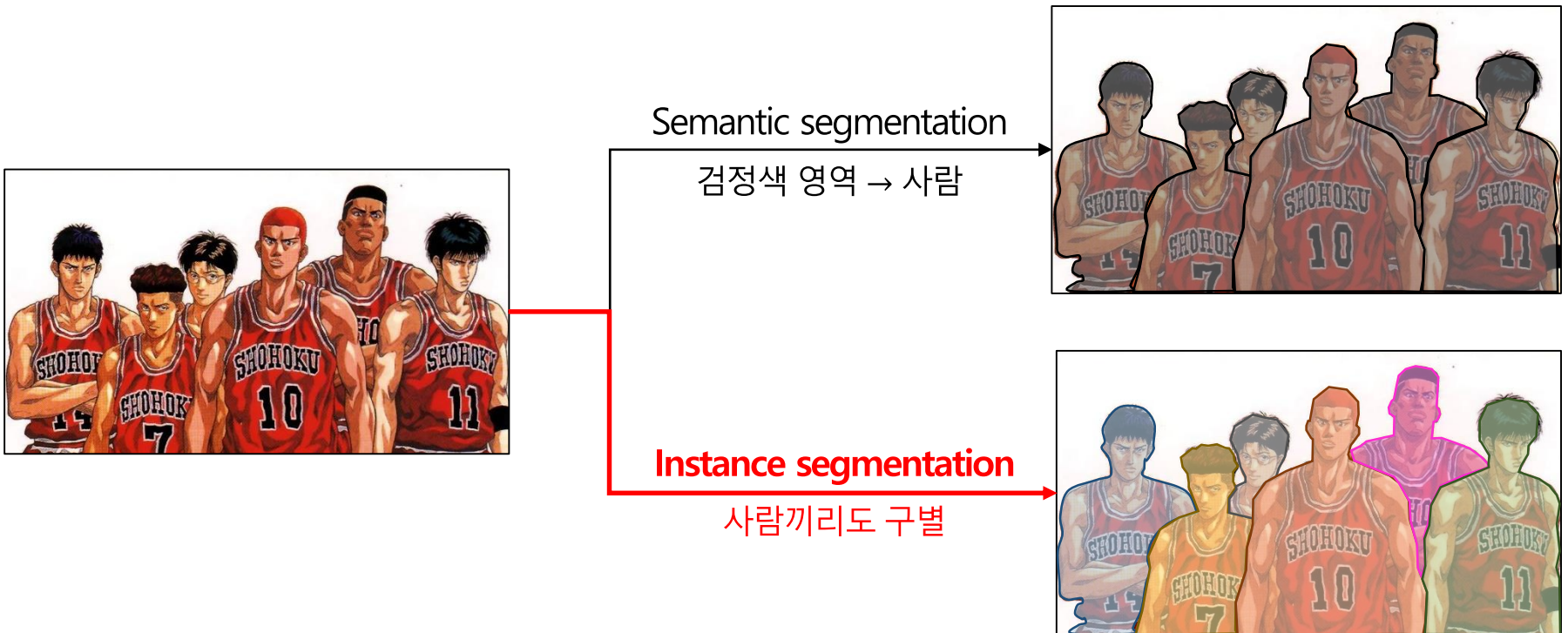
- 강백호
- 서태웅
- 정대만

# Multi-Person Pose Estimation

## - Overview of Mask R-CNN

### ❖ Object Detection and Instance segmentation

- 두 문제 모두 일반적으로 RGB 이미지를 입력 데이터로 사용
- Object detection: 탐지하고자 하는 범주에 대해 Bounding box regression & classification
- Instance segmentation: 관심있는 객체를 찾고 찾은 객체에 대해 Pixel-wise classification

