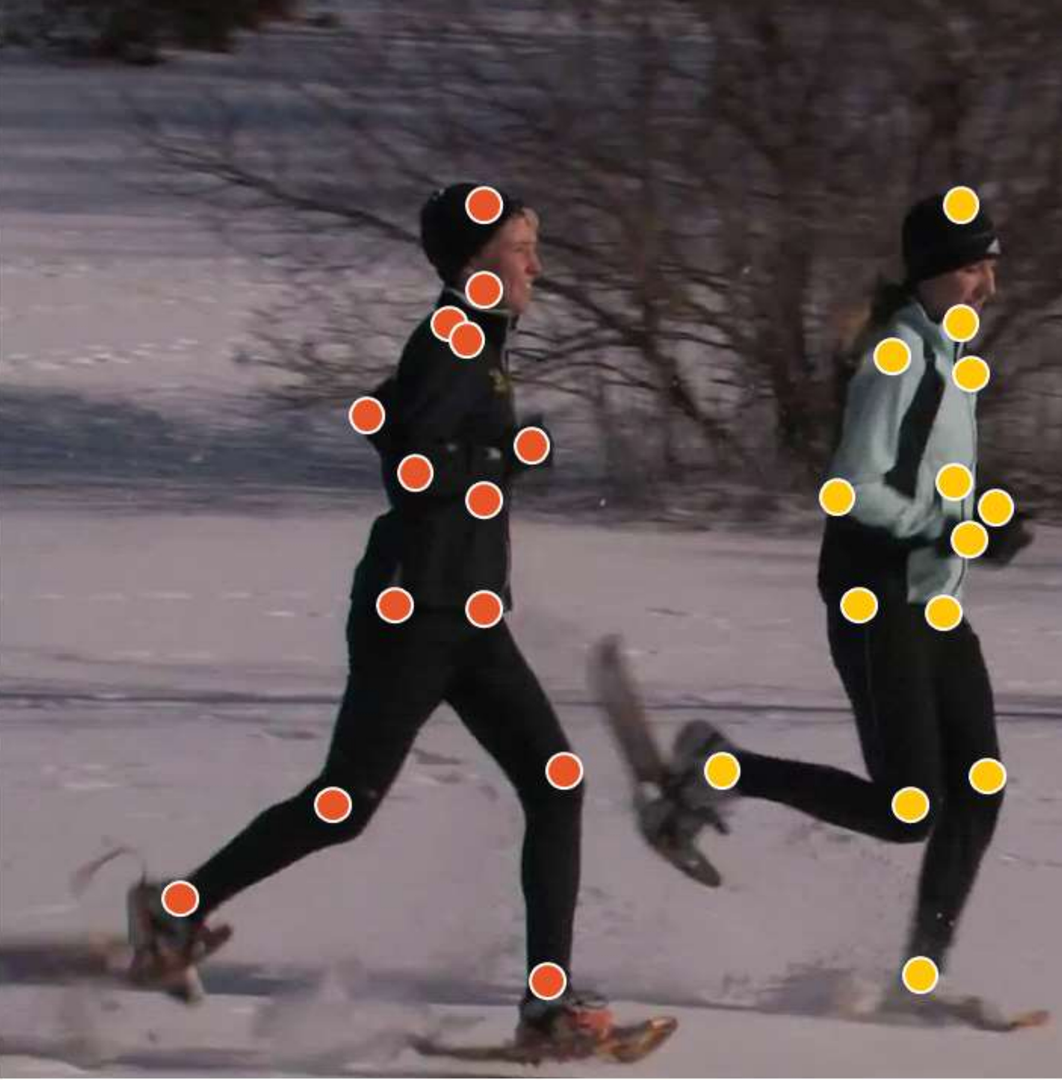
# Lecture 16
# Deep Learning for Human Pose Estimation



Human pose estimation localizes human body parts in images or videos.

- RGB vs. RGBD
- image vs. video
- single person vs. multiple person
- 3D pose estimation
- reconstruct 3D pose from 2D space

# Human pose estimation

Applications

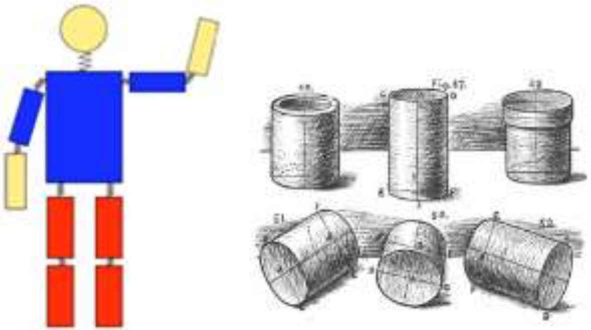
Action recognition

Human Parsing


Game / Animation

Challenges

# Traditional Methods



## Figure Drawing

- Cylinders for each body parts
- Join up the cylinders

## Pictorial Structures

- Unary templates
- Pairwise springs

Fischler & Elschlager 1973
Felzenszwalb & Huttenlocher 2005

## Mixtures of Parts

- Unary template for each mixture type

Yang & Ramanan 2011

# DeepPose

First deep learning based algorithms for human pose estimation

Alexander Toshev × Christian Szegedy

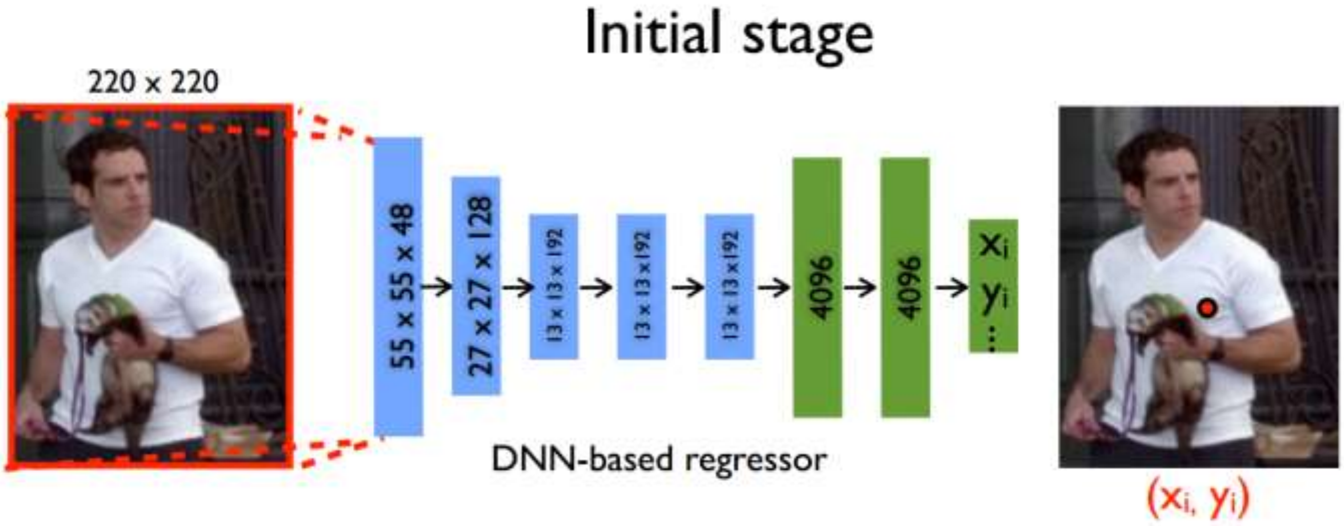
- ❑ By Google in **CVPR 2014**
- ❑ Cascade of DNN regressors

# Can you tell which part is from an image patch?

# What about this patch?

- Local appearance is weak
- Global consistency is important

# DeepPose: Holistic human pose estimation as a DNN

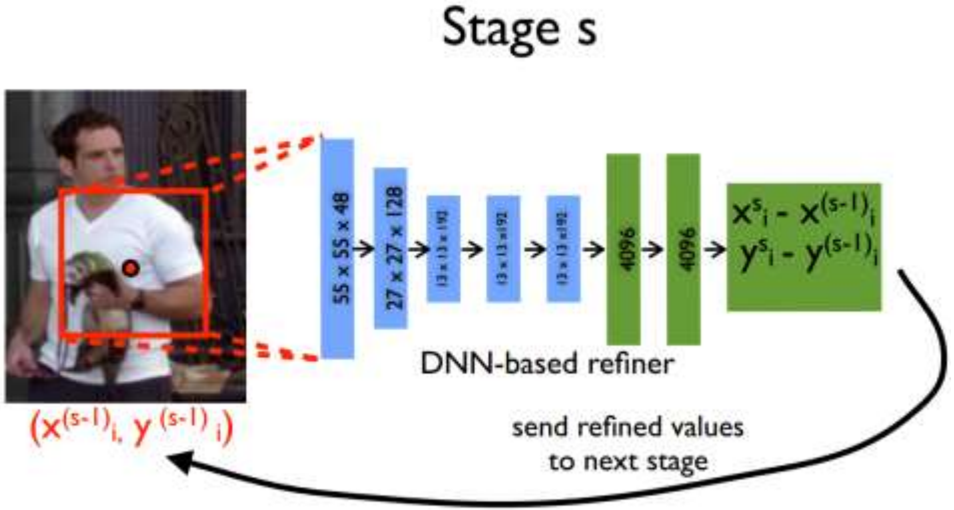



**Training:** Minimizing L2 distance between the prediction and the true pose vector.

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

# Cascade of Pose Regressors

- due to the fixed input size of 220 × 220, the network has limited capacity to look at detail
    - it learns filters capturing pose properties at coarse scale



