

## ※ Pose Estimation

- Video에서 사람이 여러명 있을 때 관절을 어떻게 찾아 내는 것 (Object Detection과 유사)

## ※ Human Pose Estimation

- Pose : 사람 Key Point의 집합 (Tree Structure)
- 3D Pose Estimation은 Motion Capture에 주로 사용
- Detection이 어려운 이유
  - : 관절들이 명확히 Define이 되어야 함 (Appearance가 좋아야 함)
  - : 안보이는 영역 (Occlusion)은 어떻게 할 것인지?
  - : 보이지 않는 Body (Ground Truth가 없다)을 어떻게 학습 할 것인지?

## ※ Two Approaches

- Top-Down Approach : Bounding Box를 찾고 Body Point를 찾는 것
- Bottom-Up Approach : Body Point를 찾고 Grouping 해서 사람을 찾는 것

## ※ Classical Approach

- Pictorial Structure : Graph의 각 Node가 Key Point. Edge는 Body Part 간의 연결선
- 가장 많이 사용하는 Structure는 Tree Structure Graph
- $L^* = \operatorname{argmin}$  공식참조 !! , 두개의 Body Point가 얼마나 벗어나 있는가?
- Pictorial Structure Training
  - : 정형화된 것을 벗어난 것을 잘 Inference 할 수 있다
  - : Template, Body Part를 학습 해야 함
  - : 벗어난 정도의 Penalty 주기 위해 Weight을 줌 (Body Part별 위치에 대한 Weight)

**단점 : Hand Designed Feature라서 한계가 있다**

## ※ Deep Learning based Approaches

- Body Part Detection의 Feature를 배우자
- 사람의 Priory Structure가 들어가지 않음
- CNN은 Body Part의 문제를 Regression 하는 것

## ※ CNN for Pose Estimation (Deep Pose, Stacked Hour Glass)

### ※ Deep Pose

- Local Appearance is Weak
- Global Consistency is Import
- Body Part만 보면 어딘지 알기 어려워 Context가 중요하다.
- Body Part의 전체 영역을 다루는 것
- Image로 부터 2D Matrix로 표현 하는 것 : CNN

**단점 : Receptive Field를 키우면 Localization 정보를 잃는다. Accuracy가 떨어진다. 용량을 많이 써서 비효율**

- Body Part를 찾고 주어진 Size로 Window하고 Neural Net을 써서 얼마나 이동을 해야 하는지 Prediction 함  
→ Body Part를 수정

## ※ Pose Estimation

- Heat Map(Score Map) : Special Domain에서 Heat Map을 뿌리는 것. Body Part 마다 Score Map 보여 줌
- Image Space에서 Mapping 이기 때문에 간단 함

## ※ Multi Resolution Heat Map Regressor

- FCN와 유사한 Architecture (그러나 Input이 2개 이상임)
- Input Size보다 작은 Resolution에서 Output이 나옴
- Localization은 2D → 2D로 사용하기 쉽다
- Feature를 Cropping

## ※ Multi Resolution : Atrous, Pyramid, Pooling

## ※ Resolution : Decoding, Skip Connect, Pooling Switches

## ※ Local Evidence

- Incorporating a Large Context : Receptive Field를 늘려서 더 많은 Body Part를 보자
- : Receptive Field를 늘리는 방법 : Pooling Window 키우기, Convolution Filter Size 키우기
- Spatial Pyramid Pooling, Multiple Image Resolution 등

## ※ Convolution Pose Machine

- CNN with Iterative Refinement
- Receptive Field를 키워서 정확도를 높이자

## ※ Iterative Update of Prediction

- Refinement를 정의해 놓고 점점 고쳐 가는 것

## ※ Iterative Error Feedback

- Refinement해서 Error가 작아지는 방향으로 조금 민다 (반복 수행)

## ※ Stacked Hourglass

- Architecture로 제안 한 것
- Global Pathway : Encode global context with a large receptive field(stacking poolings)
- Residual (Skip) Connections : abstract spatial information at various scales