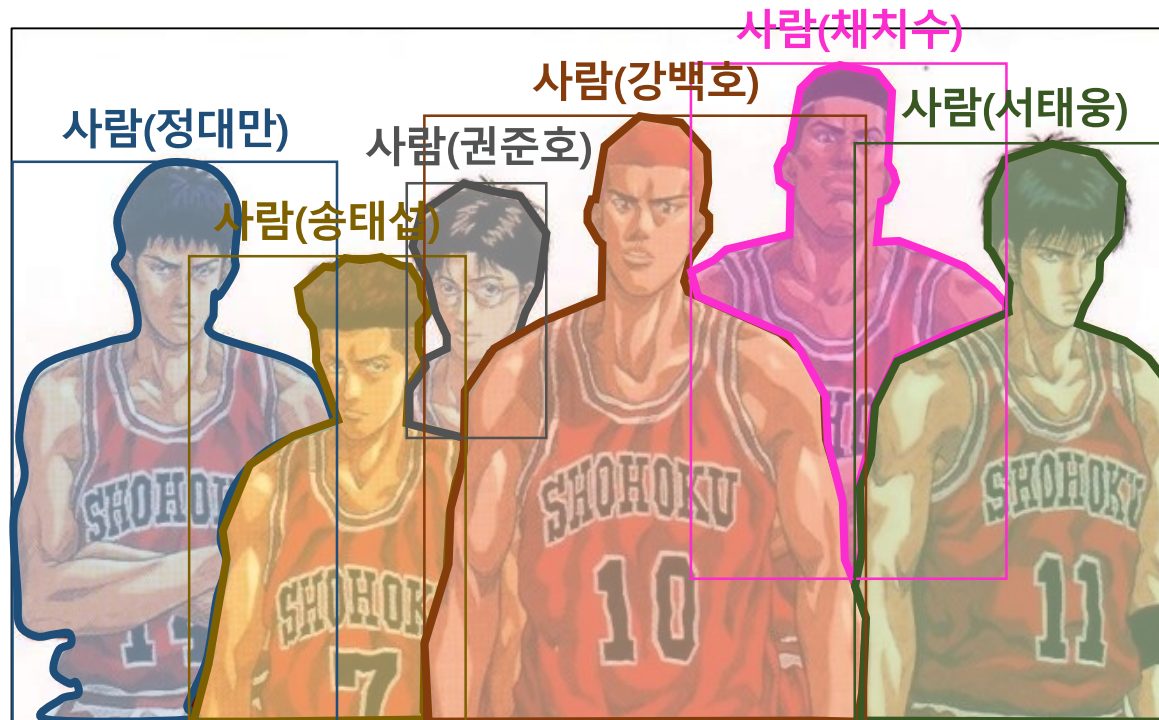


# Multi-Person Pose Estimation

- Overview of Mask R-CNN

## ❖ Mask R-CNN 기본 아이디어

- ① 객체가 있을만한 영역 탐지 (Bounding box regression) – 사각형 상자
- ② 탐지한 영역 내 어떠한 범주가 있을지 예측 (Classification) – 상자에 대한 범주
- ③ 상자 내 픽셀 ②에서 탐지한 범주인지 아닌지 분류 (Segmentation, Pixel-wise classification)