# Pipeline

Initial stage
220 x 220
55 x 55 x 48
27 x 27 x 128
4096
4096
$x_i$
$y_i$
...
DNN-based regressor
$(x_i, y_i)$
Stage s
55 x 55 x 48
27 x 27 x 128
4096
4096
$x^s_i - x^{(s-1)}_i$
$y^s_i - y^{(s-1)}_i$
DNN-based refiner
$(x^{(s-1)}_i, y^{(s-1)}_i)$
send refined values
to next stage

# Experiments

| Method | Arm | | Leg | | Ave. |
|---|---|---|---|---|---|
| | Upper | Lower | Upper | Lower | |
| DeepPose-st1 | 0.5 | 0.27 | 0.74 | 0.65 | 0.54 |
| DeepPose-st2 | **0.56** | 0.36 | **0.78** | 0.70 | 0.60 |
| DeepPose-st3 | **0.56** | **0.38** | 0.77 | **0.71** | **0.61** |
| Dantone et al. [2] | 0.45 | 0.25 | 0.65 | 0.61 | 0.49 |
| Tian et al. [24] | 0.52 | 0.33 | 0.70 | 0.60 | 0.56 |
| Johnson et al. [13] | 0.54 | **0.38** | 0.75 | 0.66 | 0.58 |
| Wang et al. [25] | **0.565** | 0.37 | 0.76 | 0.68 | 0.59 |
| Pishchulin [17] | 0.49 | 0.32 | 0.74 | 0.70 | 0.56 |

Percentage of Correct Parts (PCP) at 0.5 on LSP.

# Limitations

→Low accuracy in high precision region

→One prediction per image. No candidate.

→What if the initial estimate is very far from the groundtruth?

**Solution**: use **heatmaps**

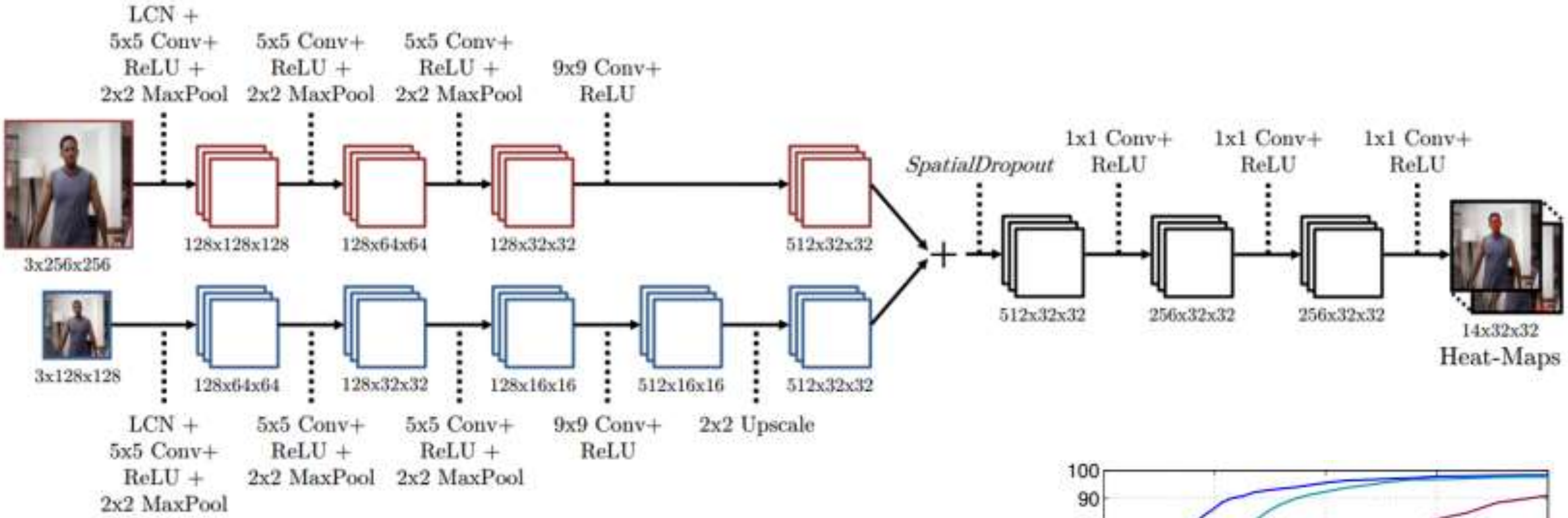
# Solution

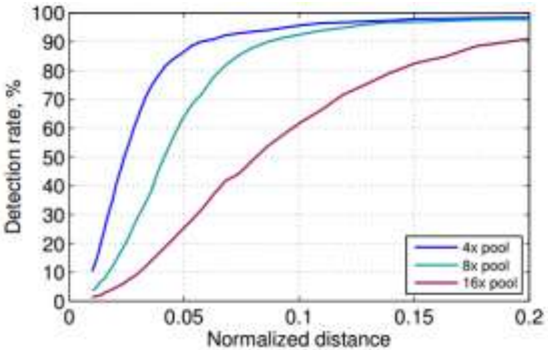
## Heatmaps regression

Efficient Object Localization Using Convolutional Networks
**CVPR 2015**

- ❑ NYU team (Yann LeCun)
- ❑ Larger and wilder dataset
  - ❑ MPII Human Pose Estimation DB
  - ❑ 20000+ images from YouTube

