CSE 252C: Advanced Computer Vision
Manmohan Chandraker

Lecture 12: Human Pose Estimation 2
Virtual classrooms

• Virtual lectures on Zoom
  – Only host shares the screen
  – Keep video off and microphone muted
  – But please do speak up (remember to unmute!)

• Virtual interactions on Zoom
  – Ask and answer plenty of questions
  – “Raise hand” feature on Zoom when you wish to speak
  – Post questions on chat window
  – TA will help keep track of raised hands and chat window

• Lectures recorded and upload on Kaltura
  – Available under “Media Gallery” on Canvas
Course details

• Class webpage:

• Instructor email:
  – mkchandraker@eng.ucsd.edu

• TAs: Zhengqin Li and You-Yi Jau
  – Emails: zhl378@eng.ucsd.edu and yjau@eng.ucsd.edu

• Aim is to learn together, discuss and have fun!

• Homework 2 released
  – Due Thu, May 21 at 4pm
  – Data access issues hopefully resolved
  – Face recognition training takes time, get started early!
“Lightning” Presentations

What is your preference for student presentations?

- 58.6%: As now: each student gives a 5 minute presentation
- 27.6%: Change: 2 students collaborate to give a 5-minute presentation
- 13.8%: Change: 2 students collaborate to give a 10-minute presentation

- A suggestion that we want to try:
  - Students record the presentation (with speech)
  - Post the question they ask beforehand
Recap
Local and Global Needed for Pose Estimation

- Local part-based detectors are not sufficient for pose estimation

- Local appearance is insufficient
- Global appearance is helpful
Deep Networks for Holistic Pose Estimation

• Advantages:
  • Simple, yet holistic
  • No need to define losses that capture interactions
  • Instead, all hidden layers are shared by joint regressors

• Disadvantages:
  • Limited ability to consider details

[Toshev and Szegedy, DeepPose]
Cascade of Regressors

- Advantages:
  - Simple, yet holistic
  - No need to define losses that capture interactions
  - Instead, all hidden layers are shared by joint regressors
  - Increasingly detailed prediction along cascade stages

- Disadvantages:
  - Limited ability to consider details
  - One prediction per image, no candidates
  - Depends on quality of initial prediction

[ Toshev and Szegedy, DeepPose ]
Cascade of Heat Maps

• Advantages:
  • Simple, yet holistic
  • No need to define losses that capture interactions
  • Instead, all hidden layers are shared by joint regressors
  • Increasingly detailed prediction along cascade stages
  • Encodes probability of joint appearing at a pixel

• Disadvantages:
  • Limited ability to consider details
  • One prediction per image, no candidates
  • Depends on quality of initial prediction
  • Not enough modeling of spatial structure
  • Cannot reason about occluded parts, or those outside window

[Tompson et al., Efficient Object Localization]
Spatial Structure Helps Localization

- Shoulder, neck and head localization correct error in location for elbow
Spatial Structure Helps Localization

- Goal: achieve large receptive fields to learn complex and long-range interactions
**Convolutional Pose Machines**

- **Goal:** achieve large receptive fields to learn complex and long-range interactions
- **Each stage sees two inputs**
  - Image features
  - Context features based on output of previous stage

---

[Wei et al., Convolutional Pose Machines]
Convolutional Pose Machines: Stage 1

- Predict part belief largely based on local image values
- Output P+1 heat maps (P parts and 1 background)
- Small receptive field
  - Captures relation between head and shoulders, but not head and knees
Convolutional Pose Machines: Next Stages

- Image features $x'$ from the previous stage
- Context function $\psi$ encodes landscape of belief maps around part locations
  - In practice, $\psi$ is just the receptive field
  - Network decides how to combine features and learn higher relations
  - Previously: hand-defined potential functions in graphical model

- Three ways to increase size of receptive field
  - More pooling: lose local details
  - Larger filters: increase number of parameters
  - More layers: vanishing gradients
Convolutional Pose Machines: Gradients

- Magnitude of backpropagated gradients decreases rapidly in initial layers
- Use intermediate supervision to ensure greater variance in gradients

\[
f_t = \sum_{p=1}^{P+1} \sum_{z \in \mathcal{Z}} \| b^p_t(z) - b^p_\star(z) \|_2^2, \quad \mathcal{F} = \sum_{t=1}^{T} f_t.
\]

