Announcements!

- HW #3 misunderstandings
  - Deadline for HW #3 re-try is next Thursday April 6
- Final Project milestones due next Tuesday April 4
- Vote for Final Day and Location
Decorators

How to write a decorator:

```python
import time
def timeit(method):
    def timed(*args, **kw):
        ts = time.time()
        result = method(*args, **kw)
        te = time.time()
        print '%r (%r, %r) %2.2f sec' %\n             (method.__name__, args, kw, te-ts)
        return result
    return timed
```

How to use a decorator:

```python
class Foo(object):
    @timeit
def foo(self, a=2, b=3):
        time.sleep(0.2)
    @timeit
def f1():
        time.sleep(1)
        print 'f1'
```
Grouping in vision

• Goals:
  – Gather features that belong together
  – Obtain an intermediate representation that compactly describes key image (video) parts

• Top down vs. bottom up segmentation
  – Top down: pixels belong together because they are from the same object
  – Bottom up: pixels belong together because they look similar

• Hard to measure success
  – What is interesting depends on the app.
Examples of grouping in vision

Determine image regions

Group video frames into shots

Object-level grouping

Figure-ground
Edge and line detection

• Canny edge detector = smooth → derivative → thin → threshold → link

• Generalized Hough transform = points vote for shape parameters

• Straight line detector = canny + gradient orientations → orientation binning → linking → check for straightness
Segmentation

- Inspiration from human perception
  - Gestalt properties
- MRFs
- Segmentation with Graph Cuts
Computer Vision Tasks

Classification

Classification + Localization

Object Detection

Segmentation

CAT, DOG, DUCK

Single object

Multiple objects

CAT

CAT

CAT, DOG, DUCK

CAT, DOG, DUCK

* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n

comp150dl
Computer Vision Tasks

Classification

Classification + Localization

Object Detection

Segmentation

Today
These intensities define the three groups.
We could label every pixel in the image according to which of these primary intensities it is.
• i.e., segment the image based on the intensity feature.
• What if the image isn’t quite so simple?
• Now how to determine the three main intensities that define our groups?
• We need to *cluster.*
Clustering

• With this objective, it is a “chicken and egg” problem:
  – If we knew the cluster centers, we could allocate points to groups by assigning each to its closest center.
  – If we knew the group memberships, we could get the centers by computing the mean per group.
Smoothing out cluster assignments

- Assigning a cluster label per pixel may yield outliers:

- How to ensure they are spatially smooth?

Kristen Grauman
Solution

Encode dependencies between pixels

\[ P(y; \theta, data) = \frac{1}{Z} \prod_{i=1..N} f_1(y_i; \theta, data) \prod_{i,j \in \text{edges}} f_2(y_i, y_j; \theta, data) \]

Labels to be predicted
Individual predictions
Pairwise predictions

Normalizing constant
Writing Likelihood as an “Energy”

\[
P(y; \theta, data) = \frac{1}{Z} \prod_{i=1..N} p_1(y_i; \theta, data) \prod_{i,j \in \text{edges}} p_2(y_i, y_j; \theta, data)
\]

\[
\text{Energy}(y; \theta, data) = \sum_{i} \psi_1(y_i; \theta, data) + \sum_{i,j \in \text{edges}} \psi_2(y_i, y_j; \theta, data)
\]

“Cost” of assignment \( y_i \)

“Cost” of pairwise assignment \( y_i, y_j \)
Markov Random Fields

Cost to assign a label to each pixel

Cost to assign a pair of labels to connected pixels

\[
\text{Energy}(y; \theta, \text{data}) = \sum_{i} \psi_1(y_i; \theta, \text{data}) + \sum_{i, j \in \text{edges}} \psi_2(y_i, y_j; \theta, \text{data})
\]
Markov Random Fields

• Example: “label smoothing” grid

\[
\begin{array}{c|c|c}
0 & 1 & \\
0 & 0 & K \\
1 & K & 0 \\
\end{array}
\]

Unary potential
0: \(-\log P(y_i = 0 \mid data)\)
1: \(-\log P(y_i = 1 \mid data)\)

Pairwise Potential

\[
\text{Energy}(y; \theta, \text{data}) = \sum_{i} \psi_1(y_i; \theta, \text{data}) + \sum_{i,j \in \text{edges}} \psi_2(y_i, y_j; \theta, \text{data})
\]

Slide: Derek Hoiem
Solving MRFs with graph cuts

\[ \text{Energy}(y; \theta, \text{data}) = \sum_i \psi_1(y_i; \theta, \text{data}) + \sum_{i,j \in \text{edges}} \psi_2(y_i, y_j; \theta, \text{data}) \]
Solving MRFs with graph cuts

\[
\text{Energy}(y; \theta, \text{data}) = \sum_{i} \psi_1(y_i; \theta, \text{data}) + \sum_{i, j \text{edges}} \psi_2(y_i, y_j; \theta, \text{data})
\]
GrabCut segmentation

User provides rough indication of foreground region.