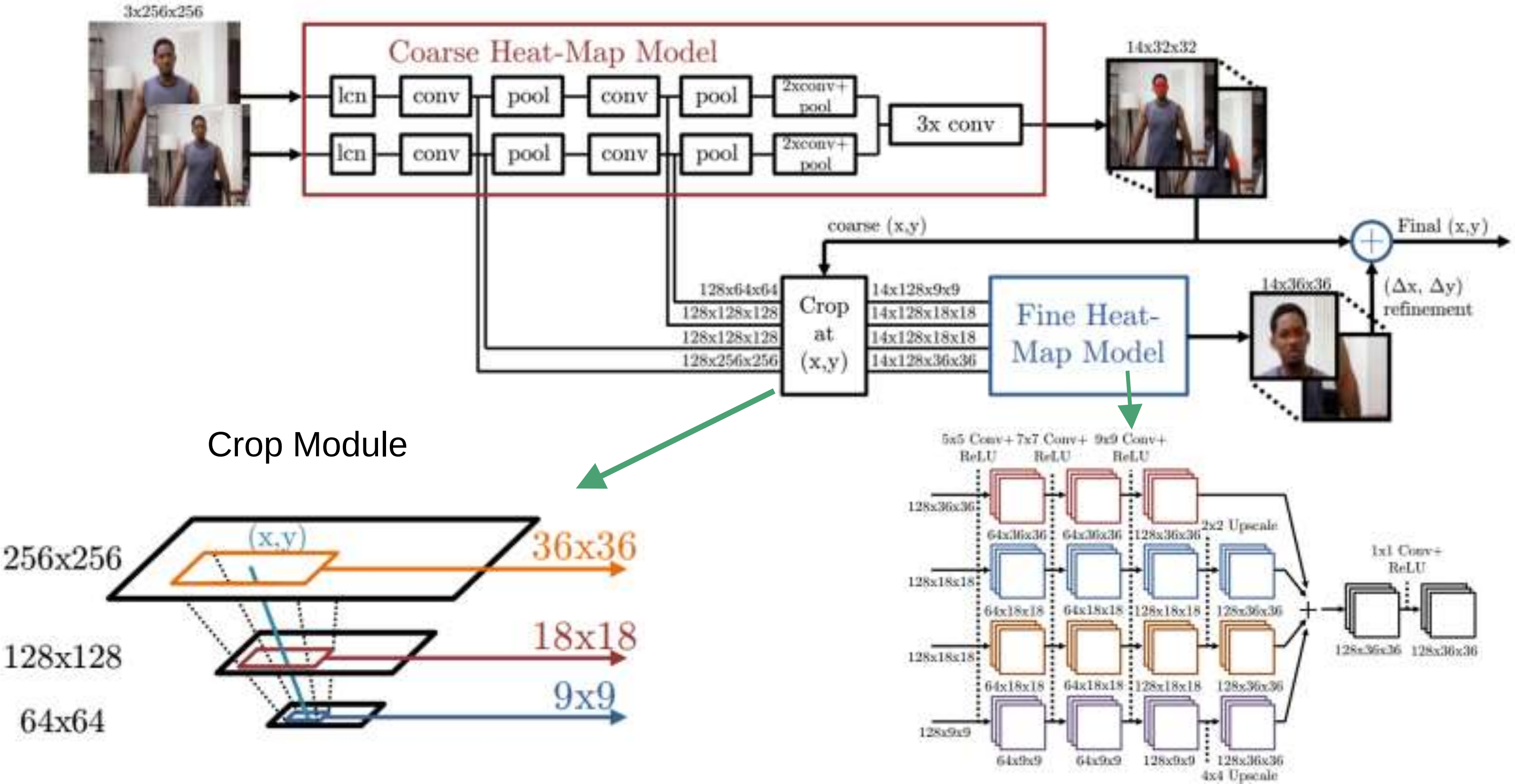
# Multi-Resolution Heatmap Regressor



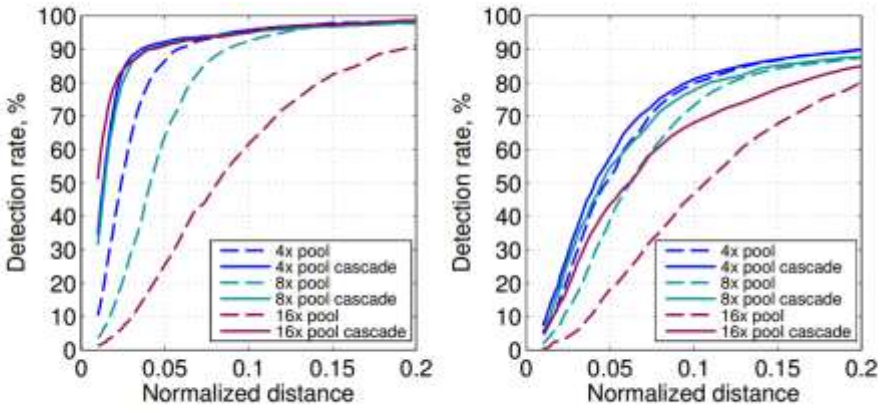




**Observation**: The spatial invariance achieved by pooling layers comes at the price of limiting spatial localization accuracy.

# Cascade Heatmap Regression Model



# Experiments




| | Head | Shoulder | Elbow | Wrist | Hip | Knee | Ankle | Upper Body | Full Body |
|---|---|---|---|---|---|---|---|---|---|
| Gkioxari et al. | - | 36.3 | 26.1 | 15.3 | - | - | - | 25.9 | - |
| Sapp & Taskar | - | 38.0 | 26.3 | 19.3 | - | - | - | 27.9 | - |
| Yang & Ramanan | 73.2 | 56.2 | 41.3 | 32.1 | 36.2 | 33.2 | 34.5 | 43.2 | 44.5 |
| Pishchulin et al. | 74.2 | 49.0 | 40.8 | 34.1 | 36.5 | 34.4 | 35.1 | 41.3 | 44.0 |
| This work 4x | **96.0** | **91.9** | **83.9** | **77.7** | **80.9** | **72.2** | **64.8** | **84.5** | **82.0** |

Performance improvement from cascaded model

Comparison with prior-art: MPII (PCKh @ 0.5)

# Missing point?

- ➔ These methods lack of structure modeling
- ➔ Still have the same problem as DeepPose: What if the true part is not in the cropped region?

# Context matters

Which part corresponds to a body part?

# Context matters

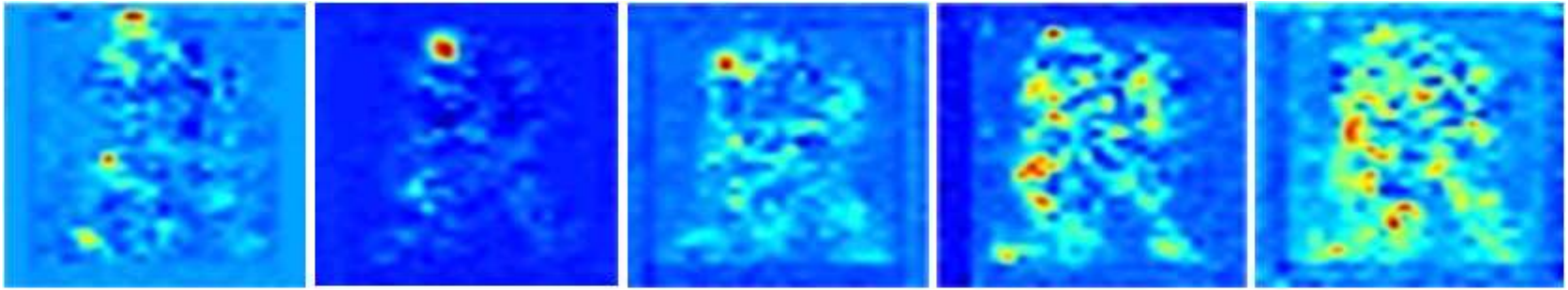
Which part corresponds to a body part?

# Context matters

- ❖ Local evidence is weak
- ❖ Part context is a strong cue
- ❖ Larger context = larger receptive field
  - ➢ More pooling
  - ➢ Deeper network

# Local Image Evidence is Weak

Certain parts are easier to detect than others



# Part Context is a Strong Cue

Part detection confidences provide spatial context cues

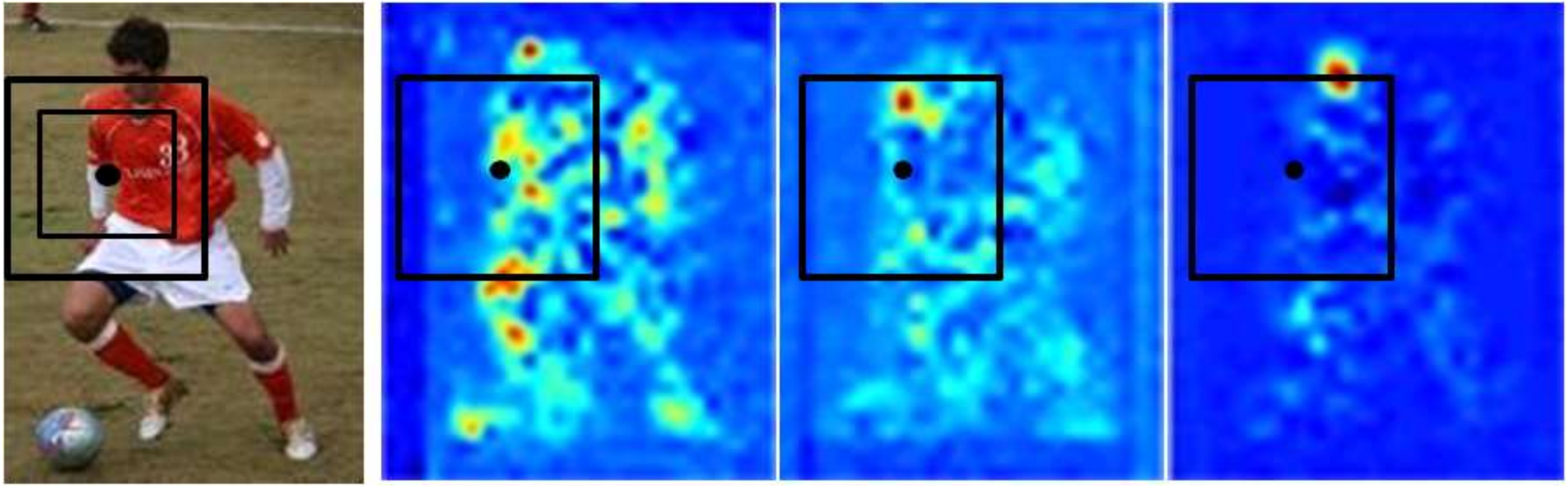
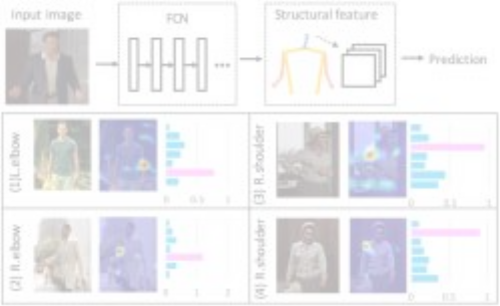
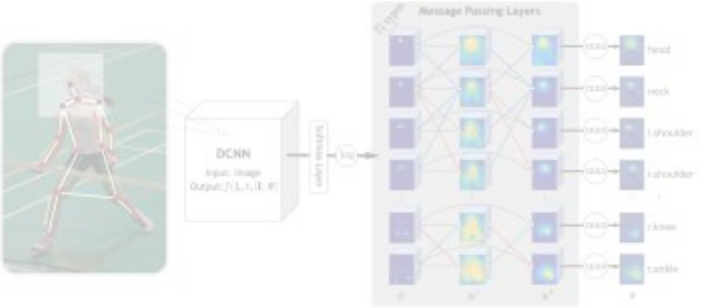


Image | Left elbow | Left shoulder | Neck

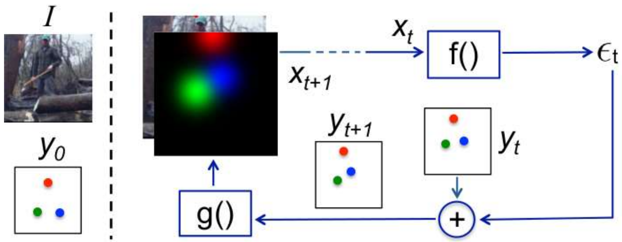
# Structure also matters...



