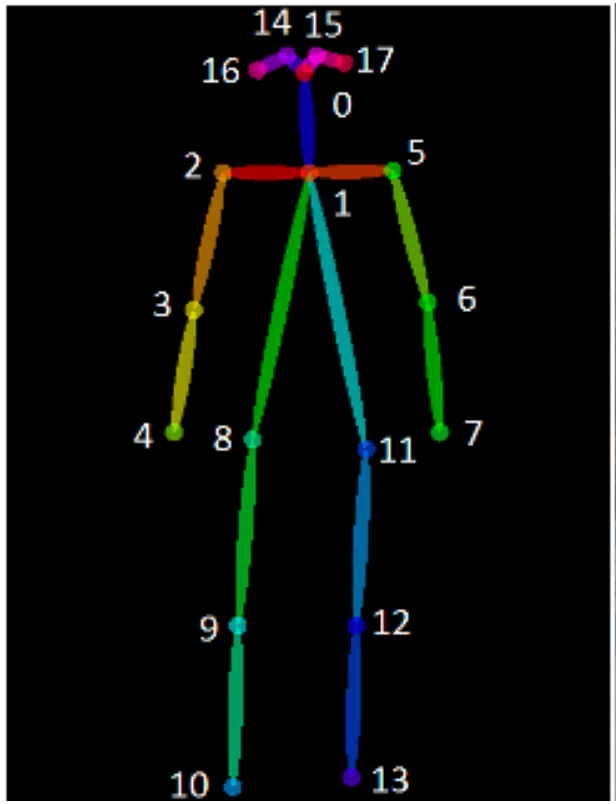
2D Human Pose Estimation

Junjie Cao @ DLUT

Spring 2019

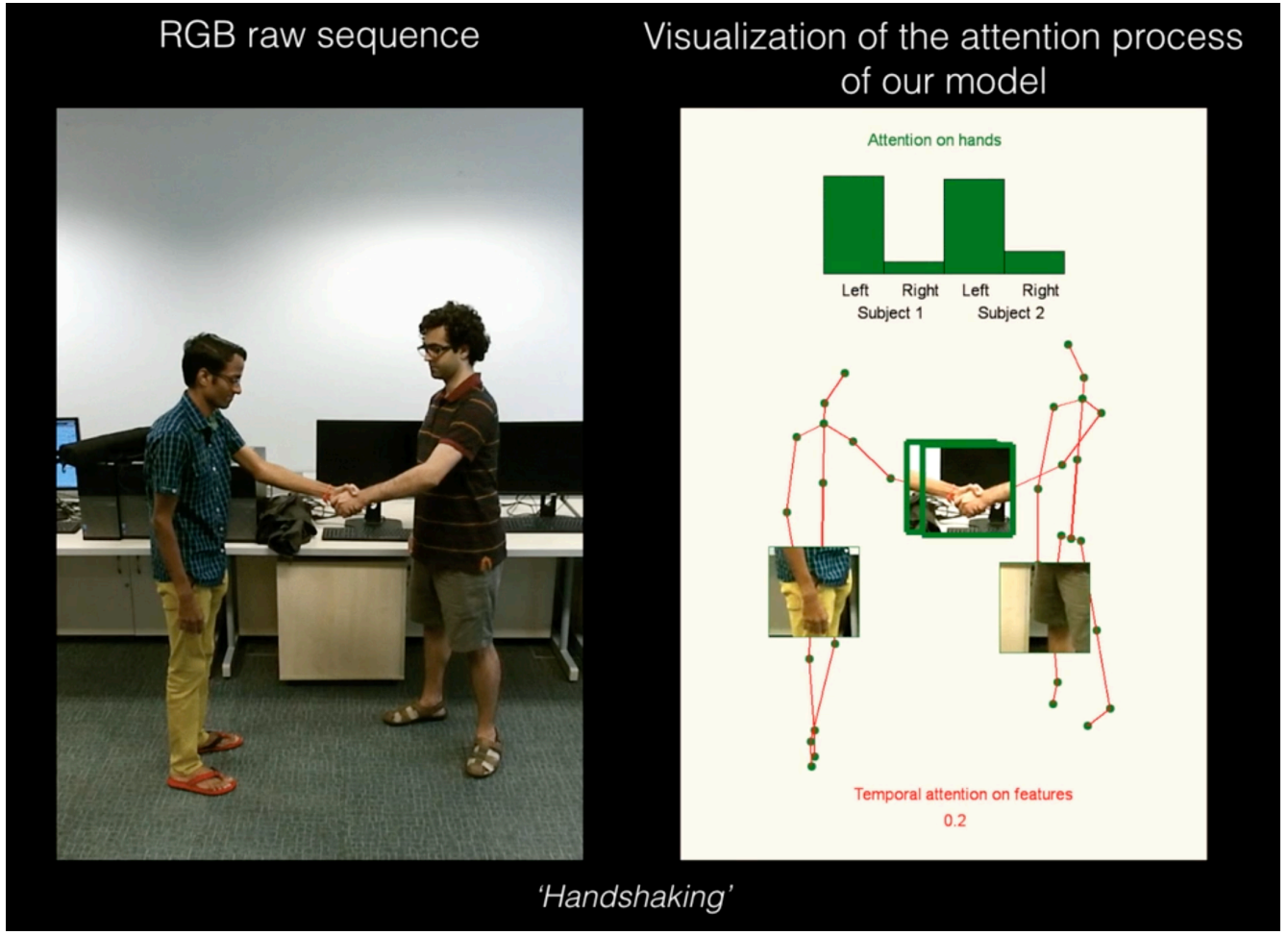
What is Human Pose Estimation?

- localization of human **joints** (also known as **keypoints** - elbows, wrists, etc) in images or videos
- Single pose v.s. multiple poses



Why Human Pose Estimation?

- a crucial step towards **understanding people** in images and videos
- heavily used in Action recognition, Animation, Gaming



Pose-conditioned Spatio-Temporal Attention for Human Action Recognition

SHOT TYPE

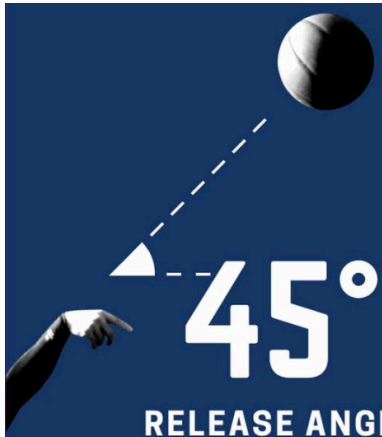
HomeCourt can identify these types of shots:

Catch and Shoot
Off the Dribble
Layup
Free Throw

0.7s

RELEASE TIME

The time from catching the ball to releasing it out of your hands. A fast release time will increase your chances of getting your shot off without being blocked.



45°

RELEASE ANGLE

The higher it is, the more arc in your shot. Developing a good arc in your shot allows the ball to have a bigger target as it enters the hoop.

8.5 MPH

SPEED

How fast you are moving before you shoot. Running your drills at game speed will improve your performance when it counts the most.

115°

LEG ANGLE

A smaller angle means bending your knees and getting low. A tighter leg angle puts you in a more stable position to rise up for a shot or drive to the hoop.

12″

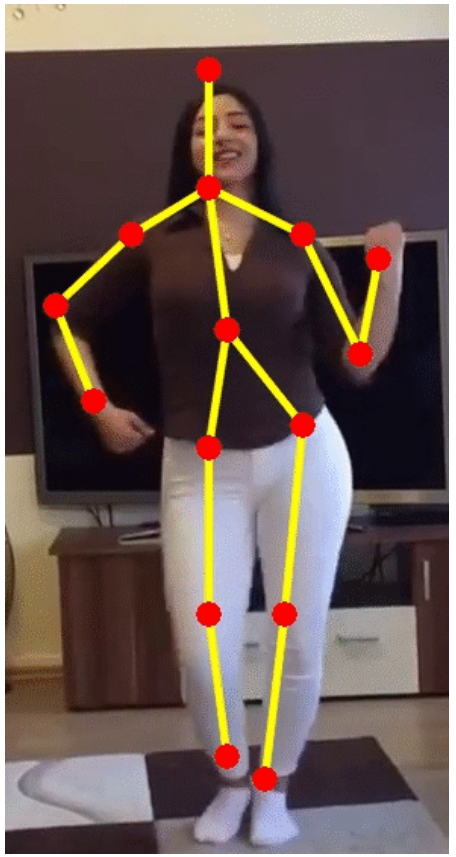
VERTICAL

Your distance from the ground at the peak of your jump. To optimize the power of your jump shot, release the ball right before you reach the peak of your jump.

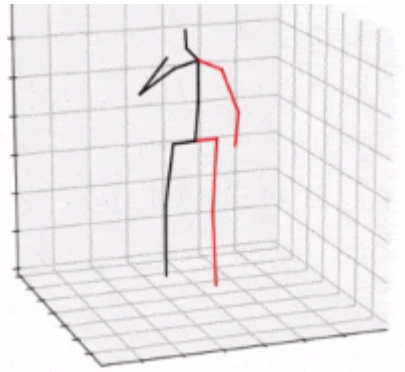
https://www.homecourt.ai/

2D vs 3D pose

- localization of human **joints** (also known as **keypoints** - elbows, wrists, etc) in images or videos



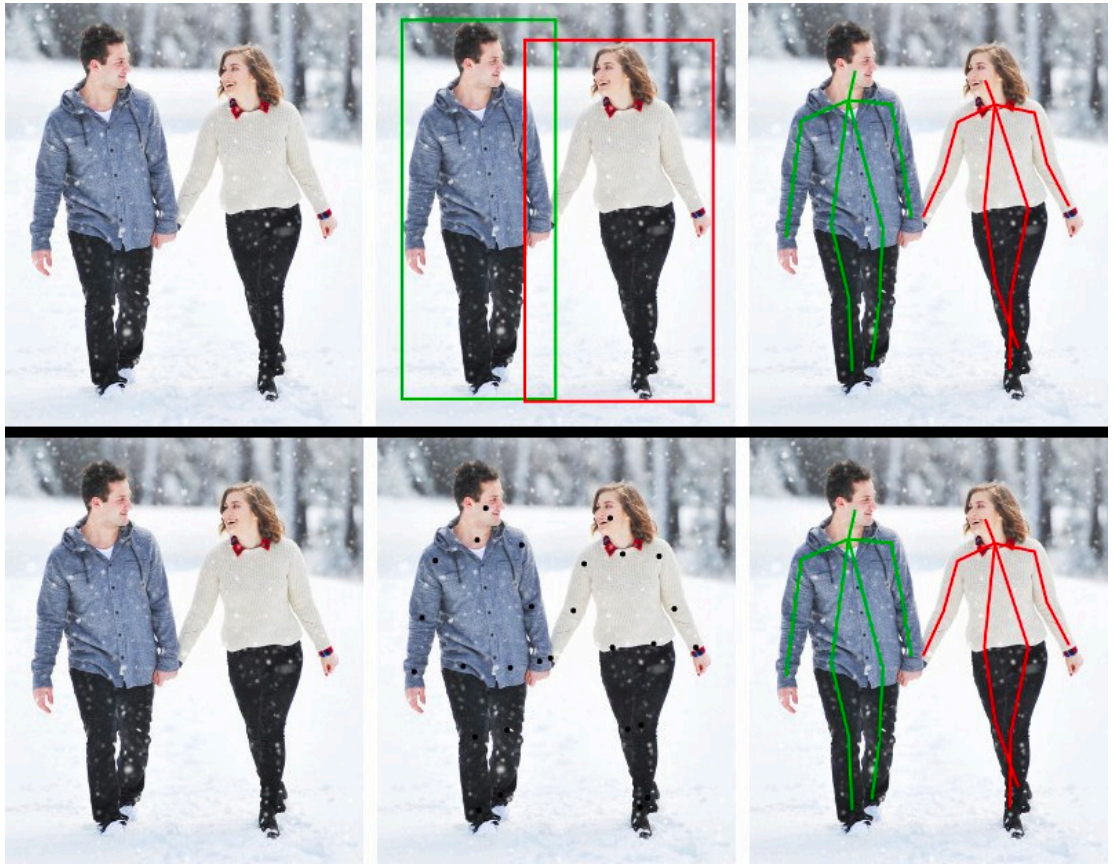
Key points: (x_i, y_i)



Key points: (x_i, y_i, z_i)

2 categories of approaches: Top-down vs Button-up

- **top-down**
  - incorporate a person detector first, followed by estimating the parts and then calculating the pose for each person.
- **bottom-up**
  - detect all parts in the image (i.e. parts of every person), followed by associating/grouping parts belonging to distinct persons.



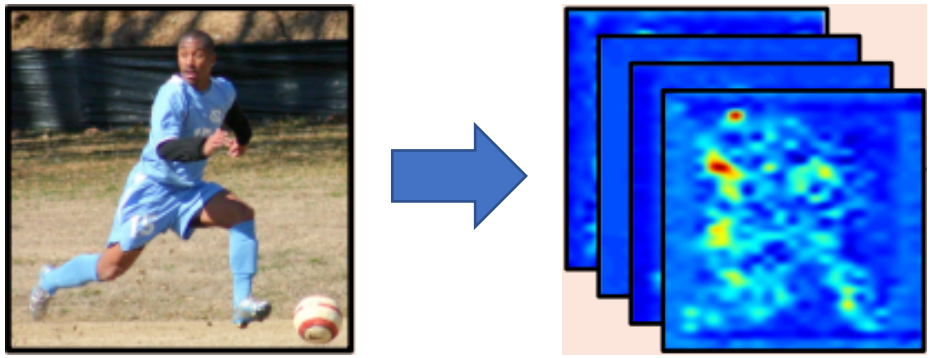
Detection VS. Regression

Detection

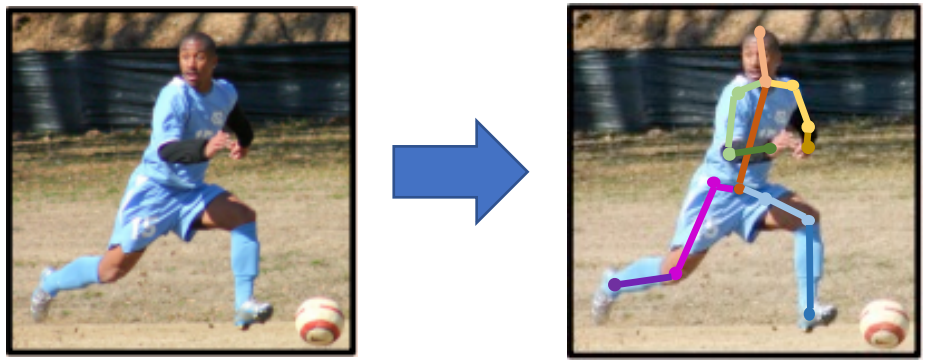
- Per-pixel classification
- Output: likelihood score maps

Regression

- Location regression
- Output: key points location



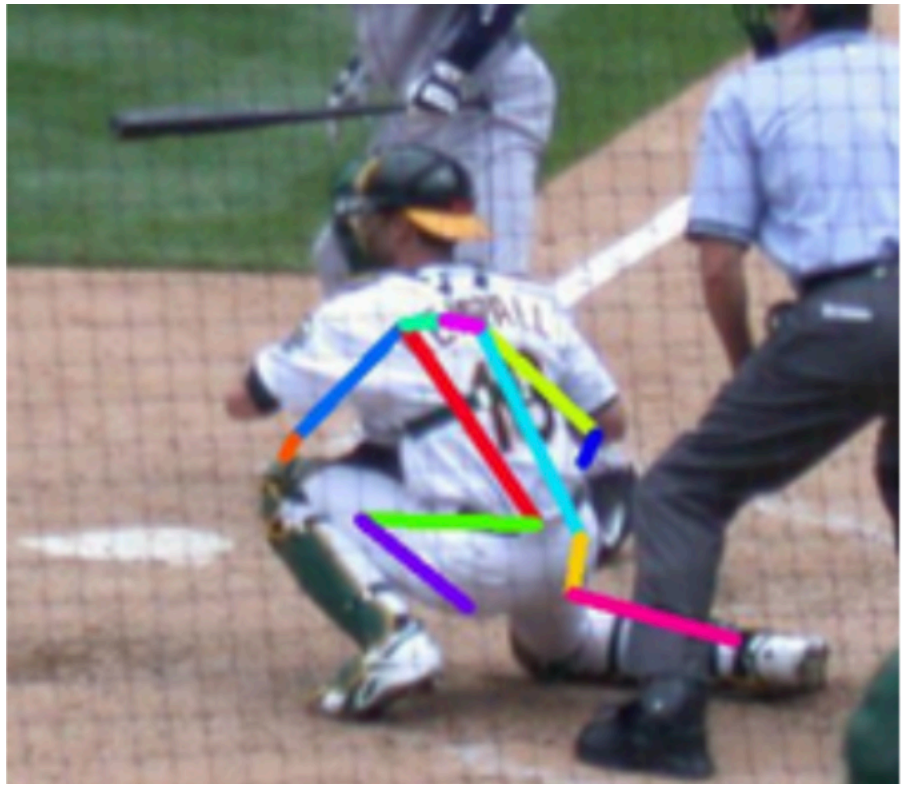
$\mathbf{H}_k$: Heatmap



$\mathbf{J}_k$: Joint

Why is it hard?

- Strong articulations, small and barely visible joints, occlusions, clothing, and lighting changes make this a difficult problem.



Deep Learning based approaches

- "DeepPose" by Toshev et al, researches began to shift from classic approaches to Deep Learning.
- Deep learning strategy has yielded drastic improvements on standard benchmarks.
- Main building block: ConvNets

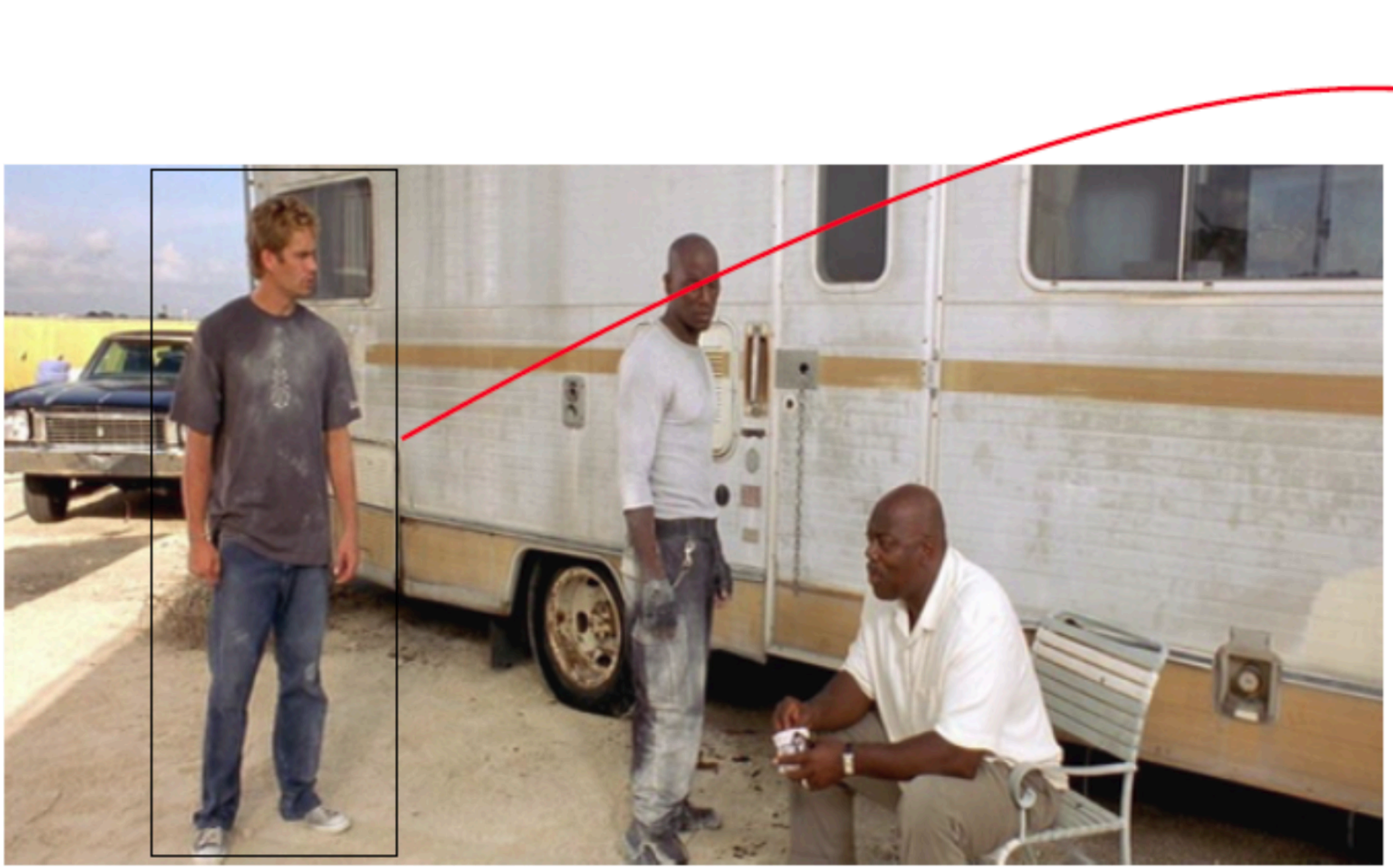
2D pose

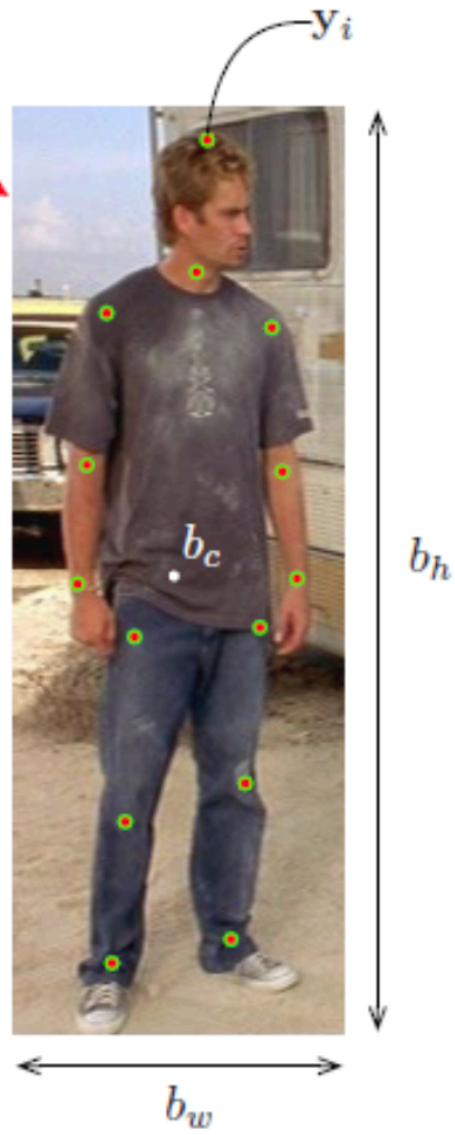
DeepPose: Cascade of CNN, CVPR'14

Alexander Toshev, Christian Szegedy
google

- pose estimation == a CNN-based regression problem towards body joints.
  - use a cascade of such regressors to refine the pose estimates
- Even if certain joints are hidden, they can be estimated if the pose is reasoned about holistically.
- CNNs naturally provide this sort of reasoning and demonstrate strong results.

