DGCN: Dynamic Graph Convolutional Network for Efficient Multi-Person Pose Estimation

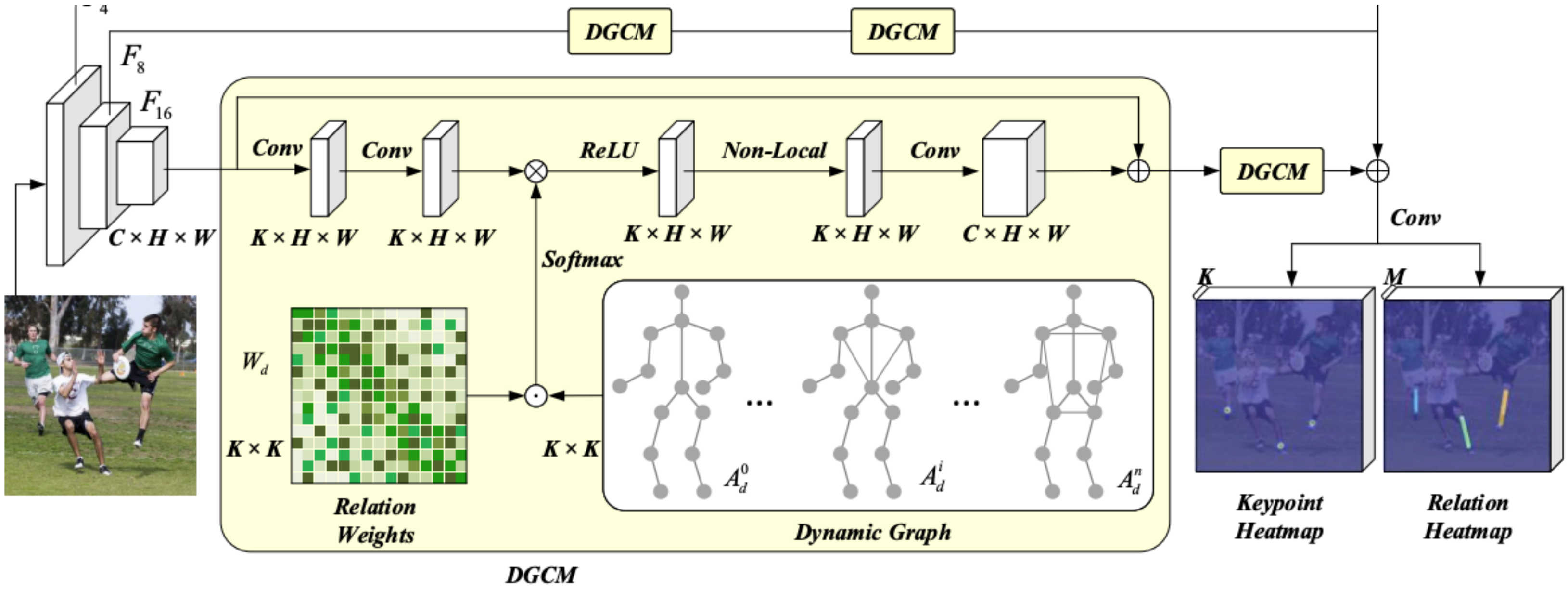
Bui Tien Cuong – 2019-35731

Presentation Overview

- Problem definition
- Dynamic Graph Convolutional Network (DGCN)
- Evaluation

Problem Definition

- Multi-person pose estimation = detecting human keypoints in an image
- Bottom-up methods: keypoints localization & grouping
- Relations between two nodes $\overset{keys}{\Longrightarrow}$ group points
- A graph of keypoints: **Node = point, Edge = 2-points relation**