Histograms of gradient magnitude during training.
Hourglass Network

- Local appearance: needed for accurate part detection
- Global reasoning: orientation of body, limb arrangement, part relationships
- Simple design to process multiple scales and achieve pixel-wise predictions
Hourglass Network

- Local appearance: needed for accurate part detection
- Global reasoning: orientation of body, limb arrangement, part relationships
- Simple design to process multiple scales and achieve pixel-wise predictions
- Bottom-up reasoning: Convolution and max-pooling to very low 4x4 resolution
  - Can apply small filters to learn global relationships
- Top-down reasoning: Upsample and combine with skip connection
  - Preserve spatial information at each resolution
Stacked Hourglass Network

- Multiple iterative stages allow refinement
- Each stage does full bottom-up and top-down processing (no weights shared)
Stacked Hourglass Network

- Multiple iterative stages allow refinement
- Each stage does full bottom-up and top-down processing (no weights shared)
- Repeated inference, with intermediate supervision for each stage
  - Network has had a chance to reason both locally and globally
  - Subsequent hourglass modules can reassess high-order spatial relations
  - Ask network to repeatedly reason across scales
  - 1x1 convolution to add intermediate heatmaps to feature channels
Stacked Hourglass Network

- Multiple iterative stages allow refinement
- Each stage does full bottom-up and top-down processing (no weights shared)
- Repeated inference, with intermediate supervision for each stage
  - Network has had a chance to reason both locally and globally
  - Subsequent hourglass modules can reassess high-order spatial relations
  - Ask network to repeatedly reason across scales
  - 1x1 convolution to add intermediate heatmaps to feature channels
- Intermediate supervision not straightforward for single hourglass module
  - Cannot apply before pooling since only local information available
  - High-order features only present at low resolutions
  - After upsampling, have high-order information and higher resolution
  - But cannot re-evaluate features globally relative to each other
Stacked Hourglass Network

- Depth helps: single long hourglass better than short hourglass
- Cascading better than just depth: stacked hourglass better than long hourglass
- Intermediate supervision helps both single and stacked models
Evaluation of Human Pose Estimation

- Percentage of Correct Parts
  - A part (limb) is correct if sum of its joint errors is less than half limb-length
  - Penalizes shorter limbs since they have smaller thresholds
Evaluation of Human Pose Estimation

• Percentage of Correct Parts
  • A part (limb) is correct if sum of its joint errors is less than half limb-length
  • Penalizes shorter limbs since they have smaller thresholds

• Percentage of Correct Keypoints
  • A predicted joint is correct if it lies within a threshold of true location
  • Threshold: 0.2 of torso length, or 0.5 of head bone length
  • Usually shorter limbs have smaller torsos, so does not penalize them
Evaluation of Human Pose Estimation

- **Percentage of Correct Parts**
  - A part (limb) is correct if sum of its joint errors is less than half limb-length.
  - Penalizes shorter limbs since they have smaller thresholds.

- **Percentage of Correct Keypoints**
  - A predicted joint is correct if it lies within a threshold of true location.
  - Threshold: 0.2 of torso length, or 0.5 of head bone length.
  - Usually shorter limbs have smaller torsos, so does not penalize them.

- **Object Keypoint Similarity**
  - Define OKS = \( \frac{\sum_i \exp \left( -\frac{d_i^2}{2s^2k_i^2} \right) \delta (v_i > 0)}{\sum_i \delta (v_i > 0)} \)
  - \( d_i \): distance between predicted and ground truth keypoint.
  - \( v_i \): visibility of keypoint \( i \).
  - \( s \): scale of object.
  - OKS plays similar role as IoU in detection.
  - AP50: average precision at OKS = 50.
  - mAP: mean of AP at OKS = [0.50, 0.55, ..., 0.90, 0.95]
Evaluation of Human Pose Estimation

- Average response from flipped inputs
- Keypoint location quarter offset from highest response to second highest

Convolutional Pose Machines
Recipes for Human Pose Estimation

- Local appearance: needed for accurate part detection
- Global reasoning: orientation of body, limb arrangement, part relationships
Recipes for Human Pose Estimation

- Local appearance: needed for accurate part detection
- Global reasoning: orientation of body, limb arrangement, part relationships
- CNNs are inherently well-suited for this task
  - Multiscale feature extraction
  - Shared hidden layers capture part interactions
- But mechanisms needed to coax local and global performance out of CNNs
Recipes for Human Pose Estimation