1. Pose Vector





1. Pose Vector

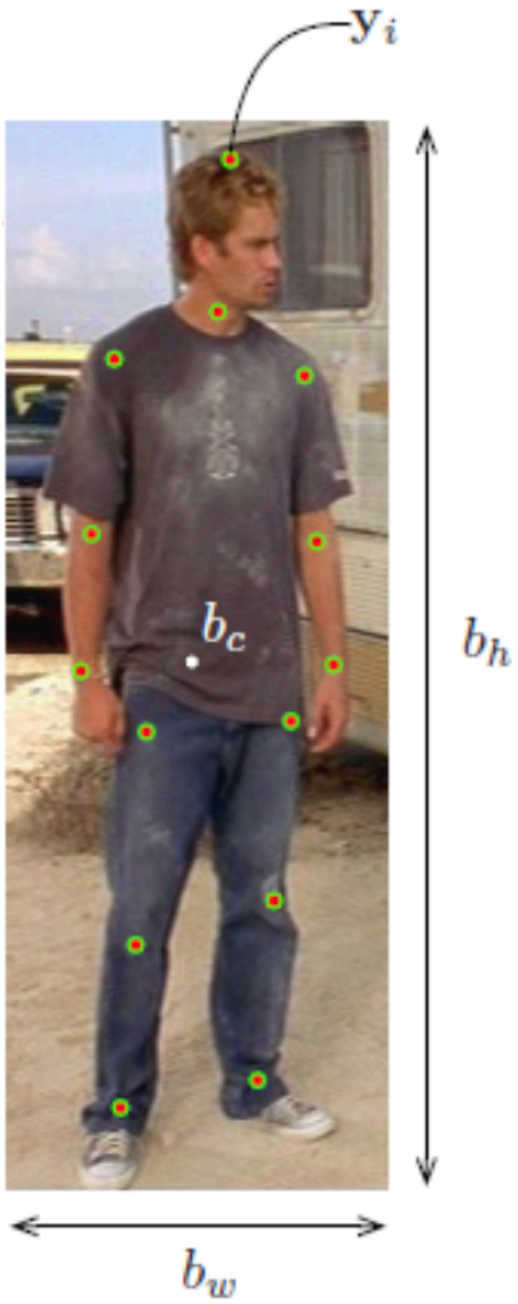
- pose vector **y**: $\mathbf{y} = (\ldots, \mathbf{y}_i^T, \ldots)^T, i \in \{1, \ldots, k\},$
  - The ith joint: **yi**=(xi, yi)
  - xi, yi are absolute coordinates within the image.

- A labeled image: (*x*, ***y***)
  - *x* is image data
  - ***y*** is the ground truth pose vector

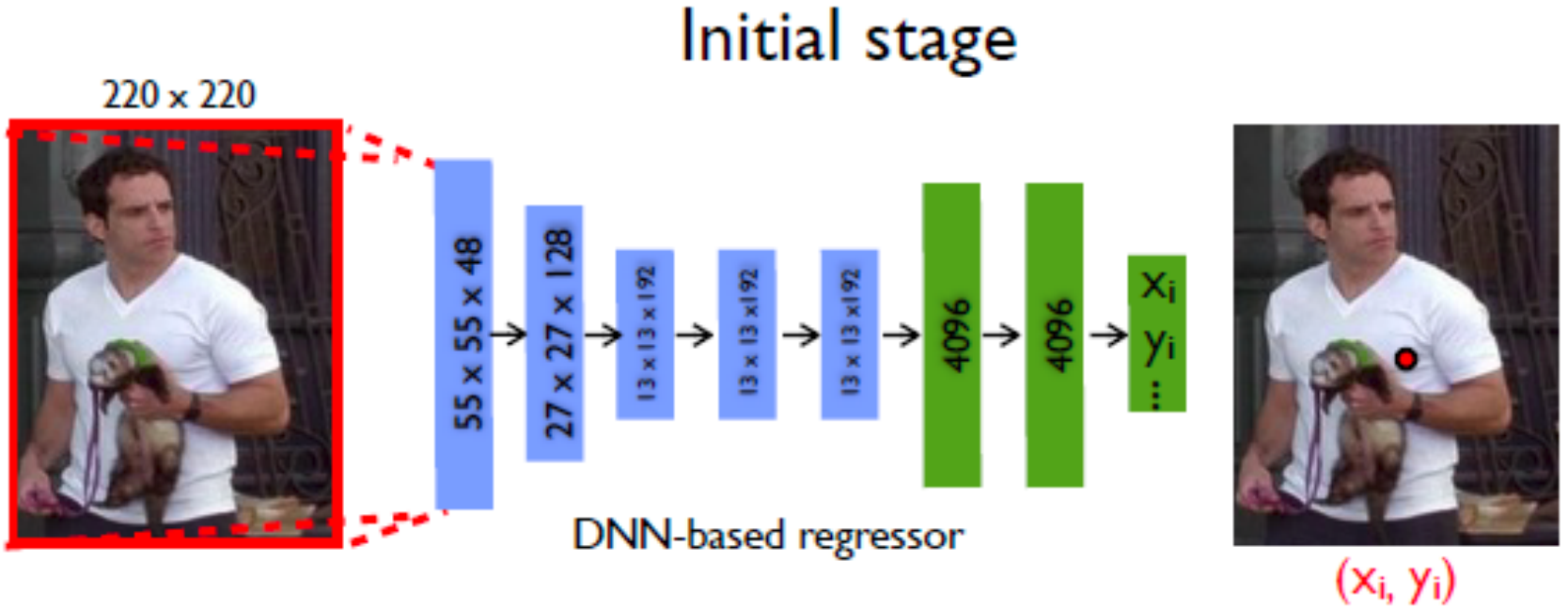
- **Normalize the coordinates *yi*** w.r.t. a box *b*

$$N(\mathbf{y}_i; b) = \begin{pmatrix} 1/b_w & 0 \\ 0 & 1/b_h \end{pmatrix} (\mathbf{y}_i - b_c)$$

- ***N*(*y*; *b*)** is the normalized pose vector.
- *N*(*x*; *b*) is a crop of the image *x* by the bounding box *b*.



CNN As Pose Regressor



$$y^* = N^{-1}(\psi(N(x);\theta))$$

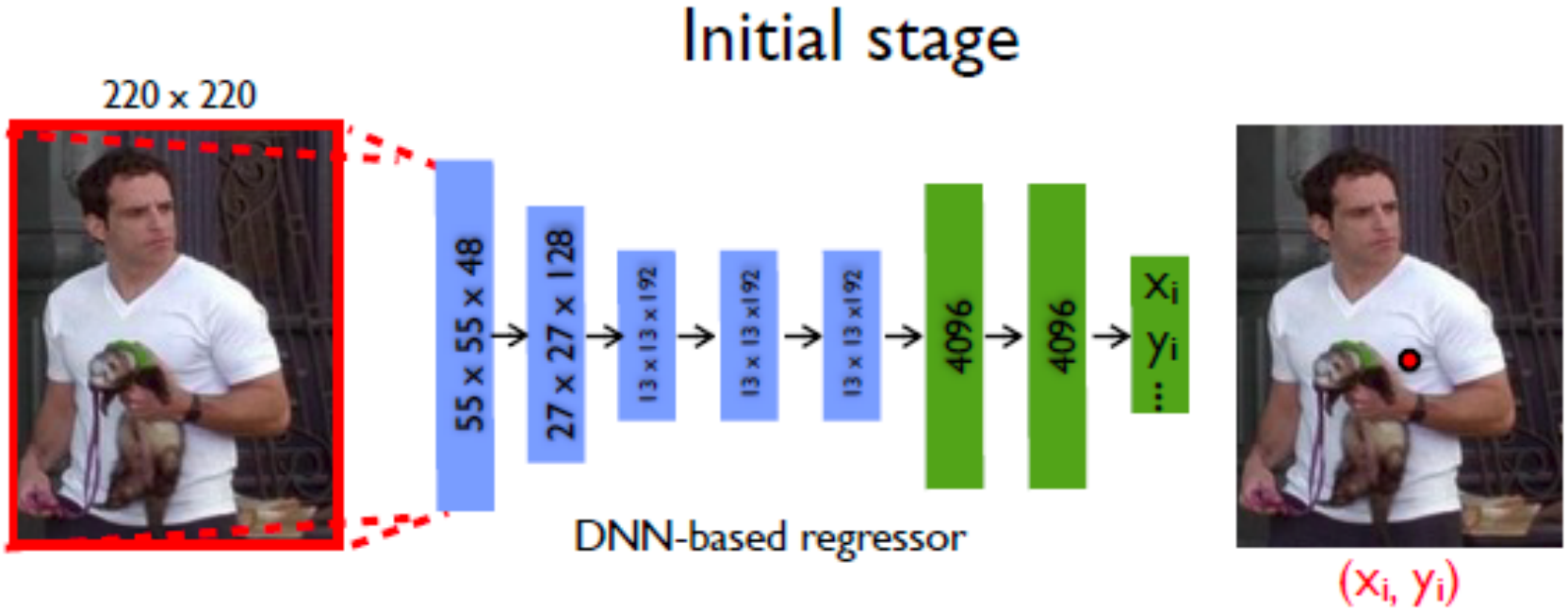
AlexNet

Input: an image of predefined size, N(x)

With trained parameters θ, ψ outputs the normalized 2k prediction of joints, N(**y**) .

y* can be obtained by denormalization *N*^-1.

CNN As Pose Regressor



$$y^* = N^{-1}(\psi(N(x);\theta))$$

Loss:

$$\arg\min_{\theta} \sum_{(x,y)\in D_N} \sum_{i=1}^{k} ||\mathbf{y}_i - \psi_i(x;\theta)||_2^2$$

$$D_N = \{(N(x), N(\mathbf{y}))|(x,\mathbf{y}) \in D\}$$

Mini-batch size is 128. Data is augmented with random translation and left/right flip.

Total number of parameters is 40M.

Cascade of Pose Regressors

- joint estimation is based on the full image and thus relies on context.
- fixed input size of 220 × 220 =>
  - limited capacity to look at detail
  - it learns filters capturing pose properties at coarse scale.
- It is insufficient to precisely localize the body joints.

Cascade of Pose Regressors

$$\text{Stage } 1: \quad \mathbf{y}^1 \leftarrow N^{-1}(\psi(N(x; b^0); \theta_1); b^0)$$

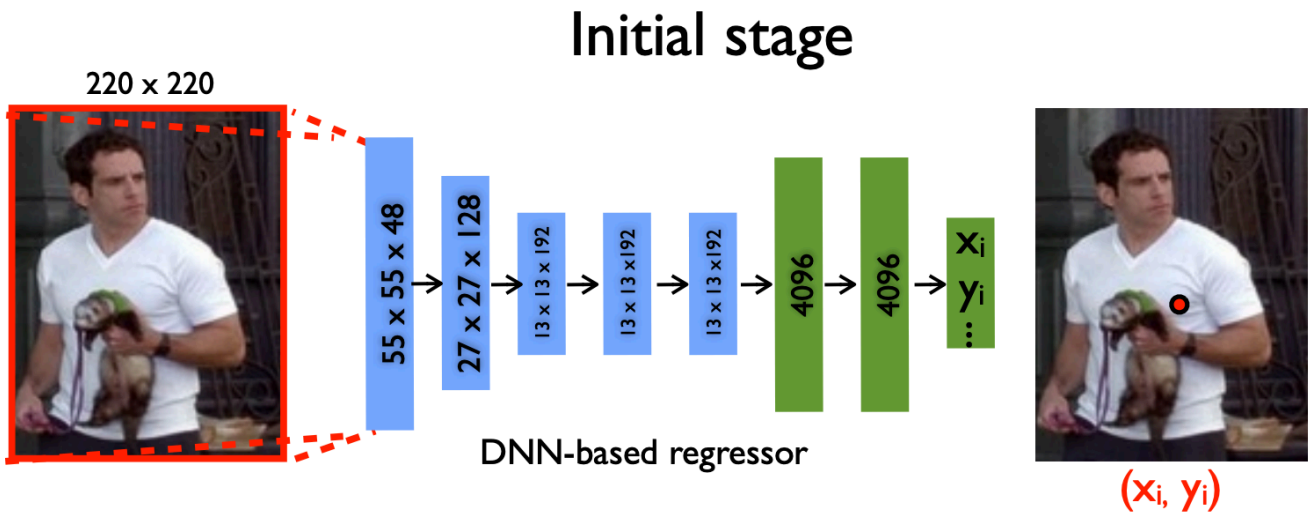
$$\text{Stage } s: \quad \mathbf{y}_i^s \leftarrow \mathbf{y}_i^{(s-1)} + N^{-1}(\psi_i(N(x; b); \theta_s); b)$$

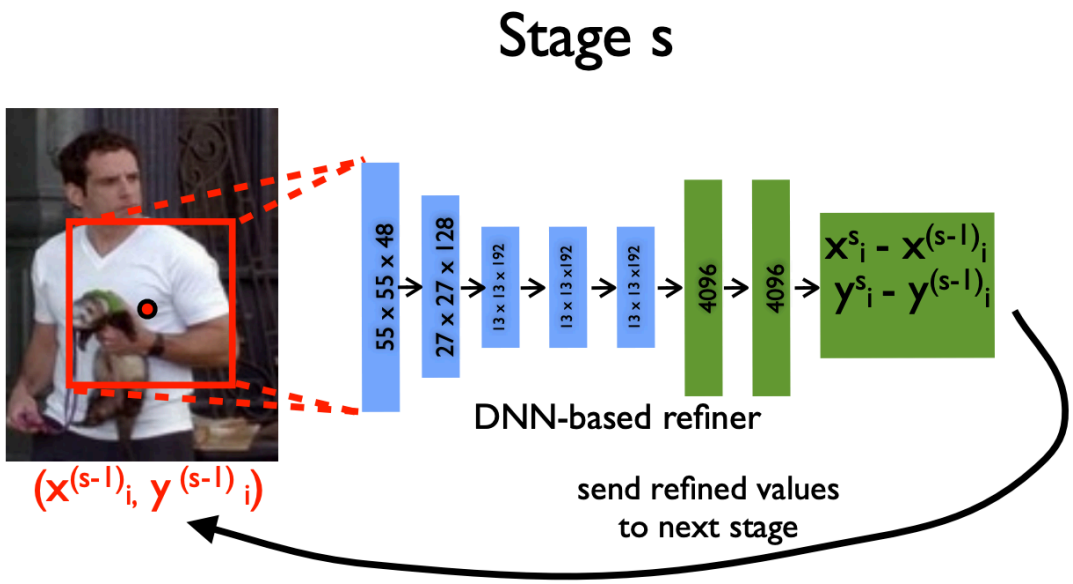
$$\text{for } b = b_i^{(s-1)}$$

$$b_i^s \leftarrow (\mathbf{y}_i^s, \sigma \text{diam}(\mathbf{y}^s), \sigma \text{diam}(\mathbf{y}^s))$$

- b^0 is full image or a box obtained by person detector.

Cascade of Pose Regressors





Stage 1: a network for all k joints predict locations

$$D_N = \{(N(x), N(\mathbf{y}))|(x, \mathbf{y}) \in D\}$$

$$\arg\min_{\theta} \sum_{(x,y)\in D_N} \sum_{i=1}^{k} ||\mathbf{y}_i - \psi_i(x;\theta)||_2^2$$

Stage s: a network for a joint or a Siamese Network? predict a displacement

$$\begin{aligned} D_A^s = \{&(N(x;b), N(\mathbf{y}_i;b))| \\ &(x, \mathbf{y}_i) \sim D, \delta \sim \mathcal{N}_i^{(s-1)}, \\ &b = (\mathbf{y}_i + \delta, \sigma \mathbf{diam}(\mathbf{y}))\} \end{aligned}$$

$$\theta_s = \arg\min_{\theta} \sum_{(x,\mathbf{y}_i)\in D_A^s} ||\mathbf{y}_i - \psi_i(x;\theta)||_2^2$$

Datasets

- **Frames Labeled In Cinema (FLIC)**: 4000 training and 1000 test images from Hollywood movies with diverse poses and diverse clothing. For each labeled human, 10 upper body joints are labeled.

- **Leeds Sports Dataset (LSP)**: 11000 training and 1000 testing images from sports activities with challenging in terms of appearance and especially articulations. The majority of people have 150 pixel height. For each person the full body is labeled with total 14 joints.

Metrics

- **Percentage of Correct Parts (PCP)**: measures detection rate of limbs, where a limb is considered detected if the distance between the two predicted joint locations and the true limb joint locations is at most half of the limb length.

- **Percent of Detected Joints (PDJ)**: A joint is considered detected if the distance between the predicted and the true joint is within a certain fraction of the torso diameter. By varying this fraction, detection rates are obtained for varying degrees of localization precision.

Cascading CNN for refinement helps to improve the results

Comments

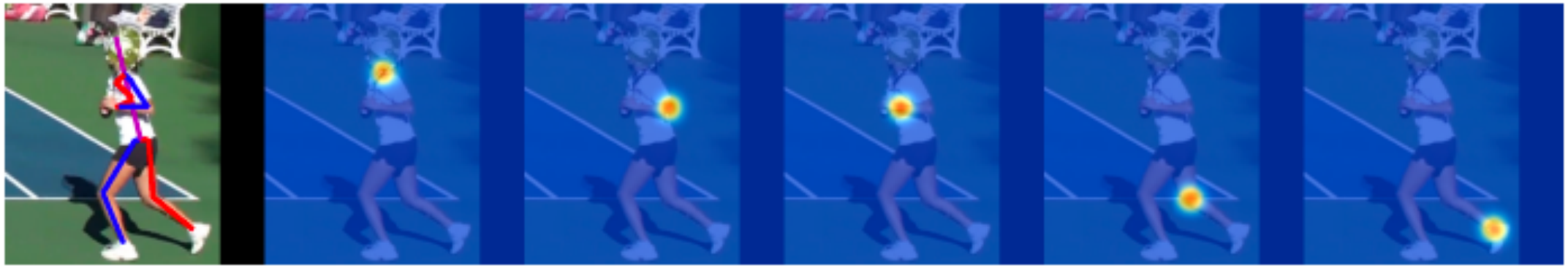
- This paper applied Deep Learning (CNN) to Human Pose Estimation and pretty much kicked off research in this direction.

- **Regressing to XY** locations is **difficult** and adds learning complexity which **weakens generalization** and hence performs poorly in certain regions.