Goal: Automatically provide a pixel-level segmentation.
What is easy or hard about these cases for graphcut-based segmentation?
Easier examples
More difficult Examples

Camouflage & Low Contrast

Initial Rectangle

Initial Result

Fine structure

Harder Case

GrabCut – Interactive Foreground Extraction
Using graph cuts for recognition

TextonBoost (Shotton et al. 2009 IJCV)
Unsupervised Segmentation

- Classic Example: **Superpixels**

- Cluster pixels with their neighbors

- Break clusters when large gradient occurs between neighboring pixels

- Figure from Achanta et al. — “SLIC Superpixels Compared to State-of-the-art Superpixel Methods,” May 2012
Further reading and resources

• Graph cuts
  – Classic paper: *What Energy Functions can be Minimized via Graph Cuts?* (Kolmogorov and Zabih, ECCV '02/PAMI '04)

• Belief propagation

• Normalized cuts and image segmentation (Shi and Malik)

• N-cut implementation
  [http://www.seas.upenn.edu/~timothee/software/ncut/ncut.html](http://www.seas.upenn.edu/~timothee/software/ncut/ncut.html)
Semantic Segmentation

Label every pixel!

Don’t differentiate instances (cows)

Figure credit: Shotton et al, “TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context”, IJCV 2007

* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Instance Segmentation

Detect instances, give category, label pixels

“simultaneous detection and segmentation” (SDS)

Lots of recent work (MS-COCO Challenges)

Figure credit: Dai et al, "Instance-aware Semantic Segmentation via Multi-task Network Cascades", arXiv 2015
Semantic Segmentation
Semantic Segmentation
Semantic Segmentation

Extract patch
Semantic Segmentation

Extract patch

Run through a CNN

* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Semantic Segmentation

Extract patch → Run through a CNN → Classify center pixel

* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Semantic Segmentation

Extract patch

Run through a CNN

Classify center pixel

Repeat for every pixel
Semantic Segmentation

Run “fully convolutional” network to get all pixels at once

Smaller output due to pooling
Semantic Segmentation: Multi-Scale

Semantic Segmentation: Multi-Scale

Resize image to multiple scales


* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Semantic Segmentation: Multi-Scale

Semantic Segmentation: Multi-Scale

Resize image to multiple scales

Run one CNN per scale

Upscale outputs and concatenate

Semantic Segmentation: Multi-Scale

Resize image to multiple scales

Run one CNN per scale

Upscale outputs and concatenate

External “bottom-up” segmentation

Semantic Segmentation: Multi-Scale

Resize image to multiple scales
Run one CNN per scale
Upscale outputs and concatenate
Combine everything for final outputs

External “bottom-up” segmentation

Semantic Segmentation: Refinement

Apply CNN once to get labels

Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014
Semantic Segmentation: Refinement

Apply CNN once to get labels

Apply AGAIN to refine labels

Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014
Semantic Segmentation: Refinement

Apply CNN once to get labels

Apply AGAIN to refine labels

And again!

Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Semantic Segmentation: Refinement

Apply CNN once to get labels

Apply AGAIN to refine labels

And again!

Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014
Semantic Segmentation: Refinement

Apply CNN once to get labels

Apply AGAIN to refine labels

And again!

Same CNN weights: recurrent convolutional network

More iterations improve results

Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Semantic Segmentation: Upsampling


* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Semantic Segmentation: Upsampling

Learnable upsampling!


* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Semantic Segmentation: Upsampling


* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Semantic Segmentation: Upsampling

"skip connections"


* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Semantic Segmentation: Upsampling


* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Learnable Upsampling: “Deconvolution”

Typical 3 x 3 convolution, stride 1 pad 1

Input: 4 x 4

Output: 4 x 4
Learnable Upsampling: “Deconvolution”

Typical 3 x 3 convolution, stride 1 pad 1

Input: 4 x 4

Dot product between filter and input

Output: 4 x 4

* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Learnable Upsampling: “Deconvolution”

Typical 3 x 3 convolution, stride 1 pad 1

Input: 4 x 4

Dot product between filter and input

Output: 4 x 4
Learnable Upsampling: “Deconvolution”

Typical 3 x 3 convolution, \textit{stride} 2 pad 1

Input: 4 x 4

Output: 2 x 2
Learnable Upsampling: “Deconvolution”

Typical 3 x 3 convolution, stride 2 pad 1

Input: 4 x 4

Dot product between filter and input

Output: 2 x 2
Learnable Upsampling: “Deconvolution”

Typical 3 x 3 convolution, stride 2 pad 1

Input: 4 x 4

Output: 2 x 2

Dot product between filter and input
Learnable Upsampling: “Deconvolution”

3 x 3 “deconvolution”, stride 2 pad 1

Input: 2 x 2

Output: 4 x 4
Learnable Upsampling: “Deconvolution”

3 x 3 “deconvolution”, stride 2 pad 1

Input: 2 x 2

Input gives weight for filter

Output: 4 x 4
Learnable Upsampling: “Deconvolution”

3 x 3 “deconvolution”, stride 2 pad 1

Input: 2 x 2  
Output: 4 x 4

Input gives weight for filter
Learnable Upsampling: “Deconvolution”

3 x 3 “deconvolution”, stride 2 pad 1

Input gives weight for filter

Sum where output overlaps

Input: 2 x 2

Output: 4 x 4
Learnable Upsampling: “Deconvolution”

3 x 3 “deconvolution”, stride 2 pad 1

Input gives weight for filter

Input gives weight for filter

Sum where output overlaps

Same as backward pass for normal convolution!

Input: 2 x 2

Output: 4 x 4

* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Learnable Upsampling: "Deconvolution"

3 x 3 "deconvolution", stride 2 pad 1

Input gives weight for filter

Sum where output overlaps

Same as backward pass for normal convolution!

"Deconvolution" is a bad name, already defined as "inverse of convolution"

Better names: convolution transpose, backward strided convolution, 1/2 strided convolution, upconvolution
Learnable Upsampling: “Deconvolution”

1It is more proper to say “convolutional transpose operation” rather than “deconvolutional” operation. Hence, we will be using the term “convolutional transpose” from now.


“A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions)


“Deconvolution” is a bad name, already defined as “inverse of convolution”

Better names: convolution transpose, backward strided convolution, 1/2 strided convolution, upconvolution
Semantic Segmentation: Upsampling

Semantic Segmentation: Upsampling


6 days of training on Titan X…
Instance Segmentation
Instance Segmentation

Detect instances, give category, label pixels

“simultaneous detection and segmentation” (SDS)

Lots of recent work (MS-COCO)

Figure credit: Dai et al, "Instance-aware Semantic Segmentation via Multi-task Network Cascades", arXiv 2015

* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Instance Segmentation

Similar to R-CNN, but with segments


* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Instance Segmentation

Similar to R-CNN, but with segments

Proposal Generation

External Segment proposals


* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Instance Segmentation


Similar to R-CNN, but with segments
Instance Segmentation

Similar to R-CNN, but with segments

Instance Segmentation

Mask out background with mean image

Hariharan et al, "Simultaneous Detection and Segmentation", ECCV 2014

External Segment proposals

Proposal Generation

Feature Extraction

Region Classification

Person?

+1.8

Box CNN

Region CNN

* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Instance Segmentation

Instance Segmentation: Hypercolumns


* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Instance Segmentation: Cascades

Similar to
Faster R-CNN

Won COCO 2015 challenge
(with ResNet)

Instance Segmentation: Cascades

Similar to Faster R-CNN

Won COCO 2015 challenge (with ResNet)

Instance Segmentation: Cascades

Similar to Faster R-CNN

Region proposal network (RPN)

Won COCO 2015 challenge (with ResNet)

Instance Segmentation: Cascades

Similar to Faster R-CNN

Region proposal network (RPN)
Reshape boxes to fixed size, figure / ground logistic regression

Won COCO 2015 challenge (with ResNet)


* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Instance Segmentation: Cascades

Similar to Faster R-CNN

Region proposal network (RPN)

Reshape boxes to fixed size, figure / ground logistic regression

Mask out background, predict object class

Won COCO 2015 challenge (with ResNet)


* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Instance Segmentation: Cascades

Similar to Faster R-CNN

Region proposal network (RPN)

Reshape boxes to fixed size, figure / ground logistic regression

Learn entire model end-to-end!