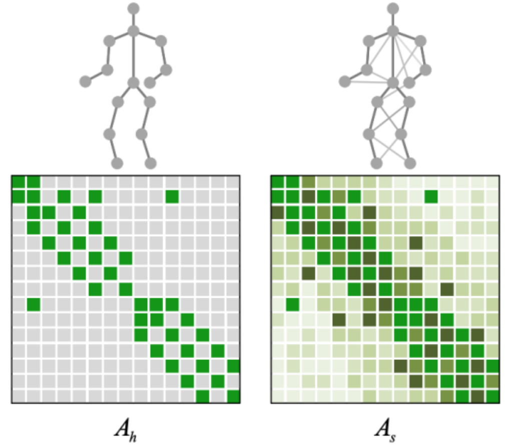


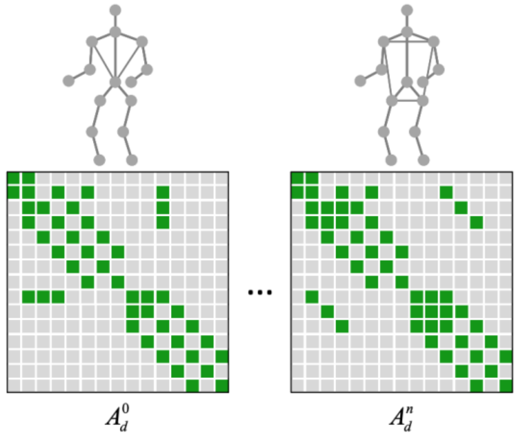
Dynamic Graph Convolutional Network (DGCN)

- Using Backbone CNNs (ResNet-50) => K keypoints
- Each keypoint <=> an output feature map
- Proposing a dynamic graph modeling method
- Generating keypoint heatmaps & relation heatmaps

GCN for Keypoints Graph Modeling

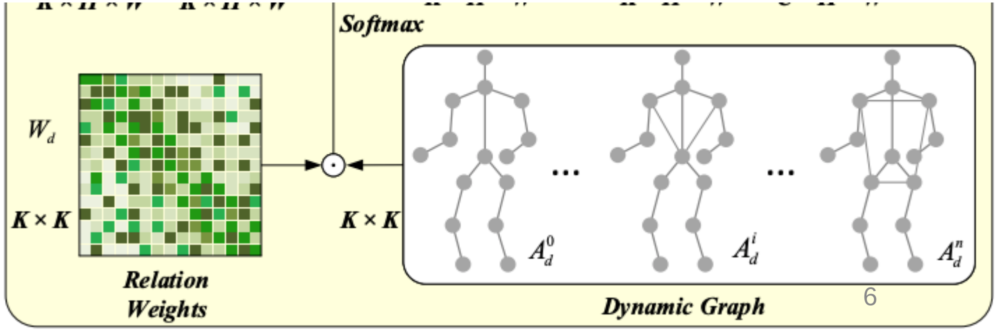
- Keypoint graph G = (V, E)
  - A keypoint set V = {v_i|i=1,..,K}
  - A limb set E = {e_i |i=1,…,M}
    - M: # hand-crafted limbs
- $A \in [0,1]^{kxk}$: adjacency matrix of G
- X_i^l: representation of v_i at the level l^{th}
  - $X^{l+1} = \sigma(WX^l\tilde{A})$
    - $\tilde{A}$: symmetrically normalized matrix of A





Dynamic Graph Convolutional Module (DGCM)

- $A_s^{ij} = softmax\left(\gamma \frac{1}{M_d^{ij}}\right)$, where $M_d^{ij} = \frac{1}{N}\sum_{n=0}^{N} \frac{1}{s_n} \left\|V^i - V^j\right\|_2$
- $A_d^{ij} \sim B(x, A_s^{ij})$: Generate x times of A_d^{ij} with Bernoulli distribution of A_s^{ij}
- $X^{l+1} = \sigma\left(WX^l\tilde{A}\right)$ => $X^{l+1} = \sigma\left(WX^l softmax(W_d \odot \tilde{A})\right)$
- Add a learnable weight matrix W_d
- Use a softmax operation => automatically learn edges
- All elements of $\tilde{A}$ is 1 in testing



Evaluation

- Backbone: using ResNet-50 pre-trained on ImageNet
- Evaluating on two pose estimation datasets: MS COCO and MPII
- The approach is straightforward and easy to apply to other models.

THANK YOU !