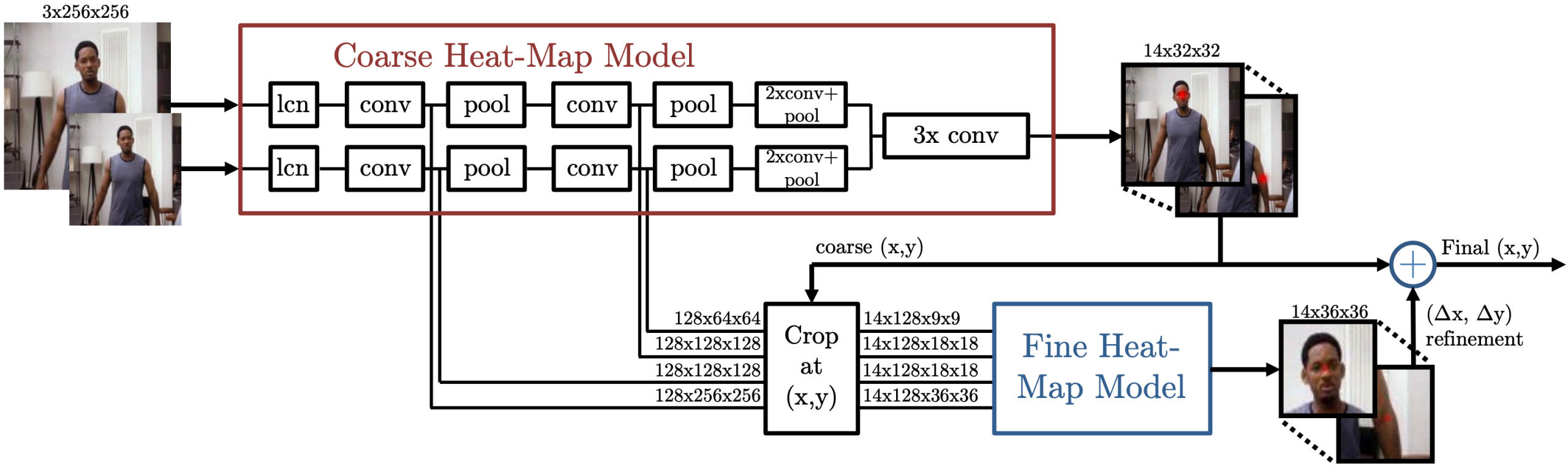
- Recent methods transform the problem to estimating K heatmaps of size W0×H0,{H1,H2,...,Hk}, where each heatmap Hk indicates the location confidence of the kth keypoint. (K keypoints in total).

Efficient Object Localization Using Convolutional Networks (CVPR'15)

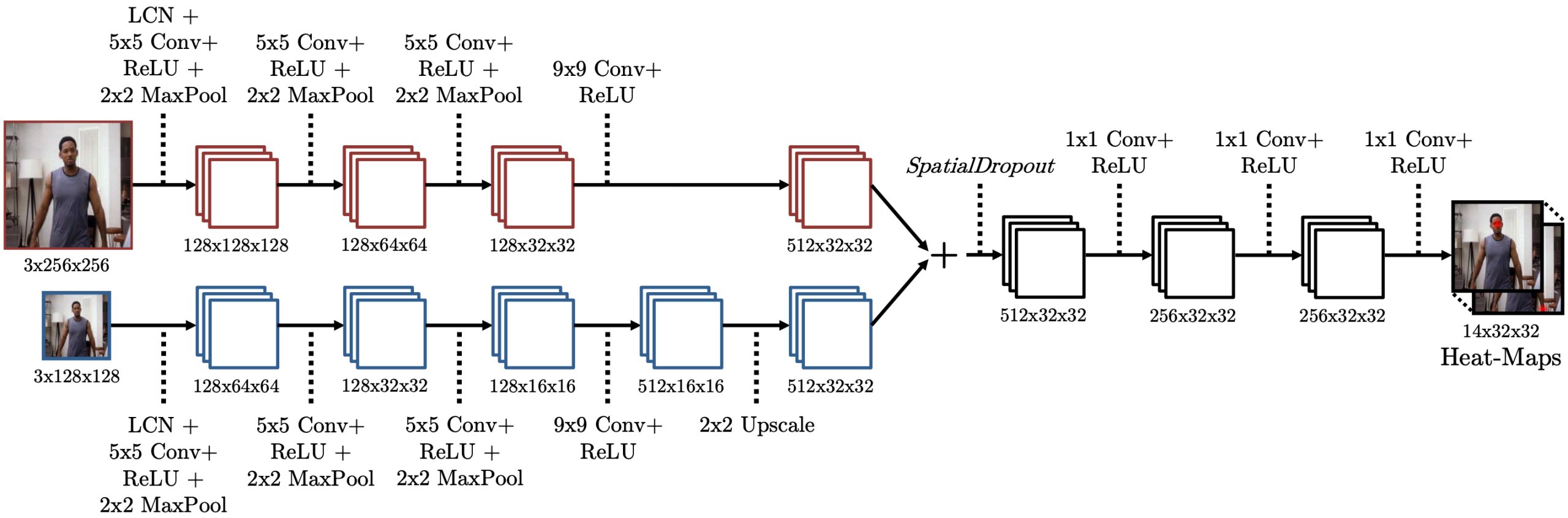
Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, Christoph Bregler
New York University



Overview of the Cascaded Architecture



Coarse Heat-Map Regression



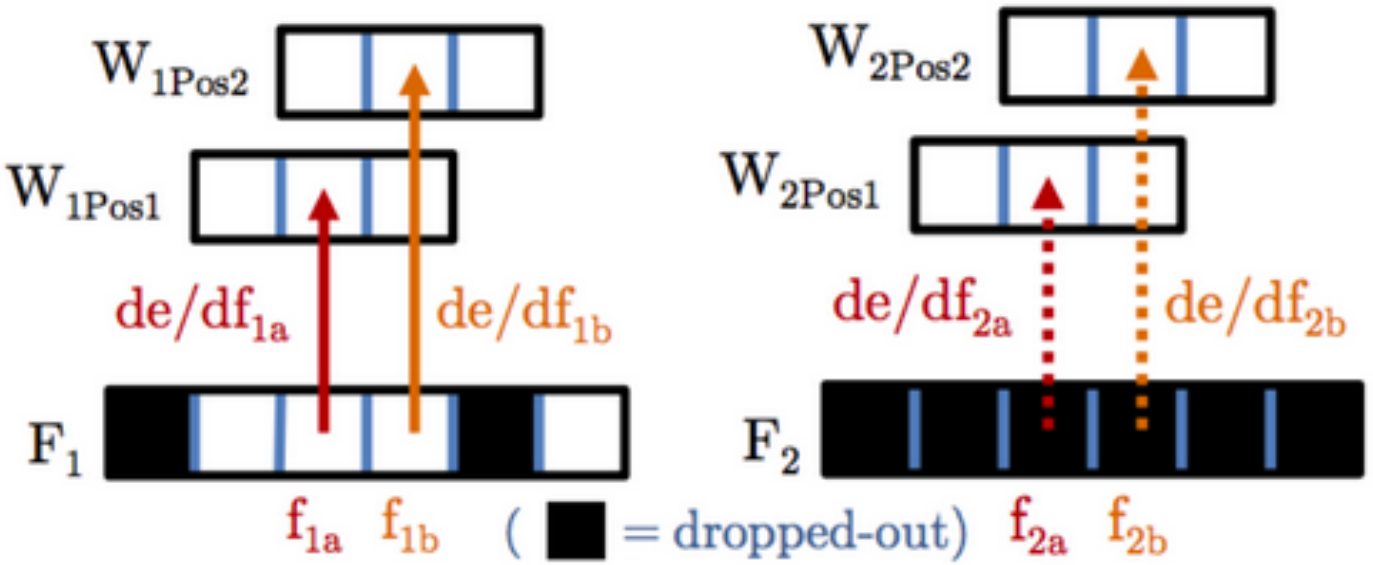
- **Inputs of 3 different scales** are used, and output a heat-map for each joint.
- LCN: Local-Contrast-Normalization
- SpatialDropout

Mean Square Error (MSE)

$$E_1 = \frac{1}{N} \sum_{j=1}^{N} \sum_{xy} \left\| H'_j(x,y) - H_j(x,y) \right\|^2$$

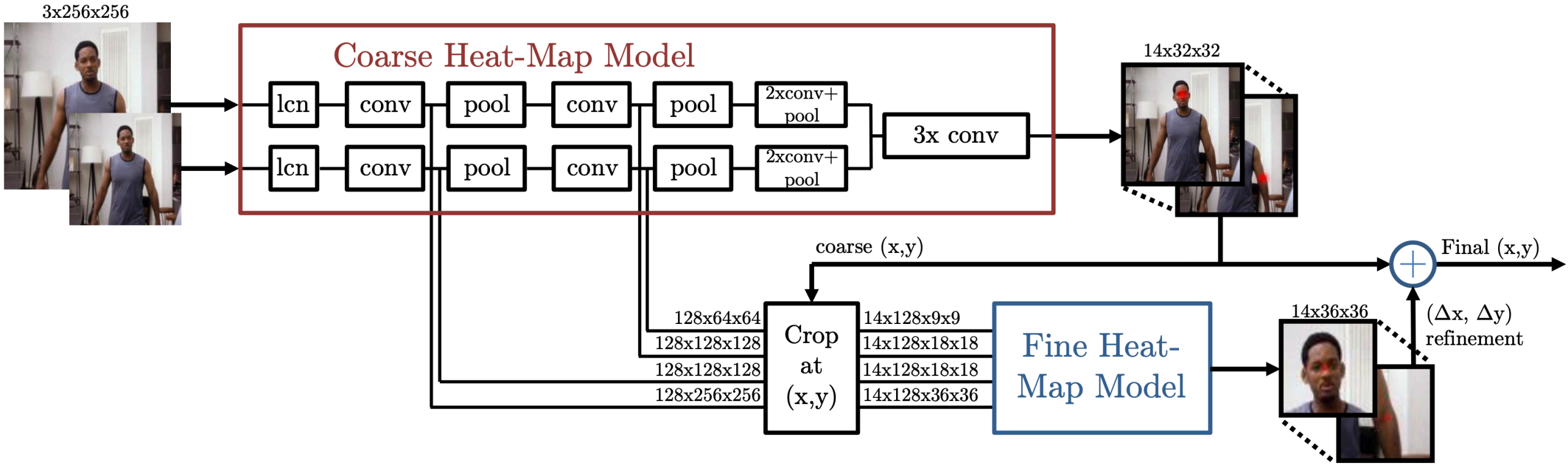
- where *H'j* and *Hj* are the predicted and ground-truth heat-map for *j*-th joint

SpatialDropout



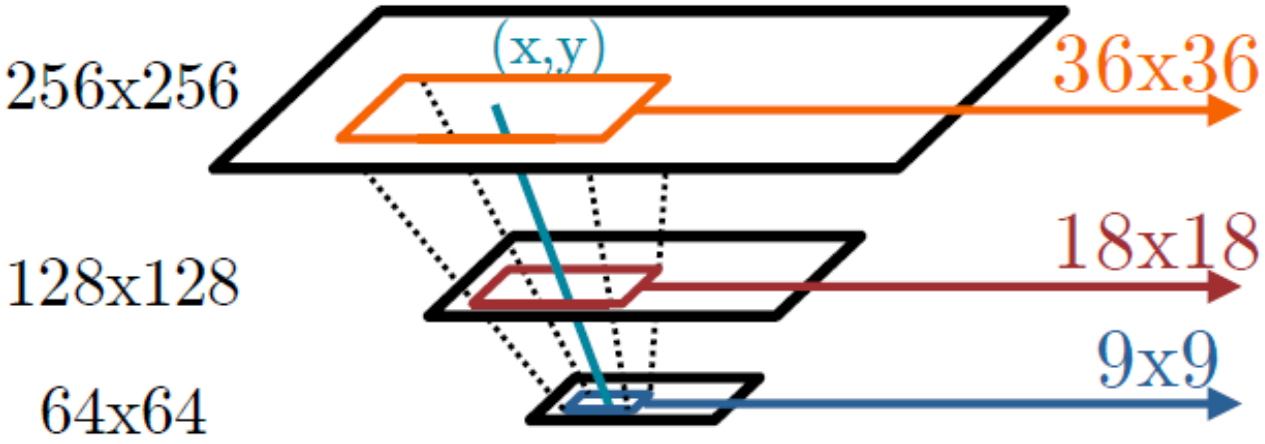
- randomly set entire feature maps (also known as channels) to 0, rather than individual 'pixels.'
- regular dropout would not work so well on images because adjacent pixels are highly correlated.
  - So if you hide pixels randomly I can still have a good idea of what they were by just looking at the adjacent pixels.

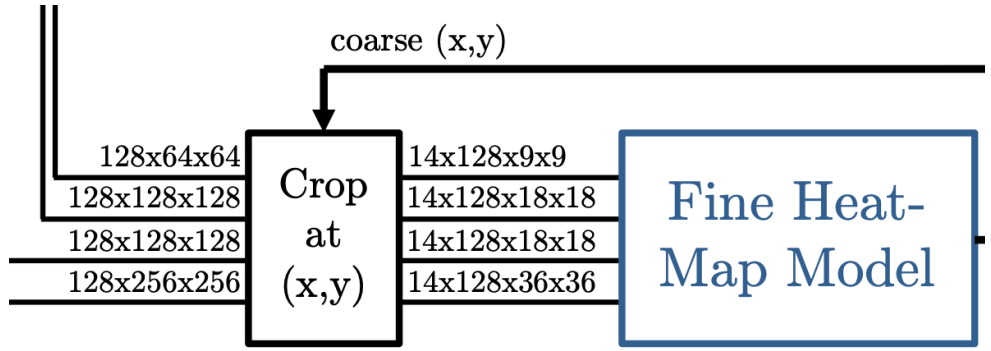
Fine Heat-Map Regression Using Cascaded Architecture



- Goal: **recover the spatial accuracy lost due to pooling** of the coarse heat-map regression model
- unlike DeepPose, we **reuse existing convolution features**
  - reduces the number of trainable parameters
  - acts as a **regularizer** for the coarse heat-map model since the coarse & fine models are **trained jointly**.

Crop Module

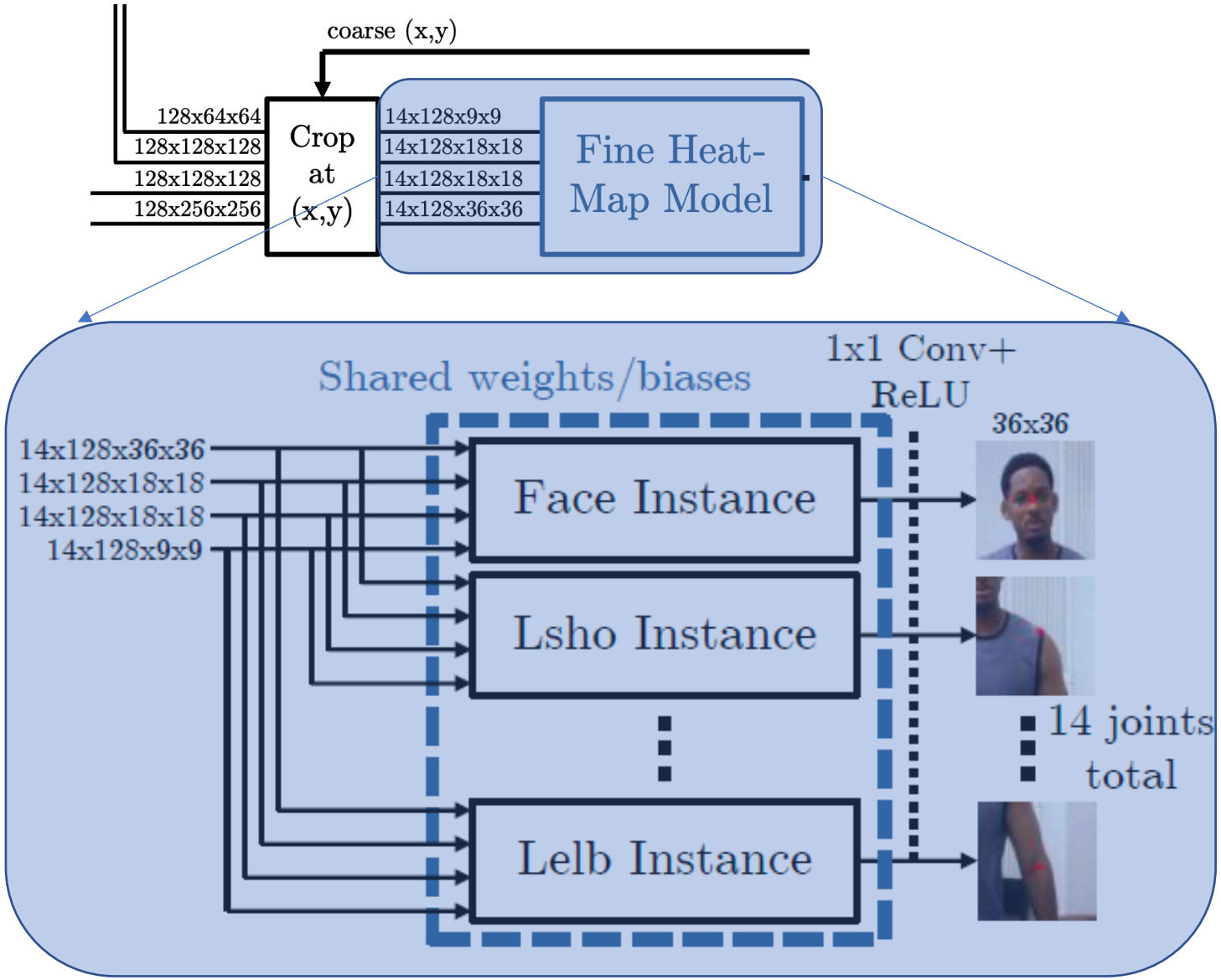




- crop out a window centered at the coarse joint (x, y) location in each resolution feature map

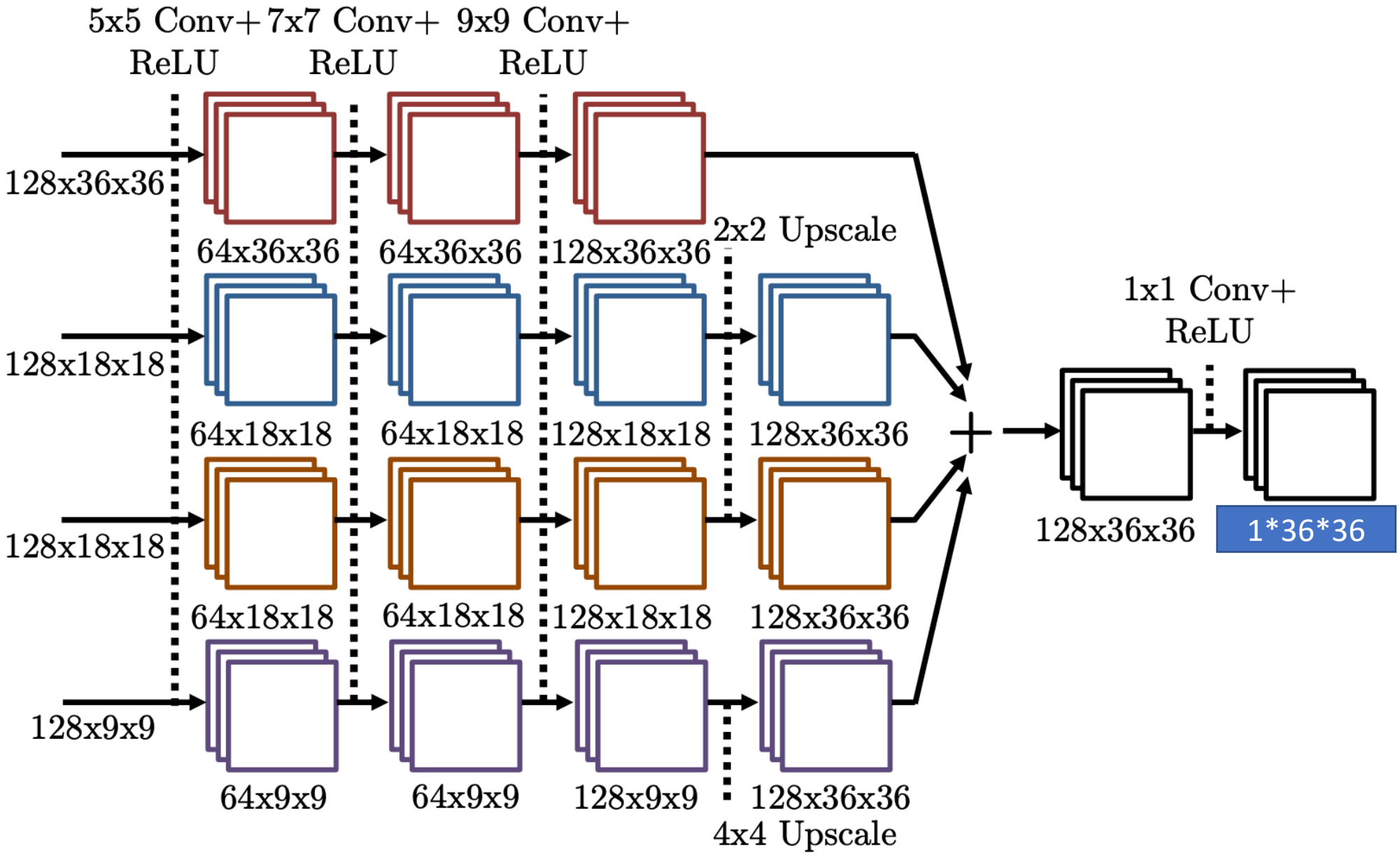
- keeping the contextual size of the window constant by scaling the cropped area at each higher resolution level.

Fine heat-map model: 14 joint Siamese network



- 14 networks shared parameters
- This can reduce the number of parameters and to prevent over-training.
- Finally, a 1×1 convolution, without any weight sharing, is used to output a detailed-resolution heat-map for each joint.