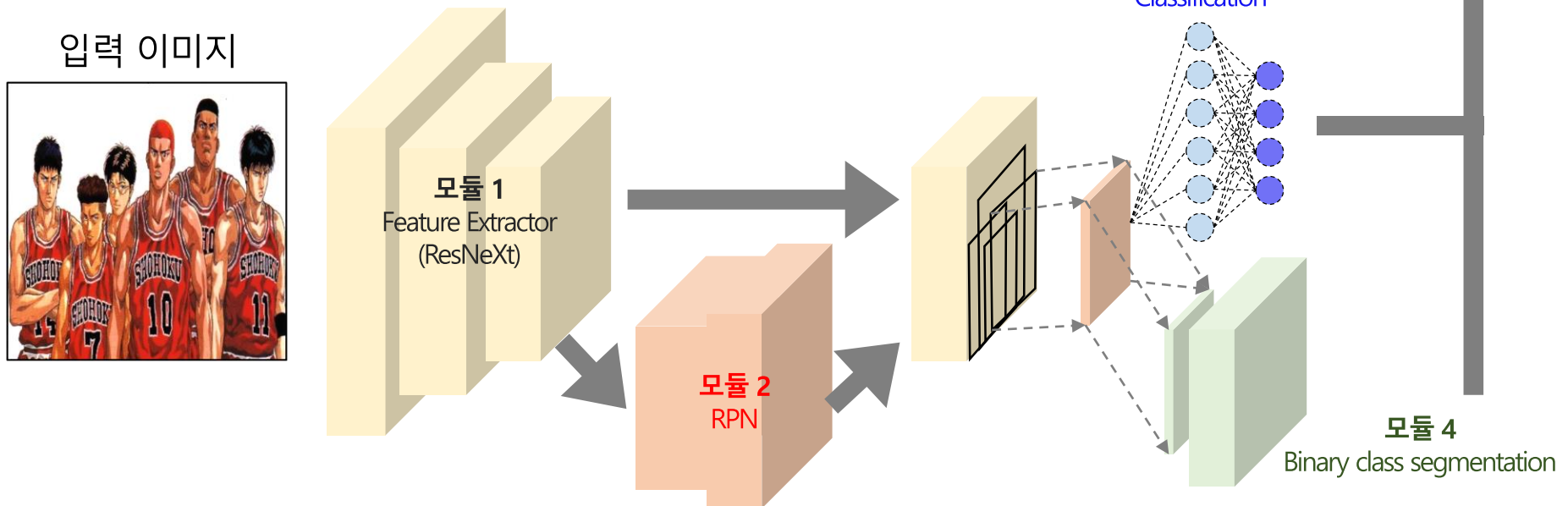


# Multi-Person Pose Estimation

- Model architecture of Mask R-CNN

## ❖ Mask R-CNN 구조 – 네 가지 모듈로 구성

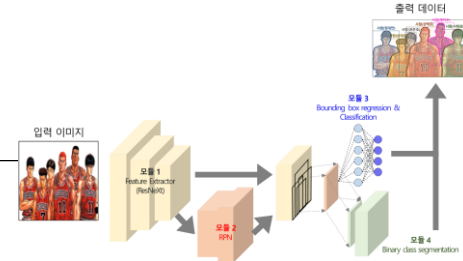
- **모듈 1:** Feature extractor (backbone) - ResNeXt
- **모듈 2:** Region Proposal Network (RPN)
- **모듈 3:** Bounding box regression and Classification
- **모듈 4:** Binary class segmentation



- Xie, S, Girshick, R, Dollár, P, Tu, Z, & He, K. (2017). Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1492-1500).