Won COCO 2015 challenge (with ResNet)

Instance Segmentation: Cascades

Evaluation metrics

Detection Score: Average Precision (mAP is mean AP over all object categories)

- AP is averaged over multiple IoU values between 0.5 and 0.95 (and categories, size)
- More comprehensive comparison metric than the traditional AP at Intersection over Union (IoU) threshold of 0.5.
Evaluation Metrics

Detection Score: AP

- AP is averaged over multiple IoU values between 0.5 and 0.95
- AP is averaged over groups of objects or over object size in an image
- More comprehensive metric than the traditional AP at Intersection over Union (IoU) threshold of 0.5.
Evaluating Detection or Segmentation

To calculate AP we need:

- Bounding Box IoU
- Mask IoU
Segmentation Overview

- **Graph Cut**
  - Classic algorithm to quickly get foreground from background segmentation
  - Not trained to consider object shape prior
  - Can fail with complicated backgrounds

- **Unsupervised Segmentation**
  - Superpixels
  - Open area of research in ML

- **Semantic segmentation**
  - Classify all pixels
  - Fully convolutional models, downsample then upsample
  - Learnable upsampling: fractionally strided convolution
  - Skip connections can help

- **Instance Segmentation**
  - Detect instance, generate mask
  - Similar pipelines to object detection

* Original slides borrowed from Andrej Karpathy and Li Fei-Fei, Stanford cs231n
Pose Estimation
(aka Keypoints)
Classic Problem: Activity Recognition

What is this person doing?
Human pose estimation & Object detection

Human pose estimation is challenging.

- Difficult part appearance
- Self-occlusion
- Image region looks like a body part

- Felzenszwalb & Huttenlocher, 2005
- Ren et al, 2005
- Ramanan, 2006
- Ferrari et al, 2008
- Yang & Mori, 2008
- Andriluka et al, 2009
- Eichner & Ferrari, 2009
Human pose estimation is challenging.

- Felzenszwalb & Huttenlocher, 2005
- Ren et al, 2005
- Ramanan, 2006
- Ferrari et al, 2008
- Yang & Mori, 2008
- Andriluka et al, 2009
- Eichner & Ferrari, 2009
Human pose estimation & Object detection

Facilitate

Given the object is detected.

Slide Credit: Yao/Fei-Fei
Human pose estimation & Object detection

Small, low-resolution, partially occluded

Image region similar to detection target

Object detection is challenging

- Viola & Jones, 2001
- Lampert et al, 2008
- Divvala et al, 2009
- Vedaldi et al, 2009

Slide Credit: Yao/Fei-Fei
Human pose estimation & Object detection

Object detection is challenging

- Viola & Jones, 2001
- Lampert et al, 2008
- Divvala et al, 2009
- Vedaldi et al, 2009
Human pose estimation & Object detection

Facilitate

Given the pose is estimated.

Slide Credit: Yao/Fei-Fei
Human pose estimation & Object detection

Mutual Context
Multi-Person Pose Estimation using Part Affinity Fields

Zhe Cao, Shih-En Wei, Tomas Simon, Yaser Sheikh
Carnegie Mellon University
Top-down Approach: Person Detection + Pose Estimation

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models
Top-down Approach: Person Detection + Pose Estimation

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models
Top-down Approach: Person Detection + Pose Estimation

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models”
Top-down Approach: Person Detection + Pose Estimation
Top-down Approach: Person Detection + Pose Estimation

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models"
Our Method: Parts Detection + Parts Association

Top-down

Ours

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models"
Our Method: Parts Detection + Parts Association

Top-down

Ours

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models
Our Method: Parts Detection + Parts Association

Top-down

Ours

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models"
Our Method: Parts Detection + Parts Association

* Slide: Zhe Cao: "Multi-Person Pose Estimation Using Part Affinity Models"
Our Method: Parts Detection + Parts Association

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models"
Our Method: Parts Detection + Parts Association

Top-down

Part Affinity Fields

* Slide: Zhe Cao: "Multi-Person Pose Estimation Using Part Affinity Models
Our Method: Parts Detection + Parts Association