Fine Heat-Map Network for a Single Joint



Joint Training

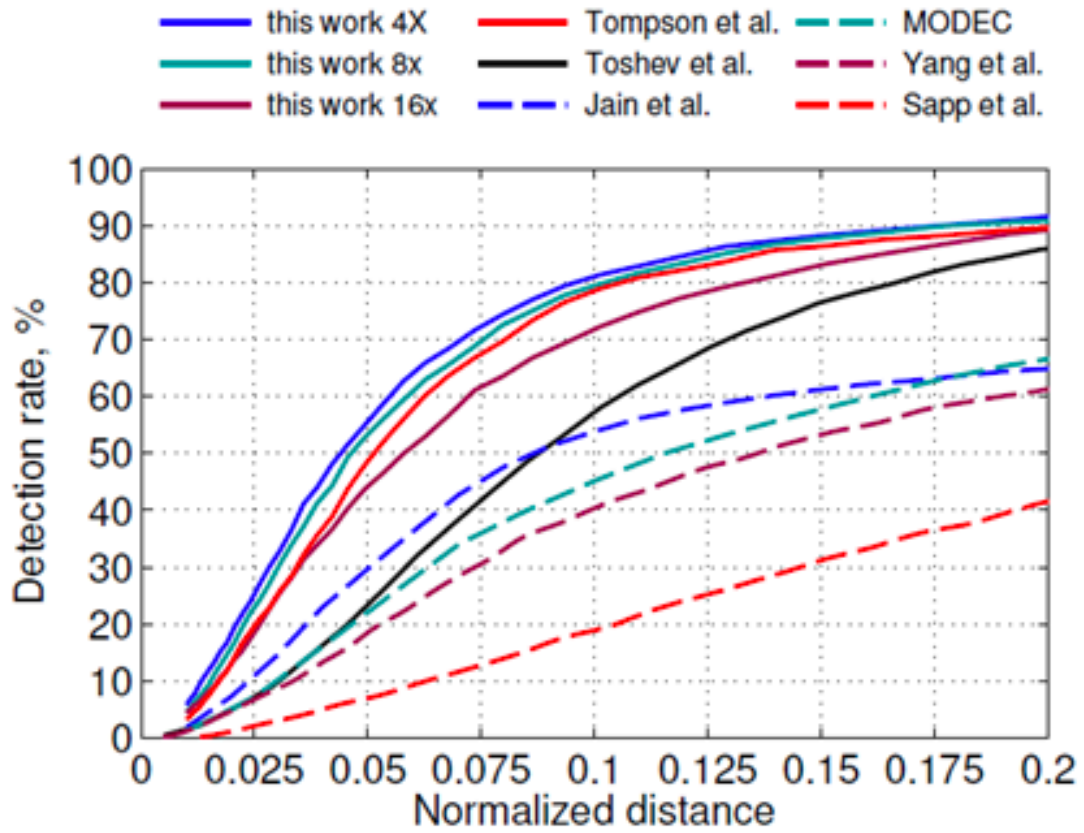
- First, the coarse heat-map model is pre-trained.
- Then the coarse heat-map model is fixed and train the fine heat-map model using below loss function:

$$E_2 = \frac{1}{N}\sum_{j=1}^{N}\sum_{x,y}\left\|G'_j(x,y) - G_j(x,y)\right\|^2$$

  - where *G'j* and *Gj* are the predicted and ground-truth heat-map for *j*-th joint.

- Finally, the coarse and fine models are jointly trained by minimizing E3 = *E1* + *λ*×*E2*, where *λ* = 0.1.

FLIC — FCK Performance, Average (Left) Individual Joints (Right)



	Head	Shoulder	Elbow	Wrist
Yang et al.	-	-	22.6	15.3
Sapp et al.	-	-	6.4	7.9
Eichner et al.	-	-	11.1	5.2
MODEC et al.	-	-	28.0	22.3
Toshev et al.	-	-	25.2	26.4
Jain et al.	-	42.6	24.1	22.3
Tompson et al.	90.7	70.4	50.2	55.4
This work 4x	**92.6**	73.0	**57.1**	**60.4**
This work 8x	92.1	**75.8**	55.6	56.6
This work 16x	91.6	73.0	47.7	45.5

Comments

- Heatmaps work better than direct joint regression

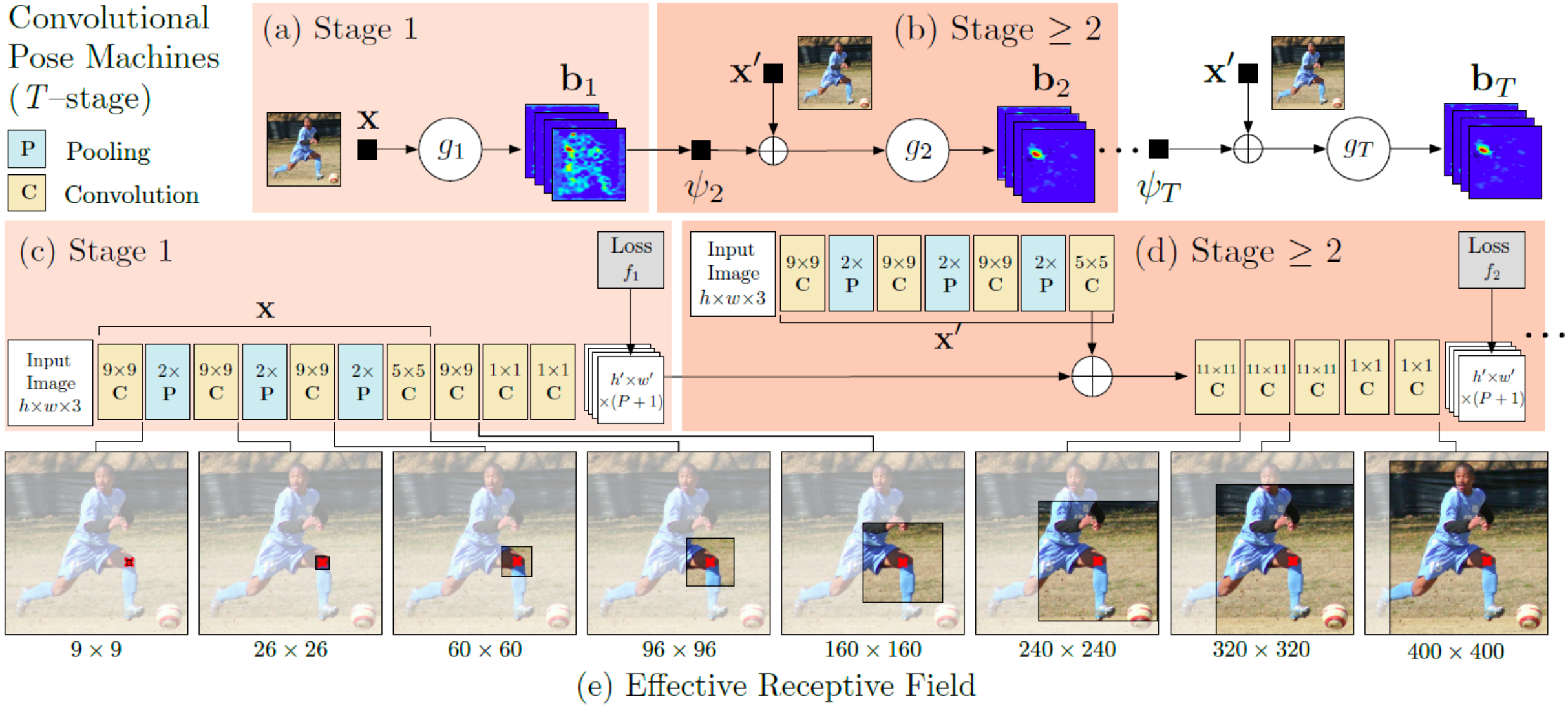
- However, these methods **lack structure modelling**.
  - The space of 2D human poses is highly structured because of body part proportions, left-right symmetries, interpenetration constraints, joint limits (e.g. elbows do not bend back) and physical connectivity (e.g. wrists are rigidly related to elbows), among others.
  - Modelling this structure should make it easier to pinpoint the visible keypoints and make it possible to estimate the occluded ones.
  - The next few papers tackle this, in their own novel ways.

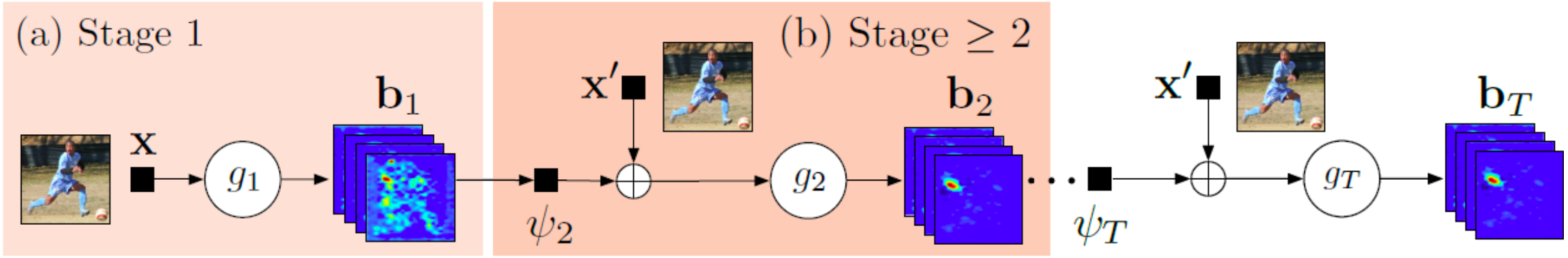
Convolutional Pose Machines (CPM) (CVPR'16, 852 citations)

Shih-En Wei, Varun Ramakrishna, Takeo Kanade, Yaser Sheikh
Carnegie Mellon University

Input Image (a) Stage 1 (b) Stage 2 (c) Stage 3

- a **sequential** architecture that composed of CNN which directly operate on **belief maps** from previous stages, producing increasingly refined estimates for part locations, such as the right elbow.


Convolutional Pose Machines (T–stage)
P Pooling
C Convolution
(a) Stage 1
$\mathbf{x}$
g_1
$\mathbf{b}_1$
(b) Stage ≥ 2
$\mathbf{x}'$
ψ_2
g_2
$\mathbf{b}_2$
ψ_T
g_T
$\mathbf{b}_T$
(c) Stage 1
Loss f_1
Input Image $h \times w \times 3$
9×9 C
2× P
5×5 C
1×1 C
$h' \times w' \times (P+1)$
(d) Stage ≥ 2
Loss f_2
11×11 C
9×9
26×26
60×60
96×96
160×160
240×240
320×320
400×400
(e) Effective Receptive Field

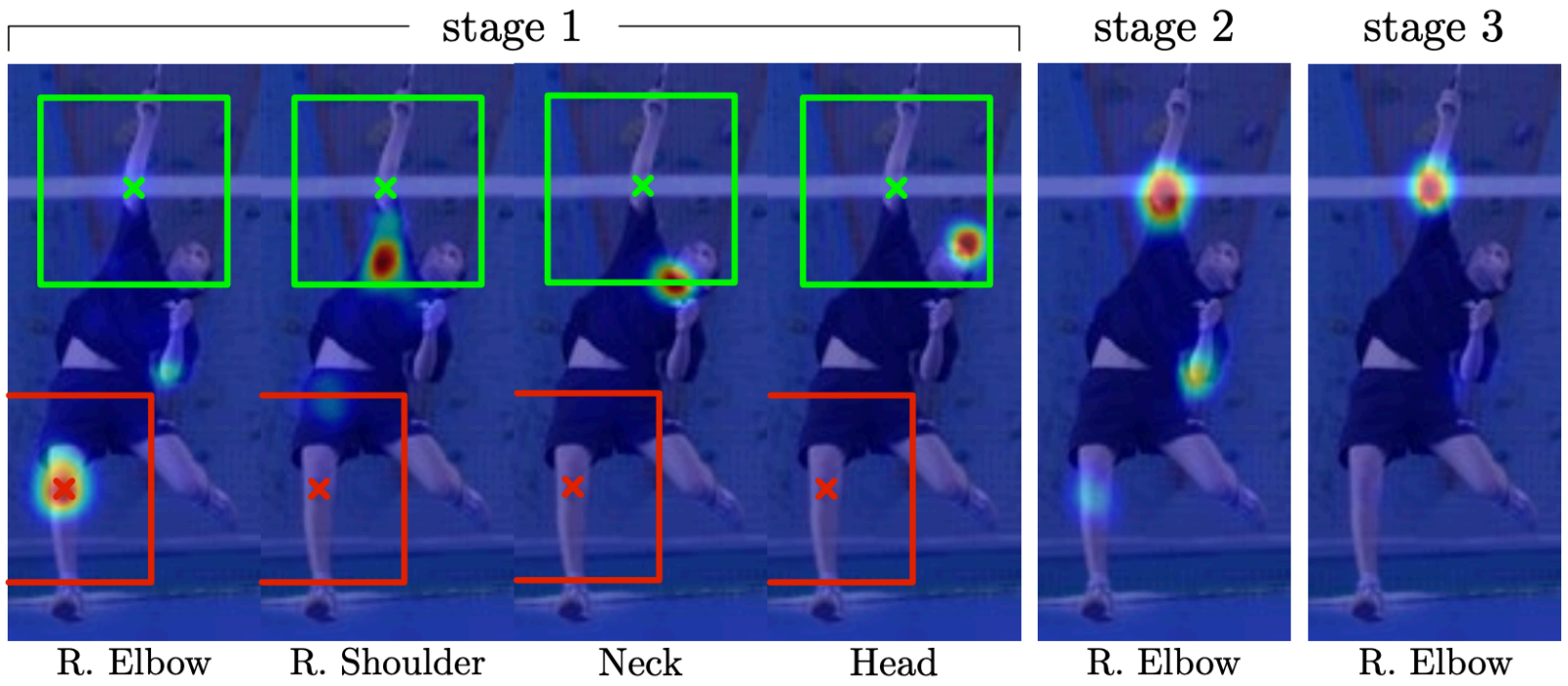


- Input of gt: **features** X or X' + $\psi_{t-1}(b_{t-1})$
- Output of gt: => $\{b_t^p$, p=0, 1, …, P} heatmaps,
- 0…P: P joints plus one for background

- $\psi_{t>1}(\cdot)$ serves to encode beliefs b_{t-1} to better features ideally.
  - Actually, it is an identity function here (I guess).
- we allow features x' for t>1 stages to be different from the feature x used in the first stage.

Spatial context of heatmap

- detection of head and shoulders can be favorable
- Accuracies of elbow may be low
- gt uses heatmaps of all joints together, instead separately, improve predictions by leveraging the fact that parts occur in consistent geometric configurations



Loss Function

$$f_t = \sum_{p=1}^{P+1} \sum_{z \in \mathcal{Z}} \| b_t^p(z) - b_*^p(z) \|_2^2.$$

$$\mathcal{F} = \sum_{t=1}^{T} f_t.$$

- By enforcing **supervision in intermediate stages** through the network, it can somehow **address the vanishing gradient problem** as the intermediate loss functions replenish the gradients at each stage.

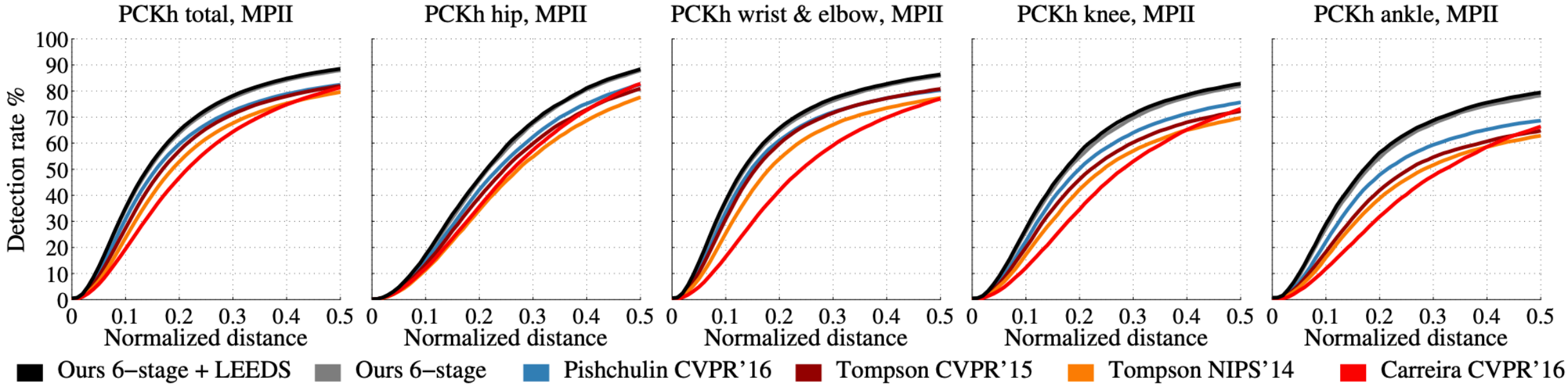


Figure 8: Quantitative results on the MPII dataset using the PCKh metric. We achieve state of the art performance and outperform significantly on difficult parts such as the ankle.

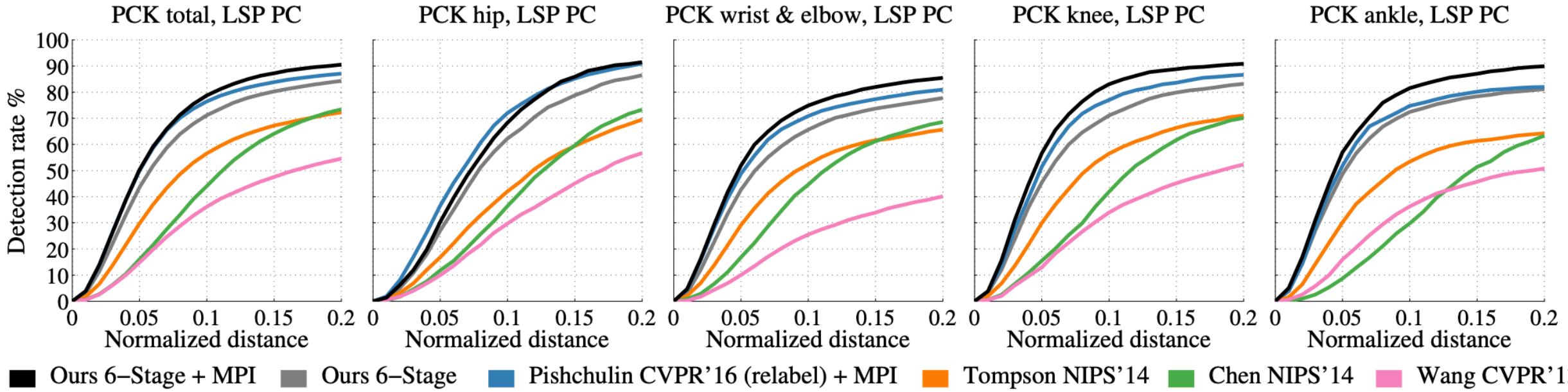


Figure 9: Quantitative results on the LSP dataset using the PCK metric. Our method again achieves state of the art performance and has a significant advantage on challenging parts.

Pose Estimation on MPII Human Pose

Rank	Method	PCKh-0.5	Paper Title	Year	Paper	Code
1	Cascade Feature Aggregation	93.9%	Cascade Feature Aggregation for Human Pose Estimation	2019		
2	Spatial Context	92.5%	Human Pose Estimation with Spatial Contextual Information	2019		
3	HRNet-32	92.3%	Deep High-Resolution Representation Learning for Human Pose Estimation	2019		
4	Multi-Scale Structure-Aware Network	92.1%	Multi-Scale Structure-Aware Network for Human Pose Estimation	2018		
5	Pyramid Residual Modules (PRMs)	92.0%	Learning Feature Pyramids for Human Pose Estimation	2017		
6	Multi-Context Attention	91.5%	Multi-Context Attention for Human Pose Estimation	2017		
7	DU-Net	91.2%	Quantized Densely Connected U-Nets for Efficient Landmark Localization	2018		
8	Stacked hourglass + Inception-resnet	91.2%	Knowledge-Guided Deep Fractal Neural Networks for Human Pose Estimation	2017		
9	Integral Regression	91.0%	Integral Human Pose Regression	2017		
10	Stacked Hourglass Networks	90.9%	Stacked Hourglass Networks for Human Pose Estimation	2016		
11	CU-Net	89.4%	CU-Net: Coupled U-Nets	2018		
12	Convolutional Pose Machines	88.52%	Convolutional Pose Machines	2016		

Stacked Hourglass Networks for Human Pose Estimation (ECCV'16)

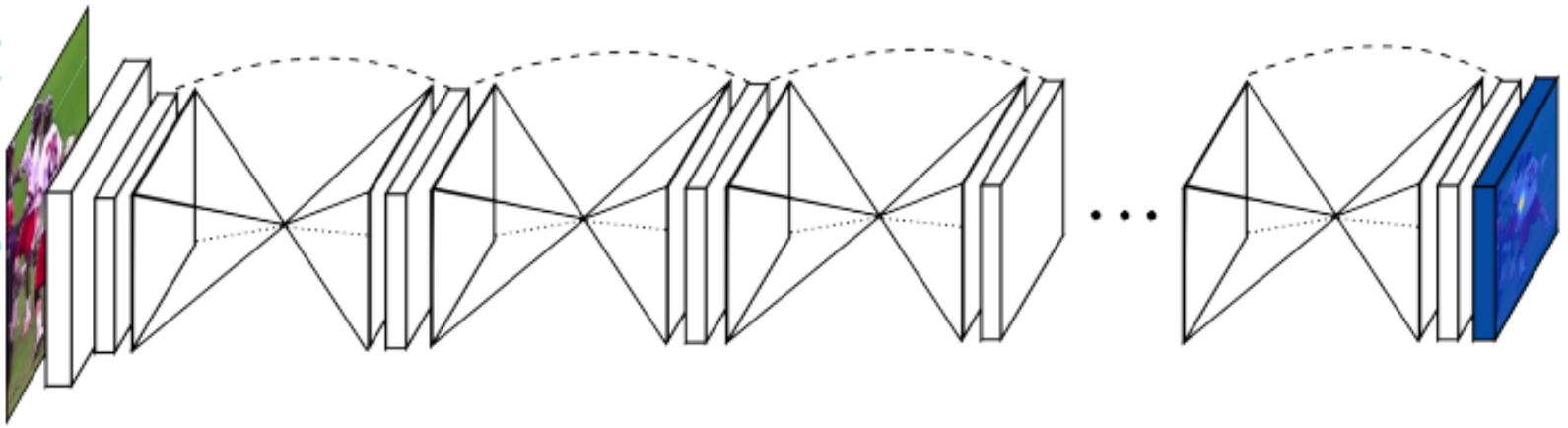


Fig. 1. Our network for pose estimation consists of multiple stacked hourglass modules which allow for repeated bottom-up, top-down inference.