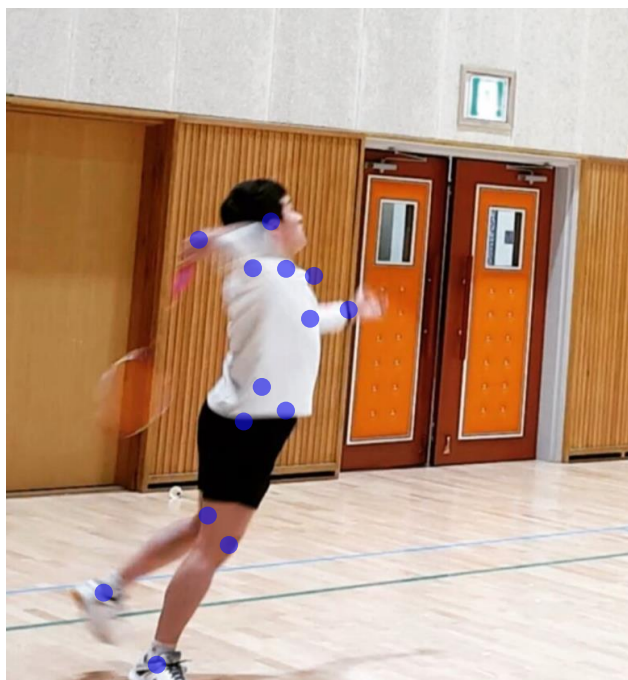
# Multi-Person Pose Estimation

- Mask R-CNN for human pose estimation

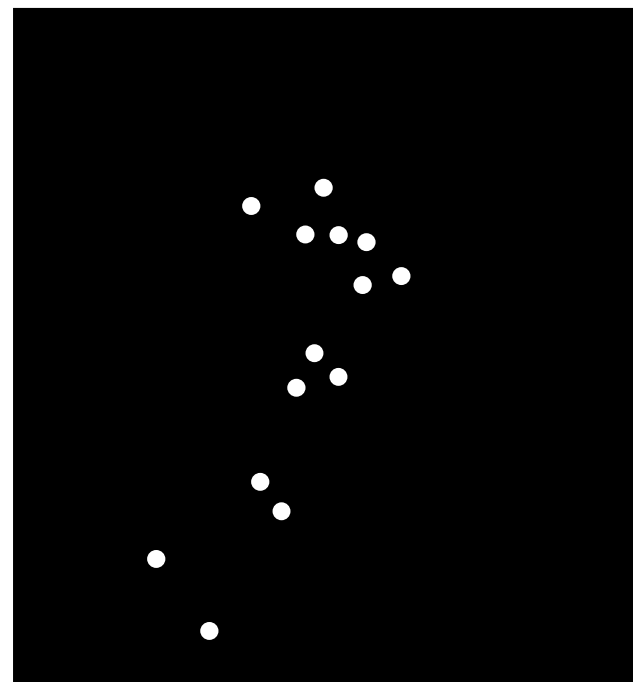


## ❖ Mask R-CNN 학습을 위한 관절별 좌표 값 변환

- 관절별 좌표를 픽셀 단위로 지정 → Mask 생성이라고 명명
- 특정 관절 좌표에 해당하는 픽셀은 1(흰색), 나머지는 0(검정색)인 Mask 생성



	X 좌표	Y 좌표
머리	120	370
왼쪽 어깨	120	330
왼쪽 팔꿈치	100	330
왼쪽 손목	80	350
.....	.....	.....
오른쪽 발목	50	70