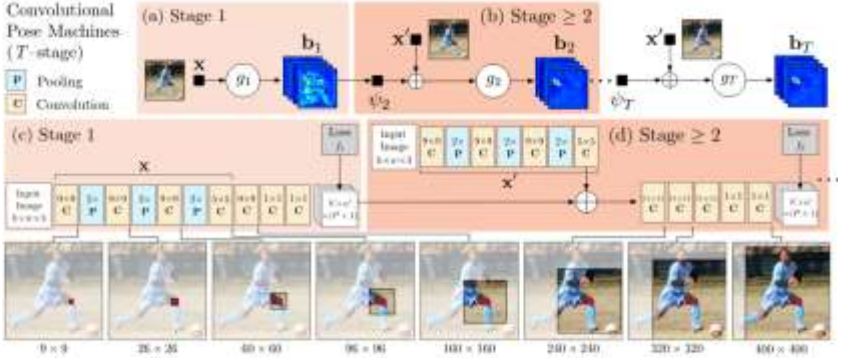

Structured Feature Learning



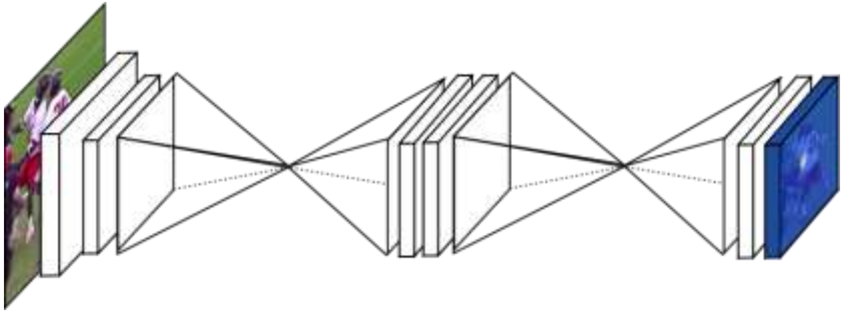

Deep Mixture of Parts




Iterative Error Feedback




Convolutional Pose Machine

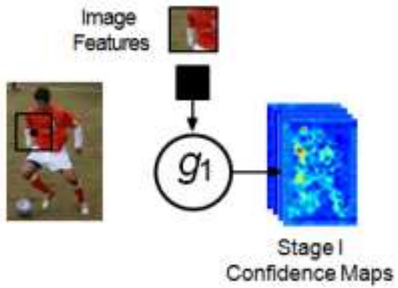


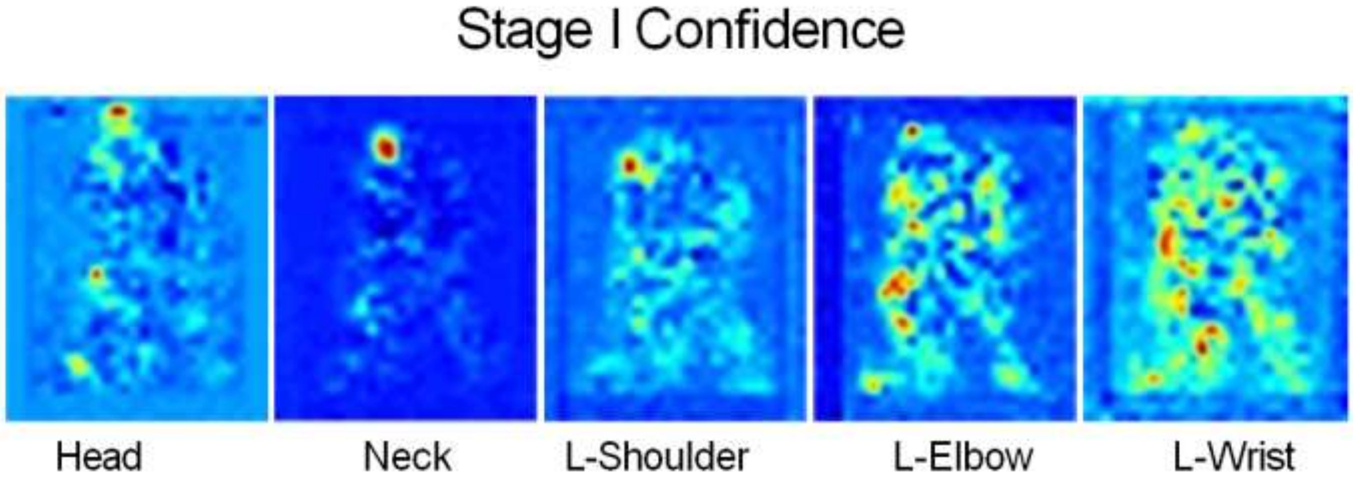
Stacked Hourglass

# Convolutional Pose Machine

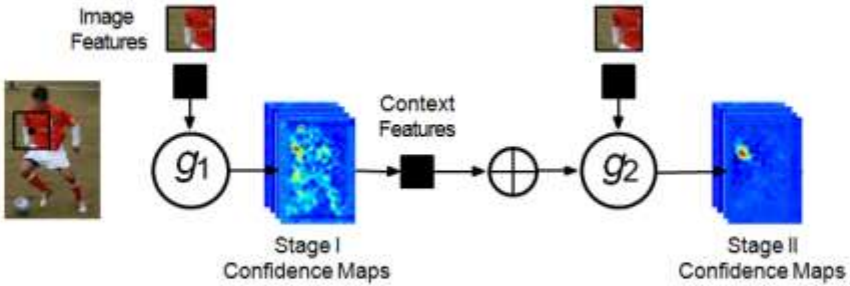
CVPR 2016
CMU

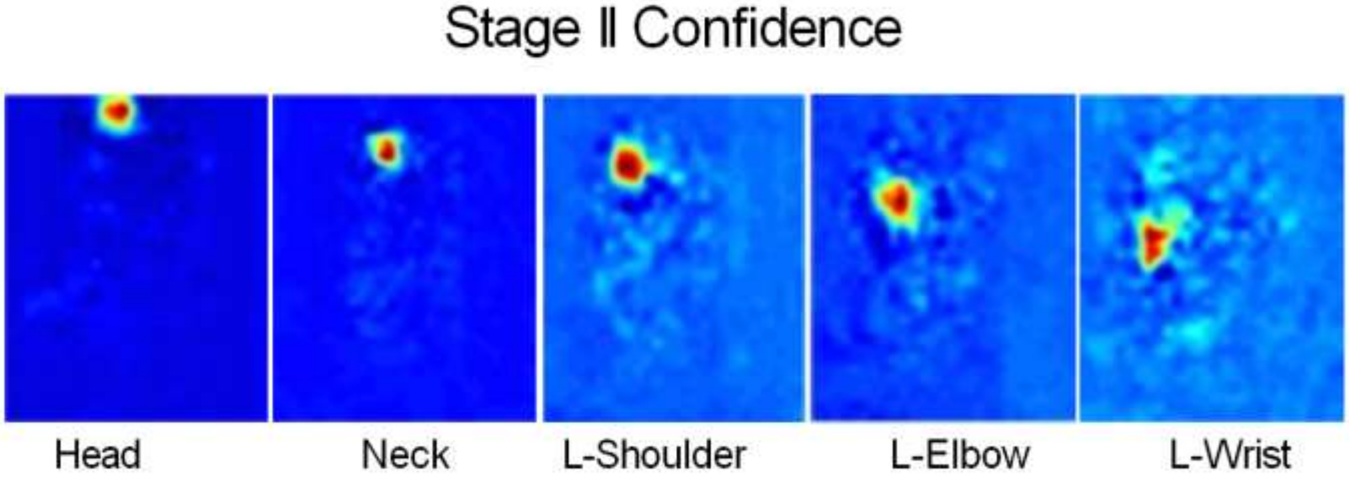
# Previous Work: Pose Machine (ECCV 14)