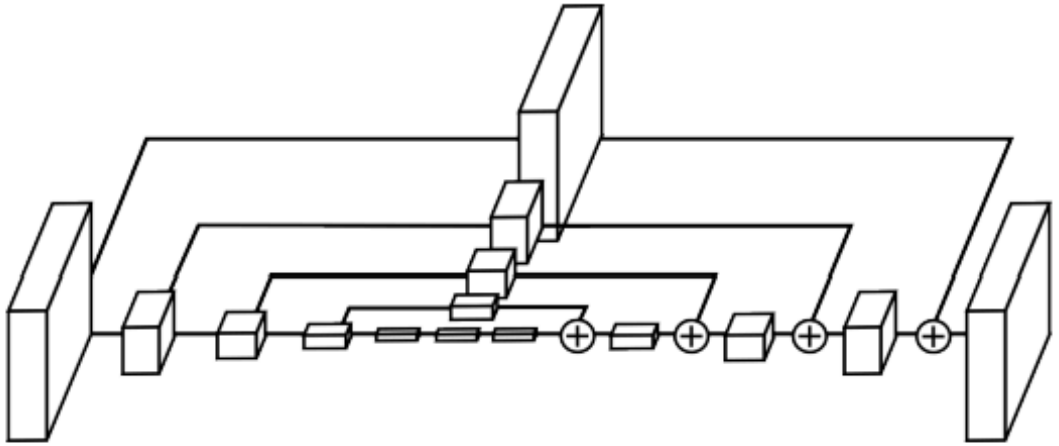


Fig. 3. An illustration of a single "hourglass" module. Each box in the figure corresponds to a residual module as seen in Figure 4. The number of features is consistent across the whole hourglass.

Cascaded pyramid network for multi-person pose estimation. CVPR (2018)

Integral Human Pose Regression, 2nd of eccv18 PoseTrack Challenge

Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, Yichen Wei

Rank	Method	PCKh-0.5	Paper Title	Year	Paper	Code
1	**Cascade Feature Aggregation**	93.9%	Cascade Feature Aggregation for Human Pose Estimation	2019		
2	**Spatial Context**	92.5%	Human Pose Estimation with Spatial Contextual Information	2019		
3	**HRNet-32**	92.3%	Deep High-Resolution Representation Learning for Human Pose Estimation	2019		
4	**Multi-Scale Structure-Aware Network**	92.1%	Multi-Scale Structure-Aware Network for Human Pose Estimation	2018		
5	**Pyramid Residual Modules (PRMs)**	92.0%	Learning Feature Pyramids for Human Pose Estimation	2017		
6	**Multi-Context Attention**	91.5%	Multi-Context Attention for Human Pose Estimation	2017		
7	**DU-Net**	91.2%	Quantized Densely Connected U-Nets for Efficient Landmark Localization	2018		
8	**Stacked hourglass + Inception-resnet**	91.2%	Knowledge-Guided Deep Fractal Neural Networks for Human Pose Estimation	2017		
9	**Integral Regression**	91.0%	Integral Human Pose Regression	2017		
10	**Stacked Hourglass Networks**	90.9%	Stacked Hourglass Networks for Human Pose Estimation	2016		
11	**CU-Net**	89.4%	CU-Net: Coupled U-Nets	2018		
12	**Convolutional Pose Machines**	88.52%	Convolutional Pose Machines	2016		

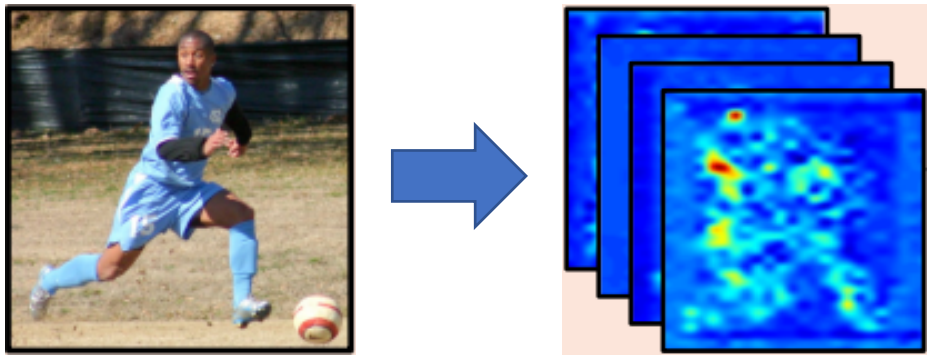
Detection VS. Regression

Detection

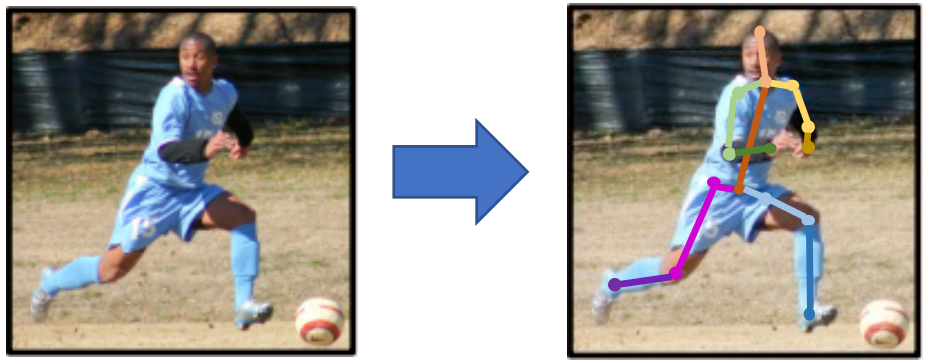
- Per-pixel classification
- Output: likelihood score maps

Regression

- Location regression
- Output: key points location



$\mathbf{H}_k$: Heatmap

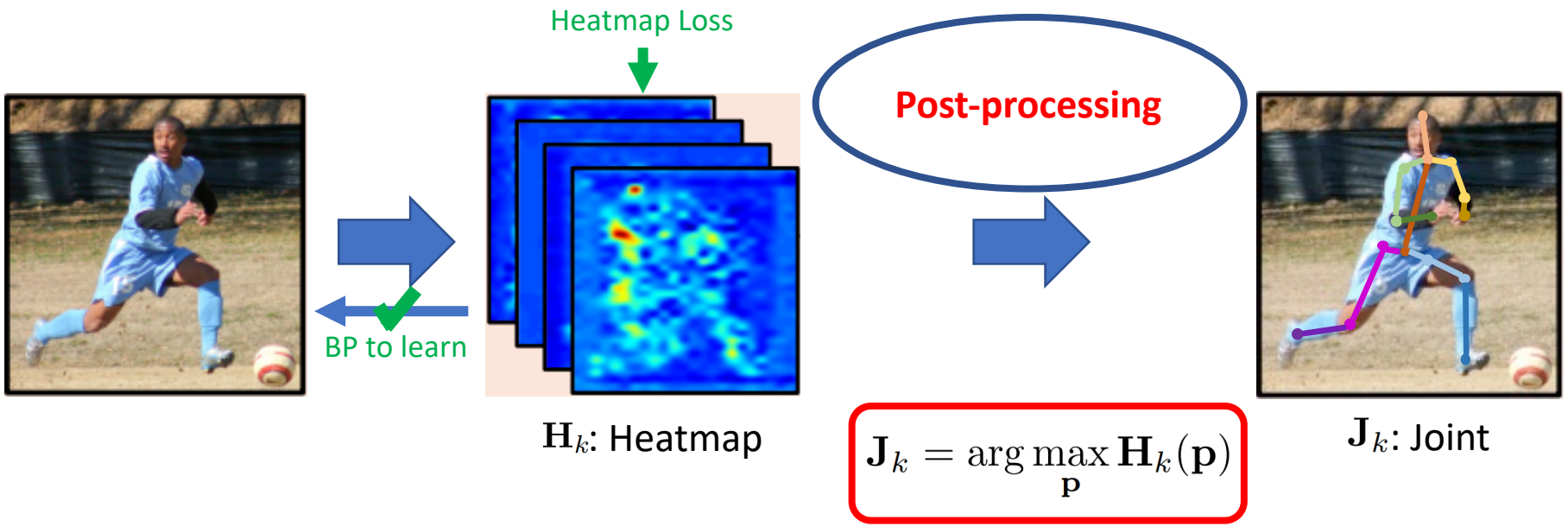


$\mathbf{J}_k$: Joint

Detection: Post-processing

Detection

- Per-pixel classification
- Output: likelihood score maps



Detection: Better performance

Detection

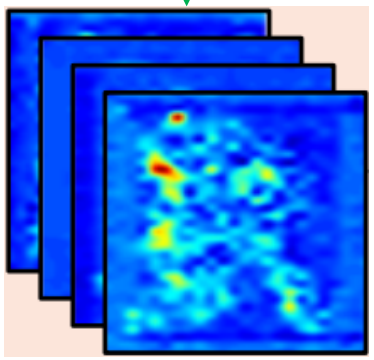
- Per-pixel classification
- Output: likelihood score maps

Better performance

Divide and Conquer: It divides the joint localization task into local image classification tasks. The latter is easier to train, because it effectively reduces the feature and target dimensions for the gradient based learning system.

BP to learn

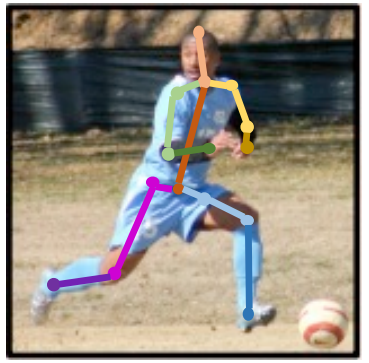
Heatmap Loss



$\mathbf{H}_k$: Heatmap

Post-processing

$$\mathbf{J}_k = \arg\max_{\mathbf{p}} \mathbf{H}_k(\mathbf{p})$$



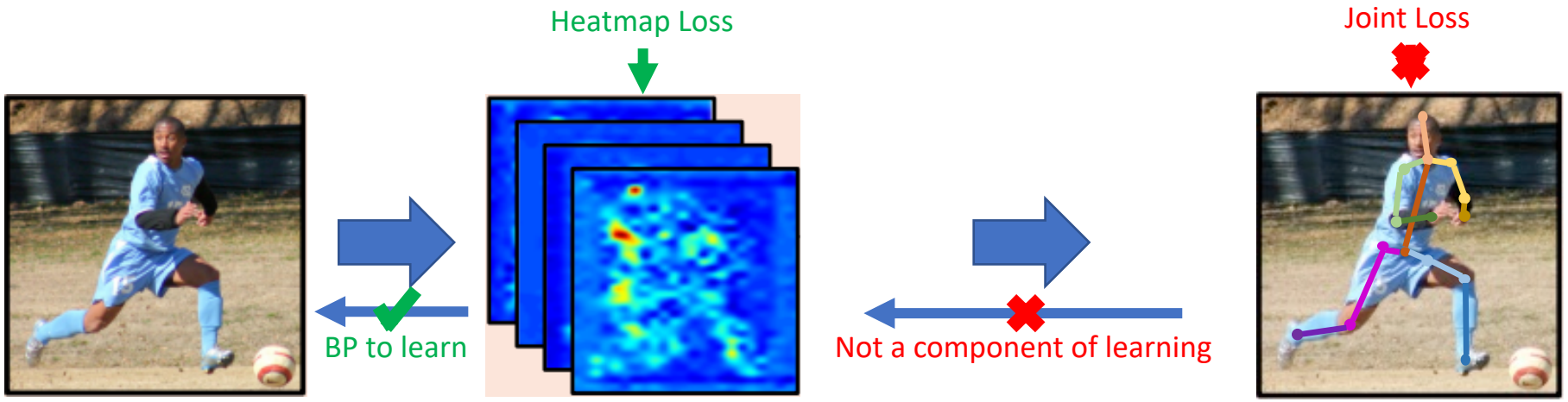
$\mathbf{J}_k$: Joint

Detection: Drawbacks

Detection

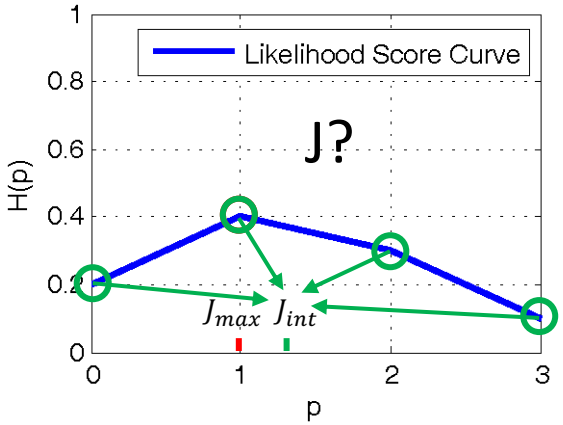
- Per-pixel classification
- Output: likelihood score maps

- **Not-differentiable**
- **Quantization error**
- **Ambiguity**



Taking Maximum VS. Taking Expectation

Example: Given the likelihood curve H(p), where is the most probable joint location J?



Argmax

$$J_{max} = \underset{p}{argmax}\, H(p)$$

= 1

- Not-differentiable
- Quantization Error

Integration

$$J_{int} = \sum_{p} p * H(p)$$

= 0 * 0.2 + 1 * 0.4 + 2 * 0.3 + 3 * 0.1

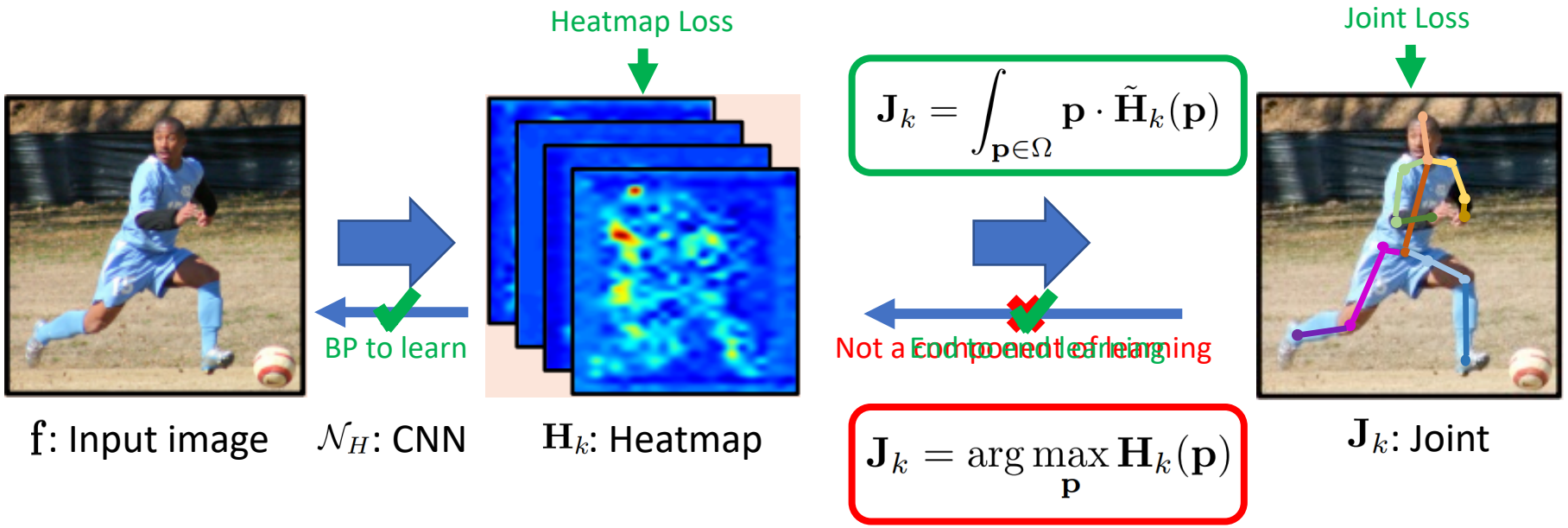
= 1.3

- Differentiable
- Continuous Output

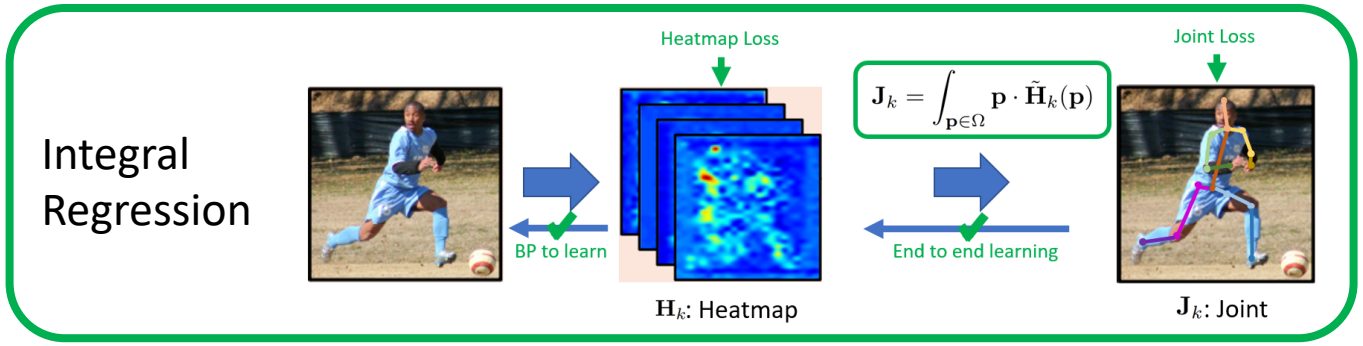
Integral Regression: Taking Expectation

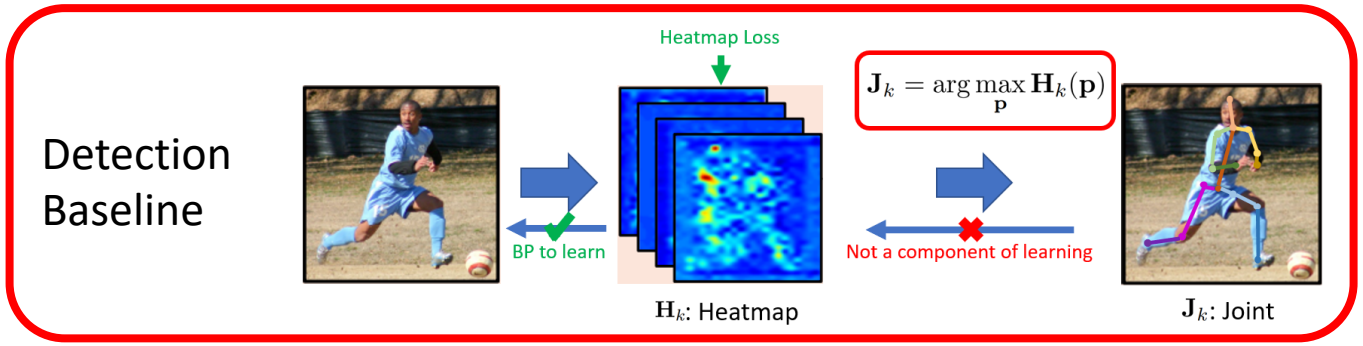
- **Not-differentiable**
- **Quantization error**
- Ambiguity

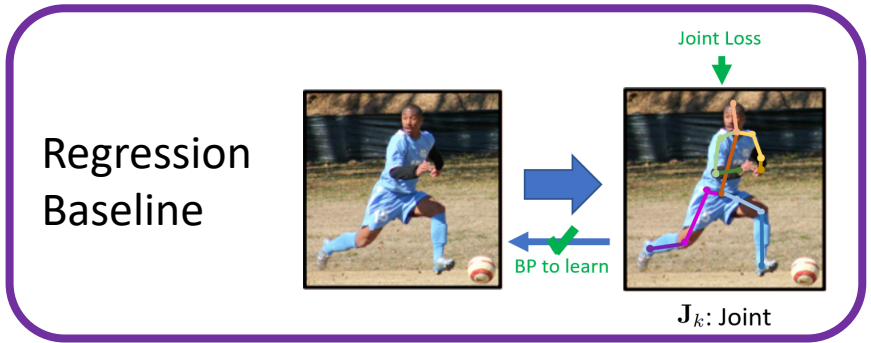
- Differentiable
- Continuous Output
- Single Mode



Share the Merits of Both







It shares the merits of both heat map representation and joint regression approaches.

1. Divide and Conquer (Easy to train)
2. End-to-end learning
3. Continuous output
4. Simple, fast, no extra parameters
5. Compatible with any heat map based methods
6. **Effective (Greatly improve the accuracy)**

Example Visualization

Ground Truth

Regression Baseline

Detection Baseline

Integral Regression

Ablation Study: Network Architecture

	Network Architecture (Multi-stage HourGlass [2])	Coarse-to-Fine. [3] (mm)	Ours H1 (mm)	Ours I1 (mm)
Two-stage HourGlass	Stage 1	85.8	85.5	78.7 (8.0%)
	Stage 2	69.8	68.0	64.1 (5.7%)

- Multi-stage HourGlass architecture sets heatmap based state-of-the-art.
- Our re-implementation is already slightly better, setting a valid baseline.
- Integral Regression improves both stages and sets new state-of-the-art.