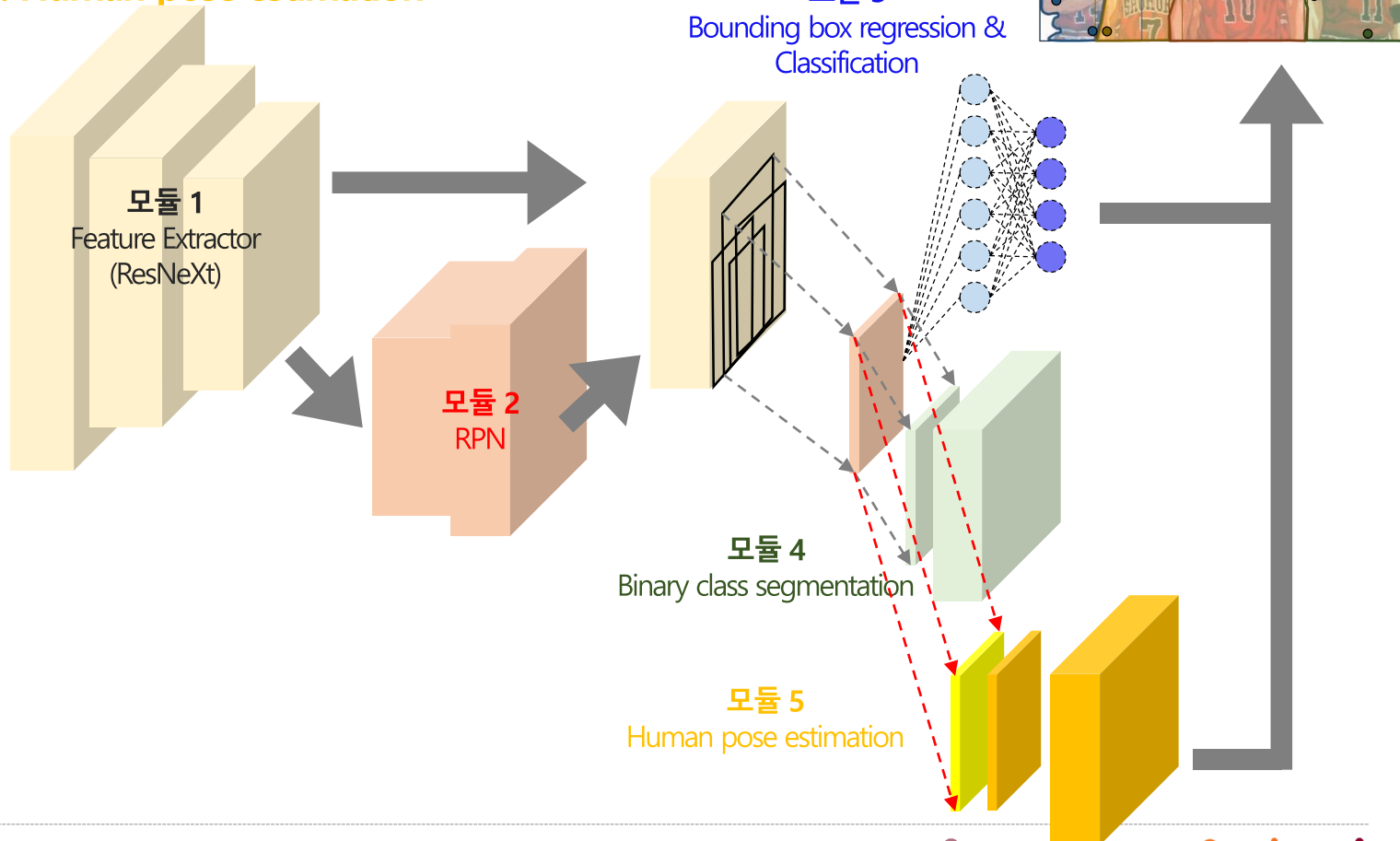
# Multi-Person Pose Estimation

- Mask R-CNN for human pose estimation

## ❖ Mask R-CNN 기본 구조에 Human pose estimation 가지 추가

- 기존 모듈 3, 4에 human pose estimation을 위한 모듈 5를 추가
- **모듈 5: Human pose estimation**