- Local appearance: needed for accurate part detection
- Global reasoning: orientation of body, limb arrangement, part relationships

- CNNs are inherently well-suited for this task
  - Multiscale feature extraction
  - Shared hidden layers capture part interactions

- But mechanisms needed to coax local and global performance out of CNNs

- Heat maps
  - Predict probability of joint appearing at a pixel

- Cascades
  - Iteratively refine predictions for fine localization
  - Long-range interactions through wider receptive fields

- Intermediate supervision
  - Prevent vanishing gradients through cascade stages
  - Allow cascade to repeatedly assess local and global information
Human Pose: 2D and 3D

- Human pose estimation in 2D works quite well
Human Pose: 2D and 3D

- Human pose estimation in 2D works quite well
- But that might not be sufficient
  - Hard to tell 3D structure from 2D joint locations
Human Pose: 2D and 3D

- Human pose estimation in 2D works quite well
- But that might not be sufficient
  - Hard to tell 3D structure from 2D joint locations
- Many applications might require 3D joint locations
Human Pose: 2D and 3D

- Human pose estimation in 2D works quite well
- But that might not be sufficient
  - Hard to tell 3D structure from 2D joint locations
- Many applications might require 3D joint locations
- Some applications might even require full 3D human shape
Estimating 3D Pose: Challenges

- Pose annotation is easy in 2D, hard in 3D
  - Large-scale 2D pose data, but limited 3D pose data

• Captured in control-environment with accurate sensors.
Estimating 3D Pose: Challenges

- Pose annotation is easy in 2D, hard in 3D
  - Large-scale 2D pose data, but limited 3D pose data
- Need to estimate greater number of parameters
- Occlusions, large appearance variations, ....
- Greater bar on realism for 3D outputs
Human 3D Pose: Challenges

- Do 2D pose estimation, then lift solution to 3D using optimization
Human 3D Pose: Challenges

• Do 2D pose estimation, then lift solution to 3D using optimization
• Suffers from partial view ambiguity
  • Multiple 3D solutions can explain 2D observations
Use a Morphable Model for Human Shapes

- Use a model-based approach, similar to 3DMM for faces
Use a Morphable Model for Human Shapes

- Use a model-based approach, similar to 3DMM for faces
- Variations more complex for human bodies
Skinned Multi-Person Linear Model (SMPL)

- Blend skinning (skeleton subspace deformation)
  - Attach surface of mesh to underlying skeletal structure
  - Vertex on mesh surface transformed using weights from nearby bones
  - Typically, weights are linear, represented by $|\text{vertices}| \times |\text{joints}|$ matrix

Mean template shape $\bar{T} \in \mathbb{R}^{3N}$

Blend weights $\mathbb{R}^{N \times K}$

Shape function $\mathbb{R}^{|\bar{b}|} \rightarrow \mathbb{R}^{3K}$

CSE 252C, SP20: Manmohan Chandraker

[Looper et al., SMPL]
Skinned Multi-Person Linear Model (SMPL)

- \( N = 6890 \) vertices, \( K = 23 \) joints, \( |\tilde{\beta}| = 10 \) shape parameters
- Pose encoded by orientation of each part with respect to parent in tree
- \( |\tilde{\theta}| = 3 \times 23 + 3 = 72 \) pose parameters (orientation of each part and root)

- Shape function
  \[
  B_S(\tilde{\beta}; S) = \sum_{n=1}^{N} \beta_n S_n , \text{ where } S = [S_1, \ldots, S_{|\beta|}] \in \mathbb{R}^{3N \times |\beta|} \text{ is shape basis}
  \]

- Pose function for deformation of vertex from template
  \[
  B_P(\tilde{\theta}; \mathcal{P}) = \sum_{n=1}^{9K} (R_n(\tilde{\theta}) - R_n(\tilde{\theta}^*)) P_n , \text{ where } R(\tilde{\theta}) \text{ is pose as rotation matrix,}
  \]
  \[
  \mathcal{P} = [P_1, \ldots, P_{9K}] \in \mathbb{R}^{3N \times 9K} \text{ is per-vertex deformation basis}
  \]