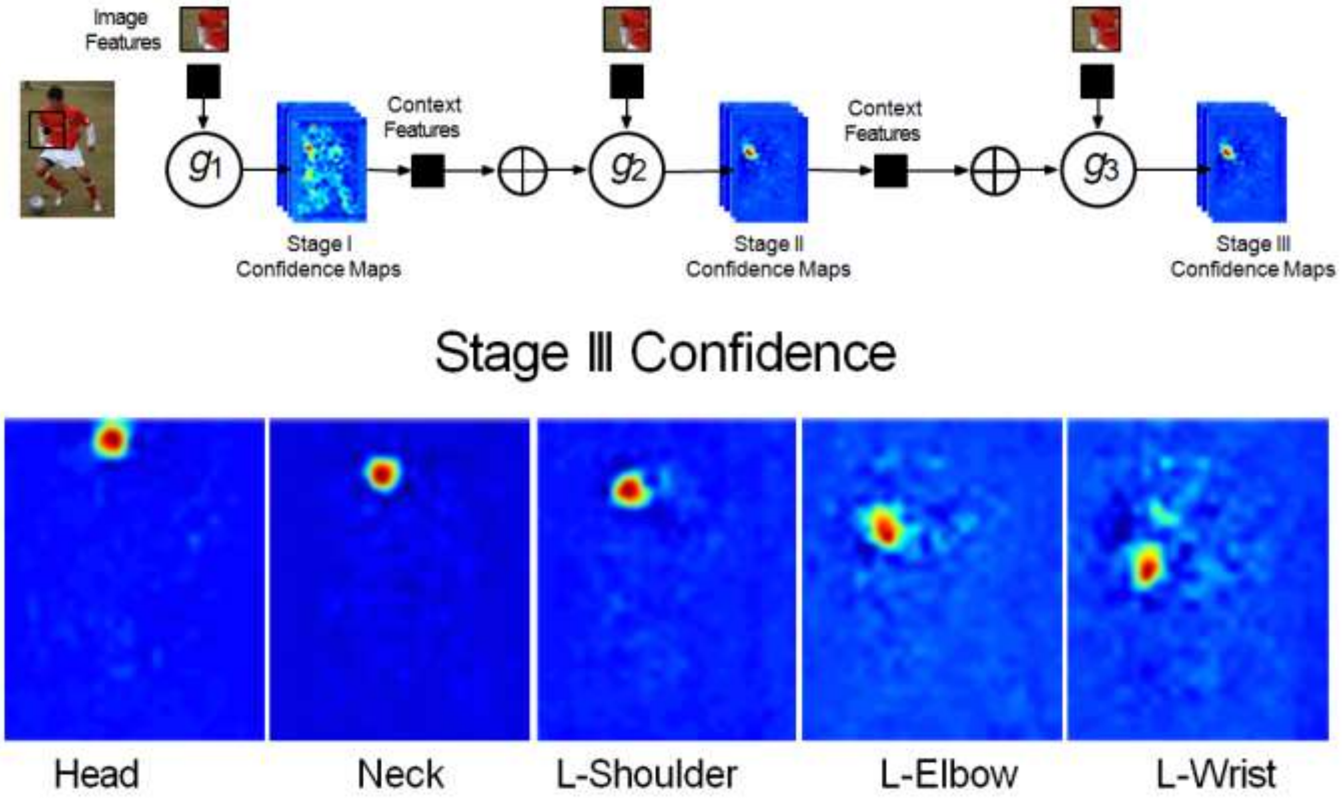



# Previous Work: Pose Machine (ECCV 14)



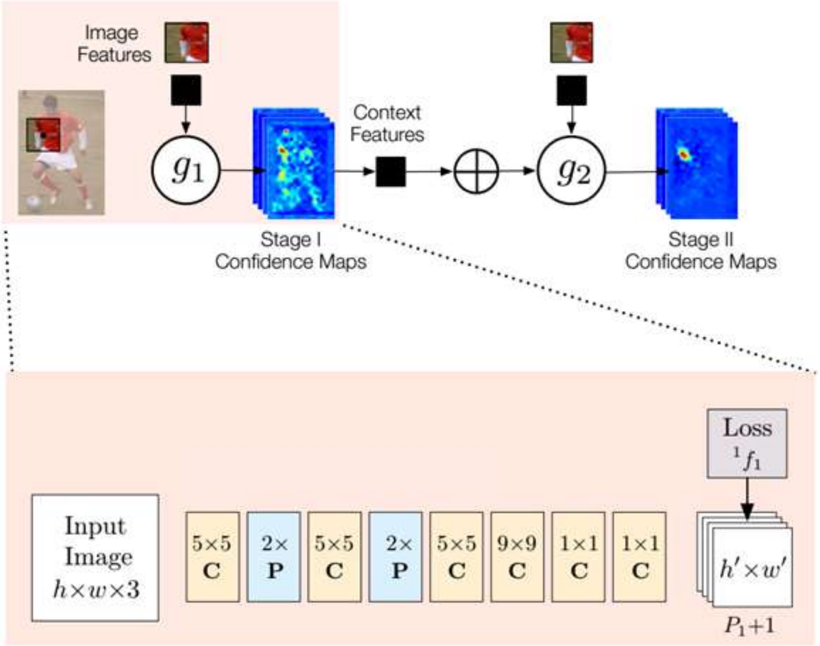



# Previous Work: Pose Machine (ECCV 14)



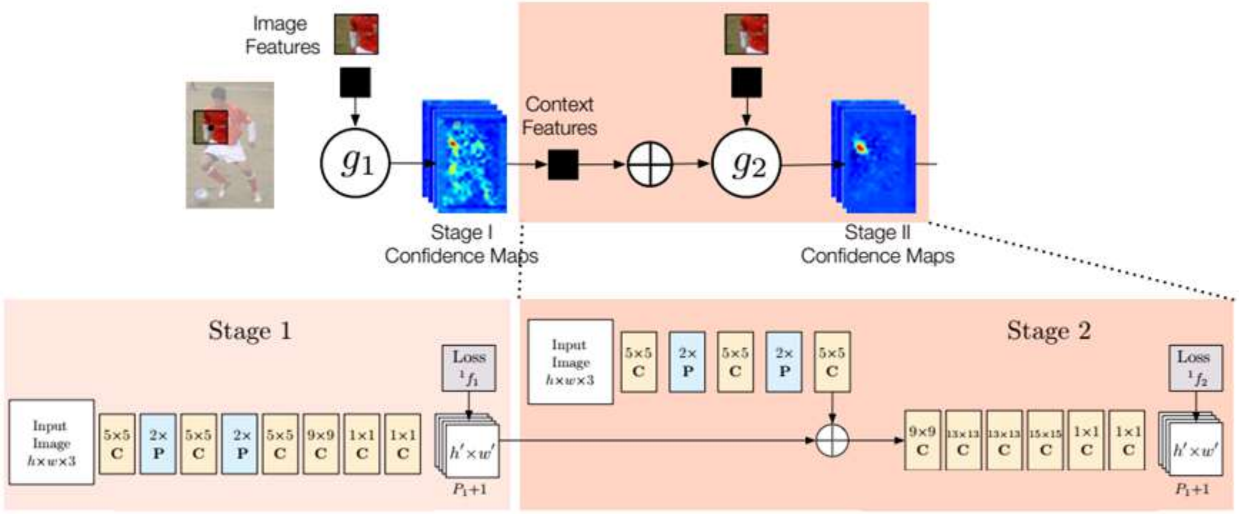
# CPM: Learning Feature Representations

Convolutional Architectures for Feature Embedding

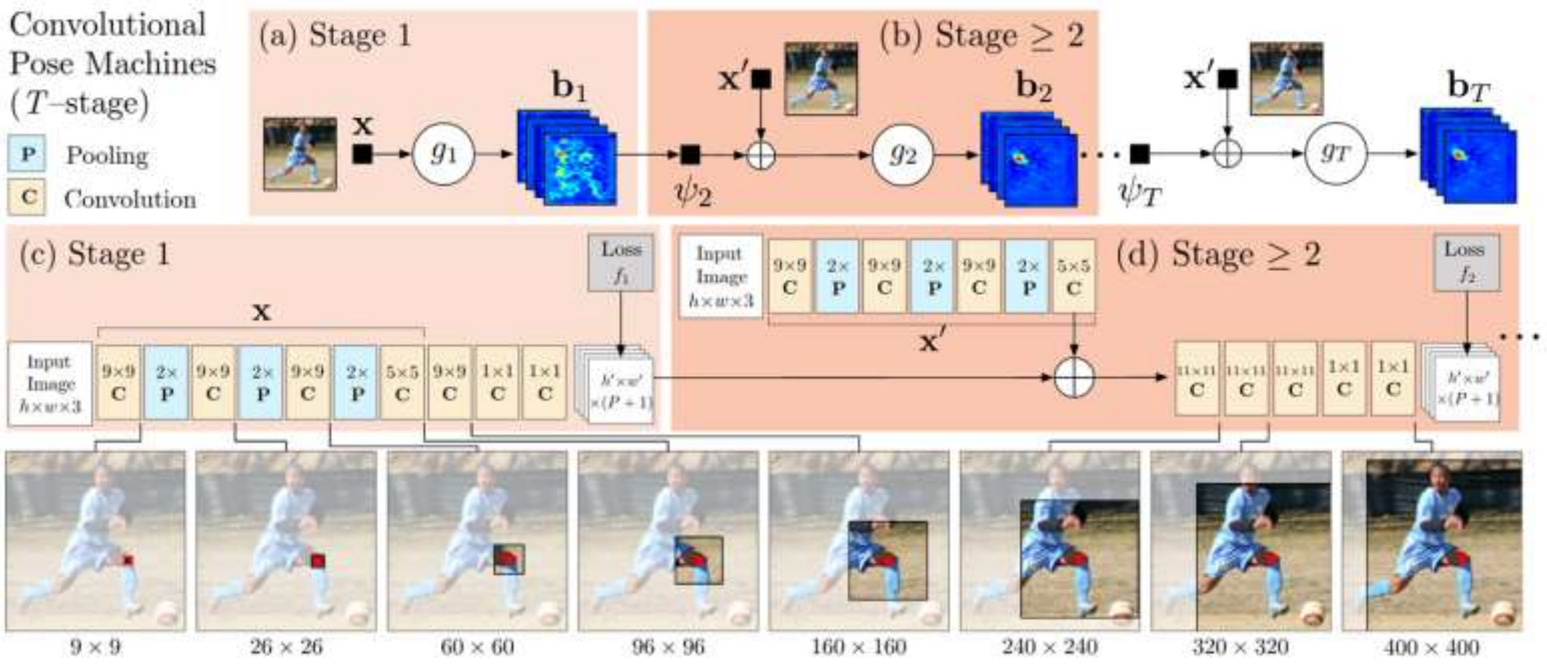