입력 이미지

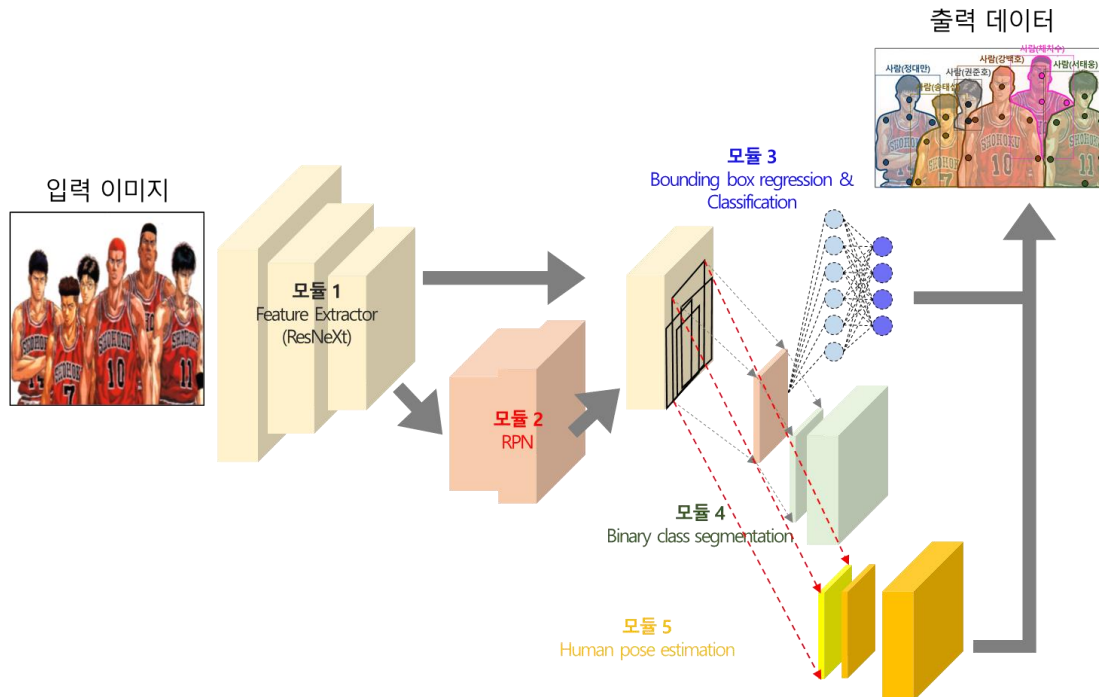


# Multi-Person Pose Estimation

- Mask R-CNN for human pose estimation

## ❖ Mask R-CNN 손실 함수

- $LOSS_{Mask\ R-CNN} = LOSS_{RPN\_reg} + LOSS_{RPN\_clf} + LOSS_{BB\_reg} + LOSS_{BB\_clf} + LOSS_{segment} + LOSS_{hpe}$
- 유사한 Task를 동시에 수행하면 성능이 증가할 것(Multi-task learning)
- $LOSS_{RPN\_reg}$ ,  $LOSS_{BB\_reg}$ : L1 손실 함수 사용
- 나머지는 Cross entropy 손실 함수 사용

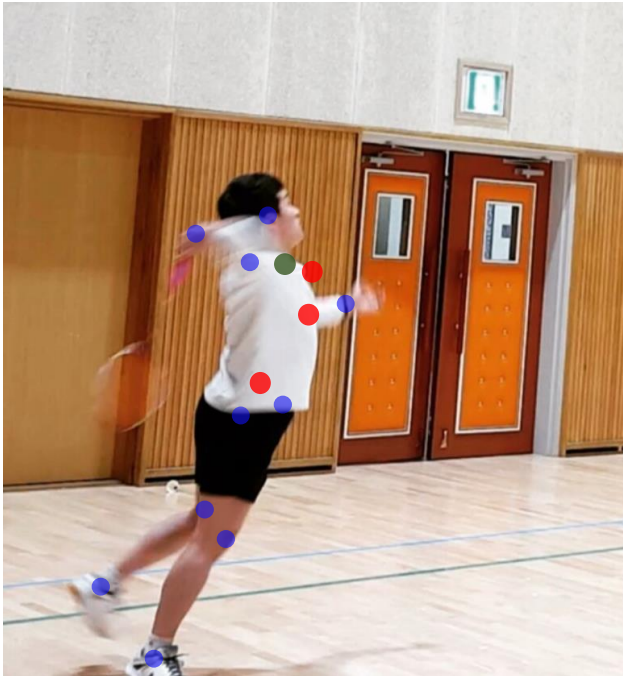


# Multi-Person Pose Estimation

- Dataset

## ❖ Microsoft common objects in context (MS COCO)

- Computer vision 관련 알고리즘을 위해 마이크로소프트에서 제공하는 벤치마크
  - Object detection, Instance segmentation, human pose estimation 등 많은 레이블 존재
- Human pose estimation을 위한 데이터 소개



	X 좌표	Y 좌표	V
머리	120	370	2
왼쪽 어깨	120	330	0
왼쪽 팔꿈치	100	330	0
왼쪽 손목	80	350	2
.....	.....	.....	.....
목	130	320	1

- 만약  $v == 0$ , 이미지에 존재 X
- 만약  $v == 1$ , 가려진 경우
- 만약  $v == 2$ , 확실하게 보임

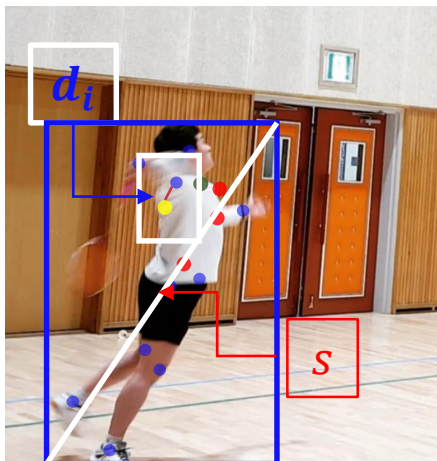


# Multi-Person Pose Estimation

- Evaluation metric for Multi-person pose estimation in Mask R-CNN

## ❖ Multi-person pose estimation 평가 지표

- Object keypoint similarity (OKS) based mean Average Precision
- ( $v > 0$ ) 인 좌표 수: 전체 관절 수( $k$ ) - 3 = 13 → 분모 (예제에선  $k = 16$ )
- ( $v > 0$ ) 인 관절에 대해서 아래 식 계산
  - $d_i$ :  $i$ 번째 **관절 예측 좌표**와 **실제 좌표 사이** 유클리디언 거리
  - $s$ : 해당 객체에 대한 Bounding box 대각선 길이
  - $k_i$ :  $i$ 번째 관절에 대한 상수 (사전에 지정)



