Human pose estimation
The task

- Mark joint locations for person
- Nose
- Right/left shoulder
- Right/left elbow
- Right/left hip
- ...
Two versions of task

- Assume people have been detected
- Rough bounding box given
- Key info available:
  - scale
  - only 1 location per joint
- Pros: disentangles detection and pose estimation
- Cons: unrealistic

- Tabula rasa without detections
- Challenge: no idea of scale or number
- Possible opportunity: use keypoint estimates to improve detections
- Pros: realistic
- Cons: conflates detection and pose estimation
Pose estimation given detection
Evaluation metric - given detection

- Evaluate every keypoint separately
- For each person, check if keypoint is correct
- Compute fraction of people for which keypoint is correct: PCK (Probability of Correct Keypoint)
Evaluation metric - given detection

\[ \frac{d}{h} < \alpha \]
R-CNN: Regions with CNN features

Input image
Extract region proposals (~2k / image)
Compute CNN features
Classify regions (linear SVM)

Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation

R. Girshick, J. Donahue, T. Darrell, J. Malik

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014

Slide credit: Ross Girshick
Bounding-box regression

Original:

- $w$
- $h$
- $(x, y)$

Predicted:

- $(\Delta x \times w + x, \Delta y \times h + h)$
- $\Delta w \times w + w$
- $\Delta h \times h + h$

Slide credit: Ross Girshick
Strategy 1: Regression

Right elbow x: 0.45
Right elbow y: 0.12
Left elbow x: 0.98
...

Strategy 1: Regression

• Assumes global object features has enough information for accurate localization
  • Localization info missing due to subsampling?
• Solution: Refinement!

Strategy 1: Regression

- Multimodal distributions?

Minimizer of

$$\mathbb{E}(\| X - x_{pred} \|^2)$$
Strategy 2: Heatmaps
Strategy 2: Heatmaps

• Still have the resolution issue
• Same solutions
  • Dilation?
  • Multiple layers?
  • Multiple image scales?
Heatmaps + Regression

Are all keypoints independent?
Are all keypoints independent?

• $\mathbf{l}$ is a candidate location for each keypoint

\[ E(\mathbf{l}) = \sum_{i} s_i(l_i) + \phi(\mathbf{l}) \]
Are all keypoints independent?

- \( \mathbf{l} \) is a candidate location for each keypoint

\[
E(l) = \sum_i s_i(l_i) + \sum_{ij} \phi_{ij}(l_i, l_j)
\]
Joint prediction of keypoints

\[ l^* = \operatorname{arg\,min} \ E(l) \]

- Conditional Random Field
- But not just smoothness: \( \phi \) is unknown!
- Needs to be learnt
Pictorial structures
Flexible Mixture of Parts

\[ S(I, L) = \sum_{i \in V} \alpha_i \cdot \phi(I, l_i) + \sum_{ij \in E} \beta_{ij} \cdot \psi(l_i, l_j) \]

- \( \psi(l_i, l_j) \): Spatial features between \( l_i \) and \( l_j \)
- \( \beta_{ij} \): Pairwise springs between part \( i \) and part \( j \)

Flexible Mixture of Parts

• Learning?
• Structured SVMs
  • Very large output spaces
  • A scoring function that scores input-output pairs $h_w(x, y)$
  • Predicted output is $\arg \max$ of scoring function
  • Loss is $\max(0, 1 + \max_{y \neq y^*} h_w(x, y) - h_w(x, y^*))$
Inference?

\[ E(l) = \sum_i s_i(l_i) + \sum_{ij} \phi_{ij}(l_i, l_j) \]

\[ l^* = \arg \min E(l) \]

\[ \min_l \sum_i s_i(l_i) + \sum_{ij} \phi_{ij}(l_i, l_j) \]

\[ l^*_i = \arg \min_{l_i} s_i(l_i) + \sum_j \phi_{ij}(l_i, l^*_j) \]

\[ l_i^{(t+1)} \leftarrow \arg \min_{l_i} s_i(l_i) + \sum_j \phi_{ij}(l_i, l_j^{(t)}) \]
Combining learning and inference

- Inference in MRFs and CRFs usually iterative and approximate

\[ l_i^{(t+1)} \leftarrow \arg\min_{l_i} s_i(l_i) + \sum_j \phi_{ij}(l_i, l_j^{(t)}) \]