[2] Newell et al., Stacked Hourglass Networks for Human Pose Estimation, ECCV 2016.
[3] Georgios et al., Coarse-to-fine volumetric prediction for single-image 3d human pose, CVPR2017.

Conclusions

- Integral regression enables end-to-end training for detection-based approach.
- It allows for continuous location estimates rather than coarse quantization.
- It leads to significant improvement over the state of the art.

OpenPose – CMU, bottom-up approach

- **2D real-time multi-person keypoint detection**:
  - 15 or 18 or **25-keypoint body/foot keypoint estimation**. **Running time invariant to number of detected people**.
  - **2x21-keypoint hand keypoint estimation**. Currently, **running time depends** on **number of detected people**.
  - **70-keypoint face keypoint estimation**. Currently, **running time depends** on **number of detected people**.
- **3D real-time single-person keypoint detection**:
  - 3-D triangulation from multiple single views.
  - Synchronization of Flir cameras handled.
  - Compatible with Flir/Point Grey cameras, but provided C++ demos to add your custom input.
- **Calibration toolbox**:
  - Easy estimation of distortion, intrinsic, and extrinsic camera parameters.

CVPR2017, Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields
CVPR2017, Hand Keypoint Detection in Single Images using Multiview Bootstrapping
CVPR2016, Convolutional pose machines

OpenPose

- **detects parts (keypoints) of an image**
- **assigning parts to distinct individuals**

Deepcut

(a) (b) (c)

Figure 1. Method overview: (a) initial detections (= part candidates) and pairwise terms (graph) between all detections that (b) are jointly clustered belonging to one person (one colored subgraph = one person) and each part is labeled corresponding to its part class (different colors and symbols correspond to different body parts); (c) shows the predicted pose sticks.

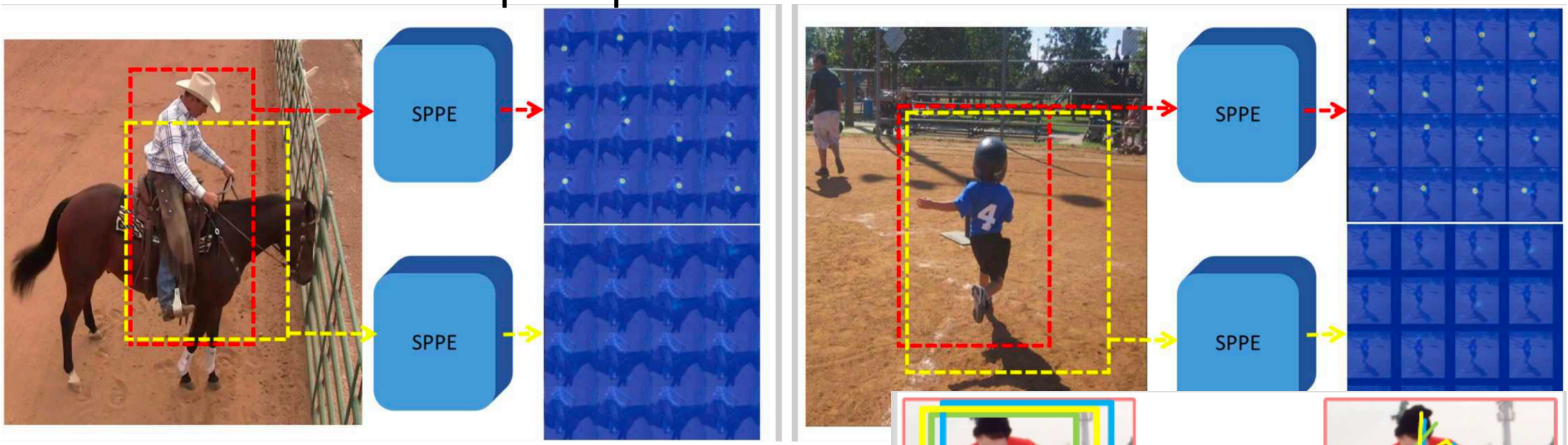
AlphaPose

RMPE: Regional Multi-Person Pose Estimation, iccv 2017

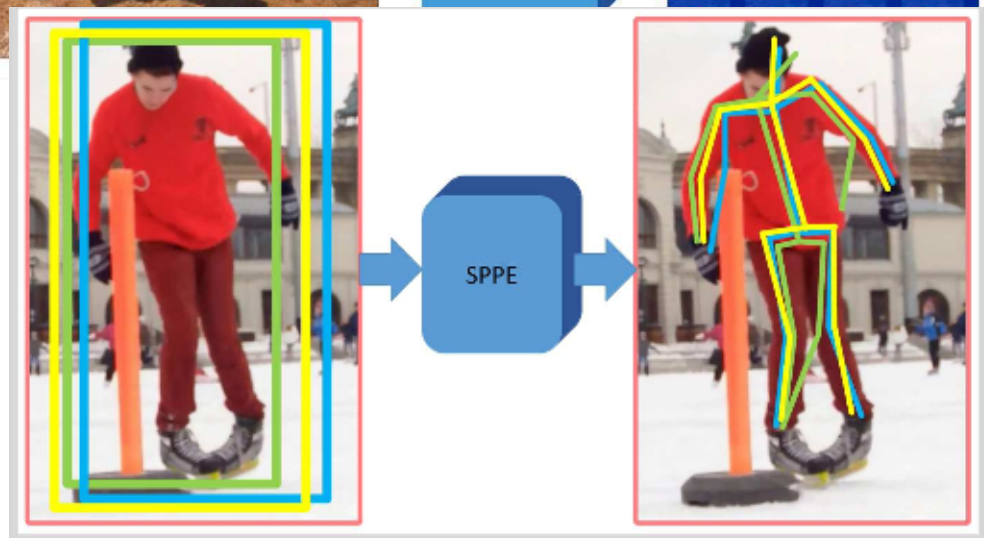
Haoshu Fang, Shuqin Xie, Yuwing Tai, Cewu Lu
Shanghai Jiao Tong University, Tencent Inc.

2 issues of top-down methods: Faster RCNN+ Single Person Pose Estimator (SPPE)

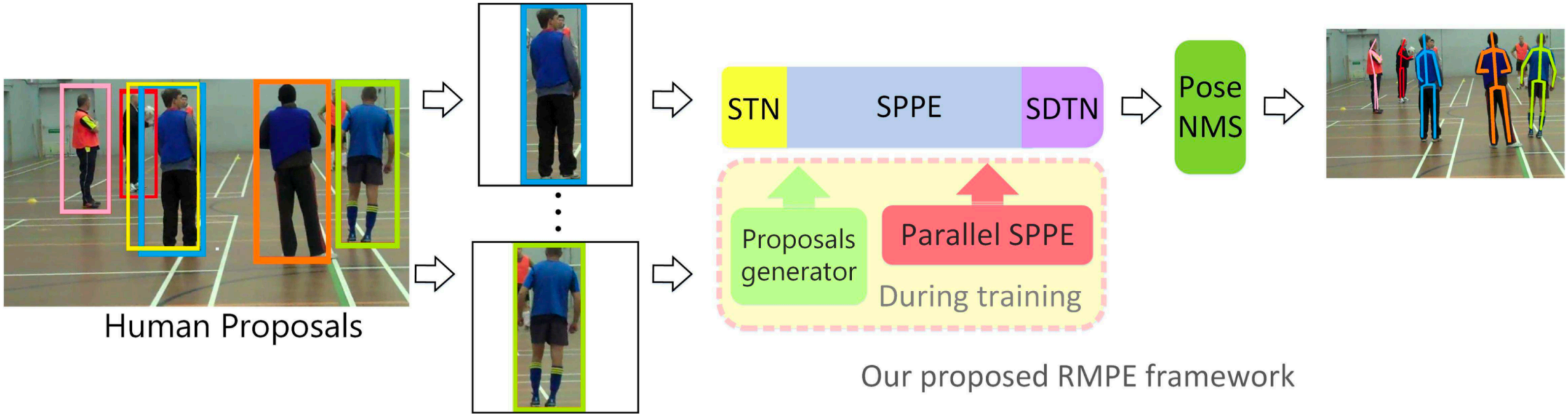
1. inaccurate bbox => poor pose



2. duplicate bbox => redundant poses

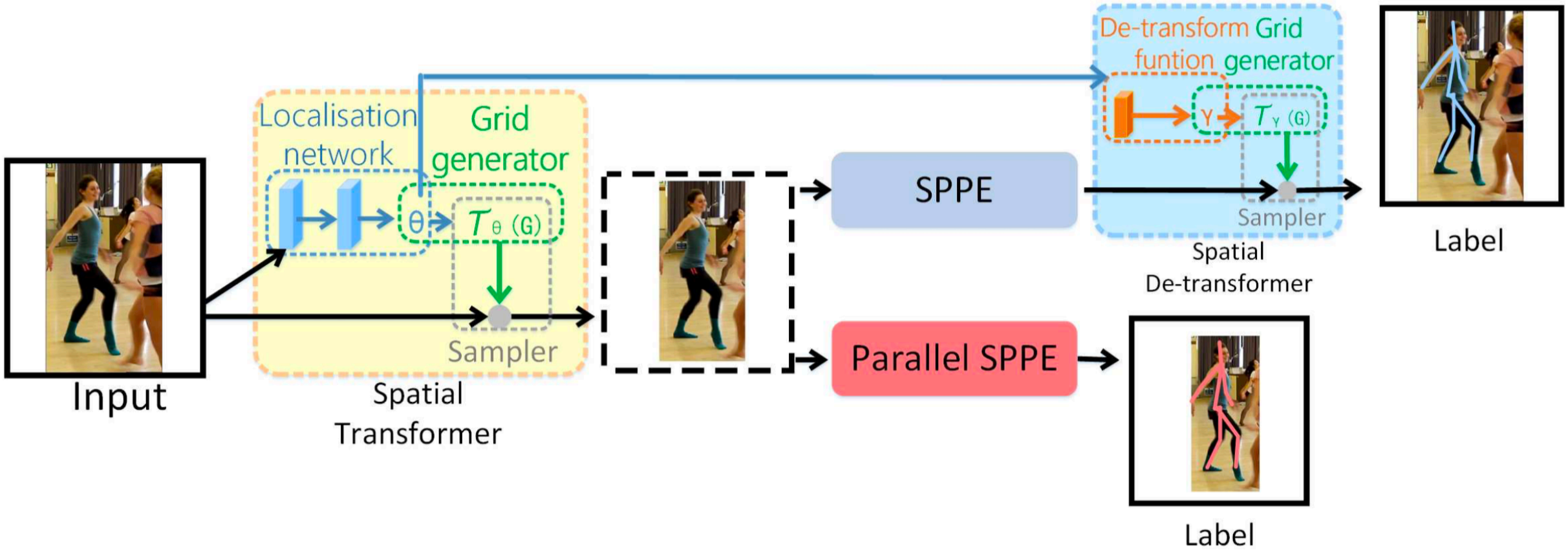


Pipeline



- Symmetric Spatial Transformer Network (SSTN) => bounding box
- Pose NMS => remove redundant poses

STN-SPPE-SDTN



- Symmetric Spatial Transformer Network (SSTN) => high-quality single person region from an inaccurate bounding box
- A Spatial De-Transformer Network (SDTN) is used to remap the estimated human pose back to the original image coordinate system.

Pose Guided Proposals Generator

- to augment training samples that can better help train the SPPE and SSTN networks.

Parametric Pose NMS

- duplicate bbox => redundant poses

效果　MPII datasets mAP提高17%

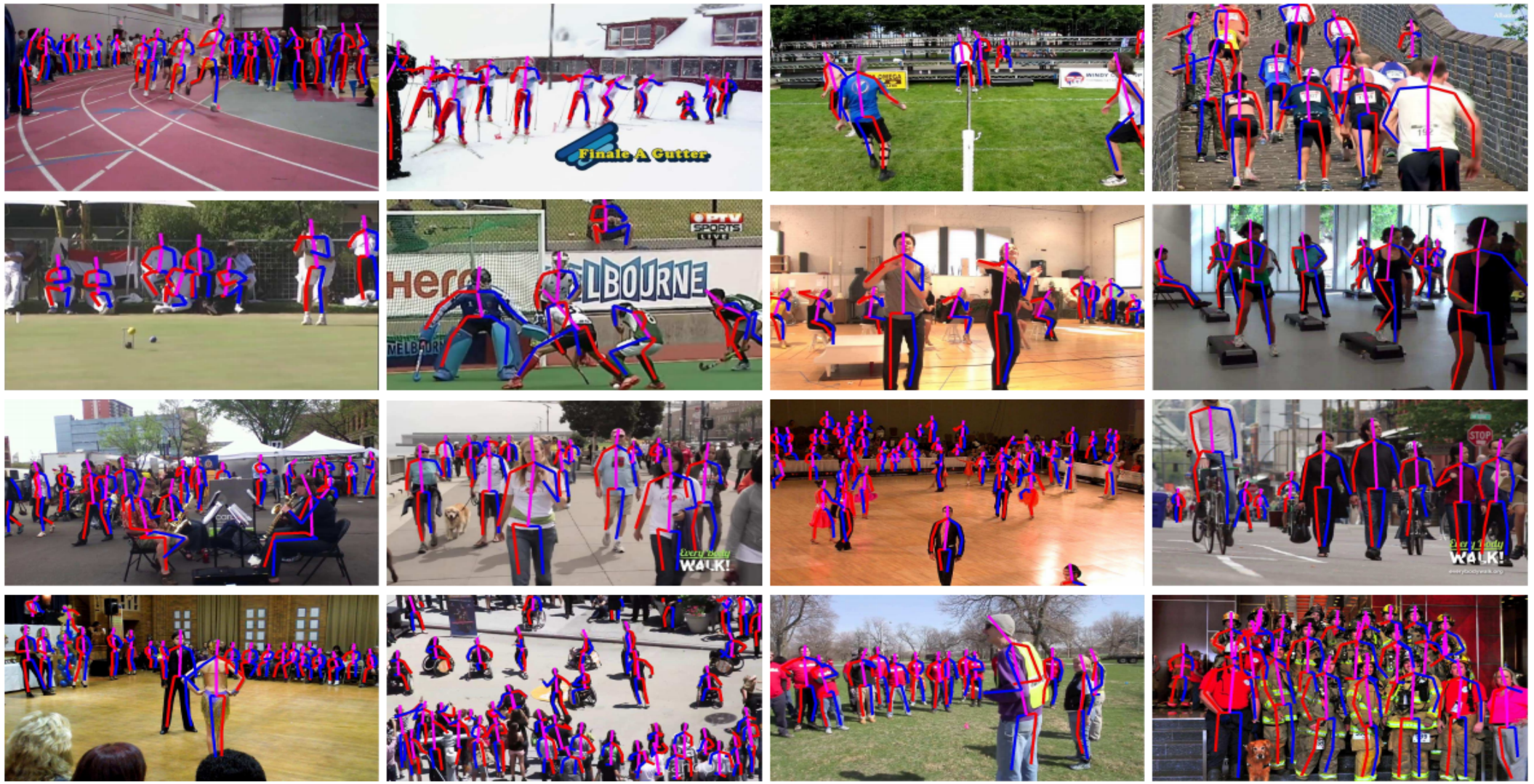


Figure 6. Some results of our model's predictions.

一些失败的例子

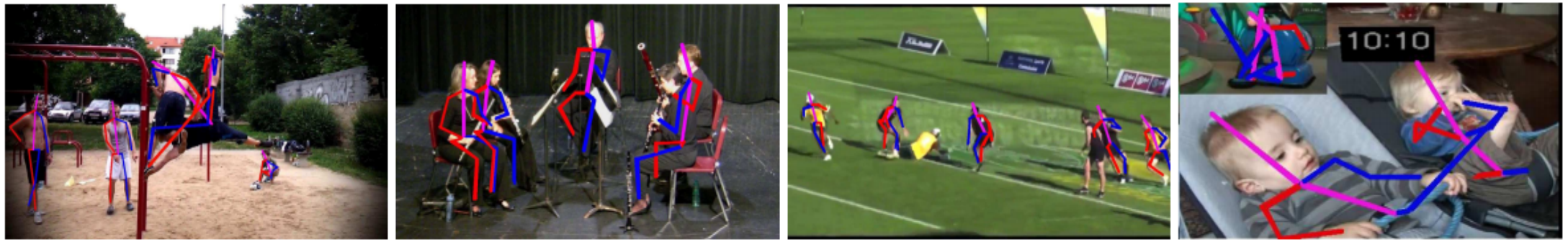


Figure 7. Example failure cases of our model.

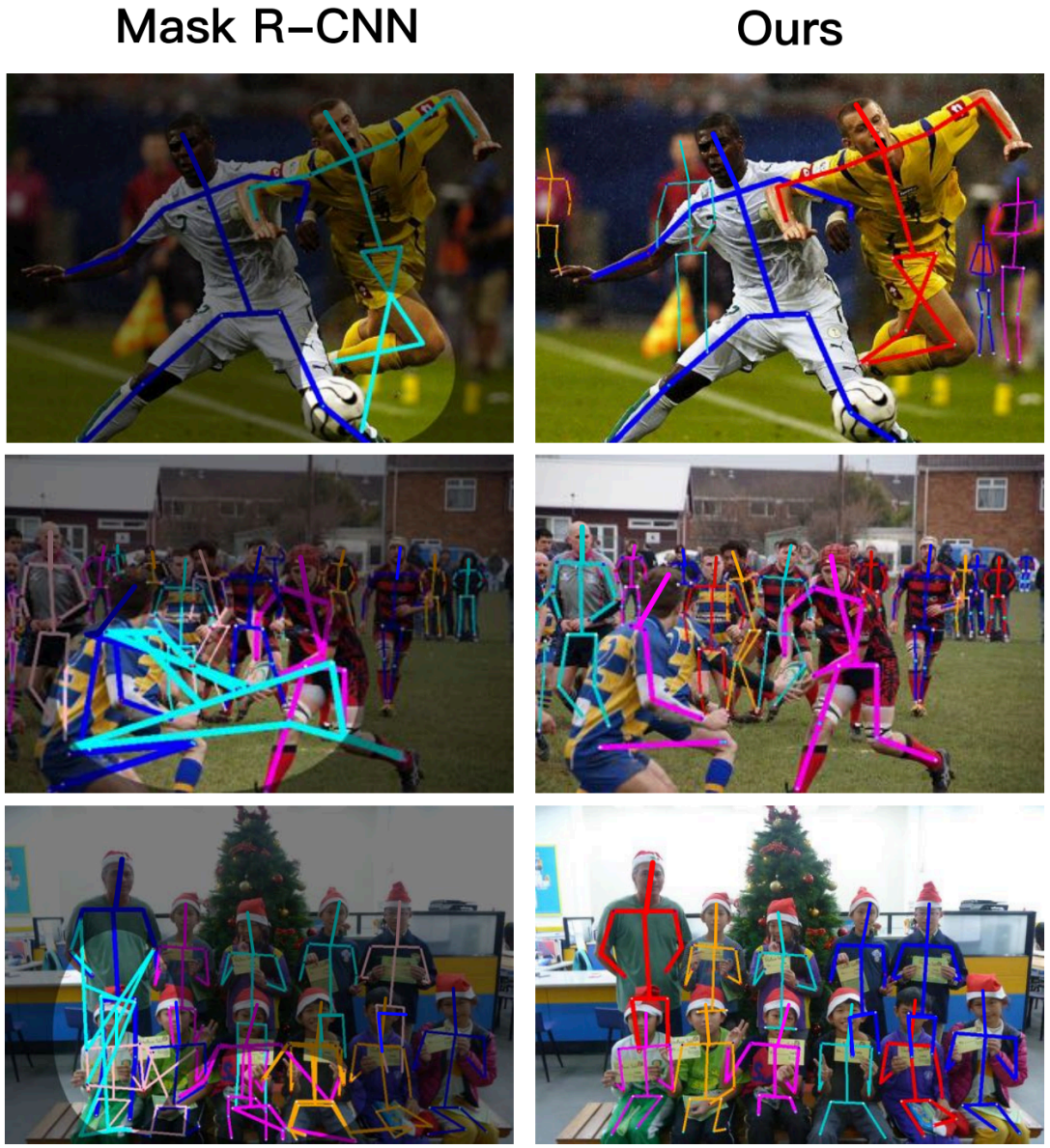
存在的问题：

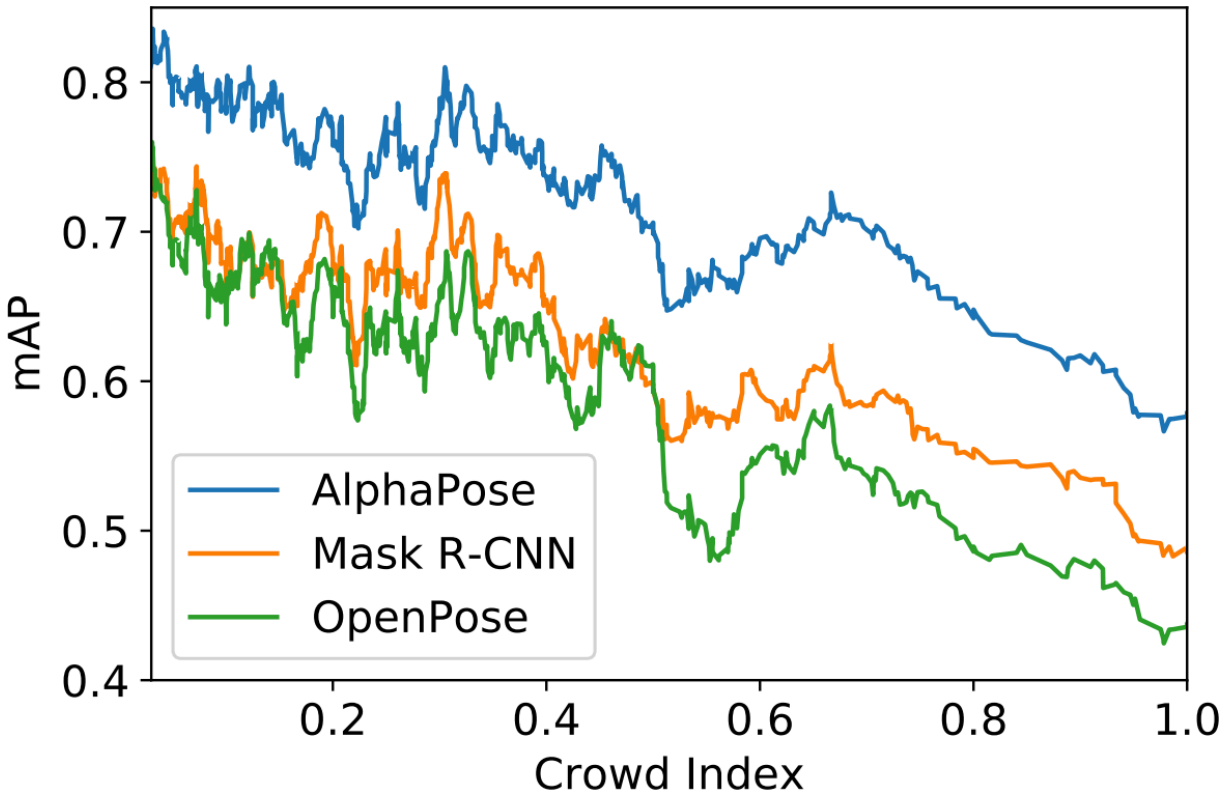
SPPE不能处理不通常出现的姿势
当发生高度遮挡时会产生混淆，无法区分
漏检时无法pose估计
当背景物体的形状很像人姿势时会被误检