	X 좌표	Y 좌표	V
머리	120	370	2
<b>왼쪽 어깨</b>	<b>120</b>	<b>330</b>	<b>0</b>
<b>왼쪽 팔꿈치</b>	<b>100</b>	<b>330</b>	<b>0</b>
왼쪽 손목	80	350	2
.....	.....	.....	.....
<b>목</b>	<b>130</b>	<b>320</b>	<b>1</b>

파란색 상자 내 객체에 대한 OKS

$$\frac{\sum_{i=1}^k \exp\left(-\frac{d_i^2}{2s^2k_i^2}\right) \delta(v_i > 0)}{\delta(v_i > 0)}$$

- <https://ctkim.tistory.com/101?category=906618>

# Multi-Person Pose Estimation

- Evaluation metric for Multi-person pose estimation in Mask R-CNN

## ❖ 실험 결과

- 여러 문제 상황을 한번에 학습하는 Multi-task learning 방식의 Mask R-CNN 학습 방식
- 인간에 대한 여러 정보를 사용해 Human pose estimation 성능 증가를 확인

	$AP_{person}^{BB}$ (OD 성능)	$AP_{person}^{Mask}$ (IS 성능)	$AP^{HPE}$ (HPE 성능)
Faster R-CNN (OD)	52.5	-	-
Mask R-CNN (OD and IS)	<b>53.6</b>	<b>45.8</b>	
Mask R-CNN (OD and HPE)	50.7		64.2
Mask R-CNN (OD, IS, and HPE)	52.0	45.1	<b>64.7</b>

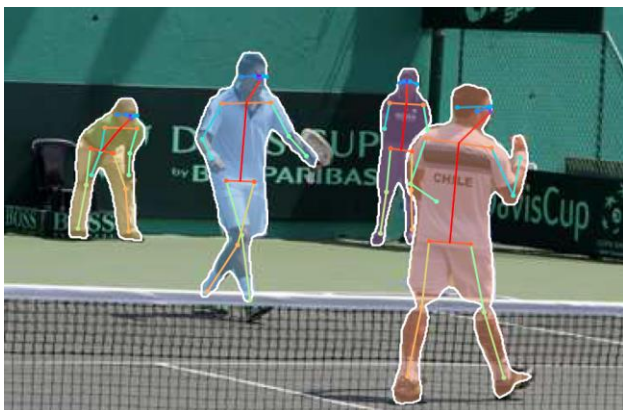
- Object detection (OD), Instance segmentation (IS), and Human pose estimation (HPE)

# Multi-Person Pose Estimation

- Evaluation visualization for Multi-person pose estimation in Mask R-CNN

## ❖ 모델 예측 결과 시각화

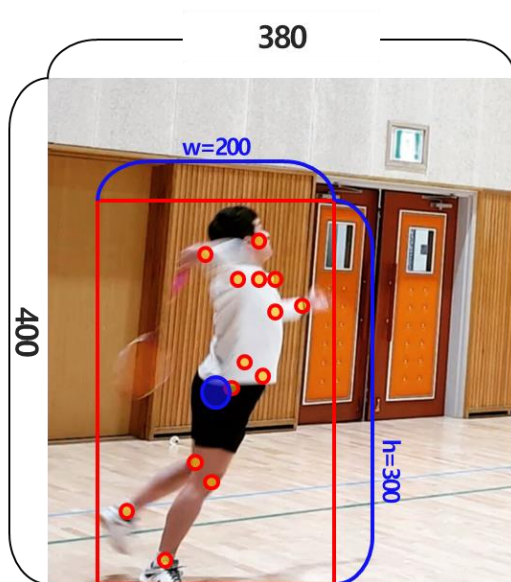
- 입력 이미지 내 여러 사람에 대해서도 정확히 탐지하는 것을 확인 가능