- Except trees: FMP
- Instead of learning scoring function, then *approximately* minimizing it
- Learn iterative inference procedure?
Combining learning and inference

• In each iteration, beliefs of one variable are updated using current beliefs of the others
  \[ l_i^{(t+1)} \leftarrow \arg \min_{l_i} s_i(l_i) + \sum_j \phi_{ij}(l_i, l_j^{(t)}) \]

• Frame each iteration of inference as a differentiable function
• Write inference as a convolutional network

Combining learning and inference

\[ P(\text{eye at } p) = \sum_q P(\text{eye at } p \mid \text{nose at } q) \cdot P(\text{nose at } q) \]

\[ f(p) = \sum_q w(p, q) \cdot g(q) \]

\[ f = w \ast g \]
Combining learning and inference

Autocontext and Inference Machines

• Instead of learning model that does iterative improvement
• Learn model that does refinement
Autocontext and Inference Machines

- Shared parameters: *Inference Machines*
- Unshared parameters: *Autocontext*

Inference machines

Stacked Hourglass Networks

• Each refinement round has to
  • Combine global information about pose
  • Use global pose information to produce new precise pose estimate

• “Hourglass structure”
Stacked hourglass networks

MPII Results
Pose estimation without detection
Evaluation metric - tabula rasa

• Algorithm detects keypoints + scores
• Match keypoint to a ground truth keypoint if d/h is less than threshold
• Compute precision-recall curve
• Compute AP (called APK : AP Keypoint)
Two strategies

• First detect, then estimate keypoints
  • Can use any of previous techniques
  • Similar to instance segmentation
  • Easy to get object level information
  • Hard to recover from bad detections
  • e.g. Mask R-CNN

• Detect keypoints, then group into people
  • Need a way to group keypoints: hard problem, requires heuristics
  • No simple way to have object level information
Top-down keypoint detection

Bottom-up keypoint detection

Pose estimation in 3D

Pose estimation in 3D

• Key idea: know relative lengths of each limb

• Assume *scaled orthographic projection*
  • Valid when variation in depth much smaller than depth

\[
x = \frac{X}{Z} \approx \frac{X}{Z_0}
\]
\[
y = \frac{Y}{Z} \approx \frac{Y}{Z_0}
\]

Pose estimation in 3D

\[ l^2 = (X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2 \]

\[ (u_1 - u_2) = s(X_1 - X_2) \]

\[ (v_1 - v_2) = s(Y_1 - Y_2) \]

\[ dZ = (Z_1 - Z_2) \]

\[ \Rightarrow dZ = \sqrt{l^2 - ((u_1 - u_2)^2 + (v_1 - v_2)^2)/s^2} \]
Pose estimation for rigid objects
Pose estimation for rigid objects

Pose estimation for rigid objects

Cyclo-rotation / roll

Elevation / pitch

Azimuth / yaw
Viewpoint-conditioned pose

Viewpoint-conditioned pose

Fitting viewpoint to keypoints

• Idea: minimize reprojection error

\[
\mathbf{p}_i = \mathbf{K}[\mathbf{R} | \mathbf{t}] \mathbf{P}_i
\]

\[
x_i = \frac{\mathbf{p}_i[0]}{\mathbf{p}_i[2]} \quad \quad y_i = \frac{\mathbf{p}_i[1]}{\mathbf{p}_i[2]}
\]

\[
\min_{\mathbf{R}, \mathbf{t}} \sum_i (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2
\]
Shape models
Shape models
Shape models

\[ \text{N} \quad 3p \quad \text{X} \quad \text{N} \quad \equiv \quad \text{N} \quad c \quad \text{A} \quad \text{N} \quad \equiv \quad \text{N} \quad c \quad \text{B} \quad 3p \]
Shape models

\[ P_i = \sum_j \alpha_j B_{ij} \]

\[ \tilde{p}_i = K[R|t]\tilde{P}_i \]

\[ x_i = \tilde{p}_i[0]/\tilde{p}_i[2] \quad \quad \quad y_i = \tilde{p}_i[1]/\tilde{p}_i[2] \]

\[ \min_{R,t,\alpha} \sum_i (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \]
Fitting viewpoints to keypoints