- Overall model for SMPL:
  \[
  M(\tilde{\beta}, \tilde{\theta}) = W(T_P(\tilde{\beta}, \tilde{\theta}), J(\tilde{\beta}), \tilde{\theta}, \mathcal{W})
  
  T_P(\tilde{\beta}, \tilde{\theta}) = \tilde{T} + B_S(\tilde{\beta}) + B_P(\tilde{\theta})
  \]

[Lober et al., SMPL]
SMPL: Multi-Shape Data
SMPL: Multi-Pose Data
Shape Fitting Pipeline

Training

Shape Database

Model
Parametric 3D body model (SCAPE)

Single image

Initialization
pose
GrabCut

Optimization
silhouette
edges
shading

Applications
Animation Caricature Measurement

Output
3D pose and shape model

CSE 252C, SP20: Manmohan Chandraker
Predicting 3D Human Shape and Pose

• Use data-driven priors from deep network to avoid optimization difficulties
• Large-scale data exists for 2D keypoints and silhouettes
• Train a multi-task stacked hourglass network for heat maps and silhouette

[Figures and diagrams related to the prediction of 3D human shape and pose]
Predicting 3D Human Shape and Pose

- Use heat maps to predict pose, silhouette to predict shape parameters in SMPL
- Ignore inter-dependence, in favor of disentanglement
- Train based on data rendered from ground truth SMPL instances
- Objectives: 3D vertex loss and joint loss

\[ \mathcal{L}_M = \sum_{i=1}^{N} \| \hat{P}_i - P_i \|_2^2, \quad \mathcal{L}_J = \sum_{i=1}^{M} \| \hat{J}_i - J_i \|_2^2. \]

[Image of a diagram showing the process of predicting 3D human shape and pose, including heatmaps, pose priors, shape priors, and rendered images.]
Predicting 3D Human Shape and Pose

- Fine-tune end-to-end using differentiable rendering
- Project vertex estimates to obtain silhouette, joint estimates for keypoints

\[ \Pi(\hat{P}) = \hat{S}, \quad \Pi(\hat{J}) = \hat{W} \in \mathbb{R}^{M \times 2}. \]

- Supervised loss to minimize reprojection error:

\[ \mathcal{L}_\Pi = \mu \sum_{i} \left( \| \hat{W}_i - W_i \|_2^2 + \| \hat{S} - S \|_2^2 \right) \]
Predicting 3D Human Shape and Pose
Object 3D Pose Estimation

**Synthetic Training Data**

- Render CAD models for car category from different views
- Random images for background
- Manually annotated 3D keypoints for each CAD model
- Pose, visibility, 3D keypoints and 2D keypoints become available from renderer
Deep Supervision with Intermediate Concepts

- **Standard CNN training objective:**
  \[ W^* = \arg\min_W \sum_{(x,y) \in \mathcal{Z}} l(y, f(x, W)) \]
  - **Overfitting:** poor generalization to new population (real instead of synthetic)

- **Introduce regularization that biases solutions to better \( W \)**
  - Reproduce physical quantities causally related to the desired output
  - CNNs can capture increasingly complex concepts in deeper layers
  - Enforce hidden layers to reach a sequence of known intermediate concepts
  - Can derive supervision for intermediate concepts with synthetic rendering
Deep Supervision with Intermediate Concepts

CSE 252C, SP20: Manmohan Chandraker
3D Keypoints and Instance Segmentation

Chair 3D-INN DISCO

Sofa 3D-INN DISCO

CSE 252C, SP20: Manmohan Chandraker
Suppose a 2-layer network: \( y = \sigma(w_2 \sigma(w_1 x + b_1) + b_2) \) where \( \sigma \) is ReLU.

Suppose the true model is \( (w_1, w_2, b_1, b_2) = (3, -1, -2, -7) \)

We have training data \( S=\{(x,y)\} = \{(1,0), (2,0), (3,0)\} \)

A learning algorithm may obtain \( (w_1, w_2, b_1, b_2) = (1, 3, -1, -10) \). It fails to generalize when \( x=4, 5 \).

If we have an auxiliary cue that tells us the value of intermediate output \( y' = \sigma(w_1 x + b_1) \)

For example, we have an augmented training data \( S=\{(x, y', y)\} = \{(1,0,0), (2,0,0), (3,1,0)\} \), the incorrect model above can be removed.