# Conclusion

## ❖ 결론

- Human pose estimation: 인간의 관절별 좌표를 예측하는 문제
  - 입력 데이터는 일반적으로 RGB 이미지 (3D human pose estimation 문제도 존재)
  - 출력 데이터는 인간에 해당하는 Bounding box와 관절별 좌표
- 해당 문제는 Single person estimation과 Multi-person estimation으로 구분



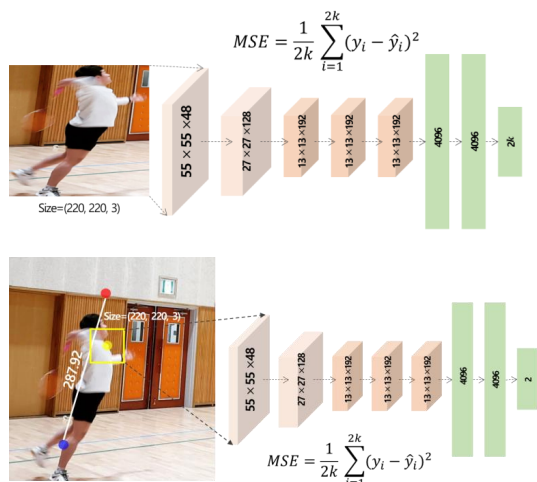
- 사람에 대한 Bounding Box:  $(x, y, h, w)$ 
  - Box의 중심:  $(x, y) = (120, 15)$
  - 높이와 너비:  $h, w$
- 탐지하고자 하는 관절 개수 = 16개
  - ① 머리:  $(x, y) = (120, 370)$
  - ② 왼쪽 어깨:  $(x, y) = (120, 330)$
  - ③ 왼쪽 팔꿈치:  $(x, y) = (100, 330)$
  - ④ 왼쪽 손목:  $(x, y) = (80, 350)$
  - 
  - 
  - 
  - ⑬ 오른쪽 발목:  $(x, y) = (50, 70)$

# Conclusion

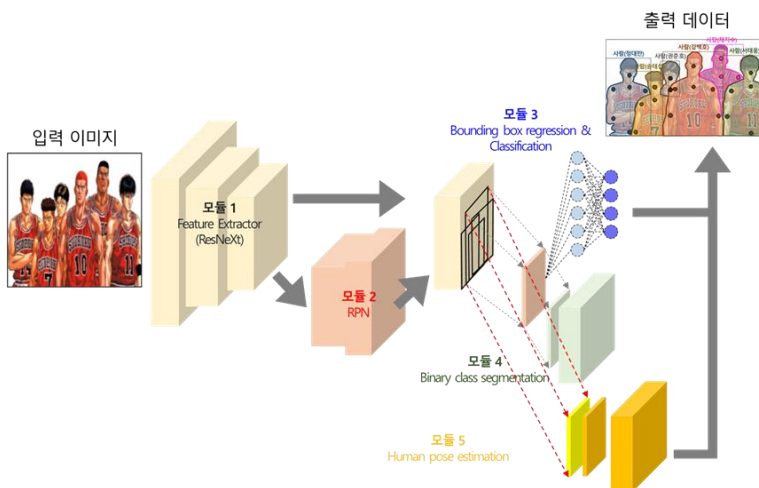
## ❖ 결론

- Human pose estimation: 인간의 관절별 좌표를 예측하는 문제
  - 입력 데이터는 일반적으로 RGB 이미지 (3D human pose estimation 문제도 존재)
  - 출력 데이터는 인간에 해당하는 Bounding box와 관절별 좌표
- 해당 문제는 **Single person estimation**과 **Multi-person estimation**으로 구분
  - **Single person estimation** 대표 알고리즘: DeepPose (2014)
  - **Multi-person estimation** 대표 알고리즘: Mask R-CNN (2017)

### Single person estimation



### Multi-person estimation



---

# 감사합니다.