# CPM: Learning Feature Representations

## Convolutional Architectures for Feature Embedding

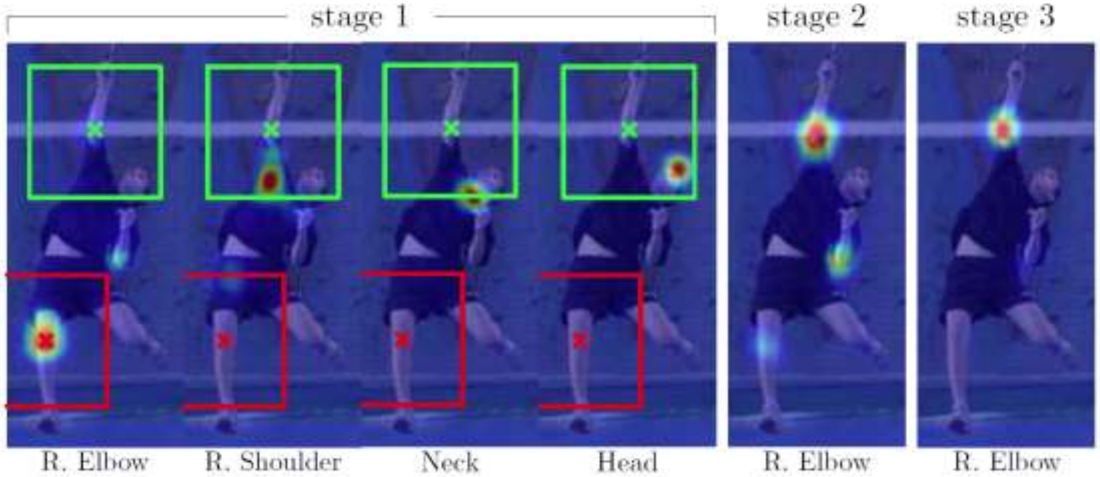


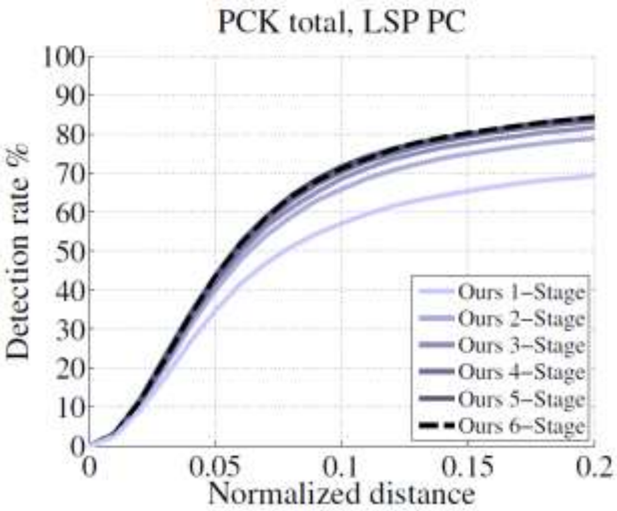
# Full Pipeline



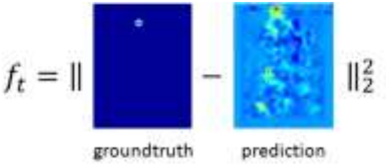

**Receptive field is quite important**

# Spatial Context from Heatmaps

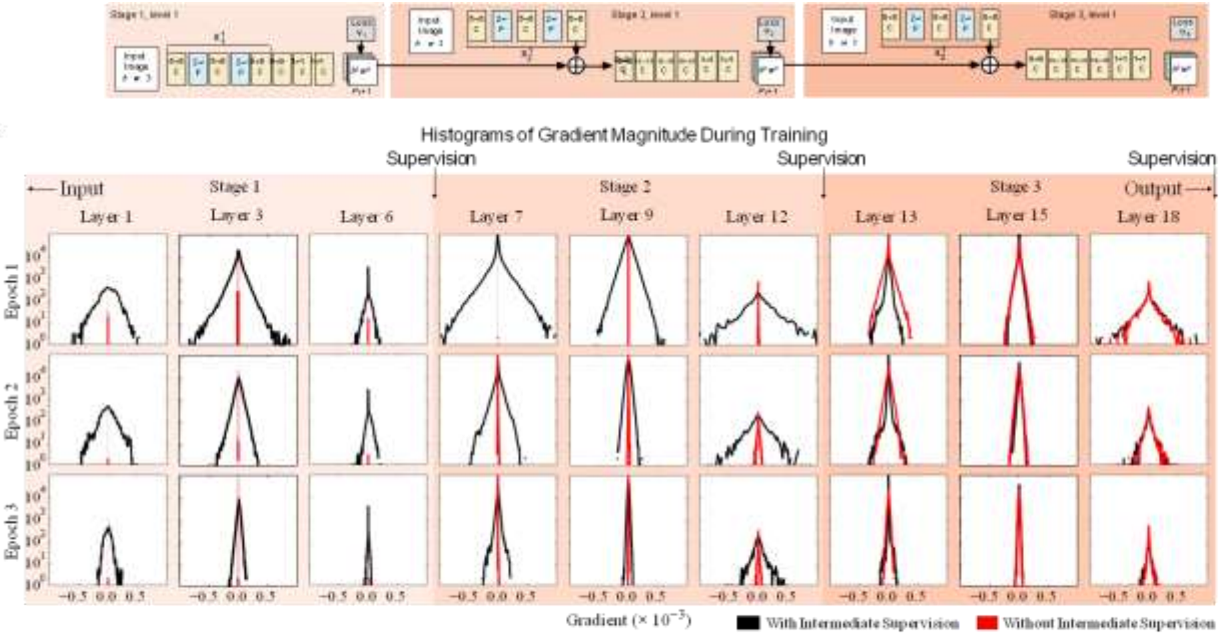


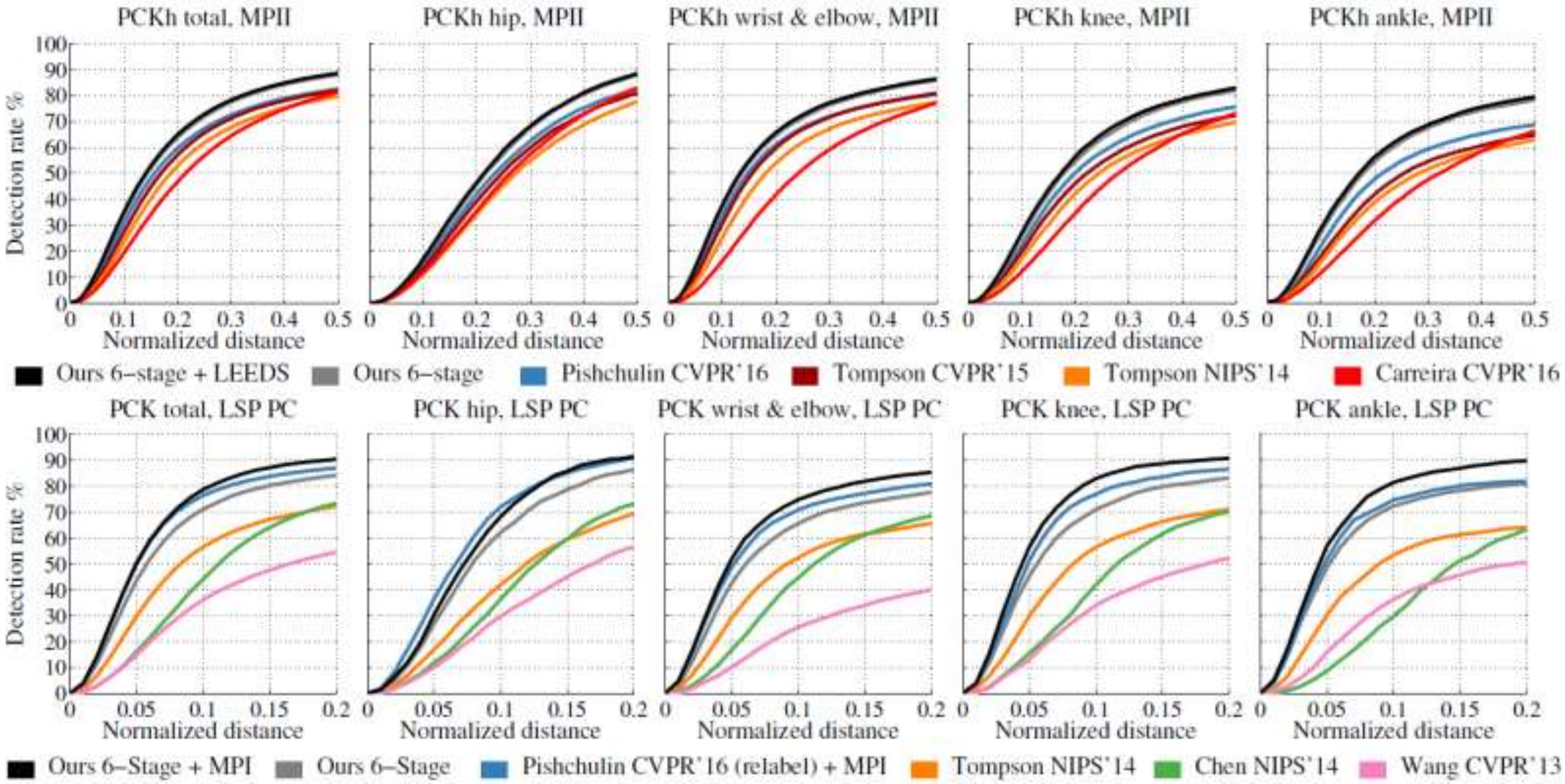
# Intermediate Supervision Addresss Vanishing Gradients