CrowdPose

Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, Cewu Lu1
Shanghai Jiao Tong University, Tsinghua University

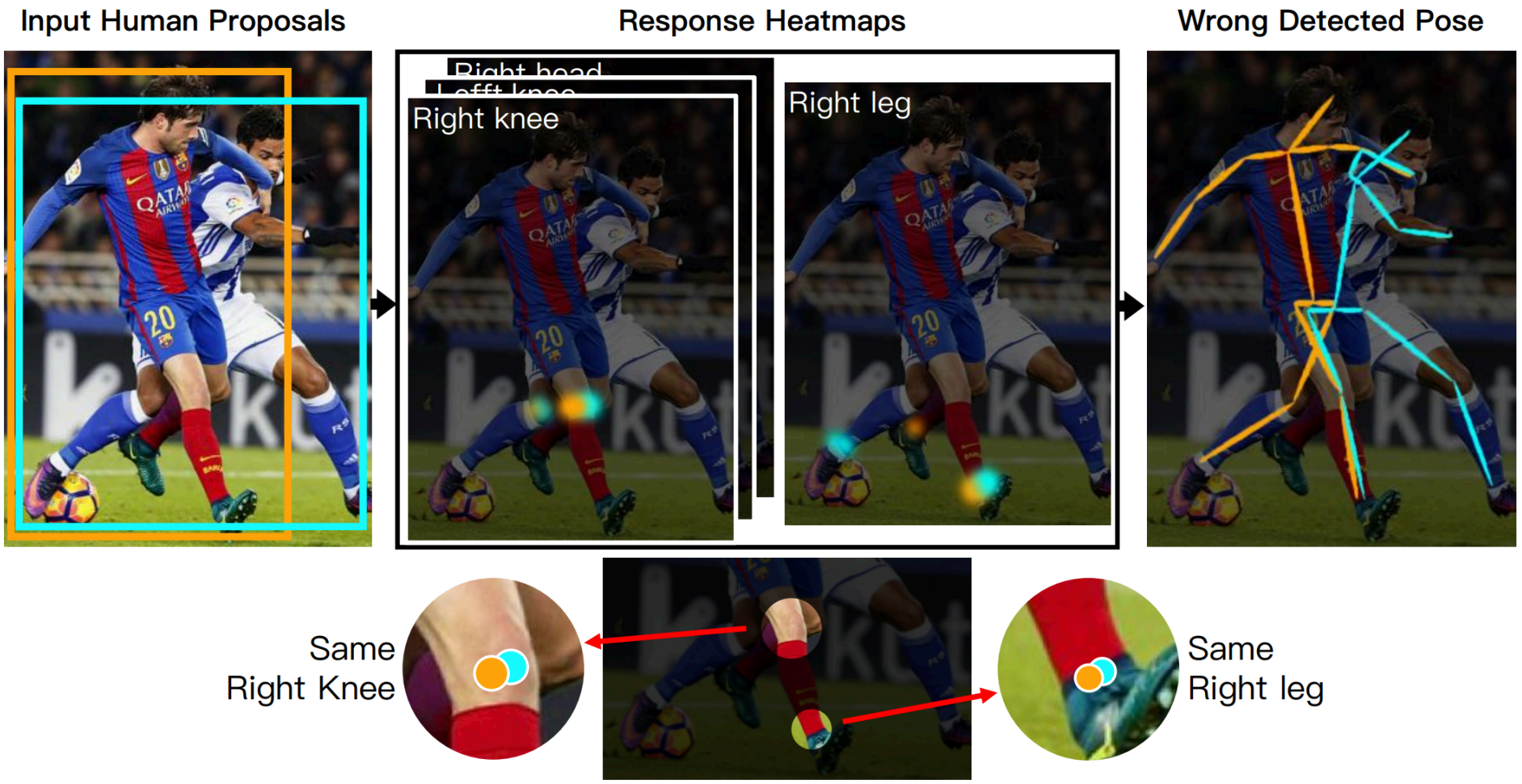
CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark, cvpr19





當前的methods隨擁擠程度上升，效能下降很快

How to solve this?



- Frame sequence
- Single frame

Joint-candidate SPPE – a single frame solution

- SPPE: region Ri (for the ith human proposal) => heatmap Pi .
- two types of candidate joints in Ri
  - **target** joints: joints of the ith person
  - **interference** joints: joints of other person
- How to determine the joint position from a heatmap?
  - max(H(p))
  - Integral pose regression ($J_{int} = \sum_p p * H(p)$)
  - **extract more joints**, then **filter them** by global association, since it is hard or **impossible to distinguish** them in Ri

Joint-candidate

Heatmap Loss For the k^{th} joint in the i^{th} person, we denote the target joint heatmap as $\mathbf{T}_i^k$, consisting of a 2D Gaussian $G(\mathbf{p}_i^k|\sigma)$, centered at the target joint location $\mathbf{p}_i^k$, with standard deviation σ.

For interference joints, we denote them as a set Ω_i^k. The heatmap of interference joints is denoted as $\mathbf{C}_i^k$, consisting of a Gaussian mixture distribution $\sum_{p\in\Omega_i^k} G(\mathbf{p}|\sigma)$.

Our proposed loss is defined as,

$$Loss_i = \frac{1}{K}\sum_{k=1}^{K} MSE[\mathbf{P}_i^k, \mathbf{T}_i^k + \mu\mathbf{C}_i^k] \qquad (1)$$

- μ = 0.5: interference joints should be attenuated but not over-suppressed
- conventional heatmap loss is a special case where μ = 0.

Person-Joint Graph -- Joint Node Building

$$\mathcal{J} = \{v_j^k : \text{for } k \in \{1, \ldots, K\}, j \in \{1, \ldots, N_k\}\}$$

where N_k is the number of joint nodes of body part k, v_j^k is the j^{th} node of body part k. The total number of joint nodes in $\mathcal{J}$ is $\sum_k N_k$.

- Candidate joints k of same person may appears on heatmaps of different boxes, i.e. person proposals, using the Joint-candidate loss.
- How to group candidate joints of joint k located at pk1 and pk2

$$||p_1^{(k)} - p_2^{(k)}||_2 \leq \min\{u_1^k, u_2^k\}\delta^{(k)}$$

where u_1^k and u_2^k are the Gaussian response size of two joints on heatmaps, determined by the Gaussian response deviation. $\delta^{(k)}$ is the parameter for controlling deviation of the k^{th} joint, which we directly adopt from MSCOCO keypoint dataset [19]. The reason why we use $\min\{u_1, u_2\}$

Person-Joint Graph -- Person Node Building

- Person nodes represent the human proposals detected by human detector.

$$\mathcal{H} = \{h_i : \forall i \in \{1, \ldots, M\}\}$$

- M is the number of detected human proposals.
- There are many redundant proposals, which will be eliminated during global person-joint matching

Person-Joint Graph -- Person-Joint Edge

After obtaining the node of both joints and persons, we connect them to construct our person-joint graph. If a joint node v_j^k contains a candidate joint from person node h_i, we build an edge $e_{i,j}^k$ between them. The weight of $e_{i,j}^k$ is the response score of that candidate joint, which is denoted as $w_{i,j}^k$. In this way, we can construct edge set $\mathcal{E} = \{e_{i,j}^k : \forall i, j, k\}$.

The person-joints graph can then be written as:

$$\mathcal{G} = ((\mathcal{H}, \mathcal{J}), \mathcal{E}). \tag{3}$$

Globally Optimizing Association

$$\max_d \mathcal{G} = \max_d \sum_{i,j,k} w_{i,j}^{(k)} \cdot d_{i,j}^{(k)} \tag{4}$$

$$s.t. \quad \sum_j d_{i,j}^{(k)} \leq 1, \quad \begin{array}{l} \forall k \in \{1, \ldots, K\}, \\ \forall i \in \{1, \ldots, M\} \end{array} \tag{5}$$

$$\sum_i d_{i,j}^{(k)} \leq 1, \quad \begin{array}{l} \forall k \in \{1, \ldots, K\}, \\ \forall j \in \{1, \ldots, N_k\} \end{array} \tag{6}$$

$$d_{i,j}^{(k)} \in \{0, 1\}, \quad \forall i, j, k \tag{7}$$

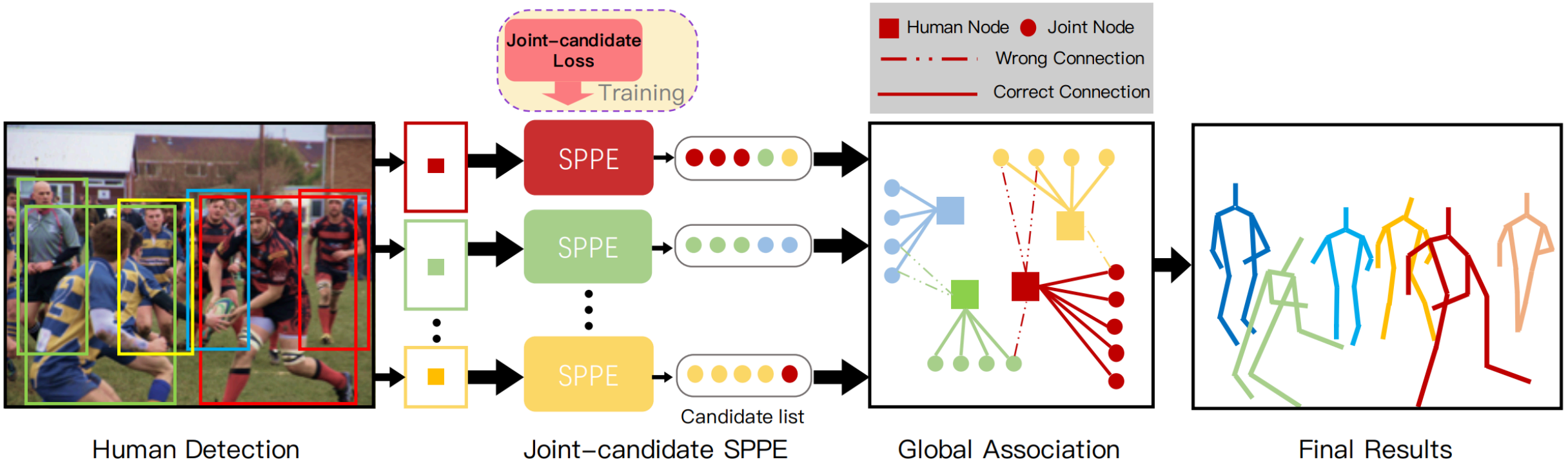
$$\max_d \mathcal{G} = \max_d \sum_{i,j,k} w_{i,j}^{(k)} \cdot d_{i,j}^{(k)}$$

$$= \sum_{k=1}^{K} (\max_{d^{(k)}} \sum_{i,j} w_{i,j}^{(k)} \cdot d_{i,j}^{(k)})$$

$$= \sum_{k=1}^{K} \max_{d^{(k)}} \mathcal{G}_k.$$

$$\mathcal{O}(n^2) = \mathcal{O}((|\mathcal{H}| + |\mathcal{J}^k|)^2) \leq \mathcal{O}(|\mathcal{H}|^2)$$

As shown in Eq. 10, solving the global assignment problem in person-joint graph $\mathcal{G}$ is mathematically equivalent to solving its sub-graph $\mathcal{G}_k$ separately. $\mathcal{G}_k$ is a bipartite graph that composed of person subset and the k^{th} joint subset. For each sub-graph, the updated Kuhn-Munkres algorithm [6] is applied to get the optimized result. By addressing each $\mathcal{G}_k$ respectively, we obtain the final result set $\mathcal{R}$.



- joint-candidate SPPE
- global association

SPPE: single person pose estimation

效果 MSCOCO提高5.2mAP

Figure 7. Qualitative results of our models predictions is presented. Different person poses are painted in different colors to achieve better visualization.

Pose Estimation on MPII Human Pose

Rank	Method	PCKh-0.5	Paper Title	Year	Paper	Code
1	Cascade Feature Aggregation	93.9%	Cascade Feature Aggregation for Human Pose Estimation	2019		
2	Spatial Context	92.5%	Human Pose Estimation with Spatial Contextual Information	2019		
3	HRNet-32	92.3%	Deep High-Resolution Representation Learning for Human Pose Estimation	2019		
4	Multi-Scale Structure-Aware Network	92.1%	Multi-Scale Structure-Aware Network for Human Pose Estimation	2018		
5	Pyramid Residual Modules (PRMs)	92.0%	Learning Feature Pyramids for Human Pose Estimation	2017		
6	Multi-Context Attention	91.5%	Multi-Context Attention for Human Pose Estimation	2017		
7	DU-Net	91.2%	Quantized Densely Connected U-Nets for Efficient Landmark Localization	2018		
8	Stacked hourglass + Inception-resnet	91.2%	Knowledge-Guided Deep Fractal Neural Networks for Human Pose Estimation	2017		
9	Integral Regression	91.0%	Integral Human Pose Regression	2017		
10	Stacked Hourglass Networks	90.9%	Stacked Hourglass Networks for Human Pose Estimation	2016		
11	CU-Net	89.4%	CU-Net: Coupled U-Nets	2018		
12	Convolutional Pose Machines	88.52%	Convolutional Pose Machines	2016		

Keypoint Detection on COCO

Rank	Method	Test AP	Validation AP	Paper Title	Year
1	MSPN	76.1		Rethinking on Multi-Stage Networks for Human Pose Estimation	2019
2	HRNet-48	75.5	76.3	Deep High-Resolution Representation Learning for Human Pose Estimation	2019
3	CPN+	73.0		Cascaded Pyramid Network for Multi-Person Pose Estimation	2017
4	PifPaf – single-scale (ours)	66.7		PifPaf: Composite Fields for Human Pose Estimation	2019
5	PersonLab	66.5		PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model	2018
6	Mask R-CNN	63.1	69.2	Mask R-CNN	2017
7	HG-jd	63.0		Objects as Points	2019
8	Pose-AE	62.8		Associative Embedding: End-to-End Learning for Joint Detection and Grouping	2016
9	HRNet-32		75.8	Deep High-Resolution Representation Learning for Human Pose Estimation	2019
10	ResNet-50		72.2	Simple Baselines for Human Pose Estimation and Tracking	2018
11	Pose Residual Network		69.6	MultiPoseNet: Fast Multi-Person Pose Estimation using Pose Residual Network	2018

Simple Baselines for Human Pose Estimation and Tracking, eccv 2018

Bin Xiao1, Haiping Wu2, and Yichen Wei
1. Microsoft Research Asia, 2 University of Electronic Science and Technology of China

- The previous approaches work very well but are complex.
- This work follows the question – ***how good could a simple method be****?*
- And achieved the state-of-the-art at mAP of 73.7% on COCO.
  - 2016年冠军PAF（openpose，mAP=60.5）和2017年冠军CPN（mAP=72.1）；

Missions

- Pose estimation
- Pose tracking

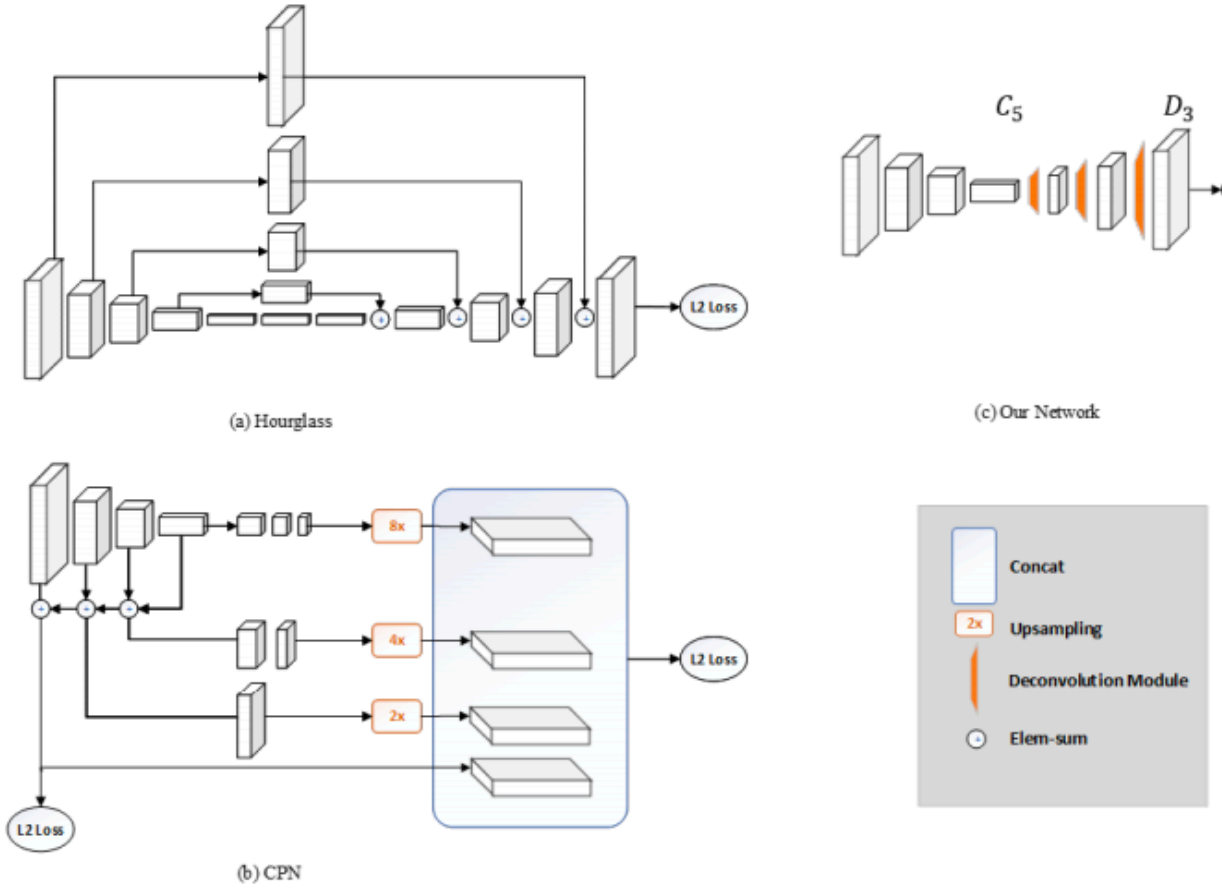
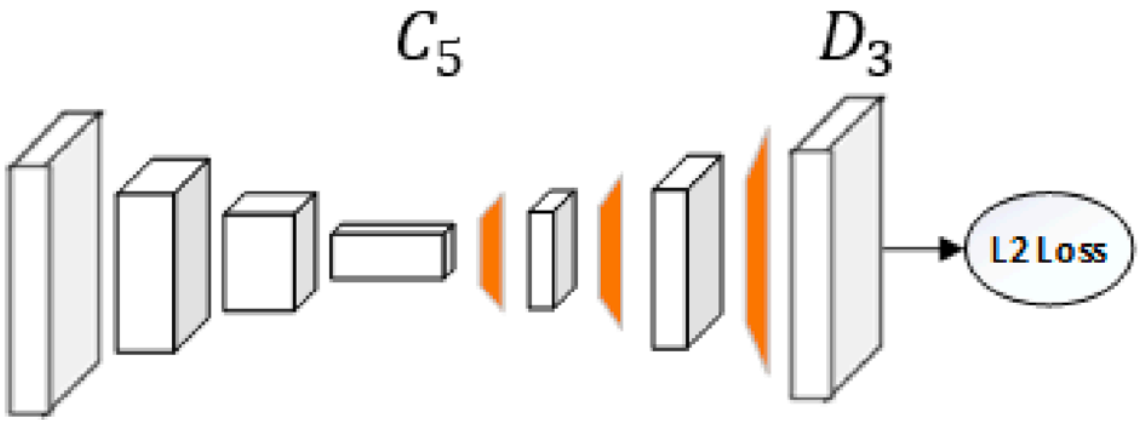


Fig. 1. Illustration of two state-of-the-art network architectures for pose estimation (a) one stage in Hourglass [22], (b) CPN [6], and our simple baseline (c).

