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

$\text{original}$

$\text{predicted}$

$(x, y)$

$\Delta w \times w + w$

$\Delta h \times h + h$

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

Slide credit: Ross Girshick
Strategy 1: Regression

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?
Strategy 2: Heatmaps
Strategy 2: Heatmaps - Training

• Each keypoint is a separate binary heatmap

• Keypoint location is positive, all other locations are negative. Options:
  • *Softmax* over all locations in an image
    • $p(x, y) = \frac{e^{s(x,y)}}{\sum_{x', y'} e^{s(x', y')}}$
  • *Sigmoid* at each location
    • $p(x, y) = \frac{1}{1 + e^{-s(x,y)}}$
Strategy 2: Heatmaps

• Still have the resolution issue

• Same solutions
  • Dilation?
  • Multiple layers?
  • Multiple image scales?
Heatmaps + Regression

• Use heatmap to predict coarse location
• Also predict at each coarse location an offset ($\Delta x, \Delta y$).
Are all keypoints independent given the image?

Equally likely locations for right elbow

Equally likely locations for right wrist
Are all keypoints independent given the image?
Capturing keypoint dependence!

- **Structured prediction**
- \( l \) is a candidate location for each keypoint

\[
E(l) = \sum_i s_i(l_i) + \phi(l)
\]

- Score from heatmap
- Consistency between joints
Are all keypoints independent?

• $l$ 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

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

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

- $\psi(l_i, l_j)$: 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 margin rescaled loss

Inference?

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

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

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

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

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

• 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 \textit{approximately} minimizing it

• Learn iterative inference procedure?
  • Similar to autocontext/inference machines
Autocontext and Inference Machines

Image → Pose

Image, Current Pose estimate → New Pose estimate
Autocontext and Inference Machines

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


Iterative models

• 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

Iterative models

- \( P(\text{eye at } p) = \sum_q P(\text{eye at } p \mid \text{nose at } q) P(\text{nose at } q) \)
- \( P(\text{eye at } p \mid \text{nose at } q) \) only depends on relative location of \( p \) and \( q \)
- \( f(p) = \sum_q w(p - q) g(q) \) : convolution!
- \( f = w * g \)
Iterative models

More iterative models

- Why only one convolution?
- Each iteration can involve multiple convolution/subsampling layers over beliefs from previous iteration
Stacked Hourglass Networks

• Each refinement round has to
  • Combine global information about pose
  • Use global pose information to produce new precise pose estimate
• Rounds need not share parameters
• ”Hourglass structure”
Stacked hourglass networks
Pose estimation without detection
Evaluation metric - tabula rasa

• Algorithm detects keypoints + scores
• Match keypoint to a ground truth keypoint if \( \frac{d}{h} \) 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

• Need to group keypoints. Can be really ambiguous
• Idea: detect not just keypoints but also limbs + limb orientation
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

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

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

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

• Problem: we need 3D location of each keypoint in canonical frame
  • E.g., CAD model
• If we have a set of CAD models, then we can try each one
Shape models
Shape models: shape basis

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

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

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

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

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