Loss: Euclidean distance



# Experiments



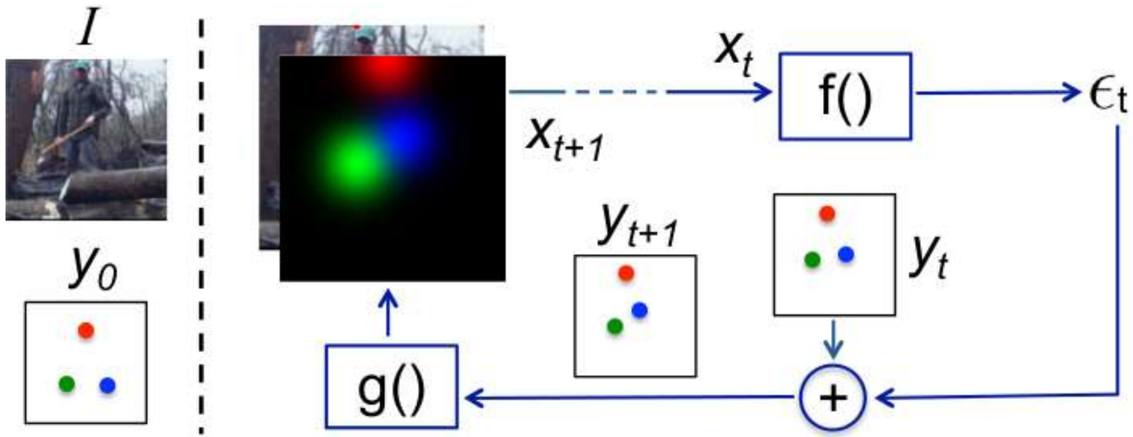
# Iterative Error Feedback

CVPR 2016

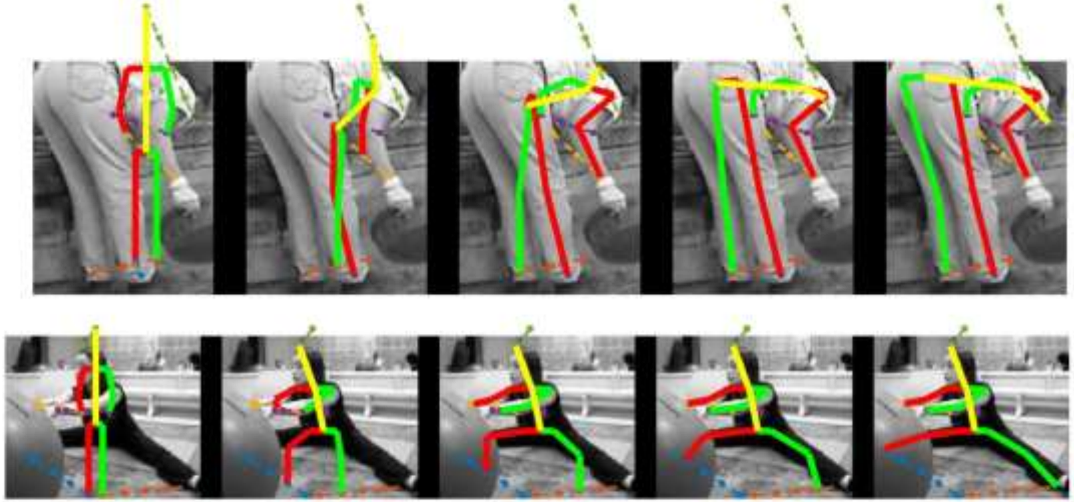
- ❖ Model dependencies in the output spaces
- ❖ Learning error feedback is easier than learning prediction directly
- ❖ Similar goal with Active Appearance Models (AAMs): end-to-end learning

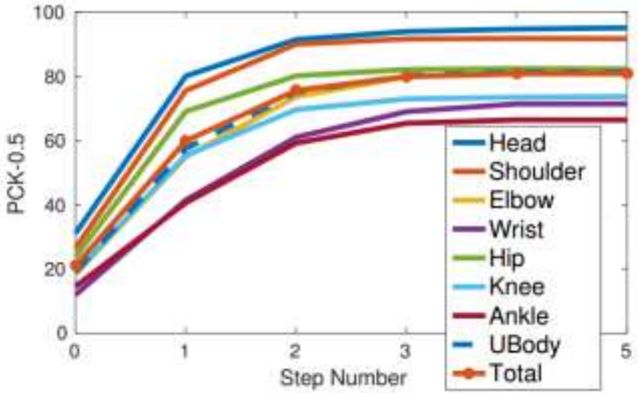
# Framework

$$\epsilon_t = f(x_t)$$
$$y_{t+1} = y_t + \epsilon_t$$
$$x_{t+1} = I \oplus g(y_{t+1}),$$



# Running Example





# Stacked Hourglass

Submitted to ECCV 16
Jia Deng's Group

Keypoints

- ❖ Repeated bottom-up, top-down
- ❖ Intermediate supervision

# Why is cascade not efficient enough?

Refinement of position within a local window could not offer much in the way of improvement for:

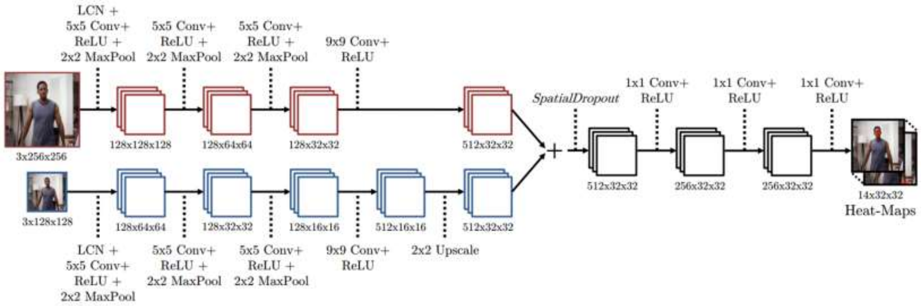
- occluded limb
- limb out of the window

We need to search over all scales of image

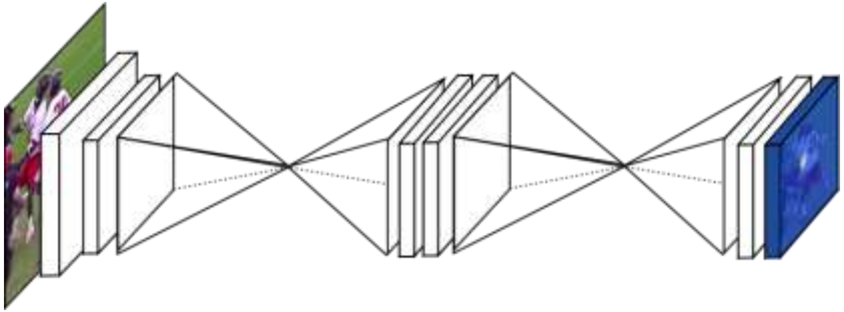
# Hourglass Design: Capture information at every scale

Review:

- ❑ Local appearance: essensial for part detection
- ❑ Global understanding: orientation of the body, limb arrangement, relationships between parts, ...