- It was quite surprising to see such a simple architecture perform **better than one with skip connections** that preserve the information for each resolution, such as hourglass, CPN

Resnet + Deconvolution + MSE



Method	Backbone	Input Size	OHKM	*AP*
8-stage Hourglass	-	256 × 192	✗	66.9
8-stage Hourglass	-	256 × 256	✗	67.1
CPN	ResNet-50	256 × 192	✗	68.6
CPN	ResNet-50	384 × 288	✗	70.6
CPN	ResNet-50	256 × 192	✓	69.4
CPN	ResNet-50	384 × 288	✓	71.6
Ours	ResNet-50	256 × 192	✗	70.4
Ours	ResNet-50	384 × 288	✗	72.2

- Deconvolution layers
  - Deconvolution 层数越多，生成的热力图分辨力就越大。三层Deconvolution 生成的热力图大小为64*48，两层为32*24，实验结果三层结果较好。
- Input size和Backbone
  - input_size越大，backbone越深，模型性能越好，但是同时计算量也大大增加。
- Kernel size
  - Kernel size减小，AP也稍微降低。所以本文的Kernel size都选取4。

Pose tracking – optical flow

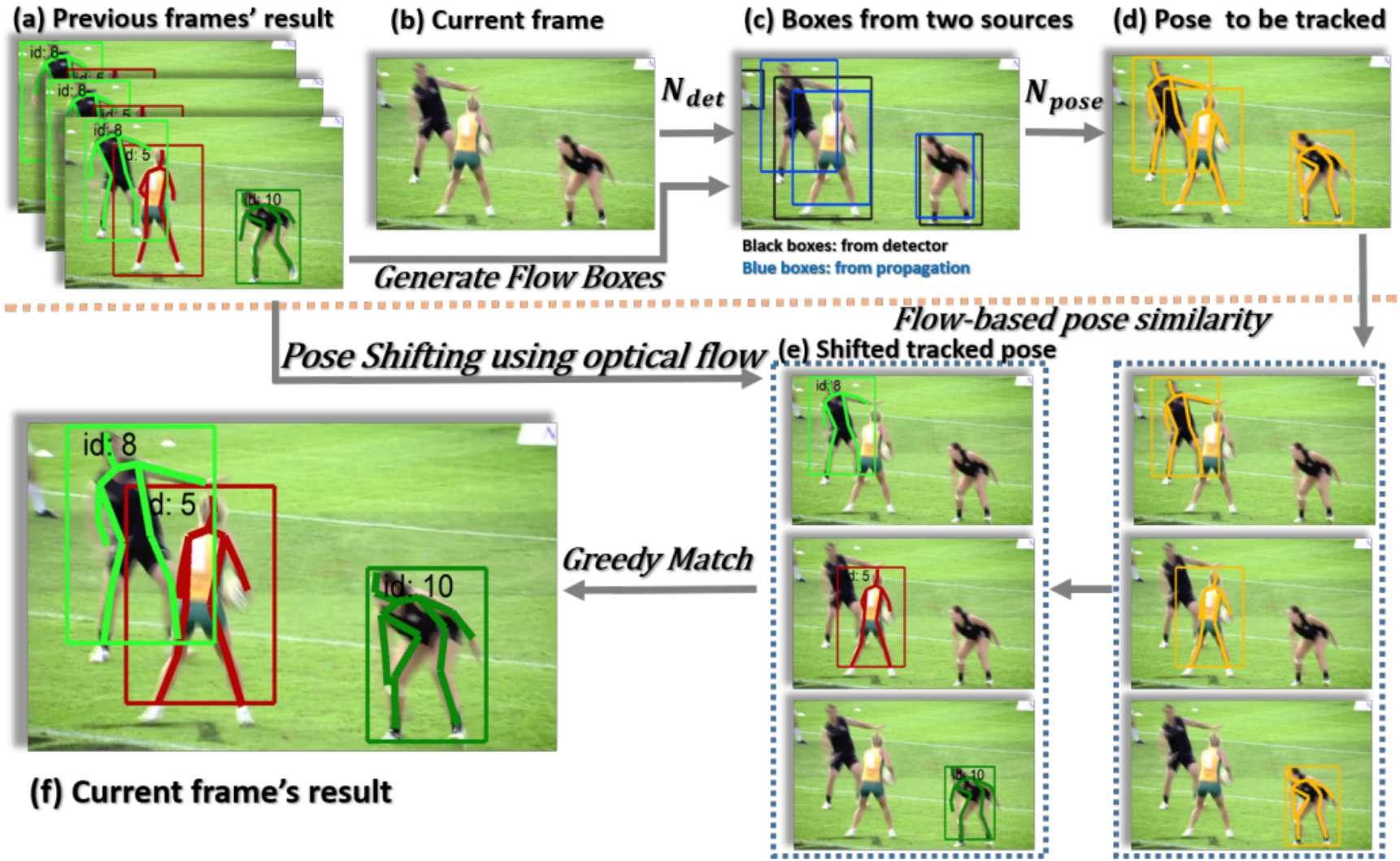


Fig. 2. The proposed flow-based pose tracking framework.

Pose tracking – Object Keypoint Similarity (OKS) v.s. IOU of box

Object Keypoint Similarity (OKS)

$$OKS = \sum_i [\exp(-d_i^2 / 2s^2 k_i^2)\delta(v_i > 0)] / \sum_i [\delta(v_i > 0)]$$

d_i 是标注和预测关节点之间的欧氏距离

sk_i 为标准差

每个关节点的相似度都会在$[0, 1]$之间，完美的预测将会得到$OKS = 1$，预测值与真实值差距太大将会得到$OKS \sim 0$

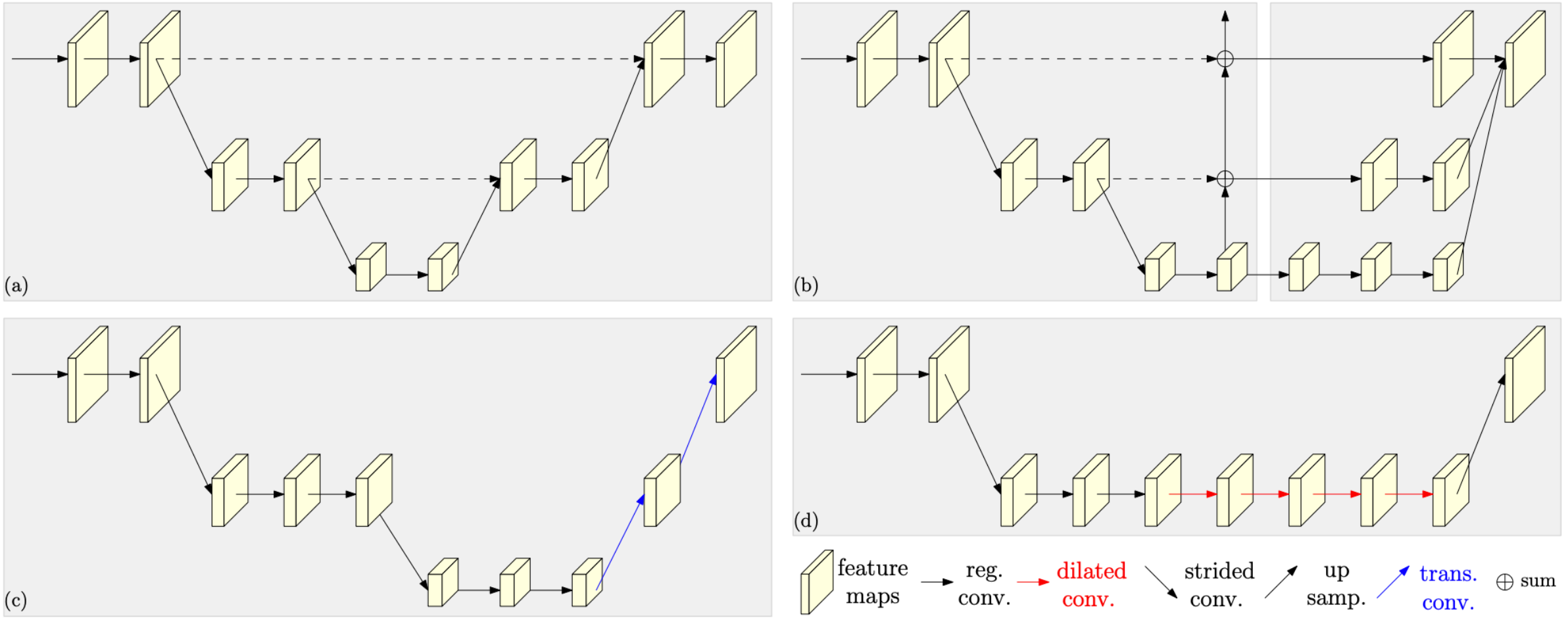
Comments

- High resolution heatmap matters!!

Deep High-Resolution Representation Learning for Human Pose Estimation, cvpr19

Ke Sun, Bin Xiao, Haiping Wu, and Yichen Wei
1. Microsoft Research Asia, 2 University of Electronic Science and Technology of China

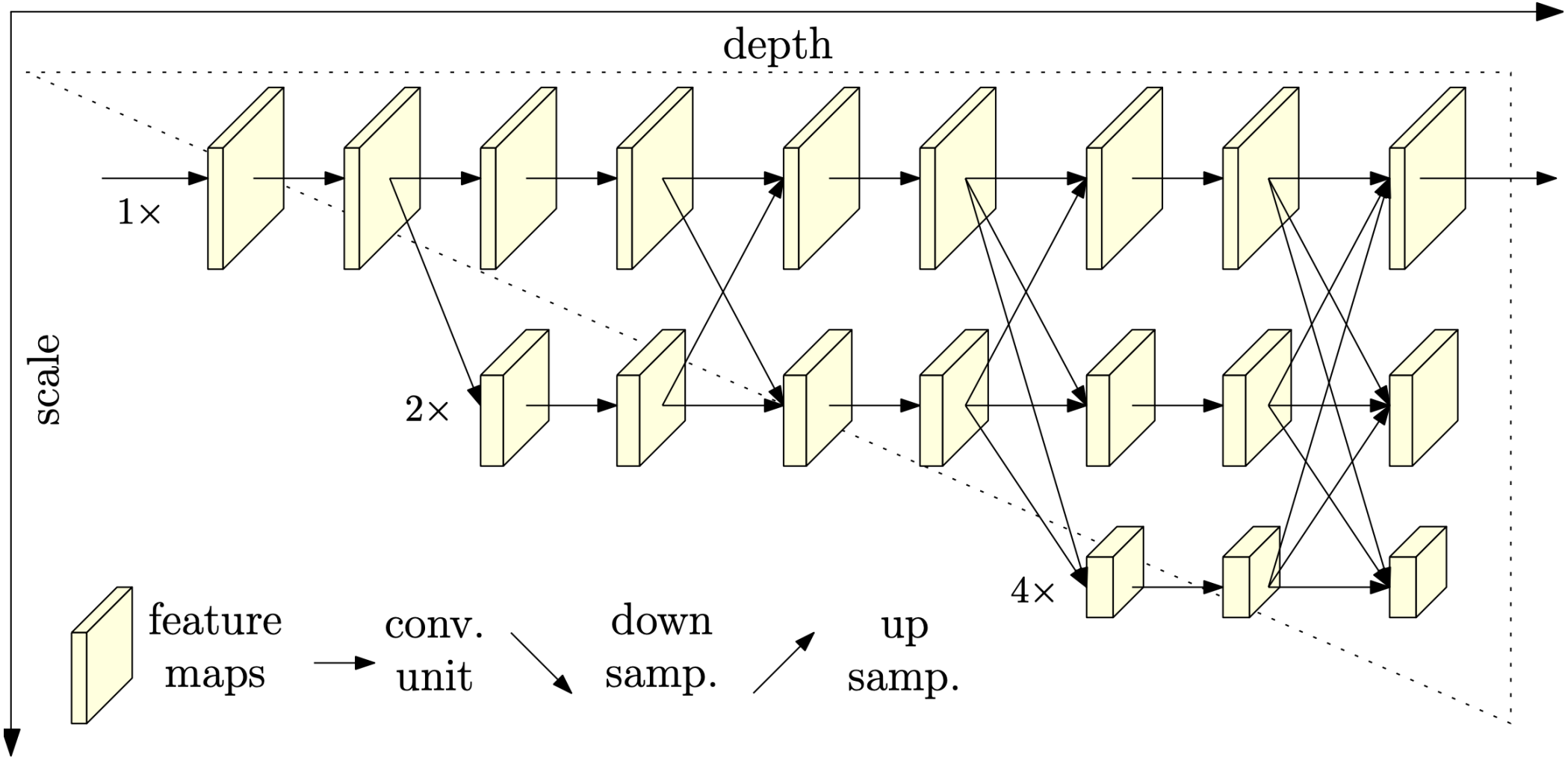
Previous network



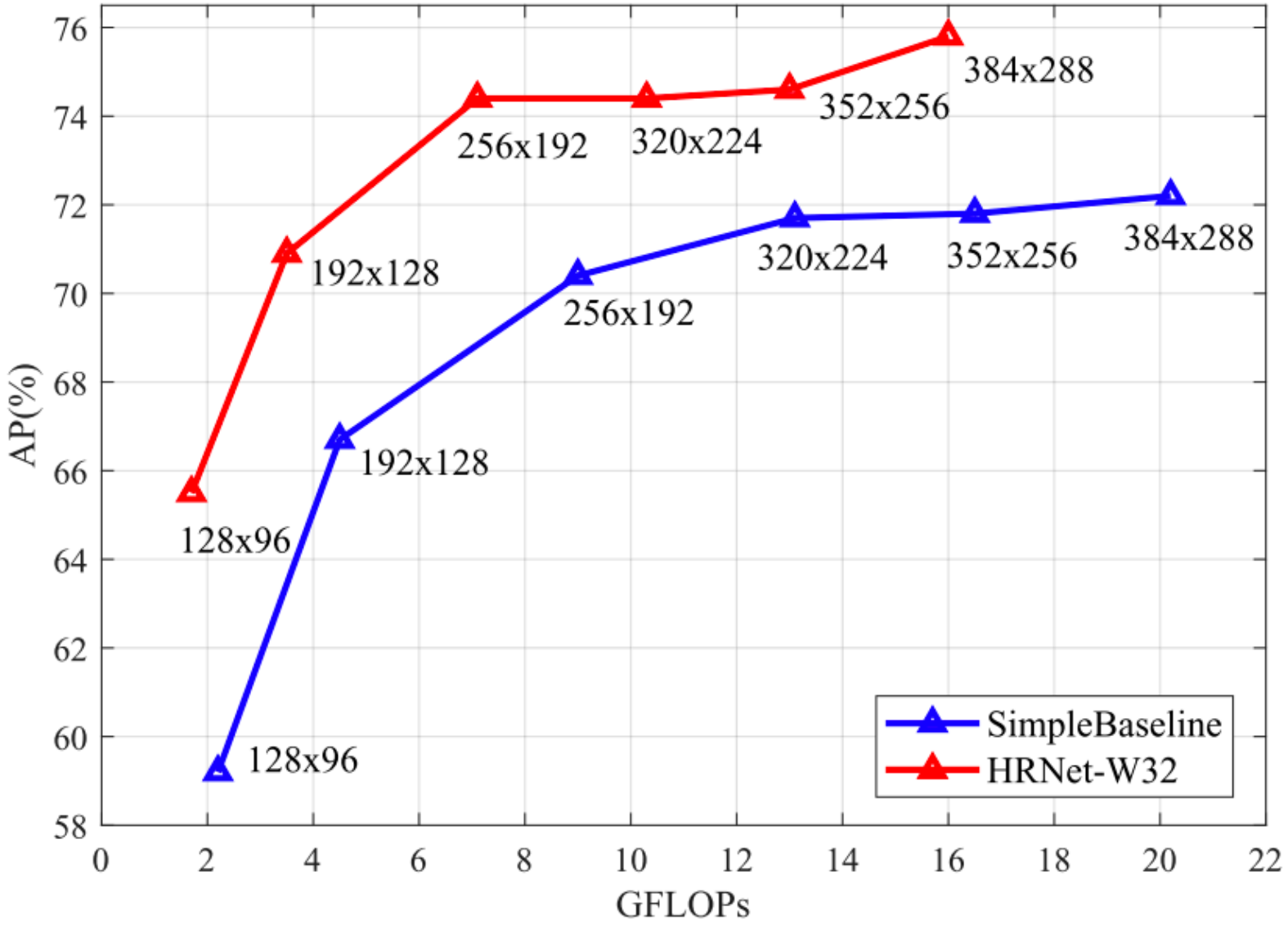
- (a) Hourglass [40]. (b) Cascaded pyramid networks [11]. (c) SimpleBaseline [72] (d) Combination with dilated convolutions [27]

the architecture of the proposed HRNet

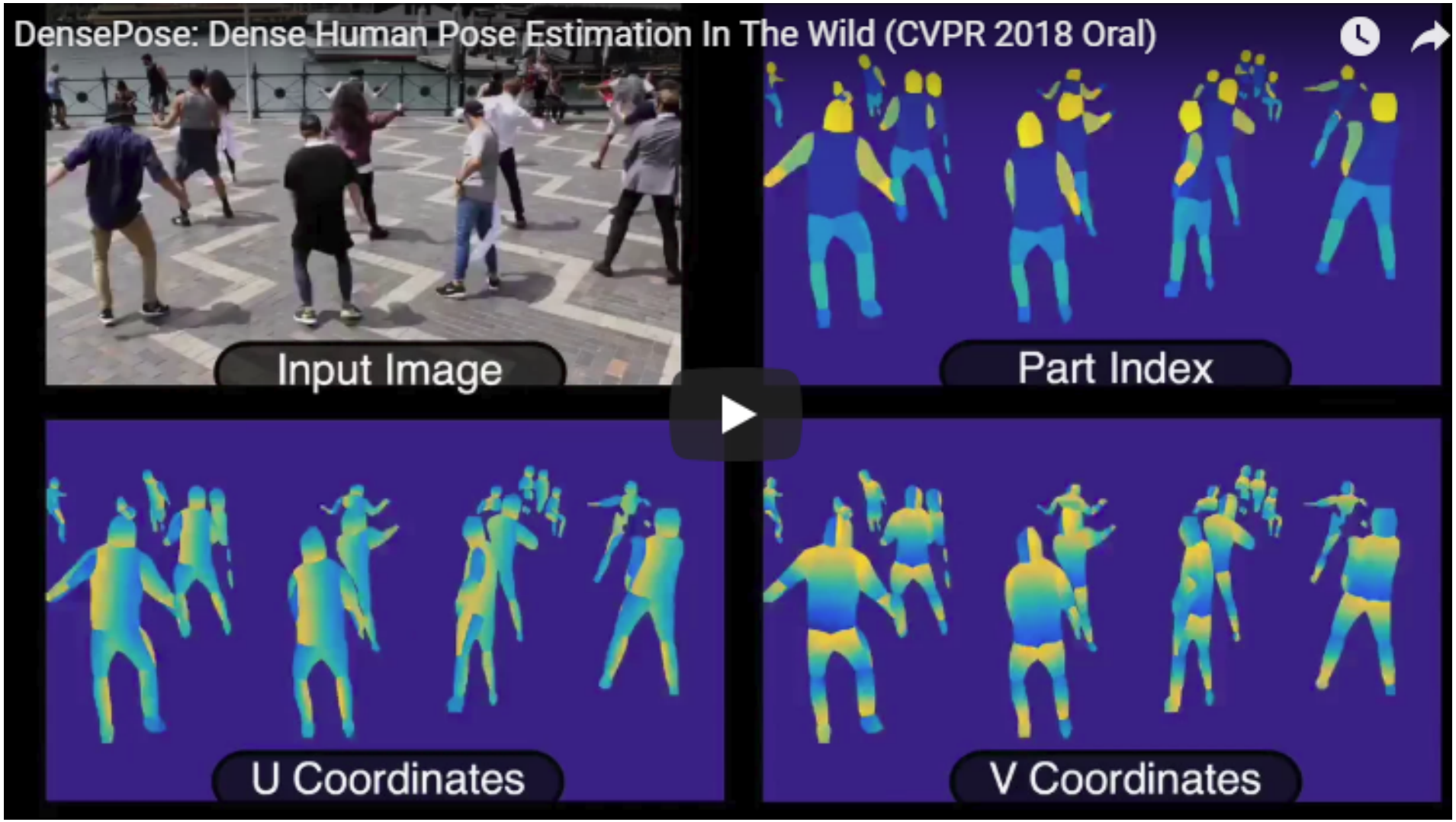
- maintain highresolution representations through the whole process



Performance & input size



Dense Human Pose Estimation In The Wild



Dense Human Pose Estimation In The Wild, cvpr 2018, facebook

Thanks