Top-down

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models”
Our Method: Parts Detection + Parts Association

Top-down

Part Affinity Fields

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models
Our Method: Parts Detection + Parts Association

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models"
Novelty: Jointly Learning Parts Detection and Parts Association

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models"
Sequential Prediction with Learned Spatial Context

Stage 1

Right shoulder

Right wrist

Right knee

Convolutional Pose Machines, Wei, Ramakrishna, Kanade, Sheikh, CVPR 2016
Sequential Prediction with Learned Spatial Context

Stage 1

Convolutional Pose Machines, Wei, Ramakrishna, Kanade, Sheikh, CVPR 2016

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models
Sequential Prediction with Learned Spatial Context

Stage 1

Right Wrist - Stage 1

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models
Sequential Prediction with Learned Spatial Context

Stage 1

CNN

Stage 2

CNN

Right Wrist - Stage 1

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models"
Sequential Prediction with Learned Spatial Context

Stage 1

Stage 2

Right Wrist - Stage 1

Right Wrist - Stage 2

* Slide: Zhe Cao: "Multi-Person Pose Estimation Using Part Affinity Models"
Sequential Prediction with Learned Spatial Context

Stage 1

Stage t = 1

Stage t = 2

Stage t = 3

Input Image

CNN

Stage 2

Stage t = 1

Stage t = 2

Stage t = 3

Input Image

CNN

……

Stage T

Input Image

CNN

……

Right Wrist - Stage 1

Right Wrist - Stage 2

Right Wrist - Stage T

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models"
Part-Person Association for Multi-Person Pose Estimation

* Slide: Zhe Cao: "Multi-Person Pose Estimation Using Part Affinity Models"
Part-Person Association for Multi-Person Pose Estimation

- Part detections
- Pose

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models”
Part-to-Part Association for Multi-Person Pose Estimation

*Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models*
Part Affinity Score Guides the Connection

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models
Part Affinity Score Guides the Connection

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models
Part Affinity Score Guides the Connection

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models
How to Obtain the Part Affinity Score

* Slide: Zhe Cao: "Multi-Person Pose Estimation Using Part Affinity Models"
Part Affinity Score is Dependent on Visual Appearance

* Slide: Zhe Cao: "Multi-Person Pose Estimation Using Part Affinity Models"
Part Affinity Score is Dependent on Visual Appearance
Key Idea: Encode the Part Affinity Score on the Image Plane

Part Affinity Fields encode **direction** and **position**

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models"
Part Affinity Fields Avoid Spatial Ambiguity

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models
Jointly Learning Parts Detection and Parts Association

Stage 1

Stage 2

Stage $T$

CNN

CNN

CNN

Input Image

Jointly Learning Parts Detection and Parts Association

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models"
Jointly Learning Parts Detection and Parts Association

Stage 1

2nd Branch
Part Affinity
Fields

Stage 2

Stage T

CNN

P

CNN

P

CNN

P

CNN

P

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models"
* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models
* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models"
Greedy Algorithm for Body Parts Association

* Slide: Zhe Cao: "Multi-Person Pose Estimation Using Part Affinity Models"
Greedy Algorithm for Body Parts Association

*Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models*
Greedy Algorithm for Body Parts Association

* Slide: Zhe Cao: "Multi-Person Pose Estimation Using Part Affinity Models"
Greedy Algorithm for Body Parts Association

* Slide: Zhe Cao: “Multi-Person Pose Estimation Using Part Affinity Models"
* Slide: Zhe Cao: "Multi-Person Pose Estimation Using Part Affinity Models"
Results on COCO Challenge Validation Set

<table>
<thead>
<tr>
<th>Method</th>
<th>AP on val</th>
</tr>
</thead>
<tbody>
<tr>
<td>GT bbox + CPM [1]</td>
<td>63</td>
</tr>
<tr>
<td>Our Method</td>
<td>58.5</td>
</tr>
<tr>
<td>Ours + Refinement</td>
<td>61</td>
</tr>
</tbody>
</table>

* Slide: Zhe Cao: "Multi-Person Pose Estimation Using Part Affinity Models

https://youtu.be/pW6nZXeWlGM
Evaluation: Keypoints

To calculate AP we need:

Bounding Box IoU

Mask IoU

Object Keypoint Similarity
Object Keypoint Similarity - OKS