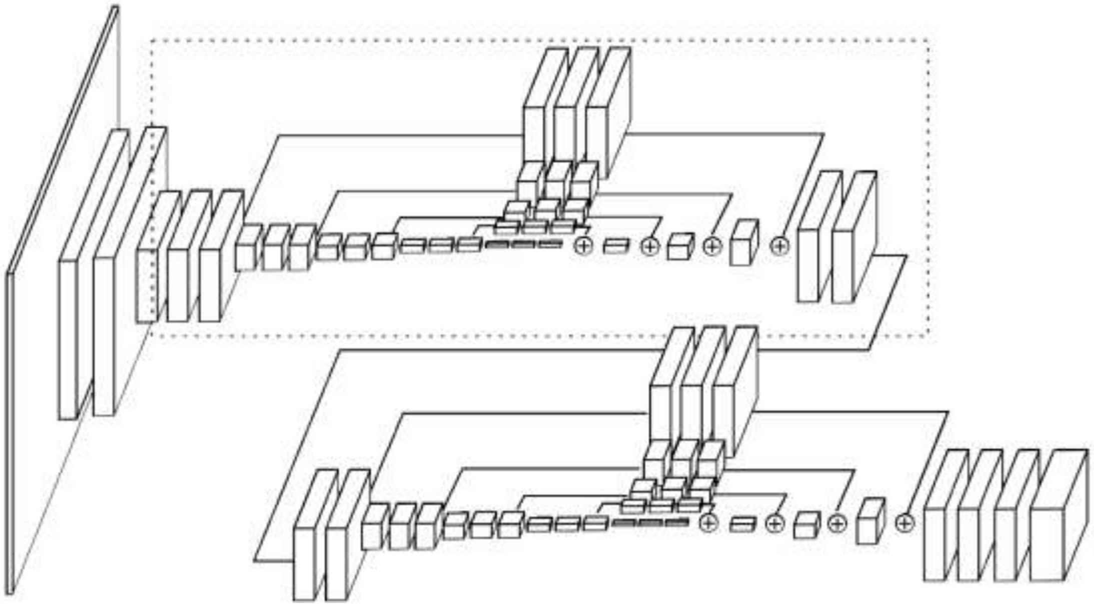



Previous work: Tompson et al. CVPR15
Combine features from multi-resolution image

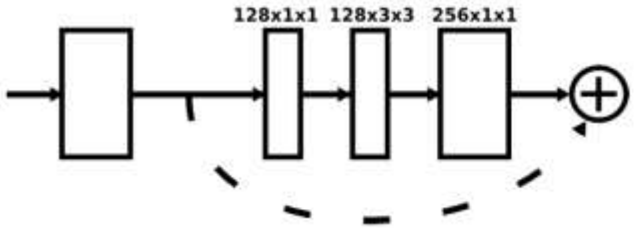


Single branch with bottom-up,top-down mechanism

# Hourglass Design



- ❖ Two stacked hourglass module
- ❖ Layers are identical across each module

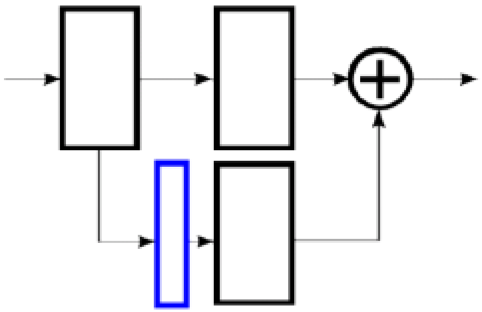


# Intermediate Supervision

❖ Generate predictions

➢ The network splits and produces a set of heatmaps (blue)

➢ The loss are applied on the prediction

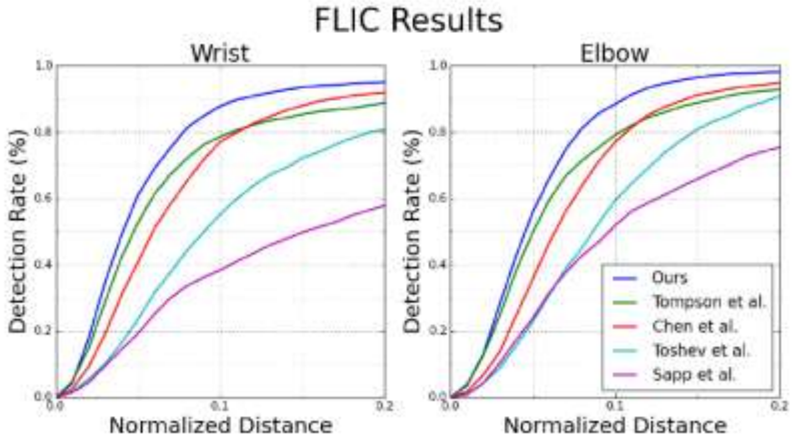


❖ A 1x1 convolution re-maps the heatmaps to match the number of channels of the intermediate features.

➢ These are added together before continuing forward.

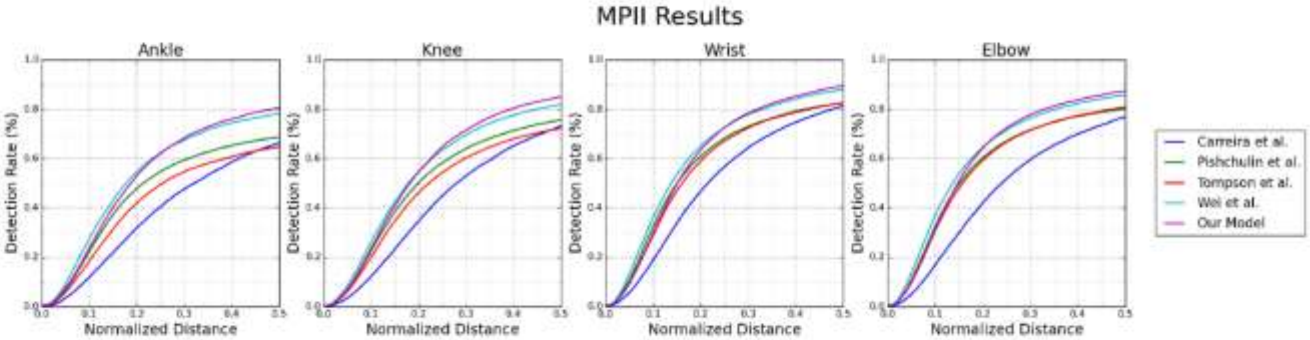
# Experiments

Results on FLIC dataset (PCK@0.2)




| | Elbow | Wrist |
|---|---|---|
| Tompson et al.[16] | 93.1 | 89.0 |
| Chen et al.[25] | 95.3 | 92.4 |
| Toshev et al.[24] | 92.3 | 82.0 |
| Sapp et al.[1] | 76.5 | 59.1 |
| Ours | **98.2** | **95.2** |

# Results: MPII dataset (PCKh@0.5)




| | Head | Shoulder | Elbow | Wrist | Hip | Knee | Ankle | Total |
|---|---|---|---|---|---|---|---|---|
| Tompson et al., NIPS'14 | 95.8 | 90.3 | 80.5 | 74.3 | 77.6 | 69.7 | 62.8 | 79.6 |
| Carreira et al., arXiV'15 | 95.7 | 91.7 | 81.7 | 72.4 | 82.8 | 73.2 | 66.4 | 81.3 |
| Tompson et al., CVPR'15 | 96.1 | 91.9 | 83.9 | 77.8 | 80.9 | 72.3 | 64.8 | 82.0 |
| Pishchulin et al., arXiV'15 | 94.1 | 90.2 | 83.4 | 77.3 | 82.6 | 75.7 | 68.6 | 82.4 |
| Wei et al., arXiV'16 | **97.7** | 94.5 | 88.3 | 83.4 | 87.9 | 81.9 | 78.3 | 87.9 |
| Our model | 97.6 | **95.4** | **90.0** | **85.2** | **88.7** | **85.0** | **80.6** | **89.4** |

# Component Analysis

Initial prediction vs. final prediction

| | Head | Shoulder | Elbow | Wrist | Hip | Knee | Ankle |
|---|---|---|---|---|---|---|---|
| Initial prediction | 96.7 | 92.9 | 85.0 | 79.8 | 83.7 | 78.6 | 74.3 |
| Final prediction | 97.7 | 94.6 | 88.5 | 88.3 | 87.5 | 83.0 | 79.0 |

# Summary

- Receptive field size is essential
    - larger receptive field incorporates more context information
- Process features at both local and global context is essential
    - Iterative Error Feedback
    - Convolutional Pose Machine
    - Stacked Hourglasses

# Other interesting papers


- Toshev, Alexander, and Christian Szegedy. "Deeppose: Human pose estimation via deep neural networks." CVPR 2014.
- Tompson, Jonathan J., et al. "Joint training of a convolutional network and a graphical model for human pose estimation." NIPS, 2014.
- Chen, Xianjie, and Alan L. Yuille. "Articulated pose estimation by a graphical model with image dependent pairwise relations." NIPS, 2014.
- Jain, Arjun, et al. "Learning human pose estimation features with convolutional networks." ICLR, 2014.
- Jain, Arjun, et al. "Modeep: A deep learning framework using motion features for human pose estimation." ACCV, 2014.
- Tompson, Jonathan, et al. "Efficient object localization using convolutional networks." CVPR. 2015.
- Fan, Xiaochuan, et al. "Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation." CVPR, 2015.
- Pfister, Tomas, James Charles, and Andrew Zisserman. "Flowing convnets for human pose estimation in videos." ICCV, 2015.
- Chu, Xiao, et al. "Structured feature learning for pose estimation." CVPR, 2016.
- Yang, Wei, et al. "End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation." CVPR, 2016.
- Gkioxari, Georgia, Alexander Toshev, and Navdeep Jaitly. "Chained Predictions Using Convolutional Neural Networks." ECCV,2016.